



Система конфигурирования реле интеллектуальных  
серии Элсима "BitLogic"

## РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Страниц 81

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

август 2019

Литера



## СОДЕРЖАНИЕ

<b>СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ</b> .....	<b>4</b>
<b>1 ВВЕДЕНИЕ</b> .....	<b>5</b>
1.1 ОБЛАСТЬ ПРИМЕНЕНИЯ .....	5
1.2 КРАТКОЕ ОПИСАНИЕ СИСТЕМЫ BitLOGIC .....	5
1.3 УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЯ.....	5
1.4 ПЕРЕЧЕНЬ ЭКСПЛУАТАЦИОННОЙ ДОКУМЕНТАЦИИ .....	5
<b>2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ</b> .....	<b>6</b>
2.1 НАЗНАЧЕНИЕ СИСТЕМЫ BitLOGIC.....	6
2.2 УСЛОВИЯ ПРИМЕНЕНИЯ СИСТЕМЫ BitLOGIC.....	6
<b>3 ПОДГОТОВКА К РАБОТЕ</b> .....	<b>7</b>
3.1 ПОДКЛЮЧЕНИЕ КОНТРОЛЛЕРА.....	7
3.2 ЗАДАНИЕ РЕЖИМА РАБОТЫ КОНТРОЛЛЕРА .....	7
3.3 СОСТАВ И СОДЕРЖАНИЕ ДИСТРИБУТИВНОГО НОСИТЕЛЯ ДАННЫХ .....	9
3.4 УСТАНОВКА СИСТЕМЫ BitLOGIC .....	9
<b>4 ИНТЕРФЕЙС СИСТЕМЫ BitLOGIC</b> .....	<b>12</b>
4.1 ГЛАВНОЕ ОКНО.....	12
4.2 СОЗДАНИЕ ПРОЕКТА .....	13
4.3 РАБОТА С ОКНАМИ .....	15
4.3.1 <i>Главное окно</i> .....	15
4.3.2 <i>Библиотека (Library)</i> .....	16
4.3.3 <i>Свойства (Properties)</i> .....	17
4.3.4 <i>Стек действий (Undo Stack)</i> .....	20
4.3.5 <i>Консоль (Console)</i> .....	20
4.3.6 <i>Настройки (Settings)</i> .....	20
4.3.7 <i>Настройка сетевых параметров контроллера</i> .....	22
4.3.8 <i>Обновление прошивки</i> .....	22
4.3.9 <i>О программе BitLogic</i> .....	26
4.4 СОСТАВЛЕНИЕ И РЕДАКТИРОВАНИЕ СХЕМ ИЗ ФБ.....	26
4.4.1 <i>Создание экземпляра блока</i> .....	26
4.4.2 <i>Создание соединения ФБ</i> .....	26
4.4.3 <i>Перенос сегмента линии</i> .....	28
<b>5 ПРИНЦИПЫ СОЗДАНИЯ ПРОГРАММ</b> .....	<b>30</b>
5.1 ПЕРЕД ПЕРВЫМ ПОДКЛЮЧЕНИЕМ К КОНТРОЛЛЕРУ .....	30
5.2 СОЕДИНЕНИЕ С КОНТРОЛЛЕРОМ .....	30
5.3 ДОБАВЛЕНИЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ.....	30
5.4 ЗАГРУЗКА ПРОЕКТА .....	30
5.5 ЗАПУСК ПРОЕКТА .....	30
5.6 ИНДИКАЦИЯ КОНТРОЛЛЕРА .....	31
<b>6 ПРИМЕР ПРОГРАММЫ</b> .....	<b>32</b>
<b>7 ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ</b> .....	<b>33</b>
<b>8 ГОРЯЧИЕ КЛАВИШИ</b> .....	<b>75</b>
<b>9 СООБЩЕНИЯ ОБ ОШИБКАХ И ПРЕДУПРЕЖДЕНИЯХ</b> .....	<b>76</b>

## Список терминов и сокращений

BitLogic	– Система конфигурирования реле интеллектуальных серии Элсима
HR	– Holding Register
IR	– Input Register
MB	– Modbus
TCP	– Transmission Control Protocol
PWM	– Pulse-Width Modulation (ШИМ)
Контроллер	– Интеллектуальное реле Элсима-RL-DA02, Элсима-RL-D02, Элсима-RL-D01, Элсима-RL-A01
ЛКМ	– Левая клавиша мыши
ПК	– Персональный компьютер
ПКМ	– Правая клавиша мыши
ПЛК	– Программируемый логический контроллер
ПО	– Программное обеспечение
ФБ	– Функциональный блок (в соответствии с IEC 61131-3)
ШИМ	– Широтно-импульсная модуляция
ЭНП	– Энергонезависимая память

# **1 Введение**

## **1.1 Область применения**

Программа «Система конфигурирования контроллера BitLogic» (далее, система BitLogic) предоставляет пользователю возможность быстрой разработки программного обеспечения, реализующего заданный алгоритм работы, и внедрения на основе интеллектуального реле из линейки Элсима-RL-DA02, Элсима-RL-D02, Элсима-RL-D01, Элсима-RL-A01 системы автоматизации процессов, востребованных в условиях технологического производства, бытового применения или коммерческого использования. Приоритетным направлением использования Системы BitLogic является построение на основе интеллектуальных реле системы «умный дом».

## **1.2 Краткое описание системы BitLogic**

Способом разработки программного обеспечения является программирование функциональными блоками (графическое программирование). При программировании используются наборы библиотечных блоков (в том числе, самостоятельно разработанных пользователем). Блоком является подпрограмма, созданная на базе процедурного программирования. Каждый блок имеет входы и выходы для данных. Пользователь выбирает необходимые блоки и соединяет входы и выходы блоков в соответствии с решаемой задачей.

## **1.3 Уровень подготовки пользователя**

Для работы с системой BitLogic пользователю необходимо иметь навыки работы в операционной системе семейства Windows/Linux.

## **1.4 Перечень эксплуатационной документации**

Перед применением системы BitLogic пользователю необходимо ознакомиться с руководством пользователя «Система конфигурирования контроллера BitLogic», а также с руководством по эксплуатации «Интеллектуальное реле Элсима-RL-DA02» (в зависимости от используемого модуля – DA02, D02, D01 или A01).

## 2 Назначение и условия применения

### 2.1 Назначение системы BitLogic

Программа «Система конфигурирования контроллера BitLogic» предназначена для разработки программного обеспечения, реализующего заданный алгоритм работы на основе интеллектуального реле Элсима-RL-DA02, Элсима-RL-D02, Элсима-RL-D01 или Элсима-RL-A01, представляющих собой отдельные программируемые логические контроллеры (ПЛК) и построенных на основе микроконтроллера STM32F407.

Система BitLogic предназначена для разработки технических решений задач автоматизации, построенных на основе интеллектуального реле Элсима-RL-DA02, Элсима-RL-D02, Элсима-RL-D01 или Элсима-RL-A01.

### 2.2 Условия применения системы BitLogic

Программа устанавливается на компьютеры с операционными системами семейства Linux, Windows (Windows 7, Windows 8, Windows 10).

Программа имеет установочный файл (для систем с разрядностью x64 – BitLogic\_x64\_setup.exe, для систем с разрядностью x32 – BitLogic\_x86\_setup.exe).

В системе BitLogic имеется ряд ограничений:

- максимальное количество констант – 1000;
- максимальное количество переменных – 1000;
- максимальное количество ФБ в системе – 1100;
- максимальное количество входов в одном ФБ – 50;
- максимальное количество выходов в одном ФБ – 15;
- максимальное количество регистров Holding Register (HR) Modbus (MB) – 500;
- максимальное количество регистров Input Register (IR) MB – 500.

## 3 Подготовка к работе

### 3.1 Подключение контроллера

Контроллер требуется обеспечить напряжением питания 24 В (интеллектуальное реле DA02, D02, D01, A01 версии 24P). Для передачи данных контроллер необходимо подключить к сети Ethernet с использованием кабеля не ниже CAT UTP5.

Количество одновременно подключаемых к контроллеру МВ TCP клиентов не должно превышать 5 (пять).

### 3.2 Задание режима работы контроллера

Выбор режима работы контроллера задается с помощью переключателя SW (см. таблицу 1), расположенных на боковой стороне контроллера (см. рис. 1).

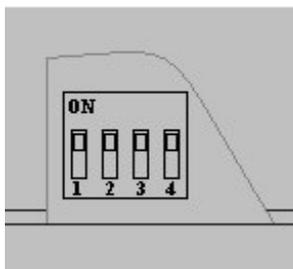


Рисунок 1. Переключатели SW.

Таблица 1. Установка режимов работы интеллектуального реле.

Режим работы контроллер	Состояние переключателя SW				Выполняемая функция
	DIP1	DIP2	DIP3	DIP4	
<b>Рабочий режим</b> (переход к загрузке основной программы при включении контроллера)	0	0	1	0	Контроллер находится в режиме «Стоп»  При включении питания контроллер загружается и переходит в состояние «Стоп» (даже при наличии сохраненной задачи во флэш-памяти).
	0	0	1	1	Контроллер переходит в режим выполнения задачи (при наличии задачи во флэш)  При включении питания контроллер загружается и при наличии сохраненной задачи во флэш-памяти начинается ее выполнение.
<b>Сервисный режим</b> (загрузка задачи пользователя, установка IP-адреса, обновление прошивки)	1	0	0	0	Контроллер переходит в режим bootloader при его включении  При выборе этого режима у пользователя появляется возможность обновить прошивку (системное ПО) контроллера.
	1	0	1	0	Работа контроллера с пользовательскими настройками (IP-адрес пользователя)  Работа с контроллером (в том числе, и загрузка проекта из системы BitLogic)

					осуществляется с сетевыми параметрами, установленными пользователем.
	1	0	1	1	Сброс настроек контроллера до заводских (IP-адрес: 10.14.0.254) Работа с контроллером (в том числе, и загрузка проекта из системы BitLogic) осуществляется с сетевыми параметрами, установленными заводом-изготовителем.
	1	1	1	1	Контроллер переходит в режим калибровки Данное положение DIP-переключателей позволяет осуществить калибровку контроллера (применимо только для контроллеров Elsyma-RL-DA02 и Elsyma-RL-A01).

*Примечание:*

1. Остальные положения DIP-переключателей, отличные от представленных в таблице, являются зарезервированными.
2. Для использования контроллера в том режиме, который был установлен с помощью DIP-переключателей, контроллер необходимо перезагрузить по питанию. Например, для установки пользовательского IP-адреса (при условии, что изначально IP-адрес не был известен пользователю) DIP-переключатели необходимо перевести в положение 1011. Выключить-включить питание контроллера. В системе BitLogic в окне **Настройки (Settings)** во вкладке **Подключение (Connection)** установить IP-адрес 10.14.0.254, подключиться к контроллеру. Далее в окне **Настройки устройства (Device Settings)** заменить IP-адрес на пользовательский и нажать кнопку **Записать (Write)**. Затем установить DIP-переключатели в положение 1010 и перезагрузить контроллер по питанию. Теперь подключение к контроллеру осуществляется по установленному пользователем IP-адресу.

### 3.3 Состав и содержание дистрибутивного носителя данных

Программа «Система конфигурирования контроллера BitLogic» может находиться в составе SCADA-системы. Программа состоит из исполняемого файла bitlogic.exe, папок с настройками и библиотек DLL.

### 3.4 Установка системы BitLogic

Установка системы BitLogic не отличается от стандартного алгоритма установки программного обеспечения:

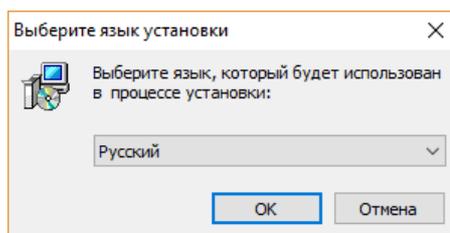


Рисунок 2. Выбор языка установки.

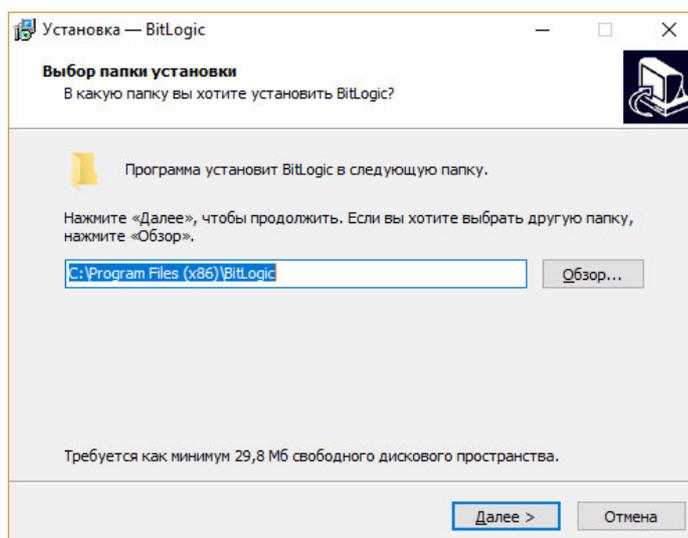


Рисунок 3. Выбор директории установки.

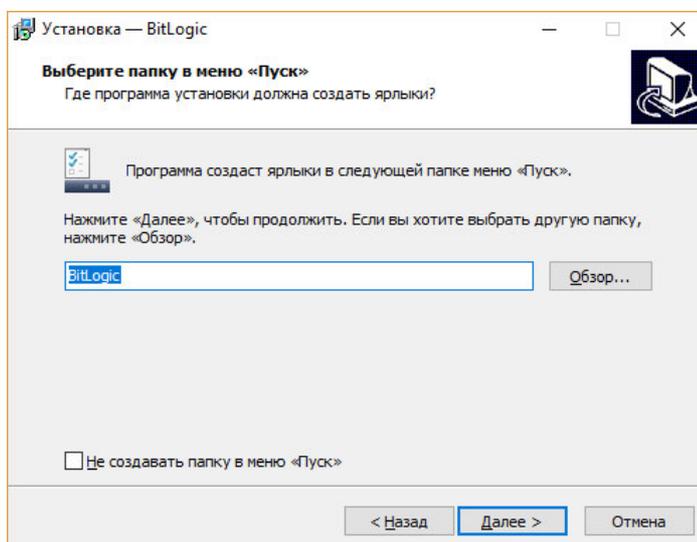


Рисунок 4. Создание ярлыков программы.

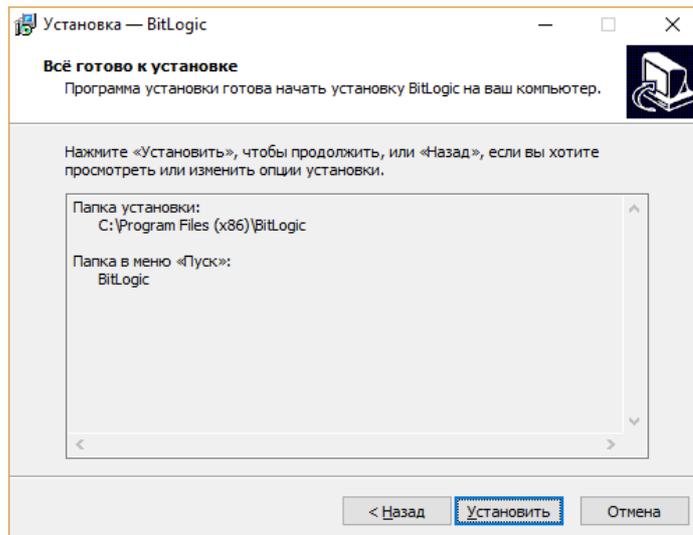


Рисунок 5. Проверка параметров установки.

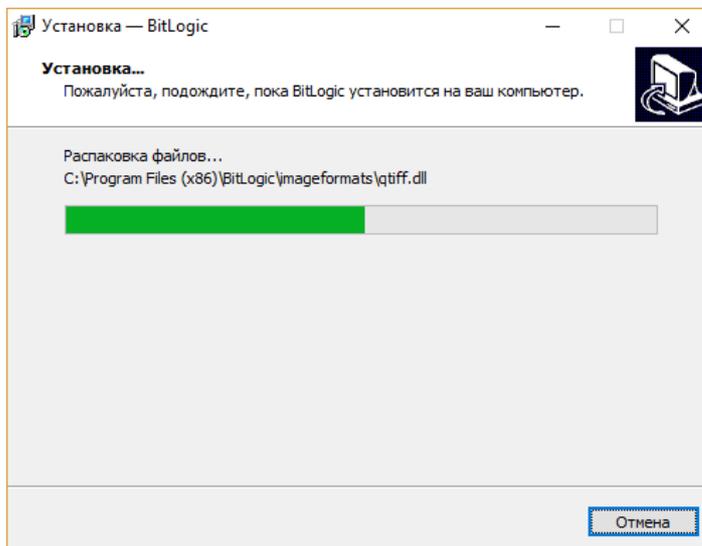


Рисунок 6. Установка программы.

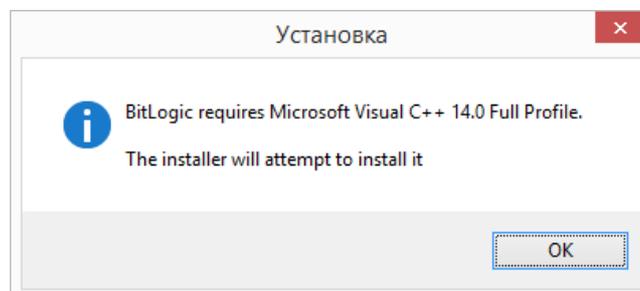


Рисунок 7. Информирование пользователя о необходимости установки пакета Microsoft Visual C++ (в случае его отсутствия или старой версии).

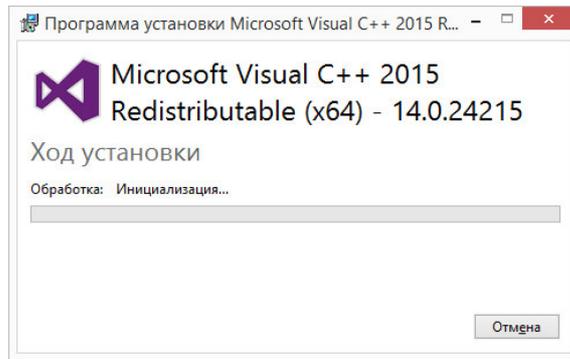


Рисунок 8. Процесс установки пакета Microsoft Visual C++.

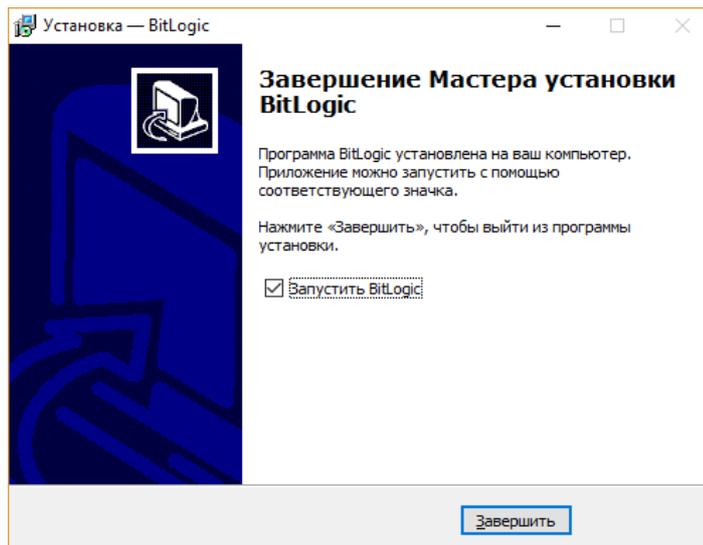


Рисунок 9. Завершение установки.

## 4 Интерфейс системы BitLogic

В данном разделе приведены рекомендации по освоению и эксплуатации системы BitLogic.

Данный редактор конфигурационных интерфейсов может работать в составе SCADA-системы для управления контроллерами.

### 4.1 Главное окно

Программа запускается вызовом исполняемого файла непосредственно или по нажатию соответствующего ярлыка способом, предусмотренным стандартно в операционной системе.

На рис. 10 представлен вид главного окна (для лучшего изучения структуры главного окна в системе BitLogic загружен проект, на поле проектов представлена часть проекта).

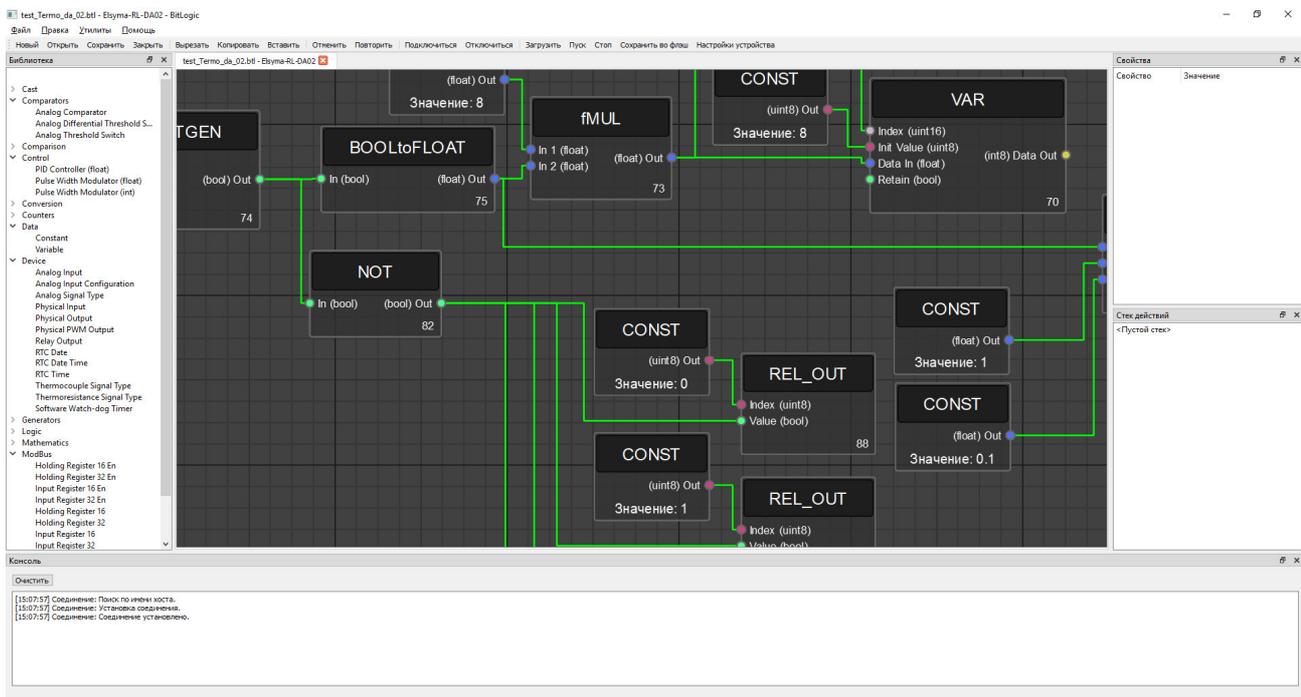


Рисунок 10. Окно системы BitLogic.

