ЦИФРОВЫЕ
РЕШЕНИЯ

Owen Logic (версия 3.2)

Руководство пользователя

09.2026
версия 2.10

Содержание

1 О программе ... 6
1.1 Системные требования для работы Owen Logic ... 6
1.2 Используемые аббревиатуры и термины ... 6
2 Установка ... 8
3 Описание интерфейса ... 11
3.1 Главное меню ... 11
3.2 Панели инструментов ... 13
3.3 Панель Библиотека компонентов ... 14
3.4 Панель Свойства ... 16
3.5 Панель переменных ... 16
3.6 Рабочая область ... 18
3.7 Строка состояния ... 20
3.8 Менеджер и редактор экранов ... 21
4 Базовые сведения ... 23
4.1 Принцип выполнения программы ... 23
4.2 Разработка проекта и порядок работы ... 23
4.2.1 Автосохранение проекта ... 25
4.2.2 Защита проекта ключом ... 26
4.3 Разработка программы ... 28
4.3.1 Блок комментариев ... 31
4.3.2 Блок переменной ... 32
4.3.3 Блок константы ... 36
4.3.4 Линия задержки ... 37
4.3.5 Блок сетевой переменной ... 38
4.3.6 Блок чтения/записи в ФБ ... 40
4.3.7 Блок преобразования ... 41
4.3.8 Перенумеровать компоненты ... 42
4.3.9 Порядок исполнения ... 43
4.4 Программирование дисплея прибора ... 44
4.4.1 Монохромный текстовый ЖКИ ... 46
4.4.2 Графический цветной ЖКИ ... 50
4.5 Режим симулятора ... 58
4.5.1 Симуляция визуализации ... 62
4.5.2 Симуляция ST кода ... 63
4.6 Настройка порта и подключение прибора ... 67
4.7 Запись программы в прибор ... 70
4.8 Онлайн отладка ... 71
4.9 Сведения о проекте ... 75
4.10 Менеджер компонентов ... 77
4.11 Работа с макросами ... 80
4.12 Работа с элементами на языке ST ... 90
4.12.1 Работа с функциями на языке ST ... 90
4.12.2 Работа с функциональными блоками на языке ST ... 95
5 Настройка прибора ... 99
5.1 Экран ... 100
5.2 Часы ... 101
5.3 Обмен по сети ... 102
5.3.1 Интерфейсы ... 102
5.3.1.1 RS-485 ... 104
5.3.1.2 Ethernet ... 105
5.3.1.3 LoRa ... 106
5.3.2 Работа по протоколу Modbus ... 110
5.3.2.1 Режим Master ... 115
5.3.2.2 Режим Slave ... 122
5.3.2.3 Шаблоны сетевых устройств ... 126

5.4 Модули расширения ... 128
5.5 Входы и выходы ... 131
5.6 Пароль ... 133
5.7 Подключение к OwenCloud ... 135
6 Переменные ... 137
6.1 Типы переменных ... 143
6.2 Сервисные переменные ... 144
6.3 Переменные интерфейса связи ... 145
6.4 Привязка переменных к параметрам ... 146
6.5 Копирование переменных ... 147
7 Библиотека ... 148
7.1 Функции ... 148
7.1.1 Логические функции ... 148
7.1.1.1 И (AND) ... 149
7.1.1.2 ИЛИ (OR) ... 149
7.1.1.3 НЕ (NOT) ... 150
7.1.1.4 Исключающее ИЛИ (XOR) ... 151
7.1.2 Арифметические функции ... 151
7.1.2.1 Сложение (ADD, fADD) ... 152
7.1.2.2 Вычитание (SUB, fSUB) ... 152
7.1.2.3 Умножение (MUL, fMUL) ... 153
7.1.2.4 Деление (DIV, fDIV) ... 154
7.1.2.5 Деление с остатком (MOD) ... 154
7.1.2.6 Возведение числа в степень (fPOW) ... 155
7.1.2.7 Взятие модуля от числа (fABS) ... 155
7.1.3 Функции сравнения ... 155
7.1.3.1 Равно (EQ) ... 156
7.1.3.2 Больше (GT, fGT) ... 156
7.1.3.3 Выбор (SEL, fSEL) ... 157
7.1.4 Сдвиговые функции ... 158
7.1.4.1 Побитовый логический сдвиг влево (SHL) ... 158
7.1.4.2 Побитовый логический сдвиг вправо (SHR) ... 159
7.1.5 Битовые функции ... 159
7.1.5.1 Чтение бита (EXTRACT) ... 159
7.1.5.2 Запись бита (PUTBIT) ... 160
7.1.5.3 Дешифратор (DC32) ... 160
7.1.5.4 Шифратор (CD32) ... 161
7.2 Функциональные блоки ... 162
7.2.1 Триггеры ... 162
7.2.1.1 RS-триггер с приоритетом выключения ... 162
7.2.1.2 SR-триггер с приоритетом включения ... 163
7.2.1.3 Детектор переднего фронта импульса (RTRIG) ... 163
7.2.1.4 Детектор заднего фронта импульса (FTRIG) ... 164
7.2.1.5 D-триггер (DTRIG) ... 164
7.2.2 Таймеры ... 165
7.2.2.1 Импульс включения заданной длительности (TP) ... 165
7.2.2.2 Таймер с задержкой включения (TON) ... 166
7.2.2.3 Таймер с задержкой отключения (TOF) ... 167
7.2.2.4 Интервальный таймер (CLOCK) ... 169
7.2.2.5 Интервальный таймер с недельным циклом (CLOCKWEEK) ... 169
7.2.3 Генераторы ... 170
7.2.3.1 Генератор прямоугольных импульсов (BLINK) ... 171
7.2.4 Счетчики ... 172
7.2.4.1 Инкрементный счетчик с автосбросом (CT) ... 172
7.2.4.2 Универсальный счетчик (CTN) ... 173
7.2.4.3 Инкрементный счетчик (CTU) ... 175
7.2.5 Регуляторы ... 176
7.2.5.1 ПИД-регулятор для приборов первого поколения (PID) ... 176
7.2.5.2 ПИД-регулятор для приборов второго поколения (PID REG) ... 179
7.3 Макросы проекта ... 184
7.4 Функция на ST ... 185

7.5 Функциональный блок на ST ... 188
7.6 Элементы управления ... 191
7.6.1 Метка ... 191
7.6.2 Ввод/вывод (int/float) ... 193
7.6.3 Ввод/вывод (boolean) ... 195
7.6.4 Динамический текст ... 197
7.6.5 Комбинированный список (ComboBox) ... 199
7.7 Базовые элементы ... 201
7.7.1 Текст ... 209
7.7.2 Индикатор ... 211
7.7.3 Индикатор выполнения ... 212
7.7.4 Динамический текст ... 213
7.7.5 Ввод/вывод int/float ... 215
7.7.6 Ввод/вывод времени ... 219
7.7.7 Ввод/вывод IP ... 221
7.7.8 Линия ... 223
7.7.9 Многоугольник ... 224
7.7.10 Круг ... 225
7.7.11 Кнопка ... 227
7.7.12 Переключатель ... 228
7.7.13 Группа переключателей ... 230
7.7.14 Изображение ... 233
7.7.15 Меню ... 236
7.7.16 График ... 238
7.7.17 Стрелочный индикатор ... 242
8 Работа с прибором ... 247
8.1 Информация о приборе ... 247
8.2 Время цикла ... 248
8.3 Обновление встроенного ПО ... 249
8.4 Юстировка ... 251
8.4.1 Юстировка входов ... 251
8.4.2 Юстировка выходов ... 255
8.5 Смена целевой платформы ... 256
9 Расширения ... 258
9.1 Мастер тиражирования ... 259
9.1.1 Запись программы в прибор через мастер тиражирования на ОС Windows ... 263
9.1.2 Запись программы в прибор через мастер тиражирования на ОС Linux ... 266
9.2 Экспорт устройства в OPC ... 267
9.3 Экспорт устройства в OwenCloud ... 267
10 Сочетания клавиш ... 270
11 Примеры проектов ... 272
11.1 Вывод значения переменной на экран ... 272
11.2 Задание уставок таймеров с экрана ... 274
11.3 Подключение ПР200 к OwenCloud через шлюз ПМ210 по протоколу Modbus RTU ... 276
11.4 Прямое подключение приборов второго поколения к OwenCloud ... 280
11.5 Упаковка/распаковка бит/целых чисел ... 285
11.6 Обработка битовых переменных с помощью маски ... 286
12 Язык программирования ST ... 287
12.1 Синтаксис ... 287
12.1.1 Побитовый доступ к переменным ... 287
12.1.2 Использование функций в других элементах программы на ST ... 287
12.1.3 Использование одного функционального блока в другом ... 288
12.1.4 Комментирование в редакторе ST ... 289
12.1.5 Копирование элементов на ST между проектами ... 289
12.2 Документирование в редакторе ST ... 290

