

Концепция технологического ПО

Содержание

1. Условные обозначения и термины.....	3
1.1. Условные обозначения.....	3
1.2. Перечень терминов и сокращений.....	3
2. Введение.....	7
3. Роли функциональных блоков на узлах РСУ.....	8
3.1. Роль функционального блока в контроллере РСУ.....	8
3.2. Роль функционального блока на станции инженера.....	8
3.3. Роль функционального блока на станции оператора.....	8
3.4. Роль функционального блока на станции интеграции.....	9
4. Атрибуты функциональных блоков.....	10
4.1. Базовый набор атрибутов функционального блока.....	12
4.2. Атрибут STATUS.....	13
4.3. Атрибут ADT_STATUS.....	16
5. Объединение функциональных блоков в контур.....	18
6. Режим функционального блока.....	20
6.1. Режим "Выключен".....	21
6.2. Режим "Автоматический".....	21
6.3. Режим "Маскирование".....	21
6.4. Режим "Имитация".....	21
6.5. Режим "Имитация с маскированием".....	21
6.6. Режим "Каскадный".....	22
6.7. Режим "Ручной".....	22
7. Рекомендации по невалидным ссылкам.....	23

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".

Контур

Группа функциональных блоков, связанных между собой как внутри технологической программы контроллера РСУ, так и между контроллерами РСУ в случае удаленного соединения.

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.

Межконтроллерное соединение

Обмен данными между технологическими программами двух контроллеров РСУ.

Мнемосхема

Схематическое отображение технологического процесса посредством среды визуализации на экране станции оператора.

Окно конфигурации функционального блока

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

ПО

Программное обеспечение.

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.

Сигнализация

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

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

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.

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.

OPC UA

Открытый протокол, предназначенный для обмена данными в системах промышленной автоматизации.

2. Введение

Документ "Концепция технологического ПО" относится к комплексу эксплуатационных документов программного обеспечения распределенной системы управления (ПО РСУ).

Данный документ предназначен для ознакомления пользователей системы с концепцией технологического ПО РСУ и принципами ее функционирования.

Технологическое программное обеспечение состоит из экземпляров функциональных блоков, которые выполняют функции автоматизации. Каждый экземпляр уникален и имеет собственное состояние и настройки.

Экземпляр принадлежит к определенному классу. Класс задает назначение и алгоритм работы функционального блока.

Например, технологическая программа обрабатывает несколько аналоговых сигналов с помощью функциональных блоков класса "Аналоговый блок". Каждый экземпляр функциональных блоков этой программы привязан к соответствующему входному аналоговому каналу и имеет собственные настройки. При этом все экземпляры работают по одному и тому же алгоритму, определенному их классом.

Функциональный блок имеет такие параметры как атрибуты и режим. Атрибуты предоставляют доступ к данным функционального блока. Режим определяет работу алгоритма функционального блока.



Внимание: Справочная информация доступна:

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

3. Роли функциональных блоков на узлах РСУ

3.1. Роль функционального блока в контроллере РСУ

В контроллере РСУ функциональный блок представляет собой компонент технологического ПО.

В каждой фазе системного цикла ⁴ выполняются функциональные блоки определенного типа:

- входных данных;
- технологические;
- выходных данных;
- системные.

3.2. Роль функционального блока на станции инженера

На станции инженера функциональный блок представляет собой объект настройки технологического ПО.

Станция инженера позволяет:

- хранить функциональный блок и его настройки в базе данных;
- загружать/выгружать функциональный блок и его настройки в контроллер/из контроллера;
- поддерживать контроль версий настроек функционального блока.

3.3. Роль функционального блока на станции оператора

Функциональный блок на станции оператора выполняет следующие функции:

- Выступает в качестве визуализированного компонента технологического ПО. Для визуального отображения функционального блока используются лицевая панель и окно конфигурации.
- Обеспечивает доступ к данным для анимаций посредством имени и атрибута.
- Обеспечивает доступ к историческим данным для трендов.
- Формирует сигнализации.

⁴ Подробнее о системном цикле см. "Архитектура и принципы построения распределенной системы управления" п. 4.1. Среда исполнения контроллера РСУ.

3.4. Роль функционального блока на станции интеграции

На станции интеграции функциональный блок представляет собой OPC UA объект, параметры которого представляют собой OPC UA теги.

4. Атрибуты функциональных блоков

Атрибуты функциональных блоков по своему назначению подразделяются на типы, представленные в таблице 1.

Таблица 1. Типы атрибутов функциональных блоков.