На верхней панели размещается главное меню приложения с кнопками: **Файл (File)**, **Правка (Edit)**, **Утилиты (Utilities)**, **Помощь (Help)** имеющиеся в стандартном наборе главного меню любой среды разработки. При нажатии на эти кнопки выпадает меню с набором кнопок, часть из которых для удобства продублирована на **Панели инструментов (Toolbar)**, расположенного ниже главного меню.

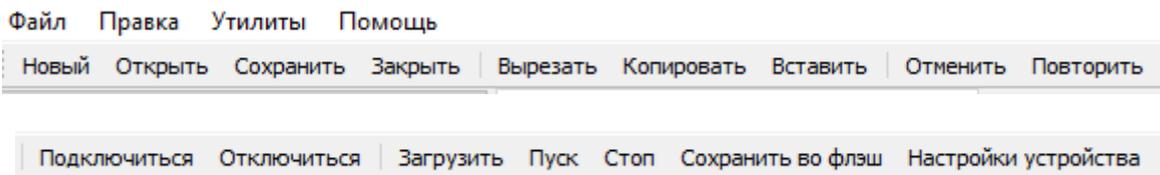


Рисунок 11. Главное меню и Панель инструментов системы BitLogic.

Кнопки **Панели инструментов (Toolbar)** имеют следующее назначение:

**Новый (New)** – создать новый проект

**Открыть (Open)** – открыть существующий проект

**Сохранить (Save)** – сохранить изменения в проекте

**Закреть (Close)** – закрыть проект

**Вырезать (Cut)** – вырезать элементы проекта (ФБ, соединительные линии)

**Копировать (Copy)** – копировать элементы проекта (ФБ, соединительные линии)

**Вставить (Paste)** – вставить элементы проекта (ФБ, соединительные линии)

**Отменить (Undo)** – отменить последние действия

**Повторить (Redo)** – повторить последние действия

**Подключиться (Connect)** – установить соединение с контроллером

**Отключиться (Disconnect)** – разорвать соединение с контроллером

**Загрузить (Upload)** – проект в оперативную память контроллера

**Пуск (Run)** – запустить выполнение проекта

**Стоп (Stop)** – остановить выполнение проекта

**Сохранить во флэш (Save in Flash)** – сохранить проект во флэш-памяти контроллера

**Настройки устройства (Device Settings)** – установить параметры соединения контроллера (IP-адрес, MAC-адрес, установка времени и др.)

Под **Панелью инструментов (Toolbar)** располагается окно **Библиотеки (Library)** функциональных блоков, самое большое окно приложения – поле проекта, окно **Свойства (Properties)**. **Библиотека (Library)** хранит набор классифицированных по категориям функциональных блоков (ФБ), необходимых для разработки программы конфигурирования контроллера. Поле проектов предназначено для непосредственного составления схемы решения задачи автоматизации из ФБ, соединенных друг с другом. В окне **Свойства (Properties)** можно задать свойства выбранного ФБ. Под окном **Свойства (Properties)** находится окно **Стека действий (Undo Stack)**, предназначенное для удобной отмены или повторения как одного, так и нескольких за один раз действий. В самом низу приложения расположена **Консоль (Console)**, информирующая пользователя о выполненных действиях, ошибках и предупреждениях. Несмотря на предустановленный порядок расположения окон приложения, для удобства их расположение можно менять, а также закрыть их.

## 4.2 Создание проекта

Для создания нового проекта необходимо нажать на кнопку **Новый (New)**. Запустится Мастер создания проекта и откроется окно создания нового проекта:

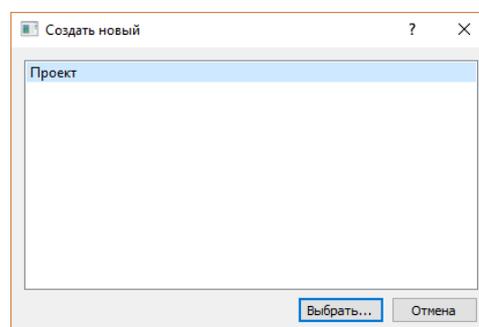


Рисунок 12. Окно создания нового проекта.

Далее необходимо выбрать тип проекта (в данном случае, **Проект (Project)**) и нажать кнопку **Выбрать... (Selected...)**.

Откроется окно с предложением указать имя проекта и директорию его расположения. Для выбора директории расположения проекта используйте кнопку **Обзор... (Browse...)**. После заполнения всех полей кнопка **Далее (Next)** станет активной.

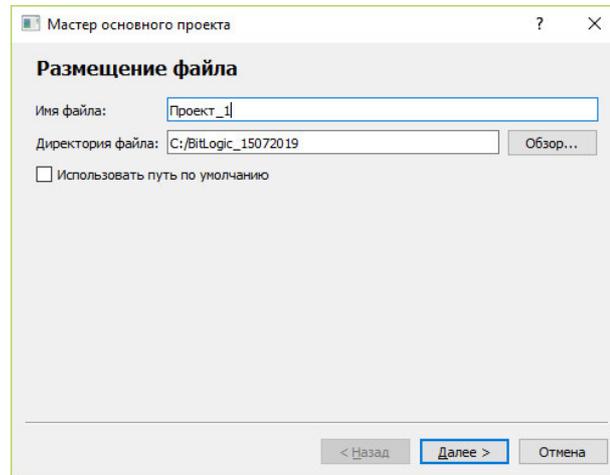


Рисунок 13. Окно уставки имени и директории расположения проекта.

Затем появится окно выбора модели контроллерного устройства, перечень которых приведен в раскрывающемся списке:

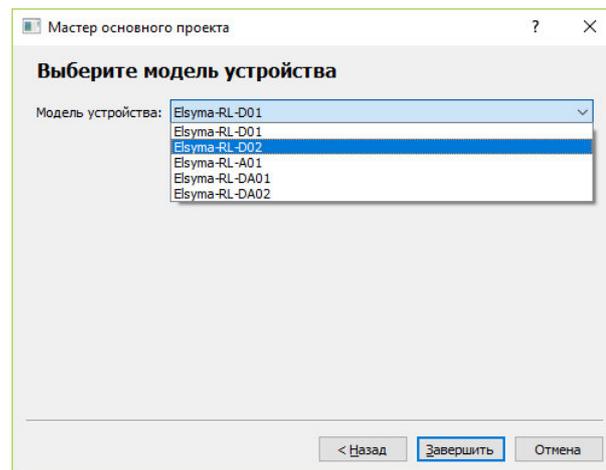


Рисунок 14. Окно выбора типа контроллера.

Наконец, после нажатия кнопки **Завершить (Finish)** откроется пустой проект:

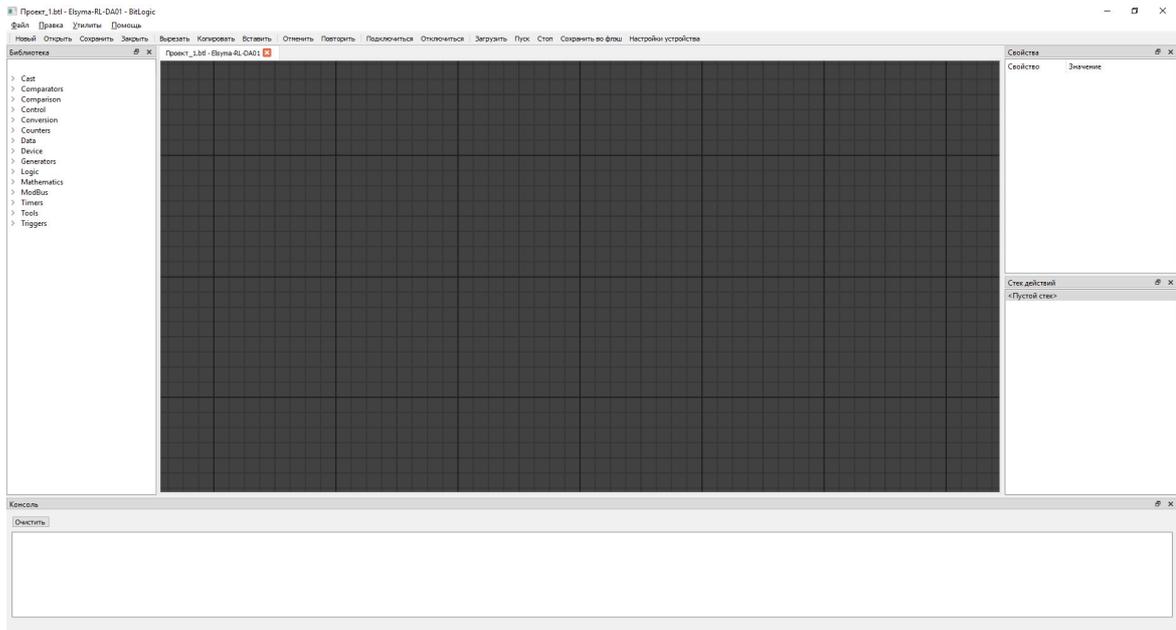


Рисунок 15. Окно пустого проекта в системе BitLogic.

## 4.3 Работа с окнами

### 4.3.1 Главное окно

В приложении имеется возможность одновременной работы с несколькими проектами. При этом название проекта и тип выбранного устройства отображаются на вкладках окна поля проектов, с помощью которых можно переключаться между проектами.

Если название проекта отмечено звездочкой, то проект не был сохранен после последних выполненных в нем действий.

При необходимости блоки и соединительные линии можно удалить с поля проекта. Для этого, кликом правой клавиши мыши (ПКМ) нужно выбрать необходимый блок или соединительную линию и нажать **Удалить (Delete)** либо нажать на клавиатуре кнопку **Delete**. Для одновременного удаления нескольких блоков и соединительных линий нужно выделить мышью требуемый участок схемы, навести мышь на этот участок схемы и кликом ее ПКМ вызвать команду **Удалить (Delete)** либо нажать кнопку **Delete**.

При одновременно нажатой кнопке **Ctrl** и клике левой кнопки мыши (ЛКМ) можно также выбрать необходимые блоки. При этом все соединительные линии, входящие в блок и исходящие из него, также будут выделены. При последующем удалении блоков, линии также будут удалены.

*Следует отметить*, при удалении ФБ все связанные с ним линии удаляются. Однако, для удаления только соединительных линий не требуется удалять блоки, достаточно удалить только ненужные линии.

При прокрутке колеса мыши можно изменять масштаб поля проекта. Перемещение поля проекта осуществляется при одновременно нажатой кнопке **Alt** и ЛКМ, а также при нажатом колесе мыши.

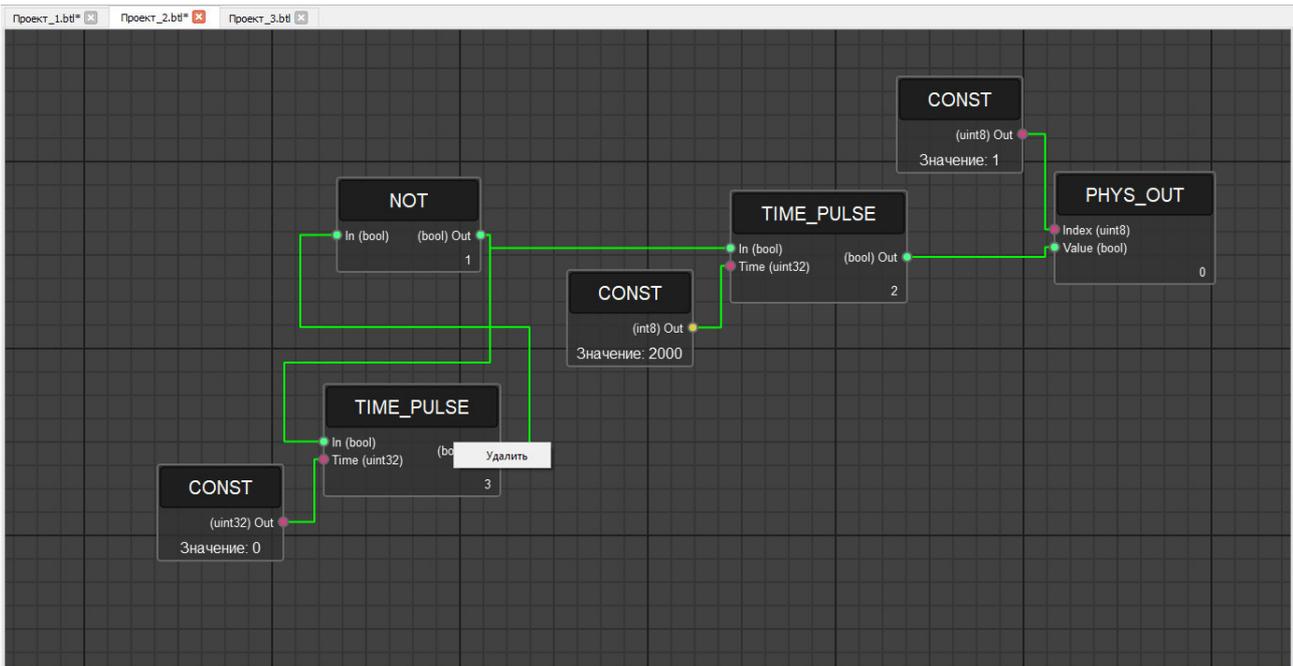


Рисунок 16. Работа с несколькими одновременно открытыми проектами в системе BitLogic.

### 4.3.2 Библиотека (Library)

Для добавления ФБ на поле проекта необходимо выбрать нужную категорию ФБ, нажать ЛКМ на ФБ и перетащить его (*Drag-and-Drop*) на поле проекта. После чего на поле проекта отобразится выбранный ФБ.

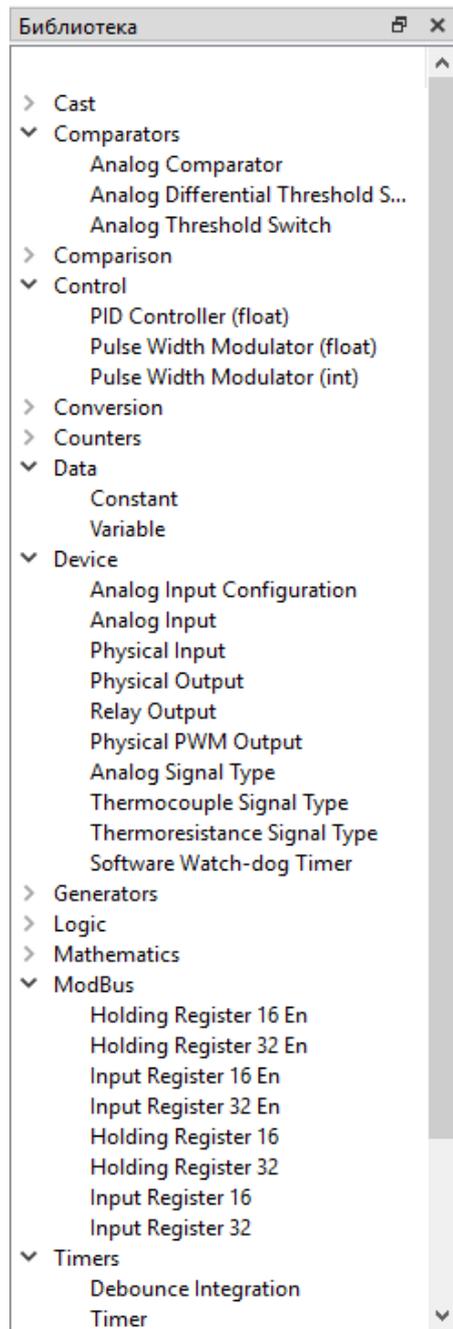


Рисунок 17. Пример библиотеки ФБ.

### 4.3.3 Свойства (Properties)

Если установленный на поле проекта ФБ активен (для этого нужно щелкнуть по нему ЛКМ), информация об этом блоке отображается в окне **Свойства (Properties)** (см. рис. 18): название блока, количество входов/выходов, тип данных для каждого из входов/выходов, порядок вызова ФБ. При этом для некоторых ФБ группы **Logic** и группы **Mathematics** предусмотрена возможность увеличения количества входов для проведения одновременной операции, например, операции логического AND или OR, арифметического Addition (Суммирование), Multiplication (Умножение) или некоторых других операций для нескольких сигналов (больше двух), одновременно поступающих на входы ФБ. В окне **Свойства (Properties)** при двойном клике ЛКМ на поле со значением количества входов ФБ активируется кнопка со стрелкой вверх, позволяющая увеличить количество входов ФБ. При количестве входов со значением большим двух активируется также кнопка со стрелкой вниз, нажимая которую ЛКМ количество входов уменьшится. Пример ФБ с увеличенным количеством входов приведен на рис. 19.

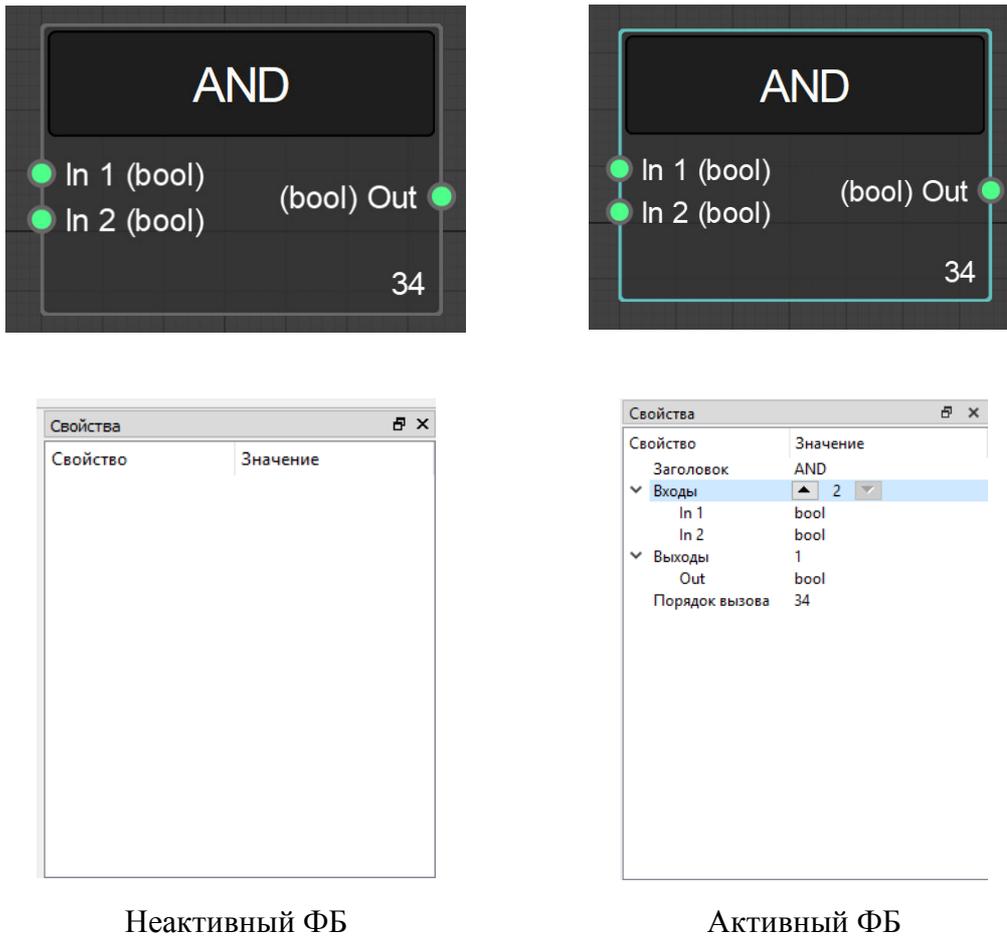


Рисунок 18. Пример неактивного (слева) и активного ФБ (справа) с отображением информации о нем в окне **Свойства (Properties)**.

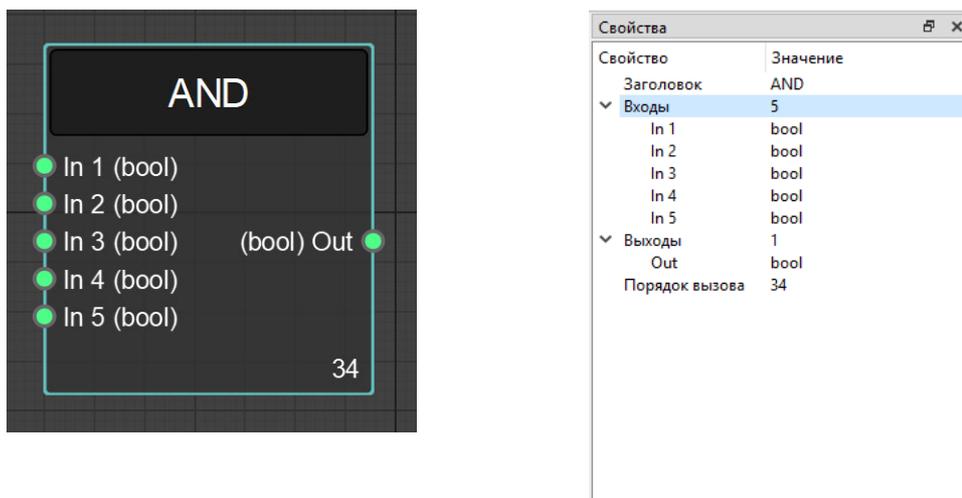


Рисунок 19. Пример ФБ с увеличенным количеством входов.

Для согласования типов данных между блоком **Constant** (группа **Data**) и каким-либо другим блоком, на вход которого поступают данные с выхода блока **Constant**, в окне **Свойства (Properties)** можно также задать (выбрать из списка) для выхода блока **Constant** нужный тип данных.

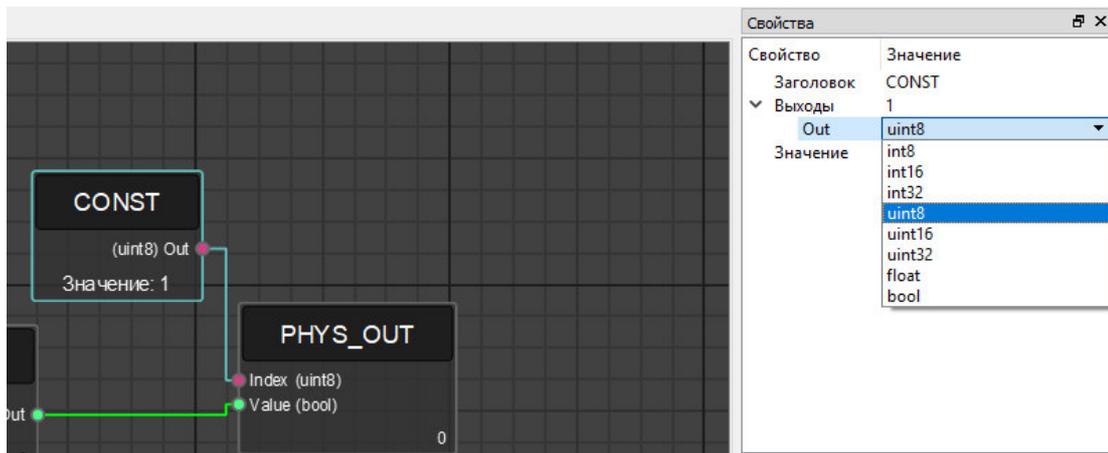


Рисунок 20. Согласование типов данных выходного и входного портов двух ФБ.