12.3 Типы данных ... 292
12.3.1 Зарезервированные ключевые слова ... 293
12.3.2 Массивы ... 293
12.4 Конструкции языка ... 295
12.4.1 Операции ... 296
12.4.1.1 Арифметические операции ... 296
12.4.1.2 Битовые операции ... 296
12.4.1.3 Операции преобразования типов данных ... 296
12.4.1.4 Логические операции ... 297
12.4.1.5 Операции сравнения ... 297
12.4.1.6 Приоритеты операций ... 297
12.4.2 Присвоение ... 298
12.4.3 Оператор IF ... 299
12.4.4 Оператор CASE ... 300
12.4.5 Оператор RETURN ... 300
12.4.6 Цикл FOR ... 301
12.4.7 Цикл WHILE ... 302
12.4.8 Цикл REPEAT UNTIL ... 302
12.5 Системные функции ... 303
12.5.1 Функции работы с датой и временем ... 303
12.5.1.1 Функция GET_TIME ... 303
12.5.1.2 Функция GET_DATE_TIME ... 303
12.5.1.3 Функция SYS.COMPARE_DATE_TIME ... 303
12.5.1.4 Функция SYS.IS_LEAP_YEAR ... 304
12.5.2 Математические функции ... 304
12.5.2.1 Математические константы (CONST.*) ... 304
12.5.2.2 MIN/FMIN ... 305
12.5.2.3 MAX/FMAX ... 305
12.5.2.4 SQRT ... 305
12.5.2.5 EXP ... 305
12.5.2.6 FACT ... 306
12.5.3 Логарифмические функции ... 306
12.5.3.1 LOG ... 306
12.5.3.2 LG ... 307
12.5.3.3 LN ... 307
12.5.4 Тригонометрические функции ... 307
12.5.4.1 SIN ... 308
12.5.4.2 COS ... 308
12.5.4.3 TAN ... 308
12.5.4.4 ASIN ... 308
12.5.4.5 ACOS ... 308
12.5.4.6 ATAN ... 309
12.6 Системные функциональные блоки ... 309
12.6.1 Триггеры ... 310
12.6.1.1 RS-триггер с приоритетом выключения (SYS.RS) ... 310
12.6.1.2 SR-триггер с приоритетом включения (SYS.SR) ... 310
12.6.1.3 Детектор переднего фронта импульса (SYS.RTRIG) ... 311
12.6.1.4 Детектор заднего фронта импульса (SYS.FTRIG) ... 311
12.6.1.5 D-триггер (SYS.DTRIG) ... 312
12.6.2 Таймеры ... 313
12.6.2.1 Импульс включения заданной длительности (SYS.TP) ... 313
12.6.2.2 Таймер с задержкой включения (SYS.TON) ... 314
12.6.2.3 Таймер с задержкой выключения (SYS.TOF) ... 315
12.6.2.4 Интервальный таймер (SYS.CLOCK) ... 316
12.6.2.5 Интервальный таймер с недельным циклом (SYS.CLOCKWEEK) ... 317
12.6.3 Генераторы ... 318
12.6.3.1 Генератор прямоугольных импульсов (SYS.BLINK) ... 318
12.6.4 Счетчики ... 319
12.6.4.1 Инкрементный счетчик с автосбросом (SYS.CT) ... 319
12.6.4.2 Универсальный счетчик (SYS.CTN) ... 320
12.6.4.3 Инкрементный счетчик (SYS.CTU) ... 321

1 О программе

Owen Logic – среда программирования, предназначенная для создания алгоритмов работы коммутационных приборов, относящихся к классу программируемых реле, в частности, приборов серий ПР1xx, ПР200 и панели ИПП120 производства компании ОВЕН.

Функционал Owen Logic отличается для следующих групп приборов:

- **приборы первого поколения** – ПР100, ПР102, ПР200 и панели ИПП120;
- **приборы второго поколения** – ПР103, с графическим цветным ЖКИ (ПР205, ПР225), ОВК-П.

Основной функционал Owen Logic доступен для всех приборов, функционал и интерфейсы для приборов второго поколения недоступны для других приборов.

Owen Logic позволяет пользователю разработать программу автоматизации системы по собственному алгоритму и записать ее в энергонезависимую память прибора. Для составления программы используется графический язык FBD стандарта МЭК 61131-3.

Перечень приборов, для программирования которых может использоваться Owen Logic, представлен на сайте компании ОВЕН.

1.1 Системные требования для работы Owen Logic

Операционная система:

- Windows 7 (SP1+);
- Windows 8.1;
- Windows 10;
- Windows 11.

Системные библиотеки:

- Microsoft .NET Framework 4.8;
- Microsoft .NET Desktop Runtime 6.0.8;
- Microsoft Visual C++ 2015-2022

Системные требования:

- процессор Intel Core i3 2 ГГц
- оперативная память 4 Гб;
- свободное место на диске 700 Мб;

Подключение к Интернету требуется для следующих действий:

- обновление Owen Logic;
- загрузка шаблонов сетевых устройств;
- загрузка макросов в Менеджере компонентов.

ПРИМЕЧАНИЕ
Недоступно в версии Owen Logic 2.9 и ниже. Для получения доступа к макросам используйте актуальную версию Owen Logic

1.2 Используемые аббревиатуры и термины

Drag and drop — механизм перемещения элементов интерфейса, который позволяет перетаскивать нужный элемент, удерживая его кнопкой мыши, из одной области в другую.

FBD (Function Block Diagram) — графический язык диаграмм функциональных блоков, который предназначен для визуального программирования.

LoRa – технология беспроводной передачи данных.

LoRaWAN (Long Range Wide Area Network) — сетевой протокол, определяющий правила взаимодействия в сети LoRa.

OwenCloud — облачный сервис компании «ОВЕН», применяемый для удаленного мониторинга, управления и хранения архивов данных приборов, используемых в системах автоматизации. Доступ к сервису осуществляется с помощью web-браузера или мобильного приложения (подробнее см. owen.ru/owencloud).

Owen Configurator — ПО для настройки и задачи параметров устройствам компании «ОВЕН».

ST (Structured Text) — язык программирования стандарта IEC61131-3. Предназначен для программирования промышленных контроллеров и операторских станций.

ЖКИ — жидкокристаллический индикатор.

Коммутационная программа, программа — разработанный пользователем на схеме проекта алгоритм работы для последующей записи в прибор.

Контекстное меню — элемент графического интерфейса, представляющий собой список команд, вызываемый пользователем для выбора необходимого действия над выбранным объектом. В Owen Logic вызывается нажатием ПКМ по объекту.

Логический сигнал («сигнал») — дискретная физическая величина (напряжение или ток), принимающая только два значения: включено – соответствует логической «1» (лог. «1»), и отключено – логическому «0» (лог. «0»).

ЛКМ — левая кнопка мыши.

Макрос — функциональный блок, разработанный пользователем.

ОЗУ (Оперативное Запоминающее Устройство) — энергозависимая часть памяти прибора, в которой во время работы хранится выполняемый код программы, а также входные, выходные и промежуточные данные, обрабатываемые процессором.

ПЗУ (Постоянное Запоминающее Устройство) — энергонезависимая память, которая используется для хранения массива неизменяемых данных.

ПК — персональный компьютер.

ПКМ — правая кнопка мыши.

ПО — программное обеспечение.

Преобразователь — устройство, через которое прибор подключается к ПК.

Прибор — программируемое устройство, например ПР200.

Приборы второго поколения — ПР103, с графическим цветным ЖКИ (ПР205, ПР225), ОВК-П;

Приборы первого поколения — ПР100, ПР102, ПР200 и панели ИПП120;

Проект — файл, созданный в Owen Logic, включающий в себя программу и настройки для последующего хранения на ПК.

Резервная копия (или файл автосохранения) — файл проекта с расширением *.owle*, который создается в фоновом режиме в результате срабатывания таймера автосохранения.

Слот — разъем в приборе для подключения интерфейса связи или модуля расширения.

Уставка — заданное значение параметра функционального блока (установленный режим).

Функциональный блок (ФБ) — структурная единица программы, которая после выполнения выдает одно или более значений. Может быть создано множество поименованных экземпляров (копий) функционального блока.

Функция — структурная единица программы, которая после выполнения выдает только одно значение. Функция не хранит информацию о своем внутреннем состоянии, то есть вызов функции с одними и теми же фактическими параметрами выдает то же значение.

Холст, схема — поле для размещения графических компонентов проекта и редактирования соединительных цепей между ними.

Цикл — время выполнения прибором заданной программы (зависит от количества выполняемых операций в программных цепях).

ЭСППЗУ (Электронно-Стираемое Программируемое Постоянное Запоминающее Устройство) — часть интегральной схемы микропроцессора, которая используется для хранения данных.

