

Библиотека функциональных блоков

Содержание

1. Условные обозначения и термины.....	13
1.1. Условные обозначения.....	13
1.2. Перечень терминов и сокращений.....	13
2. Введение.....	17
3. Встроенные функциональные блоки.....	19
3.1. Функциональный блок System.....	19
3.1.1. Атрибуты System.....	19
3.1.2. Режимы System.....	20
4. Функциональные блоки для создания удаленного соединения с контроллером.....	21
4.1. Функциональный блок RemoteConnection.....	22
4.1.1. Атрибуты RemoteConnection.....	22
4.1.2. Режимы RemoteConnection.....	22
4.1.3. Алгоритм RemoteConnection.....	22
4.2. Функциональный блок RemoteRealPoint.....	23
4.2.1. Атрибуты RemoteRealPoint.....	23
4.2.2. Режимы RemoteRealPoint.....	24
4.2.3. Алгоритм RemoteRealPoint.....	24
4.2.3.1. Чтение данных из контроллера-источника.....	25
4.2.3.2. Запись.....	25
4.2.3.3. Диагностика.....	25
4.3. Функциональный блок RemoteBooleanPoint.....	26
4.3.1. Атрибуты RemoteBooleanPoint.....	26
4.3.2. Режимы RemoteBooleanPoint.....	27
4.3.3. Алгоритм RemoteBooleanPoint.....	27
4.3.3.1. Чтение данных из контроллера-источника.....	28
4.3.3.2. Запись.....	28
4.3.3.3. Диагностика.....	28
4.4. Функциональный блок RemoteIntegerPoint.....	29
4.4.1. Атрибуты RemoteIntegerPoint.....	29
4.4.2. Режимы RemoteIntegerPoint.....	30
4.4.3. Алгоритм RemoteIntegerPoint.....	30
4.4.3.1. Чтение данных из контроллера-источника.....	31
4.4.3.2. Запись.....	31

4.4.3.3. Диагностика.....	31
5. Функциональные блоки для настройки ввода/вывода.....	32
5.1. Корзина MKRack.....	32
5.1.1. Атрибуты MKRack.....	32
5.1.2. Режимы MKRack.....	33
5.2. Модуль питания MK550024PSU.....	33
5.2.1. Атрибуты MK550024PSU.....	33
5.2.2. Режимы MK550024PSU.....	34
5.2.3. Алгоритм MK550024PSU.....	34
5.2.3.1. Коммуникация.....	35
5.2.3.2. Обработка данных.....	35
5.2.3.3. Формирование сигнализаций.....	35
5.3. Модуль ЦПУ MK502142CPU.....	36
5.3.1. Атрибуты MK502142CPU.....	36
5.3.2. Режимы MK502142CPU.....	37
5.3.3. Алгоритм MK502142CPU.....	37
5.3.3.1. Самоидентификация.....	38
5.3.3.2. Запуск.....	38
5.3.3.3. Диагностика.....	39
5.3.3.4. Формирование сигнализаций.....	40
5.4. Модули аналогового ввода MK513016AI и MK513016AAI.....	40
5.4.1. Атрибуты MK513016AI.....	40
5.4.2. Режимы MK513016AI.....	42
5.4.3. Алгоритм MK513016AI.....	42
5.4.3.1. Диагностика.....	43
5.4.3.2. Формирование сигнализаций.....	43
5.4.3.3. Преобразование и запись.....	44
5.5. Модуль аналогового ввода MK516008AAI.....	44
5.5.1. Атрибуты MK516008AAI.....	44
5.5.2. Режимы MK516008AAI.....	45
5.5.3. Алгоритм MK516008AAI.....	46
5.5.3.1. Диагностика.....	46
5.5.3.2. Формирование сигнализаций.....	47
5.5.3.3. Преобразование и запись.....	47
5.6. Модули дискретного ввода MK521032DI и MK521032ADI.....	47
5.6.1. Атрибуты MK521032DI.....	48
5.6.2. Режимы MK521032DI.....	49
5.6.3. Алгоритм MK521032DI.....	49
5.6.3.1. Диагностика.....	50
5.6.3.2. Формирование сигнализаций.....	50
5.6.3.3. Запись данных модуля ввода/вывода.....	51
5.7. Модуль дискретного ввода MK523032ADI.....	51

5.7.1. Атрибуты МК523032ADI.....	51
5.7.2. Режимы МК523032ADI.....	53
5.7.3. Алгоритм МК523032ADI.....	53
5.7.3.1. Диагностика.....	55
5.7.3.2. Формирование сигнализаций.....	55
5.7.3.3. Запись данных модуля ввода/вывода.....	55
5.7.3.4. Поддержка сигналов NAMUR.....	56
5.7.3.5. Работа в составе резервной пары.....	56
5.8. Модули аналогового вывода МК514008АО и МК514008ААО.....	57
5.8.1. Атрибуты МК514008АО.....	58
5.8.2. Режимы МК514008АО.....	59
5.8.3. Алгоритм МК514008АО.....	60
5.8.3.1. Диагностика.....	60
5.8.3.2. Формирование сигнализаций.....	61
5.8.3.3. Преобразование.....	61
5.8.3.4. Запись данных в модуль ввода/вывода.....	61
5.9. Модули дискретного вывода МК531032DO и МК531032ADO.....	61
5.9.1. Атрибуты МК531032DO.....	62
5.9.2. Режимы МК531032DO.....	63
5.9.3. Алгоритм МК531032DO.....	63
5.9.3.1. Диагностика.....	64
5.9.3.2. Формирование сигнализаций.....	65
5.9.3.3. Запись данных в модуль ввода/вывода.....	65
5.10. Модуль дискретного ввода МК532032ADO.....	65
5.10.1. Атрибуты МК532032ADO.....	65
5.10.2. Режимы МК532032ADO.....	67
5.10.3. Алгоритм МК532032ADO.....	68
5.10.3.1. Диагностика.....	68
5.10.3.2. Формирование сигнализаций.....	69
5.10.3.3. Запись данных в модуль ввода/вывода.....	69
5.10.3.4. Поддержка сигналов NAMUR.....	69
5.10.3.5. Работа в составе резервной пары.....	70
5.11. Модуль аналогового ввода МК576008ААІHART.....	71
5.11.1. Атрибуты МК576008ААІHART.....	71
5.11.2. Режимы МК576008ААІHART.....	73
5.11.3. Алгоритм МК576008ААІHART.....	73
5.11.3.1. Диагностика.....	74
5.11.3.2. Формирование сигнализаций.....	75
5.11.3.3. Преобразование и запись.....	75
5.11.3.4. Работа HART модулей в составе PDM систем.....	75
5.12. Модуль аналогового ввода МК576016ААІHART.....	76
5.12.1. Атрибуты МК576016ААІHART.....	76
5.12.2. Режимы МК576016ААІHART.....	79

5.12.3.	Алгоритм МК576016ААИНАРТ.....	79
5.12.3.1.	Диагностика.....	81
5.12.3.2.	Формирование сигнализаций.....	81
5.12.3.3.	Преобразование и запись.....	82
5.12.3.4.	Работа НАРТ в режиме резервирования.....	82
5.13.	Модуль аналогового вывода МК574008ААОНАРТ.....	84
5.13.1.	Атрибуты МК574008ААОНАРТ.....	84
5.13.2.	Режимы МК574008ААОНАРТ.....	86
5.13.3.	Алгоритм МК574008ААОНАРТ.....	87
5.13.3.1.	Диагностика.....	89
5.13.3.2.	Формирование сигнализаций.....	89
5.13.3.3.	Преобразование.....	89
5.13.3.4.	Обработка данных в модуле ввода/вывода.....	90
5.13.3.5.	Работа НАРТ в режиме резервирования.....	90
5.14.	Модуль коммуникации МК541002 с поддержкой Modbus.....	91
5.14.1.	Атрибуты МК541002.....	91
5.14.2.	Режимы МК541002.....	93
5.14.3.	Алгоритм МК541002.....	93
5.14.3.1.	Конфигурация.....	93
5.14.3.2.	Коммуникация.....	94
5.14.3.3.	Диагностика.....	94
5.14.3.4.	Формирование сигнализаций.....	95
6.	Функциональные блоки коммуникации по ModBus.....	96
6.1.	Устройство ModBusTCPDevice.....	98
6.1.1.	Атрибуты ModBusTCPDevice.....	98
6.1.2.	Режимы ModBusTCPDevice.....	99
6.1.3.	Алгоритм ModBusTCPDevice.....	99
6.1.3.1.	Коммуникация с устройством ModBusTCP.....	99
6.1.3.2.	Диагностика коммуникации.....	99
6.2.	Устройство ModBusRTUDevice.....	100
6.2.1.	Атрибуты ModBusRTUDevice.....	100
6.2.2.	Режимы ModBusRTUDevice.....	101
6.2.3.	Алгоритм ModBusRTUDevice.....	101
6.2.3.1.	Коммуникация с устройством Modbus.....	101
6.2.3.2.	Диагностика коммуникации.....	101
6.3.	Карта ModBusRealInput32.....	102
6.3.1.	Атрибуты ModBusRealInput32.....	102
6.3.2.	Режимы ModBusRealInput32.....	103
6.3.3.	Алгоритм ModBusRealInput32.....	103
6.3.3.1.	Чтение данных устройства Modbus.....	104
6.3.3.2.	Диагностика коммуникации.....	104
6.4.	Карта ModBusRealOutput32.....	105

6.4.1.	Атрибуты ModBusRealOutput32.....	105
6.4.2.	Режимы ModBusRealOutput32.....	106
6.4.3.	Алгоритм ModBusRealOutput32.....	107
6.4.3.1.	Запись данных в устройство Modbus.....	107
6.4.3.2.	Диагностика коммуникации.....	108
6.5.	Карта ModBusIntegerInput32.....	108
6.5.1.	Атрибуты ModBusIntegerInput32.....	108
6.5.2.	Режимы ModBusIntegerInput32.....	109
6.5.3.	Алгоритм ModBusIntegerInput32.....	109
6.5.3.1.	Чтение данных устройства Modbus.....	110
6.5.3.2.	Диагностика коммуникации.....	110
6.6.	Карта ModBusIntegerOutput32.....	111
6.6.1.	Атрибуты ModBusIntegerOutput32.....	111
6.6.2.	Режимы ModBusIntegerOutput32.....	112
6.6.3.	Алгоритм ModBusIntegerOutput32.....	113
6.6.3.1.	Запись данных в Modbus устройство.....	113
6.6.3.2.	Диагностика коммуникации.....	114
6.7.	Карта ModBusUIntegerInput32.....	114
6.7.1.	Атрибуты ModBusUIntegerInput32.....	114
6.7.2.	Режимы ModBusUIntegerInput32.....	115
6.7.3.	Алгоритм ModBusUIntegerInput32.....	116
6.7.3.1.	Чтение данных устройства Modbus.....	116
6.7.3.2.	Диагностика коммуникации.....	117
6.8.	Карта ModBusUIntegerOutput32.....	117
6.8.1.	Атрибуты ModBusUIntegerOutput32.....	117
6.8.2.	Режимы ModBusUIntegerOutput32.....	118
6.8.3.	Алгоритм ModBusUIntegerOutput32.....	119
6.8.3.1.	Запись данных в Modbus устройство.....	119
6.8.3.2.	Диагностика коммуникации.....	120
6.9.	Карта ModBusBooleanInput32.....	120
6.9.1.	Атрибуты ModBusBooleanInput32.....	120
6.9.2.	Режимы ModBusBooleanInput32.....	121
6.9.3.	Алгоритм ModBusBooleanInput32.....	122
6.9.3.1.	Чтение данных устройства Modbus.....	122
6.9.3.2.	Диагностика коммуникации.....	123
6.10.	Карта ModBusBooleanOutput32.....	123
6.10.1.	Атрибуты ModBusBooleanOutput32.....	123
6.10.2.	Режимы ModBusBooleanOutput32.....	124
6.10.3.	Алгоритм ModBusBooleanOutput32.....	125
6.10.3.1.	Запись данных в Modbus устройство.....	125
6.10.3.2.	Диагностика коммуникации.....	126
6.11.	Карта ModBusShortInput32.....	126
6.11.1.	Атрибуты ModBusShortInput32.....	126

6.11.2. Режимы ModBusShortInput32.....	127
6.11.3. Алгоритм ModBusShortInput32.....	127
6.11.3.1. Чтение данных устройства Modbus.....	128
6.11.3.2. Диагностика коммуникации.....	128
6.12. Карта ModBusShortOutput32.....	129
6.12.1. Атрибуты ModBusShortOutput32.....	129
6.12.2. Режимы ModBusShortOutput32.....	130
6.12.3. Алгоритм ModBusShortOutput32.....	131
6.12.3.1. Запись данных в Modbus устройство.....	131
6.12.3.2. Диагностика коммуникации.....	132
6.13. Карта ModBusUshortInput32.....	132
6.13.1. Атрибуты ModBusUshortInput32.....	132
6.13.2. Режимы ModBusUshortInput32.....	133
6.13.3. Алгоритм ModBusUshortInput32.....	133
6.13.3.1. Чтение данных устройства Modbus.....	134
6.13.3.2. Диагностика коммуникации.....	134
6.14. Карта ModBusUshortOutput32.....	135
6.14.1. Атрибуты ModBusUshortOutput32.....	135
6.14.2. Режимы ModBusUshortOutput32.....	136
6.14.3. Алгоритм ModBusUshortOutput32.....	137
6.14.3.1. Запись данных в Modbus устройство.....	137
6.14.3.2. Диагностика коммуникации.....	138
7. Технологические функциональные блоки.....	139
7.1. Функциональный блок Container16int.....	139
7.1.1. Атрибуты Container16int.....	139
7.1.2. Режимы Container16int.....	139
7.2. Функциональный блок Container16uInt.....	140
7.2.1. Атрибуты Container16uInt.....	140
7.2.2. Режимы Container16uInt.....	140
7.3. Функциональный блок Container32int.....	140
7.3.1. Атрибуты Container32int.....	140
7.3.2. Режимы Container32int.....	141
7.4. Функциональный блок Container32uInt.....	141
7.4.1. Атрибуты Container32uInt.....	141
7.4.2. Режимы Container32uInt.....	142
7.5. Функциональный блок ContainerBool.....	142
7.5.1. Атрибуты ContainerBool.....	142
7.5.2. Режимы ContainerBool.....	143
7.6. Функциональный блок ContainerReal.....	143
7.6.1. Атрибуты ContainerReal.....	143
7.6.2. Режимы ContainerReal.....	144
7.7. Функциональный блок ControlButton.....	144

7.7.1. Атрибуты ControlButton.....	144
7.7.2. Режимы ControlButton.....	145
7.7.3. Алгоритм ControlButton.....	145
7.8. Входной аналоговый блок AnalogInputPoint.....	145
7.8.1. Атрибуты AnalogInputPoint.....	145
7.8.2. Режимы AnalogInputPoint.....	149
7.8.3. Алгоритм AnalogInputPoint.....	149
7.8.3.1. Контроль достоверности.....	150
7.8.3.2. Масштабирование.....	151
7.8.3.3. Контроль ручного значения.....	151
7.8.3.4. Формирование сигнализаций.....	151
7.8.3.5. Фильтрация шума сигнала.....	151
7.9. Входной аналоговый блок AnalogInputPointExtended.....	152
7.9.1. Атрибуты AnalogInputPointExtended.....	152
7.9.2. Режимы AnalogInputPointExtended.....	158
7.9.3. Алгоритм AnalogInputPointExtended.....	158
7.9.3.1. Контроль достоверности.....	158
7.9.3.2. Масштабирование.....	159
7.9.3.3. Контроль ручного значения.....	159
7.9.3.4. Формирование сигнализаций.....	160
7.10. Выходной аналоговый блок AnalogOutputPoint.....	160
7.10.1. Атрибуты AnalogOutputPoint.....	160
7.10.2. Режимы AnalogOutputPoint.....	161
7.10.3. Алгоритм AnalogOutputPoint.....	161
7.10.3.1. Контроль достоверности.....	162
7.10.3.2. Масштабирование.....	162
7.10.3.3. Диагностика состояния физического канала.....	163
7.11. Дискретный блок DiscretePoint.....	164
7.11.1. Атрибуты DiscretePoint.....	164
7.11.2. Режимы DiscretePoint.....	165
7.11.3. Алгоритм DiscretePoint.....	165
7.11.3.1. Инверсия.....	166
7.11.3.2. Фильтрация дребезга.....	166
7.11.3.3. Формирование сигнализаций.....	166
7.12. ПИД регулятор PIDController.....	167
7.12.1. Атрибуты PIDController.....	167
7.12.2. Режимы PIDController.....	174
7.12.3. Алгоритм PIDController.....	175
7.12.3.1. Регулирование по ПИД-закону.....	178
7.12.3.2. Контроль выходного значения.....	181
7.12.3.3. Фильтрация шума сигнала.....	181
7.13. Функциональный блок Motor.....	182
7.13.1. Атрибуты Motor.....	182

7.13.2. Режимы Motor.....	185
7.13.3. Алгоритм Motor.....	186
7.13.3.1. Диагностика и управление.....	186
7.13.3.2. Формирование сигнализаций.....	189
7.14. Функциональный блок Program.....	191
7.14.1. Атрибуты Program.....	191
7.14.2. Режимы Program.....	192
7.14.3. Алгоритм блока Program.....	192
7.14.3.1. Чтение.....	193
7.14.3.2. Интерпретация и запись.....	193
7.14.3.3. Диагностика.....	194
7.15. Резервируемый дискретный функциональный блок RedundantDiscretePoint.....	194
7.15.1. Атрибуты RedundantDiscretePoint.....	194
7.15.2. Режимы RedundantDiscretePoint.....	196
7.15.3. Алгоритм RedundantDiscretePoint.....	196
7.15.3.1. Диагностика и резервирование.....	197
7.15.3.2. Обработка данных.....	197
7.15.3.3. Формирование сигнализаций.....	198
7.16. Резервируемый входной аналоговый функциональный блок RedundantAnalogInputPoint.....	198
7.16.1. Атрибуты RedundantAnalogInputPoint.....	198
7.16.2. Режимы RedundantAnalogInputPoint.....	202
7.16.3. Алгоритм RedundantAnalogInputPoint.....	203
7.16.3.1. Диагностика и резервирование.....	203
7.16.3.2. Масштабирование и запись данных.....	204
7.16.3.3. Контроль ручного значения.....	204
7.16.3.4. Формирование сигнализаций.....	205
7.16.3.5. Фильтрация шума сигнала.....	205
7.17. Функциональный блок Valve.....	205
7.17.1. Атрибуты Valve.....	205
7.17.2. Режимы Valve.....	209
7.17.3. Алгоритм Valve.....	209
7.17.3.1. Диагностика и управление Valve.....	210
7.17.3.2. Формирование сигнализаций.....	213
7.18. Функциональный блок ValveUnstoppable.....	214
7.18.1. Атрибуты ValveUnstoppable.....	214
7.18.2. Режимы ValveUnstoppable.....	217
7.18.3. Алгоритм ValveUnstoppable.....	218
7.18.3.1. Диагностика и управление ValveUnstoppable.....	218
7.18.3.2. Формирование сигнализаций.....	221
7.19. Функциональный блок LogicalBlock.....	222
7.19.1. Атрибуты LogicalBlock.....	222

7.19.2. Режимы LogicalBlock.....	222
7.19.3. Алгоритм LogicalBlock.....	223
7.19.3.1. Чтение.....	223
7.19.3.2. Обработка данных.....	223
7.19.3.3. Диагностика.....	224
7.20. Функциональный блок MinMaxBlock.....	225
7.20.1. Атрибуты MinMaxBlock.....	225
7.20.2. Режимы MinMaxBlock.....	226
7.20.3. Алгоритм MinMaxBlock.....	226
7.20.3.1. Чтение.....	227
7.20.3.2. Обработка данных.....	227
7.20.3.3. Диагностика.....	228
7.21. Функциональный блок AccumulatorBlock.....	229
7.21.1. Атрибуты AccumulatorBlock.....	229
7.21.2. Режимы AccumulatorBlock.....	230
7.21.3. Алгоритм AccumulatorBlock.....	230
7.21.3.1. Чтение.....	230
7.21.3.2. Обработка данных.....	231
7.21.3.3. Диагностика.....	231
7.22. Резервируемый ПИД регулятор RedundantPIDController.....	231
7.22.1. Атрибуты RedundantPIDController.....	232
7.22.2. Режимы RedundantPIDController.....	239
7.22.3. Алгоритм RedundantPIDController.....	241
7.22.3.1. Диагностика и резервирование.....	242
7.22.3.2. Масштабирование и запись данных.....	243
7.22.3.3. Контроль ручного значения.....	243
7.22.3.4. Формирование сигнализаций.....	243
7.22.3.5. Регулирование по ПИД-закону.....	243
7.22.3.6. Контроль выходного значения.....	246
7.22.3.7. Фильтрация шума сигнала.....	246
7.23. Функциональный блок DiscreteInputsPacker.....	247
7.23.1. Атрибуты DiscreteInputsPacker.....	247
7.23.2. Режимы DiscreteInputsPacker.....	247
7.23.3. Алгоритм DiscreteInputsPacker.....	248
7.23.3.1. Чтение.....	248
7.23.3.2. Обработка данных.....	248
7.24. Функциональный блок DigitalInputUnpacker.....	248
7.24.1. Атрибуты DigitalInputUnpacker.....	249
7.24.2. Режимы DigitalInputUnpacker.....	249
7.24.3. Алгоритм DigitalInputUnpacker.....	250
7.24.3.1. Чтение.....	250
7.24.3.2. Обработка данных.....	250
7.25. Функциональный блок TimeDelay.....	251

7.25.1. Атрибуты TimeDelay.....	251
7.25.2. Режимы TimeDelay.....	251
7.25.3. Алгоритм TimeDelay.....	252
7.25.3.1. Чтение.....	252
7.25.3.2. Обработка данных.....	252
7.26. Функциональный блок PulseDuration.....	253
7.26.1. Атрибуты PulseDuration.....	253
7.26.2. Режимы PulseDuration.....	254
7.26.3. Алгоритм PulseDuration.....	254
7.26.3.1. Чтение.....	255
7.26.3.2. Обработка данных.....	255
7.27. Функциональный блок RSSRTrigger.....	256
7.27.1. Атрибуты RSSRTrigger.....	256
7.27.2. Режимы RSSRTrigger.....	257
7.27.3. Алгоритм RSSRTrigger.....	257
7.27.3.1. Чтение.....	258
7.27.3.2. Обработка данных.....	258
7.28. Функциональный блок ToWords.....	260
7.28.1. Атрибуты ToWords.....	260
7.28.2. Режимы ToWords.....	261
7.28.3. Алгоритм ToWords.....	261
7.29. Функциональный блок WordsTo.....	262
7.29.1. Атрибуты WordsTo.....	262
7.29.2. Режимы WordsTo.....	263
7.29.3. Алгоритм WordsTo.....	263
7.30. Функциональный блок NotBlock.....	264
7.30.1. Атрибуты NotBlock.....	264
7.30.2. Режимы NotBlock.....	264
7.30.3. Алгоритм NotBlock.....	265
7.31. Функциональный блок ChangeAnalyzer.....	265
7.31.1. Атрибуты ChangeAnalyzer.....	265
7.31.2. Режимы ChangeAnalyzer.....	265
7.31.3. Алгоритм ChangeAnalyzer.....	265
7.32. Функциональный блок DiscreteControl.....	266
7.32.1. Атрибуты DiscreteControl.....	266
7.32.2. Режимы DiscreteControl.....	267
7.32.3. Алгоритм DiscreteControl.....	267
7.33. Функциональный блок TwoPositionController.....	269
7.33.1. Атрибуты TwoPositionController.....	269
7.33.2. Режимы TwoPositionController.....	270
7.33.3. Алгоритм TwoPositionController.....	270

8. Функциональные блоки противоаварийной защиты..... 272

8.1. Функциональный блок DiscreteVoter4/16.....	272
8.1.1. Атрибуты DiscreteVoter4/16.....	272
8.1.2. Алгоритм DiscreteVoter4/16.....	276
8.2. Функциональный блок CauseEffectMatrix16.....	279
8.2.1. Атрибуты CauseEffectMatrix16.....	279
8.2.2. Режимы CauseEffectMatrix16.....	282
8.2.3. Алгоритм CauseEffectMatrix16.....	283
8.2.3.1. Работа канала в состоянии "Защита".....	284
8.2.3.2. Работа канала в состоянии "Норма".....	285
8.2.3.3. Блокировка переходов каналов подсистемы защиты....	285
9. Настройка видимости атрибутов в редакторе диаграмм.....	286
10. Приложение. Базовый функционал.....	288
10.1. Формирование сигнализаций.....	288

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

Атрибут¹ функционального блока - одно из значений, характеризующих функциональный блок, позволяющее задавать его свойства².

Контроллер РСУ

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

¹ 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".

² 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 оператора³.

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.

³ "Руководство по созданию технологического программного обеспечения станции оператора" п. 3.1 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.

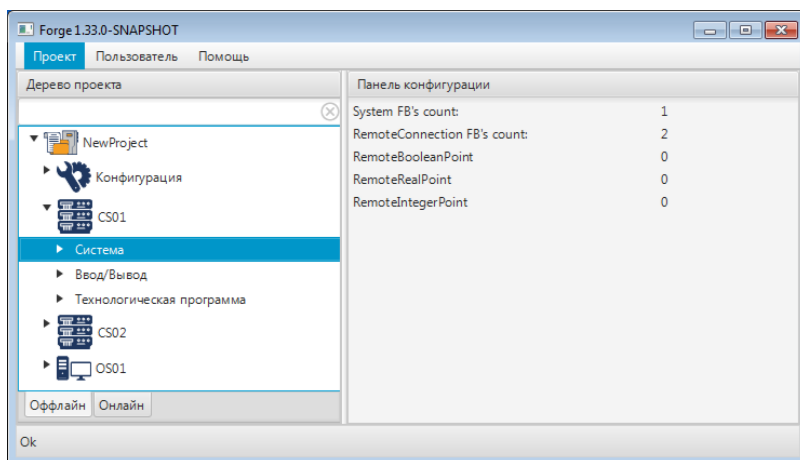


Рисунок 1. Счетчик функциональных блоков

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 РСУ.

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 времени⁴.

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

⁴ "Руководство по инсталляции", Настройка синхронизации внутреннего времени узлов РСУ.

Имя атрибута	Тип данных	Доступ	Описание атрибута
RawTime	DATE_AND_TIME	чтение/ запись	Текущее время
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 TC PCY
Конфигурационные атрибуты			
TimeServerIP	STRING	чтение/ запись	IP сервера времени
Сигнализационные атрибуты			
VER_NMATCH	BOOL	чтение	Версии не совпадают

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.

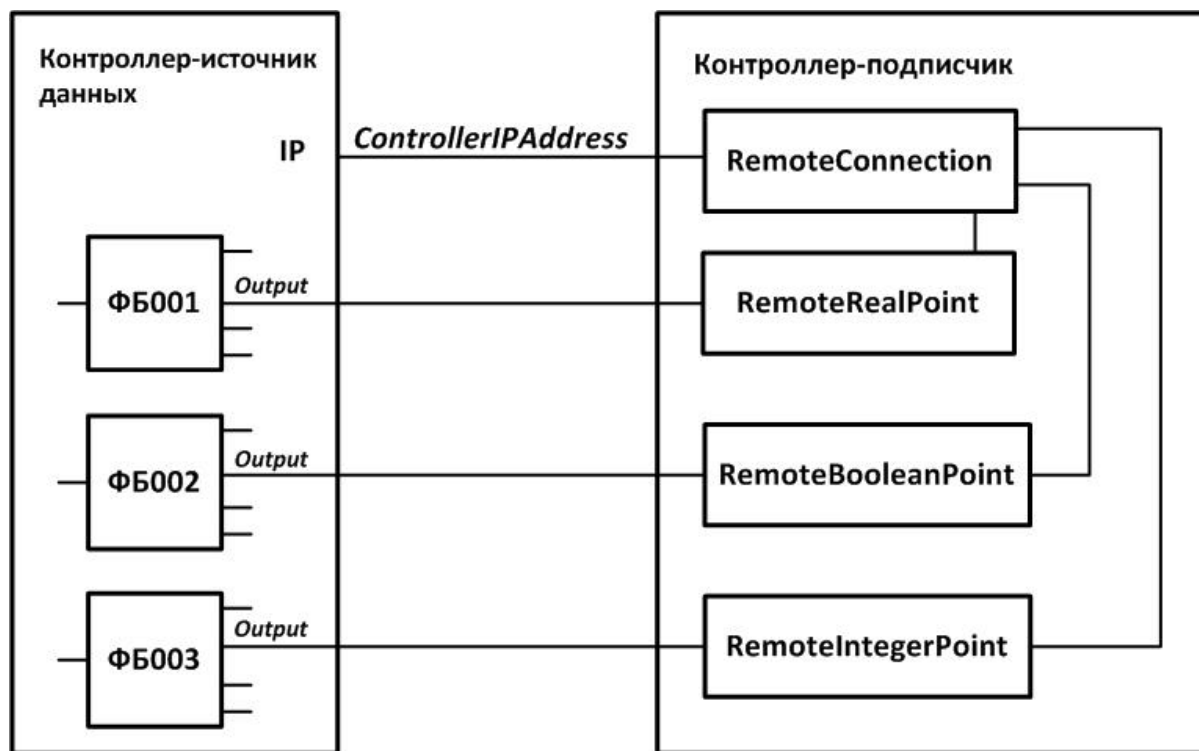


Рисунок 2. Межконтроллерное соединение с контроллером-источником данных

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>			
CTRL_IP	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 **CTRL_IP** 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>			
IN	REAL	чтение	Вход
OUT	REAL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
MV	REAL	чтение/ запись	Ручное значение
REM_CONN	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 Status 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.

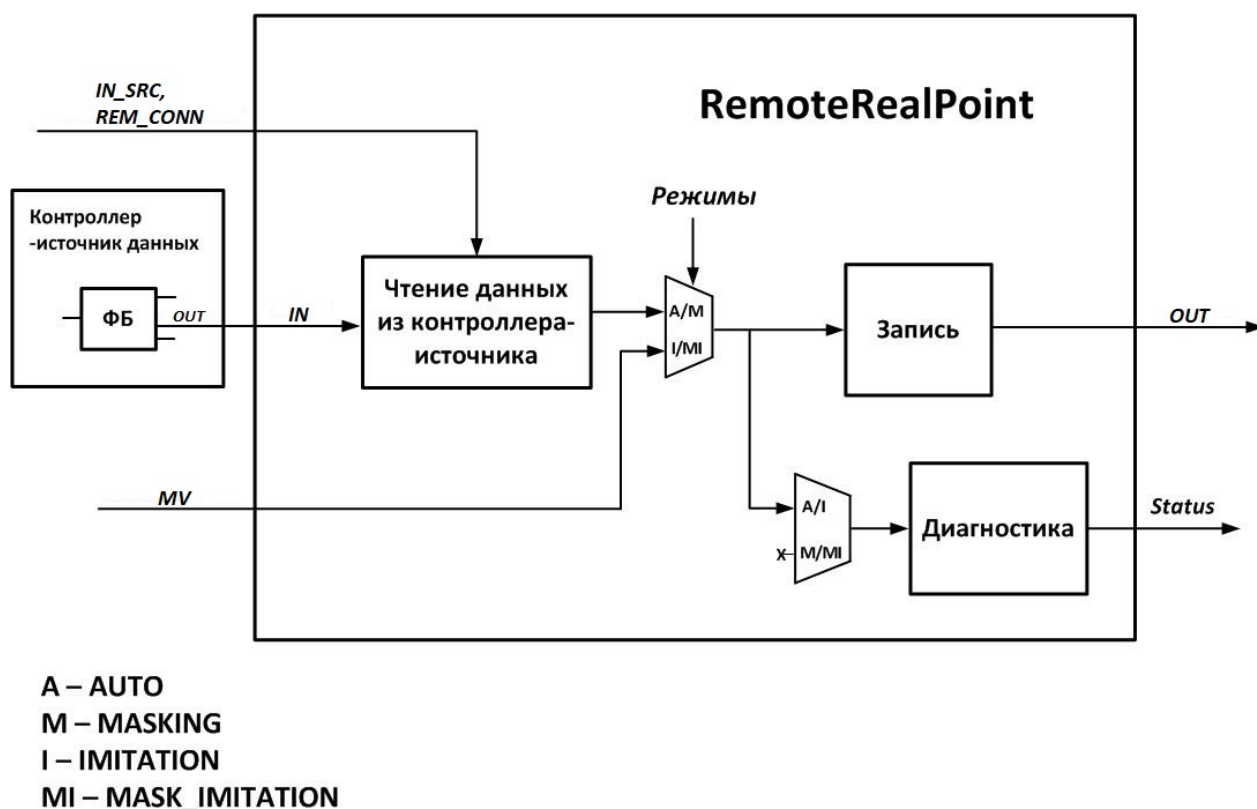


Рисунок 3. Схема процесса обработки данных RemoteRealPoint

4.2.3.1. Чтение данных из контроллера-источника

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

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

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

В режимах IMITATION и MASK_IMITATION входные данные вводятся вручную и записываются в атрибут **MV**.

4.2.3.2. Запись

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

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
Входные/выходные атрибуты			
IN	BOOL	чтение	Вход
OUT	BOOL	чтение	Выход
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник
MV	BOOL	чтение/ запись	Ручное значение
REM_CONN	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 Status 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.

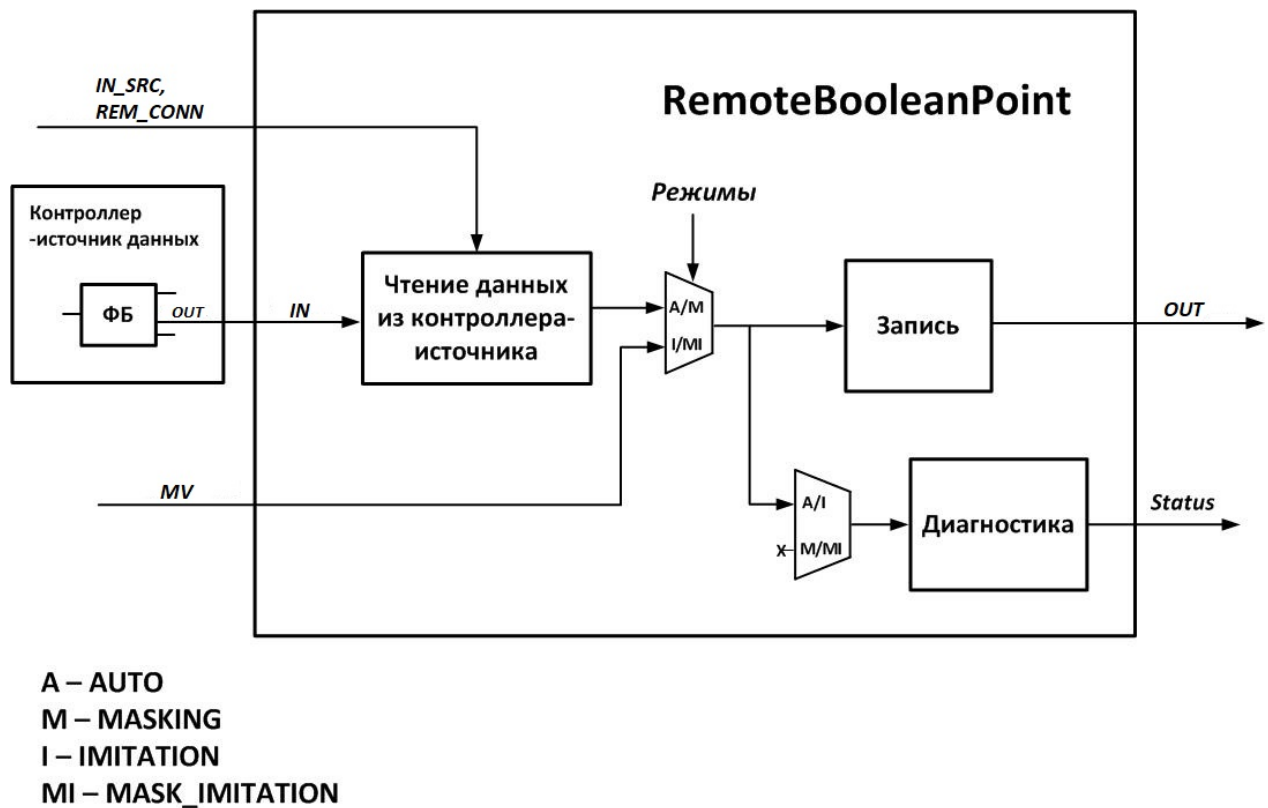


Рисунок 4. Схема процесса обработки данных RemoteBooleanPoint

4.3.3.1. Чтение данных из контроллера-источника

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

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

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

В режимах IMITATION и MASK_IMITATION входные данные вводятся вручную и записываются в атрибут **MV**.

4.3.3.2. Запись

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

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
<i>Входные/выходные атрибуты</i>			
IN	DINT	чтение	Вход
OUT	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
MV	DINT	чтение/ запись	Ручное значение
REM_CONN	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 Status 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.

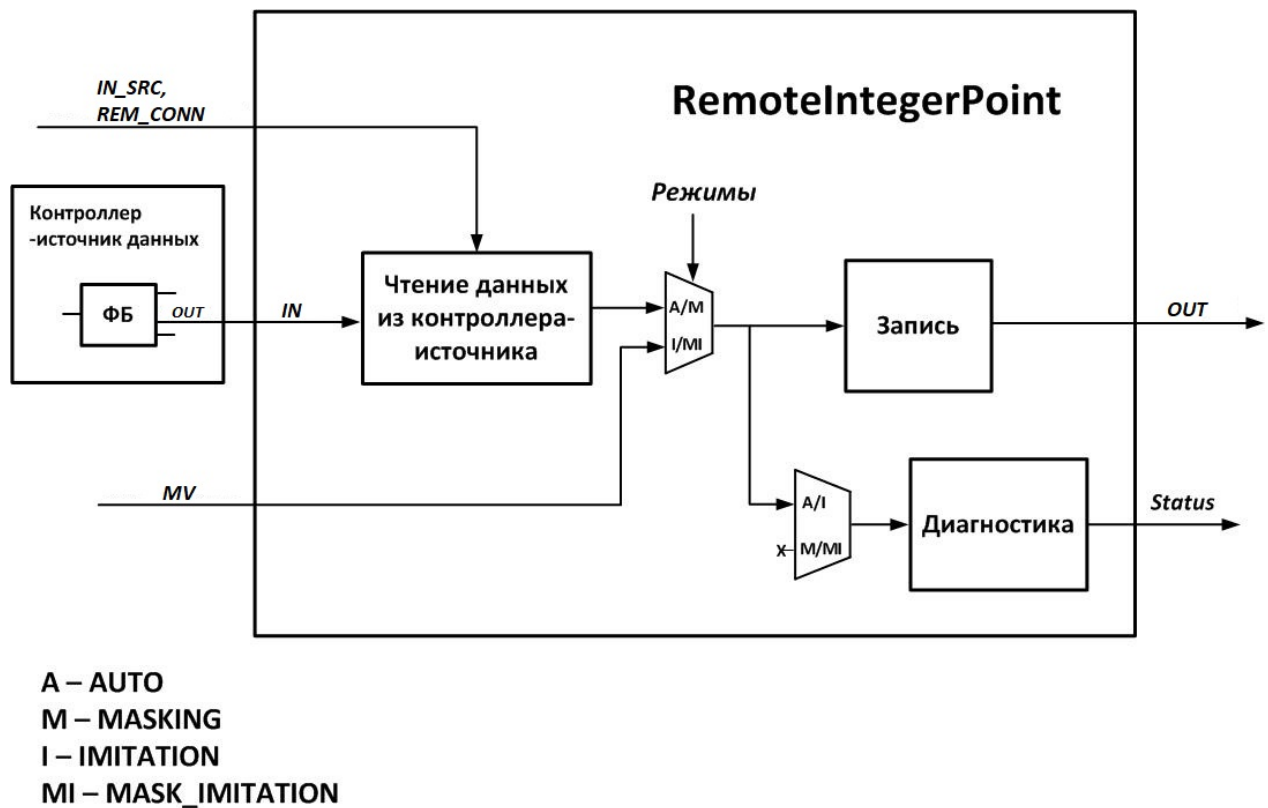


Рисунок 5. Схема процесса обработки данных RemoteIntegerPoint

4.4.3.1. Чтение данных из контроллера-источника

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

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

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

В режимах IMITATION и MASK_IMITATION входные данные вводятся вручную и записываются в атрибут *MV*.

4.4.3.2. Запись

Полученные данные записываются в выходной атрибут *OUT*.

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
BUS_ADDR	DINT	чтение/запись	Адрес корзины
Сигнализационные атрибуты			
NA	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
Конфигурационные атрибуты			
RACK	STRING	чтение/запись	Корзина
SLOT	DINT	чтение/запись	Позиция модуля питания
Диагностические атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
VTG	REAL	чтение	Выходное напряжение
Сигнализационные атрибуты			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
VTG_LO	BOOL	чтение	Низкое выходное напряжение

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

5.2.2. Режимы MK550024PSU

Режимы MK550024PSU:

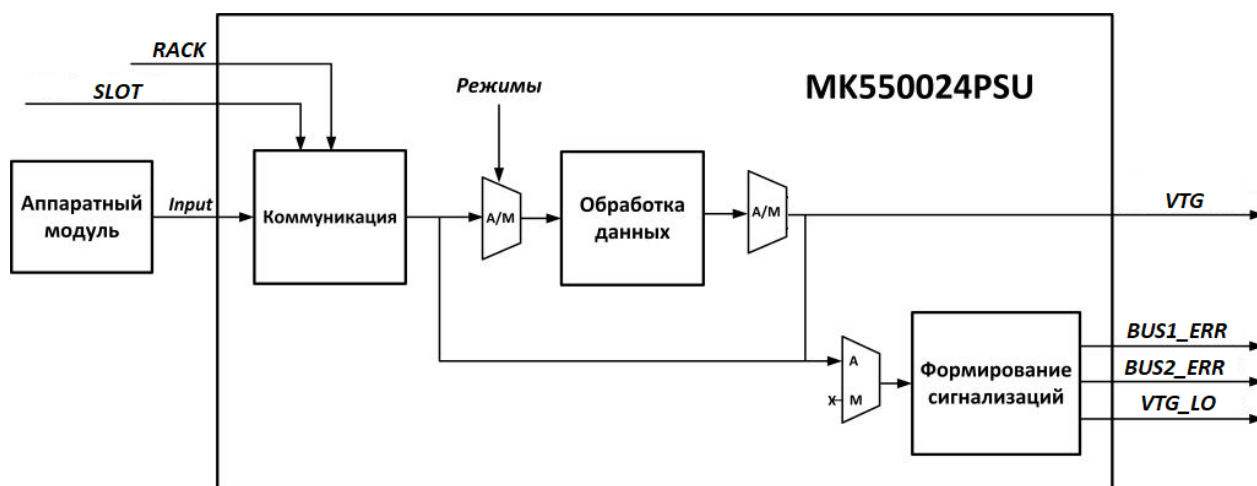
- OFF ("Выключен") - алгоритм не выполняется;
- AUTO ("Автоматический") - алгоритм выполняется в полном объеме;
- MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.

5.2.3. Алгоритм MK550024PSU

Выполнение алгоритма MK550024PSU зависит от установленного режима.