Для трех ФБ группы **Device: Analog Signal Type, Thermocouple Signal Type** и **Thermoresistance Signal Type**, можно выбрать необходимый тип сигнала, который будет считываться на аналоговом входе контроллера.

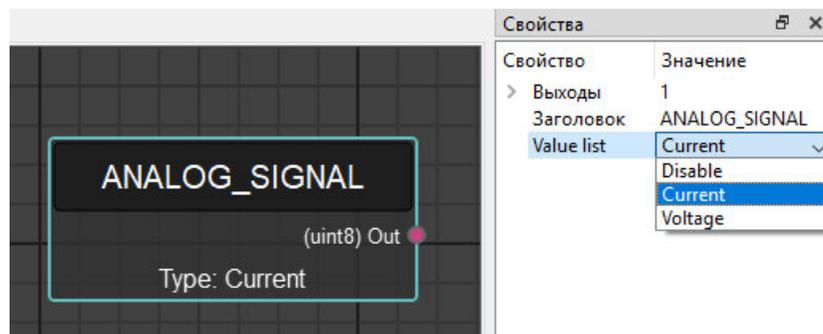


Рисунок 21. ФБ выбора типа входного аналогового сигнала.

Для ФБ следующих групп: **Control, Counters, Generators, Modbus, Timers, Triggers** и двух блоков **RTC\_DATE** и **RTC\_TIME** (группы **Device**), большинство выходов по умолчанию являются скрытыми. Однако, пользователь для решения своих задач может их задействовать и получать с них дополнительные данные. Для этого, нужно выбрать необходимый ФБ из вышеперечисленных групп, установить его на поле проекта, кликнуть по блоку ЛКМ и в окне **Свойства (Properties)** сделать требуемый выход выбранного блока видимым – поставить ЛКМ галочку. При этом указанный выход блока отобразится на поле проекта и станет доступным для подключения.

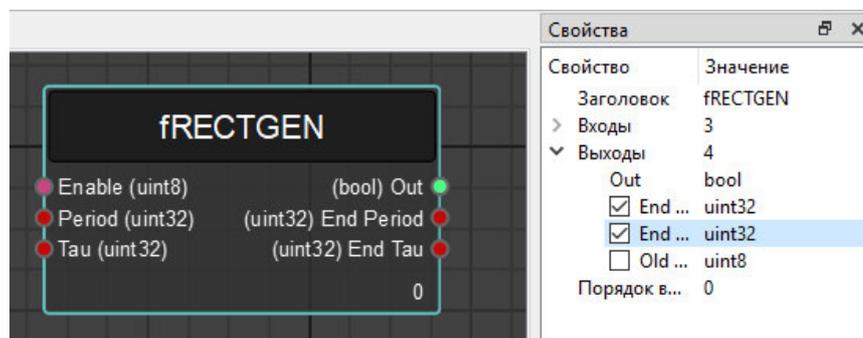


Рисунок 22. Пример установки скрытых выходов ФБ.

### 4.3.4 Стек действий (Undo Stack)

Для того, чтобы отменить/повторить одно или несколько действий, выполненных на поле проекта, необходимо кликом ЛКМ выбрать ту строку в окне **Стек действий (Undo Stack)**, с которой должно быть отменено/повторено требуемое количество действий. Точкой отсчета количества совершенных действий является строка **Пустой стек (Empty stack)**.

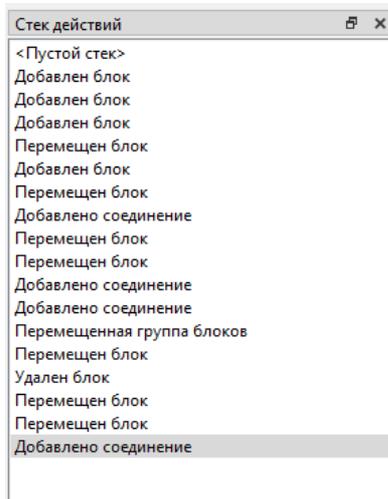


Рисунок 23. Пример окна стека действий.

### 4.3.5 Консоль (Console)

В консоль выводится информация взаимодействия системы и контроллера (например, сообщения о состоянии подключения к контроллеру, о сборке сконфигурированной программы в программный код, о загрузке проекта в оперативную или флэш-память контроллера, о запуске или остановке проекта, и т.д.), отладочная информация, ошибки проектирования программы, предупреждения.

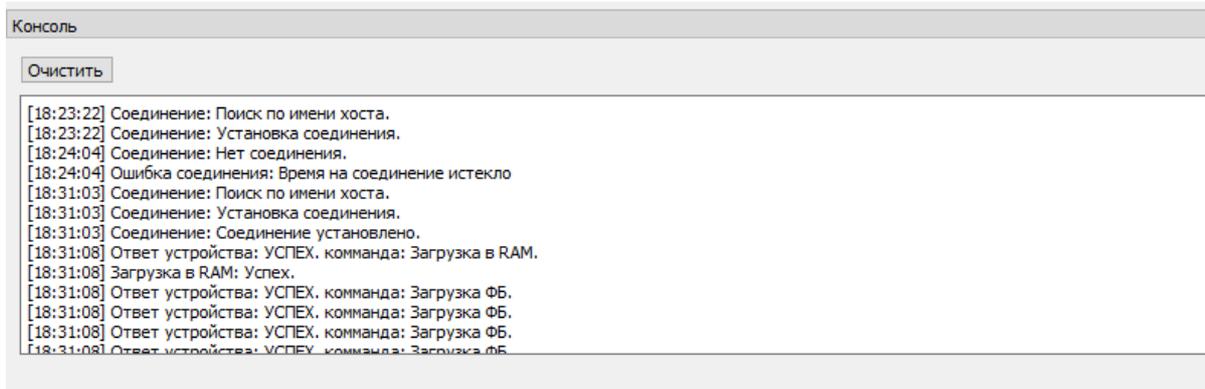


Рисунок 24. Пример консольного окна.

### 4.3.6 Настройки (Settings)

Для изменения настроек системы нужно открыть окно настроек **Файл→Настройки...** (**File→Settings...**). Окно настроек представляет из себя три вкладки: **Общие (General)**, **Подключение (Connection)**, **Вид (View)**.

На вкладке **Общие (General)** имеется возможность задать директорию расположения проектов и выбрать язык пользовательского интерфейса.

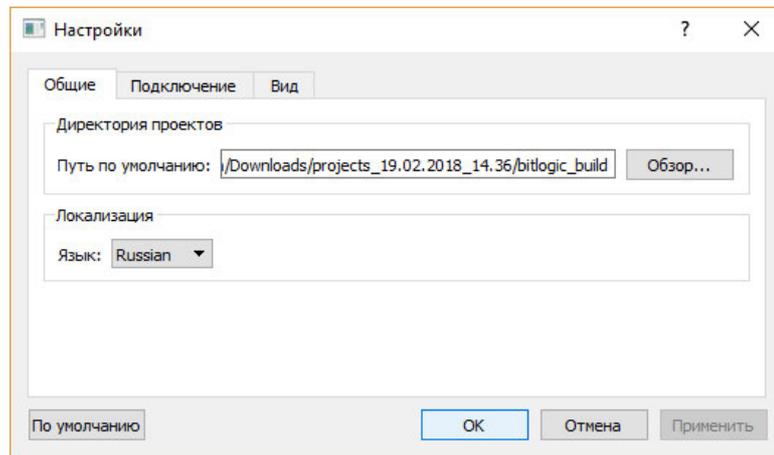


Рисунок 25. Окно настроек системы (вкладка Общие).

На вкладке **Подключение (Connection)** задаются параметры соединения: IP-адрес и порт подключения контроллера.

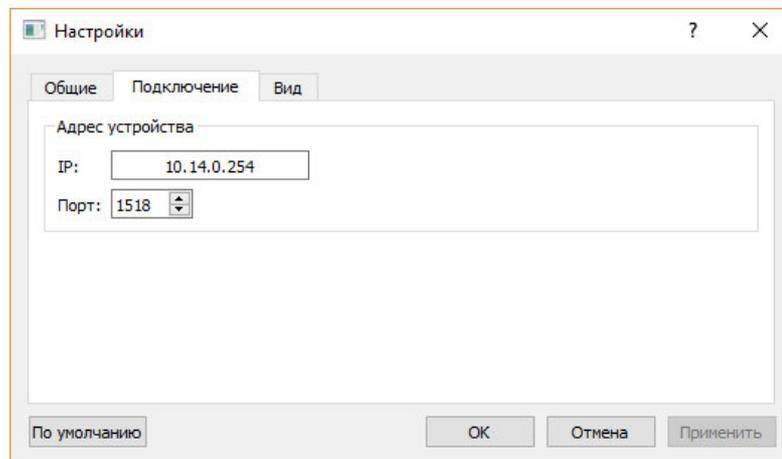


Рисунок 26. Окно настроек системы (вкладка Подключение).

Вкладка **Вид (View)** позволяет пользователю управлять видимостью таких окон приложения, как **Панель инструментов (Toolbar)**, **Библиотека (Library)**, **Свойства (Properties)**, **Стек действий (Undo Stack)** и **Консоль (Console)**, а также изменять **Тему (Theme)** сцены проекта.

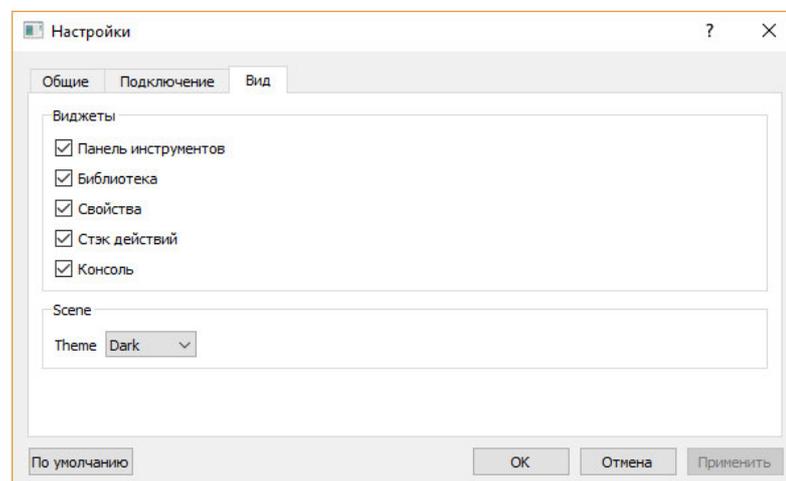


Рисунок 27. Окно настроек системы (вкладка Вид).

С помощью кнопки **По умолчанию (Default)** можно вернуть настройки системы, заданных по умолчанию.

### 4.3.7 Настройка сетевых параметров контроллера

Задать/просмотреть сетевые параметры контроллера можно с помощью кнопки **Настройки устройства (Device settings)**. Установка галочки напротив **Использовать по умолчанию (Use default)** возвращает настройки контроллера к первоначальным (маска 255.255.0.0, IP-адрес 10.14.0.254, шлюз 10.14.0.1, порт 1518, TCP-таймаут 300 сек). Кроме того, на этой же вкладке можно задать дату и время (или синхронизировать с датой и временем ПК), с которыми будет работать контроллер. В этом окне также приводится информация о версии прошивки контроллера, о типе подключенного устройства, а также о статусе калибровки устройства.

При нажатии на кнопку **Прочитать (Read)** выводятся MAC-адрес и сетевые параметры, сохраненные в памяти контроллера. Для сохранения в памяти контроллера установленных пользовательских сетевых параметров необходимо нажать кнопку **Записать (Write)**.

Установка MAC-адреса разрешена только для разработчиков контроллера.

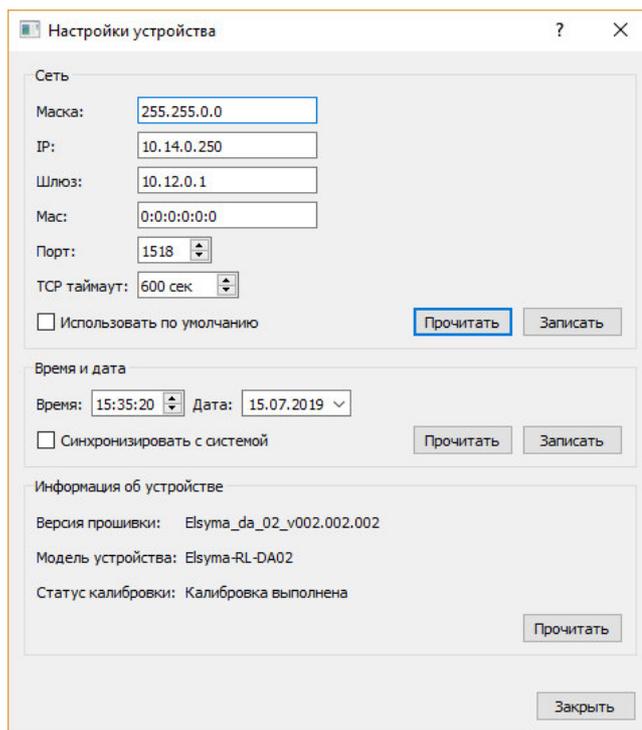


Рисунок 28. Окно настроек устройства.

### 4.3.8 Обновление прошивки

В системе BitLogic имеется возможность обновления прошивки контроллера по сети Ethernet. Для этого необходимо на вкладке **Утилиты (Utilities)** выбрать **Обновить прошивку (Firmware updater)**, откроется окно мастера обновления прошивки (см. рис. 29). Для реализации указанной функции требуется установить DIP-переключатели в положение, которое соответствует работе контроллера в режиме bootloader (см. таблицу 1). Далее необходимо перезагрузить контроллер.

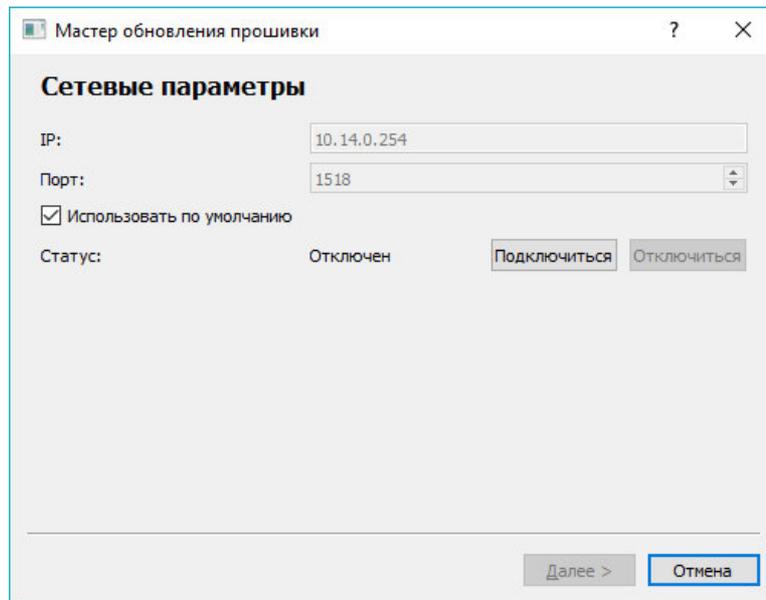


Рисунок 29. Окно мастера обновления прошивки: установка параметров сетевого подключения.

На рис. 29 приведено окно, в котором необходимо установить параметры специального сетевого соединения системы BitLogic с контроллером для дальнейшей загрузки прошивки на контроллер. По умолчанию заданы следующие сетевые параметры: IP-адрес 10.14.0.254, порт 1518. После задания сетевых параметров необходимо нажать кнопку **Подключиться (Connect)**. Результат подключения указывается в поле **Статус (Status)**. Окно успешного подключения приведено на рис. 30. При этом кнопка **Далее (Next)** становится активной и дает возможность перехода к следующему окну мастера обновления прошивки.

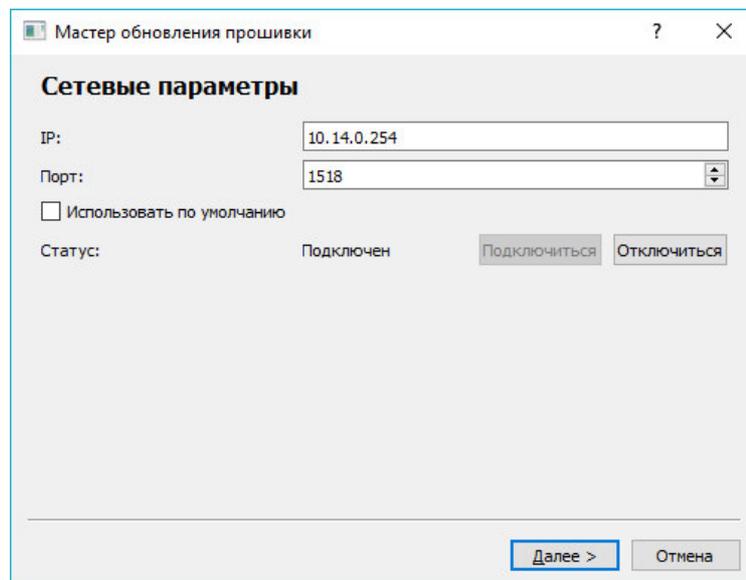


Рисунок 30. Окно мастера обновления прошивки: успешное подключение.

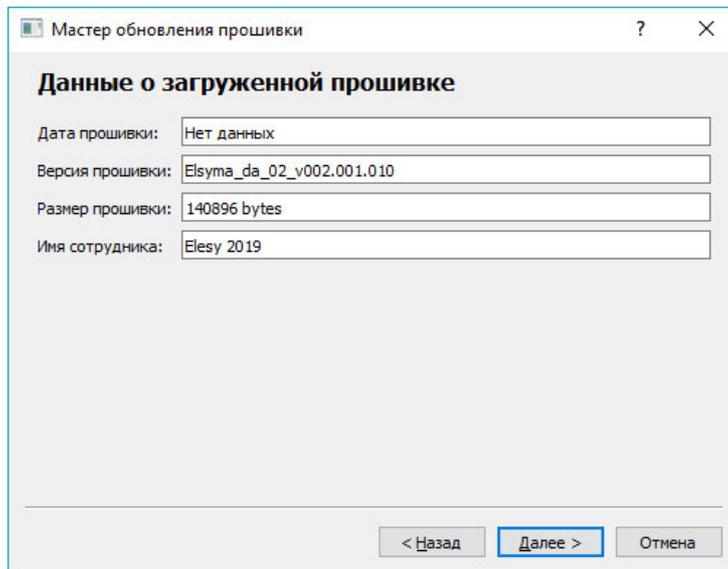


Рисунок 31. Окно мастера обновления прошивки: параметры предыдущей прошивки.

На рис. 31 можно увидеть следующую информацию о параметрах предыдущей прошивки контроллера: дата прошивки, версия прошивки, размер прошивки, имя сотрудника (пользователя), загрузившего прошивку. Ознакомившись с информацией об имеющейся прошивке, нужно перейти к следующему окну мастера обновления прошивки (см. рис. 32).

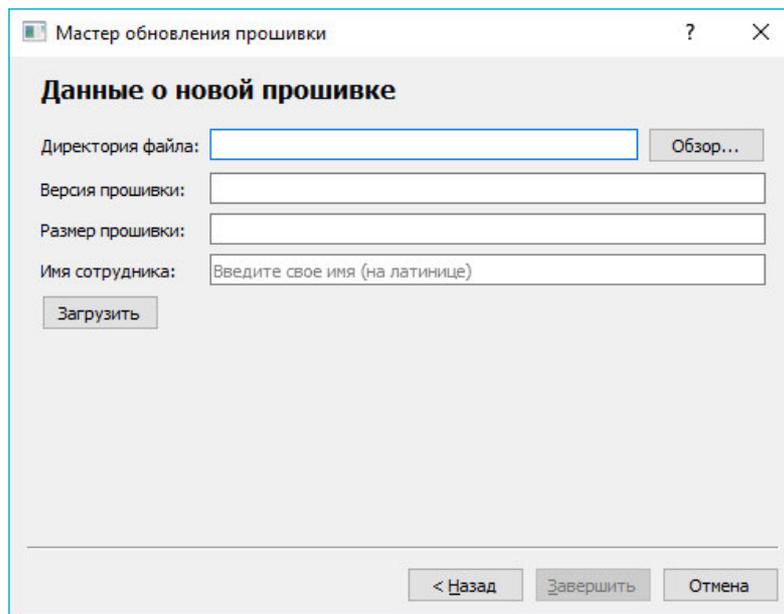


Рисунок 32. Окно мастера обновления прошивки: поля для ввода параметров новой прошивки.

В появившемся окне (см. рис. 32) необходимо выбрать путь до нужного файла прошивки, загрузив который отобразится информация о его параметрах (версия и размер прошивки). В этом окне также необходимо указать на латинице имя пользователя (сотрудника), обновляющего прошивку.

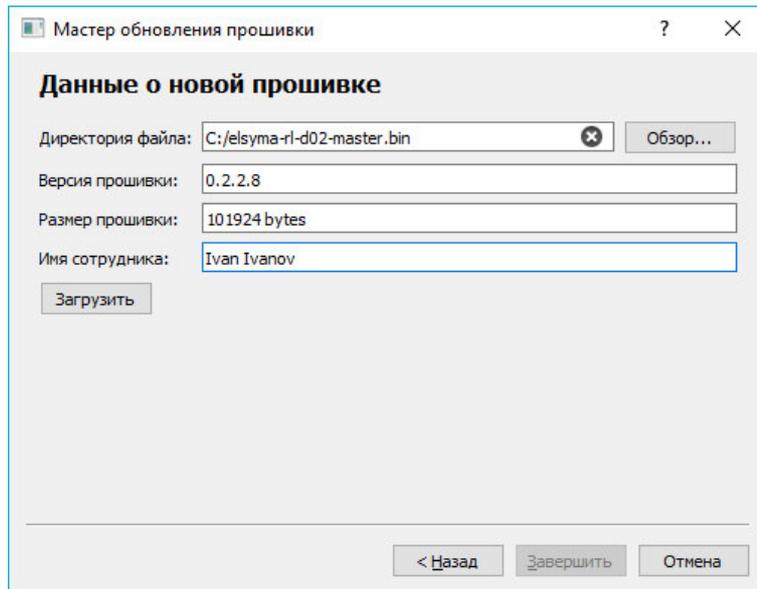


Рисунок 33. Окно мастера обновления прошивки: параметры новой прошивки.