2 Установка

1. Скачайте на ПК и запустите файл Owen Logic.exe.
2. Откроется окно мастера установки. Ознакомьтесь с информацией и нажмите кнопку **Далее**.

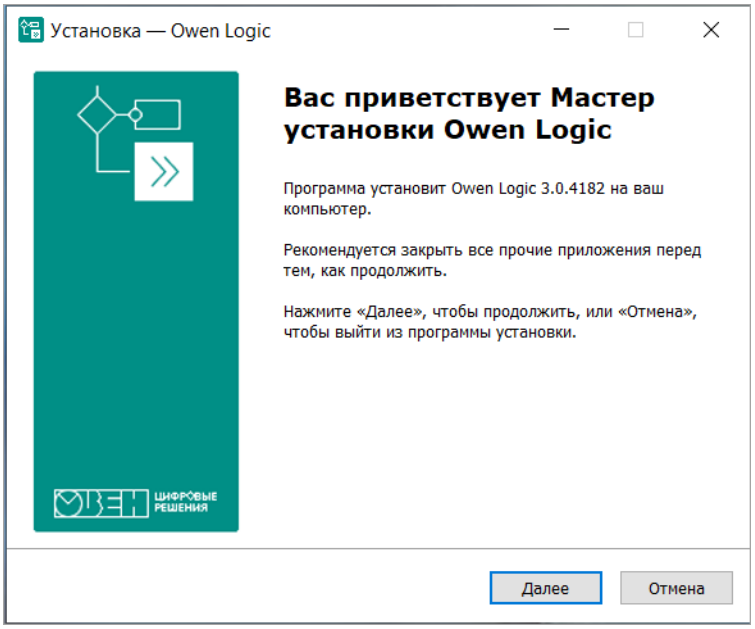


Рисунок 2.1 – Окно мастера установки

3. Ознакомьтесь с лицензионным соглашением и, в случае согласия, выберите **Я принимаю условия соглашения**. Нажмите кнопку **Далее**.

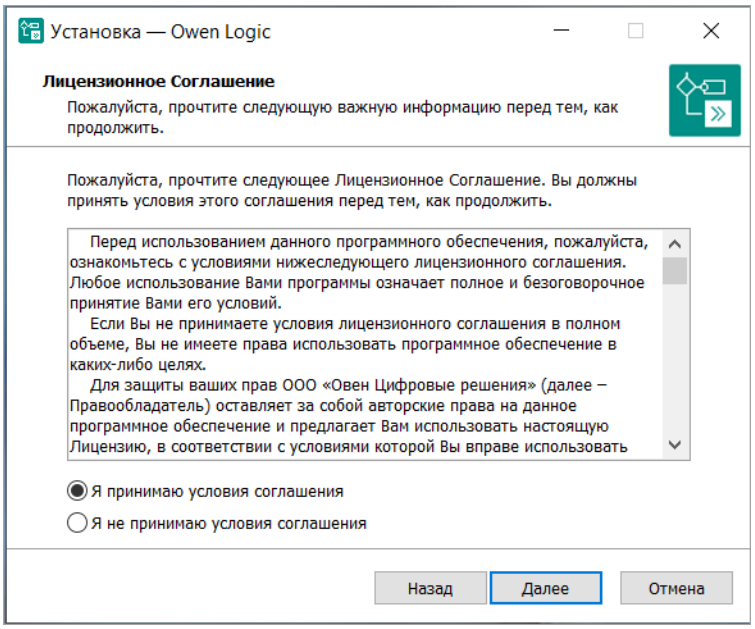


Рисунок 2.2 – Окно лицензионного соглашения

4. Выберите папку для установки Owen Logic и папку для ярлыка в меню Пуск. Нажмите кнопку **Далее**.
5. В открывшемся окне, при необходимости, установите чекбоксы:
   - Создать ярлык на рабочем столе.
   - Создать ярлык в Панели быстрого запуска.
   - Связать Owen Logic с файлами, имеющими расширение .owl, .owle — для автоматического открытия файлов с указанным расширением в Owen Logic.
   - Установить .NET 6 SDK — если библиотека не установлена, то мастер тиражирования на ОС Linux работать не будет.

   Нажмите кнопку **Далее**.

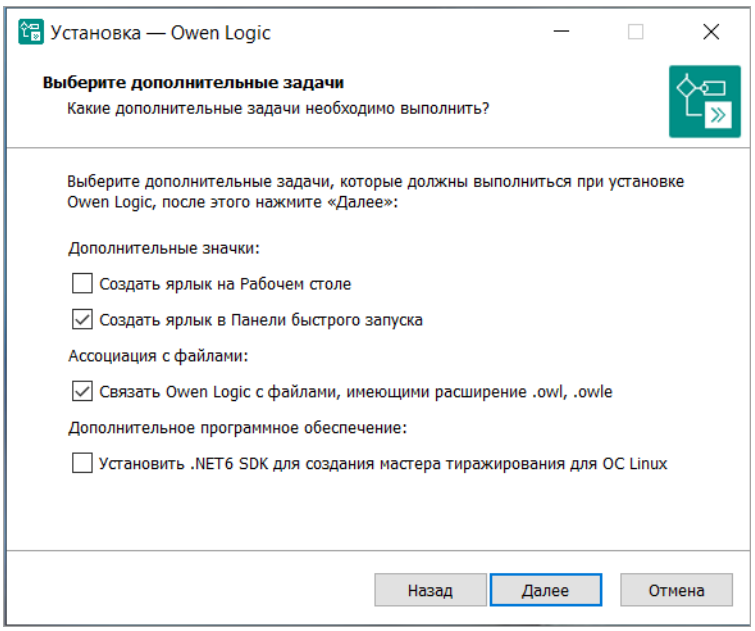


Рисунок 2.3 – Выбор дополнительных задач

6. Ознакомьтесь с информацией об установке и нажмите кнопку **Установить**.

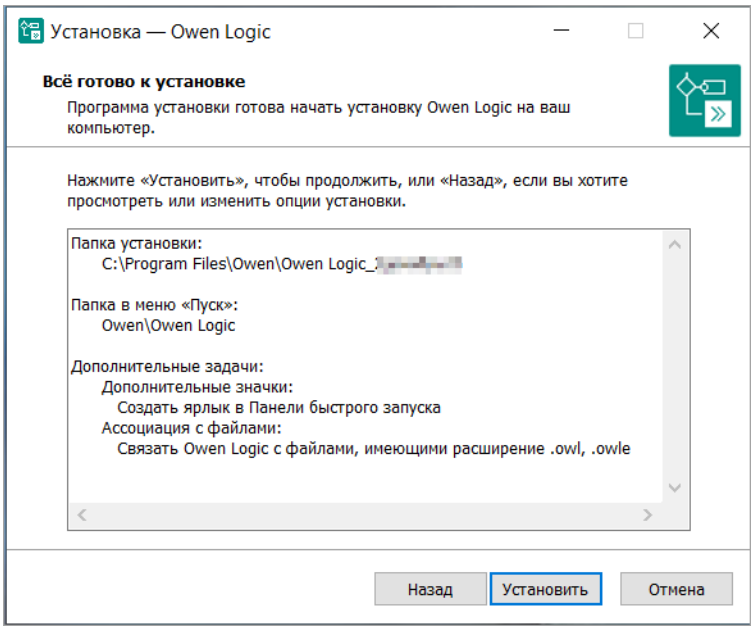


Рисунок 2.4 – Окно запуска установки

7. Откроется окно, в котором будет отображаться процесс установки.

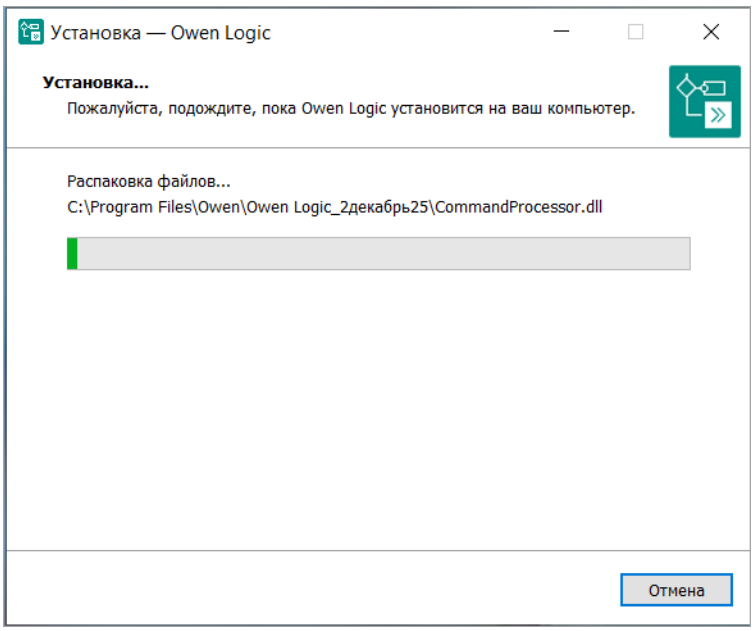


Рисунок 2.5 – Процесс установки программы

8. Дождитесь окончания установки и при необходимости установите чекбокс **Запустить Owen Logic**.