Алгоритм MK550024PSU содержит следующие элементы:

- коммуникация;
- обработка данных;
- формирование сигнализаций.



A – AUTO
M – MASKING

Рисунок 6. Схема процесса обработки данных MK550024PSU

5.2.3.1. Коммуникация

Коммуникация определяет соответствие идентификационных данных MK550024PSU данным аппаратного модуля.

Коммуникация включает в себя следующие элементы:

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

Происходит вычисление номера корзины по атрибуту **RACK** и уникального номера MK550024PSU с использованием номера корзины и атрибута **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными аппаратного модуля.

4. Считывание данных аппаратного модуля.

Производится при положительном результате выполнения элементов 1-3 коммуникации.

5.2.3.2. Обработка данных

На данном этапе выходные значения напряжения записываются в атрибут **VTG**.

5.2.3.3. Формирование сигнализаций

На данном этапе происходит сравнение выходного напряжения блока с уставкой рабочего напряжения.

Сигнализации формируются 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 модуля питания (атрибуты сигнализации **BUS1_ERR** и **BUS2_ERR** принимают значение true);
- выходное напряжение ниже уставки рабочего напряжения (атрибут сигнализации **VTG_LO** принимает значение true).

5.3. Модуль ЦПУ MK502142CPU

Модуль ЦПУ MK502142CPU (далее MK502142CPU) обеспечивает выполнение следующих функций:

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

Максимальная возможная загрузка ЦП для MK502142CPU — 85%.

5.3.1. Атрибуты MK502142CPU

MK502142CPU имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 11. Дополнительные атрибуты MK502142CPU

Имя атрибута	Тип данных	Доступ	Описание атрибута
Конфигурационные атрибуты			
RACK	STRING	чтение/ запись	Корзина
SLOT	DINT	чтение/ запись	Позиция модуля ЦПУ
Диагностические атрибуты			
CPU_LOAD	REAL	чтение	Загрузка модуля ЦПУ, %
SYNC_ROLE	DINT	чтение	Роль модуля ЦПУ (primary(1)/secondary(2))

Имя атрибута	Тип данных	Доступ	Описание атрибута
SYNC_STS	DINT	чтение	Текущее состояние модуля ЦПУ (stop(0)/active(1)/standby(2)/error(3)/disabled(4))
UTIL_MEM	DINT	чтение	Память
Сигнализационные атрибуты			
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 MK502142CPU

Режимы MK502142CPU:

- 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 MK502142CPU

Выполнение алгоритма MK502142CPU зависит от установленного режима.

Алгоритм MK502142CPU включает в себя следующие элементы:

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

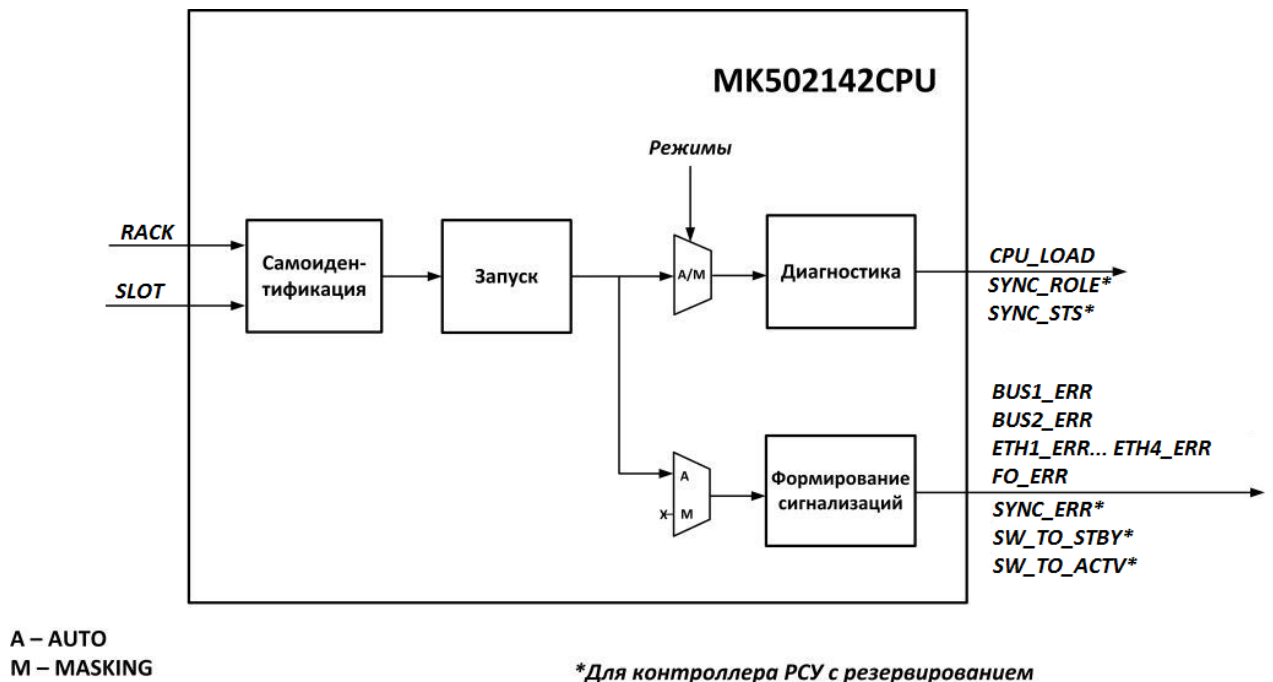


Рисунок 7. Схема процесса обработки данных MK502142CPU

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 MK502142CPU:

- "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 MK502142CPU:

Таблица 12. Элементы этапа запуска MK502142CPU

Резервирование	
Есть	Нет

Роль модуля ЦПУ		
<i>current</i>	<i>redundant</i>	
• осуществляет физическую настройку шин ввода/вывода; • запускает драйвер ввода/вывода; • настраивает драйвер ввода/вывода на работу в активном либо пассивном режиме в зависимости от значения <i>SYNC_STS</i>; • запускает технологическую программу.	• получает и записывает данные о коммуникации от соседнего аппаратного модуля.	• осуществляет физическую настройку шин ввода/вывода; • запускает драйвер ввода/вывода; • запускает технологическую программу.

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>	
• диагностирует коммуникацию сетевых интерфейсов; • диагностирует синхронизацию с соседним аппаратным модулем ЦПУ; • формирует и записывает данные для диагностических атрибутов (Таблица 11. Дополнительные атрибуты МК502142CPU).	• получает данные о текущем состоянии сетевых интерфейсов от соседнего аппаратного модуля ЦПУ; • получает данные о значении диагностических атрибутов соседнего аппаратного модуля ЦПУ; • получает данные от основного модуля ЦПУ о текущей синхронизации с соседним аппаратным модулем ЦПУ.	• диагностирует коммуникацию сетевых интерфейсов; • формирует и записывает данные для диагностических атрибутов (Таблица 11. Дополнительные атрибуты МК502142CPU).

5.3.3.4. Формирование сигнализаций

Формирование сигнализаций зависит от наличия резервирования.

MK502142CPU 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.

MK502142CPU в контроллере без резервирования формирует сигнализации только из группы 1.

5.4. Модули аналогового ввода MK513016AI и MK513016AAI

Модули аналогового ввода MK513016AI и MK513016AAI (далее MK513016AI) обеспечивает выполнение следующих функций:

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



Внимание: Модуль MK513016AAI является модификацией модуля MK513016AI.

5.4.1. Атрибуты MK513016AI

Кроме базового набора атрибутов MK513016AI имеет дополнительные атрибуты:

Таблица 14. Дополнительные атрибуты МК513016AI

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT16	REAL	чтение	Выход - значение соответствующего канала в процентах
Конфигурационные атрибуты			
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 аналогового входа (0...20 мА либо 4...20 мА)
MV1...MV16	REAL	чтение/запись	Ручное значение - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION в процентах
Диагностические атрибуты			
RAW1...RAW16	DINT	чтение	Код АЦП
Сигнализационные атрибуты			
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 МК513016AI имеют дополнительное описание:

Таблица 15. Дополнительное описание значений атрибута *Status* MK513016AI

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

5.4.2. Режимы MK513016AI

Режимы MK513016AI:

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

5.4.3. Алгоритм MK513016AI

Выполнение алгоритма MK513016AI зависит от установленного режима.

Алгоритм MK513016AI 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 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 **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

5.4.3.2. Формирование сигнализаций

MK513016AI 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 MK513016AI (атрибуты сигнализации **BUS1_ERR** и **BUS2_ERR** 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 **OUT1...OUT16**.

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

5.5. Модуль аналогового ввода MK516008AAI

Модуль аналогового ввода MK516008AAI (далее MK516008AAI) обеспечивает выполнение следующих функций:

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

5.5.1. Атрибуты MK516008AAI

Кроме базового набора атрибутов MK516008AAI имеет дополнительные атрибуты:

Таблица 16. Дополнительные атрибуты MK516008AAI

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
OUT1... OUT8	REAL	чтение	Выход - значение соответствующего канала в процентах
Конфигурационные атрибуты			
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 аналогового входа (0...20 мА либо 4...20 мА)
MV1...MV8	REAL	чтение/запись	Ручное значение - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION в процентах
Диагностические атрибуты			
RAW1...RAW8	DINT	чтение	Код АЦП
Сигнализационные атрибуты			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

5.5.2. Режимы MK516008AAI

Режимы MK516008AAI:

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

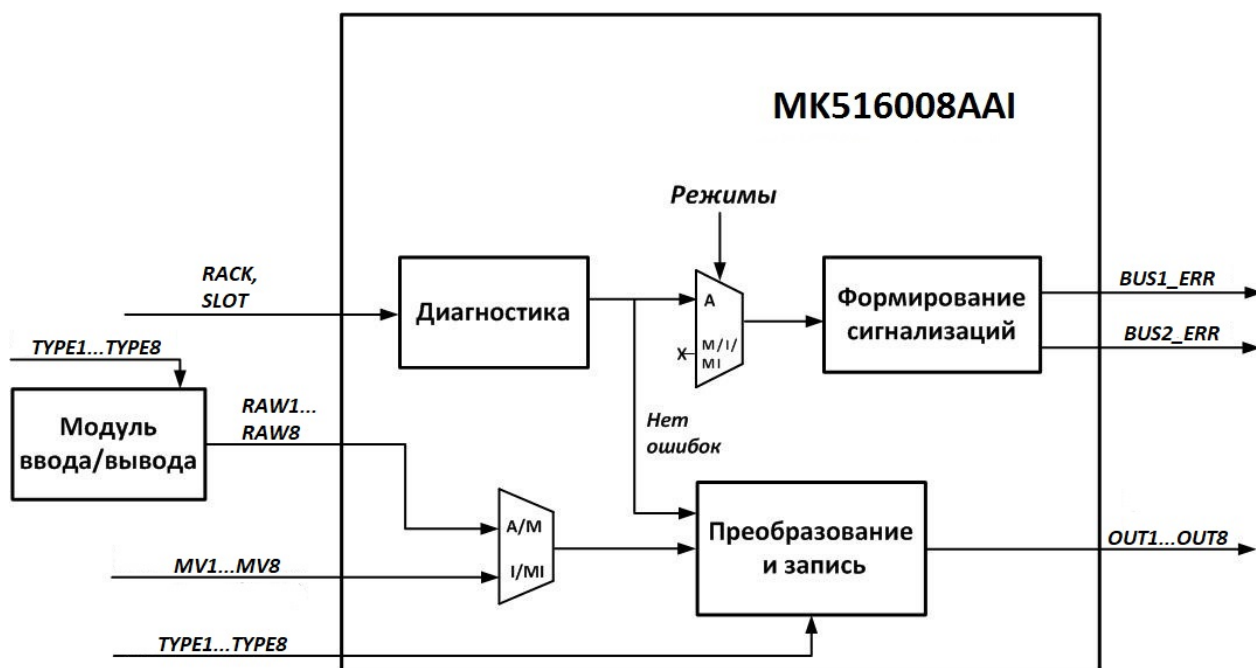
- MASK_IMITATION ("Имитация и маскирование").

5.5.3. Алгоритм MK516008AAI

Выполнение алгоритма MK516008AAI зависит от установленного режима.

Алгоритм MK516008AAI включает в себя следующие элементы:

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



A – AUTO
M – MASKING
I – IMITATION
MI – MASK_IMITATION

Рисунок 9. Схема процесса обработки данных MK516008AAI

5.5.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK516008AAI 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 MK516008AAI. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

5.5.3.2. Формирование сигнализаций

MK516008AAI 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 MK516008AAI (атрибуты сигнализации **BUS1_ERR** и **BUS2_ERR** 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 **OUT1...OUT8**.

В режимах IMITATION и MASK_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений **MV1...MV8** в процентах.

5.6. Модули дискретного ввода MK521032DI и MK521032ADI

Модули дискретного ввода MK521032DI и MK521032ADI (далее MK521032DI) 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
Выходные атрибуты			
OUT1... OUT32	BOOL	чтение	Выход - значение соответствующего канала
Конфигурационные атрибуты			
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 - значение, вводимое в режимах IMITATION и MASK_IMITATION
Сигнализационные атрибуты			
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 загружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля

Имя	Значение	Описание
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

5.6.2. Режимы MK521032DI

Режимы MK521032DI:

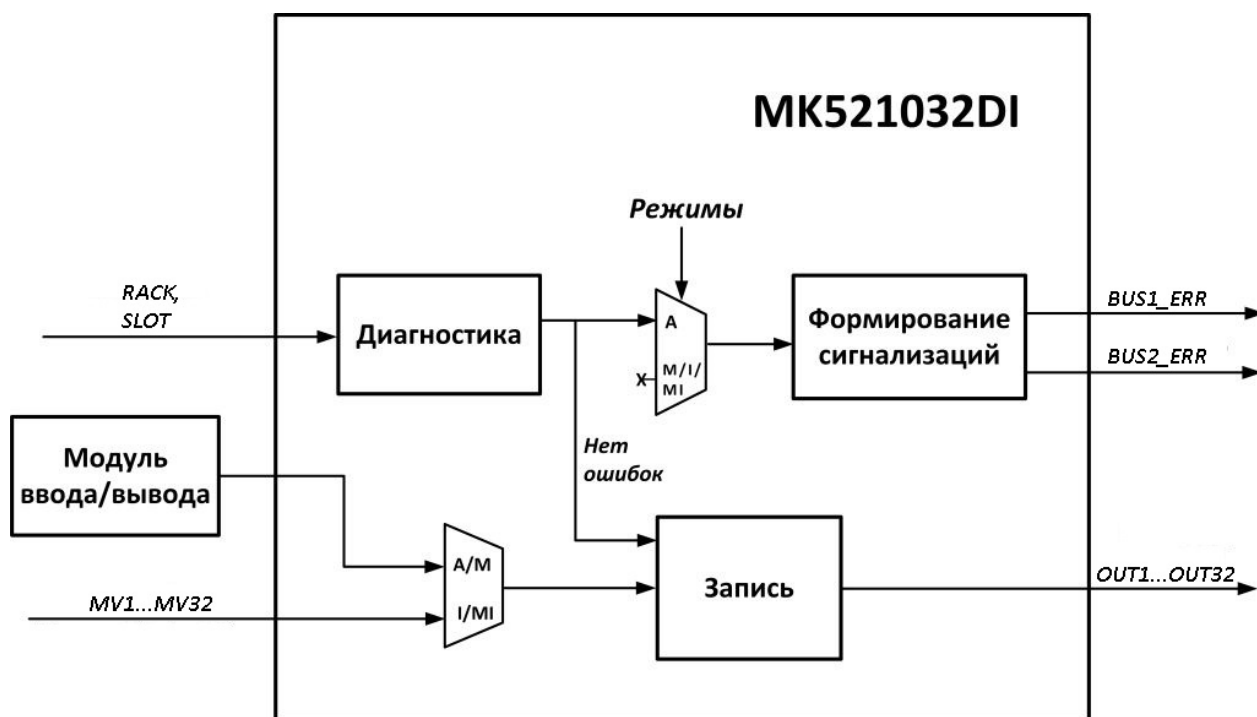
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK_IMITATION ("Имитация и маскирование").

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.



A – AUTO
M – MASKING
I – IMITATION
MI – MASK_IMITATION

Рисунок 10. 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 MK521032DI (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 MK523032ADI

Модуль дискретного ввода MK523032ADI (далее MK523032ADI) 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 MK523032ADI

Кроме базового набора атрибутов MK523032ADI имеет дополнительные атрибуты:

Таблица 19. Дополнительные атрибуты MK523032ADI

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	BOOL	чтение	Выход - значение соответствующего канала
Конфигурационные атрибуты			

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 - значение, вводимое в режимах IMITATION и MASK_IMITATION
STS1_ALM_EN... STS32_ALM_EN	BOOL	чтение/ запись	Включение сигнализации
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
NMATCH_RETR	UDINT	чтение/ запись	Нет повторных попыток совпадения каналов
BNCE_DLY1...BNCE_DLY32	UDINT	чтение/ запись	Время дребезга (минимальное время, необходимое для установления сигнала на входе)
Сигнализационные атрибуты			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
STS1_ALM... STS32_ALM	BOOL	чтение	Состояние сигнализации
Диагностические атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
STS1...STS32	UINT	чтение	Статус входа

Некоторые значения атрибута **Status** для MK523032ADI имеют дополнительное описание:

Таблица 20. Дополнительное описание значений атрибута Status MK523032ADI

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

5.7.2. Режимы MK523032ADI

Режимы MK523032ADI:

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

5.7.3. Алгоритм MK523032ADI

Выполнение алгоритма MK523032ADI зависит от установленного режима.

Алгоритм MK523032ADI включает в себя следующие элементы:

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

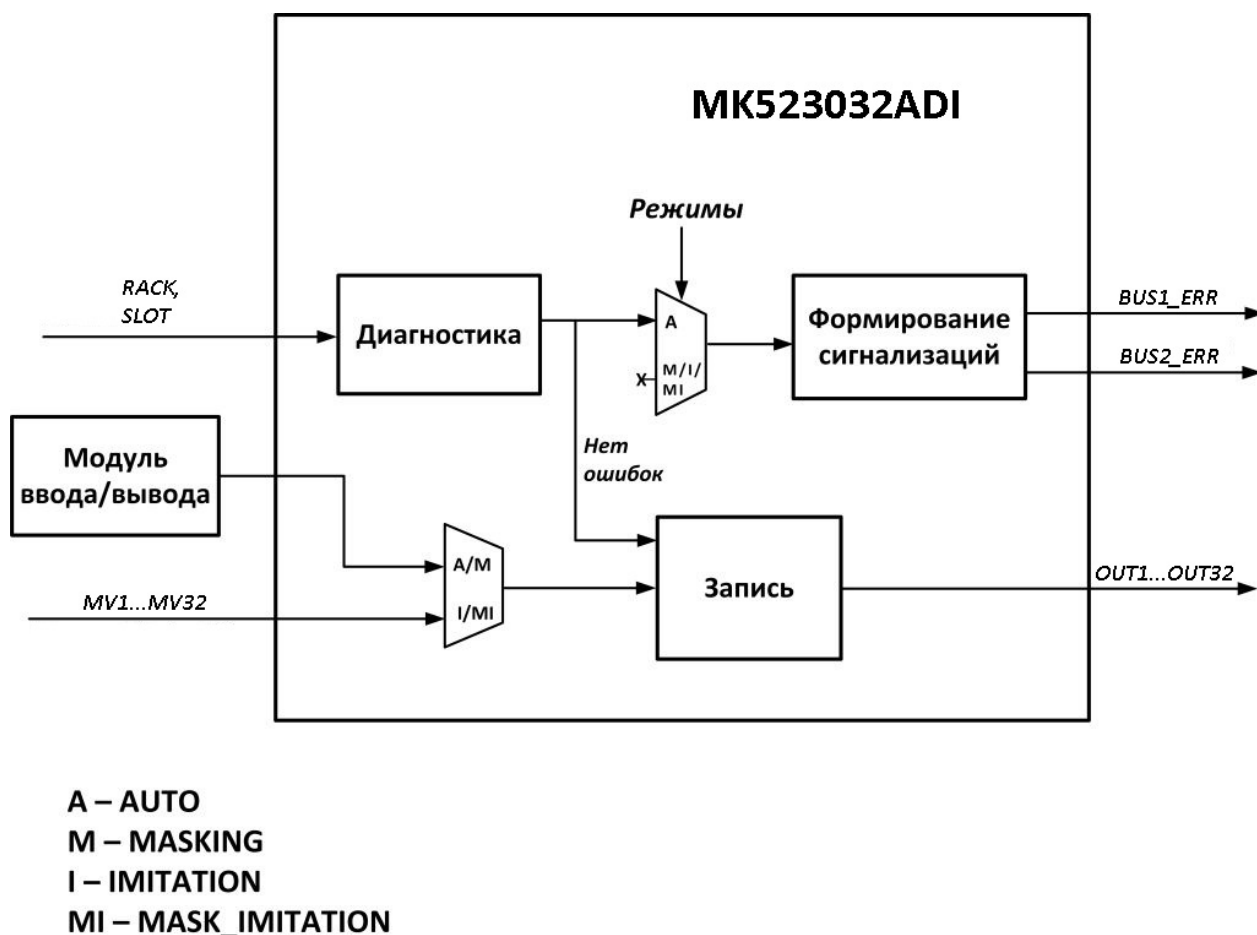


Рисунок 11. Схема процесса обработки данных MK523032ADI

MK523032ADI может формировать следующие выходные статусы каналов:

- ActiveFirst - канал первого модуля пары в состоянии логической 1 (0b0000000000000001);
- OpenCircuitFirst: - канал первого модуля в состоянии "обрыв цепи" (0b0000000000000010);
- ShortCircuitFirst - канал первого модуля в состоянии "короткое замыкание" (0b0000000000000100);
- NotRespFirst - первый модуль не отвечает (0b00000000000001000);
- InternalFaultFirst - внутренняя ошибка канала на первом модуле (0b00000000000010000);
- TerminalCableFirst - отсутствует или поврежден на первом модуле терминальный кабель (0b00000000000100000);
- DetectedBouncingFirst - дребезг на линии первого модуля (сигнал не установился за время дребезга);
- ActiveSecond - канал второго модуля пары в состоянии логической 1 (0b00000000100000000);
- OpenCircuitSecond - канал второго модуля в состоянии "обрыв цепи" (0b00000001000000000);

- ShortCircuitSecond - канал второго модуля в состоянии "короткое замыкание" (0b000000100000000000);
- NotRespSecond - второй модуль не отвечает (0b000001000000000000);
- InternalFaultSecond - внутренняя ошибка канала на втором модуле (0b000100000000000000);
- TerminalCableSecond - отсутствует или поврежден на втором модуле терминальный кабель (0b001000000000000000);
- DetectedBouncingSecond - дребезг на линии второго модуля (сигнал не установился за время дребезга);
- RedundDataNMatch - логические состояния соответствующего канала обоих модулей не совпадают (0b100000000000000000).

5.7.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных МК523032ADI 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 МК523032ADI. 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

МК523032ADI 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 МК523032ADI (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 1 модуля в резервной паре включен;
- "OpenCircuitFirst": 2 - разрыв цепи канала 1 модуля (холостой ход);
- "ShortCircuitFirst": 4 - короткое замыкание канала 1 модуля;
- "NotRespFirst": 8 - 1 модуль не отвечает;
- "ActiveSecond": 16 - канал 2 модуля в резервной паре включен;
- "OpenCircuitSecond": 32 - разрыв цепи канала 2 модуля (холостой ход);
- "ShortCircuitSecond": 64 - короткое замыкание канала 2 модуля;
- "NotRespSecond": 128 - 2 модуль не отвечает;
- "RedundDataNMatch": 256 - показание канала у 1 и 2 модуля 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 MK514008АО и MK514008ААО

Модули аналогового вывода MK514008АО и MK514008ААО (далее MK514008АО) 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 быстросъёмным разъёмом 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
Входные атрибуты			
IN1...IN8	REAL	чтение	Вход - значение соответствующего канала в процентах
Выходные атрибуты			
CH1_ERR...CH8_ERR	BOOL	чтение	Ошибка канала
Конфигурационные атрибуты			
RACK	STRING	чтение/запись	Имя корзины, в которую установлен модуль ввода/вывода
SLOT	DINT	чтение/запись	Позиция модуля в корзине
IN1_SRC...IN8_SRC	STRING	чтение/запись	Источник
TYPE1...TYPE8	DINT	чтение/запись	Тип входов (0...20 мА либо 4...20 мА)
MV1...MV8	REAL	чтение/запись	Ручное значение - значение в режиме MANUAL в процентах

Имя атрибута	Тип данных	Доступ	Описание атрибута
F_TME	USINT	чтение/ запись	Таймаут
F_MODE1...F_MODE8	BOOL	чтение/ запись	Включить режим для канала
FV1...FV8	REAL	чтение/ запись	Значение
Диагностические атрибуты			
RAW1...RAW8	DINT	чтение	Код ЦАП
Сигнализационные атрибуты			
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 MK514008AO имеют дополнительное описание:

Таблица 22. Дополнительное описание значений атрибута Status MK514008AO

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

5.8.2. Режимы MK514008AO

Режимы MK514008AO:

- OFF ("Выключен");

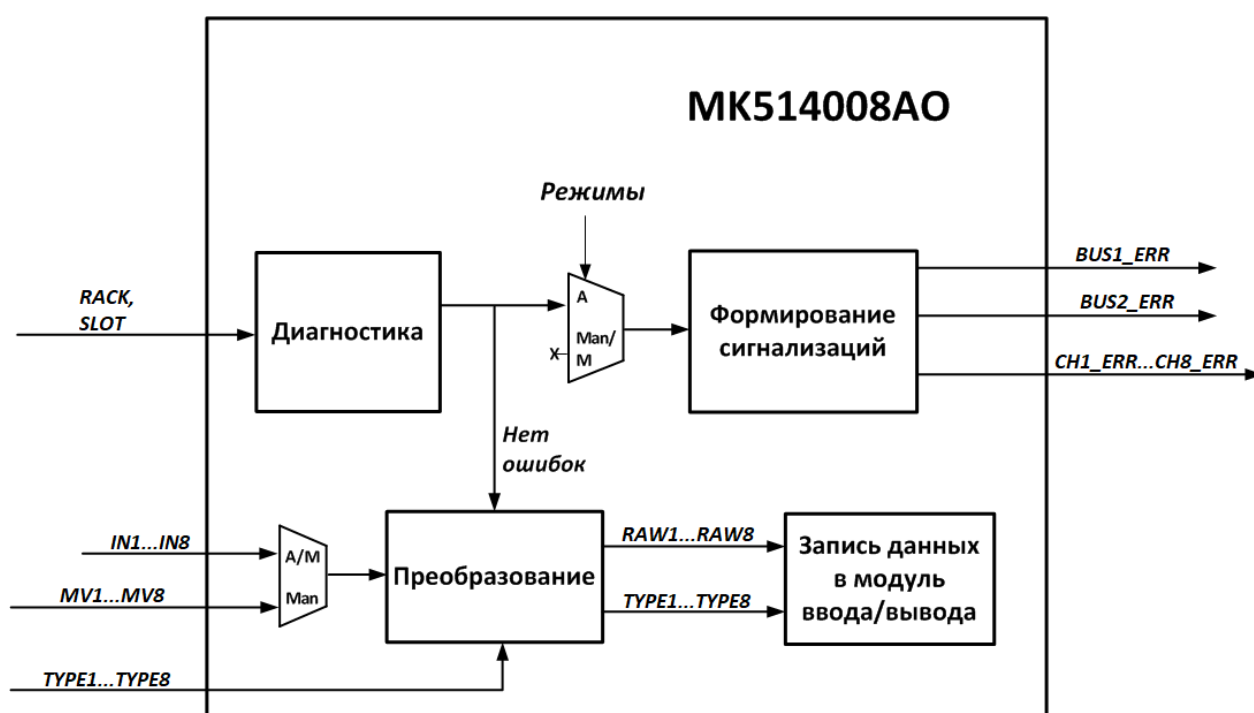
- AUTO ("Автоматический");
- MANUAL ("Ручной")
- MASKING ("Маскирование").

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 12. 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 MK514008AO. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

5.8.3.2. Формирование сигнализаций

MK514008AO 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 MK514008AO (атрибуты сигнализации **BUS1_ERR** и **BUS2_ERR** принимают значение true).

На этапе формирования сигнализаций также производится запись в выходные атрибуты **CH1_ERR...CH8_ERR**. 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 MK531032DO и MK531032ADO

Модули дискретного вывода MK531032DO и MK531032ADO (далее MK531032DO) обеспечивает выполнение следующих функций:

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



Внимание: Модуль MK531032ADO является модификацией модуля MK531032DO.

5.9.1. Атрибуты MK531032DO

MK531032DO имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 23. Дополнительные атрибуты MK531032DO

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1... IN32	BOOL	чтение	Вход (значение соответствующего канала)
<i>Конфигурационные атрибуты</i>			
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
MV1...MV32	BOOL	чтение/ запись	Ручное значение - значение, вводимое в режиме MANUAL
F_TME	USINT	чтение/ запись	Таймаут
F_MODE1...F_MODE32	BOOL	чтение/ запись	Включить режим для канала
F_VALUE1...F_VALUE32	BOOL	чтение/ запись	Значение
<i>Сигнализационные атрибуты</i>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
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 MK531032DO имеют дополнительное описание:

Таблица 24. Дополнительное описание значений атрибута *Status* MK531032DO

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

5.9.2. Режимы MK531032DO

Режимы MK531032DO:

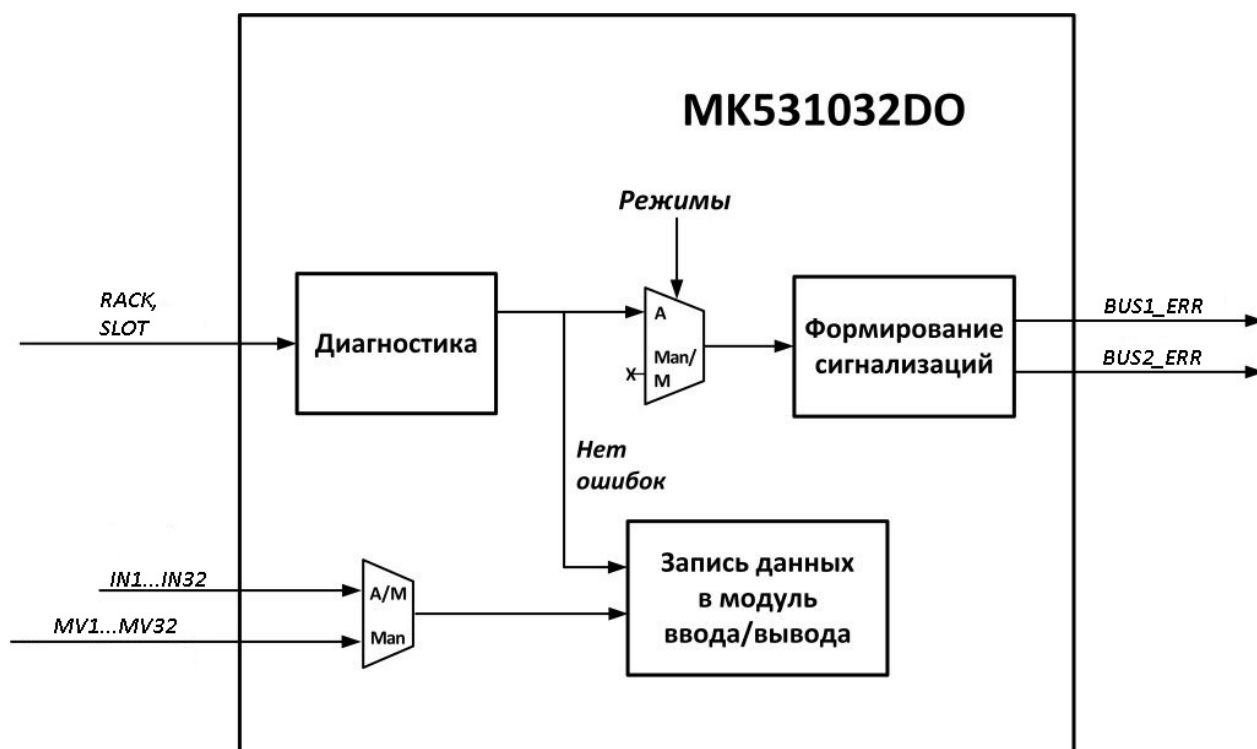
- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

5.9.3. Алгоритм MK531032DO

Выполнение алгоритма MK531032DO 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 13. 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

MK531032DO 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 MK531032DO (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 дискретного ввода MK532032ADO

Модуль дискретного ввода MK532032ADO (далее MK532032ADO) обеспечивает выполнение следующих функций:

- диагностика;
- формирование сигнализаций;
- запись данных в модуль ввода/вывода.
- работа в составе резервной пары

5.10.1. Атрибуты MK532032ADO

MK532032ADO имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 25. Дополнительные атрибуты MK532032ADO

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN1... IN32	BOOL	чтение	Вход(значение соответствующего канала)
Выходные атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
ERR_OUT1... ERR_OUT32	BOOL	чтение	Ошибка выхода
Конфигурационные атрибуты			
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
OUT1_EN... OUT32_EN	STRING	чтение/ запись	Включение канала
MV1...MV32	BOOL	чтение/ запись	Ручное значение - значение, вводимое в режиме MANUAL
F_TME	USINT	чтение/ запись	Таймаут
F_MODE1...F_MODE32	BOOL	чтение/ запись	Включить режим для канала
F_VALUE1...F_VALUE32	BOOL	чтение/ запись	Значение
STS1_ALM_EN... STS32_ALM_EN	BOOL	чтение/ запись	Включение сигнализации
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
NMATCH_RETR	UDINT	чтение/ запись	Максимальное количество несовпадений каналов
ERR_RETR	UDINT	чтение/ запись	Количество ошибок подряд

Имя атрибута	Тип данных	Доступ	Описание атрибута
Диагностические атрибуты			
STS1...STS32	UINT	чтение	Статус входа
Сигнализационные атрибуты			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
STS1_ALM... STS32_ALM	BOOL	чтение	Состояние сигнализации

Некоторые значения атрибута **Status** для MK532032ADO имеют дополнительное описание:

Таблица 26. Дополнительное описание значений атрибута Status MK532032ADO

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

5.10.2. Режимы MK532032ADO

Режимы MK532032ADO:

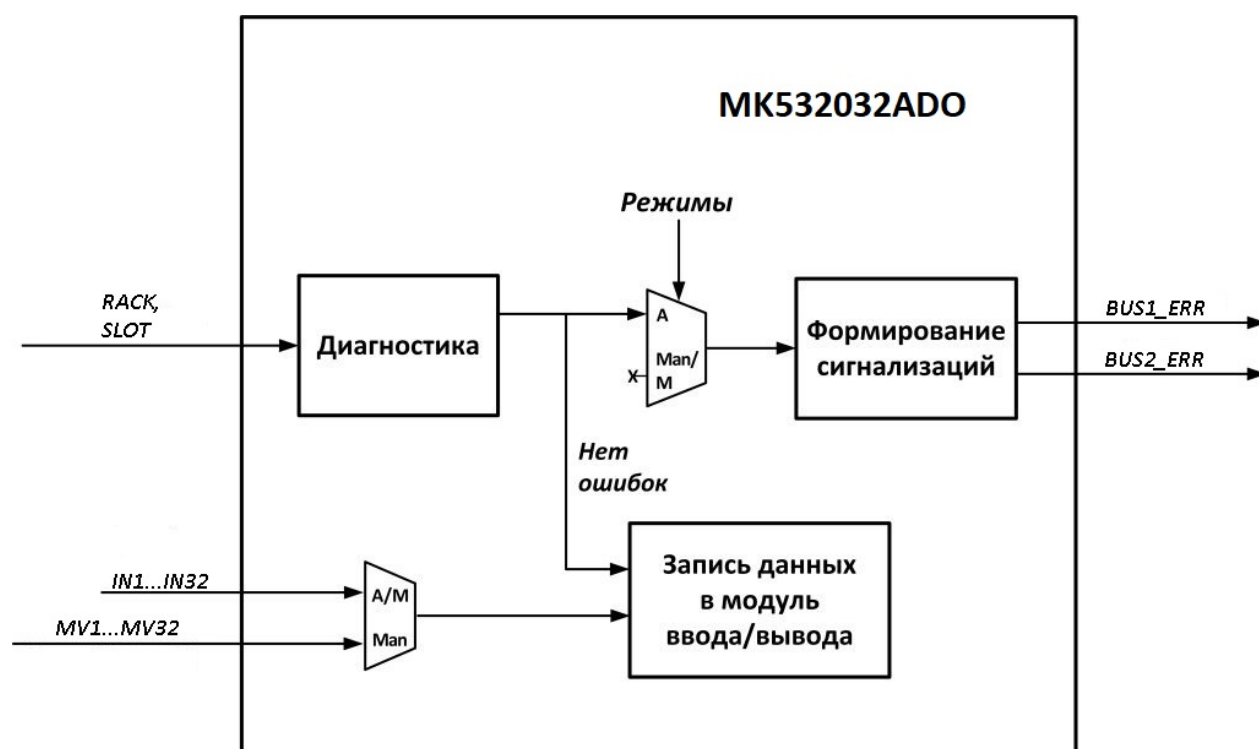
- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

5.10.3. Алгоритм MK532032ADO

Выполнение алгоритма MK532032ADO 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 14. Схема процесса обработки данных MK532032ADO

5.10.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK532032ADO 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 MK532032ADO. 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**.

5.10.3.2. Формирование сигнализаций

MK532032ADO 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 MK532032ADO (атрибуты сигнализации **BUS1_ERR** и **BUS2_ERR** принимают значение true).

5.10.3.3. 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**.

5.10.3.4. Поддержка сигналов NAMUR

MK532032ADO 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 MK532032ADO 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.

MK532032ADO формирует следующие дополнительные статусы модуля:

- "ActiveFirst": 1 - канал 1 модуля в резервной паре включен;
- "OpenCircuitFirst": 2 - разрыв цепи канала 1 модуля (холостой ход);
- "ShortCircuitFirst": 4 - короткое замыкание канала 1 модуля;
- "NotRespFirst": 8 - 1 модуль не отвечает;
- "InternalFaultFirst": 16 - внутренняя ошибка 1 модуля;
- "PowerFaultFirst": 32 - отсутствует питание у канала 1 модуля;
- "ActiveSecond": 64 - канал 2 модуля в резервной паре включен;
- "OpenCircuitSecond": 128 - разрыв цепи канала 2 модуля (холостой ход);
- "ShortCircuitSecond": 256 - короткое замыкание канала 2 модуля;
- "NotRespSecond": 512 - 2 модуль не отвечает;
- "InternalFaultSecond": 1024 - внутренняя ошибка 2 модуля;
- "PowerFaultSecond": 2048 - отсутствует питание у канала 2 модуля;
- "RedundDataNMatch": 4096 - показание канала у 1 и 2 модуля не совпадают;

Примечание. Если в ФБ модуля режим резервирования отключен, дополнительные статусы резервного модуля копируются из основного.

5.10.3.5. Работа в составе резервной пары

MK532032ADO поддерживает работу в составе резервной пары.

Режим резервирования включается с помощью атрибута ***RDNCY_ENA***.

В режиме резервирования ФБ модуля 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.

5.11. Модуль аналогового ввода MK576008AAIHART

Модуль аналогового ввода MK576008AAIHART (далее MK576008AAIHART) 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 MK576008AAIHART

Кроме базового набора атрибутов MK576008AAIHART имеет дополнительные атрибуты:

Таблица 27. Дополнительные атрибуты MK576008AAIHART

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT8	REAL	чтение	Выход - значение соответствующего канала в процентах
HART_PV1... HART_PV8	REAL	чтение	Значение первой переменной HART канала
HART_SV1... HART_SV8	REAL	чтение	Значение второй переменной HART канала
HART_TV1... HART_TV8	REAL	чтение	Значение третьей переменной HART канала

Имя атрибута	Тип данных	Доступ	Описание атрибута
HART_QV1... HART_QV8	REAL	чтение	Значение четвертой переменной HART канала
HART_ERR1... HART_ERR8	BOOL	чтение	Соответствующий HART канал в состоянии ошибки (ошибка будет выставляться при наличии тока на канале выше 3мА и при отсутствии данных по каналу при режиме 4...20 мА)
Конфигурационные атрибуты			
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 аналогового входа (0...20 мА либо 4...20 мА)
MV1...MV8	REAL	чтение/ запись	Ручное значение - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION
HART_VAR_NUM1... HART_VAR_NUM8	DINT	чтение/ запись	Количество параметров HART канала – количество параметров, передаваемых через выбранный HART канал
HART_ID1... HART_ID8	DINT	чтение/ запись	Идентификатор HART-устройства, установленного на канале
ADDR_OFFS	USINT	чтение/ запись	Смещение каналов HART
Диагностические атрибуты			
RAW1...RAW8	DINT	чтение	Код АЦП
Сигнализационные атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
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 МК576008AAIHART имеют дополнительное описание:

Таблица 28. 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
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

5.11.2. Режимы МК576008AAIHART

Режимы МК576008AAIHART:

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

5.11.3. Алгоритм МК576008AAIHART

Выполнение алгоритма МК576008AAIHART⁵ зависит от установленного режима.

Алгоритм МК576008AAIHART включает в себя следующие элементы:

- диагностика;

⁵ HART работает только при токе 4...20 мА

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

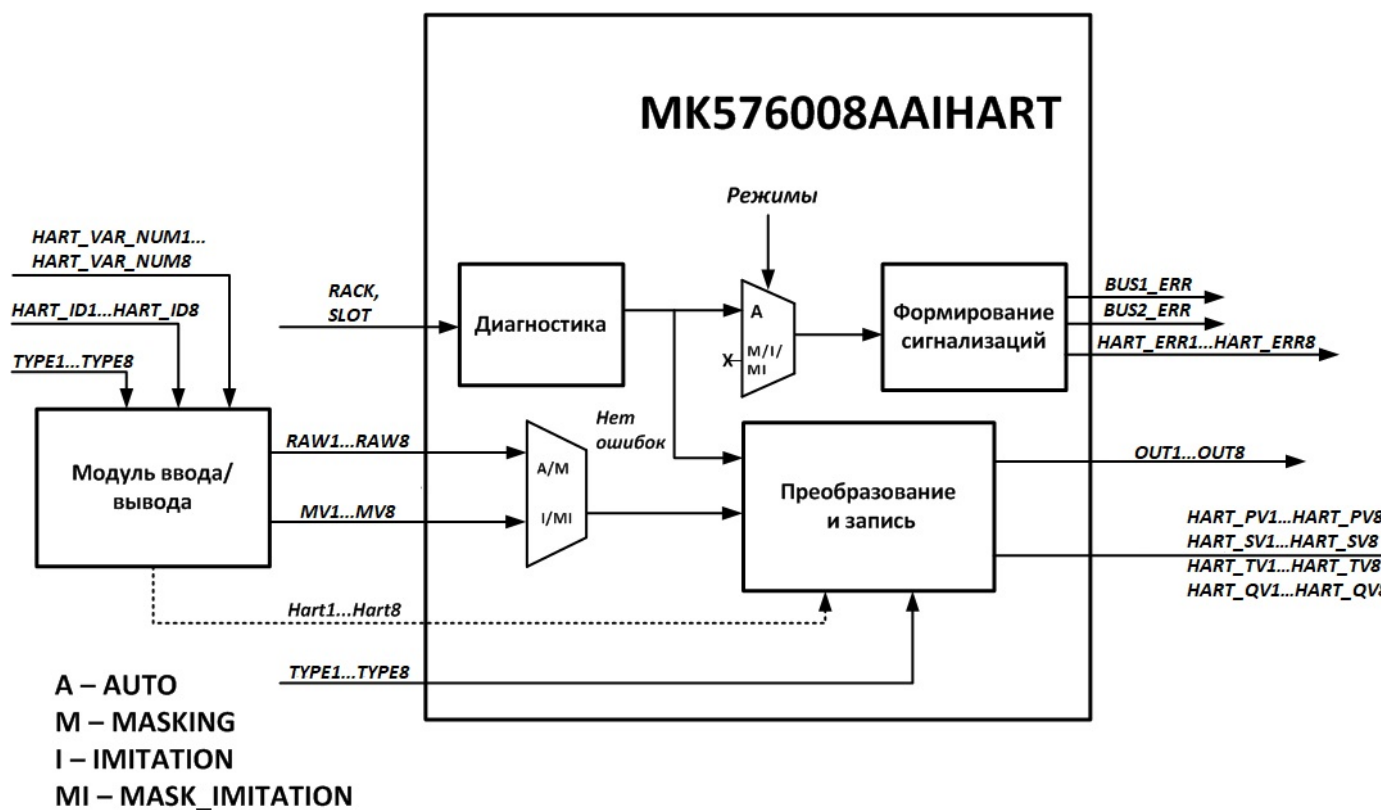


Рисунок 15. Схема процесса обработки данных MK576008AAIHART

5.11.3.1. 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 **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**.