Нажав на кнопку **Загрузить (Download)**, начнется процесс загрузки прошивки на контроллер.

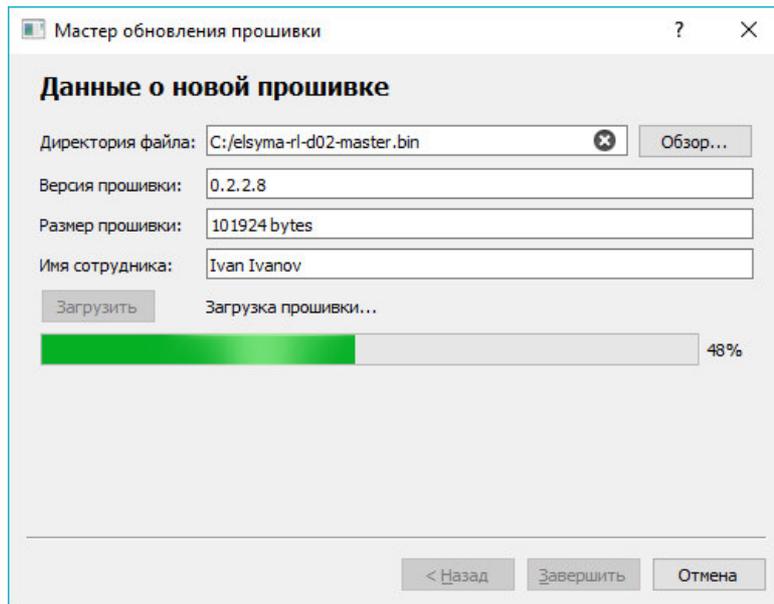


Рисунок 34. Окно мастера обновления прошивки: процесс загрузки прошивки на контроллер.

По окончании загрузки кнопка **Завершить (Finish)** станет активной. Нажатие на кнопку **Завершить (Finish)** заканчивает обновление прошивки.

### 4.3.9 О программе BitLogic

В окне о программе отображается информация о версии системы BitLogic и номере ревизии.

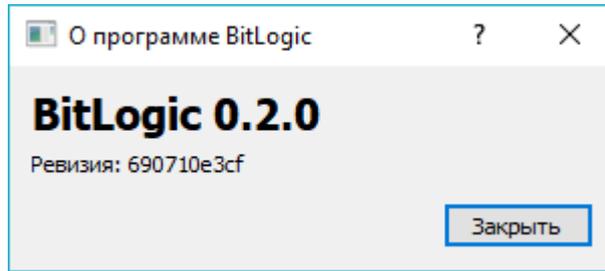


Рисунок 35. Окно о программе BitLogic.

## 4.4 Составление и редактирование схем из ФБ

### 4.4.1 Создание экземпляра блока

Создание экземпляра блока производится выносом блока из окна **Библиотеки (Library)** ФБ на поле проекта. Для этого нужно:

- в **Библиотеке (Library)** ФБ из раскрывающихся групп выбрать необходимый ФБ;
- захватить блок ЛКМ и, не отпуская кнопки мыши, перенести блок из **Библиотеки (Library)** ФБ на поле проекта;
- в точке отпускания ЛКМ возникает новый ФБ.

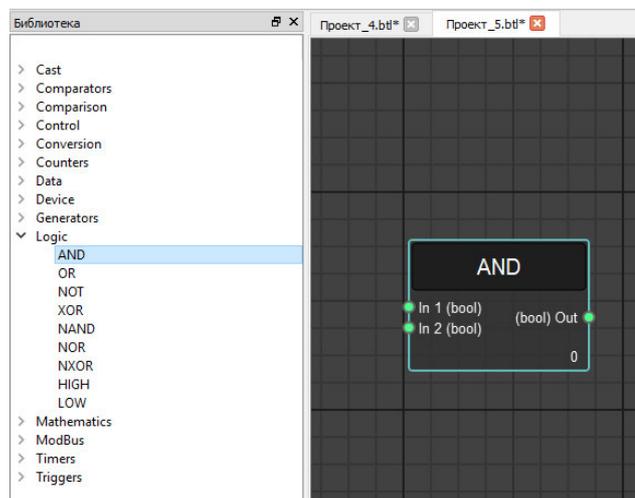


Рисунок 36. Пример размещения ФБ на поле проекта.

### 4.4.2 Создание соединения ФБ

Для создания соединительной линии можно нажатием на один из контактов блока начать процесс построения линии. После этого вслед за движениями курсора по полю проекта будет автоматически выстраиваться макет линии.

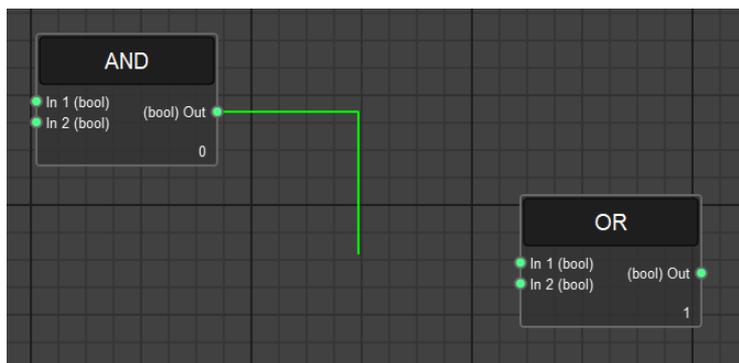


Рисунок 37. Пример начала построения соединительной линии.

Если в процессе построения линии сделать клик ЛКМ, то построенный участок линии зафиксируется, и дальнейшее построение будет производиться от места последнего клика мыши.



Рисунок 38. Пример построения участка соединительной линии.

Для соединения с блоком нужно в процессе построения линии щелкнуть ЛКМ по контакту блока, с которым планируется соединение. После этого на поле проекта появится завершенная соединительная линия.

Если до того, как соединительная линия будет окончательно построена, нажать кнопку **Esc** или **Delete**, построение соединительной линии будет отменено, линия исчезнет с поля проекта.

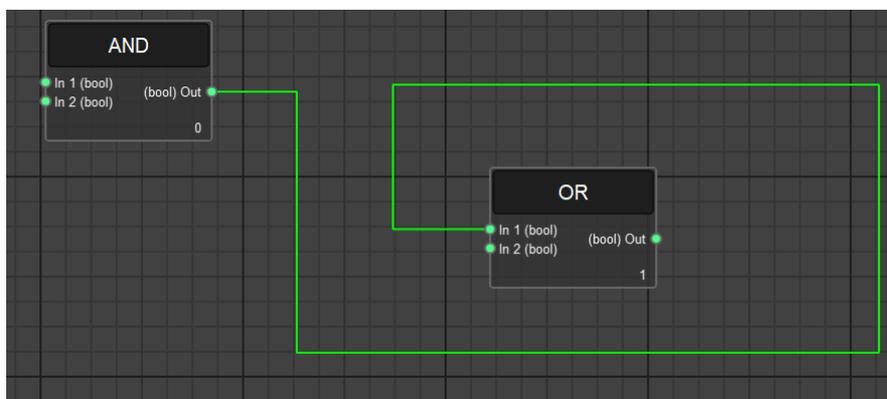


Рисунок 39. Пример построенной соединительной линии, устанавливающей связь между двумя ФБ.

### 4.4.3 Перенос сегмента линии

При необходимости можно перенести сегмент линии. Для этого нужно:

– навести стрелку мыши на соединительную линию, стрелка изменит свой вид на двухстороннюю стрелку, нажав на ЛКМ можно переместить необходимый сегмент линии;

или

– выбрать линию ЛКМ, линия станет выделенной и на линии появятся ее узлы (красные точки);

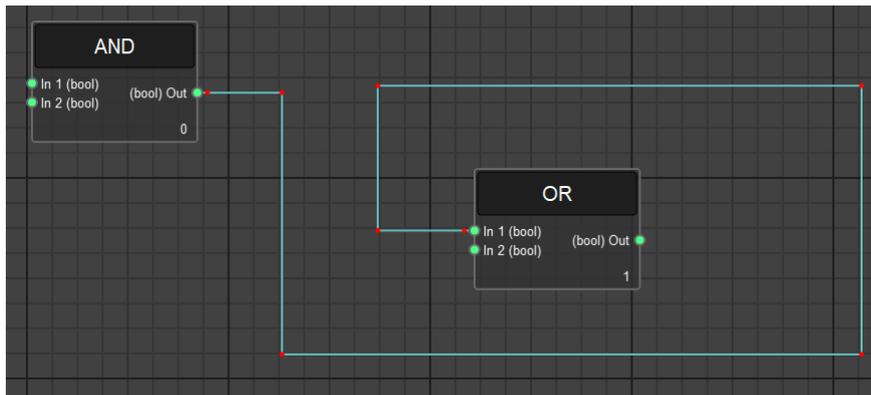


Рисунок 40. Пример соединительной линии с узлами.

– выбрать нужный узел линии ЛКМ и, не отпуская кнопку мыши, сместить его на новое место. При этом соседние узлы изменят свое положение, а участки линии, связанные с этими соседними узлами и самим перемещаемым узлом, изменят свою длину;

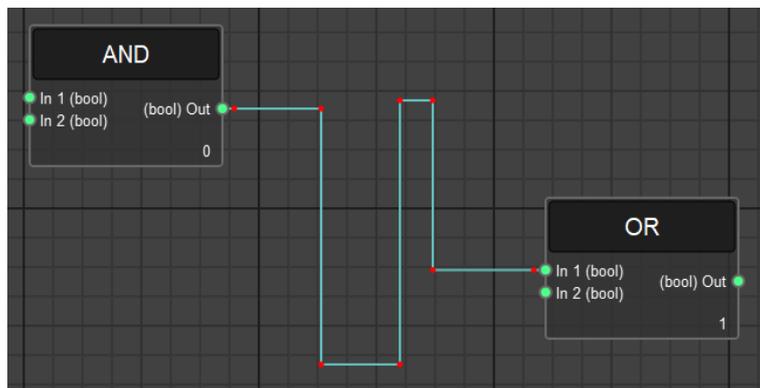


Рисунок 41. Пример перемещения узлов соединительной линии.

– при совмещении отдельных параллельных участков соединительной линии друг с другом, они «схлопываются». Таким образом, такая технология позволяет избавиться соединительную линию от лишних участков и упростить ее траекторию.

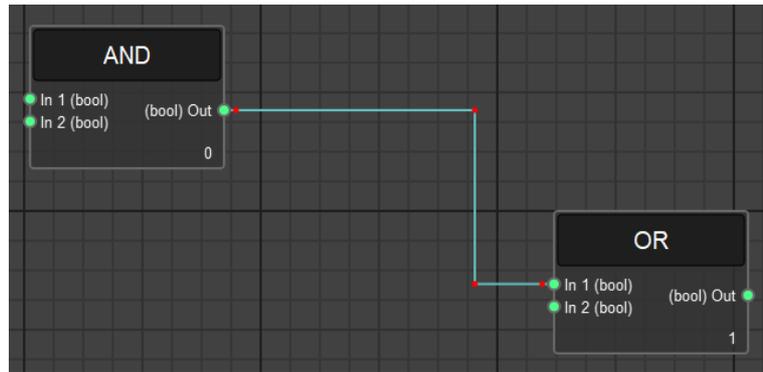


Рисунок 42. Пример упрощения траектории соединительной линии.

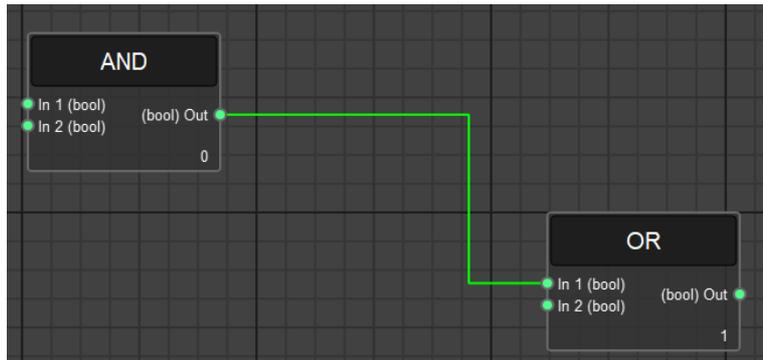


Рисунок 43. Окончательный вид соединительной линии.

## 5 Принципы создания программ

### 5.1 Перед первым подключением к контроллеру

Перед первым подключением к контроллеру необходимо установить в настройках системы IP-адрес контроллера, который приведен в руководстве по эксплуатации контроллера. Для этого нужно:

- запустить систему BitLogic;
- перейти в настройки Системы **Файл→Настройки... (File→Settings...)**;
- на вкладке **Подключение (Connection)** задать IP-адрес в соответствии с заводскими настройками контроллера, приведенными в руководстве по эксплуатации;
- на этой же вкладке задать **Порт (Port)** – 1518.

### 5.2 Соединение с контроллером

Включить контроллер. Подключение через систему BitLogic к выбранному контроллеру возможно только тогда, когда контроллер запущен.

В соответствии с п. 3.2 настоящего руководства установить в необходимое положение переключателя SW.

На **Панели инструментов (Toolbar)** нажать кнопку **Подключиться (Connect)**. В окне **Консоль (Console)** отобразятся сведения о состоянии соединения.

### 5.3 Добавление функциональных блоков

Прежде чем добавить из **Библиотеки (Library)** функциональные блоки в проект, необходимо установить подключение к включенному контроллеру, как описано выше, затем создать новый файл проекта или открыть уже существующий файл проекта.

Для добавления функционального блока в проект, нужно выбрать интересующий блок в окне **Библиотека (Library)** и перетащить его на поле проекта.

Список ФБ в **Библиотеке (Library)** следует в алфавитном порядке. Описание ФБ приведено в разделе 7.

После добавления необходимых функциональных блоков, нужно соединить их в графическом редакторе связями, перетаскивая связь мышью при нажатой ЛКМ. Выход блока соединяется с входом другого. Невозможно соединить вход блока с входом другого блока, равно как и выход блока с выходом другого.

### 5.4 Загрузка проекта

Созданный в системе BitLogic проект для интеллектуального реле загружается на контроллер через Ethernet-кабель категории не ниже CAT UTP5.

Загрузка проекта в контроллер выполняется нажатием кнопки **Загрузить (Upload)** на **Панели инструментов (Toolbar)**.

### 5.5 Запуск проекта

Перед запуском проекта необходимо убедиться, что контроллер запитан и подключен к ПК, проект собран и загружен на контроллер, как описано в пунктах 5.1.-5.4.

Запуск проекта осуществляется с помощью кнопки **Пуск (Run)** на **Панели инструментов (Toolbar)**.

## 5.6 Индикация контроллера

В зависимости от заданного режима работы, состояния соединения с компьютером, нахождения в режиме выполнения или остановки задачи, наличия аварийной ситуации можно определить состояние контроллера по индикации двух внешних светодиодов, расположенных на контроллере.

Описание индикаторов работы контроллера приведено в таблице 2.

Таблица 2. Соответствие состояний светодиодов режимам работы контроллера.

Индикатор	Состояние индикатора	Режим работы
"L1"–"L2"	Желтый цвет свечения непрерывно, красный цвет свечения непрерывно	Отсутствует прошивка на контроллере ( <b>пожалуйста, обратитесь к организации-продавцу для прошивки контроллера!</b> )
"L2"	Красный цвет свечения непрерывно	Авария модуля (проверяется в начальной фазе инициализации или в процессе работы). Для установления точной причины аварии модуля необходимо, в первую очередь, проверить правильность установки DIP-переключателей. Если режим работы модуля продолжается оставаться в аварийном состоянии, то не функционирует одно из периферийных устройств - RTC или EEPROM.
"L2"	Зеленый цвет свечения непрерывно	Инициализация модуля
"L2"	Зеленый цвет свечения длительностью 500 мс	Модуль запущен, исполнение задачи
"L1"–"L2"	Желтый цвет свечения длительностью 500 мс, зеленый цвет свечения длительностью 500 мс	Модуль запущен, задача в состоянии «Стоп»
"L1"–"L2"	Желтый цвет свечения длительностью 500 мс, зеленый цвет свечения непрерывно	Модуль запущен, нет задачи во FLASH памяти
"L1"–"L2"	Желтый цвет свечения непрерывно, красный цвет свечения длительностью 500 мс	Модуль запущен, ошибка целостности задачи во FLASH памяти
"L1"	Желтый цвет свечения длительностью 500 мс	Модуль запущен, исполнение одного шага задачи
"L1"–"L2"	Желтый цвет свечения непрерывно, зеленый цвет свечения длительностью 250 мс	Загрузка из/сохранение задачи во FLASH, проверка целостности задачи
"L2"	Зеленый цвет свечения длительностью 500 мс со сменой на красный цвет свечения длительностью 500 мс	Поломка калибровочных коэффициентов

## 6 Пример программы

На рис. 44 приведен скриншот программы, реализующий мигание лампочки, подключенной к контроллеру. Параметры реализации проекта: период свечения 2 сек. (2000 мс), длительность свечения лампочки 1 сек. (1000 мс), сигнал подается на лампочку со второго физического дискретного выхода контроллера.

Решение технической задачи мигания лампочки выполнено с использованием одного основного ФБ – генератора прямоугольных сигналов **fRECTGEN**, четырех блоков **CONST** и одного блока **PHYS\_OUT**. Описание параметров указанных блоков приведено в разделе 7 (см. таблицы 13, 15 и 14, соответственно). Время выполнения цикла задачи составляет 30 мс.

*Примечание:* Видимость некоторых выходов ФБ, получение данных с которых не является важным, но может оказаться полезным для пользователя, можно изменить в окне **Свойства (Properties)**, установив/сняв напротив названия выхода галочку. Например, на рис. 44 видно, что активны только три выхода ФБ **fRECTGEN**, один из которых является основным **Out** (*изменение видимости основных выходов ФБ недоступно*) и два дополнительных выхода **End period** и **End tau**, с которых можно получить данные об оставшемся времени до окончания периода и длительности прямоугольного импульса, соответственно.

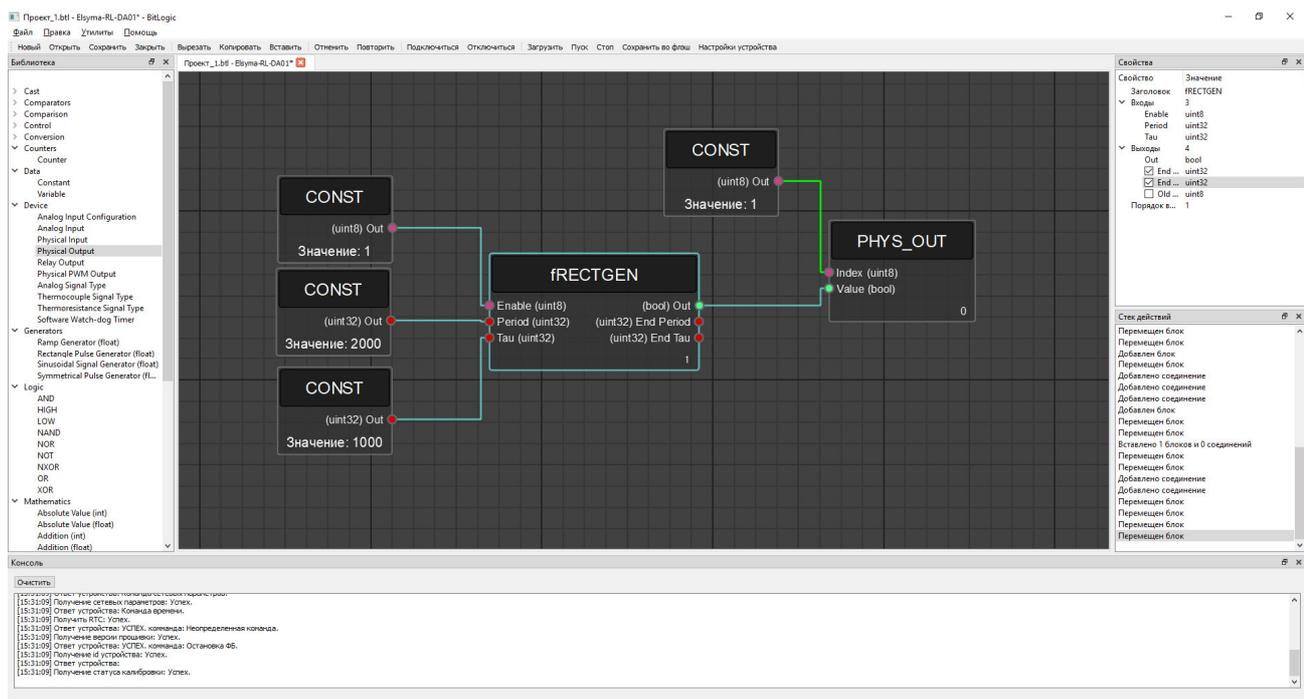
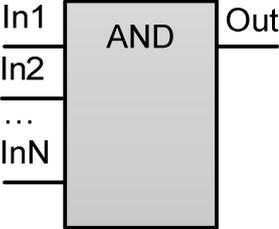
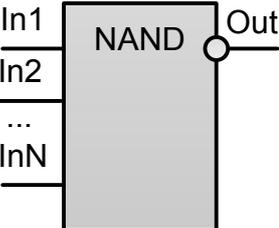


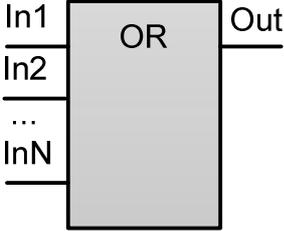
Рисунок 44. Пример реализации простого действующего проекта.