Тип атрибута	Назначение	Примеры
Входной	Содержит данные выходного атрибута функционального блока-источника данных	Атрибут IN аналогового блока, в который записываются показания аналогового датчика
Выходной	Содержит результат преобразования входных данных в зависимости от режима функционального блока	Атрибут OUT аналогового блока, в который записывается результат обработки входного значения
Конфигурационный	Содержит настройки для преобразования входных данных	Настройка шкалы масштабирования, параметры ПИД регулятора, время открытия задвижки
Диагностический	Содержит данные о состоянии функционального блока	Статус, версия конфигурации, время последнего обновления настроек функционального блока
Сигнализационный	Является источником сигнализаций. Если данный атрибут принимает значение true, функциональный блок генерирует сигнализацию в буфере сигнализаций. Сигнализация передается на станцию оператора по транспортной сети PCY	Атрибуты НН и LL аналогового блока, сигнализирующие о переходе выходного значения в диапазон соответственно верхних либо нижних аварийных значений



Внимание: Значения конфигурационных атрибутов записываются на диск и восстанавливаются после перезагрузки. Необходимо максимально минимизировать запись этих атрибутов для уменьшения износа NAND памяти.

Типы данных для атрибутов имеют диапазон максимального и минимального значений:

Таблица 2. Типы данных атрибутов функционального блока.

Тип данных	Обозначение	Объем занимаемой памяти, байт	Диапазон принимаемых значений
Целое знаковое число	INT (DINT)	4	-2147483648 / 2147483647
Целое беззнаковое число	UINT (UDINT)	4	0 / 4294967295
Целое знаковое число	SHORT	2	-32768 / 32767
Целое беззнаковое число	USHORT	2	0 / 65535
Целое знаковое число	SINT	1	-128 / 127
Целое беззнаковое число	USINT	1	0 / 255
Вещественное число	REAL	4	$\pm 1.175494351 \times 10^{-38}$ / $\pm 3.402823466 \times 10^{38}$
Строковая переменная	STRING	количество символов +2 байта	0 / 512
Булевый логический тип	BOOL	1	false/true
Тип время (дата и время)	TIME (DATE_AND_TIME)	6	1970-01-01 00:00:00.000 / 2151-06-06 23:59:59.999

4.1. Базовый набор атрибутов функционального блока

Все функциональные блоки вне зависимости от класса имеют один и тот же базовый набор атрибутов.

Таблица 3. Базовый набор атрибутов функционального блока

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
classid	INT	чтение	Идентификатор класса функционального блока
instanceid	INT	чтение	Идентификатор экземпляра функционального блока
TAGNAME	STRING	чтение	Название тэга (уникальный строковый идентификатор функционального блока)
description	STRING	чтение/ запись	Описание функционального блока
fp	INT	чтение/ запись	Имя лицевой панели
dt	STRING	чтение/ запись	Имя детальной панели
pic	INT	чтение/ запись	Имя мнемосхемы, которая откроется, если нажать на строчку сигнализации данного функционального блока в планшете сигнализаций
desc1...4	STRING	чтение/ запись	Описание панели
MODE	INT	чтение/ запись	Режим функционального блока
loop	STRING	чтение/ запись	Имя контура, к которому принадлежит функциональный блок
order	DINT	чтение/ запись	Порядок выполнения функционального блока в контуре

Имя атрибута	Тип данных	Доступ	Описание атрибута
structureUnitID	STRING	чтение/ запись	Принадлежность к структурной единице
includeInSearchResults	ARRAY OF STRING	чтение/ запись	Список станций оператора, на которых блок будет доступен из меню поиска
Диагностические атрибуты			
configVersion	INT	чтение	Номер версии конфигурации
configTimestamp	TIME	чтение	Временная метка конфигурации (время, когда были внесены последние изменения в конфигурационные атрибуты функционального блока)
timestamp	TIME	чтение	Временная метка (показывает время последнего опроса функционального блока)
STATUS	INT	чтение	Статус функционального блока
BAD	BOOL	чтение	Плохое качество (сигнализационный флаг об ошибке функционального блока)
ADT_STATUS1	DINT	чтение	Дополнительный статус основного модуля
ADT_STATUS2	DINT	чтение	Дополнительный статус резервного модуля

4.2. Атрибут STATUS

Функциональный блок предоставляет информацию о собственном состоянии при помощи атрибута **STATUS**.

Значение **STATUS** представляет собой код в виде составного числа.