5.11.3.2. Формирование сигнализаций

МК576008ААІHART 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 МК576008ААІHART (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...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 **OUT1...OUT8**.
- запись преобразованных данных из HART каналов:
 - **HART_PV1...HART_PV8**;
 - **HART_SV1...HART_SV8**;
 - **HART_TV1...HART_TV8**;
 - **HART_QV1...HART_QV8**.

В режимах IMITATION и MASK_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений **MV1...MV8** в процентах.

В режимах IMITATION и MASK_IMITATION переменные:

- **HART_PV1...HART_PV8**;
- **HART_SV1...HART_SV8**;
- **HART_TV1...HART_TV8**;
- **HART_QV1...HART_QV8**.

не опрашиваются, а принимаются их последние значения.

5.11.3.4. Работа HART модулей в составе PDM систем

Модули ввода-вывода МК576008ААІHART, МК576016ААІHART и МК574008ААОHART могут интегрироваться в PDM-системы, поддерживающие формат DTM (PactWare, FDT Container, PRM Yokogawa).

Модули MK574008AAIHART и MK576008AAIHART являются 8-ми канальными HART-модемами, модуль MK576016AAIHART – 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.

5.12. 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.

5.12.1. Атрибуты MK576016AAIHART

Кроме базового набора атрибутов MK576016AAIHART имеет дополнительные атрибуты:

Таблица 29. Дополнительные атрибуты MK576016AAIHART

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1... OUT16	REAL	чтение	Выход - значение соответствующего канала в процентах
HART_PV1... HART_PV16	REAL	чтение	Значение первой переменной HART канала
HART_SV1... HART_SV16	REAL	чтение	Значение второй переменной HART канала
HART_TV1... HART_TV16	REAL	чтение	Значение третьей переменной HART канала
HART_QV1... HART_QV16	REAL	чтение	Значение четвертой переменной HART канала
HART_ERR1... HART_ERR16	BOOL	чтение	Соответствующий HART канал в состоянии ошибки
Конфигурационные атрибуты			
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 аналогового входа (0...20 мА либо 4...20 мА)
MV1... MV16	REAL	чтение/ запись	Ручное значение - 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 – количество параметров, передаваемых через выбранный HART канал
HART_ID1... HART_ID16	DINT	чтение/ запись	Идентификатор HART-устройства, установленного на канале
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
RDNCY_MODE	USINT	чтение/ запись	Резервный режим (Тип резервирования)
ADDR_OFFS1	USINT	чтение/ запись	Смещение каналов HART
ADDR_OFFS2	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
Диагностические атрибуты			
RAW1...RAW16	DINT	чтение	Код АЦП
OUTI1...OUTI16	REAL	чтение	Входное значение тока, мА
STS1...STS16	UINT	чтение	Статус входа
Сигнализационные атрибуты			
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)

Некоторые значения атрибута **Status** для МК576016AAIHART имеют дополнительное описание:

Таблица 30. Дополнительное описание значений атрибута Status МК576016AAIHART

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

5.12.2. Режимы МК576016AAIHART

Режимы МК576016AAIHART:

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

5.12.3. Алгоритм МК576016AAIHART

Выполнение алгоритма МК576016AAIHART⁶ зависит от установленного режима.

Алгоритм МК576016AAIHART включает в себя следующие элементы:

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

⁶ HART работает только при токе 4...20 мА

- работа HART в режиме резервирования.

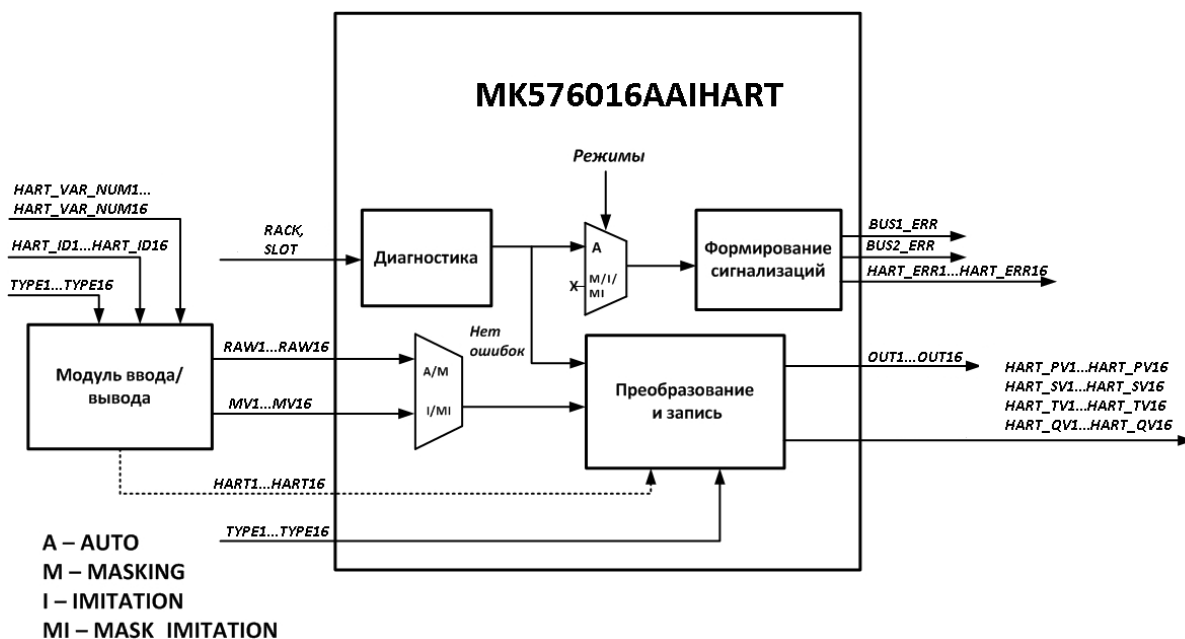


Рисунок 16. Схема процесса обработки данных MK576016AAIHART

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 допустимый 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 (0b0000000000000001);
- RawLimitOverFirst - уровень сигнала канала первого модуля пары вне диапазона значений (0b0000000000000010);

- NotRespFirst - первый модуль данного канала не отвечает (0b00000000000000100);
- RawLowLimitOverFirst - уровень сигнала канала первого модуля пары вне диапазона нижних значений (0b00000000000001000);
- ActiveSecond - канал второго модуля пары является активным (0b0000000100000000);
- RawLimitOverSecond - уровень сигнала канала второго модуля пары вне диапазона значений (0b0000001000000000);
- NotRespSecond - второй модуль данного канала не отвечает (0b0000010000000000);
- RawLowLimitOverSecond - уровень сигнала канала второго модуля пары вне диапазона нижних значений (0b0000100000000000);
- RedundDataNMatch - разброс величины аналогового сигнала получаемого с 1 и 2 модулей превысил установленный предел *NMATCH_DELTA* более чем *NMATCH_RETR* раз подряд (0b1000000000000000).

5.12.3.1. 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*.

5.12.3.2. 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).

5.12.3.3. 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.

5.12.3.4. Работа HART в режиме резервирования

Модуль МК576016AAIHART 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.

МК576016AAIHART 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 аналогового вывода MK574008AAOHART

Модуль аналогового ввода MK574008AAOHART (далее MK574008AAOHART) 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.

5.13.1. Атрибуты MK574008AAOHART

Кроме базового набора атрибутов MK574008AAOHART имеет дополнительные атрибуты:

Таблица 31. Дополнительные атрибуты MK576008AAIHART

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN1... IN8	REAL	чтение	Вход - значение соответствующего канала в процентах
Выходные атрибуты			
HART_PV1... HART_PV8	REAL	чтение	Значение первой переменной HART канала
HART_SV1... HART_SV8	REAL	чтение	Значение второй переменной HART канала
HART_TV1... HART_TV8	REAL	чтение	Значение третьей переменной HART канала
HART_QV1... HART_QV8	REAL	чтение	Значение четвертой переменной HART канала
HART_ERR1... HART_ERR8	BOOL	чтение	Соответствующий HART канал в состоянии ошибки
CH1_ERR...CH8_ERR	BOOL	чтение	Ошибка канала
Конфигурационные атрибуты			

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
TYPE1...TYPE8	DINT	чтение/ запись	Тип аналогового входа (0...20 мА либо 4...20 мА)
MV1... MV8	REAL	чтение/ запись	Ручное значение - значение соответствующего канала в процентах
HART_VAR_NUM1... HART_VAR_NUM8	DINT	чтение/ запись	Количество параметров HART канала – количество параметров, передаваемых через выбранный HART канал
HART_ID1... HART_ID8	DINT	чтение/ запись	Идентификатор HART-устройства, установленного на канале
IN1_SRC...IN8_SRC	STRING	чтение/ запись	Источник
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
RDNCY_MODE	USINT	чтение/ запись	Резервный режим (Тип резервирования)
ADDR_OFFS1	USINT	чтение/ запись	Смещение каналов HART
F_TME	USINT	чтение/ запись	Таймаут
F_MODE1...F_MODE8	BOOL	чтение/ запись	Включить режим для канала
FV1...FV8	REAL	чтение/ запись	Значение
ADDR_OFFS2	USINT	чтение/ запись	Смещение каналов HART

Имя атрибута	Тип данных	Доступ	Описание атрибута
Диагностические атрибуты			
RAW1...RAW8	DINT	чтение	Код АЦП
STS1...STS8	UINT	чтение	Статус выхода
ADT_STS1...ADT_STS8	UINT	чтение	Дополнительный статус выхода
Сигнализационные атрибуты			
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 MK574008AAOHART имеют дополнительное описание:

Таблица 32. Дополнительное описание значений атрибута Status MK574008AAOHART

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

5.13.2. Режимы MK574008AAOHART

Режимы MK574008AAOHART:

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

5.13.3. Алгоритм МК574008ААОНART

Выполнение алгоритма МК574008ААОНART⁷ 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.

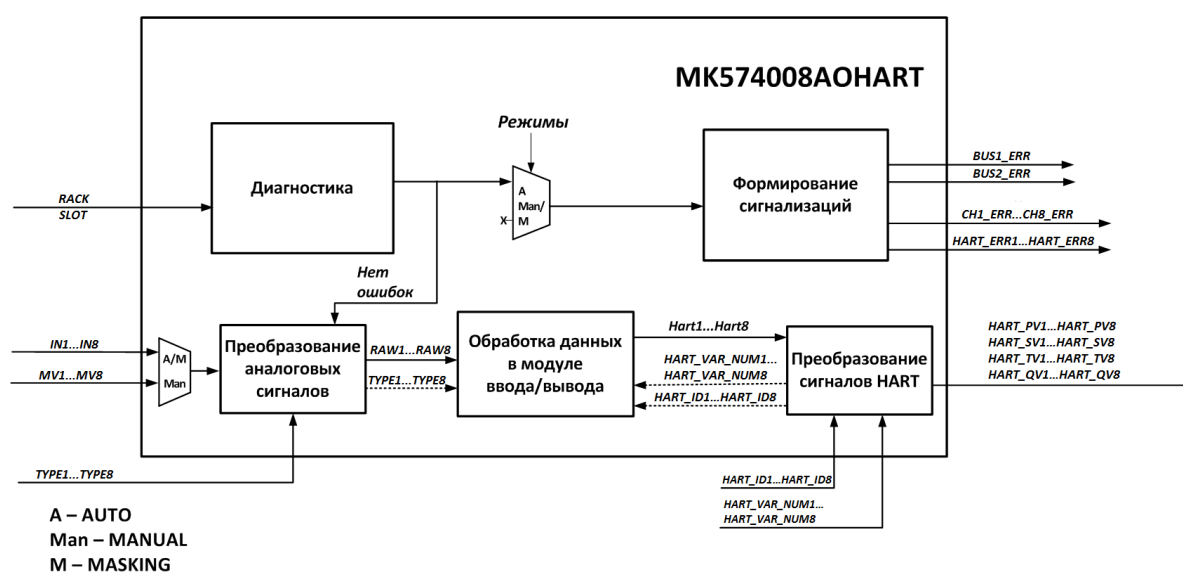


Рисунок 17. Схема процесса обработки данных МК574008ААОНART

МК574008ААОНART 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:

⁷ HART работает только при токе 4...20 мА

- **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.

МК574008ААОНАТ 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).

МК574008ААОНАТ может формировать следующие диагностические выходные статусы:

- ConfigError (1) - у резервной пары модулей разные режимы или включение;
- CurrentSetError (2) - у резервной пары модулей разное задание;
- CAPError (3) - ЦАП не отвечает или в ошибке, либо заданный ток не соответствует измеренному;
- InterModuleCommunicationError (4) - отказ перемычки UART или DI/DO;
- OpenCircuitError (5) - XX на канале, при том, что у второго модуля XX нет;
- SILFault (6) - не прошел тест ключей SIL ячейки;
- ChannelCircuitFailure (7) - измеряем ток ниже заданного, при этом второй модуль исправен;
- UnknownInternalFault (8) - отказ АЦП. АЦП не отвечает по SPI.

5.13.3.1. 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**.

5.13.3.2. 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).

5.13.3.3. 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.

5.13.3.4. 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**.

5.13.3.5. Работа HART в режиме резервирования

Модуль MK574008AAOHART поддерживает работу в составе резервной пары. Данный режим включается установкой атрибута **RDNCY_EN** 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_EN** 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.

5.14. Модуль коммуникации МК541002 с поддержкой Modbus

Модуль коммуникации МК541002 с поддержкой Modbus (далее МК541002) обеспечивает приём и передачу данных по интерфейсу RS-485 протокола ModbusRTU и выполняет следующие функции:

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

Модуль коммуникации МК541002 используется только в качестве ведущего устройства (master) протокола ModbusRTU.

5.14.1. Атрибуты МК541002

МК541002 имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 33. Дополнительные атрибуты МК541002

Имя атрибута	Тип данных	Доступ	Описание атрибута
Конфигурационные атрибуты			
RACK	STRING	чтение/ запись	Корзина
SLOT	DINT	чтение/ запись	Позиция модуля

Имя атрибута	Тип данных	Доступ	Описание атрибута
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_SET	BOOL	чтение	Установить рабочую паузу
WRK_PAUSE_RST	BOOL	чтение	Сбросить рабочую паузу
Сигнализационные атрибуты			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
Диагностические атрибуты			
WRK_PAUSE_STS	BOOL	чтение	Статус рабочей паузы

5.14.2. 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.

5.14.3. 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 18. Схема процесса обработки данных МК541002

5.14.3.1. Конфигурация

На данном этапе производится настройка коммуникации модуля МК541002 с устройствами Modbus.

Параметры настройки записываются в конфигурационные атрибуты МК541002.

5.14.3.2. Коммуникация

На данном этапе модулем МК541002 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 МК541002 и далее распределяются по блокам карт Modbus. Данная работа осуществляется в соответствии с алгоритмом работы Modbus карты.⁸

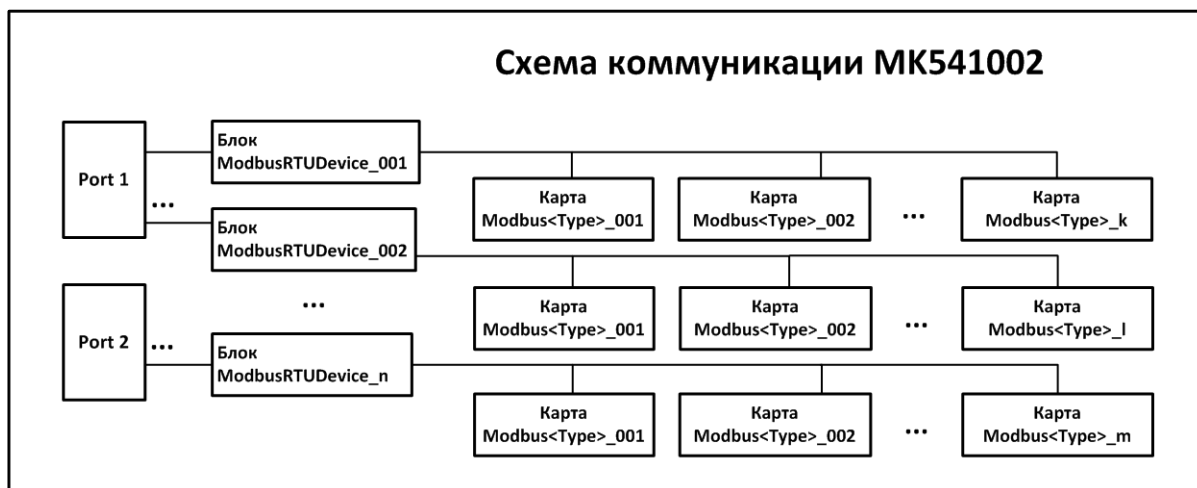


Рисунок 19. Структурная схема связи МК541002

5.14.3.3. 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).

⁸ Подробное описание алгоритмов работы карт Modbus описано в разделе [Функциональные блоки коммуникации по ModBus](#).

5.14.3.4. Формирование сигнализаций

МК541002 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 МК541002 (атрибуты сигнализации ***BUS1_ERR*** и ***BUS2_ERR*** принимают значение true).

6. Функциональные блоки коммуникации по ModBus

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

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:

ST_REG - 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.

REG_TYPE - 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).



Внимание: Количество регистров, задаваемых пользователем в Modbus карте, должно соответствовать числу регистров, доступных на подключаемом устройстве. Если указанное количество регистров превышает доступное, Modbus карте автоматически присваивается статус *BadInput*.

ST_BIT - 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.

BIT_TYPE - 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).

WORD_ENDIAN и **BYTE_ENDIAN** - 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 34. 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

Расположение байтов	Порядок (WordEndian)/значение	Порядок слов (ByteEndian)/значение	байтов
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.

SCN_RATE - 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 35. Дополнительные атрибуты ModBusTCPDevice

Имя атрибута	Тип данных	Доступ	Описание атрибута
Конфигурационные атрибуты			
DEV_ID	DINT	чтение/ запись	ID устройства
DEV_IP	STRING	чтение/ запись	IP адрес устройства
DEV_P	DINT	чтение/ запись	TCP порт
RESP_TO	DINT	чтение/ запись	Максимальное время ответа, мс
HW	STRING	чтение/ запись	Аппаратное обеспечение

Имя атрибута	Тип данных	Доступ	Описание атрибута
CLNT_TYPE	DINT	чтение/ запись	Тип клиента

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 36. Дополнительные атрибуты ModBusRTUDevice

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
DEV_ID	DINT	чтение/ запись	ID устройства
P_NUM	DINT	чтение/ запись	Номер последовательного порта (1 либо 2)
SPD	DINT	чтение/ запись	Скорость, (9600 бит/с)
SP_BITS	DINT	чтение/ запись	Стоповый бит (1 либо 2)
PTY	DINT	чтение/ запись	Четность
TO	DINT	чтение/ запись	Допустимая задержка на ответ, мс
HW	STRING	чтение/ запись	Аппаратное обеспечение
CLNT_TYPE	DINT	чтение/ запись	Тип клиента

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 **PTY** 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 37. Дополнительные атрибуты ModBusRealInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	REAL	чтение	Выход
Конфигурационные атрибуты			
ST_REG	DINT	чтение/ запись	Начальный регистр
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET32	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 38. Дополнительное описание значений атрибута Status ModBusRealInput32

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружено 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.

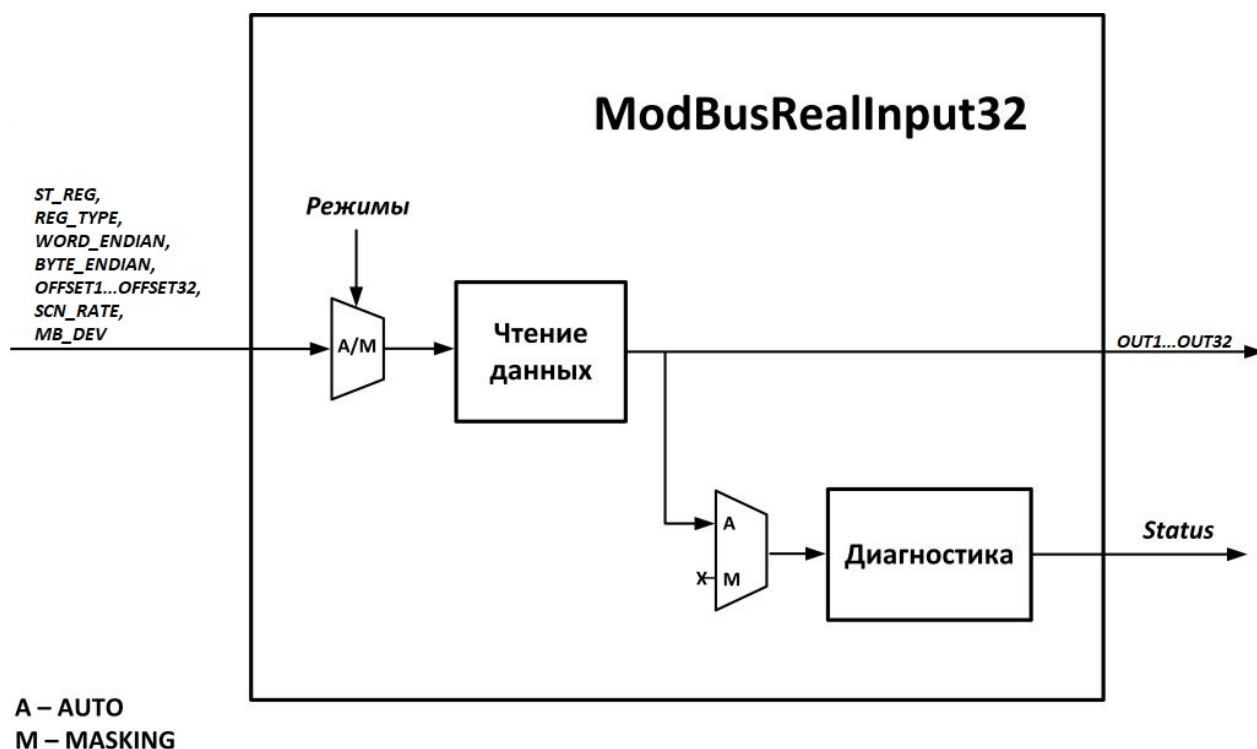


Рисунок 20. Схема процесса обработки данных ModBusRealInput32

6.3.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты **OUT1 ... OUT32** ModBusRealInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

ST_REG;

REG_TYPE;

WORD_ENDIAN и **BYTE_ENDIAN;**

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 39. Дополнительные атрибуты ModBusRealOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	REAL	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	REAL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_REG	DINT	чтение/ запись	Начальный регистр
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)

Имя атрибута	Тип данных	Доступ	Описание атрибута
IN1_SRC...IN32_SRC	STRING	чтение/ запись	Имя источника входных данных
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение

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 40. 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.

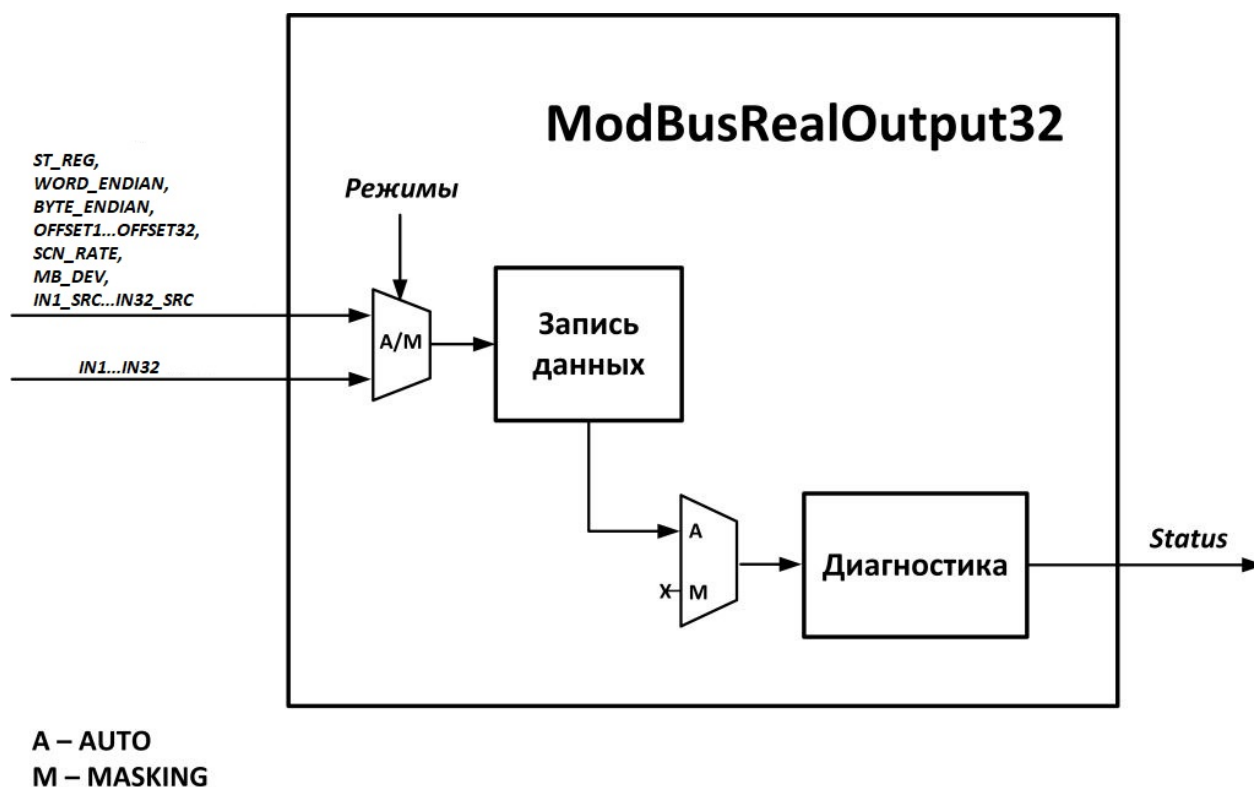


Рисунок 21. Схема процесса обработки данных ModBusRealInput32

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:

ST_REG;

WORD_ENDIAN и *BYTE_ENDIAN*;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV;

IN1_SRC...IN32_SRC.

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 41. Дополнительные атрибуты ModBusIntegerInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	DINT	чтение	Выход
Конфигурационные атрибуты			
ST_REG	DINT	чтение/ запись	Начальный регистр
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)

Имя атрибута	Тип данных	Доступ	Описание атрибута
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 42. 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:

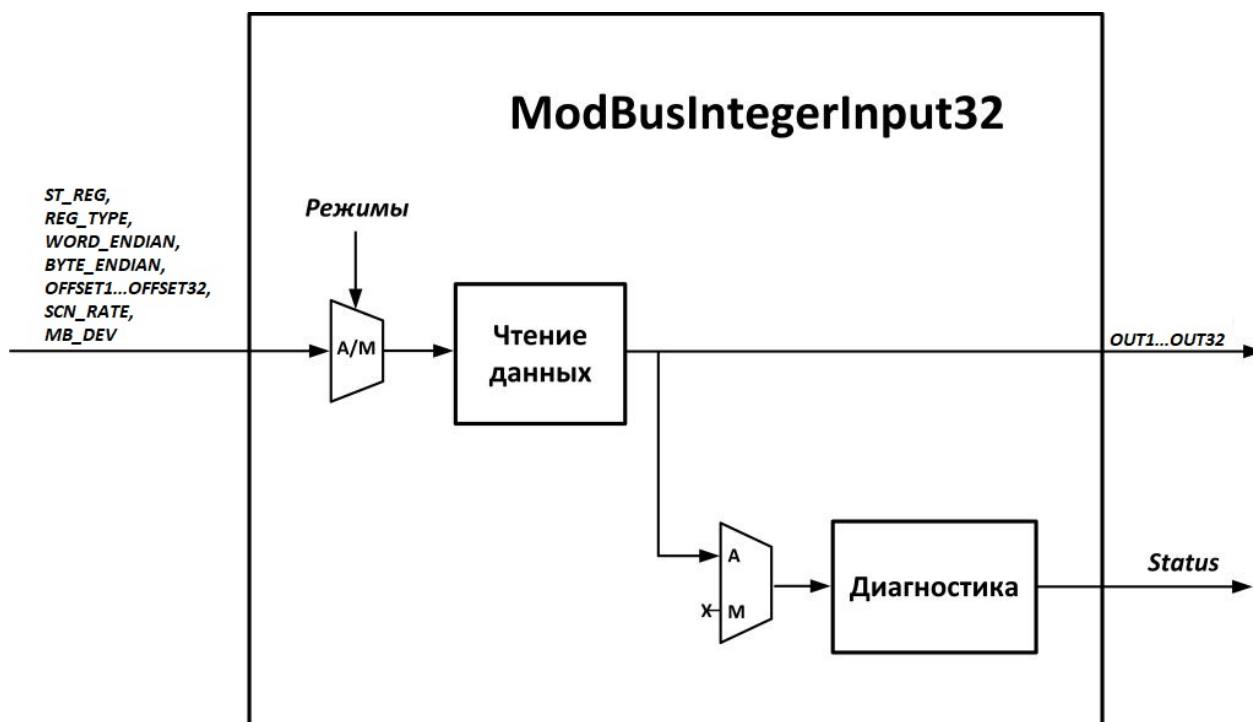
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

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 22. 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 **OUT1...OUT32** ModBusIntegerInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

ST_REG;

REG_TYPE;

WORD_ENDIAN и **BYTE_ENDIAN;**

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 43. Дополнительные атрибуты ModBusIntegerOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_REG	DINT	чтение/ запись	Начальный регистр
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (<i>WordEndian</i>)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (<i>ByteEndian</i>)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс

Имя атрибута	Тип данных	Доступ	Описание атрибута
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
IN1_SRC...IN32_SR	STRING	чтение/ запись	Имя источника входных данных
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение

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 44. 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.

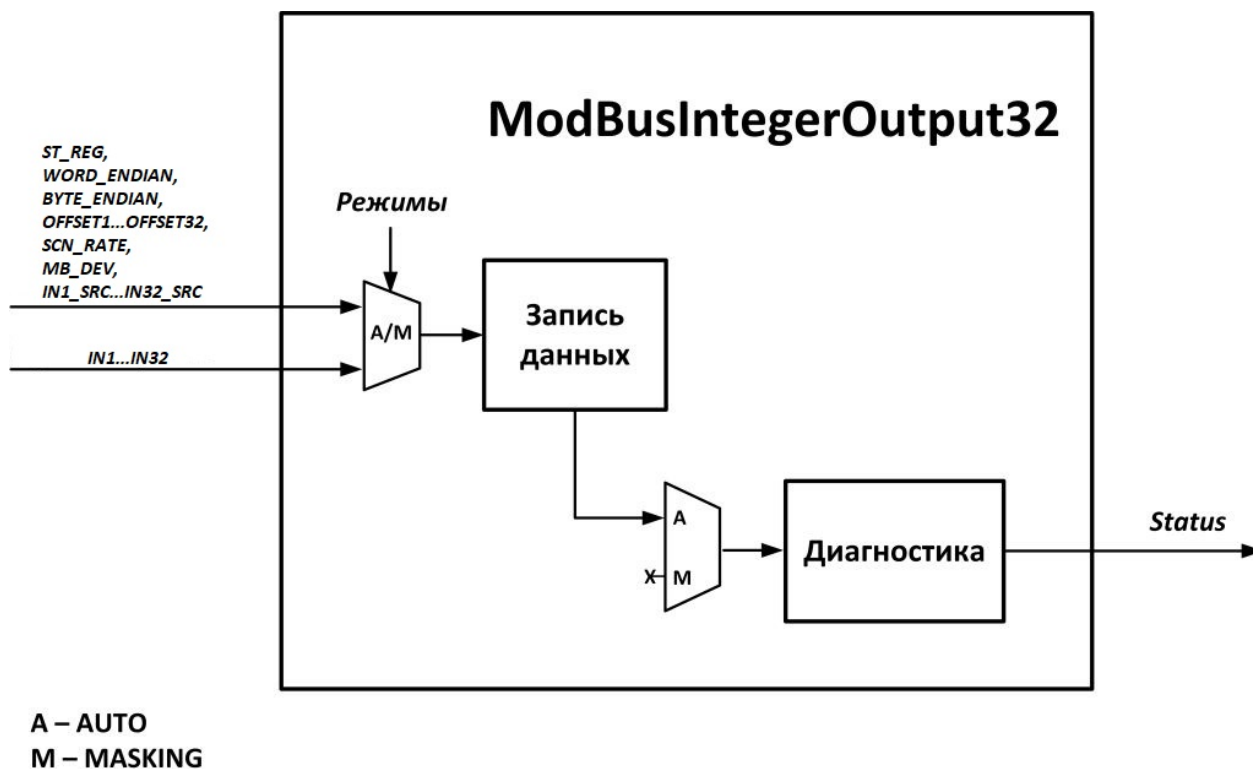


Рисунок 23. Схема процесса обработки данных ModBusIntegerOutput32

6.6.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *IN1...IN32* ModBusIntegerOutput32.

Для записи данных должны быть заданы конфигурационные атрибуты:

ST_REG;

WORD_ENDIAN и *BYTE_ENDIAN*;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV;

IN1_SRC...IN32_SRC.

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 45. Дополнительные атрибуты ModBusIntegerInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	UDINT	чтение	Выход
Конфигурационные атрибуты			
ST_REG	DINT	чтение/ запись	Начальный регистр
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)

Имя атрибута	Тип данных	Доступ	Описание атрибута
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 46. Дополнительное описание значений атрибута Status ModBusUIntegerInput32

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружено 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.

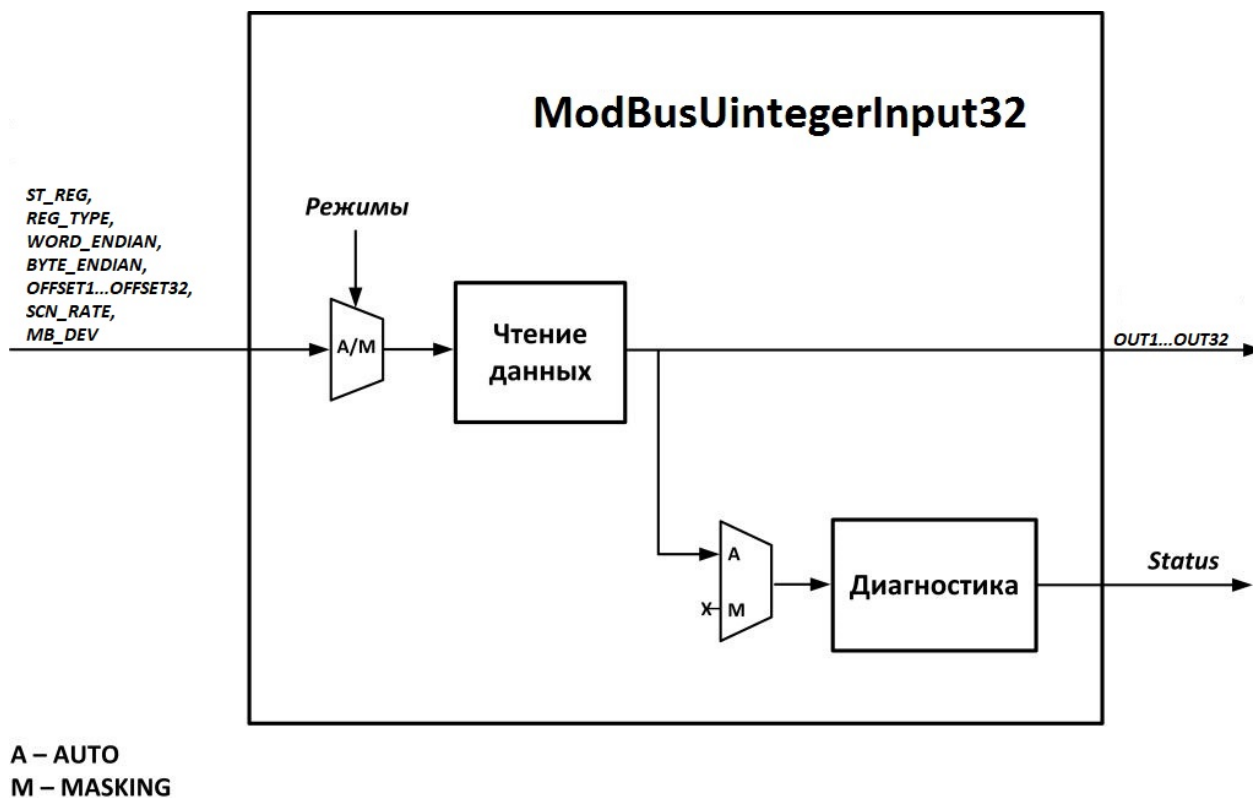


Рисунок 24. Схема процесса обработки данных ModBusUIntegerInput32

6.7.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты **OUT1...OUT32** ModBusUIntegerInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

ST_REG;

REG_TYPE;

WORD_ENDIAN и **BYTE_ENDIAN;**

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 47. Дополнительные атрибуты ModBusUIntegerOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	UDINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	UDINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_REG	DINT	чтение/ запись	Начальный регистр
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (<i>WordEndian</i>)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (<i>ByteEndian</i>)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение

Имя атрибута	Тип данных	Доступ	Описание атрибута
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
IN1_SRC...IN32_SR	STRING	чтение/ запись	Имя источника входных данных
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение

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 48. Дополнительное описание значений атрибута *Status* ModBusUIntegerOutput32

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружено 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.

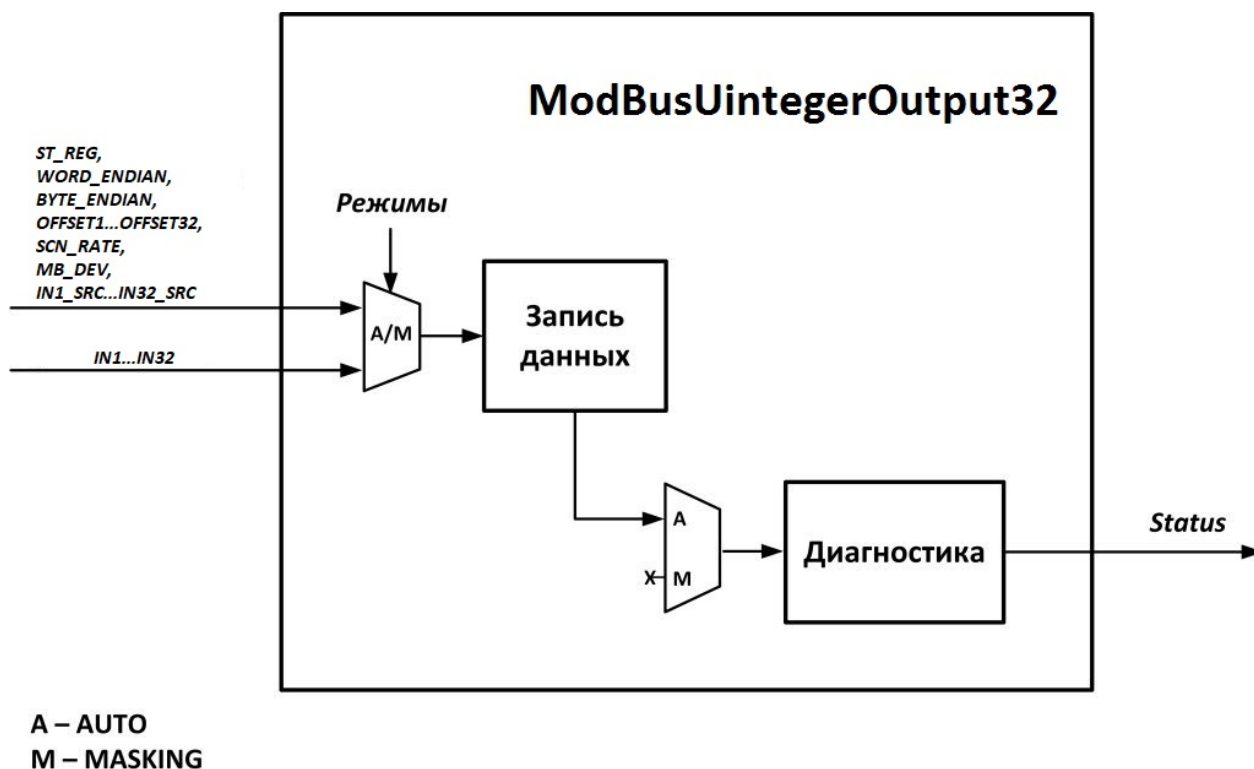


Рисунок 25. Схема процесса обработки данных ModBusUIntegerOutput32

6.8.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *IN1...IN32* ModBusUIntegerOutput32.

Для записи данных должны быть заданы конфигурационные атрибуты:

ST_REG;

WORD_ENDIAN и **BYTE_ENDIAN;**

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV;

IN1_SRC...IN32_SRC.

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 49. Дополнительные атрибуты ModBusBooleanInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
OUT1...OUT32	BOOL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_BIT	DINT	чтение/ запись	Начальный бит
BIT_TYPE	DINT	чтение/ запись	Тип бита

Имя атрибута	Тип данных	Доступ	Описание атрибута
OFFSET1_OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 50. 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.