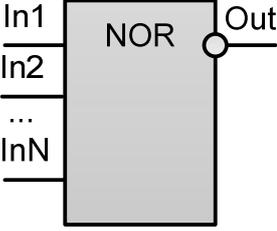
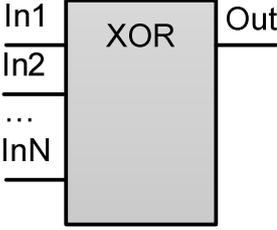
## 7 Описание функциональных блоков

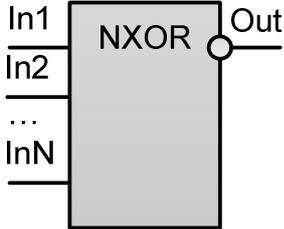
В программе реализованы следующие функциональные блоки:

Таблица 3. Блоки логические

Представление блока	Название	Описание блока																																													
	AND	<p>Операция «Логическое И» (AND)</p> <p>Данный функциональный блок (ФБ) используется для выполнения логических операций.</p> <p>На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если на все его входа (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) поступает логическая «1» (все входы включены – контакты замкнуты).</p> <p>Работе соответствует приведенная таблица состояний:</p> <table border="1" data-bbox="919 824 1382 1267"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>...</th> <th>InN</th> <th>Out</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p>Если на входы блока заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>, ..., <b>InN</b>.</p>	Тип данных					bool					In1	In2	...	InN	Out	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1
Тип данных																																															
bool																																															
In1	In2	...	InN	Out																																											
0	0	0	0	0																																											
1	0	0	0	0																																											
0	1	0	0	0																																											
0	0	1	0	0																																											
0	0	0	1	0																																											
1	1	1	1	1																																											
	NAND	<p>Операция «Логическое И-НЕ» (NAND)</p> <p>Данный ФБ используется для выполнения логических операций.</p> <p>На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если хотя бы на одном из его входов (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) установлен логический «0» (входа отключены). Если на все входа блока поступает логическая «1», то на его выходе устанавливается логический «0» (выход отключен).</p> <p>Работе соответствует приведенная таблица состояний:</p>																																													

Представление блока	Название	Описание блока																																													
		<table border="1" data-bbox="826 210 1321 645"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>...</th> <th>InN</th> <th>Out</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> <p data-bbox="762 689 1385 831">Если на входы блока заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p data-bbox="762 837 1385 904">Обязательные для инициализации входы ФБ: <b>In1, In2, ..., InN</b>.</p>	Тип данных					bool					In1	In2	...	InN	Out	0	0	0	0	1	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	1	0
Тип данных																																															
bool																																															
In1	In2	...	InN	Out																																											
0	0	0	0	1																																											
1	0	0	0	1																																											
0	1	0	0	1																																											
0	0	1	0	1																																											
0	0	0	1	1																																											
1	1	1	1	0																																											
	<p data-bbox="603 1480 651 1514">OR</p>	<p data-bbox="820 949 1326 983"><b>Операция «Логическое ИЛИ» (OR)</b></p> <p data-bbox="762 1021 1385 1088">Данный ФБ используется для выполнения логических операций.</p> <p data-bbox="762 1095 1385 1274">На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если хотя бы на один из его входов (<b>In1, In2, ..., InN</b>) поступает логическая «1» (контакты замкнуты).</p> <p data-bbox="762 1281 1385 1348">Работе соответствует приведенная таблица состояний:</p> <table border="1" data-bbox="826 1355 1321 1789"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>...</th> <th>InN</th> <th>Out</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p data-bbox="762 1839 1385 1973">Если на входы блока заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p data-bbox="762 1980 1385 2047">Обязательные для инициализации входы ФБ: <b>In1, In2, ..., InN</b>.</p>	Тип данных					bool					In1	In2	...	InN	Out	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	1	1
Тип данных																																															
bool																																															
In1	In2	...	InN	Out																																											
0	0	0	0	0																																											
1	0	0	0	1																																											
0	1	0	0	1																																											
0	0	1	0	1																																											
0	0	0	1	1																																											
1	1	1	1	1																																											

Представление блока	Название	Описание блока																																													
	NOR	<p>Операция «Логическое ИЛИ-НЕ» (NOR)</p> <p>Данный ФБ используется для выполнения логических операций.</p> <p>На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если на всех его входах (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) установлен логический «0» (все входа отключены – контакты разомкнуты).</p> <p>Работе соответствует приведенная таблица состояний:</p> <table border="1" data-bbox="903 609 1401 1048"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>...</th> <th>InN</th> <th>Out</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> <p>Если на входы блока заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>, ..., <b>InN</b>.</p>	Тип данных					bool					In1	In2	...	InN	Out	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	0
Тип данных																																															
bool																																															
In1	In2	...	InN	Out																																											
0	0	0	0	1																																											
1	0	0	0	0																																											
0	1	0	0	0																																											
0	0	1	0	0																																											
0	0	0	1	0																																											
1	1	1	1	0																																											
	XOR	<p>Операция «Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ» (XOR)</p> <p>Данный ФБ используется для выполнения логических операций.</p> <p>На выходе (<b>Out</b>) блока устанавливается логический «0» (выход отключен), если на все его входа (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) поступает логический «0» либо, если на его входах (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) устанавливается чётное количество логических «1». Если на входах блока (<b>In1</b>, <b>In2</b>, ..., <b>InN</b>) устанавливается нечётное количество логических «1», то на выход (<b>Out</b>) блока поступает логическая «1».</p> <p>Примером работы блока <b>XOR</b> может служить следующая таблица, соответствующая четырем входам блока:</p>																																													

Представление блока	Название	Описание блока																																													
		<table border="1" data-bbox="826 208 1321 645"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>In3</th> <th>In4</th> <th>Out</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> <p data-bbox="762 689 1385 831">Если на входы функции заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p data-bbox="762 837 1385 904">Обязательные для инициализации входы ФБ: <b>In1, In2, ..., InN</b>.</p>	Тип данных					bool					In1	In2	In3	In4	Out	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	1	1	0
Тип данных																																															
bool																																															
In1	In2	In3	In4	Out																																											
0	0	0	0	0																																											
1	0	0	0	1																																											
0	1	0	0	1																																											
0	0	1	0	1																																											
0	0	0	1	1																																											
1	1	1	1	0																																											
	NXOR	<p data-bbox="783 947 1362 1014" style="text-align: center;"><b>Операция «Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ» (NXOR)</b></p> <p data-bbox="762 1055 1385 1122">Данный ФБ используется для выполнения логических операций.</p> <p data-bbox="762 1128 1385 1491">На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если на всех его входах (<b>In1, In2, ..., InN</b>) установлен логический «0» либо, если на входа блока (<b>In1, In2, ..., InN</b>) подается чётное количество логических «1». Если на входах блока (<b>In1, In2, ..., InN</b>) устанавливается нечётное количество логических «1», то на выход (<b>Out</b>) блока поступает логический «0».</p> <p data-bbox="762 1498 1385 1603">Примером работы блока <b>NXOR</b> может служить следующая таблица, соответствующая четырем входам блока:</p> <table border="1" data-bbox="826 1608 1321 2045"> <thead> <tr> <th colspan="5">Тип данных</th> </tr> <tr> <th colspan="5">bool</th> </tr> <tr> <th>In1</th> <th>In2</th> <th>In3</th> <th>In4</th> <th>Out</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <p data-bbox="762 2089 1385 2157">Если на входы блока заведены целочисленные значения (0 или 1), то</p>	Тип данных					bool					In1	In2	In3	In4	Out	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1
Тип данных																																															
bool																																															
In1	In2	In3	In4	Out																																											
0	0	0	0	1																																											
1	0	0	0	0																																											
0	1	0	0	0																																											
0	0	1	0	0																																											
0	0	0	1	0																																											
1	1	1	1	1																																											

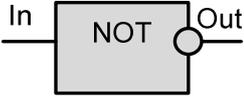
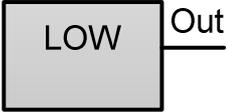
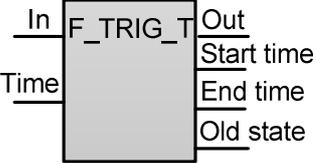
Представление блока	Название	Описание блока										
		<p>операция будет произведена над каждым битом в отдельности.</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2, ..., InN</b>.</p>										
	NOT	<p>Операция «Логическое НЕ» (NOT)</p> <p>Данный ФБ используется для инвертирования значения сигнала.</p> <p>На выходе (<b>Out</b>) блока устанавливается логическая «1» (выход включен), если на его входе (<b>In</b>) устанавливается логический «0» (контакты разомкнуты) и, наоборот: сигнал инвертируется.</p> <p>Работе соответствует приведенная таблица состояний:</p> <table border="1" data-bbox="1043 792 1256 1039"> <thead> <tr> <th colspan="2">Тип данных</th> </tr> <tr> <th colspan="2">bool</th> </tr> <tr> <th>In</th> <th>Out</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table> <p>Если на входы блока заведены целочисленные значения (0 или 1), то операция будет произведена над каждым битом в отдельности.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>	Тип данных		bool		In	Out	0	1	1	0
Тип данных												
bool												
In	Out											
0	1											
1	0											
	HIGH	<p>Операция «Постоянный высокий логический уровень» (HIGH)</p> <p>Данный ФБ формирует постоянный высокий логический уровень на своем выходе.</p> <p>На выходе (<b>Out</b>) блока устанавливается постоянный высокий логический уровень – логическая «1». Тип данных <b>Out – bool</b>.</p>										
	LOW	<p>Операция «Постоянный низкий логический уровень» (LOW)</p> <p>Данный ФБ формирует постоянный низкий логический уровень на своем выходе.</p> <p>На выходе (<b>Out</b>) блока устанавливается постоянный низкий логический уровень – логический «0». Тип данных <b>Out – bool</b>.</p>										

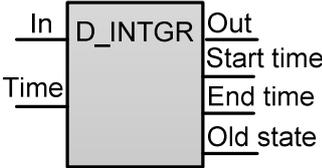
Таблица 4. Блоки выделения фронта сигнала

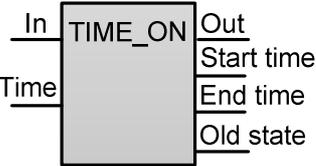
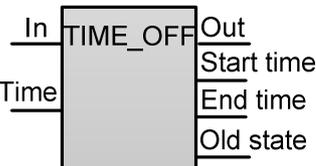
Представление блока	Название	Описание блока
	Front Trigger	<p>Операция «Выделение “восходящего” (переднего) фронта» (F_TRIG)</p> <p>Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе (<b>Out</b>) генерируется единичный импульс по переднему фронту сигнала на входе (<b>In</b>). Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In</b>, <b>Out</b> и <b>Old state</b> – <b>bool</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	Rear Trigger	<p>Операция «Выделение “нисходящего” (заднего) фронта» (R_TRIG)</p> <p>Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе (<b>Out</b>) генерируется единичный импульс по заднему фронту сигнала на входе (<b>In</b>). Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In</b>, <b>Out</b> и <b>Old state</b> – <b>bool</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	RS Trigger	<p>Операция «Установка блокировки с приоритетом сброса над установкой» (RS_TRIG)</p> <p>Блок устанавливает выход по входу <b>S</b> (Set) и сбрасывает по входу <b>R</b> (Reset). На выход <b>Direct out</b> поступает логическая «1», если логическая «1» подана на вход <b>S</b>, и логический «0», если логическая «1» подана на вход <b>R</b>. Выход <b>Inverse out</b> является инверсным относительно <b>Direct out</b>. Вход <b>R</b> имеет приоритет, т. е. если <b>R=S=1</b>, то на выходе <b>Direct out</b> поступает нуль. При <b>R=S=0</b> состояние выходов не меняется. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>S</b>. Тип данных <b>S</b>, <b>R</b>, <b>Direct out</b>, <b>Inverse out</b> и <b>Old state</b> – <b>bool</b>.</p> <p>Работе соответствует приведенная таблица состояний:</p>

Представление блока	Название	Описание блока																				
		<table border="1" data-bbox="938 210 1362 506"> <thead> <tr> <th>S</th> <th>R</th> <th>Direct out</th> <th>Inverse out</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Out<sub>i-1</sub></td> <td><math>\overline{\text{Out}_{i-1}}</math></td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table> <p data-bbox="831 546 1469 613">Обязательные для инициализации входы ФБ: S, R.</p>	S	R	Direct out	Inverse out	0	0	Out <sub>i-1</sub>	$\overline{\text{Out}_{i-1}}$	1	0	1	0	0	1	0	1	1	1	0	0
S	R	Direct out	Inverse out																			
0	0	Out <sub>i-1</sub>	$\overline{\text{Out}_{i-1}}$																			
1	0	1	0																			
0	1	0	1																			
1	1	0	0																			
	<p data-bbox="603 1133 783 1200">Front Trigger Time</p>	<p data-bbox="842 658 1458 801">Операция «Выделение “восходящего” (переднего) фронта и удержание выхода в течение заданного времени параметром Time» (F_TRIG_T)</p> <p data-bbox="831 842 1469 1384">Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе (Out) генерируется единичный импульс заданной длительности (устанавливается на входе Time) по переднему фронту сигнала на входе (In). На выходе Start time выводится время с момента прихода переднего фронта сигнала на вход In. На выходе End time выводится время, оставшееся до окончания длительности временной задержки, установленной на входе Time. Выход Old state запоминает предыдущее состояние входа In. Тип данных In, Out и Old state – bool, Time, Start time и End time – uint32.</p> <p data-bbox="831 1391 1469 1458">Обязательные для инициализации входы ФБ: In, Time.</p>																				
	<p data-bbox="611 1783 778 1850">Rear Trigger Time</p>	<p data-bbox="858 1503 1442 1646">Операция «Выделение “нисходящего” (заднего) фронта и удержание выхода в течение заданного времени параметром Time» (R_TRIG_T)</p> <p data-bbox="831 1686 1469 2154">Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе (Out) генерируется единичный импульс заданной длительности (устанавливается на входе Time) по заднему фронту сигнала на входе (In). На выходе Start time выводится время с момента прихода заднего фронта сигнала на вход In. На выходе End time выводится время, оставшееся до окончания длительности временной задержки, установленной на входе Time. Выход Old state запоминает предыдущее состояние</p>																				

Представление блока	Название	Описание блока
		входа <b>In</b> . Тип данных <b>In, Out</b> и <b>Old state</b> – <b>bool</b> , <b>Time, Start time</b> и <b>End time</b> – <b>uint32</b> . Обязательные для инициализации входы ФБ: <b>In, Time</b> .

Таблица 5. Блоки работы с таймером

Представление блока	Название	Описание блока
	Debounce Integration	<p>Операция «<b>Интегрирование входного сигнала для устранения дребезга выключателя</b>» (<b>D_INTGR</b>)</p> <p>Блок используется для устранения дребезга кнопки (выключателя). На выходе (<b>Out</b>) блока генерируется единичный импульс, если результатом по истечении заданного времени интегрирования сигнала на входе (<b>In</b>) является единичный импульс. Длительность дребезга кнопки лежит в диапазоне от 10 до 50 мс. Длительность интегрирования сигнала задается на входе <b>Time</b>, но должна быть не менее 100 мс. На выходе <b>Start time</b> выводится время с момента прихода переднего фронта сигнала на вход <b>In</b>. На выходе <b>End time</b> выводится время, оставшееся до окончания длительности временной задержки, установленной на входе <b>Time</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In, Out</b> и <b>Old state</b> – <b>bool</b>, <b>Time, Start time</b> и <b>End time</b> – <b>uint32</b>. Обязательные для инициализации входы ФБ: <b>In, Time</b>.</p>
	Timer	<p>Блок «<b>Таймер</b>» (<b>TIMER</b>)</p> <p>Блок используется для определения прошедшего времени с момента включения блока. На выходе (<b>Time</b>) блока начнется отсчет времени после того, как на вход (<b>In</b>) поступит логическая «1». Предельное значение на выходе – 49 суток. На выходе <b>Time</b> выводится время (в мс) с момента прихода сигнала на вход <b>In</b>. На выходе <b>Start time</b> выводится время (0.1 сек), определяющее точность подсчета времени на выходе <b>Time</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In, Old state</b> – <b>bool</b>, <b>Time</b> и <b>Start time</b> – <b>uint32</b>.</p>

Представление блока	Название	Описание блока
	Timer ON	<p>Обязательный для инициализации вход ФБ: <b>In</b>.</p> <p>Блок «<b>Таймер с задержкой включения</b>» (<b>TIME_ON</b>)</p> <p>Блок используется для операции задержки передачи сигнала.</p> <p>На выходе (<b>Out</b>) блока появится логическая «1» с задержкой относительно переднего фронта сигнала на входе (<b>In</b>). Выход включается логической «1», если на входе <b>Time</b> установлена продолжительность сигнала не менее длительности <b>T</b>, а выключается по спаду входного сигнала.</p> <p>Допустимый диапазон значений <b>T</b>: от 0 до 10800000 мс, или 3 часа.</p> <p>На выходе <b>Start time</b> выводится время с момента прихода переднего фронта сигнала на вход <b>In</b>. На выходе <b>End time</b> выводится время, оставшееся до окончания длительности временной задержки, установленной на входе <b>Time</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In</b>, <b>Out</b> и <b>Old state</b> – <b>bool</b>, <b>Time</b>, <b>Start time</b> и <b>End time</b> – <b>uint32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In</b>, <b>Time</b>.</p>
	Timer OFF	<p>Блок «<b>Таймер с задержкой выключения</b>» (<b>TIME_OFF</b>)</p> <p>Блок используется для задержки отключения выхода.</p> <p>На выходе (<b>Out</b>) блока появится логическая «1» по заднему фронту сигнала на входе (<b>In</b>), а начало отсчета времени задержки отключения (<math>T_{\text{зад}}</math>) (устанавливается на входе <b>Time</b>) происходит по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится логический «0» с задержкой <math>T_{\text{зад}}</math>.</p> <p>Допустимый диапазон значений <math>T_{\text{зад}}</math>: от 0 до 10800000 мс, или 3 часа.</p> <p>На выходе <b>Start time</b> выводится время с момента прихода заднего фронта сигнала на вход <b>In</b>. На выходе <b>End time</b> выводится время, оставшееся до окончания длительности временной задержки, установленной на входе <b>Time</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In</b>, <b>Out</b> и <b>Old state</b> – <b>bool</b>, <b>Time</b>, <b>Start time</b> и <b>End time</b> – <b>uint32</b>.</p>

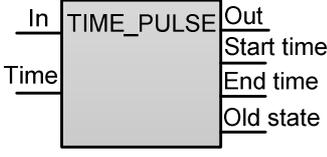
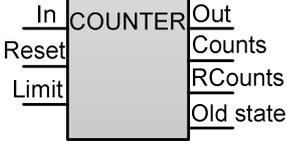
Представление блока	Название	Описание блока
		Обязательные для инициализации входы ФБ: <b>In, Time</b> .
	Time Pulse	<p><b>Блок «Генератор импульса заданной длительности» (TIME_PULSE)</b></p> <p>Блок используется для формирования импульса включения выхода на заданный интервал времени.</p> <p>На выходе (<b>Out</b>) блока появится логическая «1» по фронту сигнала на входе (<b>In</b>). После запуска выход (<b>Out</b>) не реагирует на изменение значения входного сигнала на интервале <math>T_{имп}</math> (устанавливается на входе <b>Time</b>), а по истечении этого интервала сбрасывается в «0».</p> <p>Допустимый диапазон значений <math>T_{имп}=T</math>: от 0 до 10800000 мс, или 3 часа.</p> <p>На выходе <b>Start time</b> выводится время с момента прихода переднего фронта сигнала на вход <b>In</b>. На выходе <b>End time</b> выводится время, оставшееся до окончания длительности временной задержки, установленной на входе <b>Time</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In, Out</b> и <b>Old state</b> – <b>bool</b>, <b>Time, Start time</b> и <b>End time</b> – <b>uint32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In, Time</b>.</p>

Таблица 6. Блоки счетные

Представление блока	Название	Описание блока
		<p><b>Блок «Инкрементный счетчик с управляемым сбросом» (COUNTER)</b></p> <p>Блок используется в случае необходимости формирования единичного импульса после прихода заданного количества сигналов на его вход.</p> <p>На выходе (<b>Out</b>) блока генерируется единичный импульс, после того как на вход блока (<b>In</b>) поступит последний сигнал из заданного количества (устанавливается на входе <b>Limit</b>) входных сигналов. Для сброса любого количества (меньшего, чем значение входа <b>Limit</b>) пришедших на вход (<b>In</b>) логических «1» используется подача логической «1» на вход сброса (<b>Reset</b>). С помощью обратной связи, связывающей выход <b>Out</b> со входом <b>Reset</b>, можно задать режим работы блока, при котором блок</p>
	Counter	

Представление блока	Название	Описание блока
		<p>автоматически сбрасывает в ноль накопленное значение пришедших на вход (<b>In</b>) сигналов. При этом с приходом очередного сигнала на вход (<b>In</b>) блока, цикл подсчета входных сигналов повторяется. Значение на входе <b>Limit</b> ограничено в диапазоне от 1 до 4294967295.</p> <p>На выходе <b>Counts</b> выводится текущее значение поступивших на вход <b>In</b> логических «1», на выходе <b>RCounts</b> – количество необходимых логических «1», поступающих на вход <b>In</b>, для подачи единичного импульса на выход <b>Out</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>In</b>. Тип данных <b>In</b>, <b>Reset</b>, <b>Out</b> и <b>Old state</b> – <b>bool</b>, <b>Limit</b>, <b>Counts</b> и <b>RCounts</b> – <b>uint32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In</b>, <b>Reset</b>, <b>Limit</b>.</p>

Таблица 7. Блоки сравнения

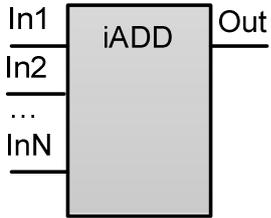
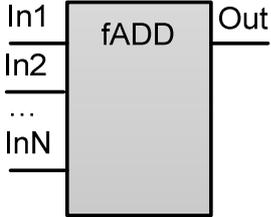
Представление блока	Название	Описание блока
	Equality int	<p>Операция «Равенство» (<b>iEQ</b>)</p> <p>Проверяет два целочисленных значения на равенство.</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая целочисленные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 = In2</b>;  <b>Out=0</b>, если <b>In1 ≠ In2</b>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – <b>int32</b>, <b>Out</b> – <b>bool</b>.</p> <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>.</p>
	Equality float	<p>Операция «Равенство» (<b>fEQ</b>)</p> <p>Проверяет два вещественных значения на равенство.</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая вещественные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 = In2</b>;  <b>Out=0</b>, если <b>In1 ≠ In2</b>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – <b>float</b>, <b>Out</b> – <b>bool</b>.</p> <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>.</p>