Код включает в себя следующие поля:

- класс;
- первичный номер;
- уточняющий номер.

Пример записи кода:

0x02030001, где

0x02 - класс;

0x0203 - первичный номер;

0x02030001 - уточняющий номер.

Таблица 4. Диапазоны полей кода

	Класс	Первичный номер	Уточняющий номер
бит	31-----24	23-----16	15-----0

Таблица 5. Классы кода

Класс	Обозначение	Описание
0x00	OK	Функциональный блок работает штатно
0x01	Uncertain	Функциональный блок находится в неопределенном состоянии
0x02	Bad ⁵	Функциональный блок работает некорректно

Таблица 6. Значения атрибута *Status*

Код	Обозначение/ сокращение	Описание
Класс OK		
0x00000000	OK	Функциональный блок работает штатно
Класс Uncertain		
0x01010000	Uninitialized UNIN	Функциональный блок не инициализирован
0x01020000	WaitForService WFS	Функциональный блок ожидает получение данных

⁵ При возникновении статусов, имеющих класс кода Bad и выше, формируется сигнализационный флаг об ошибке **BAD**.

Код	Обозначение/ сокращение	Описание
0x01030000	ResolvingReference RREF	Функциональный блок находится в поиске источника данных
0x01040000	UncertainSourceStatus USS	Функциональный блок обнаружил источник, но источник данных еще не инициализирован
Класс Bad		
0x02020000	BadSourceReference BSR	Неверная ссылка на источник данных
0x02020001	BadTypeReference BTR	Атрибут-источник данных не соответствует требуемому типу
0x02030000	BadConfiguration BCON	Неверная конфигурация функционального блока
0x02030001	BadInput BINP	Данные источника не соответствуют техническим параметрам, например, превышен допустимый диапазон
0x02040000	BadSourceStatus BSS	Данные от источника имеют признак плохого качества
0x02050000	ProgramError EPRO	Сбой программы пользователя в функциональном блоке Program или в любом функциональном блоке в режиме исполнения
0x02060000	HardwareError EHAR	Аппаратный сбой в работе, относится к модулям ВВ в том числе CPU
0x02070000	CommunicationError ECOM	Сбой передачи данных на сетевом уровне

Код	Обозначение/ сокращение	Описание
0x02080000	RedundandModulesError RME	Оба модуля из резервной пары находятся в состоянии ошибки
0x02090000	BadChannelInput BCI	Ссылка правильная, но данные исходного канала имеют неправильное значение

Атрибут **STATUS** функционального блока предоставляет информацию о его состоянии, включая возможные ошибки и некорректные состояния. Статус функционального блока может изменяться в зависимости от статуса его источника данных: если функциональный блок-источник данных находится в состоянии Bad, это состояние передается функциональному блоку-приемнику. Исключением является обращение к атрибуту **STATUS** блока-источника, при котором состояние Bad не влияет на блок-приемник.

4.3. Атрибут ADT_STATUS

Функциональный блок (модуль) предоставляет информацию о собственном уточненном состоянии при помощи атрибута **ADT_STATUS**. Атрибут **ADT_STATUS** имеет тип данных DINT, объем занимаемой памяти — 4 байта, где в первых двух байтах хранится состояние для первого (основного) модуля, а в третьем и четвертом байтах состояние второго (резервного) модуля.

Поле дополнительных статусов представляет собой 16 разрядный битовый массив. Код статуса присваивается путем установки 1 в соответствующем разряде.

Таблица 7. Дополнительная диагностика

Значение переменной	Разряд	Обозначение	Описание
0b0000 0000 0000 0000	-	NoError	Нет ошибок
0b0000 0000 0000 0001	0	kPSUPowerFault	Напряжение отсутствует (только у модулей питания и только при резерве питания)