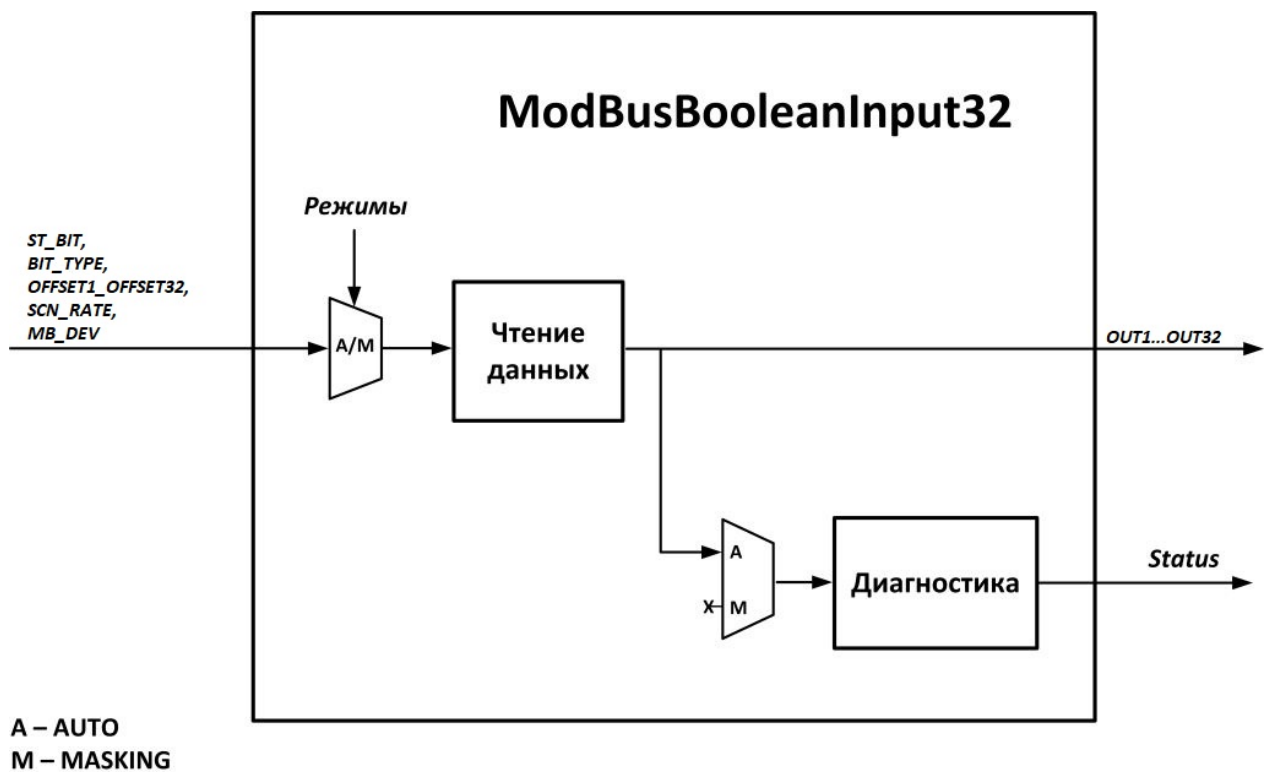


Рисунок 26. Схема процесса обработки данных ModBusBooleanInput32

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:

ST_BIT;

BIT_TYPE;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 51. Дополнительные атрибуты ModBusBooleanOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	BOOL	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	BOOL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_BIT	DINT	чтение/ запись	Начальный бит
OFFSET1_OFFSET	DINT	чтение/ запись	Смещение
SCN_RATE	DINT	чтение/ запись	Период опроса, мс

Имя атрибута	Тип данных	Доступ	Описание атрибута
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
IN1_SRC...IN32_SR	STRING	чтение/ запись	Имя источника входных данных
BIT_TYPE	DINT	чтение/ запись	Тип бита
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение

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 52. 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.

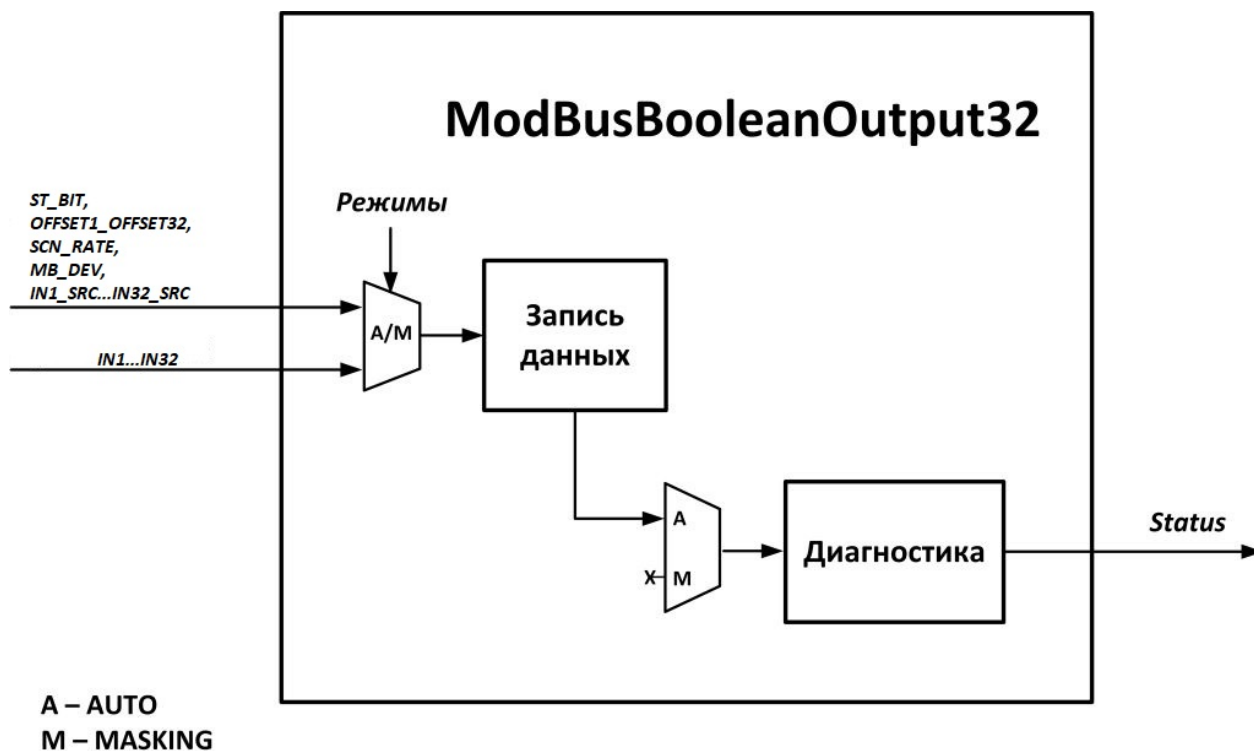


Рисунок 27. Схема процесса обработки данных ModBusBooleanOutput32

6.10.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *IN1...IN32* ModBusBooleanOutput32 в биты устройства Modbus.

Для записи данных должны быть заданы конфигурационные атрибуты:

ST_BIT;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV;

IN1_SRC...IN32_SRC.

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 53. Дополнительные атрибуты ModBusShortInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	DINT	чтение	Выход
Конфигурационные атрибуты			
ST_REG	DINT	чтение/ запись	Начальный регистр
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение

Имя атрибута	Тип данных	Доступ	Описание атрибута
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 54. Дополнительное описание значений атрибута Status ModBusShortInput32

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружено 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").

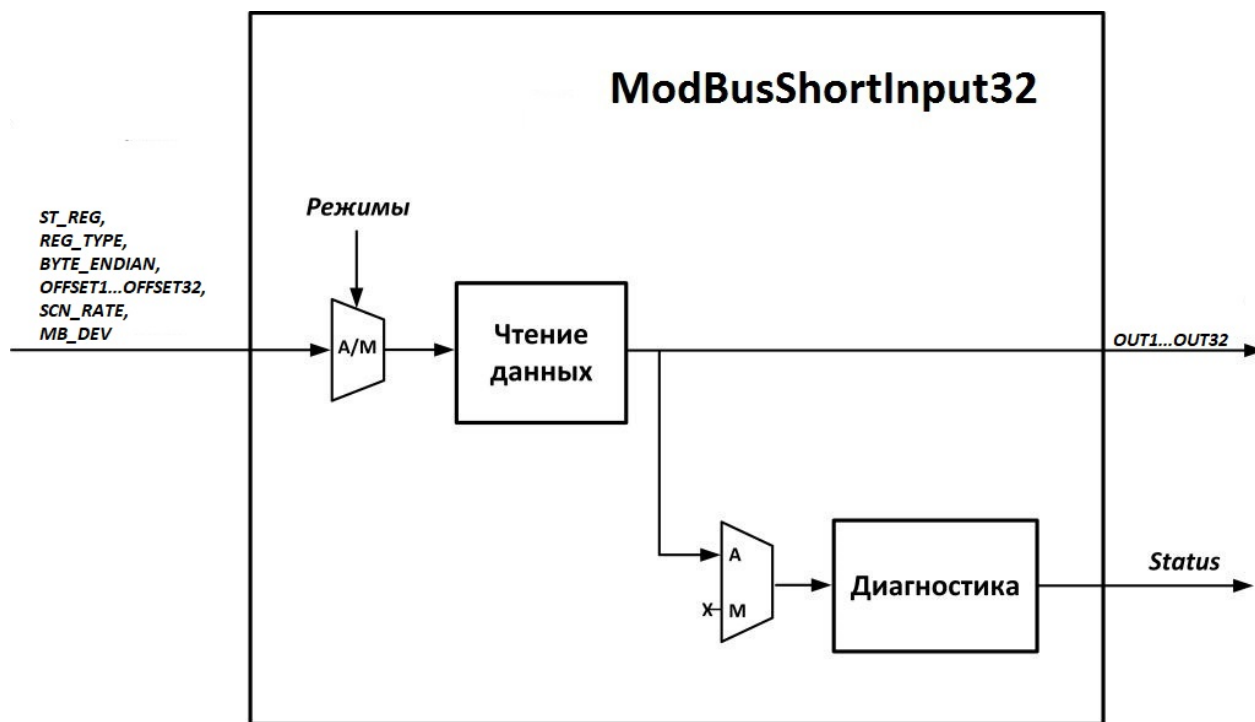
6.11.3. Алгоритм ModBusShortInput32

Выполнение алгоритма ModBusShortInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;

- диагностика коммуникации.



A – AUTO
M – MASKING

Рисунок 28. 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 **OUT1...OUT32** 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:

ST_REG;

REG_TYPE;

BYTE_ENDIAN;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 55. Дополнительные атрибуты ModBusShortOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_REG	DINT	чтение/ запись	Начальный регистр
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
IN1_SRC...IN32_SR	STRING	чтение/ запись	Имя источника входных данных

Имя атрибута	Тип данных	Доступ	Описание атрибута
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение
SCN_RATE	DINT	чтение/ запись	Период опроса, 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".

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

Таблица 56. 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.

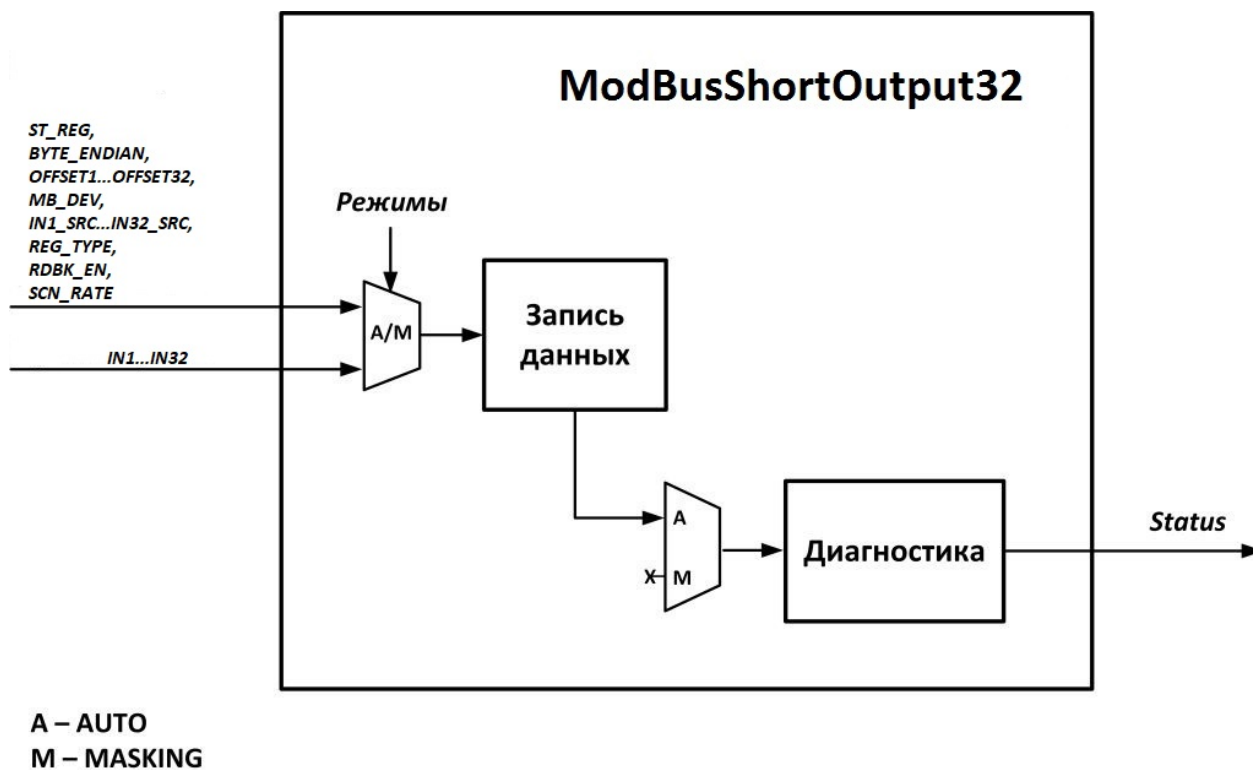


Рисунок 29. Схема процесса обработки данных ModBusShortOutput32

6.12.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *IN1...IN32* ModBusShortOutput32. Каждый входной атрибут *IN1...IN32* усекается до типа SHORT путем отбрасывания двух старших значащих байтов.

Для записи данных должны быть заданы конфигурационные атрибуты:

ST_REG;

BYTE_ENDIAN;

OFFSET1...OFFSET32;

MB_DEV;

IN1_SRC...IN32_SRC.

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 57. Дополнительные атрибуты ModBusUshortInput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT1...OUT32	DINT	чтение	Выход
Конфигурационные атрибуты			
ST_REG	DINT	чтение/ запись	Начальный регистр
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение

Имя атрибута	Тип данных	Доступ	Описание атрибута
SCN_RATE	DINT	чтение/ запись	Период опроса, мс
MB_DEV	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 58. Дополнительное описание значений атрибута Status ModBusUshortInput32

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружено 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").

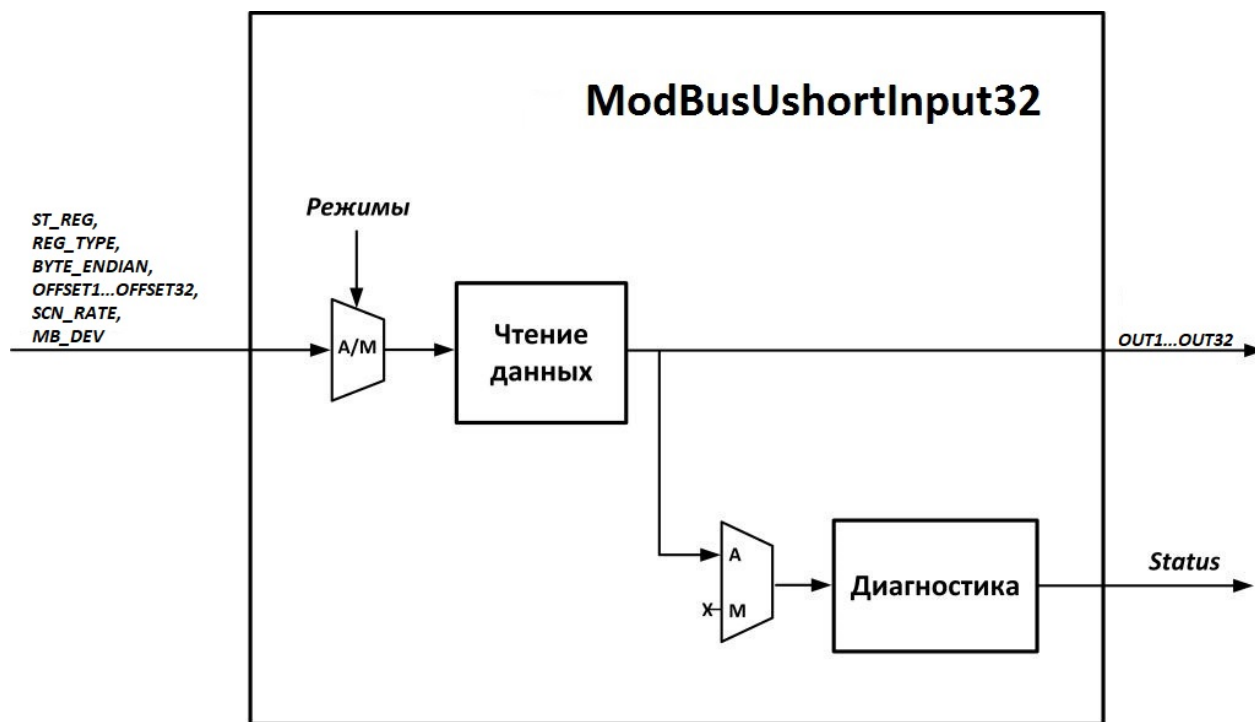
6.13.3. Алгоритм ModBusUshortInput32

Выполнение алгоритма ModBusUshortInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;

- диагностика коммуникации.



A – AUTO
M – MASKING

Рисунок 30. 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 **OUT1...OUT32** 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:

ST_REG;

REG_TYPE;

BYTE_ENDIAN;

OFFSET1...OFFSET32;

SCN_RATE;

MB_DEV.

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 59. Дополнительные атрибуты ModBusUshortOutput32

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN32	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT1...OUT32	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
ST_REG	DINT	чтение/ запись	Начальный регистр
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)
OFFSET1...OFFSET	DINT	чтение/ запись	Смещение
MB_DEV	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
IN1_SRC...IN32_SR	STRING	чтение/ запись	Имя источника входных данных

Имя атрибута	Тип данных	Доступ	Описание атрибута
REG_TYPE	DINT	чтение/ запись	Тип регистра (Input или Holding)
RDBK_EN	BOOL	чтение/ запись	Включить обратное чтение
SCN_RATE	DINT	чтение/ запись	Период опроса, 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".

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

Таблица 60. 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.

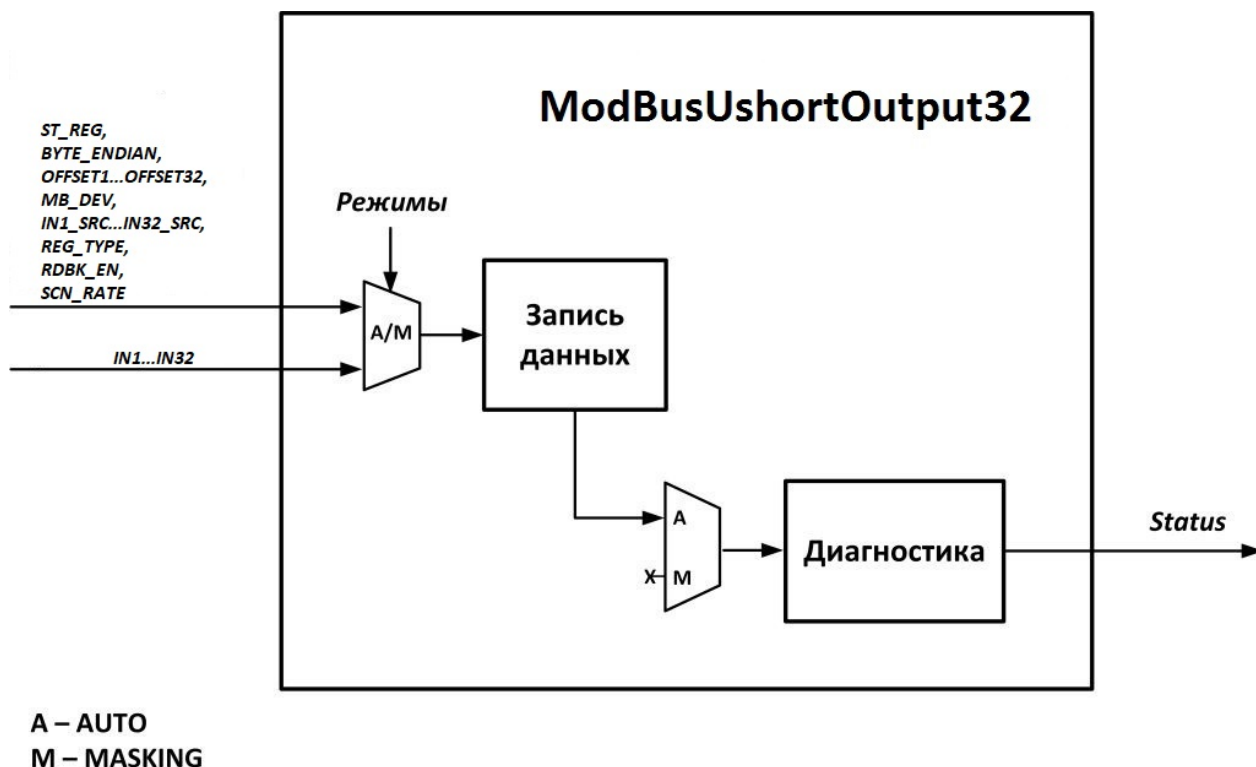


Рисунок 31. Схема процесса обработки данных ModBusUshortOutput32

6.14.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *IN1...IN32* ModBusUshortOutput32. Каждый входной атрибут *IN1...IN32* усекается до типа SHORT путем отбрасывания двух старших значащих байтов.

Для записи данных должны быть заданы конфигурационные атрибуты:

ST_REG;

BYTE_ENDIAN;

OFFSET1...OFFSET32;

MB_DEV;

IN1_SRC...IN32_SRC.

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 Container16int

7.1.1. Атрибуты Container16int

Container16int имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 61. Дополнительные атрибуты Container16int

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
MV	DINT	чтение/запись	Ручное значение
CONV_TYPE	DINT	чтение/запись	Преобразование входа



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.1.2. Режимы Container16int

Режимы Container16int:

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

7.2. Функциональный блок Container16uInt

7.2.1. Атрибуты Container16uInt

Container16uInt имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 62. Дополнительные атрибуты Container16uInt

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
MV	DINT	чтение/запись	Ручное значение



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.2.2. Режимы Container16uInt

Режимы Container16uInt:

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

7.3. Функциональный блок Container32int

7.3.1. Атрибуты Container32int

Container32int имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 63. Дополнительные атрибуты Container32int

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	DINT	чтение	Вход
<i>Выходные атрибуты</i>			
OUT	DINT	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
MV	DINT	чтение/запись	Ручное значение



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.3.2. Режимы Container32int

Режимы Container32int:

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

7.4. Функциональный блок Container32uInt

7.4.1. Атрибуты Container32uInt

Container32uInt имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 64. Дополнительные атрибуты Container32uInt

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	UDINT	чтение	Вход

Имя атрибута	Тип данных	Доступ	Описание атрибута
Выходные атрибуты			
OUT	UDINT	чтение	Выход
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник
MV	UDINT	чтение/запись	Ручное значение



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.4.2. Режимы Container32uInt

Режимы Container32uInt:

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

7.5. Функциональный блок ContainerBool

7.5.1. Атрибуты ContainerBool

ContainerBool имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 65. Дополнительные атрибуты ContainerBool

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	BOOL	чтение	Вход
Выходные атрибуты			
OUT	BOOL	чтение	Выход
NOUT	BOOL	чтение	Инвертированный выход

Имя атрибута	Тип данных	Доступ	Описание атрибута
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник
MV	BOOL	чтение/запись	Ручное значение



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.5.2. Режимы ContainerBool

Режимы ContainerBool:

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

7.6. Функциональный блок ContainerReal

7.6.1. Атрибуты ContainerReal

ContainerReal имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 66. Дополнительные атрибуты ContainerReal

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	REAL	чтение	Вход
Выходные атрибуты			
OUT	REAL	чтение	Выход
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник
MV	REAL	чтение/запись	Ручное значение



Внимание: Значение атрибута *MV* записывается на диск и восстанавливается после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

7.6.2. Режимы ContainerReal

Режимы ContainerReal:

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

7.7. Функциональный блок ControlButton

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

7.7.1. Атрибуты ControlButton

ControlButton имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 67. Дополнительные атрибуты ControlButton

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

Имя атрибута	Тип данных	Доступ	Описание атрибута
EN_OPTS	USINT	чтение/запись	Опции при отключенном управлении

7.7.2. Режимы ControlButton

Режимы ControlButton:

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

7.7.3. Алгоритм ControlButton

ControlButton имеет два типа управления

- импульс;
- удержание.

Когда на ControlButton 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.

С помощью атрибута **EN_OPTS** настраивается режим работы функционального блока: при отсутствии команды разрешения (EN=0) при типе воздействия HOLD

- None (0) – команда не отправляется;
- AllowFalse (1) – разрешается отправлять команду false;
- AllowTrue (2) – разрешается отправлять команду true.

7.8. 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.

7.8.1. Атрибуты AnalogInputPoint

AnalogInputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 68. Дополнительные атрибуты AnalogInputPoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN	REAL	чтение	Входное значение в процентах
NH_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
NH_ACT	BOOL	чтение	Сработка аларма до разрешающего бита NH_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
NH_NACT	BOOL	чтение	Инвертированный NH_ACT, имеет статус PV
HI_NACT	BOOL	чтение	Инвертированный HI_ACT, имеет статус PV
LO_NACT	BOOL	чтение	Инвертированный LO_ACT, имеет статус PV

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL_NACT	BOOL	чтение	Инвертированный LL_ACT, имеет статус PV
Конфигурационные атрибуты			
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
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 HH

Имя атрибута	Тип данных	Доступ	Описание атрибута
HI_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HI
LO_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LO
LL_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LL
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
F_COEFF	REAL	чтение/ запись	Коэффициент фильтрации
F_EN	BOOL	чтение/ запись	Фильтрация входа
F_BUFF	USINT	чтение/ запись	Буфер медианного фильтра
engineerUnit	STRING	чтение/ запись	Единица измерения
Сигнализационные атрибуты			
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
HH_Priority	BOOL	чтение	Приоритет сигнала ВАУ
HI_Priority	BOOL	чтение	Приоритет сигнала ВПУ
LO_Priority	BOOL	чтение	Приоритет сигнала НПУ
LL_Priority	BOOL	чтение	Приоритет сигнала НАУ

7.8.2. Режимы AnalogInputPoint

Режимы AnalogInputPoint:

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

7.8.3. Алгоритм AnalogInputPoint

Выполнение алгоритма AnalogInputPoint зависит от установленного режима.

Алгоритм содержит следующие элементы:

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

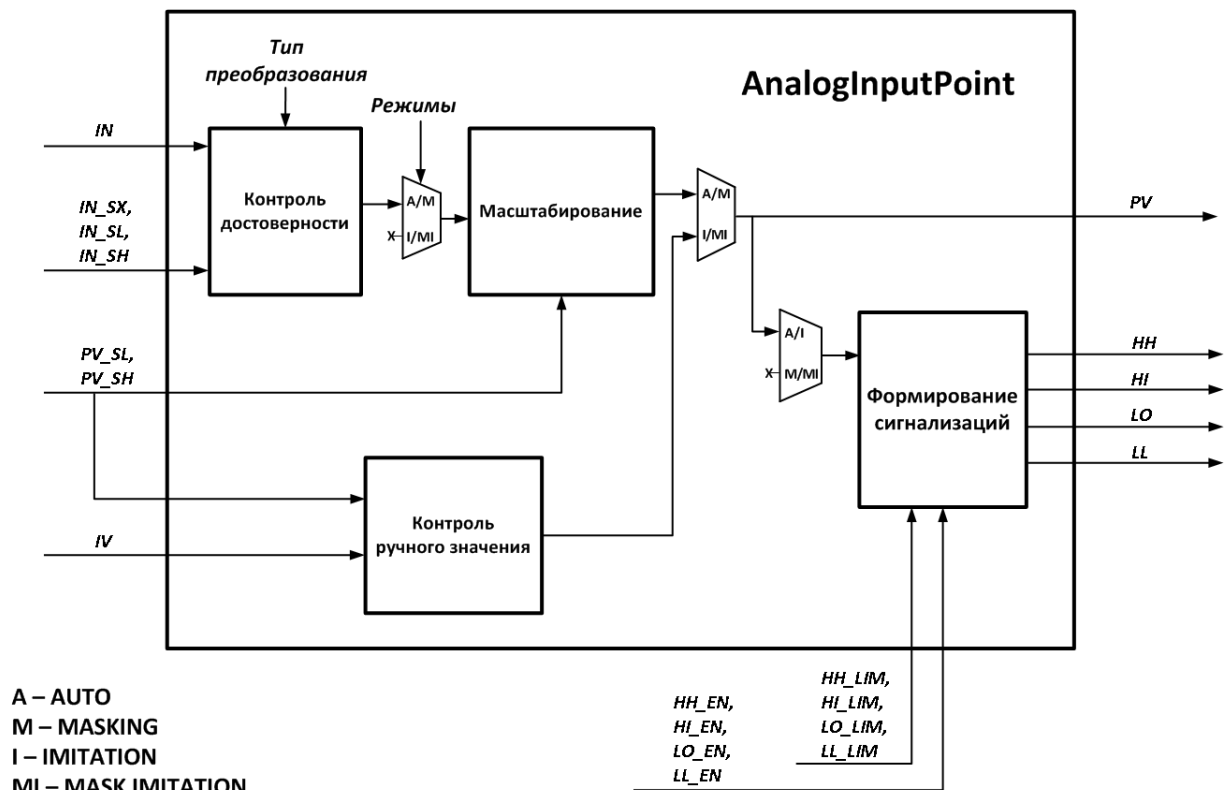


Рисунок 32. Схема процесса обработки данных AnalogInputPoint

7.8.3.1. 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.

7.8.3.2. 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:

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

7.8.3.3. 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.

7.8.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 10.1. [Формирование сигнализаций Библиотеки](#).

7.8.3.5. Фильтрация шума сигнала

На данном этапе можно включить фильтрацию аналогового входа для защиты от шумов с помощью атрибута *F_EN*.

Фильтрация шумов в модуле AnalogInputPoint происходит с помощью сочетания медианного фильтра и бегущего среднего.

F_BUFF (значение от 1-10) количество значений, которые сохраняются в памяти блока и по которым высчитывается медианное среднее.

Затем модуль берет дельту предыдущего и текущего входного значения модуля, умножает на *F_COEFF*. Полученное значение прибавляется к текущему входному значению.

7.9. Входной аналоговый блок AnalogInputPointExtended

7.9.1. Атрибуты AnalogInputPointExtended

AnalogInputPointExtended имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 69. Дополнительные атрибуты AnalogInputPointExtended

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN	REAL	чтение	Входное значение в процентах
HH_CND_EN	BOOL	чтение	Включение HiHi (вход). Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
HI_CND_EN	BOOL	чтение	Включение Hi (вход). Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
LO_CND_EN	BOOL	чтение	Включение Lo (вход). Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
LL_CND_EN	BOOL	чтение	Включение LoLo (вход). Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
HH2_CND_EN	BOOL	чтение	Включение HiHi 2 (вход)
HI2_CND_EN	BOOL	чтение	Включение Hi 2 (вход)
LO2_CND_EN	BOOL	чтение	Включение Lo 2 (вход)
LL2_CND_EN	BOOL	чтение	Включение LoLo 2 (вход)
PV	REAL	чтение	Выходное значение
HH_ACT	BOOL	чтение	Сработка 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
HH2_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HH2_EN. Учитывает гистерезис
HI2_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HI2_EN. Учитывает гистерезис
LO2_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LO2_EN. Учитывает гистерезис
LL2_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LL2_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
HH2_NACT	BOOL	чтение	Инвертированный HH2_ACT, имеет статус PV
HI2_NACT	BOOL	чтение	Инвертированный HI2_ACT, имеет статус PV
LO2_NACT	BOOL	чтение	Инвертированный LO2_ACT, имеет статус PV

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL2_NACT	BOOL	чтение	Инвертированный LL2_ACT, имеет статус PV
Конфигурационные атрибуты			
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	Уровень верхней аварийной уставки в инженерных единицах
HI_LIM	REAL	чтение/ запись	Уровень верхней предупредительной уставки в инженерных единицах
LO_LIM	REAL	чтение/ запись	Уровень нижней предупредительной уставки в инженерных единицах
LL_LIM	REAL	чтение/ запись	Уровень нижней аварийной уставки в инженерных единицах
HH2_LIM	REAL	чтение/ запись	Уровень верхней аварийной уставки 2
HI2_LIM	REAL	чтение/ запись	Уровень верхней предупредительной уставки 2
LO2_LIM	REAL	чтение/ запись	Уровень нижней предупредительной уставки 2
LL2_LIM	REAL	чтение/ запись	Уровень нижней аварийной уставки 2

Имя атрибута	Тип данных	Доступ	Описание атрибута
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	Включить проверку сигнализации HiHiAlarm
HI_EN	BOOL	чтение/ запись	Включить проверку сигнализации HiAlarm
LO_EN	BOOL	чтение/ запись	Включить проверку сигнализации LoAlarm
LL_EN	BOOL	чтение/ запись	Включить проверку сигнализации LoLoAlarm
HH2_EN	BOOL	чтение/ запись	Включить проверку сигнализации HiHiAlarm2
HI2_EN	BOOL	чтение/ запись	Включить проверку сигнализации HiAlarm2
LO2_EN	BOOL	чтение/ запись	Включить проверку сигнализации LoAlarm2
LL2_EN	BOOL	чтение/ запись	Включить проверку сигнализации LoLoAlarm2
CONV_TYPE	DINT	чтение/ запись	Преобразование входного значения: Linear - линейное, SQRT - квадратичное
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
HH2_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HH 2
HI2_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HI 2
LO2_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LO 2
LL2_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LL 2
F_COEFF	REAL	чтение/ запись	Коэффициент фильтрации
F_EN	BOOL	чтение/ запись	Фильтрация входа
F_BUFF	USINT	чтение/ запись	Буфер медианного фильтра
engineerUnit	STRING	чтение/ запись	Единица измерения
Сигнализационные атрибуты			
HH	BOOL	чтение	Сигнал верхней аварийной уставки. Переход выходного значения в диапазон верхних аварийных значений. Аларм HH == HH_ACT && HH_CND_EN && HH_EN
HI	BOOL	чтение	Сигнал верхней предупредительной уставки. Переход выходного значения в диапазон верхних предупредительных значений. Аларм HI == HI_ACT && HI_CND_EN && HI_EN

Имя атрибута	Тип данных	Доступ	Описание атрибута
LO	BOOL	чтение	Сигнал нижней предупредительной уставки. Переход выходного значения в диапазон нижних предупредительных значений. Аларм LO == LO_ACT && LO_CND_EN && LO_EN
LL	BOOL	чтение	Сигнал нижней аварийной уставки. Переход выходного значения в диапазон нижних аварийных значений. Аларм LL == LL_ACT && LL_CND_EN && LL_EN
HH2	BOOL	чтение	Сигнал верхней аварийной уставки 2
HI2	BOOL	чтение	Сигнал верхней предупредительной уставки 2
LO2	BOOL	чтение	Сигнал нижней предупредительной уставки 2
LL2	BOOL	чтение	Сигнал нижней аварийной уставки 2
HH_Priority	BOOL	чтение	Приоритет сигнала ВАУ
HI_Priority	BOOL	чтение	Приоритет сигнала ВПУ
LO_Priority	BOOL	чтение	Приоритет сигнала НПУ
LL_Priority	BOOL	чтение	Приоритет сигнала НАУ
HH2_Priority	BOOL	чтение	Приоритет сигнала ВАУ2
HI2_Priority	BOOL	чтение	Приоритет сигнала ВПУ2
LO2_Priority	BOOL	чтение	Приоритет сигнала НПУ2
LL2_Priority	BOOL	чтение	Приоритет сигнала НАУ2
Диагностические атрибуты			
FV	REAL	чтение	Значение с поля

7.9.2. Режимы AnalogInputPointExtended

Режимы AnalogInputPointExtended:

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

7.9.3. Алгоритм AnalogInputPointExtended

Выполнение алгоритма AnalogInputPointExtended зависит от установленного режима.

Алгоритм содержит следующие элементы:

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

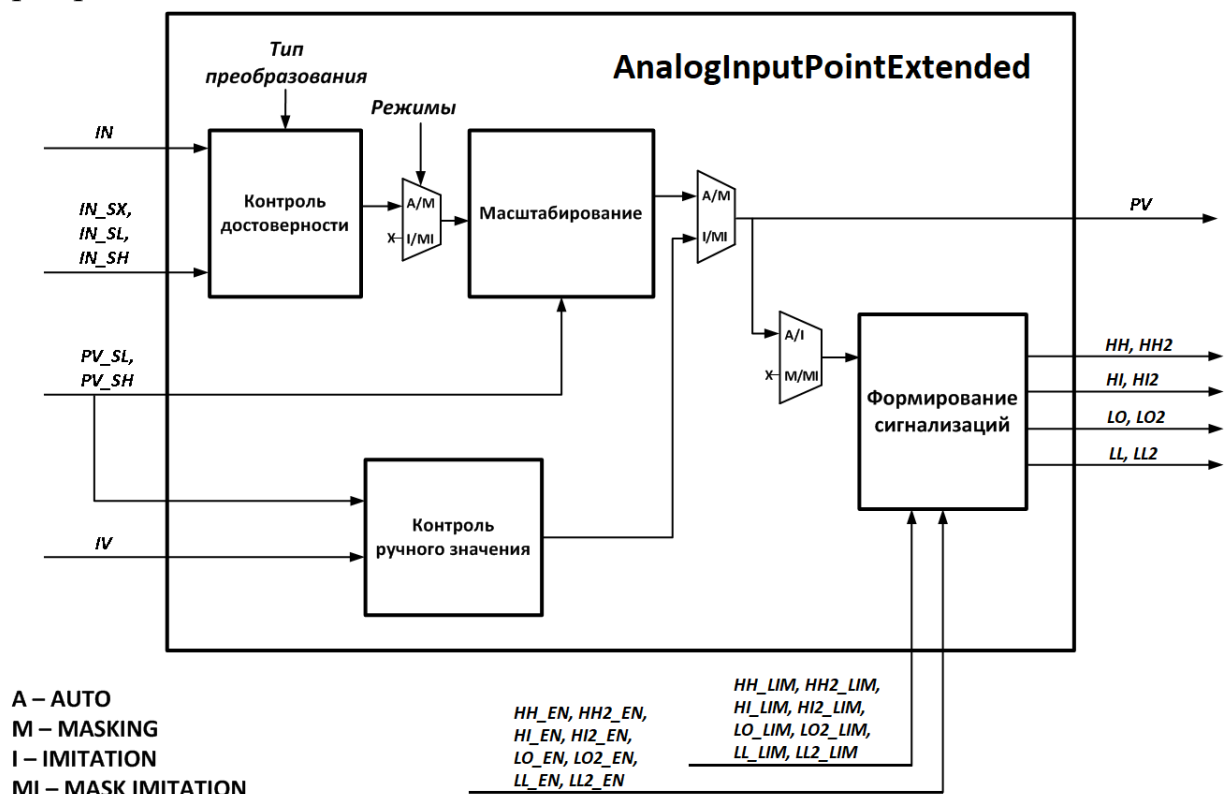


Рисунок 33. Схема процесса обработки данных AnalogInputPointExtended

7.9.3.1. 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.

7.9.3.2. 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:

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

7.9.3.3. 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.

7.9.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 10.1. [Формирование сигнализаций Библиотеки](#).

7.10. 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.

7.10.1. Атрибуты AnalogOutputPoint

AnalogOutputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 70. Дополнительные атрибуты AnalogOutputPoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные/выходные атрибуты			
IN	REAL	чтение	Входное значение в инженерных единицах
FLT	BOOL	чтение	Значение, отображающее состояние физического канала приема данных
OUT	REAL	чтение	Выходное значение
CASO	BOOL	чтение	Готовность к управлению выход
Конфигурационные атрибуты			
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
IN_SL	REAL	чтение/ запись	Нижняя граница входного сигнала
IN_SH	REAL	чтение/ запись	Верхняя граница входного сигнала
OUT_SL	REAL	чтение/ запись	Нижняя граница выходного сигнала
OUT_SH	REAL	чтение/ запись	Верхняя граница выходного сигнала
FV	REAL	чтение	Значение поля
INVERT	BOOL	чтение/ запись	Инвертирование выхода

7.10.2. Режимы AnalogOutputPoint

Режимы AnalogOutputPoint:

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

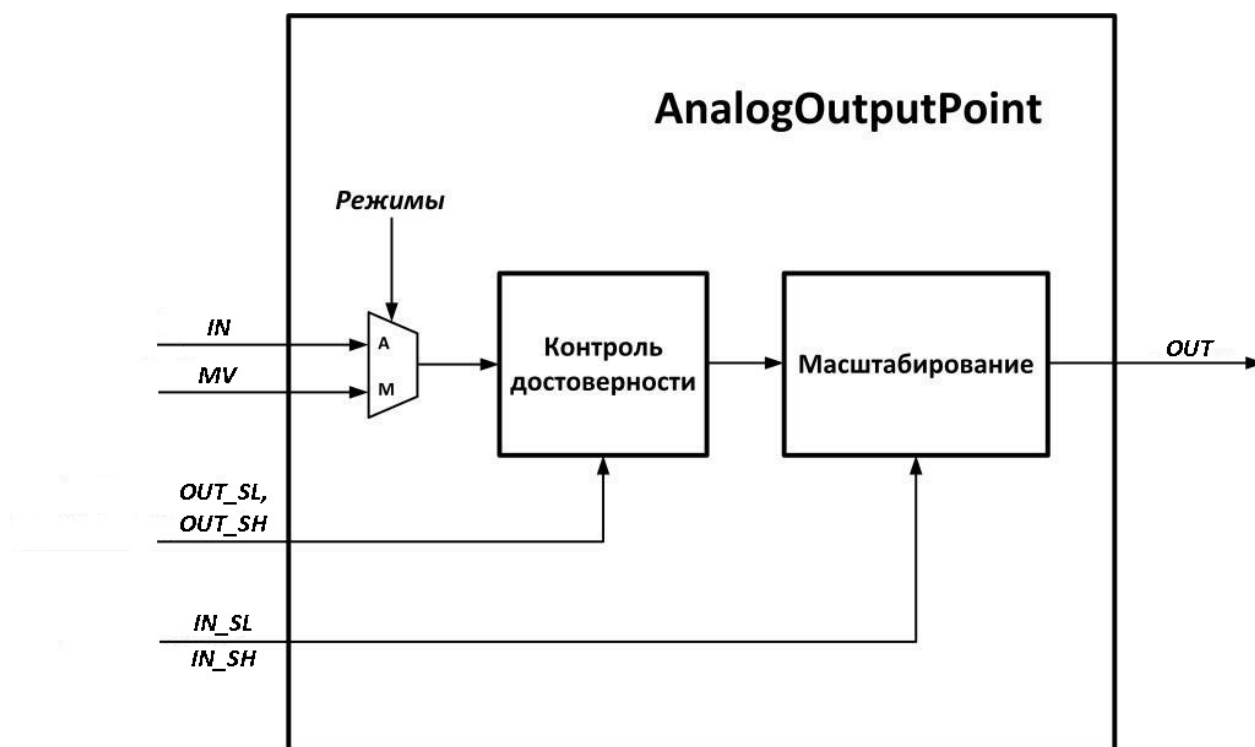
В режиме MASKING сигнализационный флаг об ошибке **Bad** не формируется.

7.10.3. Алгоритм AnalogOutputPoint

Выполнение алгоритма AnalogOutputPoint зависит от установленного режима.