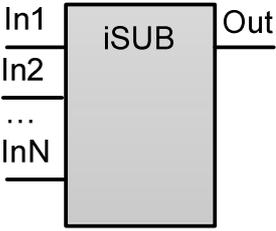
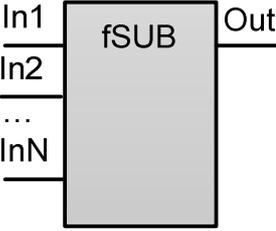
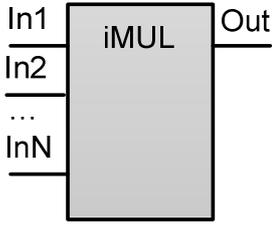
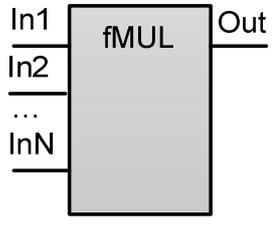
Представление блока	Название	Описание блока
	Greater Than int	<p>Операция «Больше» (iGT)</p> <p>Сравнивает два целочисленных значения по критерию «строго больше».</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая целочисленные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 &gt; In2</b>;  <b>Out=0</b>, если <b>In1 ≤ In2</b>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – <b>int32</b>, <b>Out</b> – <b>bool</b>.  Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	Greater Than float	<p>Операция «Больше» (fGT)</p> <p>Сравнивает два вещественных значения по критерию «строго больше».</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая вещественные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 &gt; In2</b>;  <b>Out=0</b>, если <b>In1 ≤ In2</b>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – <b>float</b>, <b>Out</b> – <b>bool</b>.  Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	Less Than int	<p>Операция «Меньше» (iLT)</p> <p>Сравнивает два целочисленных значения по критерию «строго меньше».</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая целочисленные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 &lt; In2</b>;  <b>Out=0</b>, если <b>In1 ≥ In2</b>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – <b>int32</b>, <b>Out</b> – <b>bool</b>.  Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	Less Than float	<p>Операция «Меньше» (fLT)</p> <p>Сравнивает два вещественных значения по критерию «строго меньше».</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая вещественные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <b>In1 &lt; In2</b>;</p>

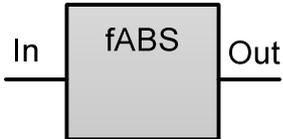
Представление блока	Название	Описание блока
		<p><b>Out=0</b>, если <math>In1 \geq In2</math>.            Тип данных <b>In1</b> и <b>In2</b> – <b>float</b>, <b>Out</b> – <b>bool</b>.            Логическая 1 на выходе соответствует числу «1».            Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	<p>Greater or Equals int</p>	<p>Операция «<b>Больше или равно</b>» (<b>iGE</b>)</p> <p>Сравнивает два целочисленных значения по критерию «больше или равно».            Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая целочисленные значения на входе <b>In1</b> и <b>In2</b> между собой:  <b>Out=1</b>, если <math>In1 \geq In2</math>;  <b>Out=0</b>, если <math>In1 &lt; In2</math>.            Тип данных <b>In1</b> и <b>In2</b> – <b>int32</b>, <b>Out</b> – <b>bool</b>.            Логическая 1 на выходе соответствует числу «1».            Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	<p>Greater or Equals float</p>	<p>Операция «<b>Больше или равно</b>» (<b>fGE</b>)</p> <p>Сравнивает два вещественных значения по критерию «больше или равно».            Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая вещественные значения на входе <b>In1</b> и <b>In2</b> между собой:  <b>Out=1</b>, если <math>In1 \geq In2</math>;  <b>Out=0</b>, если <math>In1 &lt; In2</math>.            Тип данных <b>In1</b> и <b>In2</b> – <b>float</b>, <b>Out</b> – <b>bool</b>.            Логическая 1 на выходе соответствует числу «1».            Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>
	<p>Less or Equals int</p>	<p>Операция «<b>Меньше или равно</b>» (<b>iLE</b>)</p> <p>Сравнивает два целочисленных значения по критерию «меньше или равно».            Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая целочисленные значения на входе <b>In1</b> и <b>In2</b> между собой:  <b>Out=1</b>, если <math>In1 \leq In2</math>;  <b>Out=0</b>, если <math>In1 &gt; In2</math>.            Тип данных <b>In1</b> и <b>In2</b> – <b>int32</b>, <b>Out</b> – <b>bool</b>.            Логическая 1 на выходе соответствует числу «1».            Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>

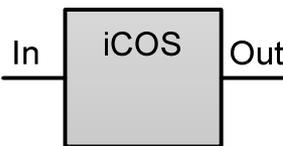
Представление блока	Название	Описание блока
	Less or Equals float	<p>Операция «Меньше или равно» (fLE)</p> <p>Сравнивает два вещественных значения по критерию «меньше или равно».</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая вещественные значения на входе <b>In1</b> и <b>In2</b> между собой:</p> <p><b>Out=1</b>, если <math>In1 \leq In2</math>;  <b>Out=0</b>, если <math>In1 &gt; In2</math>.</p> <p>Тип данных <b>In1</b> и <b>In2</b> – float, <b>Out</b> – bool.</p> <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2</b>.</p>

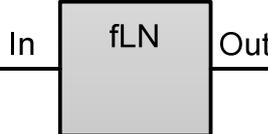
Таблица 8. Блоки арифметические

Представление блока	Название	Описание блока
	Addition int	<p>Операция «Суммирование» (iADD)</p> <p>Суммирование двух и более целочисленных чисел.</p> <p>Блок предназначен для суммирования двух и более целочисленных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее целочисленное значение суммы (<math>Out = In1 + In2 + \dots + InN</math>) двух и более целочисленных чисел, действующих на входах <b>In1, In2, ..., InN</b>. Тип данных <b>In1, In2, ..., InN</b> и <b>Out</b> – int32.</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2, ... InN</b>.</p>
	Addition float	<p>Операция «Суммирование» (fADD)</p> <p>Суммирование двух и более вещественных чисел.</p> <p>Блок предназначен для суммирования двух и более вещественных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее вещественное значение суммы (<math>Out = In1 + In2 + \dots + InN</math>) двух и более вещественных чисел, действующих на входах <b>In1, In2, ..., InN</b>. Тип данных <b>In1, In2, ..., InN</b> и <b>Out</b> – float.</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2, ... InN</b>.</p>

Представление блока	Название	Описание блока
	Subtraction int	<p align="center"><b>Операция «Разность» (iSUB)</b></p> <p>Определение разности двух и более целочисленных чисел.</p> <p>Блок предназначен для определения разности двух и более целочисленных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее целочисленное значение разности (<math>Out = In1 - In2 - \dots - InN</math>) двух и более целочисленных чисел, действующих на входах <b>In1</b>, <b>In2</b>, ..., <b>InN</b>. Тип данных <b>In1</b>, <b>In2</b>, ..., <b>InN</b> и <b>Out</b> – <b>int32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>, ... <b>InN</b>.</p>
	Subtraction float	<p align="center"><b>Операция «Разность» (fSUB)</b></p> <p>Определение разности двух и более вещественных чисел.</p> <p>Блок предназначен для определения разности двух и более вещественных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее вещественное значение разности (<math>Out = In1 - In2 - \dots - InN</math>) двух и более вещественных чисел, действующих на входах <b>In1</b>, <b>In2</b>, ..., <b>InN</b>. Тип данных <b>In1</b>, <b>In2</b>, ..., <b>InN</b> и <b>Out</b> – <b>float</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>, ... <b>InN</b>.</p>
	Multiplication int	<p align="center"><b>Операция «Умножение» (iMUL)</b></p> <p>Произведение двух и более целочисленных чисел.</p> <p>Блок предназначен для произведения двух и более целочисленных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее целочисленное значение произведения (<math>Out = In1 \cdot In2 \cdot \dots \cdot InN</math>) двух и более целочисленных чисел, действующих на входах <b>In1</b>, <b>In2</b>, ..., <b>InN</b>. Тип данных <b>In1</b>, <b>In2</b>, ..., <b>InN</b> и <b>Out</b> – <b>int32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In1</b>, <b>In2</b>, ... <b>InN</b>.</p>
	Multiplication float	<p align="center"><b>Операция «Умножение» (fMUL)</b></p> <p>Произведение двух и более вещественных чисел.</p> <p>Блок предназначен для произведения двух и более вещественных чисел.</p> <p>На выход (<b>Out</b>) блока поступает результирующее вещественное значение</p>

Представление блока	Название	Описание блока
		<p>произведения (<math>Out = In1 \cdot In2 \cdot \dots \cdot InN</math>) двух и более вещественных чисел, действующих на входах <math>In1, In2, \dots, InN</math>. Тип данных <math>In1, In2, \dots, InN</math> и <math>Out</math> – float.</p> <p>Обязательные для инициализации входы ФБ: <math>In1, In2, \dots, InN</math>.</p>
	Division int	<p>Операция «Деление» (iDIV)</p> <p>Деление двух целочисленных чисел. Блок предназначен для деления двух целочисленных чисел. На выход (<math>Out</math>) блока поступает результирующее целочисленное значение от деления (<math>Out = In1 / In2</math>) двух целочисленных чисел, действующих на входах <math>In1</math> и <math>In2</math>, соответственно. Целочисленное значение остатка от деления поступает на выход <math>Mod</math>. Тип данных <math>In1, In2, Out</math> и <math>Mod</math> – int32.</p> <p>Обязательные для инициализации входы ФБ: <math>In1, In2</math>.</p>
	Division float	<p>Операция «Деление» (fDIV)</p> <p>Деление двух вещественных чисел. Блок предназначен для деления двух вещественных чисел. На выход (<math>Out</math>) блока поступает результирующее вещественное значение деления (<math>Out = In1 / In2</math>) двух вещественных чисел, действующих на входах <math>In1</math> и <math>In2</math>. Тип данных <math>In1, In2</math> и <math>Out</math> – float.</p> <p>Обязательные для инициализации входы ФБ: <math>In1, In2</math>.</p>
	Absolute Value int	<p>Операция «Абсолютное значение» (iABS)</p> <p>Определение абсолютного значения целочисленного числа. Блок вычисляет абсолютное значение (модуль) целочисленного числа. На выход (<math>Out</math>) блока поступает абсолютное значение (модуль) целочисленного числа, действующего на его входе (<math>In</math>) (<math>Out =  In </math>). Тип данных <math>In</math> и <math>Out</math> – int32.</p> <p>Обязательный для инициализации вход ФБ: <math>In</math>.</p>
	Absolute Value float	<p>Операция «Абсолютное значение» (fABS)</p> <p>Определение абсолютного значения вещественного числа. Блок вычисляет абсолютное значение</p>

Представление блока	Название	Описание блока
		<p>(модуль) вещественного числа.</p> <p>На выход (<b>Out</b>) блока поступает абсолютное значение (модуль) вещественного числа, действующего на его входе (<b>In</b>) (<b>Out =  In </b>). Тип данных <b>In</b> и <b>Out</b> – <b>float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	SIN int	<p>Операция «Синус» (iSIN)</p> <p>Определение значения синуса целочисленного числа.</p> <p>Блок вычисляет значение синуса целочисленного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение синуса целочисленного числа, действующего на его входе (<b>In</b>) (<b>Out = SIN(In)</b>). Тип данных <b>In</b> – <b>int32</b>, <b>Out</b> – <b>float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	SIN float	<p>Операция «Синус» (fSIN)</p> <p>Определение значения синуса вещественного числа.</p> <p>Блок вычисляет значение синуса вещественного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение синуса вещественного числа, действующего на его входе (<b>In</b>) (<b>Out = SIN(In)</b>). Тип данных <b>In</b> – <b>float</b>, <b>Out</b> – <b>float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	COS int	<p>Операция «Косинус» (iCOS)</p> <p>Определение значения косинуса целочисленного числа.</p> <p>Блок вычисляет значение косинуса целочисленного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение косинуса целочисленного числа, действующего на его входе (<b>In</b>) (<b>Out = COS(In)</b>). Тип данных <b>In</b> – <b>int32</b>, <b>Out</b> – <b>float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	COS float	<p>Операция «Косинус» (fCOS)</p> <p>Определение значения косинуса вещественного числа.</p> <p>Блок вычисляет значение косинуса вещественного числа.</p>

Представление блока	Название	Описание блока
		<p>На выход (<b>Out</b>) блока поступает значение косинуса вещественного числа, действующего на его входе (<b>In</b>) (<b>Out = COS(In)</b>). Тип данных <b>In – float, Out – float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	LN int	<p>Операция «<b>Натуральный логарифм</b>» (<b>iLN</b>)</p> <p>Определение значения натурального логарифма целочисленного числа.</p> <p>Блок вычисляет значение натурального логарифма целочисленного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение натурального логарифма целочисленного числа, действующего на его входе (<b>In</b>) (<b>Out = LN(In)</b>). Тип данных <b>In – int32, Out – float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	LN float	<p>Операция «<b>Натуральный логарифм</b>» (<b>fLN</b>)</p> <p>Определение значения натурального логарифма вещественного числа.</p> <p>Блок вычисляет значение натурального логарифма вещественного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение натурального логарифма вещественного числа, действующего на его входе (<b>In</b>) (<b>Out = LN(In)</b>). Тип данных <b>In – float, Out – float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	SQRT int	<p>Операция «<b>Квадратный корень</b>» (<b>iSQRT</b>)</p> <p>Определение значения квадратного корня целочисленного числа.</p> <p>Блок вычисляет значение квадратного корня целочисленного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение квадратного корня целочисленного числа, действующего на его входе (<b>In</b>) (<b>Out = SQRT(In)</b>). Тип данных <b>In – int32, Out – float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	SQRT float	<p>Операция «<b>Квадратный корень</b>» (<b>fSQRT</b>)</p> <p>Определение значения квадратного корня вещественного числа.</p> <p>Блок вычисляет значение квадратного корня вещественного числа.</p> <p>На выход (<b>Out</b>) блока поступает значение</p>

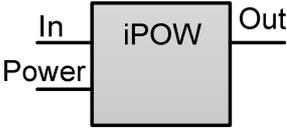
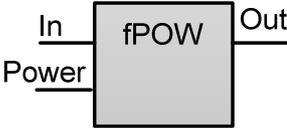
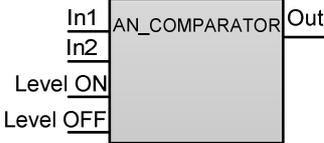
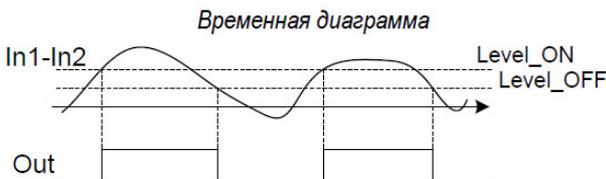
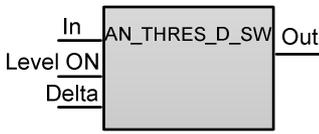
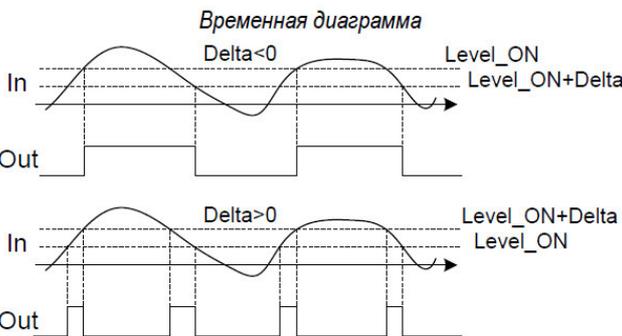
Представление блока	Название	Описание блока
		<p>квадратного корня вещественного числа, действующего на его входе (<b>In</b>) (<b>Out = SQRT(In)</b>). Тип данных <b>In – float, Out – float</b>.</p> <p>Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	POW int	<p>Операция «<b>Возведение в степень</b>» (<b>iPOW</b>)</p> <p>Определение значения возведения целочисленного числа в степень.</p> <p>Блок вычисляет значение возведения целочисленного числа в степень.</p> <p>На выход (<b>Out</b>) блока поступает значение возведения целочисленного числа, действующего на его входе (<b>In</b>), в степень на входе (<b>Power</b>) (<b>Out = POW(In, Power)</b>). Тип данных <b>In – int32, Power – uint32, Out – float</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In, Power</b>.</p>
	POW float	<p>Операция «<b>Возведение в степень</b>» (<b>fPOW</b>)</p> <p>Определение значения возведения вещественного числа в степень.</p> <p>Блок вычисляет значение возведения вещественного числа в степень.</p> <p>На выход (<b>Out</b>) блока поступает значение возведения вещественного числа, действующего на его входе (<b>In</b>), в степень на входе (<b>Power</b>) (<b>Out = POW(In, Power)</b>). Тип данных <b>In и Out – float, Power – uint32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In, Power</b>.</p>

Таблица 9. Блоки компараторов

Представление блока	Название	Описание блока
	<p>Analogue Comparator</p>	<p>Блок «<b>Аналоговый компаратор</b>» (<b>AN_COMPARATOR</b>)</p> <p>Устанавливает и сбрасывает выход, сравнивая аналоговые сигналы на входах с заданными пороговыми уровнями.</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая аналоговые сигналы на входах <b>In1</b> и <b>In2</b> с заданными пороговыми уровнями по правилам:</p> <p><b>Out=1</b>, если <math>In1 - In2 \geq Level\ ON</math> (строго больше при <math>Level\ ON = Level\ OFF</math>);</p> <p><b>Out=0</b>, если <math>In1 - In2 \leq Level\ OFF</math>.</p> <p>Тип данных <b>In1, In2, Level ON, Level OFF – float, Out – bool</b>.</p>

Представление блока	Название	Описание блока
		<p style="text-align: center;"><i>Временная диаграмма</i></p>  <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In1, In2, Level ON, Level OFF.</b></p>
	<p>Analogue Differential Threshold Switch</p>	<p style="text-align: center;"><b>Блок «Аналоговый дифференциальный пороговый выключатель» (AN_THRES_D_SW)</b></p> <p>Устанавливает и сбрасывает выход, сравнивая аналоговый сигнал на входе с заданными пороговыми уровнями.</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая аналоговый сигнал на входе <b>In</b> с заданными пороговыми уровнями по правилам:</p> <ol style="list-style-type: none"> <li>1) если параметр <b>Delta</b> отрицателен (<b>Level ON &gt; Level OFF = Level ON + Delta</b>), то       <ul style="list-style-type: none"> <li><b>Out=1</b>, если <b>In ≥ Level ON</b> (строго больше при <b>Delta = 0</b>);</li> <li><b>Out=0</b>, если <b>In ≤ Level ON + Delta</b>;</li> </ul> </li> <li>2) если параметр <b>Delta</b> положителен (<b>Level ON &lt; Level OFF = Level ON + Delta</b>), то       <ul style="list-style-type: none"> <li><b>Out=1</b>, если <b>Level ON + Delta ≥ In ≥ Level ON</b>;</li> <li><b>Out=0</b>, если <b>In &gt; Level ON + Delta</b> или <b>In &lt; Level ON</b>.</li> </ul> </li> </ol> <p>Тип данных <b>In, Level ON, Delta – float, Out – bool.</b></p> <p style="text-align: center;"><i>Временная диаграмма</i></p>  <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In, Level ON, Delta.</b></p>

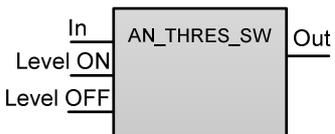
Представление блока	Название	Описание блока
	<p>Analog Threshold Switch</p>	<p><b>Блок «Аналоговый пороговый выключатель» (AN_THRES_SW)</b></p> <p>Устанавливает и сбрасывает выход, сравнивая аналоговый сигнал на входе с заданными пороговыми уровнями.</p> <p>Блок устанавливает или сбрасывает выход (<b>Out</b>), сравнивая аналоговый сигнал на входе <b>In</b> с заданными пороговыми уровнями по правилам:</p> <p><b>Out=1</b>, если <math>In \geq \text{Level ON}</math> (строго больше при <math>\text{Level ON} = \text{Level OFF}</math>);</p> <p><b>Out=0</b>, если <math>In \leq \text{Level OFF}</math>.</p> <p>Тип данных <b>In</b>, <b>Level ON</b>, <b>Level OFF</b> – <b>float</b>, <b>Out</b> – <b>bool</b>.</p> <p style="text-align: center;"><i>Временная диаграмма</i></p>  <p>Логическая 1 на выходе соответствует числу «1».</p> <p>Обязательные для инициализации входы ФБ: <b>In</b>, <b>Level ON</b>, <b>Level OFF</b>.</p>

Таблица 10. Блоки конвертации сигнала

Представление блока	Название	Описание блока
	<p>Average Value float</p>	<p><b>Операция «Среднее значение сигнала» (fAVER)</b></p> <p>Блок предназначен для вычисления среднего значения входного сигнала.</p> <p>На вход блока (<b>In</b>) подается сигнал, для которого необходимо рассчитать среднее значение. На вход <b>Period</b> задается число тактов расчёта блока, нужное для вычисления среднего значения. После завершения вычисления среднее значение появится на выходе блока (<b>Out</b>). Тип данных <b>In</b> и <b>Out</b> – <b>float</b>, <b>Period</b> – <b>uint32</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>In</b>, <b>Period</b>.</p>
	<p>Fractional Part float</p>	<p><b>Операция «Дробная часть числа в формате float» (fFRAC)</b></p> <p>Вычисляет дробную часть числа в формате <b>float</b>. Тип данных <b>In</b> и <b>Out</b> – <b>float</b>.</p> <p>Обязательный для инициализации вход ФБ:</p>

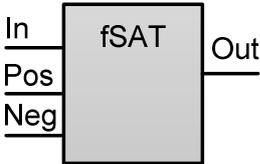
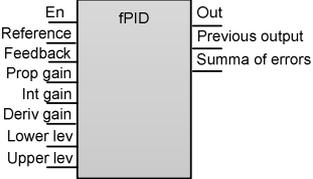
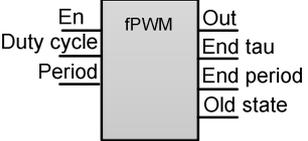
Представление блока	Название	Описание блока
	Integer Part float	<p>Операция «Целая часть числа в формате float» (fINT)</p> <p>Вычисляет целую часть числа в формате float. Тип данных <b>In</b> – float, <b>Out</b> – int32. Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	Saturation float	<p>Блок «Ограничение входного сигнала» (fSAT)</p> <p>Ограничение входного сигнала в заданном диапазоне. Блок предназначен для ограничения входного сигнала <b>In</b> в диапазоне [<b>Neg</b>, <b>Pos</b>]. Тип данных <b>In</b>, <b>Pos</b>, <b>Neg</b> и <b>Out</b> – float. Обязательные для инициализации входы ФБ: <b>In</b>, <b>Pos</b>, <b>Neg</b>.</p>
	Sign float	<p>Блок «Определение знака входного сигнала» (fSIGN)</p> <p>Определяет знак входного сигнала. Если входной сигнал равен 0, то выход <b>Out</b> = 0; Если входной сигнал положительный, то выход <b>Out</b> = 1; Если входной сигнал отрицательный, то выход <b>Out</b> = -1. Тип данных <b>In</b> – float, <b>Out</b> – int32. Обязательный для инициализации вход ФБ: <b>In</b>.</p>

Таблица 11. Блоки управления

Представление блока	Название	Описание блока
		<p>Блок «ПИД регулятор» (fPID)</p> <p>Блок пропорционально-интегрально-дифференциального (ПИД) регулятора с дополнительными настройками. <b>En</b> – Вход включения регулятора (0 - выключен, 1 - включен) <b>Reference</b> – Вход задания уставки <b>Feedback</b> – Сигнал обратной связи <b>Prop gain</b> – Пропорциональный коэффициент регулятора <b>Int gain</b> – Интегральный коэффициент регулятора <b>Deriv gain</b> – Дифференциальный</p>