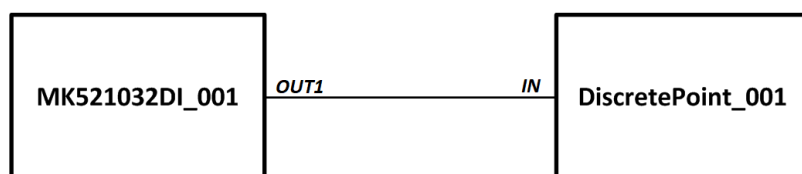
Значение переменной	Разряд	Обозначение	Описание
0b0000 0000 0000 0010	1	kPSUConfSwitchesError	Положение "Вертушков" на основном и резервном модуле питания допустимые, но не совпадают (только у модулей питания и только при резерве питания)
0b0000 0000 0000 0100	2	kPSULose	Нет связи с основным (слева) модулем питания (только у модулей питания и только при резерве питания)
0b0000 0000 0000 1000	3	kPSUSwitchesFault	"Вертушки" на модуле питания в запрещенном положении (только для модулей питания)
0b0000 0000 0001 0000	4	kLeftModuleLose	Нет связи с модулем слева (только для модулей ввода/вывода)
0b0000 0000 0010 0000	5	kCanConfError	Ошибка конфигурации сети CAN ("Вертушки" модуля питания слева в запрещенном положении) (для всех модулей)