Алгоритм содержит следующие элементы:

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



A – AUTO
M – MANUAL

Рисунок 34. Схема процесса обработки данных AnalogOutputPoint

7.10.3.1. 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).

7.10.3.2. 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*

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

где

$$k = \frac{OUT_SH - OUT_SL}{IN_SH - IN_SL}$$

или

если **INVERT** принимает значение **true**

$$OUT = OUT_SH - k \cdot (IN - IN_SL),$$

где

$$k = \frac{OUT_SH - OUT_SL}{IN_SH - IN_SL}$$

где **IN_SH** – верхняя граница входного сигнала,

IN_SL – нижняя граница входного сигнала,

OUT_SH – верхняя граница выходного сигнала,

OUT_SL – нижняя граница выходного сигнала.

7.10.3.3. 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).

Атрибут **CASO** в режиме AUTO принимает значение **true**. Его значение изменяется на **false**, если:

- режим работы переключён на MANUAL;
- на входной атрибут **FLT** поступила ошибка канала;
- статус функционального блока изменён на состояние, отличное от ОК.

7.11. Дискретный блок DiscretePoint

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

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

7.11.1. Атрибуты DiscretePoint

DiscretePoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 71. Дополнительные атрибуты DiscretePoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные/выходные атрибуты			
IN	BOOL	чтение	Входное значение
DISC_CND_EN	BOOL	чтение	Вход включения/отключения алармов
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
Конфигурационные атрибуты			
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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_SRC	STRING	чтение/ запись	Включение источника входного сигнала
<i>Сигнализационные атрибуты</i>			
DISC_ALM	BOOL	чтение	Сигнализация

7.11.2. Режимы DiscretePoint

Режимы DiscretePoint:

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

7.11.3. Алгоритм DiscretePoint

Выполнение алгоритма DiscretePoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- инверсия;
- фильтрация дребезга;
- формирование сигнализаций.

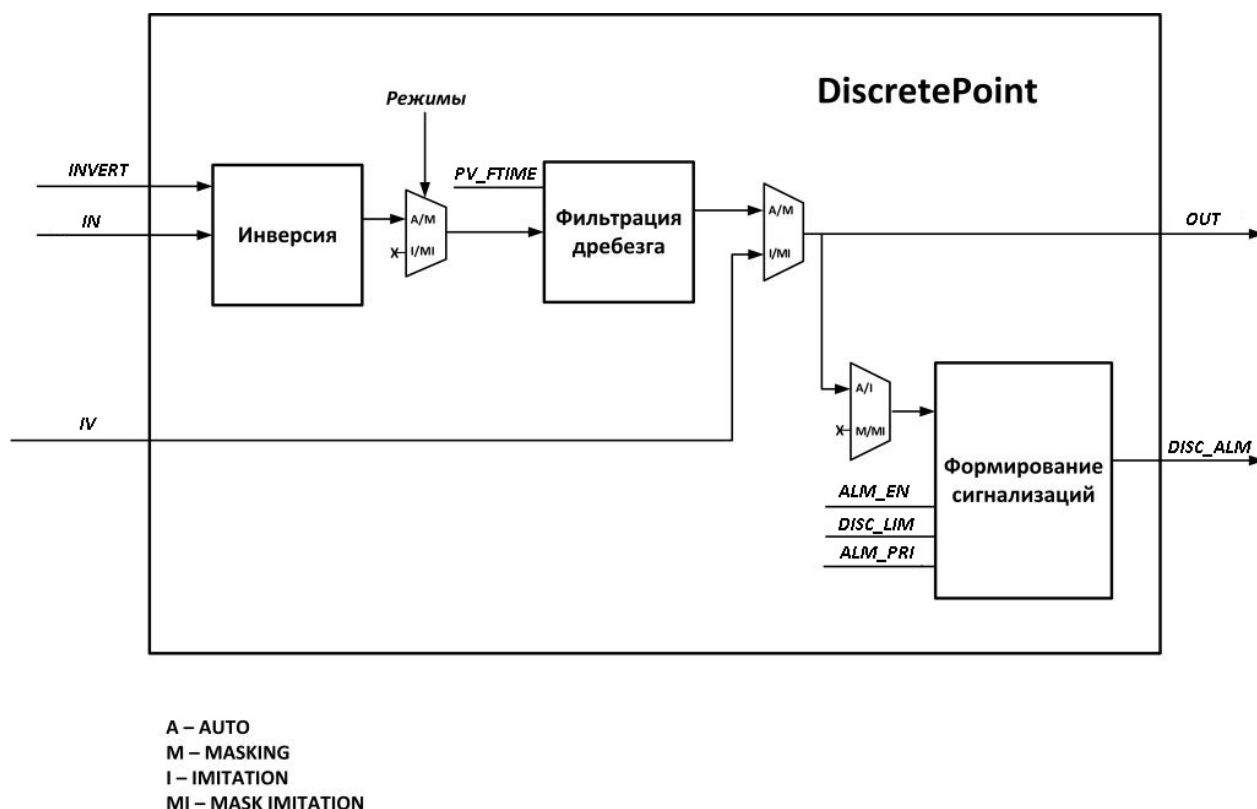


Рисунок 35. Схема процесса обработки данных DiscretePoint

7.11.3.1. Инверсия

При инверсии происходит переключение входного атрибута на противоположное значение.

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

7.11.3.2. 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.

7.11.3.3. 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**.

Если вход **DISC_CND_EN** используется, то формирование сигнализаций будет работать при условии **ALM_EN** = 1 (true) и **DISC_CND_EN** = 1 (true).

Атрибут **ALM_PRI** выставляет приоритет аларму от 0 до 999.

7.12. 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.

7.12.1. Атрибуты PIDController

PIDController имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 72. Дополнительные атрибуты PIDController

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	REAL	чтение	Входное значение
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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.
RSPI	REAL	чтение	Вход внешнего задания
RMVI	REAL	чтение	Вход внешнего выхода
Выходные атрибуты			
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
MOUT	REAL	чтение/запись	Ручное значение
INTEGRAL	REAL	чтение/запись	Накопленный интегральный компонент
PV	REAL	чтение	Выходное значение
CASO	BOOL	чтение	Готовность к управлению выход

Имя атрибута	Тип данных	Доступ	Описание атрибута
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
RSP	REAL	чтение	Внешнее задание
RMV	REAL	чтение	Задание внешнего выхода
Конфигурационные атрибуты			
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
SP_MIN	REAL	чтение/запись	Нижний предел допустимого диапазона управляющих значений, инженерные единицы

Имя атрибута	Тип данных	Доступ	Описание атрибута
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 ⁻¹
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
PV_DB	REAL	чтение/запись	Зона нечувствительности регулятора
PV_DB_HYS	REAL	чтение/запись	Гистерезис зоны нечувствительности
STB_SRC	STRING	чтение/запись	Источник стробирования
STB_EN	BOOL	чтение/запись	Включение стробирования

Имя атрибута	Тип данных	Доступ	Описание атрибута
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
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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
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
REM_SP_PRC	BOOL	чтение/ запись	Каскадная уставка в процентах
ALGORITHM	USINT	чтение/ запись	Алгоритм расчета ПИД управления
OUT_IL	REAL	чтение/ запись	Максимальная скорость приращения выхода, %/сек

Имя атрибута	Тип данных	Доступ	Описание атрибута
OUT_MSL	REAL	чтение/ запись	Нижний предел шкалы управляющего выхода
OUT_MSH	REAL	чтение/ запись	Верхний предел шкалы управляющего выхода
F_BUFF	USINT	чтение/ запись	Буфер медианного фильтра
RSPI_SRC	STRING	чтение/ запись	Источник внешнего задания
RMVI_SRC	STRING	чтение/ запись	Источник внешнего выхода
EXT_MODE	USINT	чтение/ запись	Режим внешнего управления
F_EN	BOOL	чтение/ запись	Фильтрация входа
F_COEFF	REAL	чтение/ запись	Коэффициент фильтрации
engineerUnit	STRING	чтение/ запись	Единица измерения
Сигнализационные атрибуты			
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL	BOOL	чтение	Сигнал НАУ - выходное значение перешло в диапазон нижних аварийных значений. Аларм LL == LL_ACT && LL_CND_EN && LL_EN
HH_Priority	BOOL	чтение	Приоритет сигнализации ВАУ
HI_Priority	BOOL	чтение	Приоритет сигнализации ВПУ
LO_Priority	BOOL	чтение	Приоритет сигнализации НПУ
LL_Priority	BOOL	чтение	Приоритет сигнализации НАУ
CH_ERR	BOOL	чтение	Ошибка канала
CH_ERR_Priority	BOOL	чтение	Приоритет сигнализации ошибка канала
<i>Диагностические атрибуты</i>			
RDBK	REAL	чтение	Обратная связь
TRK	BOOL	чтение	Режим отслеживания

7.12.2. Режимы PIDController

Режимы PIDController:

1. AUTO ("Автоматический").

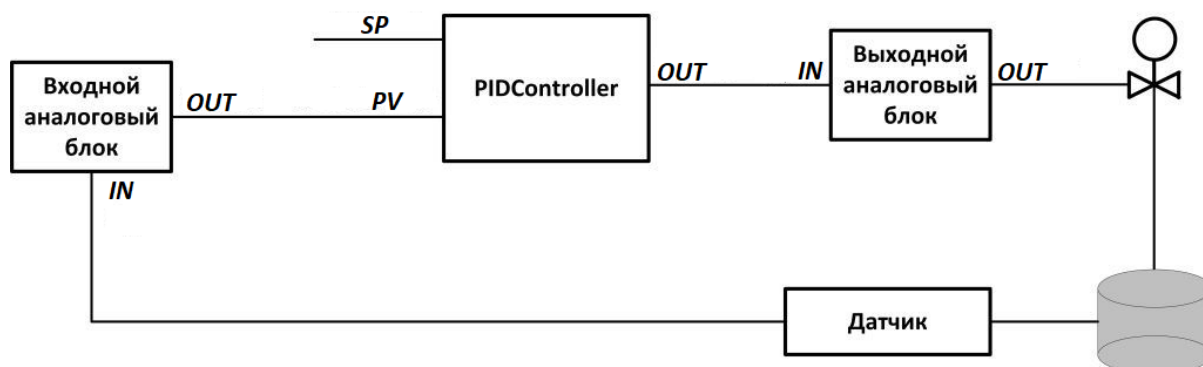


Рисунок 36. PIDController в режиме AUTO

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:

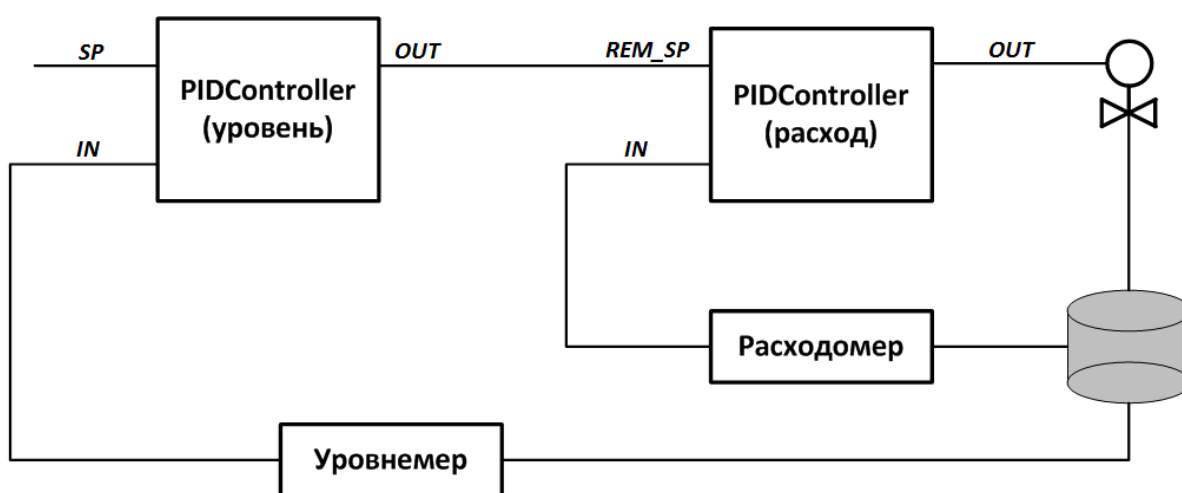


Рисунок 37. PIDController в режиме CASCADE

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").

7.12.3. Алгоритм PIDController

Выполнение алгоритма PIDController зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

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

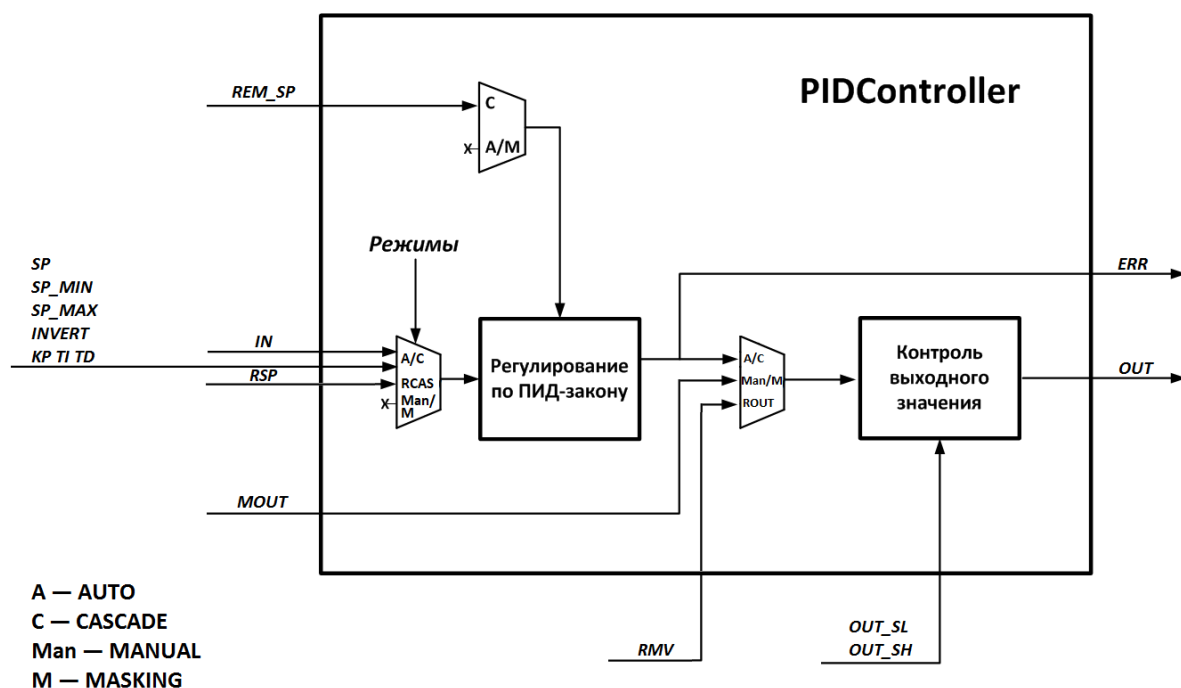


Рисунок 38. Схема процесса обработки данных PIDController

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***.

Режимы внешнего управления (***EXT_MODE***) бывают следующими:

- None
- RCAS - Режим внешнего каскада
- ROUT - Режим внешнего выхода

Когда блок находится в режиме внешнего каскада (RCAS), значение задания ***SP***, ***SP_TARG*** автоматически приравнивается к внешнему заданию ***RSP*** или ***RSPI*** (если в ***RSPI_SRC*** есть источник данных).

При работе в режиме RCAS для выполнения расчетов используется выбранный алгоритм ПИД-управления.

В режиме внешнего выхода (ROUT) обработка вычислений остановлена. Значение выхода ***OUT*** приравнивается к внешнему выходу ***RMV*** или ***RMVI*** (если в ***RMVI_SRC*** есть источник данных).

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.

Зона нечувствительности регулятора задается в процентах по формуле:

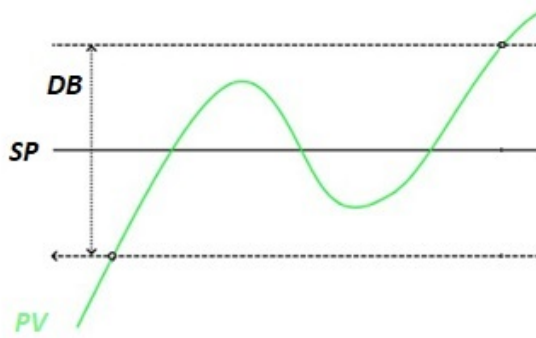
$$\frac{PV_{SH} - PV_{SL}}{100\%} \times PV_{DB}$$

Гистерезис зоны нечувствительности задается по формуле:

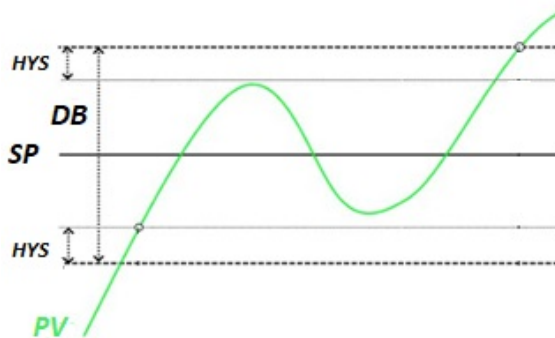
$$\frac{PV_{SH} - PV_{SL}}{100\%} \times \frac{PV_{DB_HYS}}{2}$$

Зона нечувствительности регулятора не может быть меньше гистерезиса зоны нечувствительности (***PV_DB*** ≥ ***PV_DB_HYS***):

- Если задана зона нечувствительности регулятора, но не задан гистерезис:



- Если задана зона нечувствительности регулятора и гистерезис меньше зоны нечувствительности:



7.12.3.1. 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.

Регулятор поддерживает четыре алгоритма ПИД управления. Алгоритм расчета задается в настройках ФБ через конфигурационный атрибут **ALGORITHM**. Расчет управляющего воздействия производится в соответствии с выбранным алгоритмом по формулам:

PID (при добавлении блока выбирается по умолчанию)

$$OUT = OUT_s + Inv \times \left(KP \times ERR_n + intsum_n + \frac{TD \times \Delta ERR_n}{\Delta t} \right), \text{ где}$$

OUT_s — значение выхода в момент перехода из MAN в AUTO или CAS;

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

KP — пропорциональный коэффициент;

ERR_n — отклонение, $ERR_n = SP_n - PV_n$;

SP_n — переменная задания;

PV_n — переменная процесса;

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

n — номер системного цикла;

Δt — период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$.

PID_M

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta ERR_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(\Delta ERR_n) \right\} \right),$$

где

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n – номер системного цикла;

Δt – период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$.

I-PD

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta PV_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(\Delta PV_n) \right\} \right), \text{ где}$$

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n – номер системного цикла;

Δt – период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔPV_n — приращение переменной процесса, $\Delta PV_n = PV_n - PV_{n-1}$.

PI-D

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta ERR_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(\Delta PV_n) \right\} \right), \text{ где}$$

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n — номер системного цикла;

Δt — период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$;

ΔPV_n — приращение переменной процесса, $\Delta PV_n = PV_n - PV_{n-1}$.

7.12.3.2. 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.

OUT_IL - максимальная скорость приращения выхода, задается в процентах шкалы в секунду. Если задать 1, то за одну секунду выход изменится максимум на один процент. Если значение атрибута меньше 0, то ограничение скорости приращения выхода отключено.

7.12.3.3. Фильтрация шума сигнала

На данном этапе можно включить фильтрацию аналогового входа для защиты от шумов с помощью атрибута ***F_EN***.

Фильтрация шумов в модуле PIDController происходит с помощью сочетания медианного фильтра и бегущего среднего.

F_BUFF (значение от 1-10) количество значений, которые сохраняются в памяти блока и по которым высчитывается медианное среднее.

Затем модуль берет дельту предыдущего и текущего входного значения модуля, умножает на F_COEFF . Полученное значение прибавляется к текущему входному значению.

7.13. 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.

7.13.1. Атрибуты Motor

Motor имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 73. Дополнительные атрибуты Motor

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
ANS2	BOOL	чтение	Сигнал состояния 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	Режим слежения
ANS0	BOOL	чтение	Сигнал состояния отключен
CMD2_ESD	BOOL	чтение	Команда ПАЗ пуск
CMD0_ESD	BOOL	чтение	Команда ПАЗ стоп
PERM2	BOOL	чтение	Готовность к запуску
PERM0	BOOL	чтение	Готовность к останову
Выходные атрибуты			
CMD2	BOOL	чтение	Команда "Пуск"
CMD0	BOOL	чтение	Команда "Стоп"

Имя атрибута	Тип данных	Доступ	Описание атрибута
Конфигурационные атрибуты			
ANS2_SRC	STRING	чтение/ запись	Источник состояния 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 слежения
CTRL_TYPE	DINT	чтение/ запись	Тип управляющего сигнала
PTM	DINT	чтение/ запись	Длительность импульса
MAN_ON_ERR	BOOL	чтение/ запись	Перейти в ручной режим при аварии
ANS0_SRC	STRING	чтение/ запись	Источник состояния отключен
ESD_PRI	DINT	чтение/ запись	Приоритетная команда защиты

Имя атрибута	Тип данных	Доступ	Описание атрибута
CMD2_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ пуск
CMD0_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ стоп
CTRL_TYPE_ESD	DINT	чтение/ запись	Тип управляющего сигнала при защите
PERM2_SRC	STRING	чтение/ запись	Источник готовности к запуску
PERM0_SRC	STRING	чтение/ запись	Источник готовности к останову
AUTO_RST	BOOL	чтение/ запись	Автоматическая деблокировка
OFI	BOOL	чтение/ запись	Выход следит за входом
MOTO_H_RST	BOOL	чтение/ запись	Сброс моточасов
FRC_CMD	BOOL	чтение/ запись	Форсированные команды управления
Диагностические атрибуты			
STATE	DINT	чтение	Состояние блока
MV	USINT	чтение	Значение команды ⁹
PV	USINT	чтение	Значение процесса ¹⁰
EVENTS_STATE	DINT	чтение	Состояние сообщений
MOTO_H	REAL	чтение	Моточасы

⁹ Атрибут **MV** - требуемое состояние, оно же код команды:

- 0 - выключить
- 2 - включить

¹⁰ Атрибут **PV** - переменная процесса:

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

Имя атрибута	Тип данных	Доступ	Описание атрибута
Сигнализационные атрибуты			
FLT	BOOL	чтение	Сигнал "Авария"
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	Пуск по защите
CMD0_TRP	BOOL	чтение	Стоп по защите
IP_ERR	BOOL	чтение	Неопределенное положение
UN2_ERR	BOOL	чтение	Несанкционированный запуск
UN0_ERR	BOOL	чтение	Несанкционированный останов
CMD2_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не запустился"
CMD0_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не остановился"
FLT_Priority	BOOL	чтение	Приоритет сигнализации "Авария"
CMD_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Неверная программная команда"
CMD2_TRP_Priority	BOOL	чтение	Приоритет сигнализации "Пуск по защите"
CMD0_TRP_Priority	BOOL	чтение	Приоритет сигнализации "Стоп по защите"
IP_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Неопределенное положение"
UN2_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Несанкционированный запуск"
UN0_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Несанкционированный останов"

7.13.2. Режимы Motor

Режимы Motor:

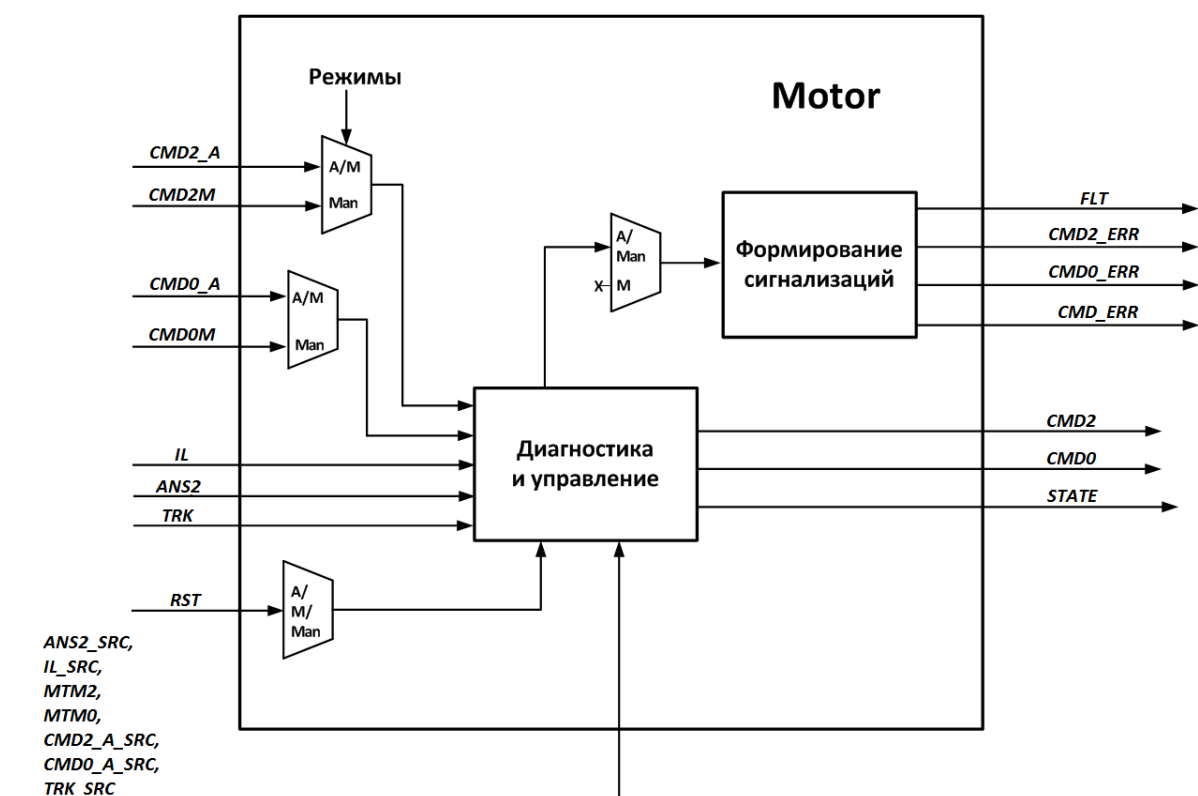
- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

7.13.3. 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 39. Процесс обработки данных Motor

7.13.3.1. 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);
- 6 - ExitTracking (выход из режима отслеживания).

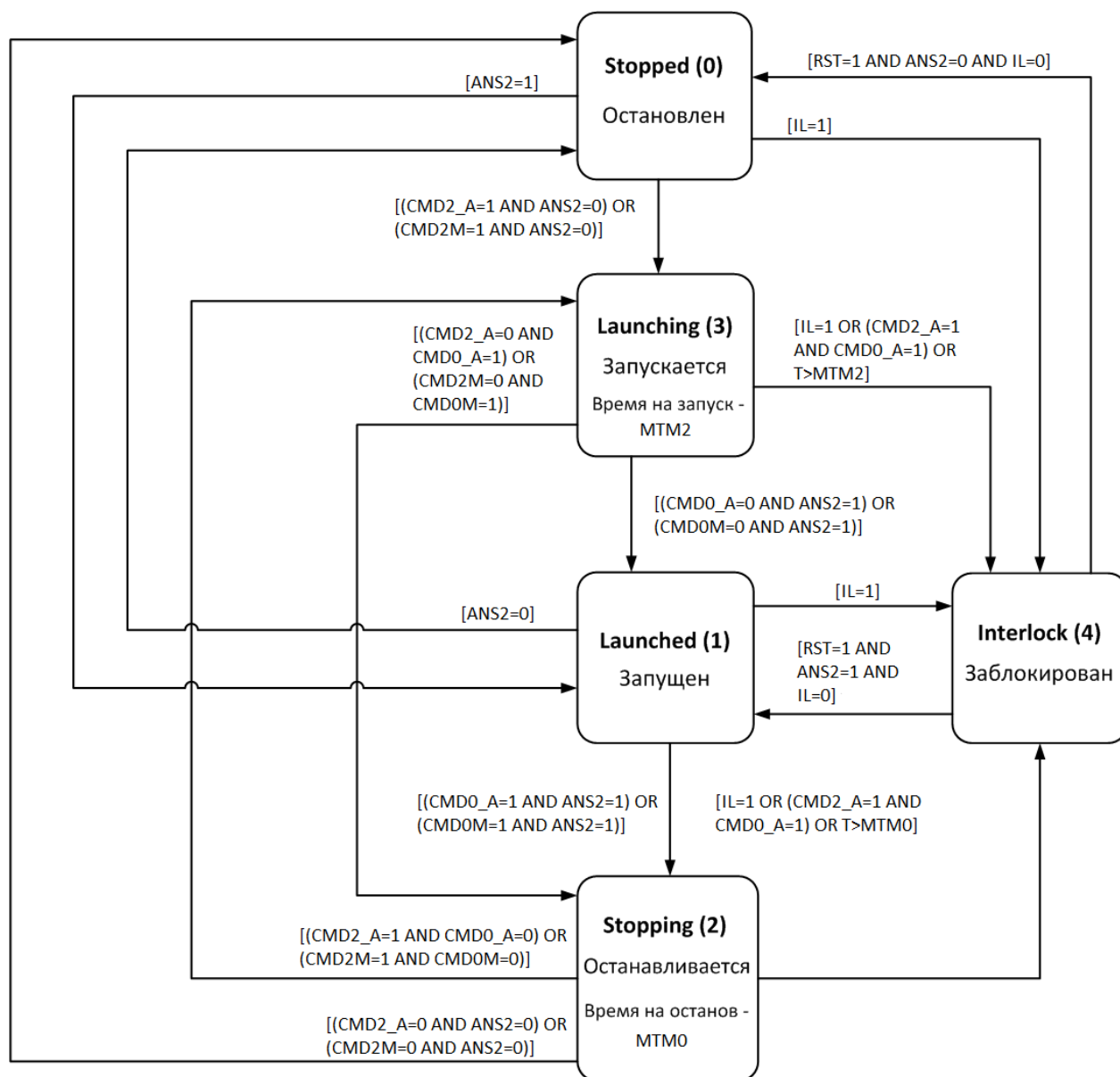


Рисунок 40. Диаграмма перехода состояний Motor

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 41. Диаграмма перехода из/в состояние Tracking.

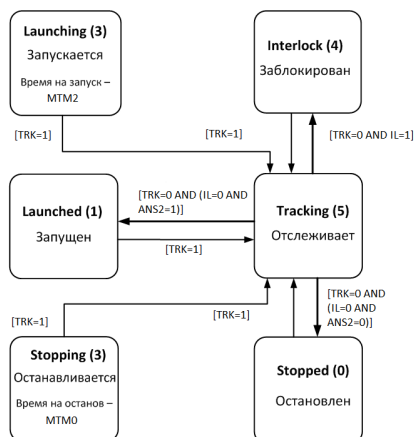


Рисунок 41. Диаграмма перехода из/в состояние Tracking



Внимание: *TRK* имеет приоритет над атрибутом *OFI*. Если к входу *TRK* подвязан функциональный блок, то отслеживание включается только через состояние Tracking, независимо от настройки *OFI*.

Настройка *FRC_CMD* = 1 дает возможность повторной подачи команд из любого состояния блока при отсутствии активных сигналов блокировок *CMD0_ESD*, *CMD2_ESD*.

Входные атрибуты *CMD0_ESD* и *CMD2_ESD* – это входы блокировки на закрытие/открытие. Если пришла блокировка, то функциональный блок переходит в состояние interlock и взводится соответствующая команда управления.

Входные атрибуты *PERM0* и *PERM2* – сигнал готовности к останову/запуску. Если нет готовности, то команды управления не проходят.

Конфигурационный атрибут *ESD_PRI* – приоритетная команда защиты, если одновременно пришла блокировка на открытие и на закрытие.

CTRL_TYPE_ESD – выбор типа управления защитой:

- ResetByState - сбрасывать управляющий выход, если пришло нужное состояние;
- HoldAnyway - удерживать независимо от состояния.

При включенной настройке **MAN_ON_ERR** блок переходит в режим MANUAL при переходе в состояние Interlock. Изменение режима также доступно, если блок находится в заблокированном состоянии.

7.13.3.2. 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 74. 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: $T > MTM2$
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: $CMD2_A = 1 \text{ AND } CMD0_A = 1$

Атрибут	Обозначение	Условие
IP_ERR	Неопределенное положение ¹¹	Нет сигналов состояния в течение времени максимума из полного времени включения и отключения
UN2_ERR	Несанкционированное включение ¹²	Не было ручной или автоматической команды на включение, и пришло состояние "включен"
UN0_ERR	Несанкционированное отключение ¹³	Не было ручной или автоматической команды на отключение, и пришло состояние "отключен"

Таблица 75. Дополнительные условия возникновения ошибки "несанкционированное включение/отключение"

Условие	Значение	Формирование ошибки
TRK есть сигнал	TRK=1	Не формируется
TRK есть сигнал	TRK=0	Формируется
TRK отсутствует сигнал	OFI=1	Не формируется
TRK отсутствует сигнал	OFI=0	Формируется

Таблица 76. Дополнительные условия возникновения ошибки "неопределенное положение"

Условие	Переход в режим Interlock	Формирование ошибки
ФБ находится в состоянии Launched или Stopped и пришло два концевика	Да	Формируется

¹¹ См. таблицу 73. Дополнительные условия возникновения ошибки "неопределенное положение"

¹² См. таблицу 72. Дополнительные условия возникновения ошибки "несанкционированное включение/отключение"

¹³ См. таблицу 72. Дополнительные условия возникновения ошибки "несанкционированное включение/отключение"

Условие	Переход в режим Interlock	Формирование ошибки
ФБ находится в состоянии Launched или Stopped, настройка "выход следит за входом" отключена и произошла пропажа двух концевиков	Да	Формируется
ФБ находится в состоянии Tracking, Stopping, Launching либо в состоянии Launched, Stopped, с включенной настройкой "выход следит за входом" и нет сигналов состояния в течение времени максимума из полного времени включения и отключения	Нет	Формируется

7.14. Функциональный блок Program

Функциональный блок Program (далее Program) запускает программу на языке Calculon¹⁴.

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

7.14.1. Атрибуты Program

Program имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 77. Дополнительные атрибуты Program

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
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)
Выходные атрибуты			
BO1...BO4	BOOL	чтение	Выходы BO1...BO4
RO1...RO2	REAL	чтение	Выходы RO1...RO2
IO1	DINT	чтение	Выход IO1

¹⁴ "Справочное руководство по языку Calculon".

Имя атрибута	Тип данных	Доступ	Описание атрибута
ERR_STR	DINT	чтение	Строка ошибки
Конфигурационные атрибуты			
TXT	STRING	чтение/ запись	Текст программы
BI1_SRC...BI8_SRC	BOOL	чтение/ запись	Источник BI1...Источник BI8
RI1_SRC...RI4_SRC	STRING	чтение/ запись	Источник RI1...Источник RI4
II1_SRC...II2_SRC	STRING	чтение/ запись	Источник II1...Источник II2

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

Таблица 78. Дополнительное описание значений атрибута Status Program

Имя	Значение	Описание
ProgramError	0x02050000	Ошибка в программе Calculon

7.14.2. Режимы Program

Режимы Program:

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

7.14.3. Алгоритм блока Program

Выполнение алгоритма Program зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение;
- интерпретация и запись;
- диагностика.

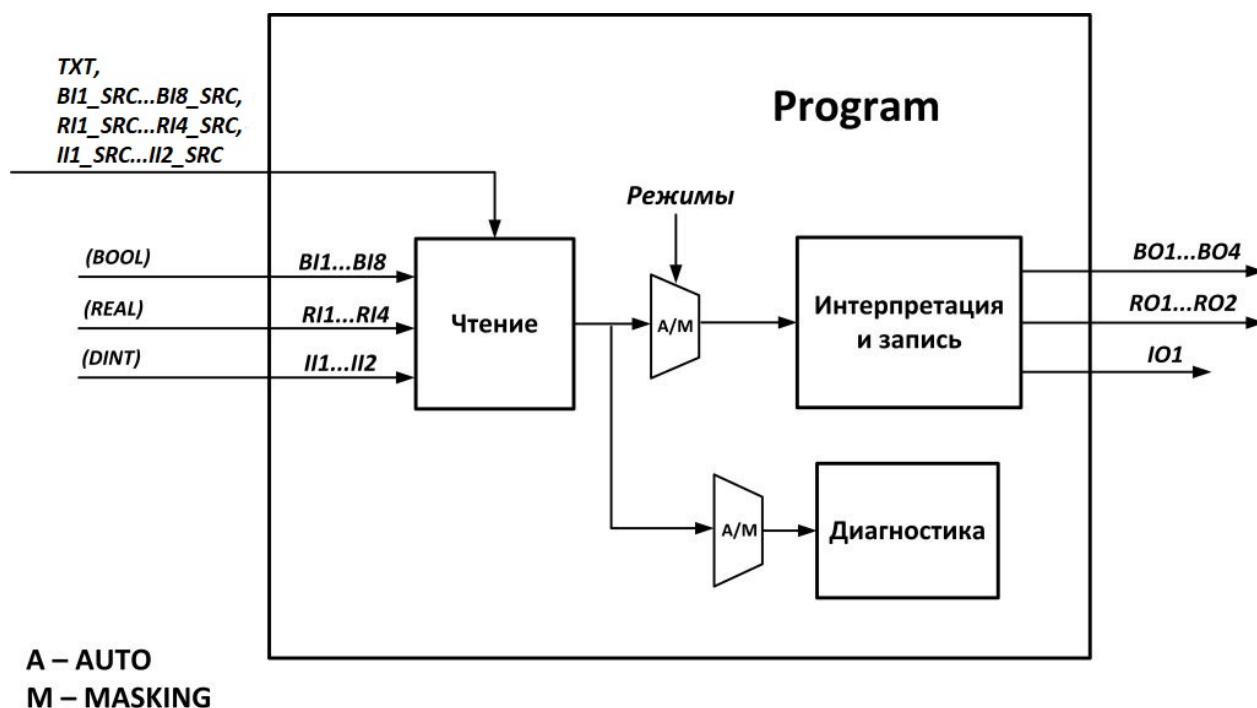


Рисунок 42. Процесс обработки данных Program

7.14.3.1. Чтение

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

Функциональные блоки привязываются к Program по нумерованным атрибутам источника.

Тип входного атрибута и атрибута источника должен соответствовать типу считываемых данных (см. Таблица 77. Дополнительные атрибуты Program, Рисунок 42. Процесс обработки данных Program).