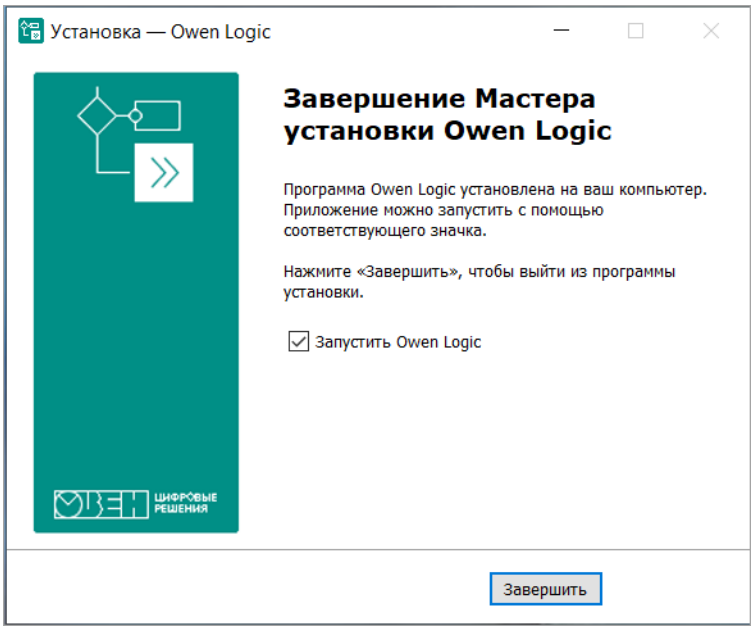


Рисунок 2.6 – Завершение установки

9. Нажмите кнопку **Завершить**.

3 Описание интерфейса

После запуска Owen Logic на мониторе ПК открывается Главное окно:

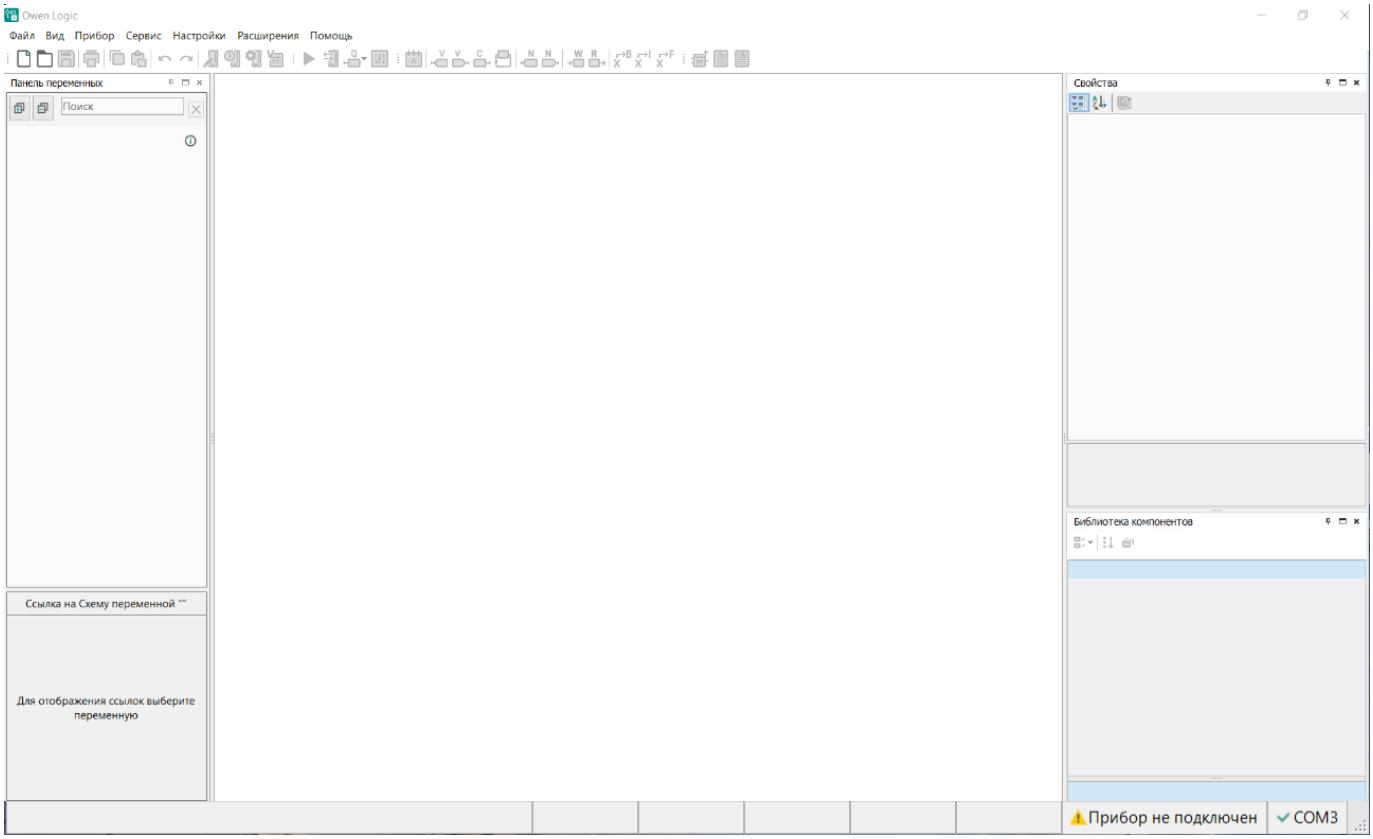


Рисунок 3.1

Главное окно содержит:

- главное меню: **Файл**, **Вид**, **Прибор**, **Сервис**, **Расширения**, **Помощь**;
- панели инструментов;
- панели Библиотека компонентов, Свойства и Переменные (до открытия или создания проекта в них нет информации);
- рабочую область проекта – поле редактирования программы (до открытия или создания проекта пустое);
- строку состояния в нижней части главного окна, показывающую информация о доступных ресурсах прибора и подключении к Owen Logic.
- менеджер экранов

3.1 Главное меню

Файл

Новый проект	Создание нового проекта. Текущий проект закрывается, перед закрытием будет предложено сохранить проект
Смена целевой платформы	Вызов окна смены модификации прибора для проекта
Открыть проект	Открытие ранее созданного и сохраненного проекта
Сохранить текущий документ	Сохранение текущего проекта
Сохранить проект	Пересохранение проекта под текущим именем
Сохранить проект как	Сохранение проекта под новым именем или в другом месте
Создать файл с ключом	Создание файла с ключом (подробнее см. Защита проекта ключом 4.2.2)
Сведения о проекте	Вызов окна сведений о проекте 4.9
Создать макрос	Вызов редактора макросов 4.11

Импорт из файла	Импорт макроса, функции или функционального блока на ST из файла в библиотеку компонентов
Экспорт в файл	Экспорт макроса функции или функционального блока на ST для сохранения на ПК
Менеджер компонентов	Вызов окна Менеджера компонентов 4.10
Печать	Печать схемы открытого проекта
Последние проекты	Список недавно открытых проектов
Выход	Выход из Owen Logic

Вид

Отменить	Отмена последнего действия в проекте
Восстановить	Восстановление отмененного действия
Индикаторы состояний	Включение/отключение индикаторов в строке состояний 3.7
Панель библиотек	Скрыть/отобразить панель Библиотека компонентов 3.3
Панель свойств	Скрыть/отобразить панель Свойства 3.4
Переменные	Скрыть/отобразить панель Переменные 3.5
Менеджер экранов	Скрыть/отобразить Менеджер экранов 4.4 (меню доступно для приборов с дисплеем)
Сбросить расположение панелей	Сброс пользовательских настроек положения панелей и возврат к настройкам по умолчанию

Прибор

Записать программу в прибор	Запуск записи созданной программы в прибор (кнопка неактивна, пока не подключен прибор)
Обновить встроенное ПО	Запуск обновления прошивки подключенного прибора
Информация	Вызов окна с информацией 8.1 о подключенном приборе
Таблица переменных	Вызов окна для работы с переменными 6 в проекте
Юстировка входов/выходов	Вызов окна юстировки 8.4 (кнопка неактивна, пока не подключен прибор)
Настройка прибора	Вызов окна настройки прибора 5
Настройка порта	Вызов окна настройки порта 4.6 связи с прибором

Сервис

Перенумеровать компоненты	Изменение нумерации ФБ в рабочей зоне проекта
Режим симулятора	Запуск режима симуляции 4.5
Режим OFFLINE	Включение/отключение запрета опроса прибора по USB

Настройки

Автосохранение	Настройка автоматического создания резервных копий проекта в фоновом режиме

Расширения

ПРИМЕЧАНИЕ
Меню **Расширения** по умолчанию содержит один пункт — **Управление расширениями**. Другие пункты появляются по мере установки расширений, подробнее см. раздел Расширения 9.