Представление блока	Название	Описание блока
	<p>PID Controller float</p>	<p>коэффициент регулятора</p> <p><b>Lower lev</b> – Ограничение нижнего уровня выходного сигнала</p> <p><b>Upper lev</b> – Ограничение верхнего уровня выходного сигнала</p> <p><b>Out</b> – Выходное значение сигнала регулятора</p> <p><b>Previous output</b> – Предыдущее выходное значение сигнала регулятора</p> <p><b>Summa of errors</b> – Сумма интегральной ошибки разности сигналов (<b>Reference</b> – <b>Feedback</b>)</p> <p>Из сигнала задания (вход <b>Reference</b>) вычитается сигнал обратной связи (вход <b>Feedback</b>), полученная ошибка обрабатывается согласно алгоритму ПИД регулятора. Блок предназначен для динамической настройки параметров регулятора.</p> <p>Тип данных <b>En</b> – uint8, <b>Reference</b>, <b>Feedback</b>, <b>Prop gain</b>, <b>Int gain</b>, <b>Deriv gain</b>, <b>Lower lev</b>, <b>Upper lev</b>, <b>Out</b>, <b>Previous output</b>, <b>Summa of errors</b> – float.</p> <p>Обязательные для инициализации входы ФБ: <b>En</b>, <b>Reference</b>, <b>Feedback</b>, <b>Prop gain</b>, <b>Int gain</b>, <b>Deriv gain</b>, <b>Lower lev</b>, <b>Upper lev</b>.</p>
	<p>Pulse Width Modulator float</p>	<p>Блок «Широтно-импульсный модулятор» (fPWM)</p> <p>Генерирует на выходе последовательность периодических импульсов с заданным коэффициентом заполнения.</p> <p>При наличии разрешения <b>En=1</b> блок генерирует импульсный сигнал на выходе <b>Out</b> с коэффициентом заполнения, заданным на входе <b>Duty cycle</b> (изменяется в диапазоне от 0 до 100 %), и длительностью периода, заданной на входе <b>Period</b>.</p> <p>Блок выдает на выходе <b>Out</b> значение 0, если вход разрешения сброшен <b>En=0</b>.</p> <p>На выходе <b>End tau</b> выводится оставшееся время до окончания формирования импульса длительностью <math>\tau = \text{Duty cycle} \cdot \text{Period}</math>. На выходе <b>End period</b> выводится оставшееся время до окончания формирования периода. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>En</b>.</p>

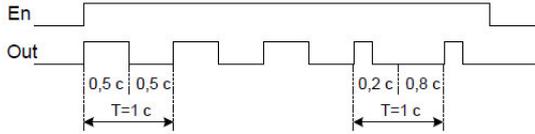
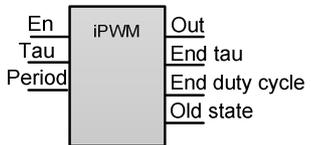
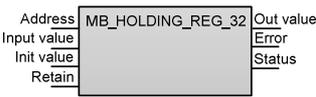
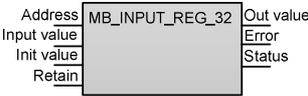
Представление блока	Название	Описание блока
		<p style="text-align: center;"><i>Временная диаграмма</i></p>  <p>Тип данных <b>En</b>, <b>Old state</b> – uint8, <b>Out</b> – bool, <b>Duty cycle</b> – float, <b>Period</b>, <b>End tau</b> и <b>End period</b> – uint32.</p> <p>Обязательные для инициализации входы ФБ: <b>En</b>, <b>Duty cycle</b>, <b>Period</b>.</p>
	<p>Pulse Width Modulator int</p>	<p style="text-align: center;"><b>Блок «Широтно-импульсный модулятор» (iPWM)</b></p> <p>Генерирует на выходе последовательность периодических импульсов с заданной скважностью.</p> <p>При наличии разрешения <b>En=1</b> блок генерирует импульсный сигнал на выходе <b>Out</b> с периодом, заданным на входе <b>Period</b>, и длительностью импульса, заданной на входе <b>Tau</b>.</p> <p>Блок выдает на выходе <b>Out</b> значение 0, если вход разрешения сброшен <b>En=0</b>.</p> <p>На выходе <b>End tau</b> выводится оставшееся время до окончания формирования импульса длительностью <b>Tau</b>. На выходе <b>End duty cycle</b> выводится оставшееся время до окончания формирования скважности, определяемой по формуле: <b>Duty cycle = Period – Tau</b>. Выход <b>Old state</b> запоминает предыдущее состояние входа <b>En</b>.</p> <p style="text-align: center;"><i>Временная диаграмма</i></p>  <p>Тип данных <b>En</b>, <b>Old state</b> – uint8, <b>Out</b> – bool, <b>Tau</b>, <b>Period</b>, <b>End tau</b> и <b>End duty cycle</b> – uint32.</p> <p>Обязательные для инициализации входы ФБ: <b>En</b>, <b>Tau</b>, <b>Period</b>.</p>

Таблица 12. Блоки Modbus

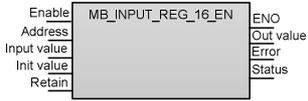
Представление блока	Название	Описание блока
	<p>Holding Register 16</p>	<p><b>Блок «Modbus Holding Register 16» (MB_HOLDING_REG_16)</b></p>

Представление блока	Название	Описание блока
		<p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 16-битовый тип данных (доступен для чтения и записи).</p> <p>Входы:  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus: 0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено. Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – uint16, <b>Retain</b>, <b>Error</b> – bool, <b>Status</b> – int16.</p> <p>Обязательные для инициализации входы ФБ: <b>Address</b>, <b>Input value</b>.</p>
	<p style="text-align: center;"><b>Holding Register 32</b></p>	<p style="text-align: center;"><b>Блок «Modbus Holding Register 32» (MB_HOLDING_REG_32)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 32-битовый тип данных (доступен для чтения и записи).</p> <p>Входы:  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p>

Представление блока	Название	Описание блока
		<p>Выходы:  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus:  0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено.  Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Retain</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.  Обязательные для инициализации входы ФБ: <b>Address</b>, <b>Input value</b>.</p>
	<p style="text-align: center;"><b>Блок «Modbus Input Register 16» (MB_INPUT_REG_16)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 16-битовый тип данных (доступен только для чтения).</p> <p>Входы:  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus:  0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено.  Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Retain</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.  Обязательные для инициализации входы ФБ: <b>Address</b>, <b>Input value</b>.</p>	<p style="text-align: center;"><b>Блок «Modbus Input Register 16» (MB_INPUT_REG_16)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 16-битовый тип данных (доступен только для чтения).</p> <p>Входы:  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus:  0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено.  Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Retain</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.  Обязательные для инициализации входы ФБ: <b>Address</b>, <b>Input value</b>.</p>

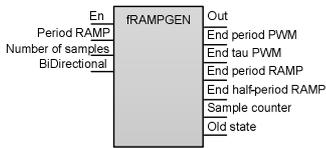
Представление блока	Название	Описание блока
	Input Register 32	<p align="center"><b>Блок «Modbus Input Register 32» (MB_INPUT_REG_32)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 32-битовый тип данных (доступен только для чтения).</p> <p><b>Входы:</b>  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p><b>Выходы:</b>  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus:  0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено.  Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Retain</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.  Обязательные для инициализации входы ФБ: <b>Address</b>, <b>Input value</b>.</p>
	Holding Register 16 En	<p align="center"><b>Блок «Modbus Holding Register 16 En» (MB_HOLDING_REG_16_EN)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus, используя 16-битовый тип данных (доступен для чтения и записи) с дополнительной функцией включения/выключения блока.</p> <p><b>Входы:</b>  <b>Enable</b> – Вход включения/выключения ФБ.  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход</p>

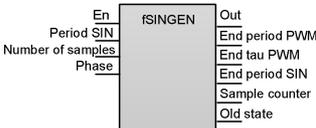
Представление блока	Название	Описание блока
		<p><b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>ENO</b> – Выход пропускает сигнал со входа <b>Enable</b>;  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus: 0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено. Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Enable</b>, <b>Retain</b>, <b>ENO</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.  Обязательные для инициализации входы ФБ: <b>Enable</b>, <b>Address</b>, <b>Input value</b>.</p>
	<p style="text-align: center;"><b>Holding Register 32 En</b></p>	<p style="text-align: center;"><b>Блок «Modbus Holding Register 32 En» (MB_HOLDING_REG_32_EN)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus RTU, используя 32-битовый тип данных (доступен для чтения и записи) с дополнительной функцией включения/выключения блока.</p> <p>Входы:  <b>Enable</b> – Вход включения/выключения ФБ.  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>ENO</b> – Выход пропускает сигнал со входа <b>Enable</b>;  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus: 0 – готов к подключению; 4 – соединение</p>

Представление блока	Название	Описание блока
		<p>установлено; 5 – соединение не установлено. Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Enable</b>, <b>Retain</b>, <b>ENO</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>Enable</b>, <b>Address</b>, <b>Input value</b>.</p>
	<p align="center"><b>Input Register 16 En</b></p>	<p align="center"><b>Блок «Modbus Input Register 16 En» (MB_INPUT_REG_16_EN)</b></p> <p>Блок осуществляет прием/передачу данных по протоколу Modbus RTU, используя 16-битовый тип данных (доступен только для чтения) с дополнительной функцией включения/ выключения блока.</p> <p><b>Входы:</b>  <b>Enable</b> – Вход включения/выключения ФБ.  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p><b>Выходы:</b>  <b>ENO</b> – Выход пропускает сигнал со входа <b>Enable</b>;  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus: 0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено. Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Enable</b>, <b>Retain</b>, <b>ENO</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.</p> <p>Обязательные для инициализации входы ФБ: <b>Enable</b>, <b>Address</b>, <b>Input value</b>.</p>
	<p align="center"><b>Input Register 32 En</b></p>	<p align="center"><b>Блок «Modbus Input Register 32 En» (MB_INPUT_REG_32_EN)</b></p> <p>Блок осуществляет прием/передачу данных</p>

Представление блока	Название	Описание блока
		<p>по протоколу Modbus RTU, используя 32-битовый тип данных (доступен только для чтения) с дополнительной функцией включения/выключения блока.</p> <p>Входы:  <b>Enable</b> – Вход включения/выключения ФБ.  <b>Address</b> – Адрес в памяти;  <b>Input value</b> – Вход данных;  <b>Init value</b> – Установка начального значения;  <b>Retain</b> – Установка сохранения данных в переменные VAR.</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:  <b>ENO</b> – Выход пропускает сигнал со входа <b>Enable</b>;  <b>Out Value</b> – Выходной сигнал данных;  <b>Error</b> – Сигнал об ошибке;  <b>Status</b> – Состояние соединения по Modbus: 0 – готов к подключению; 4 – соединение установлено; 5 – соединение не установлено. Значение выхода (из 4 в 5) меняется по истечении 2 мин.</p> <p>Тип данных <b>Address</b> – <b>uint16</b>, <b>Enable</b>, <b>Retain</b>, <b>ENO</b>, <b>Error</b> – <b>bool</b>, <b>Status</b> – <b>int16</b>.          Обязательные для инициализации входы ФБ: <b>Enable</b>, <b>Address</b>, <b>Input value</b>.</p>

Таблица 13. Блоки генераторов

Представление блока	Название	Описание блока
	Ramp Generator float	<p>Блок «Генератор треугольных импульсов» (fRAMPGEN)</p> <p>Блок генерирует треугольный (пилообразный) сигнал заданной частоты.</p> <p><b>En (uint8)</b> – Вход включения генератора (0 - выключен, 1 - включен)  <b>Period RAMP (uint32)</b> – Период треугольного сигнала (от 1 до 10000 мс)  <b>Numbers of samples (uint16)</b> – Количество отсчетов дискретизации (от 1 до 1000 единиц)  <b>BiDirectional (uint8)</b> – Вход задания вида</p>

Представление блока	Название	Описание блока
		<p>треугольного сигнала (0 - однонаправленный, 1 - двунаправленный)</p> <p><b>Out (bool)</b> – Выходной треугольный сигнал с заданными параметрами</p> <p><b>End period PWM (uint32)</b> – Окончание периода сигнала PWM (<b>End period PWM = Period RAMP / Number of samples</b>)</p> <p><b>End tau PWM (uint32)</b> – Окончание длительности сигнала PWM</p> <p><b>End period RAMP (uint32)</b> – Окончание периода треугольного сигнала</p> <p><b>End half-period RAMP (uint32)</b> – Окончание половины периода треугольного сигнала (для двунаправленного треугольного сигнала)</p> <p><b>Sample counter (uint8)</b> – Счетчик отсчетов дискретизации</p> <p><b>Old state (uint8)</b> – Сохранение предыдущего состояния входа <b>En</b></p> <p>Обязательные для инициализации входы ФБ: <b>En, Period RAMP, Number of samples, BiDirectional.</b></p>
	<p>Блок «Генератор прямоугольных импульсов» (<b>fRECTGEN</b>)</p> <p>Блок генерирует прямоугольные импульсы заданной длительности.</p> <p><b>En (uint8)</b> – Вход включения генератора (0 - выключен, 1 - включен)</p> <p><b>Period (uint32)</b> – Период прямоугольного импульса (от 1 до 10000 мс)</p> <p><b>Tau (uint32)</b> – Длительность прямоугольного импульса (от 1 до 10000 мс)</p> <p><b>Out (bool)</b> – Выходной прямоугольный импульс с заданными параметрами</p> <p><b>End period (uint32)</b> – Окончание периода прямоугольного импульса</p> <p><b>End tau (uint32)</b> – Окончание длительности прямоугольного импульса</p> <p><b>Old state (uint8)</b> – Сохранение предыдущего состояния входа <b>En</b></p> <p>Обязательные для инициализации входы ФБ: <b>En, Period, Tau.</b></p>	
	<p>Блок «Генератор синусоидального сигнала» (<b>fSINGEN</b>)</p> <p>Блок генерирует синусоидальный сигнал заданной частоты.</p> <p><b>En (uint8)</b> – Вход включения генератора (0 - выключен, 1 - включен)</p> <p><b>Period SIN (uint32)</b> – Период</p>	

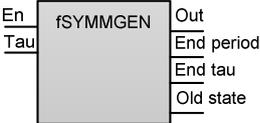
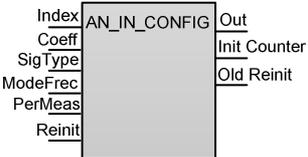
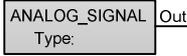
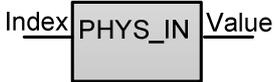
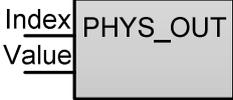
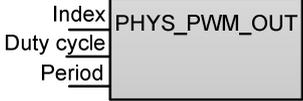
Представление блока	Название	Описание блока
		<p>синусоидального сигнала (от 1 до 10000 мс)  <b>Numbers of samples (uint16)</b> – Количество отсчетов дискретизации (от 1 до 1000 единиц)  <b>Phase (int16)</b> – Начальная фаза синусоидального сигнала  <b>Out (bool)</b> – Выходной синусоидальный сигнал с заданными параметрами  <b>End period PWM (uint32)</b> – Окончание периода сигнала PWM (<b>End period PWM = Period RAMP / Number of samples</b>)  <b>End tau PWM (uint32)</b> – Окончание длительности сигнала PWM  <b>End period SIN (uint32)</b> – Окончание периода синусоидального сигнала  <b>Sample counter (uint8)</b> – Счетчик отсчетов дискретизации  <b>Old state (uint8)</b> – Сохранение предыдущего состояния входа <b>En</b>            Обязательные для инициализации входы ФБ: <b>En, Period SIN, Number of samples, Phase.</b></p>
	<p align="center"><b>Симметричный генератор импульсов</b> Symmetrical Pulse Generator float</p>	<p align="center"><b>Блок «Генератор симметричных импульсов» (fSYMMGEN)</b></p> <p>Блок генерирует симметричные прямоугольные импульсы (<i>длительность импульса равна длительности паузы</i>) заданной длительности.  <b>En (uint8)</b> – Вход включения генератора (0 - выключен, 1 - включен)  <b>Tau (uint32)</b> – Длительность симметричного импульса (от 1 до 1000 мс)  <b>Out (bool)</b> – Последовательность симметричных прямоугольных импульсов с заданными параметрами  <b>End period (uint32)</b> – Окончание периода симметричного импульса  <b>End tau (uint32)</b> – Окончание длительности симметричного импульса  <b>Old state (uint8)</b> – Сохранение предыдущего состояния входа <b>En</b>            Обязательные для инициализации входы ФБ: <b>En, Tau.</b></p>

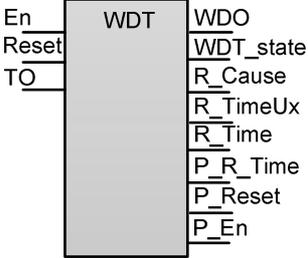
Таблица 14. Блоки аналоговых и дискретных входов, дискретных/ШИМ и релейных выходов, блок WDT, блоки даты и времени (RTC), блоки Modbus RTU

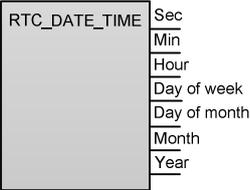
Представление блока	Название	Описание блока																																		
 <p>The diagram shows a rectangular block labeled 'AN_IN_CONFIG'. On the left side, there are input parameters: Index, Coeff, SigType, ModeFreq, PerMeas, and Reinit. On the right side, there are output parameters: Out, Init Counter, and Old Reinit.</p>	<p>Analogue Input Configuration</p>	<p>Блок «Аналоговый вход. Конфигурация» (AN_IN_CONFIG)</p> <p>Блок используется для съема и передачи аналоговых данных с аналоговых входов контроллера.</p> <p><b>Index (uint8)</b> – Номер аналогового входа контроллера (отсчет ведется с нуля; например, если номер физического аналогового входа контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p> <p><b>Coeff (float)</b> – Коэффициент фильтрации (диапазон задания параметра – 0,0001...1).</p> <p><b>SigType (uint8)</b> – Режим измерения.</p> <p>Для установки нужного значения для режима <b>SigType</b> можно воспользоваться следующими тремя блоками (тип режима устанавливается в соответствии со значениями, указанными в таблицах ниже):</p> <p>1. </p> <table border="1" data-bbox="847 1178 1485 1335"> <thead> <tr> <th>Значение</th> <th>Описание входа</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Вход отключен</td> </tr> <tr> <td>1</td> <td>Ток 0–20 мА</td> </tr> <tr> <td>2</td> <td>Напряжение 0–10 В</td> </tr> </tbody> </table> <p>2. </p> <table border="1" data-bbox="847 1467 1485 2141"> <thead> <tr> <th>Значение</th> <th>Описание входа</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>Термопара типа ТХА (К)</td> </tr> <tr> <td>4</td> <td>Термопара типа ТХА (К) с термокомпенсацией</td> </tr> <tr> <td>5</td> <td>Термопара типа ТХК (L)</td> </tr> <tr> <td>6</td> <td>Термопара типа ТХК (L) с термокомпенсацией</td> </tr> <tr> <td>7</td> <td>Термопара типа ТХКн (E)</td> </tr> <tr> <td>8</td> <td>Термопара типа ТХКн (E) с термокомпенсацией</td> </tr> <tr> <td>9</td> <td>Термопара типа ТПП10 (S)</td> </tr> <tr> <td>10</td> <td>Термопара типа ТПП10 (S) с термокомпенсацией</td> </tr> <tr> <td>11</td> <td>Термопара типа ТНН (N)</td> </tr> <tr> <td>12</td> <td>Термопара типа ТНН (N) с термокомпенсацией</td> </tr> <tr> <td>13</td> <td>Термопара типа ТПР (B)</td> </tr> <tr> <td>14</td> <td>Термопара типа ТПР (B) с</td> </tr> </tbody> </table>	Значение	Описание входа	0	Вход отключен	1	Ток 0–20 мА	2	Напряжение 0–10 В	Значение	Описание входа	3	Термопара типа ТХА (К)	4	Термопара типа ТХА (К) с термокомпенсацией	5	Термопара типа ТХК (L)	6	Термопара типа ТХК (L) с термокомпенсацией	7	Термопара типа ТХКн (E)	8	Термопара типа ТХКн (E) с термокомпенсацией	9	Термопара типа ТПП10 (S)	10	Термопара типа ТПП10 (S) с термокомпенсацией	11	Термопара типа ТНН (N)	12	Термопара типа ТНН (N) с термокомпенсацией	13	Термопара типа ТПР (B)	14	Термопара типа ТПР (B) с
Значение	Описание входа																																			
0	Вход отключен																																			
1	Ток 0–20 мА																																			
2	Напряжение 0–10 В																																			
Значение	Описание входа																																			
3	Термопара типа ТХА (К)																																			
4	Термопара типа ТХА (К) с термокомпенсацией																																			
5	Термопара типа ТХК (L)																																			
6	Термопара типа ТХК (L) с термокомпенсацией																																			
7	Термопара типа ТХКн (E)																																			
8	Термопара типа ТХКн (E) с термокомпенсацией																																			
9	Термопара типа ТПП10 (S)																																			
10	Термопара типа ТПП10 (S) с термокомпенсацией																																			
11	Термопара типа ТНН (N)																																			
12	Термопара типа ТНН (N) с термокомпенсацией																																			
13	Термопара типа ТПР (B)																																			
14	Термопара типа ТПР (B) с																																			

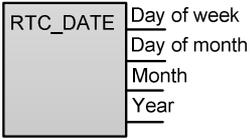
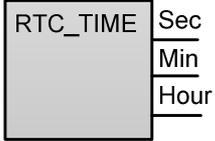
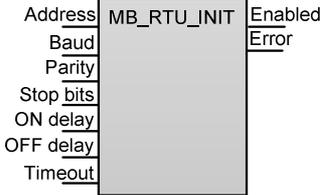
Представление блока	Название	Описание блока	
			термокомпенсацией
		15	Термопара типа ТЖК (J)
		16	Термопара типа ТЖК (J) с термокомпенсацией
		17	Термопара типа ТВР (А-1)
		18	Термопара типа ТВР (А-1) с термокомпенсацией
		19	Термопара типа ТПП13 (R)
		20	Термопара типа ТПП13 (R) с термокомпенсацией
		<div style="text-align: center;">  </div>	
		3.	
		Значение	Описание входа
		21	Термосопротивление в режиме трехпроводного подключения типа ТСМ 50М
		22	Термосопротивление в режиме трехпроводного подключения типа ТСМ 100М
		23	Термосопротивление в режиме трехпроводного подключения типа ТСМ 500М
		24	Термосопротивление в режиме трехпроводного подключения типа ТСП 50П
		25	Термосопротивление в режиме трехпроводного подключения типа ТСП 100П
		26	Термосопротивление в режиме трехпроводного подключения типа ТСП 500П
		27	Термосопротивление в режиме трехпроводного подключения типа ТСП 1000П
		28	Термосопротивление в режиме трехпроводного подключения типа ТСП Pt50
		29	Термосопротивление в режиме трехпроводного подключения типа ТСП Pt100
		30	Термосопротивление в режиме трехпроводного подключения типа ТСН 100Н
		31	Термосопротивление в режиме трехпроводного подключения типа ТСН 500Н
		32	Термосопротивление в режиме трехпроводного подключения типа ТСН 1000Н