7.14.3.2. Интерпретация и запись

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

Текст программы записывается в конфигурационный атрибут *TXT*. 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
```

7.14.3.3. Диагностика

В процессе диагностики осуществляется проверка:

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

7.15. 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.

7.15.1. Атрибуты RedundantDiscretePoint

RedundantDiscretePoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 79. Дополнительные атрибуты RedundantDiscretePoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные/выходные атрибуты			
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
Конфигурационные атрибуты			
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
Сигнализационные атрибуты			
DISC_ALM	BOOL	чтение	Сигнализация
CH1_ERR	BOOL	чтение	Ошибка канала номер 1
CH2_ERR	BOOL	чтение	Ошибка канала номер 2
CH1_ERR_Priority	BOOL	чтение	Приоритет сигнала "ошибка канала 1"
CH2_ERR_Priority	BOOL	чтение	Приоритет сигнала "ошибка канала 2"

7.15.2. Режимы RedundantDiscretePoint

Режимы RedundantDiscretePoint:

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

7.15.3. Алгоритм RedundantDiscretePoint

Выполнение алгоритма RedundantDiscretePoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

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

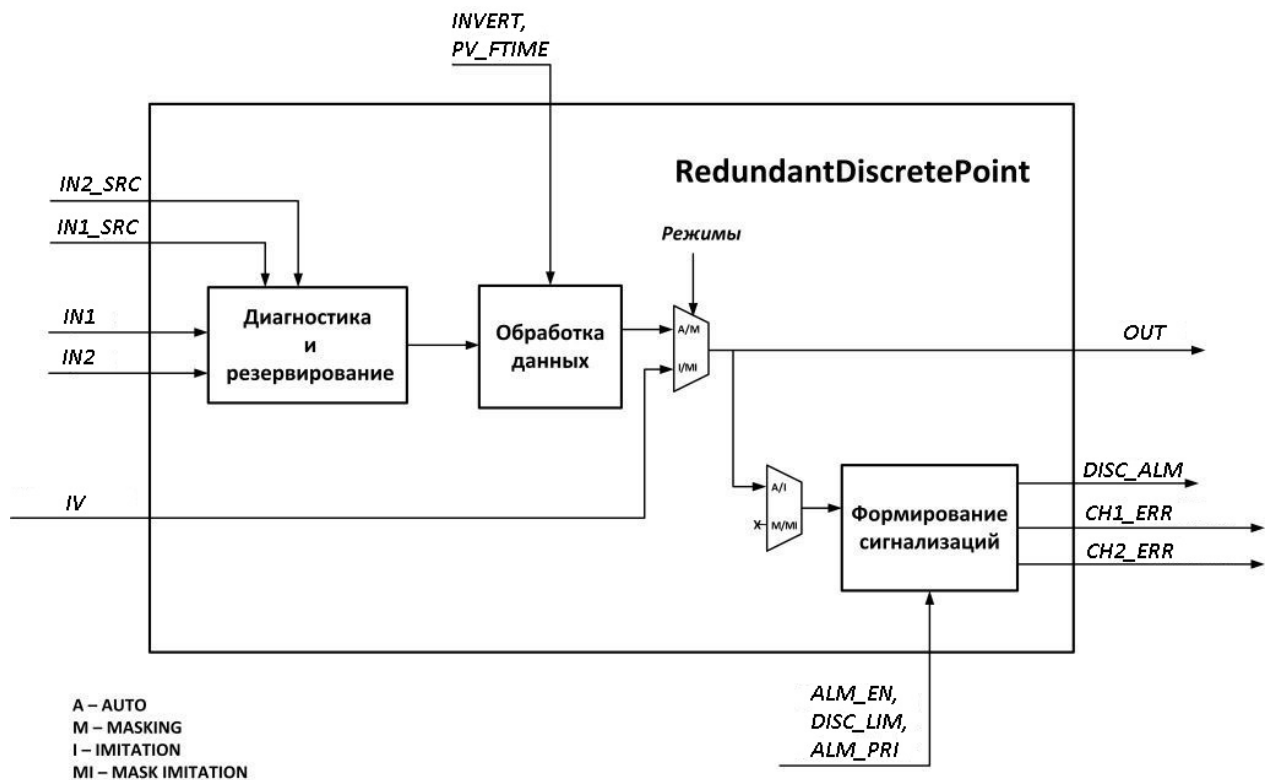


Рисунок 43. Процесс обработки данных RedundantDiscretePoint

7.15.3.1. 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**.

7.15.3.2. 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**.

7.15.3.3. 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.

7.16. 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.

7.16.1. Атрибуты RedundantAnalogInputPoint

RedundantAnalogInputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 80. Дополнительные атрибуты RedundantAnalogInputPoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные/выходные атрибуты			
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
Конфигурационные атрибуты			
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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	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
F_COEFF	REAL	чтение/ запись	Коэффициент фильтрации
F_EN	BOOL	чтение/ запись	Фильтрация входа
F_BUFF	USINT	чтение/ запись	Буфер медианного фильтра
engineerUnit	STRING	чтение/ запись	Единица измерения
Сигнализационные атрибуты			
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
CH1_ERR	BOOL	чтение	Ошибка канала 1
CH2_ERR	BOOL	чтение	Ошибка канала 2
HH_Priority	BOOL	чтение	Приоритет сигнала ВАУ
HI_Priority	BOOL	чтение	Приоритет сигнала ВПУ
LO_Priority	BOOL	чтение	Приоритет сигнала НПУ
LL_Priority	BOOL	чтение	Приоритет сигнала НАУ
CH1_ERR_Priority	BOOL	чтение	Приоритет сигнала "ошибка канала 1"
CH2_ERR_Priority	BOOL	чтение	Приоритет сигнала "ошибка канала 2"

7.16.2. Режимы RedundantAnalogInputPoint

Режимы RedundantAnalogInputPoint:

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

7.16.3. 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.

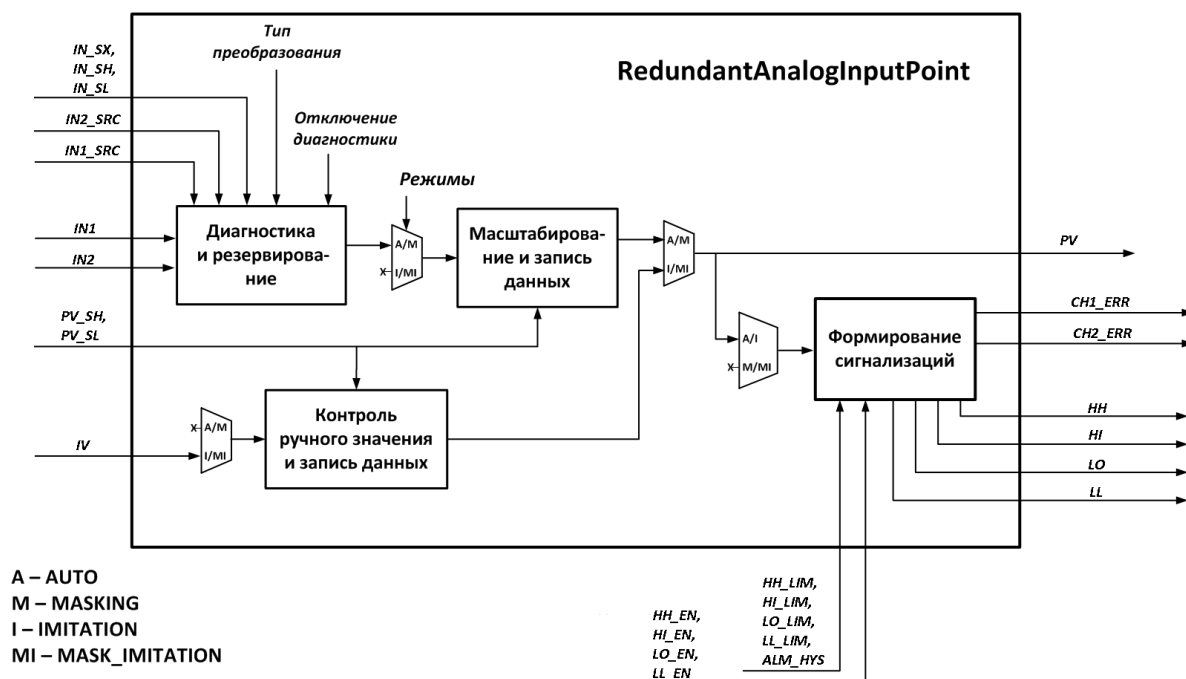


Рисунок 44. Процесс обработки данных RedundantAnalogInputPoint

7.16.3.1. 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.

7.16.3.2. 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:

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

7.16.3.3. 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.

7.16.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 10.1. Формирование сигнализаций Библиотеки.

7.16.3.5. Фильтрация шума сигнала

На данном этапе можно включить фильтрацию аналогового входа для защиты от шумов с помощью атрибута F_EN .

Фильтрация шумов в модуле RedundandAnalogInputPoint происходит с помощью сочетания медианного фильтра и бегущего среднего.

F_BUFF (значение от 1-10) количество значений, которые сохраняются в памяти блока и по которым высчитывается медианное среднее.

Затем модуль берет дельту предыдущего и текущего входного значения модуля, умножает на F_COEFF . Полученное значение прибавляется к текущему входному значению.

7.17. 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.

7.17.1. Атрибуты Valve

Valve имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 81. Дополнительные атрибуты Valve

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
ANS2	BOOL	чтение	Открыт
ANS0	BOOL	чтение	Закрыт
IL	BOOL	чтение	Сигнал блокировки

Имя атрибута	Тип данных	Доступ	Описание атрибута
CMD2_A	BOOL	чтение	Программная команда открытия
CMD1_A	BOOL	чтение	Программная команда останова
CMD0_A	BOOL	чтение	Программная команда закрытия
TRK	BOOL	чтение	Режим слежения
CMD2_ESD	BOOL	чтение	Команда ПАЗ открыть
CMD0_ESD	BOOL	чтение	Команда ПАЗ закрыть
Выходные атрибуты			
CMD2	BOOL	чтение	Открыть
CMD1	BOOL	чтение	Остановить
CMD0	BOOL	чтение	Закрыть
Конфигурационные атрибуты			
ANS2_SRC	STRING	чтение/ запись	Источник состояния открыт
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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 слежения
CTRL_TYPE	DINT	чтение/ запись	Тип управляющего сигнала
PTM	DINT	чтение/ запись	Длительность импульса
MAN_ON_ERR	BOOL	чтение/ запись	Перейти в ручной режим при аварии
CTM	DINT	чтение/ запись	Время ожидания схода с концевика
ESD_PRI	DINT	чтение/ запись	Приоритетная команда защиты
CMD2_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ открыть
CMD0_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ закрыть
CTRL_TYPE_ESD	DINT	чтение/ запись	Тип управляющего сигнала при защите
AUTO_RST	BOOL	чтение/ запись	Автоматическая деблокировка
OFI	BOOL	чтение/ запись	Выход следит за входом
FRC_CMD	BOOL	чтение/ запись	Форсированные команды управления
Диагностические атрибуты			
STATE	DINT	чтение	Состояние блока

Имя атрибута	Тип данных	Доступ	Описание атрибута
MV	USINT	чтение	Значение команды ¹⁵
PV	USINT	чтение	Значение процесса ¹⁶
EVENTS_STATE	DINT	чтение	Состояние сообщений
Сигнализационные атрибуты			
FLT	BOOL	чтение	Сигнал "Авария"
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"
ANS2_ERR	BOOL	чтение	Не сошел с концевика "Открыто"
ANS0_ERR	BOOL	чтение	Не сошел с концевика "Закрыто"
CMD2_TRP	BOOL	чтение	Открыта по защите
CMD0_TRP	BOOL	чтение	Закрыта по защите
CAP_ERR	BOOL	чтение	Несоответствие команды и положения концевика
CMD2_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не открылся"

¹⁵ Атрибут **MV** - требуемое состояние, оно же код команды:

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

¹⁶ Атрибут **PV** - переменная процесса:

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

Имя атрибута	Тип данных	Доступ	Описание атрибута
CMD1_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не остановился"
CMD0_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не закрылся"
FLT_Priority	BOOL	чтение	Приоритет сигнализации "Авария"
CMD_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Неверная программная команда"
JAMMED_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Заклинило"
ANS2_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не сошел с концевика "Открыто"
ANS0_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Не сошел с концевика "Закрыто"
CMD2_TRP_Priority	BOOL	чтение	Приоритет сигнализации "Открыта по защите"
CMD0_TRP_Priority	BOOL	чтение	Приоритет сигнализации "Закрыта по защите"
CAP_ERR_Priority	BOOL	чтение	Приоритет сигнализации "Несоответствие команды и положения концевика"

7.17.2. Режимы Valve

Режимы Valve:

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

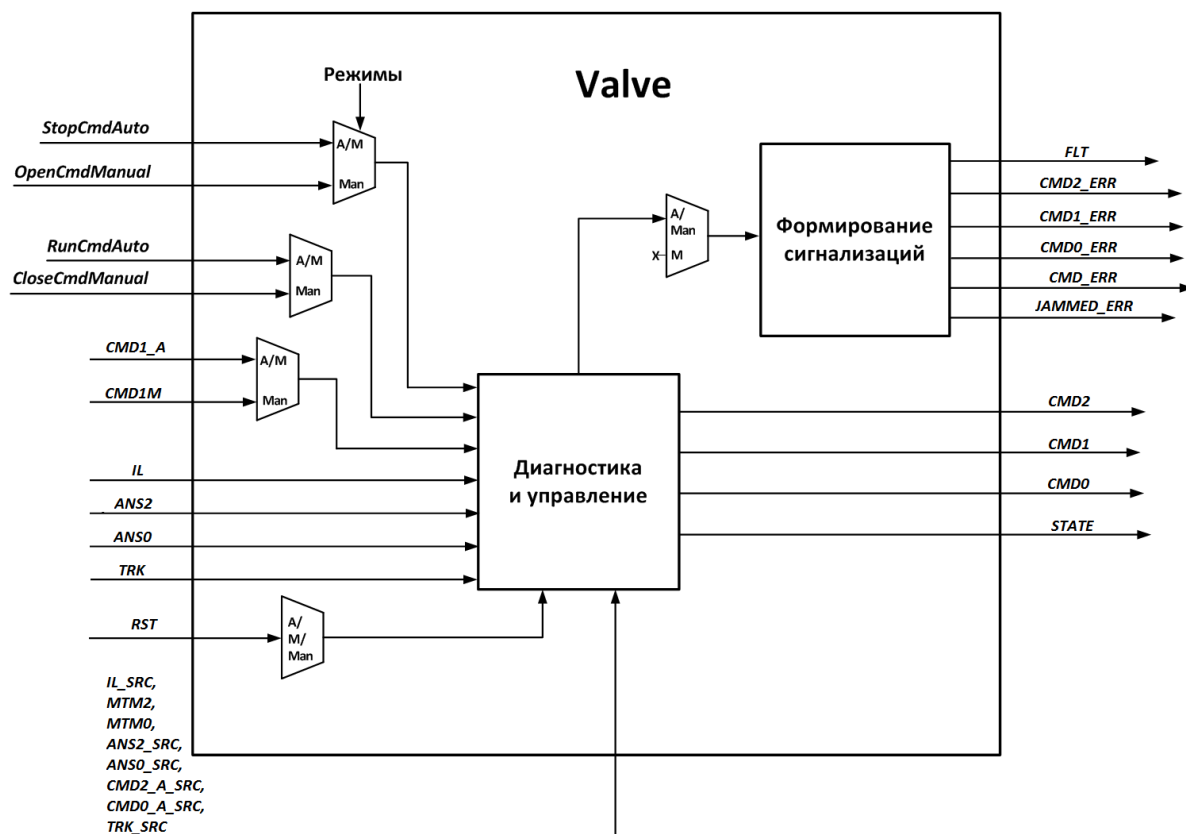
7.17.3. Алгоритм Valve

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 45. Схема процесса обработки данных Valve

7.17.3.1. 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);

9 - ExitTracking (выход из режима отслеживания).

Одно состояние переходит в другое при выполнении заданных условий.

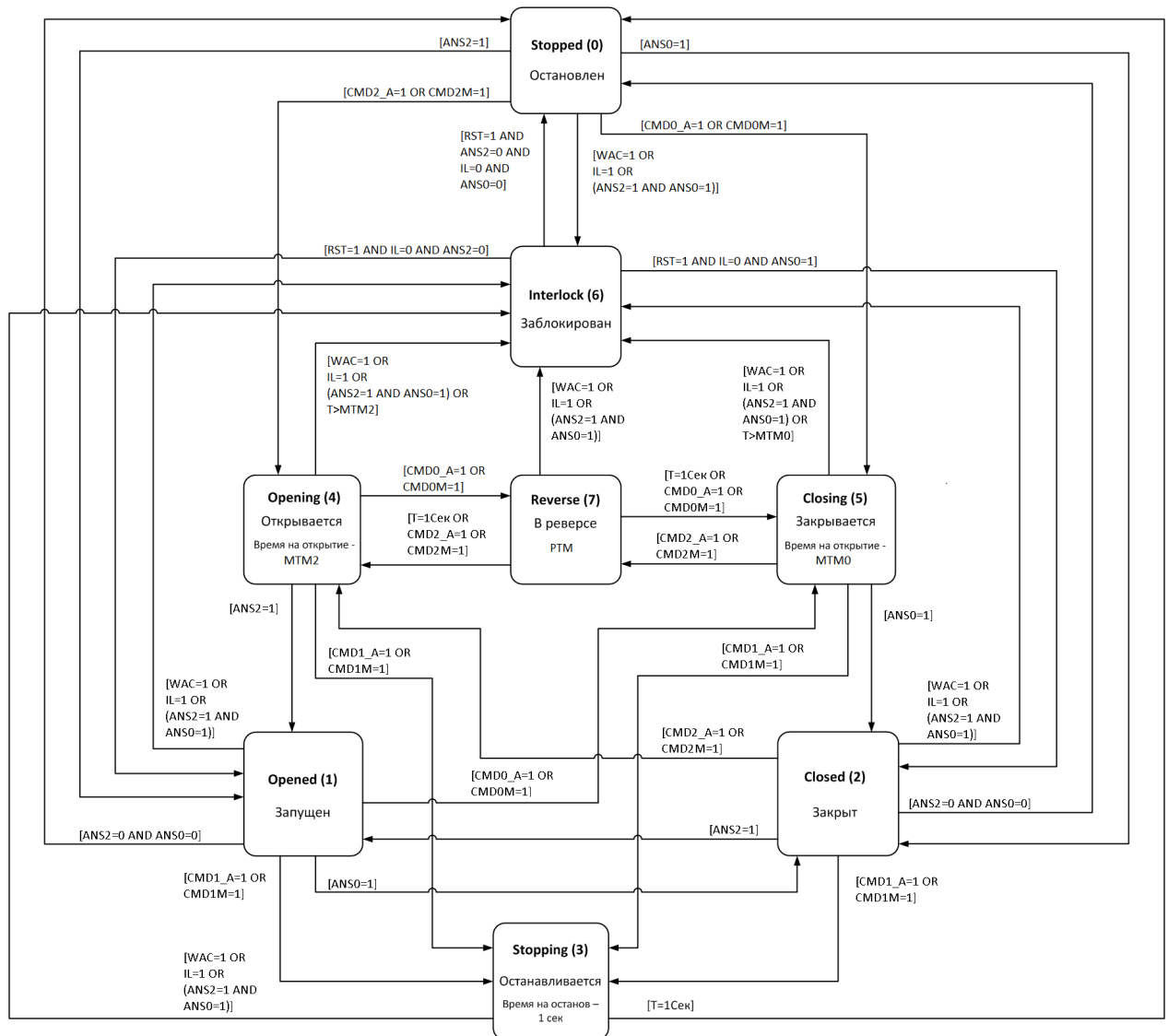


Рисунок 46. Диаграмма перехода состояний блока Valve

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 47. Диаграмма перехода из/в состояние Tracking.

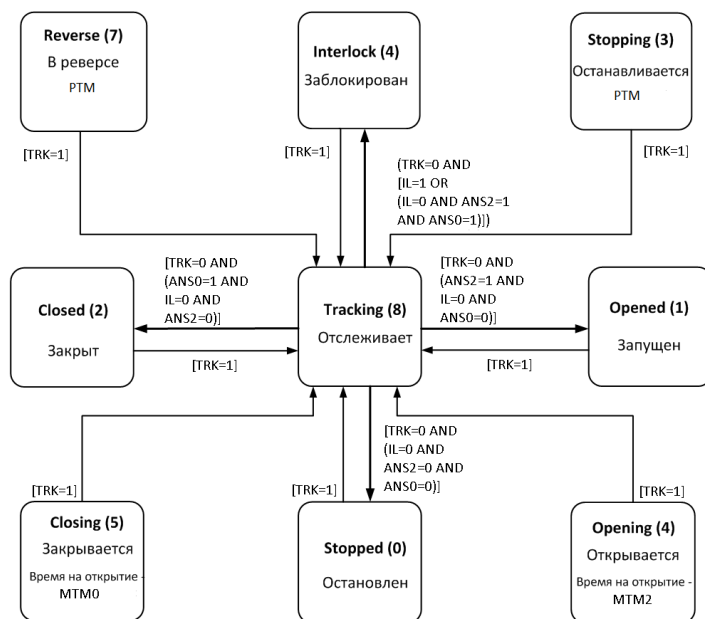


Рисунок 47. Диаграмма перехода из/в состояние Tracking



Внимание: При включенной настройке **OFI** в режиме hold - управляющее воздействие не будет сбрасываться после прихода концевика, так же выходы блока будут "следить" за состояниями концевиков (при изменении состояния будет срабатывать соответствующий выход). Если исполнительное устройство находится в промежуточном положении, выход **CMD1** срабатывать не будет.

Настройка **FRC_CMD = 1** дает возможность повторной подачи команд из любого состояния блока при отсутствии активных сигналов блокировок **CMD0_ESD**, **CMD2_ESD**.

Входные атрибуты **CMD0_ESD** и **CMD2_ESD** – это входы блокировки на закрытие/открытие. Если пришла блокировка, то функциональный блок

переходит в состояние interlock и взводится соответствующая команда управления.

Конфигурационный атрибут **ESD_PRI** – приоритетная команда защиты, если одновременно пришла блокировка на открытие и на закрытие.

CTRL_TYPE_ESD – выбор типа управления защитой:

- ResetByState - сбрасывать управляющий выход, если пришло нужное состояние;
- HoldAnyway - удерживать независимо от состояния.

При включенной настройке **MAN_ON_ERR** блок переходит в режим MANUAL при переходе в состояние Interlock. Изменение режима также доступно, если блок находится в заблокированном состоянии.

7.17.3.2. Формирование сигнализаций

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

Таблица 82. 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: $T > MTM2$
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

Атрибут	Обозначение	Условие
JAMMED_ERR	"Заклинило"	Входные атрибуты показывают одновременно открытое и закрытое состояние: $ANS2 = 1$ AND $ANS0 = 1$
CAP_ERR	"Несоответствие команды и положения концевика"	Несоответствие команды и положения концевика

В режиме MASKING сигнализации не формируются.

7.18. Функциональный блок ValveUnstoppable

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

ValveUnstoppable обеспечивает выполнение следующих функций:

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

Особенности работы:

- ValveUnstoppable имеет настройку "Управление без команды "остановить" (*NO_CMD1*), при включении которой управление электродвигателем может осуществляться без необходимости отправки команды "стоп" (выход "стоп" не срабатывает);
- при выключенной настройке "Выход следит за входом" (*OFI*) при изменении состояния задвижки (без изменения команды управления) не сбрасывает все команды, а продолжает держать последнюю команду.

7.18.1. Атрибуты ValveUnstoppable

ValveUnstoppable имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 83. Дополнительные атрибуты ValveUnstoppable

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
ANS2	BOOL	чтение	Открыт
ANS0	BOOL	чтение	Закрыт

Имя атрибута	Тип данных	Доступ	Описание атрибута
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	Режим слежения
CMD2_ESD	BOOL	чтение	Команда ПАЗ открыть
CMD0_ESD	BOOL	чтение	Команда ПАЗ закрыть
Выходные атрибуты			
CMD2	BOOL	чтение	Открыть
CMD1	BOOL	чтение	Остановить
CMD0	BOOL	чтение	Закрыть
Конфигурационные атрибуты			
ANS2_SRC	STRING	чтение/ запись	Источник состояния открыт
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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 слежения
CTRL_TYPE	DINT	чтение/ запись	Тип управляющего сигнала
PTM	DINT	чтение/ запись	Длительность импульса, с
MAN_ON_ERR	BOOL	чтение/ запись	Перейти в ручной режим при аварии
CTM	DINT	чтение/ запись	Время ожидания схода с концевика, с
ESD_PRI	DINT	чтение/ запись	Приоритетная команда защиты
CMD2_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ открыть
CMD0_ESD_SRC	STRING	чтение/ запись	Источник команды ПАЗ закрыть
CTRL_TYPE_ESD	DINT	чтение/ запись	Тип управляющего сигнала при защите
NO_CMD1	BOOL	чтение/ запись	Управление без команды "остановить"
OFI	BOOL	чтение/ запись	Выход следит за входом
AUTO_RST	BOOL	чтение/ запись	Автоматическая деблокировка
FRC_CMD	BOOL	чтение/ запись	Форсированные команды управления
Диагностические атрибуты			

Имя атрибута	Тип данных	Доступ	Описание атрибута
STATE	DINT	чтение	Состояние блока
MV	USINT	чтение	Значение команды ¹⁷
PV	USINT	чтение	Значение процесса ¹⁸
EVENTS_STATE	DINT	чтение	Состояние сообщений
Сигнализационные атрибуты			
FLT	BOOL	чтение	Сигнал "Авария"
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"
ANS2_ERR	BOOL	чтение	Не сошел с концевика "Открыто"
ANS0_ERR	BOOL	чтение	Не сошел с концевика "Закрыто"
CMD2_TRP	BOOL	чтение	Открыта по защите
CMD0_TRP	BOOL	чтение	Закрыта по защите
CAP_ERR	BOOL	чтение	Несоответствие команды и положения концевика

7.18.2. Режимы ValveUnstoppable

Режимы ValveUnstoppable:

¹⁷ Атрибут **MV** - требуемое состояние, оно же код команды:

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

¹⁸ Атрибут **PV** - переменная процесса:

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

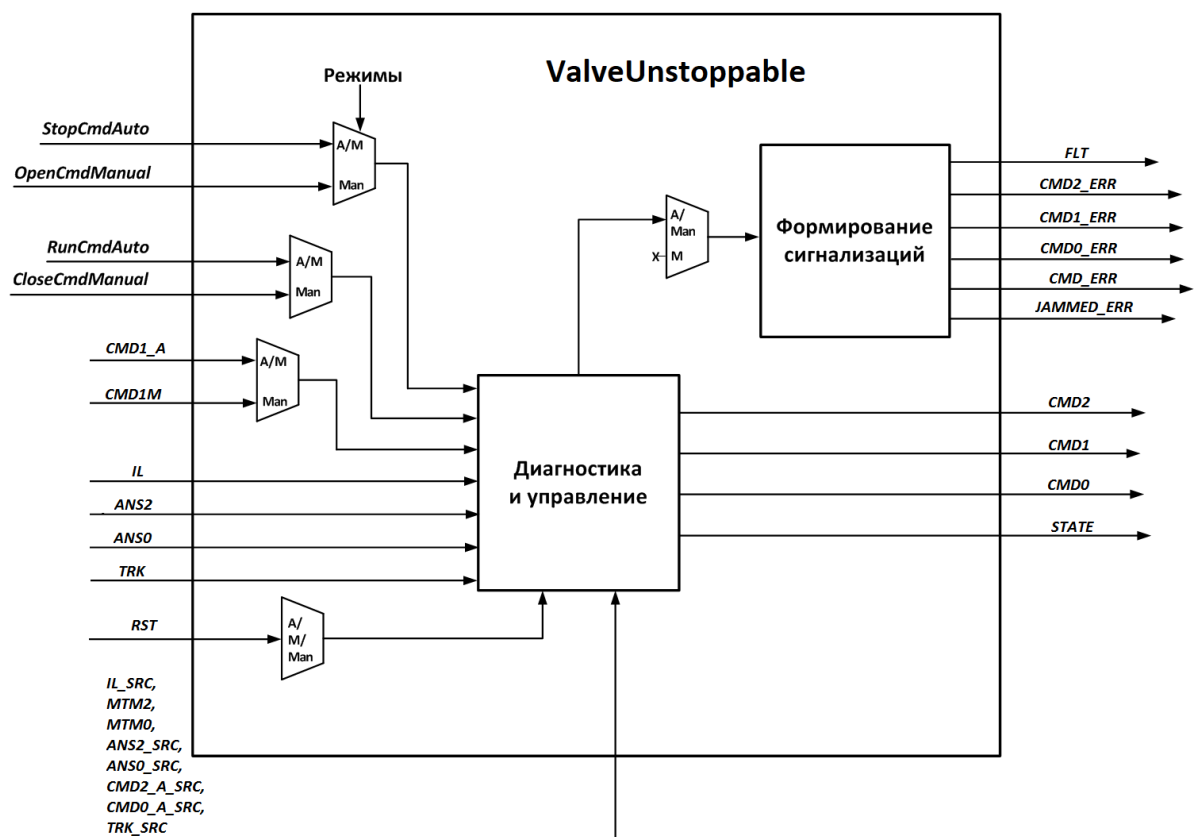
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- MANUAL ("Ручной").

7.18.3. Алгоритм ValveUnstoppable

Алгоритм блока ValveUnstoppable зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и управление;
- формирование сигнализаций.



A – AUTO
Man – MANUAL
M – MASKING

Рисунок 48. Схема процесса обработки данных ValveUnstoppable

7.18.3.1. Диагностика и управление ValveUnstoppable

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

Значения состояний:

0 - Stopped (остановлен);

- Одно состояние переходит в другое при выполнении заданных условий.

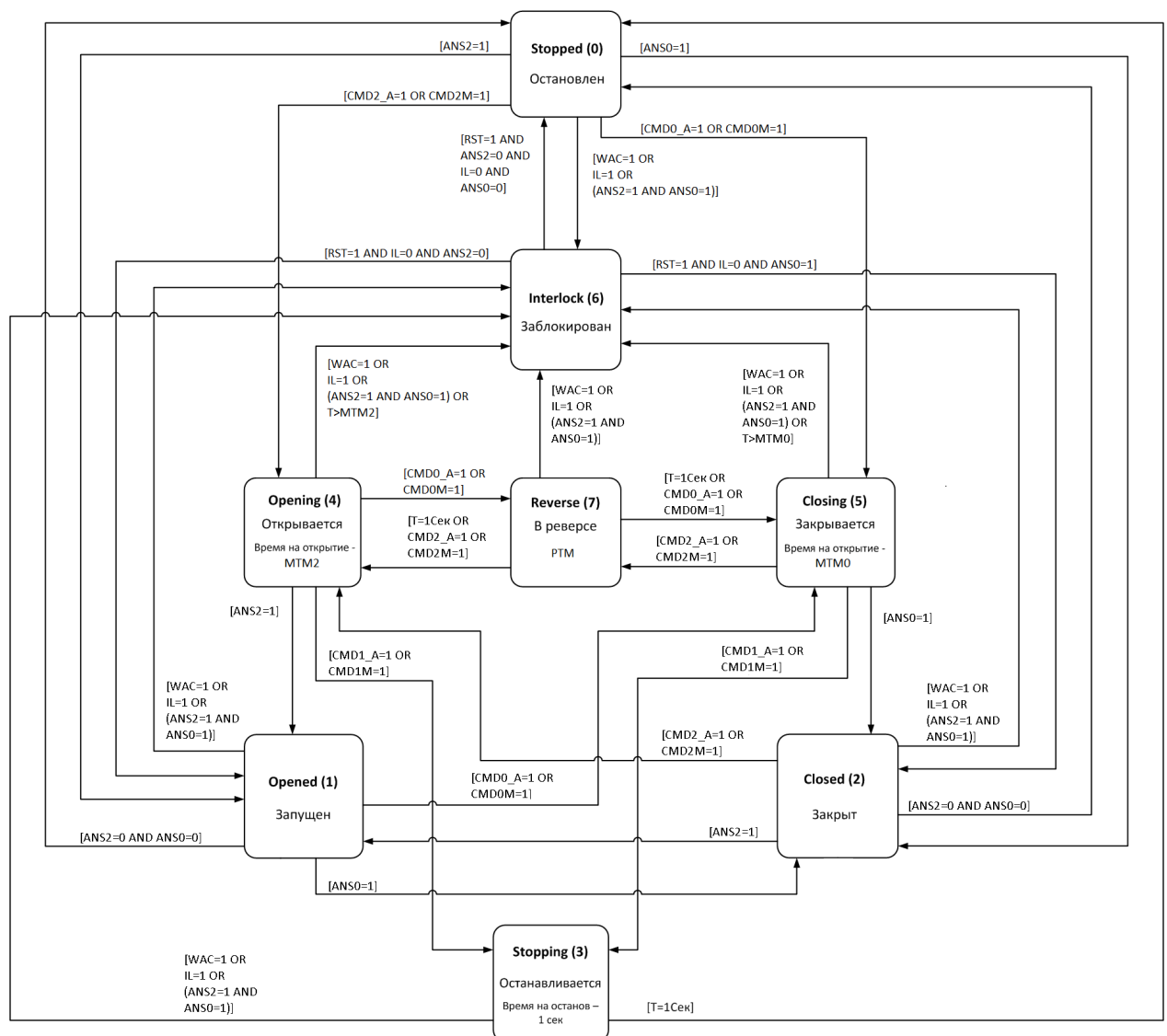


Рисунок 49. Диаграмма перехода состояний блока ValveUnstoppable

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 - "Закрыть".

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 [Диаграмма перехода из/в состояние Tracking](#).

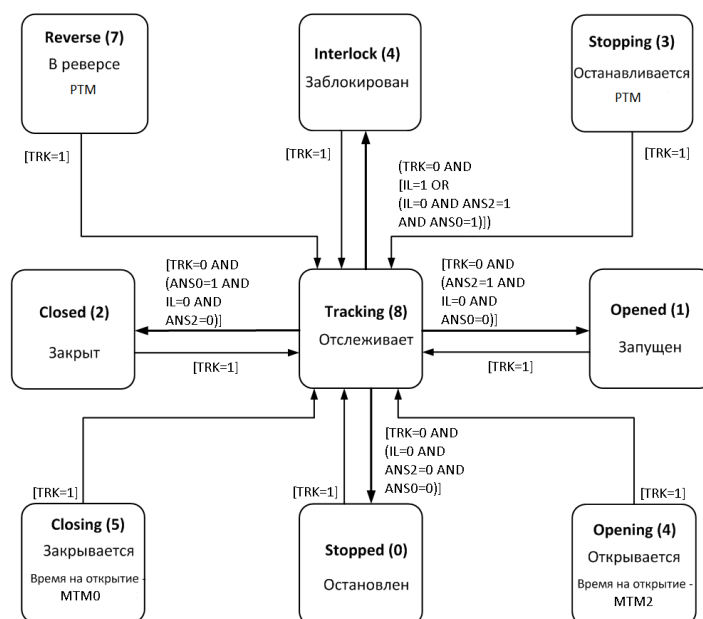


Рисунок 50. Диаграмма перехода из/в состояние Tracking

Настройка **FRC_CMD** = 1 дает возможность повторной подачи команд из любого состояния блока при отсутствии активных сигналов блокировок **CMD0_ESD**, **CMD2_ESD**.

Входные атрибуты **CMD0_ESD** и **CMD2_ESD** – это входы блокировки на закрытие/открытие. Если пришла блокировка, то функциональный блок

переходит в состояние interlock и взводится соответствующая команда управления.

Конфигурационный атрибут **ESD_PRI** – приоритетная команда защиты, если одновременно пришла блокировка на открытие и на закрытие.

CTRL_TYPE_ESD – выбор типа управления защитой:

- ResetByState - сбрасывать управляющий выход, если пришло нужное состояние;
- HoldAnyway - удерживать независимо от состояния.

При включенной настройке **MAN_ON_ERR** блок переходит в режим MANUAL при переходе в состояние Interlock. Изменение режима также доступно, если блок находится в заблокированном состоянии.

7.18.3.2. Формирование сигнализаций

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

Таблица 84. 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: $T > MTM2$
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

Атрибут	Обозначение	Условие
JAMMED_ERR	"Заклинило"	Входные атрибуты показывают одновременно открытое и закрытое состояние: $ANS2 = 1$ AND $ANS0 = 1$
CAP_ERR	"Несоответствие команды и положения концевика"	Несоответствие команды и положения концевика

В режиме MASKING сигнализации не формируются.

7.19. Функциональный блок LogicalBlock

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

7.19.1. Атрибуты LogicalBlock

LogicalBlock имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 85. Дополнительные атрибуты LogicalBlock

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
FUNC	DINT	чтение/запись	Функция

7.19.2. Режимы LogicalBlock

Режимы LogicalBlock:

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

- MASKING ("Маскирование").

7.19.3. Алгоритм LogicalBlock

Алгоритм LogicalBlock включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

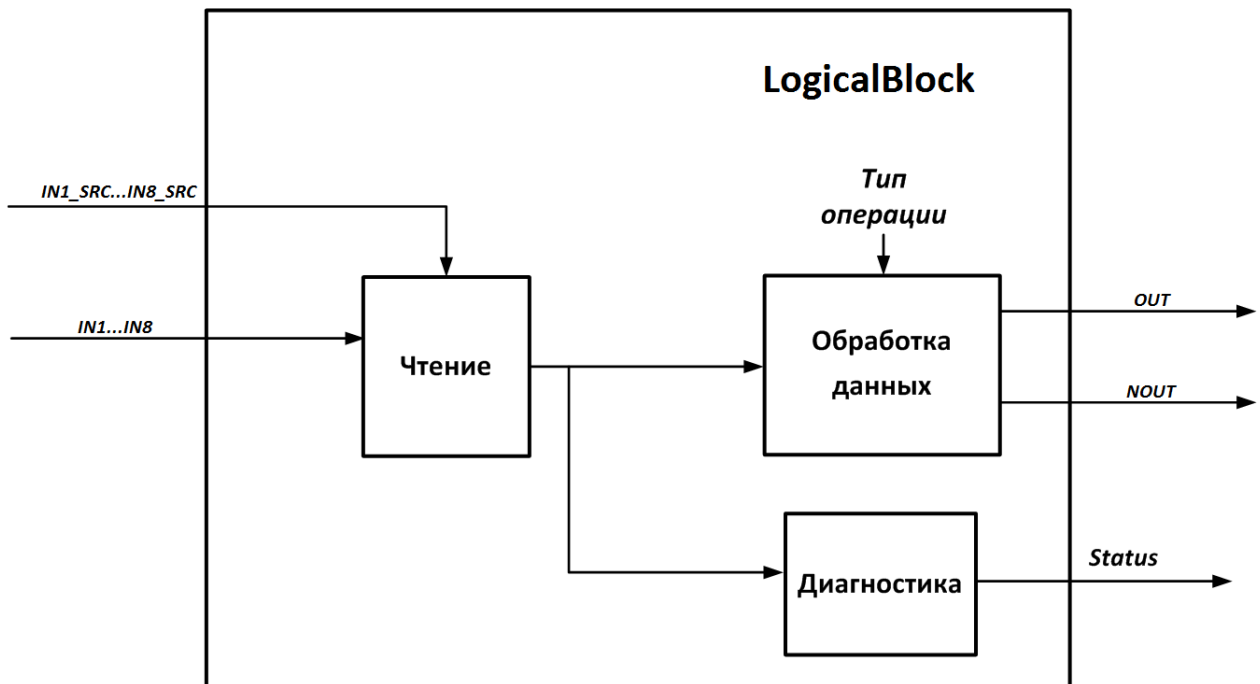


Рисунок 51. Процесс обработки данных LogicalBlock

7.19.3.1. Чтение

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

7.19.3.2. 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 86. 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

7.19.3.3. Диагностика

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

При этом учитываются следующие правила¹⁹:

- 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.
- Если один из источников *IN1_SRC...IN8_SRC* имеет статус отличный от **ОК**, атрибут *Status* блока принимает значение статуса этого источника, и обработка данных в текущем рабочем цикле прекращается.

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

¹⁹ Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

7.20. Функциональный блок MinMaxBlock

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

7.20.1. Атрибуты MinMaxBlock

MinMaxBlock имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 87. Дополнительные атрибуты MinMaxBlock

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/Выходные атрибуты</i>			
IN1...IN8	REAL	чтение	Вход
OUT	REAL	чтение	Выход
DIS1...DIS8	BOOL	чтение	Отключение входа
SELECTED	DINT	чтение	Выбранный вход
<i>Конфигурационные атрибуты</i>			
IN1_SRC...IN8_SRC	STRING	чтение/ запись	Источник
FUNC	USINT	чтение/ запись	Тип функции - значение аналогового входа: • Минимум (min); • Максимум (max); • Средний (mid); • Первый Нормальный (firstgood); • Усредненный (avg); • Горячий Резерв (hotbackup).

Имя атрибута	Тип данных	Доступ	Описание атрибута
OP_SEL	USINT	чтение/ запись	Если имеет ненулевое значение, выбирает соответствующий вход (т.е. 2 выбирает IN2), если значение нулевое, выходное значение OUT определяется параметром FUNC
DIS1_SRC... DIS8_SRC	STRING	чтение/ запись	Источник отключения входа DIS
DISABLE_NV1... DISABLE_NV8	BOOL	чтение/ запись	Отключает входы IN при отсутствии внешнего входа DIS
MIN_GOOD	USINT	чтение/ запись	Минимальное количество пригодных для использования входов, при котором результат не имеет статуса <i>Bad</i>

7.20.2. Режимы MinMaxBlock

Режимы MinMaxBlock:

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

7.20.3. Алгоритм MinMaxBlock

Алгоритм MinMaxBlock включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

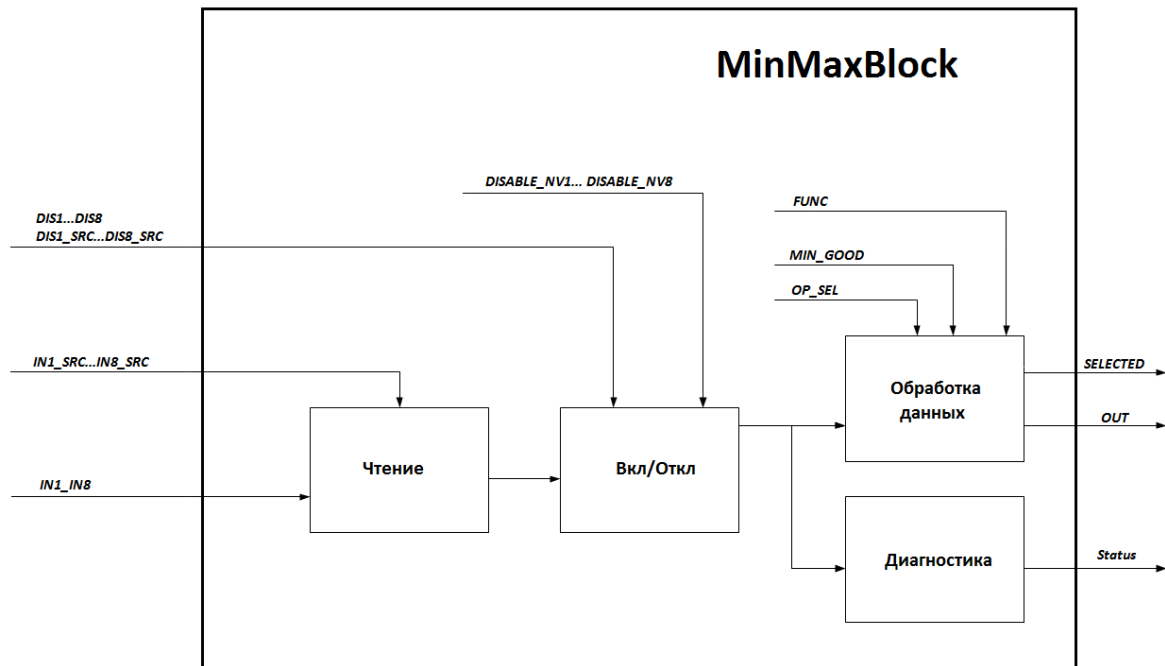


Рисунок 52. Процесс обработки данных MinMaxBlock

7.20.3.1. Чтение

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

7.20.3.2. Обработка данных

OP_SEL – если имеет ненулевое значение, выбирает соответствующий вход (т.е. 2 выбирает **IN2**). Если значение равно 0, то вход выбирает **FUNC**.

Тип функции - значение аналогового входа:

- Минимум (min) - выбирает вход с минимальным значением из тех входов, которые подключены к источникам, не выключены²⁰ и не имеют статус источника *Bad*.

²⁰ Выключенным считается вход, который:

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

- 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 *IN1*
- 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.

DIS1...8 имея истинное значение, выключает соответствующий вход (т.е. ***DIS1*** выключает ***IN1***, ***DIS2*** выключает ***IN2***). Выключенный вход не используется алгоритмом, сконфигурированным в ***FUNC***.

Если соответствующий вход ***DIS1...8*** не подключен к ***DIS1_SRC...DIS8_SRC***, то его состояние берется из параметра ***DISABLE_NV1...8*** соответственно, если подключен, то ***DISABLE_NV1...8*** игнорируется и состояние берется из ***DIS1_SRC...DIS8_SRC***.

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

7.20.3.3. Диагностика

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

Во время диагностики определяется состояние блока MinMaxBlock.

При этом учитываются следующие правила²¹:

- Если источник ***IN1_SRC...IN8_SRC*** отсутствует, он исключается из обработки данных.
- Если все источники ***IN1_SRC...IN8_SRC*** отсутствуют, атрибут ***Status*** функционального блока MinMaxBlock принимает значение ResolvingReference - обработка данных не выполняется.
- 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.

²¹ Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

7.21. Функциональный блок AccumulatorBlock

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

7.21.1. Атрибуты AccumulatorBlock

Таблица 88. Дополнительные атрибуты AccumulatorBlock

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/Выходные атрибуты</i>			
IN	REAL	чтение	Вход
RST	BOOL	чтение	Сброс по импульсу
OUT	REAL	чтение	Выход
OUT_INT	DINT	чтение	Целочисленная часть выходного значения
OUT_FRAC	REAL	чтение	Дробная часть выходного значения
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Источник
RST_SRC	STRING	чтение/ запись	Источник сброса
TIME_UNIT	DINT	чтение/ запись	Единица измерения времени, с\мин\ч\день
INTEG_OPTS	DINT	чтение/ запись	Вид суммирования: Module - по модулю, OneSide - однонаправленное, Bidirectional - двунаправленное
engineerUnit	STRING	чтение/ запись	Единица измерения
<i>Сигнализационные атрибуты</i>			
OVFL	BOOL	чтение	Переполнение

Имя атрибута	Тип данных	Доступ	Описание атрибута
OVFL_Priority	BOOL	чтение	Приоритет сигнализации "Переполнение"

7.21.2. Режимы AccumulatorBlock

Режимы AccumulatorBlock:

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

7.21.3. Алгоритм AccumulatorBlock

Алгоритм AccumulatorBlock включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

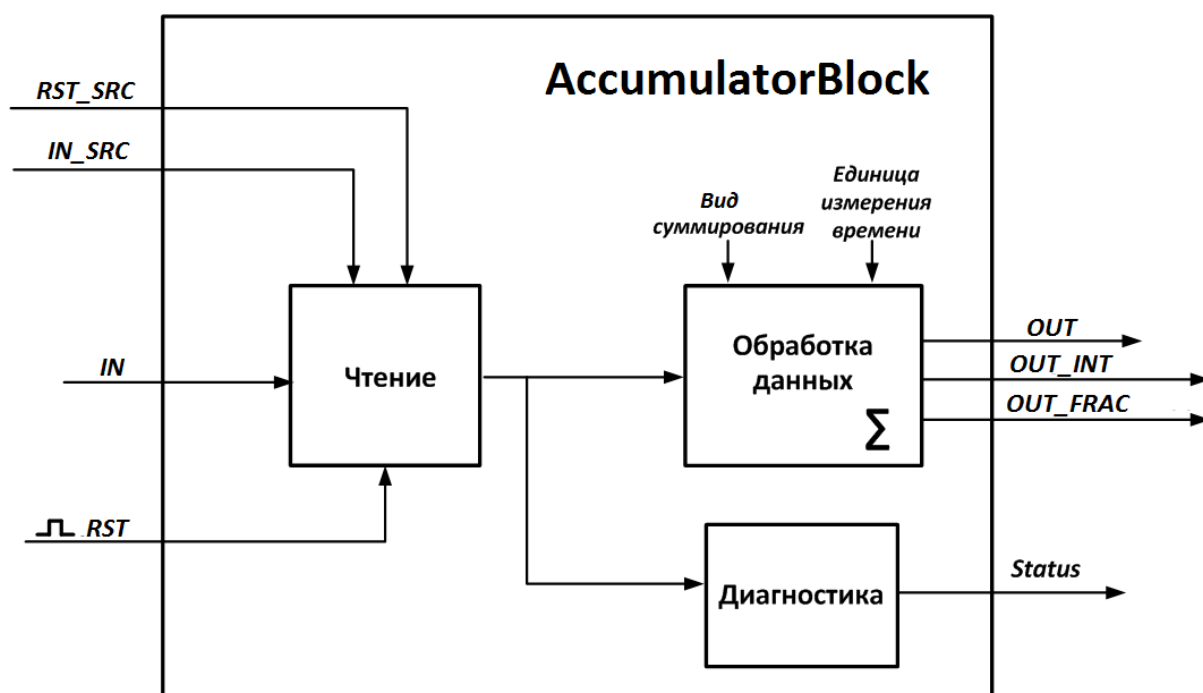


Рисунок 53. Процесс обработки данных AccumulatorBlock

7.21.3.1. Чтение

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

7.21.3.2. Обработка данных

Обработка данных заключается в суммировании по времени входного значения *IN* и запись итоговой суммы в выходные атрибуты *OUT*, *OUT_INT*, *OUT_FRAC*.

Различают типы суммирования (*INTEG_OPTS*):

- 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 *RST*, в результате которого отсчет суммы начинается заново.

7.21.3.3. Диагностика

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

Во время диагностики определяется состояние блока AccumulatorBlock.

При этом если источник *IN_SRC* отсутствует, атрибут *Status* функционального блока принимает значение ResolvingReference - обработка данных не выполняется.²²

7.22. 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;
- 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.

²² Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

7.22.1. Атрибуты RedundantPIDController

RedundantPIDController имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 89. 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 сorsa, то == 1
HI_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сorsa, то == 1
LO_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сorsa, то == 1
LL_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сorsa, то == 1
OUT	REAL	чтение	Выход

Имя атрибута	Тип данных	Доступ	Описание атрибута
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/ запись	Накопленный интегральный компонент
PV	REAL	чтение	Регулируемое значение (выход)
CASO	BOOL	чтение	Готовность к управлению выход
RMV	REAL	чтение/ запись	Задание внешнего выхода
RSPI	REAL	чтение	Вход внешнего задания

Имя атрибута	Тип данных	Доступ	Описание атрибута
RSP	REAL	чтение/ запись	Внешнее задание
RMVI	REAL	чтение	Вход внешнего выхода
Конфигурационные атрибуты			
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 HH

Имя атрибута	Тип данных	Доступ	Описание атрибута
HI_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HI
LO_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LO
LL_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LL
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 ⁻¹
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

Имя атрибута	Тип данных	Доступ	Описание атрибута
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
PV_DB	REAL	чтение/ запись	Зона нечувствительности регулятора
PV_DB_HYS	REAL	чтение/ запись	Гистерезис зоны нечувствительности
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/ запись	Переключаться в ручной режим при статусе BAD

Имя атрибута	Тип данных	Доступ	Описание атрибута
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
REM_SP_PRC	BOOL	чтение/ запись	Каскадная уставка в процентах
ALGORITHM	USINT	чтение/ запись	Алгоритм расчета ПИД управления
OUT_IL	REAL	чтение/ запись	Максимальная скорость приращения выхода, %/сек
OUT_MSL	REAL	чтение/ запись	Нижний предел шкалы управляющего выхода
OUT_MSH	REAL	чтение/ запись	Верхий предел шкалы управляющего выхода
F_BUFF	USINT	чтение/ запись	Буфер медианного фильтра
F_COEFF	REAL	чтение/ запись	Коэффициент фильтрации
F_EN	BOOL	чтение/ запись	Фильтрация входа
RSPI_SRC	STRING	чтение/ запись	Источник внешнего задания
RMVI_SRC	STRING	чтение/ запись	Источник внешнего выхода
engineerUnit	STRING	чтение/ запись	Единица измерения

Имя атрибута	Тип данных	Доступ	Описание атрибута
Сигнализационные атрибуты			
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
HH_Priority	BOOL	чтение	Приоритет сигнализации ВАУ
HI_Priority	BOOL	чтение	Приоритет сигнализации ВПУ
LO_Priority	BOOL	чтение	Приоритет сигнализации НПУ
LL_Priority	BOOL	чтение	Приоритет сигнализации НАУ
CH1_ERR_Priority	BOOL	чтение	Приоритет сигнализации "ошибка канала 1"
CH2_ERR_Priority	BOOL	чтение	Приоритет сигнализации "ошибка канала 2"
Диагностические атрибуты			
RDBK	REAL	чтение	Обратная связь
TRK	BOOL	чтение	Режим отслеживания

7.22.2. Режимы RedundantPIDController

Режимы RedundantPIDController:

1. OFF ("Выключен") - алгоритм не выполняется.
2. AUTO ("Автоматический"). В режиме AUTO входное значение записывается в атрибуты *IN1* и *IN2*. RedundantPIDController обрабатывает входное значение в соответствии с заданными значениями конфигурационных атрибутов и уставкой *SP*. Выходное управляющее значение записывается в атрибут *OUT*.

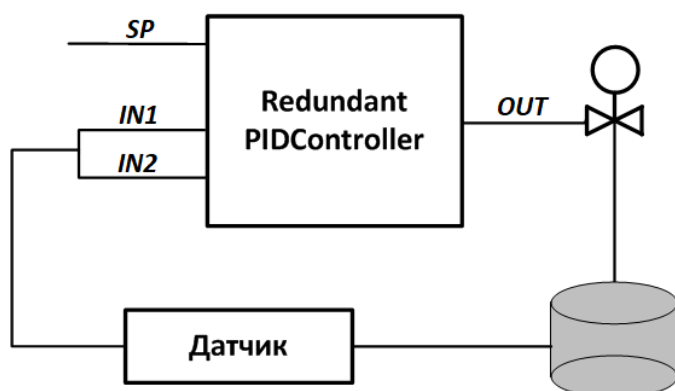


Рисунок 54. RedundantPIDController в режиме AUTO с блокировкой схемы резервирования

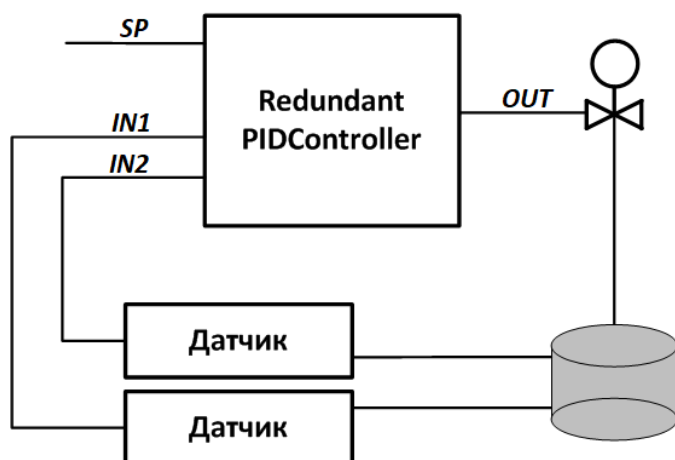


Рисунок 55. RedundantPIDController в режиме AUTO с использованием схемы резервирования

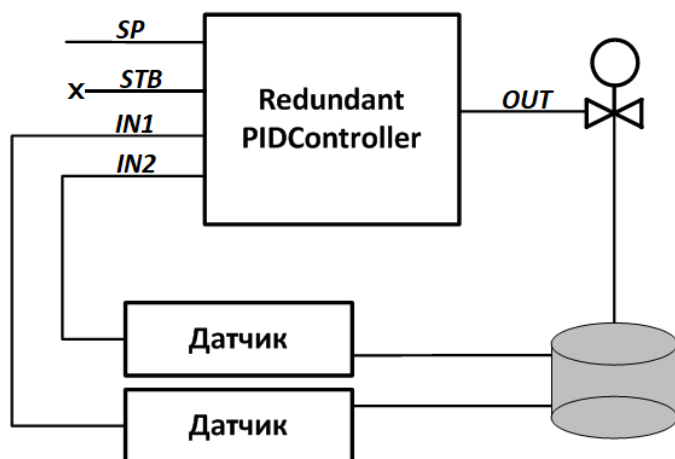


Рисунок 56. RedundantPIDController в режиме AUTO с использованием входа стробирования

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 **MOUТ** в атрибут **OUT**, при этом входы **IN1** и **IN2** 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.

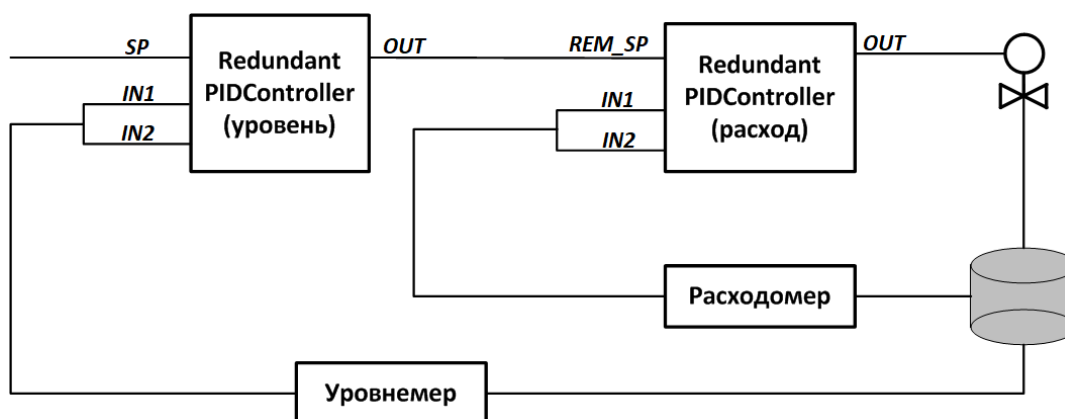


Рисунок 57. RedundantPIDController в режиме CASCADE с блокировкой схемы резервирования

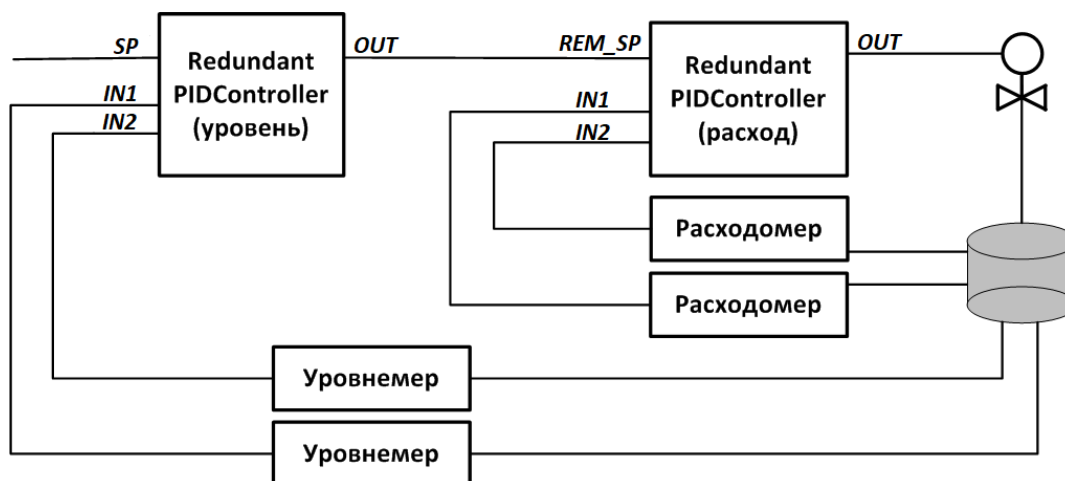


Рисунок 58. RedundantPIDController в режиме CASCADE с использованием схемы резервирования

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.

7.22.3. 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.

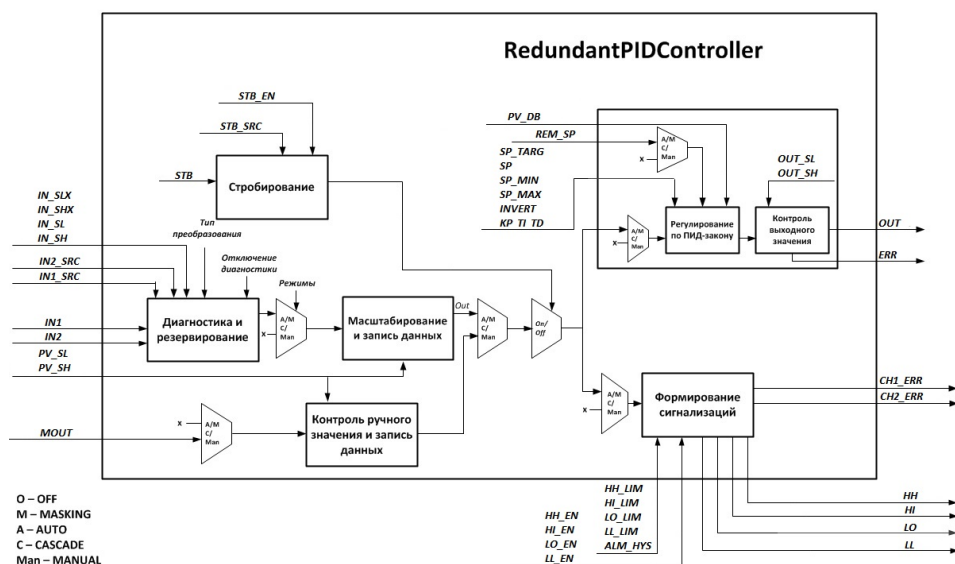


Рисунок 59. Процесс обработки данных RedundantPIDController

7.22.3.1. 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** 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 позволяет 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.

7.22.3.2. 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:

$$OUT = \frac{PV_SH - PV_SL}{IN_SH - IN_SL} \times (IN - IN_SL) + PV_SL$$

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:

$$PV = \frac{PV_SH - PV_SL}{IN_SH - IN_SL} \times (\sqrt{IN} - IN_SL) + PV_SL$$

7.22.3.3. Контроль ручного значения

Данный этап выполняется в режимах MASKING и MANUAL.

Значение атрибута *MOUТ* сравнивается со шкалой датчика в инженерных единицах.

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

7.22.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 10.1. Формирование сигнализаций Библиотеки.

7.22.3.5. 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.

Регулятор поддерживает четыре алгоритма ПИД управления. Алгоритм расчета задается в настройках ФБ через конфигурационный атрибут ***ALGORITHM***. Расчет управляющего воздействия производится в соответствии с выбранным алгоритмом по формулам:

PID (при добавлении блока выбирается по умолчанию)

$$OUT = OUT_s + Inv \times \left(KP \times ERR_n + intsum_n + \frac{TD \times \Delta ERR_n}{\Delta t} \right), \text{ где}$$

OUT_s — значение выхода в момент перехода из MAN в AUTO или CAS;

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

KP — пропорциональный коэффициент;

ERR_n — отклонение, $ERR_n = SP_n - PV_n$;

SP_n — переменная задания;

PV_n — переменная процесса;

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

n — номер системного цикла;

Δt — период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$.

PID_M

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta ERR_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(\Delta ERR_n) \right\} \right),$$

где

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n – номер системного цикла;

Δt – период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$.

I-PD

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta PV_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(\Delta PV_n) \right\} \right), \text{ где}$$

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n – номер системного цикла;

Δt – период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔPV_n — приращение переменной процесса, $\Delta PV_n = PV_n - PV_{n-1}$.

PI-D

$$OUT = OUT_{n-1} + Inv \times \left(\frac{100}{KP} \times K_s \left\{ \Delta ERR_n + \frac{\Delta t}{TI} \times ERR_n + \frac{TD}{\Delta t} \Delta(PV_n) \right\} \right), \text{ где}$$

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

KP — пропорциональный коэффициент;

K_s - коэффициент преобразования шкалы, $K_s = \frac{OUT_{SH}-OUT_{SL}}{PV_{SH}-PV_{SL}}$

ERR_n — отклонение, $ERR_n = PV_n - SP_n$;

PV_n — переменная процесса;

SP_n — переменная задания;

n — номер системного цикла;

Δt — период выборки сигнала, с;

TI — интегральный коэффициент, с;

TD — дифференциальный коэффициент, с;

ΔERR_n — приращение отклонения, $\Delta ERR_n = ERR_n - ERR_{n-1}$;

ΔPV_n — приращение переменной процесса, $\Delta PV_n = PV_n - PV_{n-1}$.

7.22.3.6. 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.

OUT_IL - максимальная скорость приращения выхода, задается в процентах шкалы в секунду. Если задать 1, то за одну секунду выход изменится максимум на один процент. Если значение атрибута меньше 0, то ограничение скорости приращения выхода отключено.

7.22.3.7. Фильтрация шума сигнала

На данном этапе можно включить фильтрацию аналогового входа для защиты от шумов с помощью атрибута ***F_EN***.

Фильтрация шумов в модуле ReduantPIDController происходит с помощью сочетания медианного фильтра и бегущего среднего.

F_BUFF (значение от 1-10) количество значений, которые сохраняются в памяти блока и по которым высчитывается медианное среднее.

Затем модуль берет дельту предыдущего и текущего входного значения модуля, умножает на *F_COEFF*. Полученное значение прибавляется к текущему входному значению.

7.23. Функциональный блок DiscreteInputsPacker

Функциональный блок DiscreteInputsPacker (далее DiscreteInputsPacker) предназначен для упаковки данных из атрибутов входного значения. На входе производится упаковка данных типа BOOL, а на выходе производится запись в тип INT.

7.23.1. Атрибуты DiscreteInputsPacker

DiscreteInputsPacker имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 90. Дополнительные атрибуты DiscreteInputsPacker

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN1... IN16	BOOL	чтение	Входное значение от каналов (Рисунок 60. Битовый формат выходного значения)
OUT	DINT	чтение	Выходное значение
<i>Конфигурационные атрибуты</i>			
IN1_SRC...IN16_SRC	STRING	чтение/ запись	Источник
MV	DINT	чтение/ запись	Ручное значение

Флаги *Bad* формируются из статусов источников входного сигнала.

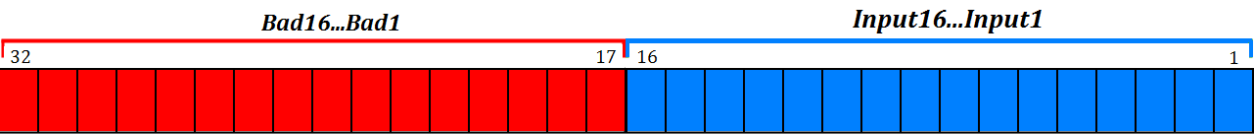


Рисунок 60. Битовый формат выходного значения

7.23.2. Режимы DiscreteInputsPacker

Режимы DiscreteInputsPacker:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK_IMITATION ("Имитация и маскирование").

7.23.3. Алгоритм DiscreteInputsPacker

Алгоритм DiscreteInputsPacker включает в себя следующие элементы:

- чтение;
- обработка данных.

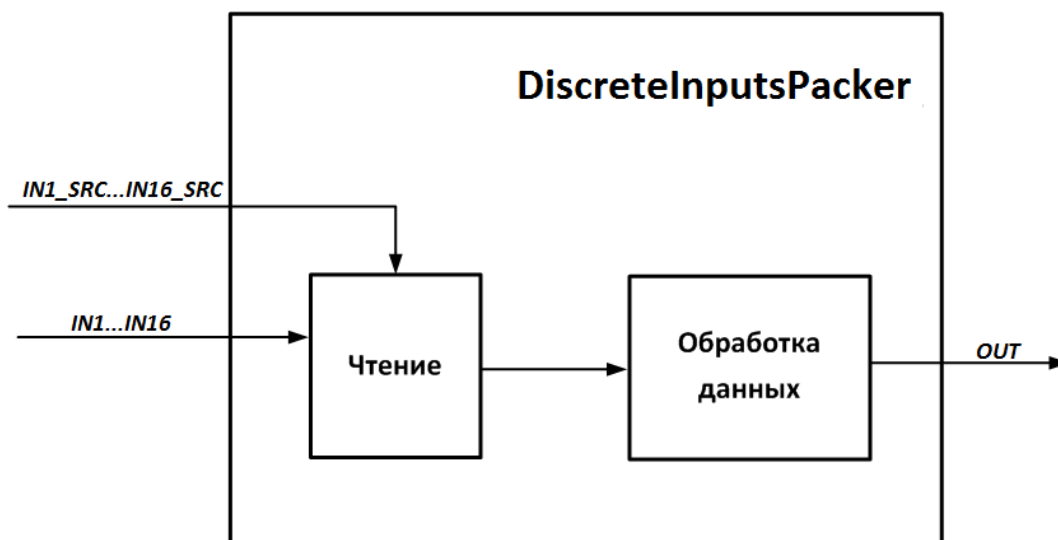


Рисунок 61. Процесс обработки данных DiscreteInputsPacker

7.23.3.1. Чтение

При чтении данные с выхода функционального блока-источника атрибуты *IN1_SRC...IN16_SRC* записываются во входной атрибут *IN*.