Управление расширениями	Вызов окна управления расширениями 9
Экспорт устройства в OPC	Запуск расширения Экспорт устройства в OPC 9.1
Экспорт устройства в OwenCloud	Запуск расширения Экспорт устройства в OwenCloud 9.3
Мастер тиражирования	Запуск расширения Мастер тиражирования 9.1

Помощь

Автопроверка нового релиза	Включение/отключение автоматической проверки новых версий и релизов Owen Logic на сервере обновления
Проверить обновления...	Вызов окна для проверки и установки новых версий и релизов Owen Logic
Справка	Вызов окна справки
История версий	Вызов в браузере страницы со списком версий ПО и описанием
О программе	Вызов окна информации о текущей версии Owen Logic

3.2 Панели инструментов

Панель инструментов

	Новый проект	Создание нового проекта. Текущий проект закрывается, перед закрытием будет предложено сохранить проект
	Открыть проект	Открытие ранее созданного и сохраненного проекта
	Сохранить проект	Сохранение текущего проекта. При первом сохранении вызывает окно для присвоения имени файлу
	Печать	Печать схемы открытого проекта
	Копировать	Копирование выделенного в рабочей области элемента
	Вставить	Вставка скопированного элемента
	Отменить	Отмена последнего действия в проекте
	Восстановить	Восстановление отмененного действия
	Записать программу в прибор	Запуск записи созданной программы в прибор
	Информация	Вызов окна с информацией о подключенном приборе
	Настройки прибора	Вызов окна с настройками прибора
	Таблица переменных	Вызов окна для работы с переменными в проекте

Панель отладки

	Режим симулятора	Запуск режима симуляции
	Онлайн отладка	Запуск режима онлайн отладки
	Порядок исполнения	Изменение порядка вычисления значений на выходах программы или макроса и порядка передачи значений по обратным связям
	Перенумеровать компоненты	Изменение нумерации ФБ в рабочей зоне проекта

Панель вставки

	Блок комментариев	Добавление комментария на схему проекта
	Создание блока выходной переменной	Создание блока для записи значения в переменную
	Создание блока входной переменной	Создание блока для считывания значения из переменной
	Создание блока константы	Создание блока с фиксированным числовым значением
	Линия задержки	Создание задержки на 1 цикл
	Создание сетевой выходной переменной	Создание блока переменной, значение которой задается по сети
	Создание сетевой входной переменной	Создание блока переменной, значение которой считывается по сети
	Создание блока записи в ФБ	Создание блока для записи значения переменной ФБ, на схеме может не отображаться соединительная линия цепи передачи сигнала
	Создание блока чтения из ФБ	Создание блока для считывания значения переменной ФБ, на схеме может не отображаться соединительная линия цепи передачи сигнала
	Преобразователь в булевское значение	Конвертация значения сигнала в булевское значение
	Преобразователь в целочисленное значение	Конвертация значения сигнала в целочисленное значение
	Преобразователь в число с плавающей точкой	Конвертация значения сигнала в число с плавающей точкой
	Создать макрос	Создание нового макроса в проекте
	Создать функцию на ST	Создание функции на языке ST для использования в проекте
	Создать функциональный блок на ST	Создание функционального блока на языке ST для использования в проекте

3.3 Панель Библиотека компонентов

Включить/выключить отображение панели **Библиотека компонентов** в рабочей области можно в главном меню **Вид**.

Если в Owen Logic открыт проект, то на панели **Библиотека компонентов** отображаются ее разделы:

- Функция 7.1;
- Функциональный блок 7.2;
- Макросы проекта 7.3;
- Функции на ST 4.12.1;
- Функциональные блоки на ST 4.12.2.

Раздел выбирается в нижней части панели.

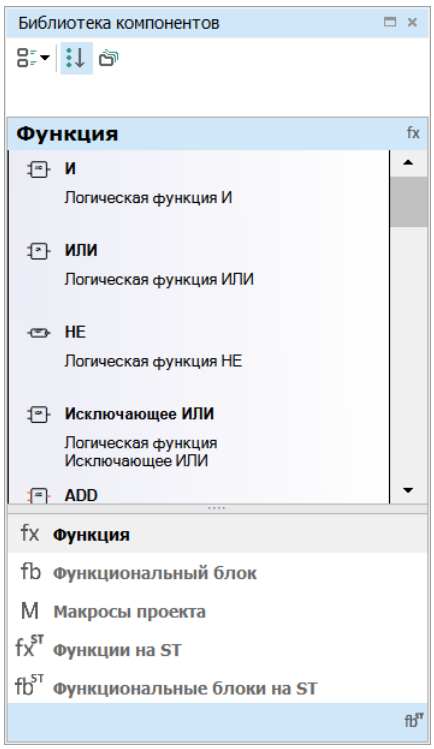


Рисунок 3.2

Вид

В выпадающем меню можно выбрать вид расположения компонентов:

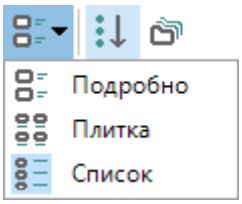


Рисунок 3.3

Способ отображения компонентов можно выбрать с помощью кнопок в верхней части панели:

- **Показать все компоненты** – отображаются все компоненты выбранного раздела;

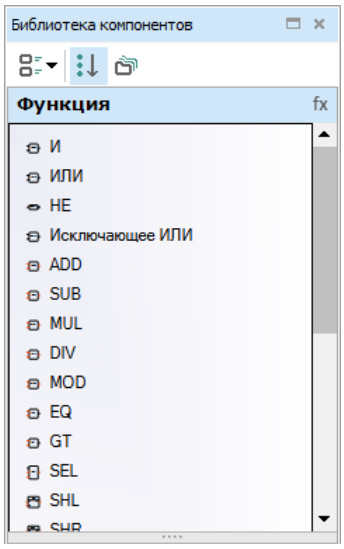


Рисунок 3.4

- **Группировка по папкам** – отображаются папки, содержащие компоненты разных типов.

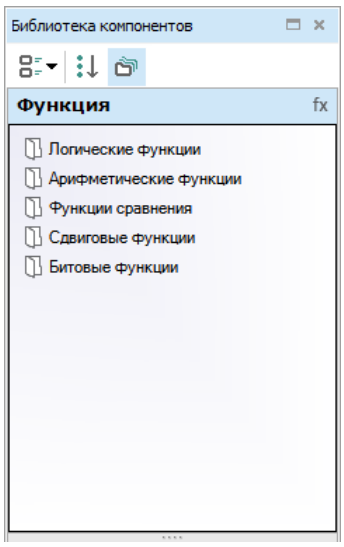


Рисунок 3.5

Описание разделов и компонентов панели **Библиотека компонентов** приведено в разделе Библиотека 7.

3.4 Панель Свойства

Включить/выключить отображение панели **Свойства** в рабочей области можно в главном меню **Вид**.

На панели отображаются и редактируются:

- параметры элементов программы;
- размеры холста;
- часто используемые свойства входов и выходов (полный перечень свойств находится в разделе Входы 5.5 настроек прибора);
- параметры комментария, свободных переменных, блоков преобразователей.

Если элемент не выбран, то на панели отображаются свойства холста.

Для отображения свойств элемента в панели следует нажать на него ЛКМ.

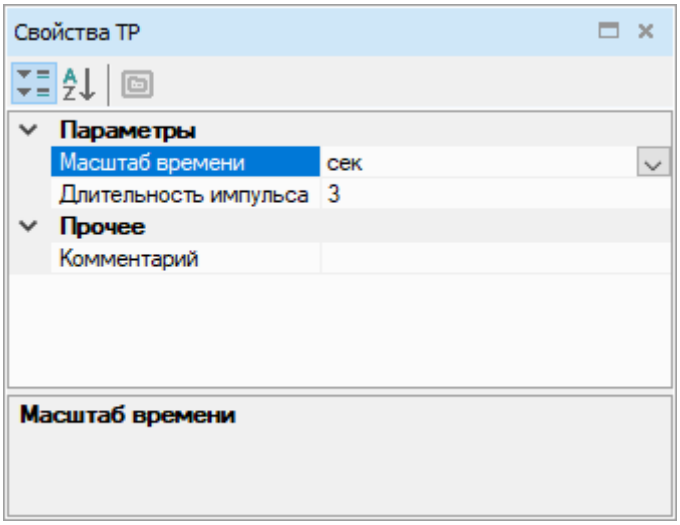


Рисунок 3.6

Способы отображения параметров в панели:

- — в алфавитном порядке;
- — по категориям.

3.5 Панель переменных

Включить/выключить отображение **Панели переменных** в рабочей области можно в главном меню **Вид**.