0b0000 0000 0100 0000	6	kInternalFault	Внутренний отказ
0b0000 0000 1000 0000	7	kFirstBusError	Обрыв связи по шине CAN0
0b0000 0001 0000 0000	8	kSecondBusError	Обрыв связи по CAN1

5. Объединение функциональных блоков в контур

Для того чтобы объединить группу функциональных блоков в контур, применяются специальные ссылки.

Ссылка представляет собой конфигурационный атрибут, содержащий имя функционального блока-источника данных и его выходной атрибут.

Пример. В среде разработки необходимо увязать выходной атрибут output1 дискретного модуля MK521032DI_001 со входом дискретного блока DiscretePoint_001:



С этой целью в дереве проекта нужно:

1. Выбрать функциональный блок DiscretePoint_001:

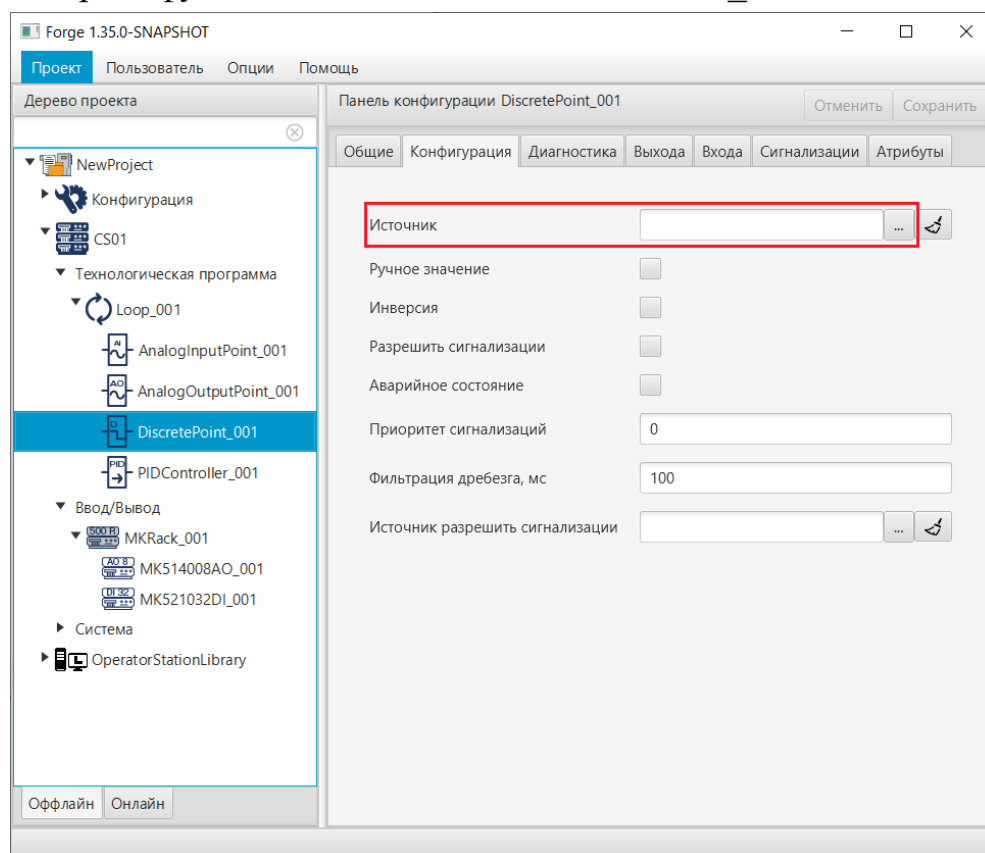
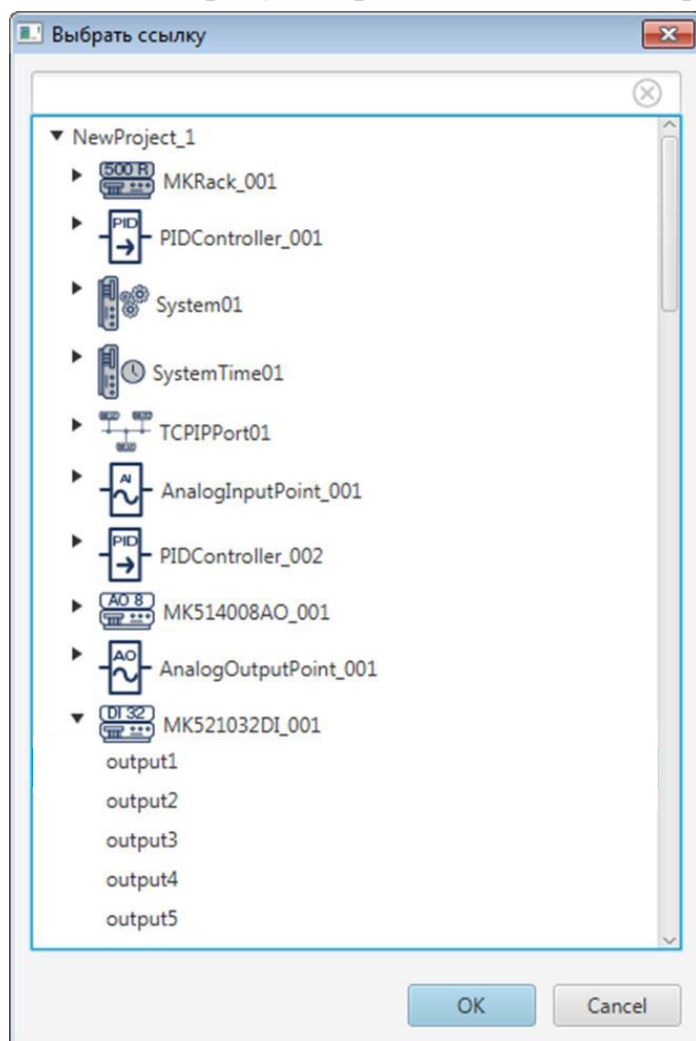