7.23.3.2. Обработка данных

Обработка данных заключается в упаковке данных из атрибутов входного значения *IN1...IN16* в тип INT и запись в выходной атрибут *OUT*.

7.24. Функциональный блок DigitalInputUnpacker

Функциональный блок DigitalInputUnpacker (далее DigitalInputUnpacker) выполняет обратную по отношению к DiscreteInputsPacker операцию - распаковку данных. На входе производится распаковка данных типа INT, а на выходе производится запись в тип BOOL.

7.24.1. Атрибуты DigitalInputUnpacker

DigitalInputUnpacker имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 91. Дополнительные атрибуты DigitalInputUnpacker

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	DINT	чтение	Входное значение (Рисунок 62. Битовый формат входного значения)
Выходные атрибуты			
OUT1... OUT16	BOOL	чтение	Выход1... Выход16
BAD1... BAD16	BOOL	чтение	Качество сигнала
Конфигурационные атрибуты			
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 преединенного блока упаковки (DiscreteInputsPacker).

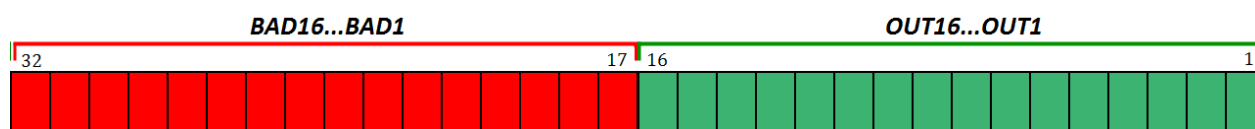


Рисунок 62. Битовый формат входного значения

7.24.2. Режимы DigitalInputUnpacker

Режимы DigitalInputUnpacker:

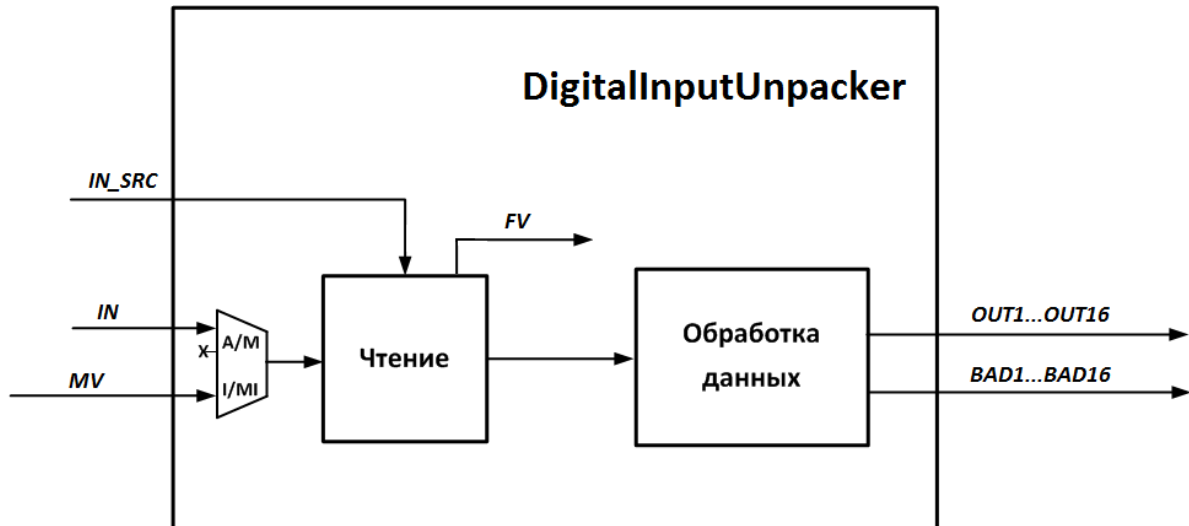
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");

- MASK_IMITATION ("Имитация и маскирование").

7.24.3. Алгоритм DigitalInputUnpacker

Алгоритм DigitalInputUnpacker включает в себя следующие элементы:

- чтение;
- обработка данных.



A – AUTO
M – MASKING
I – IMITATION
MI – MASK_IMITATION

Рисунок 63. Процесс обработки данных DigitalInputUnpacker

7.24.3.1. 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*.

7.24.3.2. 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**.

7.25. Функциональный блок TimeDelay

Функциональный блок TimeDelay (далее TimeDelay) обеспечивает исполнение таймера с задержкой включения/ выключения.

7.25.1. Атрибуты TimeDelay

TimeDelay имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 92. Дополнительные атрибуты TimeDelay

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	BOOL	чтение	Вход
RST	BOOL	чтение	Сброс
Выходные атрибуты			
OUT	BOOL	чтение	Выход
NOUT	BOOL	чтение	Инверсный выход
RT	DINT	чтение	Время до старта, с
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник входа
RST_SRC	STRING	чтение/ запись	Источник сброса
EDGE	DINT	чтение/ запись	Тип таймера: Rising - с задержкой включения, Falling - с задержкой выключения
TON	DINT	чтение/ запись	Временная задержка, с

7.25.2. Режимы TimeDelay

Режимы TimeDelay:

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

7.25.3. Алгоритм TimeDelay

Алгоритм включает в себя следующие элементы:

- чтение;
- обработка данных.

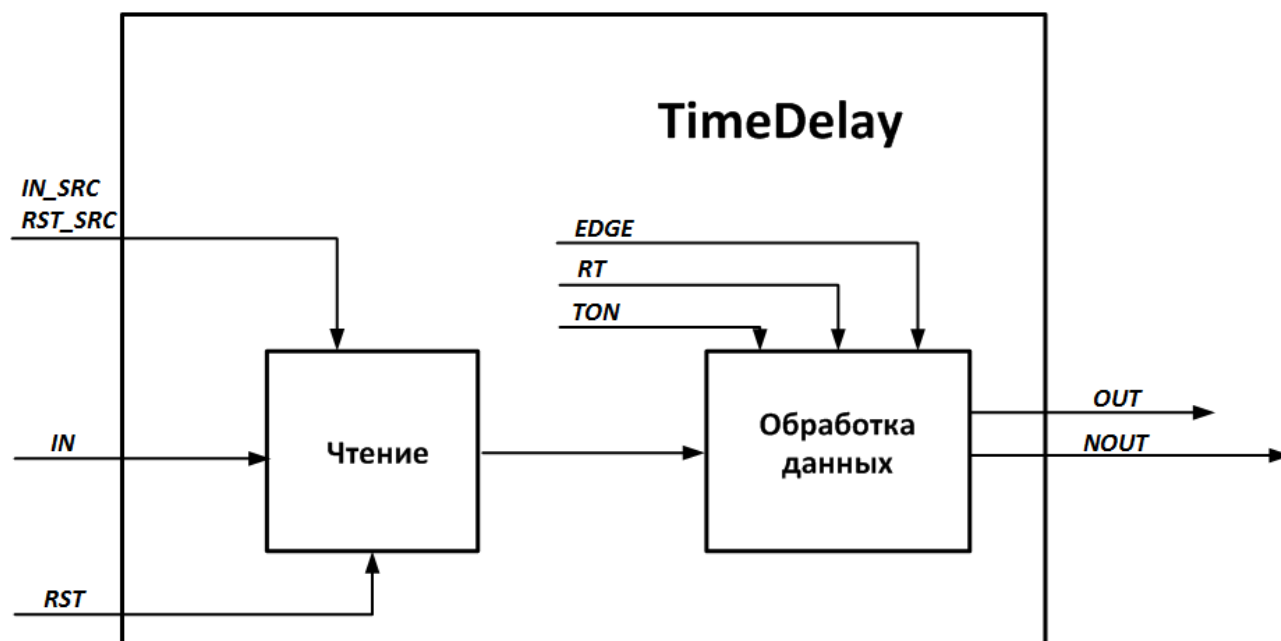


Рисунок 64. Процесс обработки данных TimeDelay

7.25.3.1. Чтение

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

7.25.3.2. Обработка данных

Различают типы таймера (атрибут **EDGE**):

- 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.

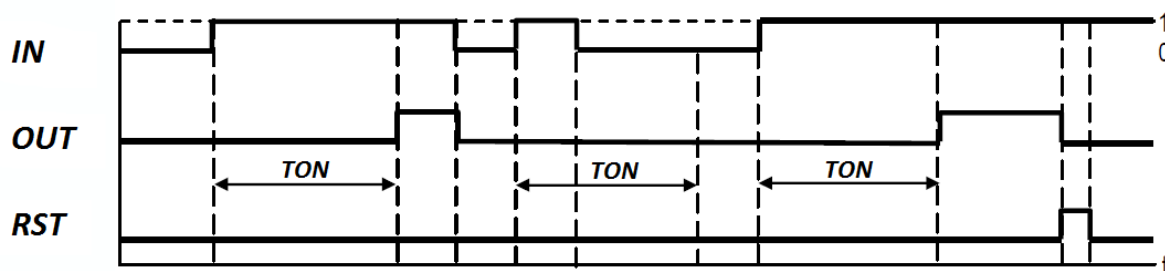


Рисунок 65. Таймер с задержкой включения

На выходе **OUT** блока появится логическая «1» с задержкой относительно фронта входного сигнала **IN**. Выход включается логической «1» на входе продолжительностью не менее длительности **TON**, а выключается по спаду входного сигнала либо когда команда сброса **RST** изменяется на "1".

Таймер с задержкой отключения (TOF)

Блок используется для задержки отключения выхода. Работу поясняет приведенная на рисунке диаграмма.

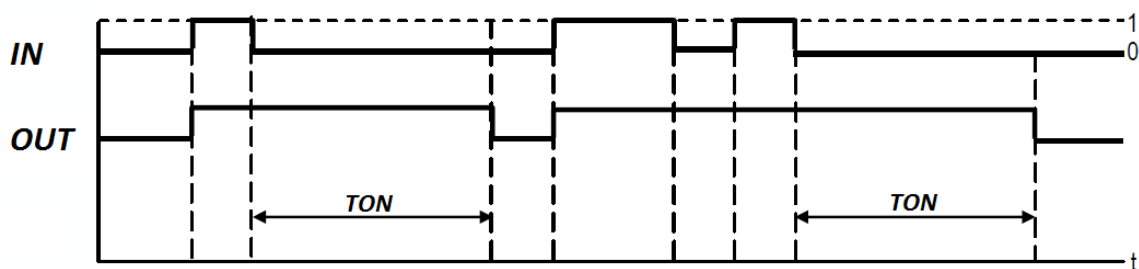


Рисунок 66. Таймер с задержкой отключения

На выходе блока **OUT** появится логическая «1» по фронту сигнала на входе **IN**, а начало отсчета времени задержки отключения (**TON**) происходит по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится логический «0» с задержкой **TON**. В данном блоке вход **RST** не используется.

7.26. Функциональный блок PulseDuration

Функциональный блок PulseDuration (далее PulseDuration) используется для генерации импульсов заданной/переменной длительности.

7.26.1. Атрибуты PulseDuration

PulseDuration имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 93. Дополнительные атрибуты PulseDuration

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	BOOL	чтение	Входное значение
RST	BOOL	чтение	Сброс
RT	DINT	чтение	Оставшееся время, с
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выходное значение
NOUT	BOOL	чтение	Выходное значение, инвертированное
<i>Конфигурационные атрибуты</i>			
RST_SRC	STRING	чтение/ запись	Источник данных для Reset
IN_SRC	STRING	чтение/ запись	Источник данных для Input
P_TYPE	DINT	чтение/ запись	Тип импульса: Fixed - с заданной длительностью, Variable - с переменной длительностью
TP	DINT	чтение/ запись	Длительность импульса, с

7.26.2. Режимы PulseDuration

Режимы PulseDuration:

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

7.26.3. Алгоритм PulseDuration

Алгоритм включает в себя следующие элементы:

- чтение;
- обработка данных.

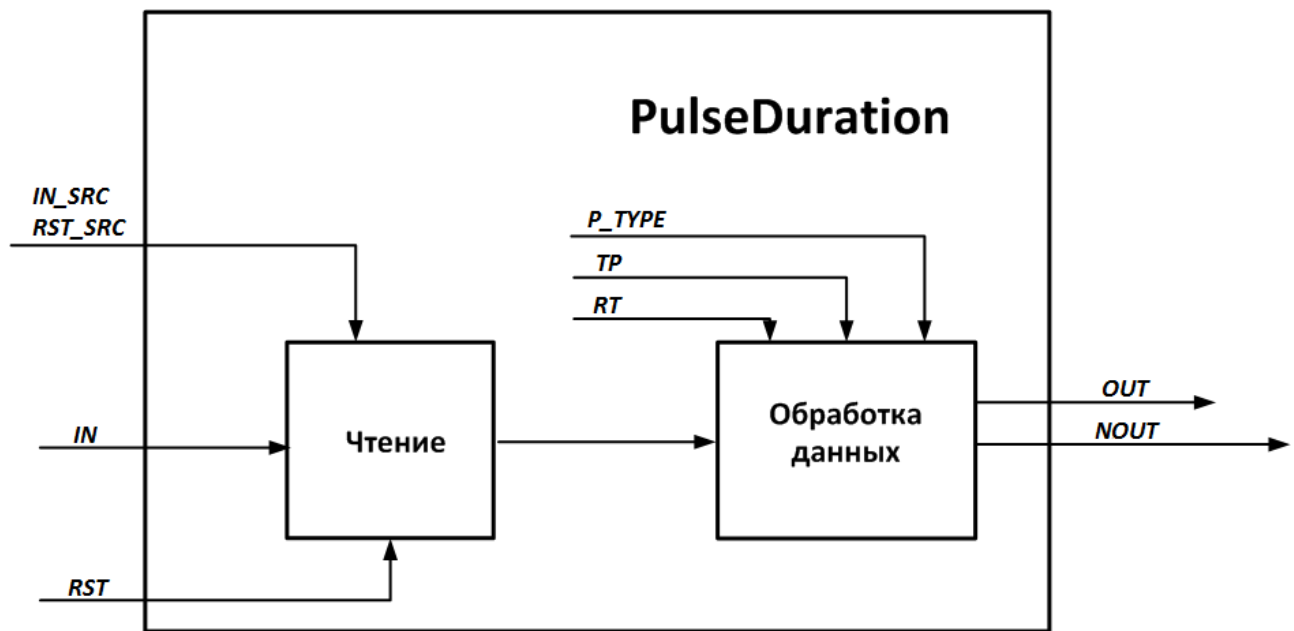


Рисунок 67. Процесс обработки данных PulseDuration

7.26.3.1. Чтение

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

7.26.3.2. Обработка данных

Различают типы импульса (атрибут *P_TYPE*):

- 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.

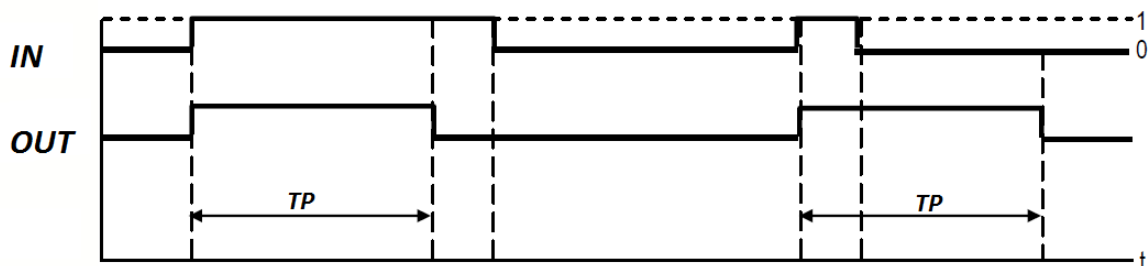


Рисунок 68. Импульс включения заданной длительности

На выходе *OUT* блока появится логическая «1» по фронту входного сигнала *IN*.

После запуска выход **OUT** не реагирует на изменение значения входного сигнала на интервале **TP**, а по истечении этого интервала сбрасывается в «0». В данном блоке вход **RST** не используется.

Импульс включения переменной длительности

Блок используется для формирования импульса включения выхода на переменный интервал времени. Работу поясняет приведенная на рисунке диаграмма.

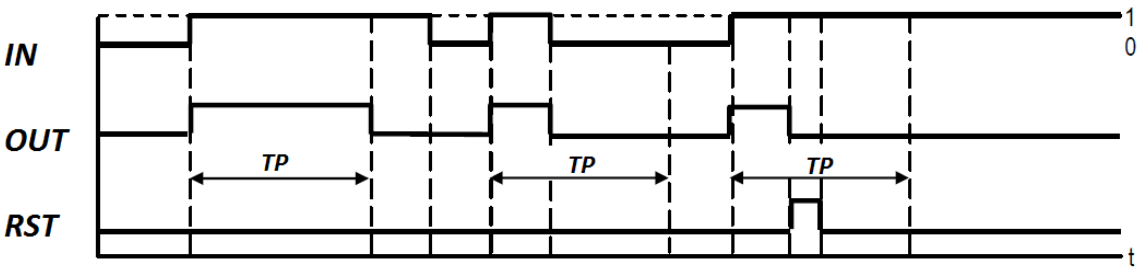


Рисунок 69. Импульс включения переменной длительности

На выходе **OUT** блока появится логическая «1» по фронту входного сигнала **IN**. Выход включается логической «1» на входе продолжительностью не менее длительности **TP**, а выключается по спаду входного сигнала либо когда команда сброса **RST** изменяется на "1".

7.27. Функциональный блок RSSRTrigger

Функциональный блок RSSR Trigger (далее RSSRTrigger) представляет собой асинхронный триггер с поддержкой тактирования.

Возможность тактирования обеспечивается подключением входа **EN** к источнику сигнала.

7.27.1. Атрибуты RSSRTrigger

RSSRTrigger имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 94. Дополнительные атрибуты RSSRTrigger

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
SET	BOOL	чтение	Входное значение установки
RST	BOOL	чтение	Входное значение сброса

Имя атрибута	Тип данных	Доступ	Описание атрибута
EN ²³	BOOL	чтение	Входное значение активации
Выходные атрибуты			
OUT	BOOL	чтение	Выходное значение
NOUT	BOOL	чтение	Выходное значение, инвертированное
Конфигурационные атрибуты			
SET_SRC	STRING	чтение/ запись	Источник данных для SET
RST_SRC	STRING	чтение/ запись	Источник данных для RST
EN_SRC	STRING	чтение/ запись	Источник данных для EN
OUT_INIT ²⁴	BOOL	чтение/ запись	Первоначальное значение выхода в нулевой момент времени
TRIG_TYPE	DINT	чтение/ запись	Тип триггера: RS - с приоритетом сброса, SR - с приоритетом установки

7.27.2. Режимы RSSRTrigger

Режимы RSSRTrigger:

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

7.27.3. Алгоритм RSSRTrigger

Алгоритм включает в себя следующие элементы:

- чтение;

²³ Атрибут **EN** предназначен для запуска триггера. При установке значения "1" включается стандартный тактируемый RSSR-триггер. При установке значения "0" включается состояние хранения, т.е. независимо от выходного значения атрибут **OUT** сохраняет предыдущее состояние.

²⁴ Атрибуту **OUT** присваивается состояние **OUT_INIT**, если в нулевой момент времени управляющими входами был сформирован состояние хранения.

- обработка данных.

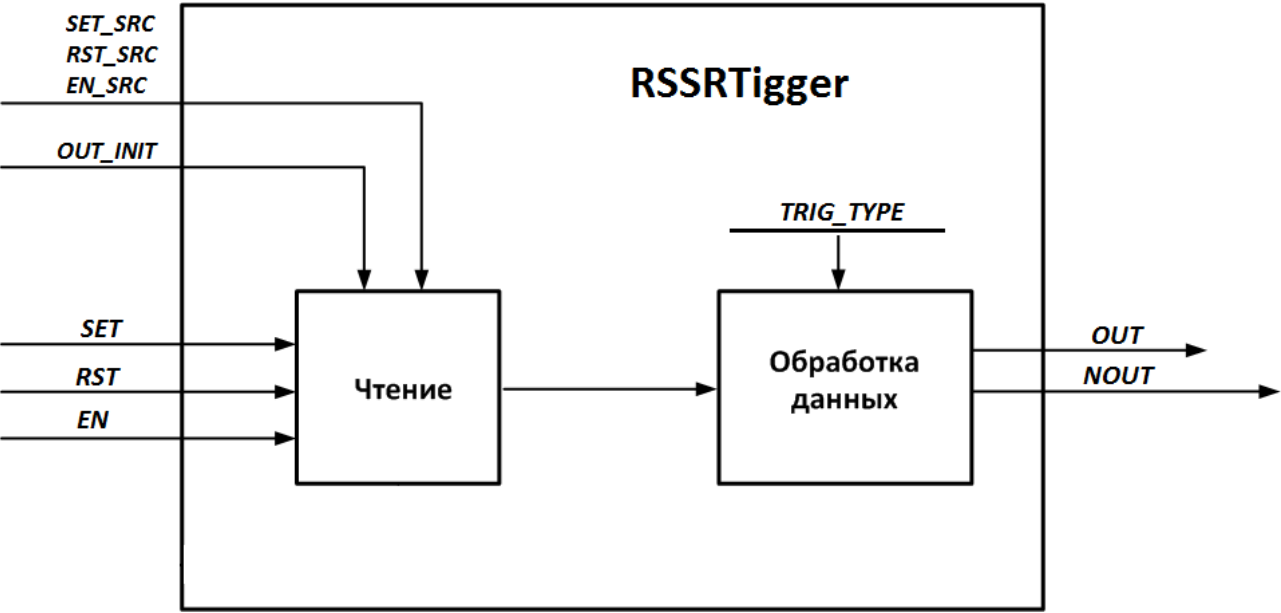


Рисунок 70. Процесс обработки данных RSSRTrigger

7.27.3.1. Чтение

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

7.27.3.2. Обработка данных

Блок RSSRTrigger поддерживает следующие типы триггеров (атрибут *TRIG_TYPE*):

- RSSRTrigger с приоритетом сброса - RS;
- RSSRTrigger с приоритетом установки - SR.

Таблица 95. Таблица перехода состояний блока RSSRTrigger 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	

R	S	Q(t)	Q(t+1)	Пояснения
1	1	0	0	R=S=1
1	1	1	0	

где $Q(t)$ - текущее состояние, $Q(t+1)$ - последующее состояние.

Таблица 96. Таблица перехода состояний блока RSSRTrigger 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 RSSRTrigger с приоритетом сброса

Работу блока RSSRTrigger с приоритетом сброса (RS) поясняет приведенная на рисунке диаграмма.

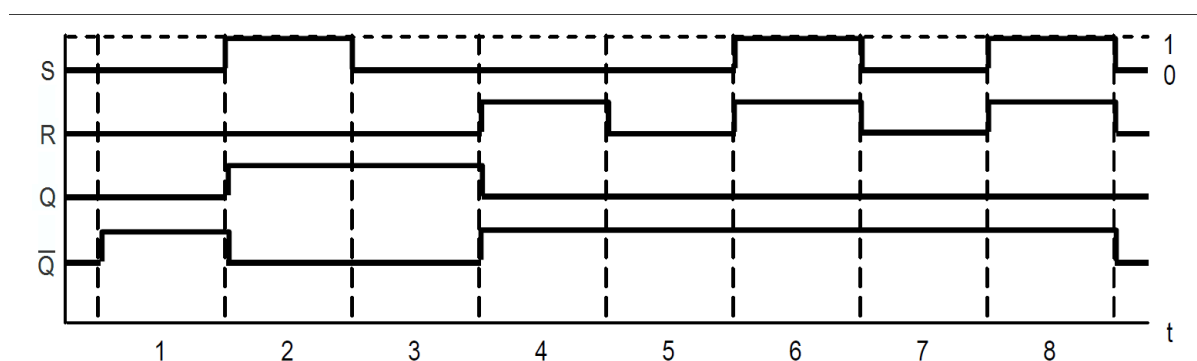


Рисунок 71. Временная диаграмма перехода состояний блока RSSRTrigger с приоритетом сброса

Функциональный блок RSSRTrigger с приоритетом установки

Работу блока RSSRTrigger с приоритетом установки (SR) поясняет приведенная на рисунке диаграмма.

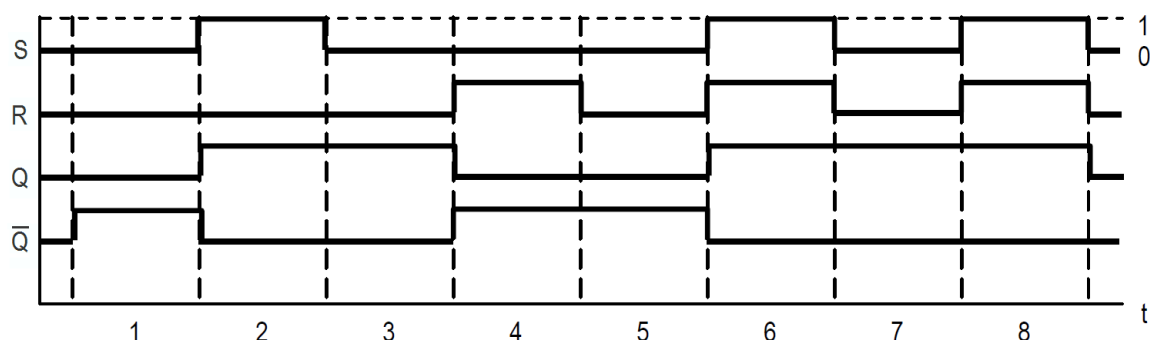


Рисунок 72. Временная диаграмма перехода состояний блока RSSRTrigger с приоритетом установки

7.28. Функциональный блок ToWords

Функциональный блок ToWords (далее ToWords) предназначен для разбиения входной величины на две части в соответствии с заданной конфигурацией.

7.28.1. Атрибуты ToWords

ToWords имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 97. Дополнительные атрибуты ToWords

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN1	REAL	чтение	Вход 1
IN2	DINT	чтение	Вход 2
Выходные атрибуты			
OUT1	DINT	чтение	Выход 1
OUT2	DINT	чтение	Выход 2
Конфигурационные атрибуты			
IN1_SRC	STRING	чтение/ запись	Источник 1
IN2_SRC	STRING	чтение/ запись	Источник 2

Имя атрибута	Тип данных	Доступ	Описание атрибута
MV1	DINT	чтение/ запись	Ручное значение 1
MV2	DINT	чтение/ запись	Ручное значение 2
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)

7.28.2. Режимы ToWords

Режимы ToWords:

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

7.28.3. Алгоритм ToWords

ToWords разбивает входную величину на две части: старшую и младшую. В качестве входной величины может быть использован тип данных REAL или DINT. 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 *IN1* или *IN2*, блок берет входные данные с подключенного входа.
2. Если подключены оба источника, приоритетным является *IN1*.

В режимах IMITATION и MASK_IMITATION входы блока отключаются, на выходы *OUT1*, *OUT2* подается значение атрибутов *MV1*, *MV2* соответственно.

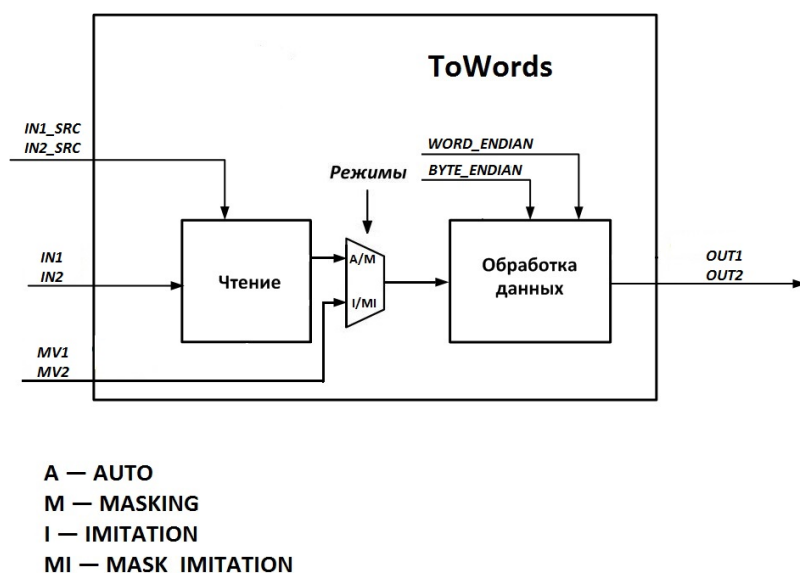


Рисунок 73. Процесс обработки данных ToWords

7.29. Функциональный блок WordsTo

Функциональный блок WordsTo (далее WordsTo) предназначен для комбинирования двух входных значений в одно выходное в соответствии с заданной конфигурацией.

7.29.1. Атрибуты WordsTo

WordsTo имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 98. Дополнительные атрибуты WordsTo

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN1	DINT	чтение	Вход 1
IN2	DINT	чтение	Вход 2
Выходные атрибуты			
OUT1	REAL	чтение	Выход 1
OUT2	DINT	чтение	Выход 2
Конфигурационные атрибуты			
IN1_SRC	STRING	чтение/ запись	Источник 1
IN2_SRC	STRING	чтение/ запись	Источник 2

Имя атрибута	Тип данных	Доступ	Описание атрибута
MV1	REAL	чтение/ запись	Ручное значение 1
MV2	DINT	чтение/ запись	Ручное значение 2
WORD_ENDIAN	DINT	чтение/ запись	Порядок слов (WordEndian)
BYTE_ENDIAN	DINT	чтение/ запись	Порядок байтов (ByteEndian)

7.29.2. Режимы WordsTo

Режимы WordsTo:

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

7.29.3. Алгоритм WordsTo

Входные величины *IN1*, *IN2* преобразуются согласно конфигурационным атрибутам *WORD_ENDIAN*, *BYTE_ENDIAN* и выдаются в *OUT1* в виде фиксированного типа данных REAL, а в *OUT2* в виде типа данных DINT.

В режимах IMITATION и MASK_IMITATION входы блока отключаются, на выходы *OUT1*, *OUT2* подается значение атрибутов *MV1*, *MV2* соответственно.

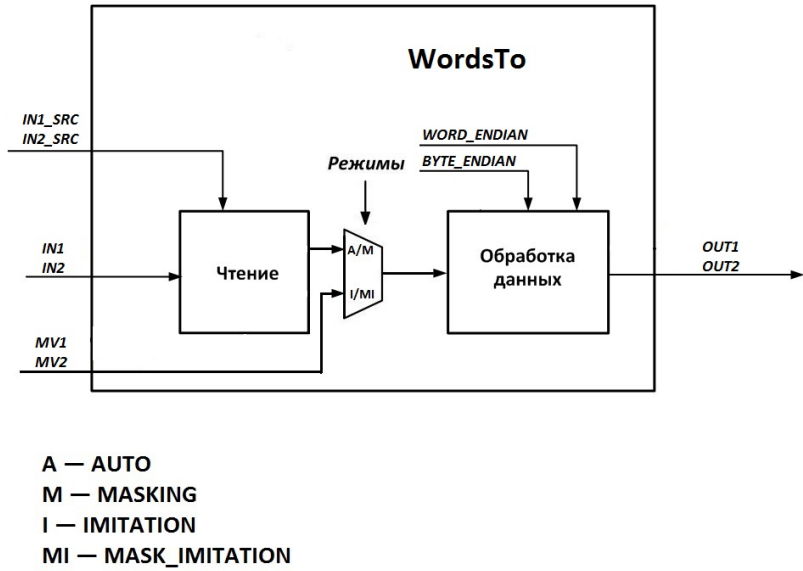


Рисунок 74. Процесс обработки данных WordsTo

7.30. Функциональный блок NotBlock

Функциональный блок NotBlock (далее NotBlock) инвертирует дискретный входной сигнал и генерирует дискретное выходное значение.

7.30.1. Атрибуты NotBlock

NotBlock имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 99. Дополнительные атрибуты NotBlock

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

7.30.2. Режимы NotBlock

Режимы NotBlock:

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

7.30.3. Алгоритм NotBlock

NotBlock формирует выходное значение, которое является отрицанием входа: если значение входа - истинно, то выход имеет ложное значение, и наоборот.

7.31. Функциональный блок ChangeAnalyzer

Функциональный блок ChangeAnalyzer (далее ChangeAnalyzer) 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".

7.31.1. Атрибуты ChangeAnalyzer

ChangeAnalyzer имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 100. Дополнительные атрибуты ChangeAnalyzer

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	Выбор фронта

7.31.2. Режимы ChangeAnalyzer

Режимы ChangeAnalyzer:

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

7.31.3. Алгоритм ChangeAnalyzer

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".

7.32. Функциональный блок DiscreteControl

Функциональный блок DiscreteControl (далее DiscreteControl) преобразует управляющий аналоговый выход ПИД регулятора в дискретный.

7.32.1. Атрибуты DiscreteControl

DiscreteControl имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 101. Дополнительные атрибуты DiscreteControl

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	REAL	чтение	Вход
Выходные атрибуты			
CMD2	BOOL	чтение	Команда "Больше"
CMD0	BOOL	чтение	Команда "Меньше"
MODEL_OUT	REAL	чтение	Положение в модели
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/запись	Источник
ARTM	DINT	чтение/запись	Время хода исполнительного механизма, с
MCTM	DINT	чтение/запись	Время цикла, с
MIN_PULSE_TM	REAL	чтение/запись	Минимальное время импульса, с

Имя атрибута	Тип данных	Доступ	Описание атрибута
OUT_TYPE	USINT	чтение/запись	Тип регулирования
PWM_TM	DINT	чтение/запись	Период ШИМ

Таблица 102. Диапазон значений

Атрибут	Условие	По умолчанию	Диапазон значений
ARTM	ARTM>MCTMx2	15	[2...9999]
MCTM	MCTM>MIN_PULSE_TMx2	3	[1...9999]
MIN_PULSE_TM	MIN_PULSE_TM>MCTM в режиме ThreePosition и MIN_PULSE_TM>PWM_TM в режиме OnOff	0,5	[0,1...9999]
PWM_TM	PWM_TM>MIN_PULSE_TMx2	15	[1...9999]

7.32.2. Режимы DiscreteControl

Режимы DiscreteControl:

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

7.32.3. Алгоритм DiscreteControl

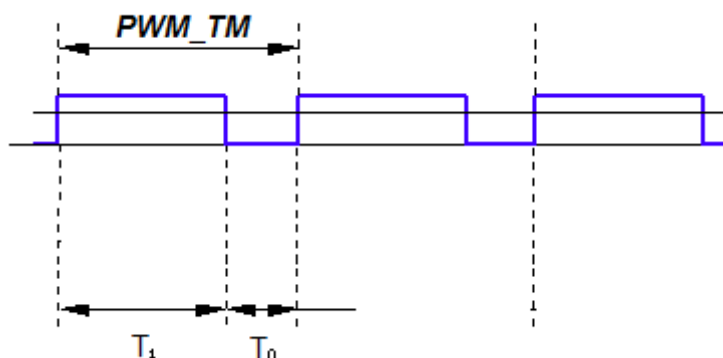
DiscreteControl предусматривает два типа управления:

- двухпозиционное (OnOff) посредством ШИМ-регулирования;
- трехпозиционное (ThreePosition).

Выбор типа управления осуществляется изменением атрибута **OUT_TYPE**, атрибут **PWM_TM** используется только в режиме OnOff, а атрибуты **ARTM**, **MCTM**, **MODEL_OUT** только в режиме ThreePosition.

В режиме OnOff блок работает как пропорциональный времени регулятор включения/выключения. Выходы **CMD2/CMD0** меняют временное соотношение состояний true/false пропорционально значению **IN**, полученного с ПИД регулятора. Выход **CMD0** является инверсией выхода **CMD2**.

Выход **CMD2** всегда выключен при 0% и включен при 100% значения управляющего входа **IN**. Например, если **PWM_TM** составляет 10 сек, при 70% значения управляющего входа **IN** время включения выхода **CMD2** составит 7 сек.



Период ШИМ (PWM_TM) величина постоянная, на этом периоде укладывается один импульс включения $T1$ ($CMD2 = true$), вычисляемый по формуле:

$$T1 = \frac{IN \times PWM_TM}{100}$$

В режиме ThreePosition блок осуществляет управление через два дискретных выхода $CMD2$ и $CMD0$.

На основании цифровой модели устройства подаются команды на открытие (больше) - выход $CMD2$, либо на закрытие (меньше) - выход $CMD0$. Синхронизация модели и реального исполнительного механизма выполняется в крайних точках путем подачи непрерывного сигнала $CMD2 = true$ при $IN = 100$ и $CMD0 = true$ при $IN = 0$.

$MCTM$ (время цикла) – это время через которое производится расчет рассогласования модели $MODEL_OUT$ и входа IN , с последующей синхронизацией модели.

Выходные управляющие импульсы ограничиваются по времени значением MIN_PULSE_TM .

Расчет и формирование времени импульса в зависимости от текущего рассогласования производится по формуле:

$$Impulse_time = \frac{|current_mismatch| \times ARTM}{100},$$

где $current_mismatch$ – текущее рассогласование.

Таблица 103. Значения $current_mismatch$

Условие	Значение
$IN - MODEL_OUT < MIN_PULSE_TM$	0

Условие	Значение
$ IN-MODEL_OUT > (MCTM-MIN_PULSE_TM) \times 100 / ARTM$ и $IN-MODEL_OUT > 0$	$MCTM \times 100 / ARTM$
$ IN-MODEL_OUT > (MCTM-MIN_PULSE_TM) \times 100 / ARTM$ и $IN-MODEL_OUT < 0$	$-MCTM \times 100 / ARTM$
В остальных случаях	$IN-MODEL_OUT$

Пока время импульса больше таймера цикла, производится формирование управляющих импульсов. Если рассогласование больше нуля, то **CMD2** = true, если рассогласование меньше нуля, то **CMD0** = true.

7.33. Функциональный блок TwoPositionController

Функциональный блок TwoPositionController (далее TwoPositionController) выполняет включение/выключение дискретного выхода (**CMD**) в зависимости от результата сравнения переменной процесса (**IN**) и значения задания (**SP**).

7.33.1. Атрибуты TwoPositionController

TwoPositionController имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 104. Дополнительные атрибуты TwoPositionController

Имя атрибута	Тип данных	Доступ	Описание атрибута
Входные атрибуты			
IN	REAL	чтение	Вход
Выходные атрибуты			
SP	REAL	чтение/запись	Задание
CMD	BOOL	чтение	Выход
CMDM	BOOL	чтение/запись	Ручное задание
Конфигурационные атрибуты			
IN_SRC	STRING	чтение/ запись	Источник
PV_DB_HYS	REAL	чтение/запись	Гистерезис

Имя атрибута	Тип данных	Доступ	Описание атрибута
GAIN	REAL	чтение/запись	Коэффициент усиления
MCTM	REAL	чтение/запись	Время итерации, с
IMPACT	USINT	чтение/запись	Направление воздействия

7.33.2. Режимы TwoPositionController

Режимы TwoPositionController:

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

7.33.3. Алгоритм TwoPositionController

Включение/отключение управляющего выхода происходит в результате сравнения отклонения входного сигнала с заданием. Гистерезис разделяет уставку на две, больше и меньше, тем самым позволяет уменьшить количество переключений реле.

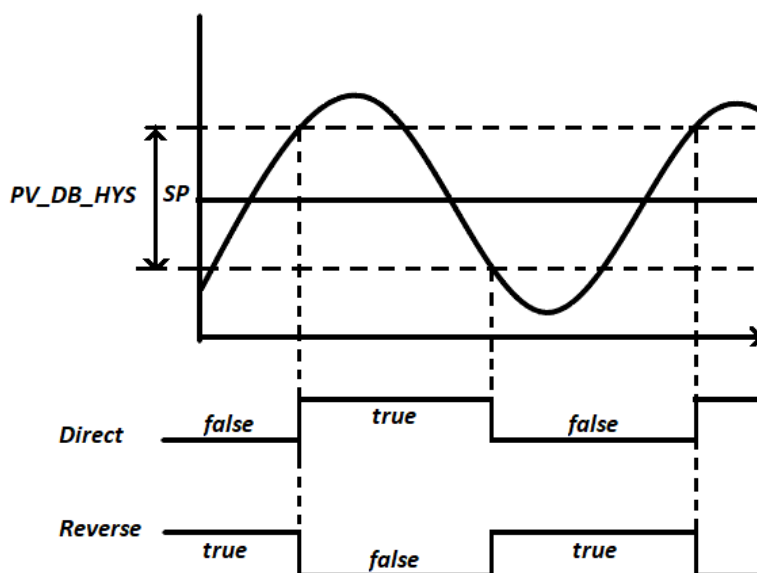


Рисунок 75. Алгоритм работы TwoPositionController

Если задан коэффициент усиления $GAIN > 0$ и период итерации $MCTM > 0$ (для периода итерации рекомендуется задавать значение кратное 0.1), тогда алгоритм работает с коррекцией по скорости изменения величины. Скорость изменения вводится в алгоритм как производная: изменение величины деленное на время, за

которое произошло измерение. Далее это измерение умножается на коэффициент **GAIN**. Коэффициент усиления подбирается в ручную и зависит от инертности системы и выбранного периода итерации.

8. Функциональные блоки противоаварийной защиты

8.1. Функциональный блок DiscreteVoter4/16

Функциональные блоки DiscreteVoter4 и DiscreteVoter16 (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 105. Дополнительные атрибуты DiscreteVoter4/16

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN16	BOOL	чтение	Вход
BPS_IN	BOOL	чтение	Байпас всех входов
BPS1_IN...BPS16_IN	BOOL	чтение	Байпас входа
RST_IN	BOOL	чтение	Сброс входного значения
BPS_LA_IN	BOOL	чтение	Байпас выхода
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выходное значение
NFOUT1...NFOUT16	BOOL	чтение	Отражение состояния входов на момент срабатывания воутера
RST_RDY	BOOL	чтение	Готовность к сбросу
TRP_UL_NO_BPS	BOOL	чтение	Выходное значение без байпасов

Имя атрибута	Тип данных	Доступ	Описание атрибута
TRP_UL	BOOL	чтение	Выходное значение без "защелок" (RQ_RST)
TRP_LA	BOOL	чтение	Выходное значение с RQ_RST, минуя выходной байпас
Конфигурационные атрибуты			
IN1_SRC...IN16_SRC	STRING	чтение/ запись	Источник входа
BPS1...BPS16	BOOL	чтение/ запись	Включение байпаса
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, с
NR_DLY	DINT	чтение/ запись	Задержка переключения в норму, с
RQ_RST	BOOL	чтение/ запись	Выход требует сброса после защиты
RST	BOOL	чтение/ запись	Сброс выходного значения

Имя атрибута	Тип данных	Доступ	Описание атрибута
BPS_PERM	BOOL	чтение/ запись	Опция, позволяющая установить байпас, если включена
AOS_TRP_DLY	DINT	чтение/ запись	Задержка переключения в защиту при неисправности всех сигналов
AOS_NR_DLY	DINT	чтение/ запись	Задержка переключения в норму после неисправности всех сигналов
BPS_IN_SRC	STRING	чтение/ запись	Источник байпаса всех данных
BPS1_IN_SRC... BPS16_IN_SRC	STRING	чтение/ запись	Источник байпаса входа
RST_IN_SRC	STRING	чтение/ запись	Источник сброса воутера
BPS_LA	BOOL	чтение/ запись	Байпас выхода конфигурационный
BPS_LA_IN_SRC	STRING	чтение/ запись	Источник байпаса выхода
PRCS_IN_STS	BOOL	чтение/ запись	Учитывать статусы входов (если статус не Ok, то воутер в BadConfiguration)
DEG_OPTS	USINT	чтение/ запись	Опции деградации
Диагностические атрибуты			
TRP_STS	DINT	чтение	Статус воутера

Имя атрибута	Тип данных	Доступ	Описание атрибута
AOS_LAST_ALIVE	DINT	чтение	Последние входа с хорошим статусом перед AOS ²⁵ (1-16 бит)
AOS_TRP_ET	DINT	чтение	Таймер переключения в защиту при неисправности всех сигналов
AOS_NR_ET	DINT	чтение	Таймер переключения в норму после неисправности всех сигналов
AOS	BOOL	чтение	Неисправность всех входов
ET1...ET16	DINT	чтение	Таймер канала
A_NUM_TO_TRP	USINT	чтение	Общее количество голосов для перевода блока в защиту
VOTE1...VOTE16	BOOL	чтение	Голосование канала
IN1_STS...IN16_STS	DINT	чтение	Статус входного канала
VOTE1_NO_BPS... VOTE16_NO_BPS	BOOL	чтение	Голосование без байпасов
LA_TS	DATE_AND_TIME	чтение	Дата и время последней сработки
TRP_VOTE_NO_DLY	UINT	чтение	Проголосовавших входов без задержки
TRP_VOTE_NO_BPS	UINT	чтение	Проголосовавших входов с задержками до байпаса
TRP_VOTE	UINT	чтение	Проголосовавших входов

²⁵ AOS — 32-битное число.

Имя атрибута	Тип данных	Доступ	Описание атрибута
A_NUM_TOT	USINT	чтение	Расчетное количество входов
BPS_CNT	USINT	чтение	Количество байпасов
BAD_CNT	USINT	чтение	Количество недостоверных входов

8.1.2. Алгоритм DiscreteVoter4/16

BypassOpts - опции байпасов:

- BypassReduceNumVoteToTrip - байпас уменьшает количество голосов, необходимых для перехода в защиту.

При подключении внешнего источника на входа байпасов (например, **BPS_IN/BPS_IN_SRC**) конфигурационный атрибут **BPS** будет перезаписываться значением из **BPS_IN**. ConfigVersion при этом обновится.

StatusOpts - опции обработки статуса входа:

- AlwaysUseValue – при голосовании используется значение входа, несмотря на его статус;
- WillNotVoteIfBad – вход не голосует, если его статус не ОК;
- VoteToTripIfBad – вход голосует за сработку, если его статус отличен от ОК.;
- DegradationBySignalMalfunction - если какой-то из входов становится неисправным (статус не ОК), то **A_NUM_TO_TRIP** уменьшается на количество неисправных входов. При этом все неисправные входа не приводят к срабатыванию;
- AutomaticOverrideSwitch - AOS - при неисправности всех входов происходит сработка воутера;
- DegradationWithAOS – DegradationBySignalMalfunction + AOS.

Примеры работы воутера с разными опциями обработки статуса входа.

Конфигурация: входов - 3, количество голосов для защиты - 2.

- AlwaysUseValue 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Выход	TripStatus
1	1	1	Норма	Normal
1	1	0	Норма	Normal
1	0	0	Защита	Tripped
1	1	1	Норма	Normal
1	1	0	Норма	Normal
1	0	0	Защита	Tripped

■ - статус канала BAD

- WillNotVoteIfBad 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Выход	TripStatus
1	1	1	Норма	Normal
1	1	0	Защита	TripInheibited
1	0	0	Защита	TripInheibited

■ - статус канала BAD

- VoteToTripIfBad 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Выход	TripStatus
1	1	1	Норма	Normal
1	1	1	Защита	Tripped
1	0	0	Защита	Tripped

■ - статус канала BAD

- DegradationBySignalMalfunction 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Выход	TripStatus
1	1	1	Норма	Normal
1	1	1	Норма	Normal
1	1	1	Норма	Normal
1	1	1	Защита	TripInheibited

■ - статус канала BAD

- AutomaticOverrideSwitch(AOS) 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Выход	TripStatus
1	1	1	Норма	Normal
1	1	0	Защита	TripInheibited
1	0	0	Защита	TrippedByAOS

■ - статус канала BAD

- DegradationWithAOS 2oo3

Значение входа 1	Значение входа 2	Значение входа 3	Архитектура	Выход	TripStatus
1	1	1	2003	Норма	Normal
1	1	1	2002	Норма	Normal
1	1	1	1001	Норма	Normal
1	1	1		Защита	TrippedByAOS

 - статус канала BAD

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 **TRP_DLY** > 0;
- VotedNormalDelayed – голосование за норму, задержка, если атрибут **NR_DLY** > 0;
- VotedToTripDelayedAOS – задержка перед срабатыванием воутера по AOS, если атрибут **AOS_TRP_DLY** > 0;
- VotedToNormalDelayedAOS - задержка перед выходом в норму после AOS, если атрибут **AOS_NR_DLY** > 0;
- ResetReady – воутер готов к сбросу (если установлен флаг **RQ_RST**).

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**).

8.2. Функциональный блок CauseEffectMatrix16

Функциональный блок CauseEffectMatrix16 (далее CauseEffectMatrix16) предназначен для определения логики блокировок и разрешений, которая связывает до 16 входов и выходов между собой и до 128 входов и выходов при каскадном подключении.

8.2.1. Атрибуты CauseEffectMatrix16

CauseEffectMatrix16 имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 106. Дополнительные атрибуты CauseEffectMatrix16

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
C1...C16	BOOL	чтение	Вход причины
RST1_PERM...RST16_PERM	BOOL	чтение	Разрешение на сброс защиты
RST1...RST16	BOOL	чтение	Сброс защиты
CHAIN_IN	USINT	чтение	Вход для каскадного блока
RST_PERM	BOOL	чтение	Разрешение на сброс в каскаде
RST_IN	BOOL	чтение	Вход сброса всех блоков в каскаде
<i>Выходные атрибуты</i>			
E1...E16	BOOL	чтение	Выход защиты

Имя атрибута	Тип данных	Доступ	Описание атрибута
STS1...STS16	DINT	чтение	Статус выхода защиты
CHAIN_OUT	USINT	чтение	Выход для каскадного блока
RST1_AVAIL...RST16_AVAIL	BOOL	чтение	Сброс доступен
RST_AVAIL	BOOL	чтение	Общий сброс доступен
Конфигурационные атрибуты			
C1_SRC...C16_SRC	STRING	чтение/ запись	Вход причины 1...16 источник
NR_DLY1...NR_DLY16	DINT	чтение/ запись	Задержка перехода в норму выход, с
TRP_DLY1...TRP_DLY16	DINT	чтение/ запись	Задержка перехода в сработку выход, с
RQ_RST1...RQ_RST16	BOOL	чтение/ запись	Требуется сброс для возвращения в норму выход
RST1_PERM_SRC... RST16_PERM_SRC	STRING	чтение/ запись	Разрешение на сброс защиты источник
RST1_SRC...RST16_SRC	STRING	чтение/ запись	Сброс защиты источник

Имя атрибута	Тип данных	Доступ	Описание атрибута
OVR_VAL1...OVR_VAL16	USINT	чтение/ запись	Безусловное переключение защиты (блокировка защиты или нормы)
CHAIN_IN_SRC	STRING	чтение/ запись	Вход для каскадного блока источник
CHAIN_MX_NUM	USINT	чтение/ запись	Номер блока в каскаде
RST_PERM_SRC	STRING	чтение/ запись	Разрешение на сброс в каскаде источник
RST_IN_SRC	STRING	чтение/ запись	Сброс всех блоков в каскаде источник
C_MASK_1_1... C_MASK8_16	UINT	чтение/ запись	Матрица причин и следствий защиты
RST	BOOL	чтение/запись	Сброс всех блоков в каскаде
Диагностические атрибуты			
ACT_CAUSES1... ACT_CAUSES16	UINT	чтение	Текущие активные причины защиты
FOUT_1_1...FOUT_8_16	UINT	чтение	Матрица первопричин
FOUT	DINT	чтение	Первопричина блока

Имя атрибута	Тип данных	Доступ	Описание атрибута
E1_TS...E16_TS	DATE_AND_TIME	чтение	Метка времени первопричины канала защиты
FOUT_TS	DATE_AND_TIME	чтение	Метка времени первопричины блока
<i>Сигнализационные атрибуты</i>			
EFFECT_FORCED	BOOL	чтение	Сигнализация форсирования защиты
EFFECT_TRIPPED	BOOL	чтение	Сигнализация сработки защиты
EFFECT_FORCED_Priority	BOOL	чтение	Приоритет сигнализации форсирования защиты
EFFECT_TRIPPED_Priority	BOOL	чтение	Приоритет сигнализации сработки защиты

8.2.2. Режимы CauseEffectMatrix16

Режимы CauseEffectMatrix16:

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

Конфигурационный атрибут *C_MASK_1_1... C_MASK_8_16* позволяет связать вход причину матрицы с выходом защиты текущей матрицы.

Для включения CauseEffectMatrix16 в каскадное соединение необходимо подключить вход для каскадного блока **CHAIN_IN** (вход цепочки) настраиваемого блока к выходу для каскадного блока **CHAIN_OUT** (выход цепочки) предыдущего блока, таким образом выстраивая цепочку блоков, которые будут обрабатываться

последовательно. Всего в каскадном соединении можно связать до 8 блоков CauseEffectMatrix16. Каждому блоку в каскаде необходимо назначить порядковый номер от 1 до 8 в атрибуте **CHAIN_MX_NUM**. Номер блока в каскаде **CHAIN_MX_NUM** должен соответствовать конфигурации **C_MASK_1_1...C_MASK_8_16** в настройках связей всех блоков, включенных в каскад. Для правильной работы необходимо замкнуть цепь блоков, соединив первый и последний блоки в каскаде. При ошибке конфигурации каскада, все блоки, включенные в группу, перейдут в состояние "BadConfiguration" (ошибка конфигурации).



Внимание: CauseEffectMatrix16 допускает только последовательное включение блоков. Подключение двух и более блоков CauseEffectMatrix16 к одному через связь **CHAIN_OUT** - **CHAIN_IN** недопустимо, и может привести к неправильной работе всего каскада.

Для включения блока CauseEffectMatrix16 в одиночный режим необходимо сбросить атрибут **CHAIN_MX_NUM** в 0. В одиночном режиме используются связи только с атрибутов **C_MASK_1_1...C_MASK_8_1**.

8.2.3. Алгоритм CauseEffectMatrix16

Выходной атрибут **E1...E16** имеет два состояния - "Норма" и "Защита". Переключение **E1...E16** определяется текущим состоянием канала подсистемы защиты, которое определяется конфигурацией блока и состоянием входных сигналов **C1...C16**. Состояние канала подсистемы защиты для выхода **E1...E16** представлено атрибутом **STS1...STS16** - статус выхода защиты. Выходные атрибуты **RST1_AVAIL...RST16_AVAIL** срабатывают при переходе канала из состояния "Защита - Норма" или в состояние "Требуется сброс". Устанавливаются в 1 (true) в случае фиксации первопричины и перехода соответствующего выхода защиты (**E1...E16**) в норму. **RST_AVAIL** - общий выход разрешения сброса для всей матрицы или цепочки матрицы. Логика срабатывания та же, что и у **RST1_AVAIL...RST16_AVAIL**.

В **E1_TS...E16_TS** записывается метка времени, соответствующая моменту фиксации первопричины при переходе канала в состояние "Защита". В **FOUT_TS** записывается метка времени, соответствующая моменту фиксации любой из первопричин какого-либо канала защиты.

Таблица 107. Основные состояния канала подсистемы защиты

Состояние канала подсистемы защиты	Состояние выхода <i>E1...E16</i>
Норма (Normal state)	Норма (1)
Защита (Tripped state)	Защита (0)
Задержка норма - защита (Normal to tripped delayed state)	Норма (1)
Задержка защита - норма (Tripped to normal delayed state)	Защита (0)
Готов к сбросу (Ready to reset)	Защита (0)

Таблица 108. Блокировка Защиты или Нормы

OverrideType	Значение	Описание
OverrideDisable	0	Блокировка отключена
OverrideToTrip	1	Заблокировать в состоянии "Защита"
OverrideToNormal	2	Заблокировать в состоянии "Норма"

8.2.3.1. Работа канала в состоянии "Защита"

При инициализации блока все выходы *E1...E16* и каналы подсистемы защиты переводятся в состояние "Tripped (Защита)". В этом состоянии анализируются все активные причины *C1...C16*, влияющие на данный канал, в соответствии с конфигурацией матрицы причин и следствий защиты *C_MASK_1_1...C_MASK_8_16*, и если таковые отсутствуют и атрибут *OVR_VAL1...OVR_VAL16* не установлен в "OverrideToTrip", канал подсистемы защиты переходит в состояние "Tripped to normal delayed (Задержка защита-норма)". По истечении времени *NR_DLY1...NR_DLY16* или если *NR_DLY1...NR_DLY16* установлен в 0, канал подсистемы защиты перейдет в состояние "Ready to reset (Готов к сбросу)". В этом состоянии проверяется настройка *RQ_RST1...RQ_RST16*, и если она сброшена, то канал переходит в состояние "Normal (Норма)", в противном случае проверяется состояние атрибутов как в следующем псевдокоде