На **Панели переменных** отображаются переменные из Таблицы переменных 6.

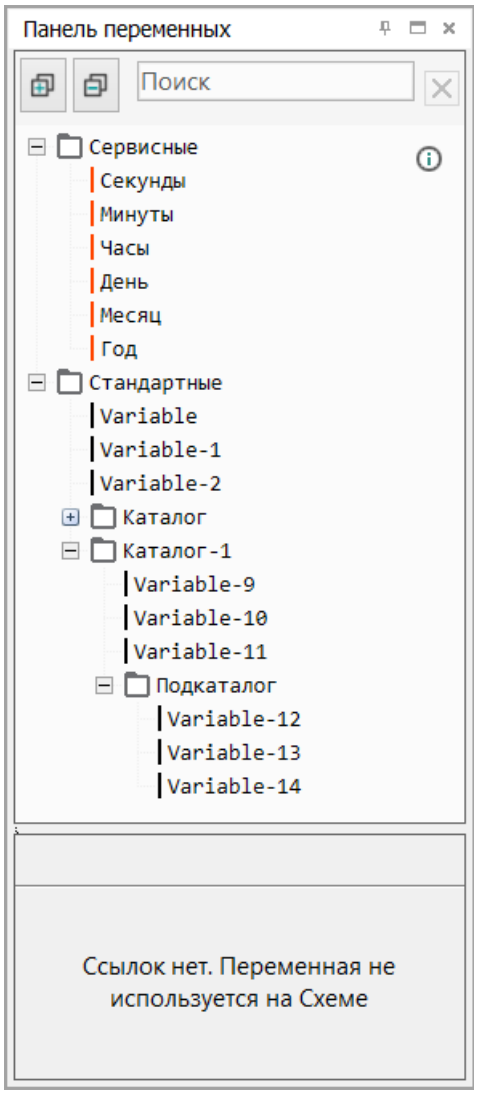


Рисунок 3.7 – Окно Панель Переменных

	Раскрыть подкаталоги
	Свернуть подкаталоги

Раскрыть/свернуть один каталог можно с помощью частных кнопок + / -, расположенных перед иконкой каталога.

Создание блока переменной

Для создания **блока входной переменной** следует перетащить переменную (методом drag & drop) на холст.

Для создания **блока выходной переменной** следует перетащить переменную на холст, удерживая клавишу Shift.

Можно перетаскивать на холст сразу несколько переменных. Для этого предварительно выделите их на панели переменных.

Способы выделения переменных:

- **CTRL + ЛКМ** по переменным. Каждый клик добавляет переменную в выделение или исключает её, если она уже выделена.
- **SHIFT + стрелки ↓ / ↑.** Не выделенные переменные выделяются, с уже выделенных выделение снимается. Каталоги/подкаталоги не выделяются.
- **Выделение диапазона (SHIFT).** Нажмите **ЛКМ** по первой переменной диапазона, затем, удерживая **SHIFT**, нажмите **ЛКМ** по последней. Все переменные между ними (включительно) будут выделены.

ПРИМЕЧАНИЕ
Выделять переменные можно только в пределах одного каталога.

Если переменную перетащить к коннектору элемента схемы, то будет создан блок переменной, присоединенный к этому коннектору. Если к коннектору элемента схемы перетащить несколько выделенных переменных, связь с коннектором создана не будет.

Ссылки

Для отображения блоков, к которым привязана переменная, следует нажать на имя переменной в панели. В области ссылок панели отобразятся ссылки на блоки. Если кликнуть по ссылке, то на холсте будет выделен блок, к которому привязана переменная.

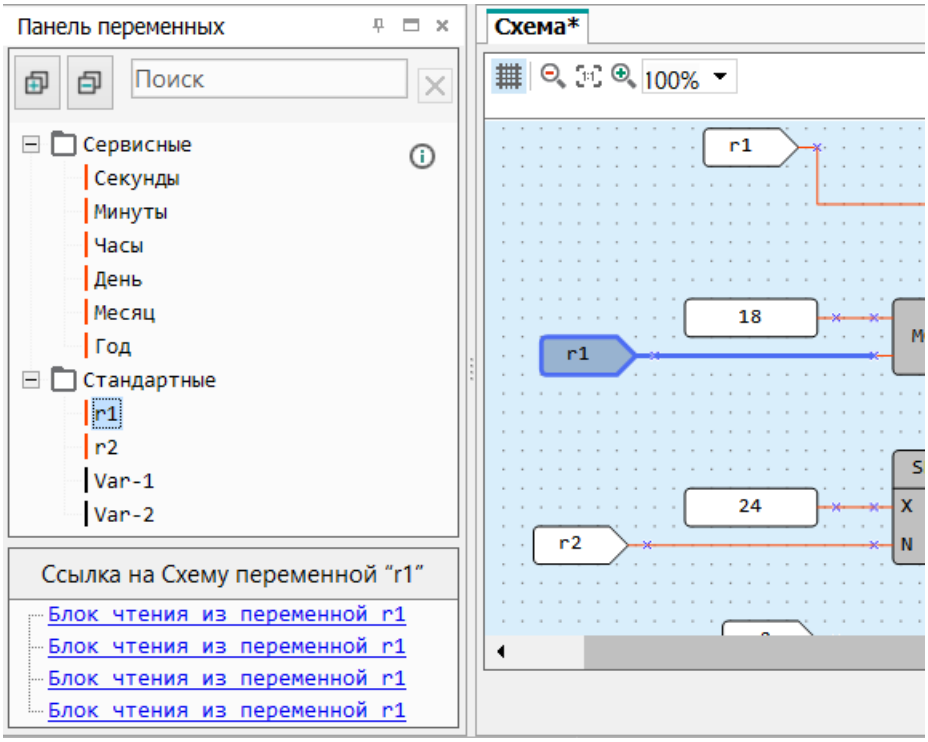


Рисунок 3.8 – Ссылка на переменную

С помощью контекстного меню переменной можно найти переменную в Панели переменных, для этого следует выбрать пункт **Найти в Панели переменных**. Привязки переменной к другим блокам можно просмотреть в пункте **Показать ссылки** контекстного меню блока переменной. Если нажать на ссылку в контекстном меню, то будет совершен переход к выбранному блоку.

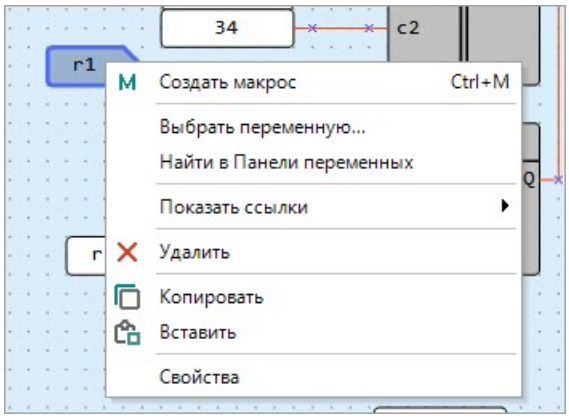


Рисунок 3.9 – Контекстное меню переменной

3.6 Рабочая область

После создания проекта или открытия сохраненного проекта в рабочей области появляется холст для разработки программы на языке FBD.

Если прибор поддерживает работу с функциями на языке ST, отобразится вкладка **Редактор функции**.