Представление блока	Название	Описание блока
		<p><b>ModeFrec (uint8)</b> – Режим интегрирования. Диапазон задания параметра: 0 – без интегрирования в АЦП, 1 – с интегрированием в АЦП.</p> <p><b>PerMeas (uint16)</b> – Количество измерений.</p> <p><b>Reinit (bool)</b> – Переинициализация параметров аналоговых входов (по переднему фронту, т.е. при переключении с логического нуля на единицу).</p> <p><b>Out (float)</b> – Входной аналоговый сигнал на физическом аналоговом входе контроллера с номером (<b>Index</b> + 1) (в зависимости от типа измеряемой физической величины (напряжение <math>U</math>, ток <math>I</math> или температура <math>T</math>) единицей измерения является <math>B, mA</math> или <math>^{\circ}C</math>).</p> <p><b>Init Counter (uint8)</b> – Счетчик количества инициализаций параметров аналоговых входов.</p> <p><b>Old Reinit (uint8)</b> – Предыдущее состояние входа переинициализаций параметров аналоговых входов.</p> <p>Обязательные для инициализации входы ФБ: <b>Index</b>, <b>Coeff</b>, <b>SigType</b>, <b>ModeFrec</b>, <b>PerMeas</b>.</p>
	<p>Аналог Input</p>	<p>Блок «Аналоговый вход» (AN_IN)</p> <p>Блок используется совместно с блоком «Аналоговый вход. Конфигурация» (AN_IN_CONFIG) для удобства разработки проекта. Например, в случае большого проекта (большого количества используемых ФБ) достаточно установить один блок AN_IN_CONFIG и проинициализировать параметры нужного аналогового входа. Блок «Аналоговый вход» (AN_IN) является дубликатом блока AN_IN_CONFIG в том смысле, что на его выходе можно снимать те же числовые данные, что и с блока AN_IN_CONFIG. При этом блок AN_IN не требует повторной инициализации параметров заданного аналогового входа. Для его работы достаточно указать только номер (<b>Index</b>) заданного аналогового входа. При использовании блока AN_IN сокращается не только время разработки проекта, но и улучшается его читаемость.</p> <p><b>Index (uint8)</b> – Номер аналогового входа контроллера (отсчет ведется с нуля; например, если номер физического аналогового входа контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p>

Представление блока	Название	Описание блока
		<p><b>Out (float)</b> – Входной аналоговый сигнал на физическом аналоговом входе контроллера с номером (<b>Index</b> + 1) (в зависимости от типа измеряемой физической величины (напряжение <math>U</math>, ток <math>I</math> или температура <math>T</math>) единицей измерения является <math>B</math>, <math>mA</math> или <math>^{\circ}C</math>).</p> <p>Обязательный для инициализации вход ФБ: <b>Index</b>.</p>
	Physical Input	<p>Блок «<b>Физический (дискретный) вход</b>» (<b>PHYS_IN</b>)</p> <p>Блок используется для съема и передачи дискретных данных: съем логических «0» (от -3 до 5 В DC) или «1» (от 15 до 30 В DC) с дискретных входов контроллера.</p> <p><b>Index (uint8)</b> – Номер дискретного входа контроллера (отсчет ведется с нуля; например, если номер физического дискретного входа контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p> <p><b>Value (bool)</b> – Входной дискретный сигнал на физическом дискретном входе контроллера с номером (<b>Index</b> + 1).</p> <p>Обязательный для инициализации вход ФБ: <b>Index</b>.</p>
	Physical Output	<p>Блок «<b>Физический (дискретный) выход</b>» (<b>PHYS_OUT</b>)</p> <p>Блок используется для управления дискретными выходами контроллера: подача логических «0» или «1» (максимальное коммутируемое напряжение 30 В DC) на дискретные выхода контроллера.</p> <p><b>Index (uint8)</b> – Номер дискретного выхода контроллера (отсчет ведется с нуля; например, если номер физического дискретного выхода контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p> <p><b>Value (bool)</b> – Выходной дискретный сигнал на физическом дискретном выходе контроллера с номером (<b>Index</b> + 1).</p> <p>Обязательные для инициализации входы ФБ: <b>Index</b>, <b>Value</b>.</p>
	Physical PWM Output	<p>Блок «<b>ШИМ выход</b>» (<b>PHYS_PWM_OUT</b>)</p> <p>Блок используется для подачи аппаратно-реализованного ШИМ-сигнала на дискретный выход контроллера.</p> <p><b>Index (uint8)</b> – Номер дискретного выхода контроллера (отсчет ведется с нуля;</p>

Представление блока	Название	Описание блока
		<p>например, если номер физического дискретного выхода контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p> <p><b>Duty cycle (float)</b> – Коэффициент заполнения периода сигналом (изменяется в диапазоне от 0 до 100 %).</p> <p><b>Period (uint16)</b> – длительность периода ШИМ-сигнала.</p> <p>Обязательные для инициализации входы ФБ: <b>Index, Duty cycle, Period</b>.</p>
	<p>Relay Output</p>	<p>Блок «Релейный выход» (<b>REL_OUT</b>)</p> <p>Блок используется для управления релейными выходами контроллера (максимальное коммутируемое напряжение 250 В AC).</p> <p><b>Index (uint8)</b> – Номер релейного выхода контроллера (отсчет ведется с нуля; например, если номер физического релейного выхода контроллера = 1, то на вход <b>Index</b> нужно подать 0).</p> <p><b>Value (bool)</b> – Выходной сигнал на физическом релейном выходе контроллера с номером (<b>Index + 1</b>).</p> <p>Обязательные для инициализации входы ФБ: <b>Index, Value</b>.</p>
	<p>Software Watchdog Timer</p>	<p>Блок «Сторожевой таймер» (<b>WDT</b>)</p> <p>Блок представляет собой аппаратно реализованную схему контроля над зависанием системы (т.е. таймер, который периодически сбрасывается контролируемой системой). Если сброса не произошло в течение некоторого интервала времени, происходит принудительная перезагрузка системы.</p> <p>Контроллер перезапускается, если <b>WDT</b> включен (перемычка не установлена).</p> <p>Входы:</p> <p><b>En (bool)</b> – Вход управляет включением / выключением блока. Вход работает по переднему фронту сигнала.</p> <p><b>Reset (bool)</b> – Вход получает сигналы от системы, поступающие на него с заданным периодом для сброса таймера. Период следования сигналов должен быть меньше времени, установленного на входе <b>TO</b>.</p> <p><b>TO (uint32)</b> – Вход позволяет установить время, по истечении которого происходит принудительная перезагрузка системы при том условии, что не произошло сброса</p>

Представление блока	Название	Описание блока
		<p>таймера.</p> <p>Выходы:</p> <p><b>WDO (bool)</b> – Выход устанавливает логическую единицу по истечении времени, установленного на входе <b>TO</b>, при этом вход <b>Reset</b> не был задействован.</p> <p><b>WDT_state (bool)</b> – Выход показывает состояние аппаратного WDT. Если WDT включен (переключатель не установлен), то на выходе «1», иначе «0».</p> <p><b>R_Cause (uint8)</b> – Выход устанавливает «1», если произошла перезагрузка модуля по истечении времени <b>TO</b>. Значение выхода сохраняется в EEPROM. Выход сбрасывается в «0», при поступлении переднего фронта сигнала на вход <b>En</b>.</p> <p><b>R_TimeUx (uint32)</b> – Выход показывает текущее время (unixtime из RTC), при котором истекло время <b>TO</b>.</p> <p><b>R_Time (uint32)</b> – Выход показывает период времени с начала работы задачи.</p> <p><b>P_R_Time (uint32)</b> – Выход показывает время, когда был получен последний передний фронт на входе <b>Reset</b> (внутренняя служебная переменная).</p> <p><b>P_Reset (uint8)</b> – Выход показывает предыдущее значение входа <b>Reset</b> (внутренняя служебная переменная).</p> <p><b>P_En (uint8)</b> – Выход показывает предыдущее значение входа <b>En</b> (внутренняя служебная переменная).</p> <p>Обязательные для инициализации входы ФБ: <b>En, Reset, TO</b>.</p>
	<p>RTC Date Time</p>	<p>Блок «Дата и время DATE &amp; TIME» (RTC_DATE_TIME)</p> <p>Блок используется для определения текущей даты и реального времени.</p> <p>На следующие выходы блока поступают:</p> <p><b>Sec (int32)</b> – Секунды.</p> <p><b>Min (int32)</b> – Минуты.</p> <p><b>Hour (int32)</b> – Часы.</p> <p><b>Day of week (int32)</b> – Дни недели.</p> <p><b>Day of month (int32)</b> – Дни месяца.</p> <p><b>Month (int32)</b> – Месяцы.</p> <p><b>Year (int32)</b> – Года.</p>

Представление блока	Название	Описание блока
	RTC Date	<p>Блок «Дата DATE» (RTC_DATE)</p> <p>Блок используется для определения текущей даты.</p> <p>На следующие выходы блока поступают:</p> <p><b>Day of week (int32)</b> – Дни недели.  <b>Day of month (int32)</b> – Дни месяца.  <b>Month (int32)</b> – Месяцы.  <b>Year (int32)</b> – Годы.</p>
	RTC Time	<p>Блок «Время TIME» (RTC_TIME)</p> <p>Блок используется для определения реального времени.</p> <p>На следующие выходы блока поступают:</p> <p><b>Sec (int32)</b> – Секунды.  <b>Min (int32)</b> – Минуты.  <b>Hour (int32)</b> – Часы.</p>
	MB RTU Init	<p>Блок «Modbus RTU Initialization» (MB_RTU_INIT)</p> <p>Блок инициализирует Modbus RTU с параметрами, установленными на входе блока.</p> <p>Входы:</p> <p><b>Address</b> – адрес станции;  <b>Baud</b> – скорость передачи данных;  <b>Parity</b> – чётность;  <b>Stop bits</b> – количество стоп-битов;  <b>ON delay</b> – преамбула (мс);  <b>OFF delay</b> – постамбула (мс);  <b>Timeout</b> – таймаут (сек).</p> <p><i>Примечание:</i></p> <ol style="list-style-type: none"> <li>Baud: <ul style="list-style-type: none"> <li>5 – 9600 бит/сек;</li> <li>6 – 19200 бит/сек;</li> <li>7 – 28800 бит/сек;</li> <li>8 – 38400 бит/сек;</li> <li>9 – 57600 бит/сек;</li> <li>10 – 115200 бит/сек.</li> </ul> </li> <li>Parity: 0 – NOPARITY; <ul style="list-style-type: none"> <li>1 – ODD;</li> <li>2 – EVEN;</li> <li>3 – всегда 0;</li> <li>4 – всегда 1.</li> </ul> </li> <li>Stop bits: 0 – 1; <ul style="list-style-type: none"> <li>1 – 1,5;</li> <li>2 – 2.</li> </ul> </li> </ol> <p>Выходы:</p>

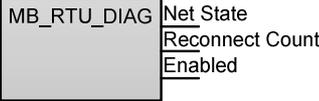
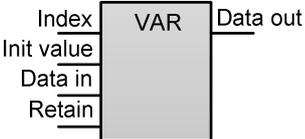
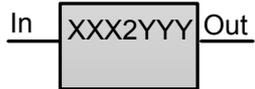
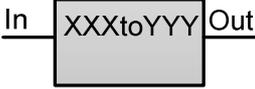
Представление блока	Название	Описание блока
		<p><b>Enabled</b> – Выдает сигнал о состоянии блока (включен/выключен);</p> <p><b>Error</b> – Сигнал об ошибке.</p> <p>Тип данных <b>Address, Baud, Parity, Stop bits, Error</b> – uint8, <b>ON delay, OFF delay</b> – int32, <b>Timeout</b> – uint32, <b>Enabled</b> – bool.</p> <p>Обязательные для инициализации входы ФБ: <b>Address, Baud, Parity, Stop bits, ON delay, OFF delay, Timeout</b>.</p>
	MB RTU Diag	<p>Блок «<b>Modbus RTU Diagnostics</b>» (<b>MB_RTU_DIAG</b>)</p> <p>Блок определяет состояние соединения.</p> <p>Выходы:</p> <p><b>Net State</b> – Выдает сигнал о состоянии сети (0 - если мастер установил соединение, 1 - соединение разорвано или мастер прислал ошибочный запрос);</p> <p><b>Reconnect Count</b> – Счетчик потери связи. Инкрементируется после разрыва связи с мастером по истечении времени Timeout ФБ MB_RTU_INIT. Если по старту программы связь была сразу установлена, то =0, иначе =1;</p> <p><b>Enabled</b> – На выходе устанавливается признак того, что данный ФБ обрабатывается =1.</p> <p>Тип данных <b>Net State, Enabled</b> – bool, <b>Reconnect Count</b> – uint32.</p>

Таблица 15. Блоки переменных и констант

Представление блока	Название	Описание блока
	Constant	<p>Блок «<b>Константы</b>» (<b>CONST</b>)</p> <p>Блок используется для задания любых постоянных величин (любого типа данных из восьми предложенных) на входах ФБ.</p> <p><b>Out</b> – Заданная постоянная величина.</p>
	Variable	<p>Блок «<b>Переменные</b>» (<b>VAR</b>)</p> <p>Блок используется для передачи данных из одной части проекта в другую часть без использования соединительных линий.</p> <p>Входы:</p> <p><b>Index</b> – Номер переменной;</p> <p><b>Init value</b> – Установка начального значения;</p>

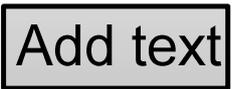
Представление блока	Название	Описание блока
		<p><b>Data in</b> – Вход блока переменных, на который поступают необходимые для передачи данные (данный вход используется только в случае передачи данных);</p> <p><b>Retain</b> – Установка сохранения данных в переменные VAR;</p> <p><i>Примечание:</i> Если на вход <b>Init value</b> задается первоначальное значение и вход <b>Retain</b> имеет значение <b>true</b>, то при выполнении программы сначала будет использовано значение со входа <b>Init value</b>, а затем будут использоваться только значения, сохраненные в переменной VAR.</p> <p>Выходы:</p> <p><b>Data out</b> – Выход блока переменных, с которого считываются переданные данные (данный выход используется только в случае приема данных).</p> <p><i>Примечание:</i> Одновременно вход <b>Data in</b> и выход <b>Data out</b> в одном ФБ не используются. Номер, указанный на входе <b>Index</b> блока, передающего данные, должен совпадать с номером на входе <b>Index</b> блока, принимающего данные.</p> <p>Тип данных <b>Index</b> – <b>uint16</b>, <b>Retain</b> – <b>bool</b>. Обязательные для инициализации входы ФБ: <b>Index</b>, <b>Data in</b>.</p>

Таблица 16. Блоки преобразования типов

Представление блока	Название	Описание блока
	xxx2yyy	<p>Операция «Приведение типов данных <b>XXX</b> к <b>YYY</b>» (<b>XXX2YYY</b>)</p> <p>Блок используется для приведения типа данных <b>XXX</b> к типу данных <b>YYY</b>. Например, операция <b>float2int</b> (блок <b>FLOAT2INT</b>).</p> <p><b>In</b> – На вход поступают данные типа <b>XXX</b>. <b>Out</b> – С выхода идут данные типа <b>YYY</b>. Обязательный для инициализации вход ФБ: <b>In</b>.</p>
	xxx to yyy	<p>Операция «Приведение типов данных <b>XXX</b> к <b>YYY</b>» (<b>XXXtoYYY</b>)</p> <p>Блок используется для приведения типа данных <b>XXX</b> к типу данных <b>YYY</b>. Например, операция <b>uint32 to int8</b> (блок <b>UINT32toINT8</b>).</p> <p><b>In</b> – На вход поступают данные типа <b>XXX</b>.</p>

Представление блока	Название	Описание блока
		<b>Out</b> – С выхода идут данные типа YYY. Обязательный для инициализации вход ФБ: <b>In</b> .

Таблица 17. Блок комментариев

Представление блока	Название	Описание блока
	Comment	Блок «Комментарий» (Comment)  Блок используется для добавления комментария на поле проекта. Двойное нажатие ЛКМ по блоку позволяет изменить текст комментария.

## 8 Горячие клавиши

Горячие клавиши, используемые при работе в системе BitLogic, приведены в списке ниже.

### *Файл*

[Ctrl]+[N] – Создать новый проект

[Ctrl]+[O] – Открыть проект

[Ctrl]+[S] – Сохранить проект

[Ctrl]+[W] – Закрыть проект

[Ctrl]+[Shift]+[W] – Закрыть все проекты

[Alt]+[F7] – Настройки системы BitLogic

### *Правка*

[Ctrl]+[Z] – Отменить действие

[Ctrl]+[Y] – Повторить действие

[Ctrl]+[X] – Вырезать ФБ

[Ctrl]+[C] – Копировать ФБ

[Ctrl]+[V] – Вставить ФБ

### *Помощь*

[F1] – Открыть руководство

### *Панель инструментов*

[Ctrl]+[R] – Подключиться к контроллеру

[Ctrl]+[Break] – Отключиться от контроллера

[Ctrl]+[D] – Загрузить проект в ОЗУ контроллера

[F5] – Запустить выполнение проекта

[Ctrl]+[F5] – Остановить выполнение проекта

[F8] – Сохранить проект во флэш-памяти контроллера

[Alt]+[F12] – Настройки контроллера

## 9 Сообщения об ошибках и предупреждениях

Область видимости ошибки / предупреждения	Ошибка/предупреждение		Значение
	английский вариант	русский вариант	
Главное окно	It is impossible to open the manual file.	Невозможно открыть файл руководства.	Файл руководства пользователя недоступен
Сцена проекта	Link error: The input has been already connected.	Ошибка соединения: Данный вход уже подключен.	К одному входу ФБ может быть подключено только одно соединение
	Link error: It is impossible to connect an input to other one.	Ошибка соединения: Нельзя подключить вход к входу.	Неверное направление передачи данных
	Link error: It is impossible to connect an output to other one.	Ошибка соединения: Нельзя подключить выход к выходу.	Неверное направление передачи данных
Проект	It is impossible to read file.	Невозможно прочитать файл.	Файл недоступен или поврежден
	It is impossible to write file.	Невозможно записать файл.	Файл недоступен или поврежден
Соединение по TCP	Running task: Success.	Запуск задачи: Успех.	
	Running task: Failed.	Запуск задачи: Сбой.	
	Stopping task: Failed.	Остановка задачи: Сбой.	
	Stopping task: Success.	Остановка задачи: Успех.	
	Loading in RAM: Failed.	Загрузка в ОЗУ: Сбой.	
	Loading in RAM: Success.	Загрузка в ОЗУ: Успех.	
	Uploading task: Failed.	Загрузка задачи: Сбой.	
	Initializing the module: Failed.	Инициализация модуля: Сбой.	
	Initializing the module: Success.	Инициализация модуля: Успех.	
	Saving task: Failed.	Сохранение задачи: Сбой.	
	Saving task: Success.	Сохранение задачи: Успех.	
	Getting of net parameters: Failed	Получение сетевых параметров: Сбой.	
	Getting of net parameters: Success	Получение сетевых параметров: Успех.	
Setting of net	Установка сетевых		

parameters: Failed.	параметров: Сбой.	
Setting of net parameters: Success.	Установка сетевых параметров: Успех.	
Connection error.	Ошибка соединения.	
No connection.	Нет соединения.	
Search by the host name.	Поиск по имени хоста.	
Establishing connection.	Установка соединения.	
A connection is established.	Соединение установлено.	
The connection is about to close.	Соединение закрывается.	
State code	Код состояния	
Connection	Соединение	
Repeat	Повтор	
ERROR	ОШИБКА	
READY. command	ГОТОВ. команда:	
Getting of function block info.	Получение информации ФБ.	
SUCCESS. command:	УСПЕХ. команда:	
UNKNOWN COMMAND code:	НЕИЗВЕСТНАЯ КОМАНДА код:	
Free space command		
Net params command.		
Busy command.		
Device response.	Ответ устройства.	
There is no free space in the module.	В модуле нет свободного места.	
Exceeding of the package size.	Превышение размера пакета.	
The program is not stopped.	Программа не остановлена.	
Invalid data size in the package.	Недопустимый размер данных в пакете.	
Wrong function block index.	Неверный индекс функционального блока.	
Error saving to flash memory.	Ошибка сохранения во флэш-память.	
Wrong password.	Неверный пароль.	
No task in flash memory.	Нет задачи во флэш-памяти.	
Flash memory is corrupted.	Флэш-память повреждена.	

The data size is larger than the data buffer.	Размер данных больше, чем буфер данных.	
Invalid state.	Недопустимое состояние.	
A task in flash memory exists.	Существует задача во флэш-памяти.	
CRC error.	Ошибка CRC.	
Invalid data.	Неверные данные.	
No error.	Нет ошибок.	
The programming mode is disabled.	Режим программирования отключен.	
Unknown error.	Неизвестная ошибка.	
Loading in RAM memory.	Загрузка в ОЗУ.	
Save to flash.	Сохранение в флэш.	
Stopping of task.	Остановка задачи.	
Stopping of function block.	Остановка ФБ.	
Step.	Шаг.	
Setting of task state.	Установка состояния задачи.	
Run from flash.	Запустить из флэш.	
Reading and uploading data.	Чтение и загрузка данных.	
Clearing of data.	Очистка данных.	
Getting of function block data.	Получение данных ФБ.	
Setting of function block data.	Установка данных ФБ.	
Getting of net parameters.	Получение сетевых параметров.	
Setting of net parameters.	Установка сетевых параметров.	
Setting of password.	Установка пароля.	
Debugging mode.	Режим отладки.	
Echo.	Эхо.	
Uploading of function block.	Загрузка ФБ.	
Uploading of function block data.	Загрузка данных ФБ.	
Getting of data MB_HR.	Получение данных MB_HR.	
Getting of MB connection status.	Получение статуса соединения MB.	
Getting of TCP port.	Получение TCP порта.	
Getting of soft		

	data.		
	Getting of hardware ID.		
	Getting of MB port.		
	User comments.		
	Undefined command.	Неопределенная команда.	
Библиотека ФБ	The file is corrupted or is not a library file of function blocks.	Файл поврежден или не является файлом библиотеки функциональных блоков.	