Рисунок 1. Вкладка "Настройки" функционального блока

2. В поле источник активировать окно **Выбрать ссылку** и выбрать требуемый выходной атрибут output1 во вкладке дискретного модуля MK521032DI_001:



Примечание. Пошаговую инструкцию см. в Руководстве по созданию технологического программного обеспечения контроллера PCY, п. Настройка соединений функциональных блоков.

Функциональные блоки могут взаимодействовать через ссылки на атрибуты других блоков. Если функциональный блок Program в своей логике обращается к атрибутам других блоков, то состояние Bad блоков-источников передается блоку Program. Исключение составляет обращение к атрибуту **STATUS**: состояние Bad источника не влияет на блок Program при чтении этого атрибута. Это позволяет функциональному блоку Program корректно анализировать состояние источников, не наследуя их статус Bad.

6. Режим функционального блока

Обработка данных функциональным блоком производится в соответствии с заданным режимом. Схематичное изображение процесса обработки данных приведено на рисунке (Рисунок 2. Процесс обработки данных с учетом режима функционального блока).

Типы режимов функциональных блоков:

OFF (0) - "Выключен" ;

MANUAL (1) - "Ручной";

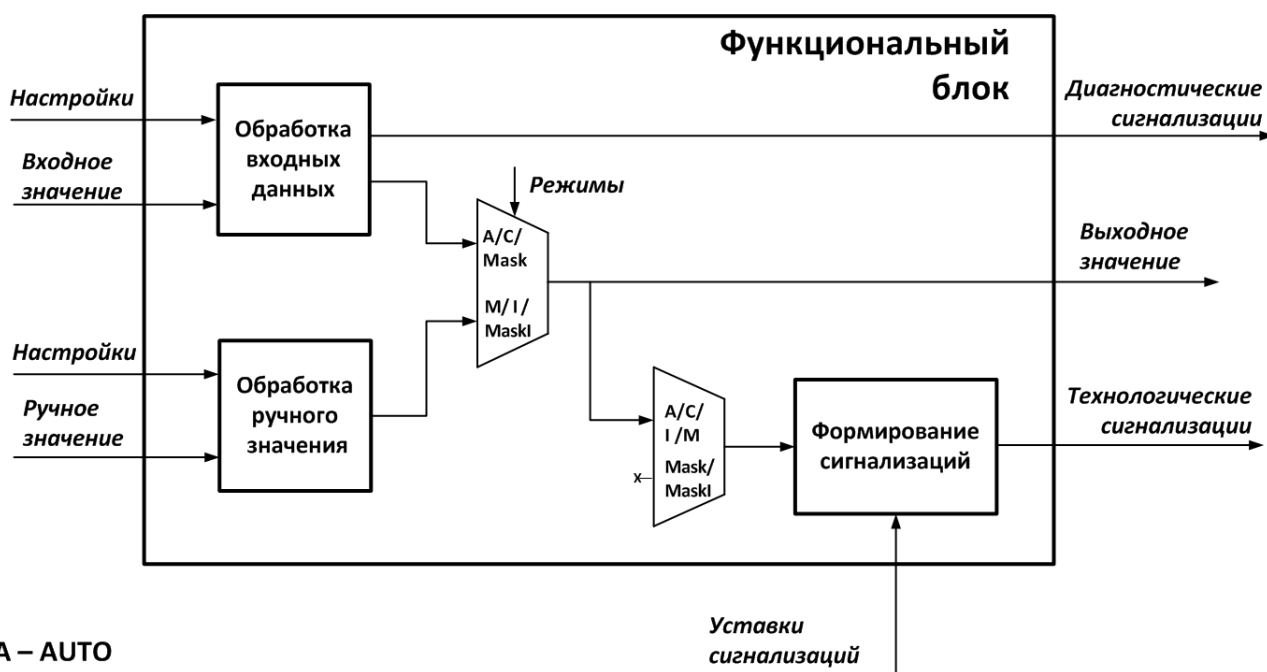
AUTO (2) - "Автоматический";

CASCADE (3) - "Каскадный";

MASKING (4) - "Маскирование";

IMITATION (5) - "Имитация";

MASK_IMITATION (6) - "Имитация с маскированием".



A – AUTO
 Mask – MASKING
 I – IMITATION
 MaskI – MASK IMITATION
 C – CASCADE
 M – MANUAL

Рисунок 2. Процесс обработки данных с учетом режима функционального блока

6.1. Режим "Выключен"

В данном режиме функциональный блок остается в том состоянии, которое он имел до момента выключения.

В атрибуты разрешена запись, но значения игнорируются.

Алгоритм функционального блока не выполняется.

Если у какого-либо функционального блока блок-источник находится в режиме "Выключен", то у него формируется статус "UnertainSourceStatus".

6.2. Режим "Автоматический"

В данном режиме считываются входные значения и производится их обработка.

Настройки обработки задаются вручную.

Результат обработки записывается в выходные атрибуты, и обновляется статус функционального блока.

Обновляются диагностические и технологические атрибуты.

6.3. Режим "Маскирование"

В данном режиме порядок выполнения алгоритма функционального блока совпадает с режимом "Автоматический". При этом часть алгоритма, отвечающая за формирование сигнализаций, не выполняется.

Статус блока всегда "ОК".

6.4. Режим "Имитация"

Данный режим позволяет передавать значение выходным атрибутам в обход алгоритма.

Статус блока всегда "ОК", сигнализационный флаг об ошибке Bad не выставляется, диагностические сигнализации подавляются, технологические сигнализации не обновляются.

6.5. Режим "Имитация с маскированием"

В данном режиме порядок выполнения алгоритма функционального блока совпадает с режимом "Имитация". При этом часть алгоритма, отвечающая за формирование сигнализаций, не выполняется.

И диагностические, и технологические сигнализации подавляются.

6.6. Режим "Каскадный"

Данный режим по своим функциям совпадает с режимом "Автоматический".

Отличие его в том, что настройки для обработки входных значений функциональный блок получает не от оператора, а от другого функционального блока.

Примером применения каскадного режима может служить ПИД регулятор.

6.7. Режим "Ручной"

В отличие от режима "Имитация", в режиме "Ручной" входные значения алгоритмов задаются посредством конфигурационных атрибутов.

7. Рекомендации по невалидным ссылкам

- При разработке и эксплуатации NaftaProcess рекомендуется максимально сокращать количество ссылок на блоке в плохом качестве.
- Переводить блоки модулей, а также другие блоки-источники сигналов в режим маскирование, имитации.
- Если количество невалидных ссылок в проекте больше 100, то возможны проблемы с работой блоков во время обновления их конфигурации.
- Количество обрабатываемых ссылок за скан равно 100. Если ссылка невалидна, она будет снова обрабатываться в следующем скане.