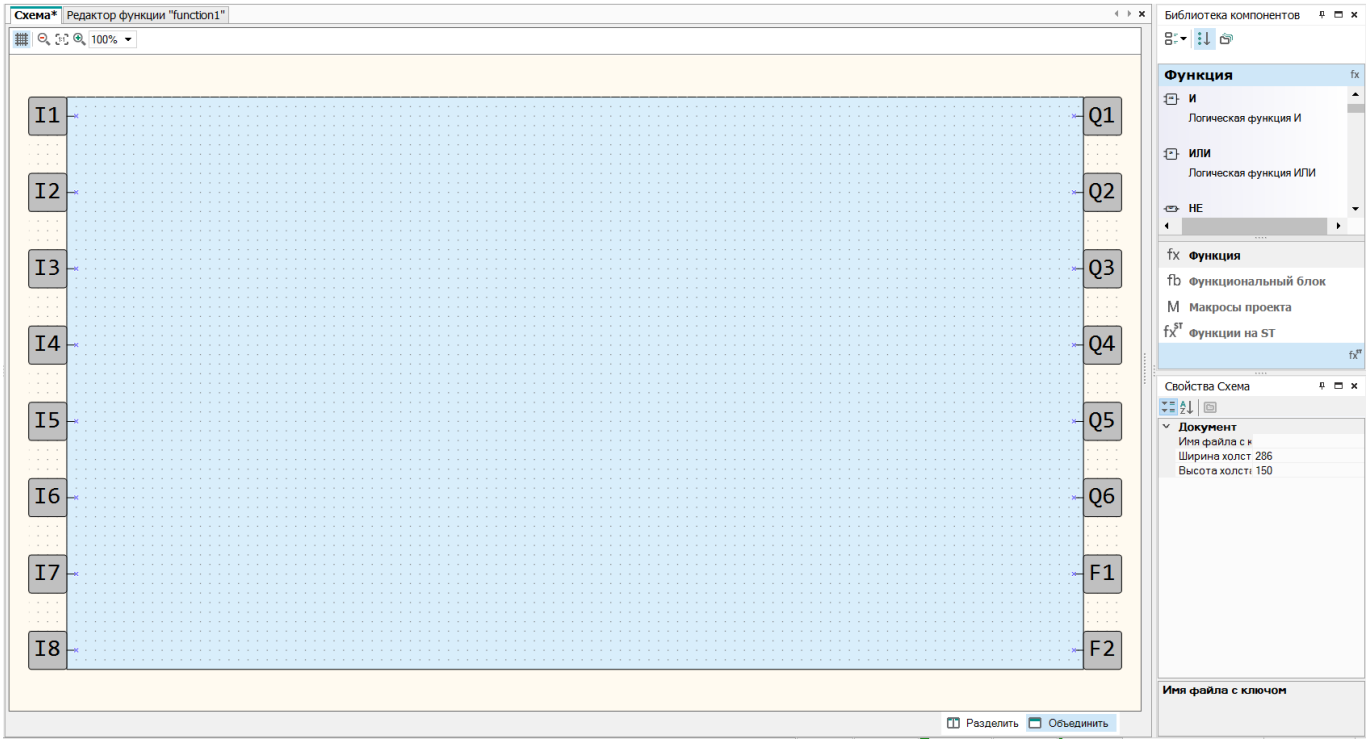


Рисунок 3.10

На холсте размещаются элементы и блоки из панели Библиотека компонентов 3.3 и панели вставки. Изменить размеры холста можно на панели Свойства 3.4.

Входы и выходы

Входы и выходы прибора обозначены в виде квадратов с маркировкой номеров:

- вдоль левого края холста расположены **входы** (**Ix** — дискретные, **FIx** — быстрые дискретные, **AIx** — аналоговые);
- вдоль правого края холста расположены **выходы** (**Qx** — дискретные, **AOx** — аналоговые, **K** — ключи, **Fx** — светодиоды).

Цифры в обозначениях соответствуют номерам физических входов и выходов прибора. Входы и выходы можно перетаскивать вверх/вниз вдоль холста методом drag & drop, удалять и добавлять для удобства компоновки схемы.

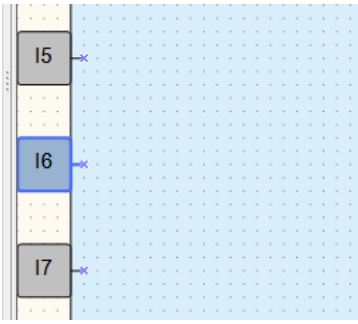


Рисунок 3.11

Управление

Над холстом располагаются кнопки:

	Включить/Выключить сетку	При включенной сетке на холсте отображаются вертикальная и горизонтальная линейки и сетка. Элементы и линии связи при размещении привязываются к сетке
	Уменьшить масштаб	Уменьшение масштаба на 10 % от первоначального

1:1	**Оригинальный размер**	Возврат к исходному масштабу
	Увеличить масштаб	Увеличение масштаба на 10 % от первоначального

Задать требуемый масштаб можно с помощью выпадающего меню справа от описанных выше кнопок.

В нижней части холста находятся кнопки:

Разделить	Используется для одновременного отображения двух областей одной схемы в разных окнах
Объединить	Используется для отображения схемы в одном окне

3.7 Строка состояния

В строке состояния отображается информация о доступных ресурсах прибора и подключении.

Вид

В зависимости от прибора, выбранного при создании проекта, перечень индикаторов в строке состояния может различаться.

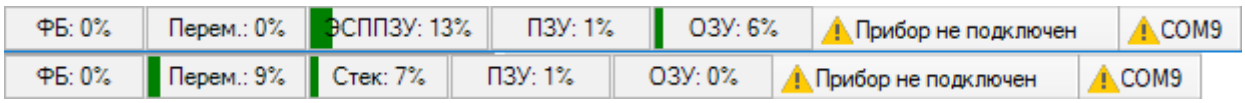


Рисунок 3.12

Индикаторы

Настройка отображения индикаторов в строке состояния доступна через **Главное меню** → **Вид** → **Индикаторы состояний**.

В строке состояния может отображаться следующая информация:

- **ФБ** — доступное количество экземпляров функциональных блоков;
- **Перем.** — доступное количество переменных;
- **Стек** — уровень использования памяти выделенной под стек. Стек используется для промежуточных вычислений в программе;
- **Сист. ЭСППЗУ** — доступное количество системной энергонезависимой памяти. Индикатор заполняется в случае использования сетевых переменных на схеме, привязки переменных к визуализации и параметрам прибора и добавления в проект модулей расширения;
- **ЭСППЗУ** — доступное количество энергонезависимой памяти. Индикатор заполняется в случае использования на схеме стандартных энергонезависимых переменных;
- **ПЗУ** — доступное ПЗУ прибора в процентах от общего объема: «ПЗУ: 0…100 %»;
- **Сист. ОЗУ** — доступное системное ОЗУ прибора.
- **ОЗУ** — доступное ОЗУ прибора в процентах от общего объема: «ОЗУ: 0…100 %». Чем больше используется в программе блоков и функций, тем больше памяти требуется для работы в приборе. Owen Logic автоматически рассчитывает доступную память прибора и в случае критического значения выводит соответствующее предупреждение;
- **Сист. ПЗУ визуал.** — доступное ПЗУ визуализации прибора в процентах от общего объема (только для приборов с графическим цветным ЖКИ);
- **Польз. ПЗУ визуал.** — доступное ПЗУ визуализации для растровых изображений (только для приборов с графическим цветным ЖКИ);
- **ОЗУ визуал.** — доступное ОЗУ визуализации прибора в процентах от общего объема. Расчет ведется для самого нагруженного экрана (только для приборов с графическим цветным ЖКИ);
- **ПРxxx-xxx** — модель подключенного прибора. Если прибор не подключен, на индикаторе отображается надпись **Прибор не подключен**. Нажатие на индикатор включает/выключает режим OFFLINE;
- **COMx** — номер выбранного пользователем порта для подключения прибора. Нажатие на индикатор открывает окно настройки порта.

ПРЕДУПРЕЖДЕНИЕ
Если какой-либо из индикаторов показывает заполнение более 100%, то запись программы в прибор будет запрещена, при попытке записи программы появится сообщение о превышении размера памяти.

При наведении мышки на индикатор отображается всплывающее сообщение с информацией о загрузке памяти прибора.

Режим OFFLINE

В режиме OFFLINE устанавливается запрет на связь с прибором.

Для перехода в режим OFFLINE следует нажать на индикатор модели прибора в строке состояния или выбрать в главном меню **Сервис** → **Режим OFFLINE**. Вид индикатора изменится на **OFFLINE**. Повторное нажатие отключает режим OFFLINE.

Подробнее см. в разделе Загрузка в прибор.

3.8 Менеджер и редактор экранов

Менеджер экранов

Если прибор оснащен дисплеем, то на него можно запрограммировать вывод информации из программы. Инструкция по работе с экранами приведена в разделе Программирование экрана прибора.

Включить/выключить отображение вкладки **Менеджер экранов** в рабочей области можно в главном меню **Вид**.

Менеджер экранов открывается нажатием на одноименную вкладку в левом крае рабочей области.

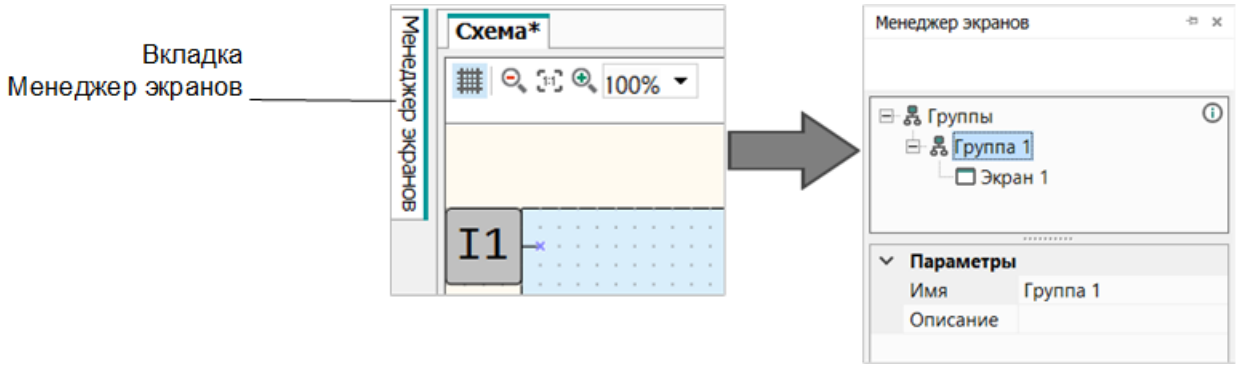


Рисунок 3.13

Менеджер экранов состоит из дерева элементов в верхней части и полей свойств в нижней части.