```
((RST1_PERM...RST16_PERM && RST1...RST16) || (RST_IN || RST) && RST_PERM) == 1
```

где *RST1...RST16*, *RST1_PERM...RST16_PERM* - поканальные входы сброса защиты и разрешения сброса защиты. *RST_IN*, *RST*, *RST_PERM* - общие входы сброса и разрешения сброса всех блоков в каскаде. При выполнении равенства канал подсистемы защиты переходит в состояние "Normal (Норма)".

8.2.3.2. Работа канала в состоянии "Норма"

В этом состоянии анализируются все активные причины $C1...C16$, влияющие на данный канал подсистемы защиты, в соответствии с конфигурацией матрицы причин и следствий защиты $C_MASK_1_1...C_MASK_8_16$, и если они обнаружены и атрибут $OVR_VAL1...OVR_VAL16$ не установлен в "OverrideToNormal", канал переходит в состояние "Normal to tripped delayed (Задержка норма-защита)" с записью матрицы первопричин $FOUT_1_1...FOUT_8_16$. По истечении времени $TRP_DLY1...TRP_DLY16$ или если $TRP_DLY1...TRP_DLY16$ установлен в 0, канал подсистемы защиты перейдет в состояние "Tripped (Защита)", также будет произведено обновление матрицы первопричин $FOUT_1_1...FOUT_8_16$.

8.2.3.3. Блокировка переходов каналов подсистемы защиты

Блокировка переходов каналов подсистемы защиты в защиту и норму осуществляется путем установки атрибутов $OVR_VAL1...OVR_VAL16$. Для отключения блокировки следует установить для данного канала $OVR_VAL1...OVR_VAL16$ в значение "OverrideDisable".

9. Настройка видимости атрибутов в редакторе диаграмм

Функциональные блоки AnalogInputPoint, CauseEffectMatrix16, DiscretePoint, DiscreteVoter4, DiscreteVoter16, Logical Block имеют атрибуты, видимость которых в редакторе диаграмм можно настроить.

Настраиваются атрибуты в **Forge > Контроллер > Технологическая программа > Loop > Функциональный блок**:

1. Перейдите во вкладку **Атрибуты**.

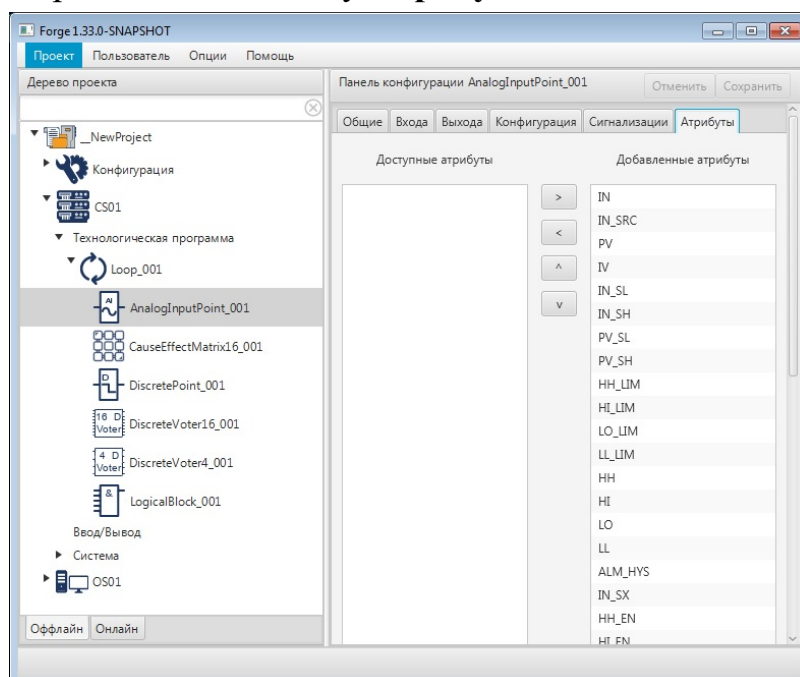


Рисунок 76. Вкладка атрибуты

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, зажмите на клавиатуре кнопку **Ctrl**, для выделения всех атрибутов зажмите на клавиатуре **Ctrl** + **A**.

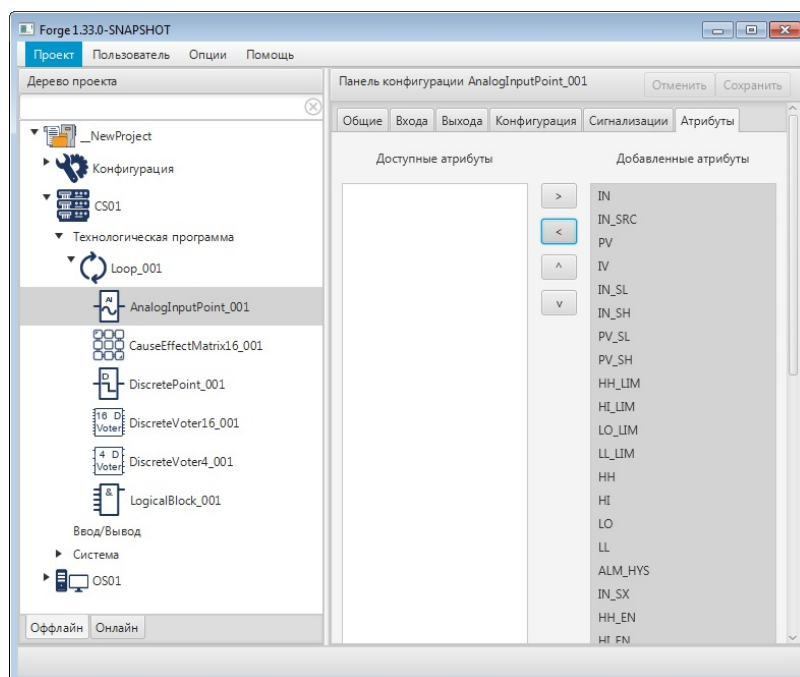


Рисунок 77. Выделение атрибутов

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).

Рисунок 78. 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 109. 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:

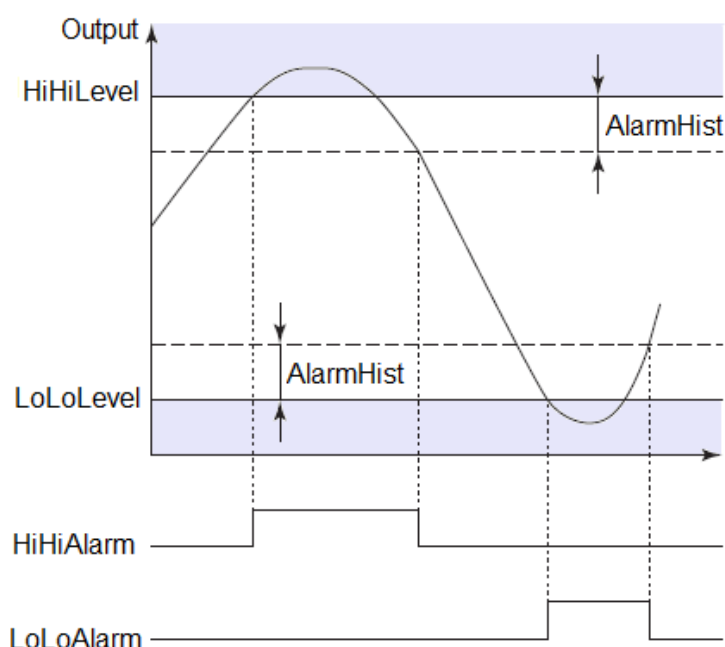


Рисунок 79. Проверка сигнализации аварийного верхнего/аварийного нижнего предела

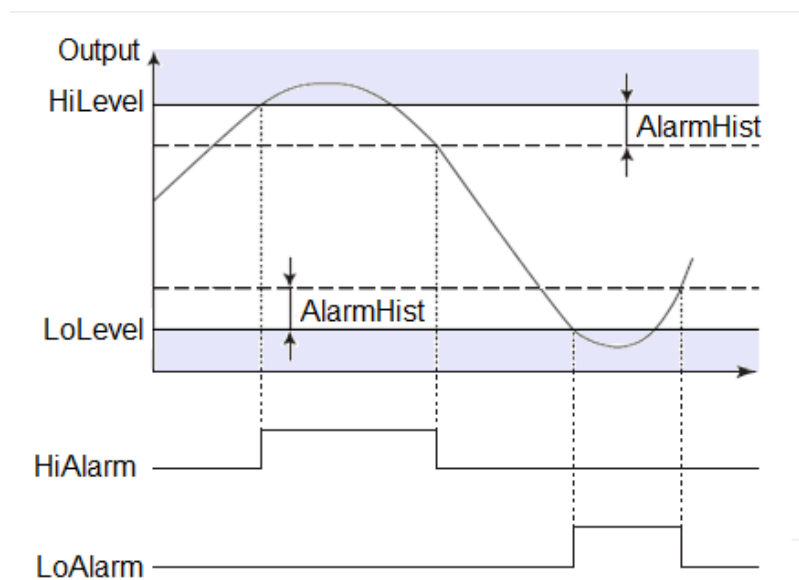


Рисунок 80. Проверка сигнализации предупредительного верхнего/предупредительного нижнего предела