В дереве экранов отображаются и редактируются группы экранов и экраны.

Основные функции:

	Добавить экран (находится в верхней части менеджера экранов и в контекстном меню группы экранов приборов с графическим цветным ЖКИ)
	Удалить элемент (находится в верхней части менеджера экранов, в контекстном меню экрана и группы экранов приборов с графическим цветным ЖКИ)
	Удалить экран (в контекстном меню экрана)
	Редактировать экран (в контекстном меню экрана)
	Редактировать группу (в контекстном меню группы экранов, доступен только для приборов с монохромным текстовым ЖКИ)

При выборе экрана или группы экранов в нижней части **Менеджера экранов** отображаются их свойства.

Редактор экранов

Редактор экранов открывается выбором команды **Редактировать экран** в контекстном меню экрана в **Менеджере экранов** или двойным нажатием ЛКМ по экрану в дереве экранов.

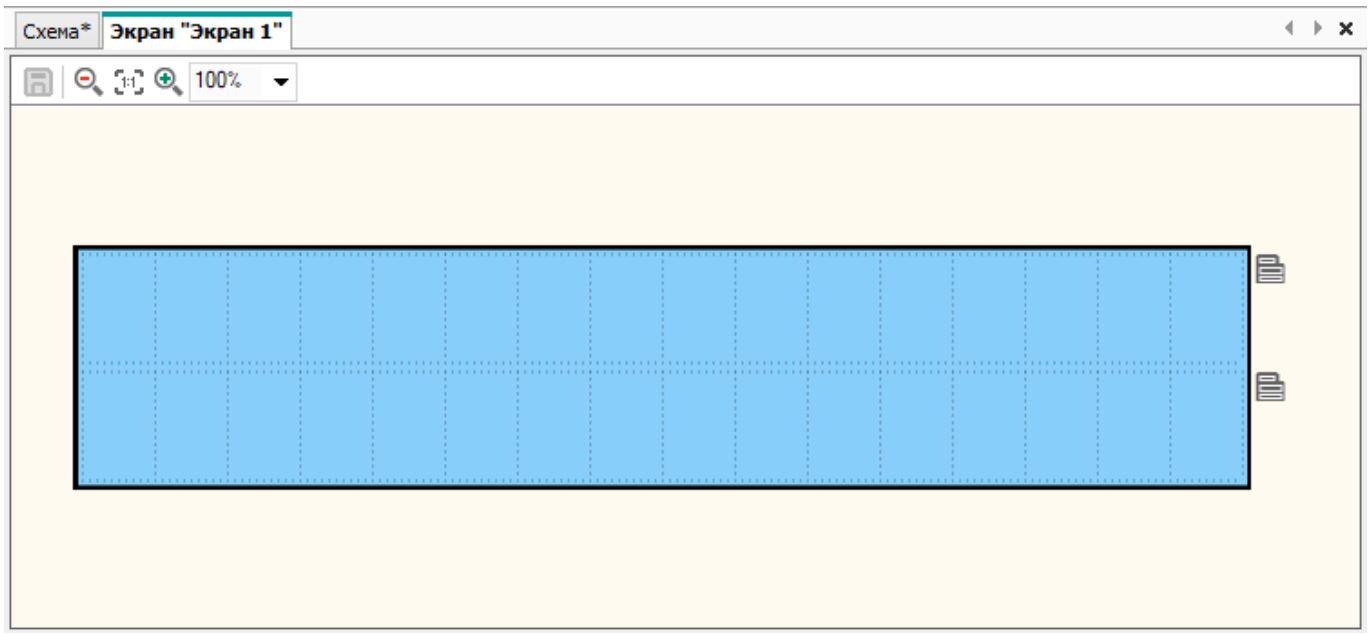


Рисунок 3.14 – Редактор экрана для прибора с монохромным текстовым ЖКИ

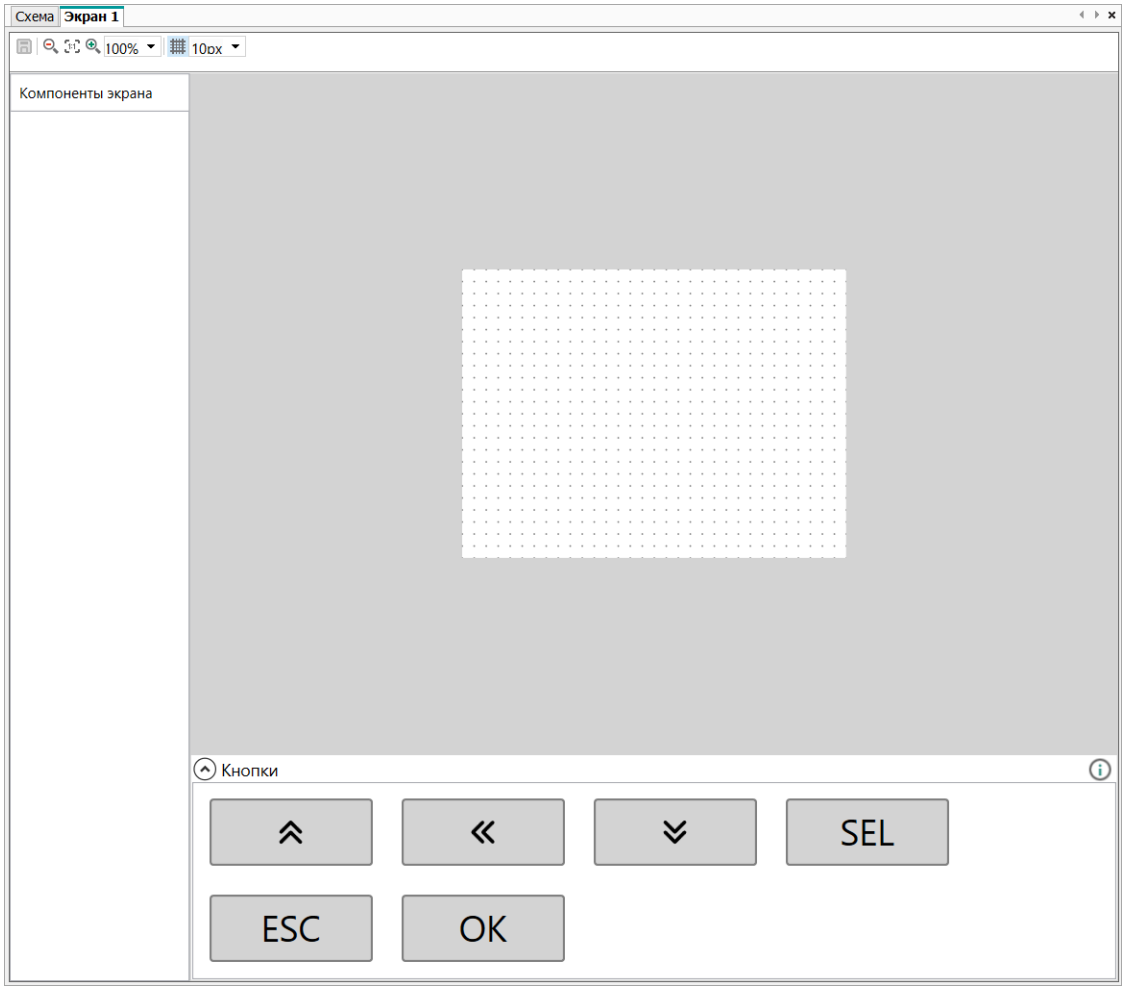


Рисунок 3.15 – Редактор экрана для прибора с графическим ЖКИ

Подробно работа с экранами рассмотрена в разделах:

- монохромный текстовый ЖКИ;
- графический цветной ЖКИ.

4 Базовые сведения

В настоящем разделе описываются основные понятия работы прибора и принцип создания программы для загрузки в прибор:

- Принцип выполнения программы 4.1;
- Разработка проекта и порядок работы 4.2;
- Разработка программы 4.3;
- Программирование дисплея прибора 4.4;
- Режим симулятора 4.5;
- Настройка порта и подключение прибора 4.6;
- Запись программы в прибор 4.7.
- Онлайн отладка 4.8;
- Сведения о проекте 4.9;
- Менеджер компонентов 4.10;
- Работа с макросами 4.11;
- Создание функций на языке ST 4.12.1.

4.1 Принцип выполнения программы

Программа для прибора составляется с учетом количества имеющихся у него входов, выходов и наличия часов реального времени.

Функциональная схема работы прибора:

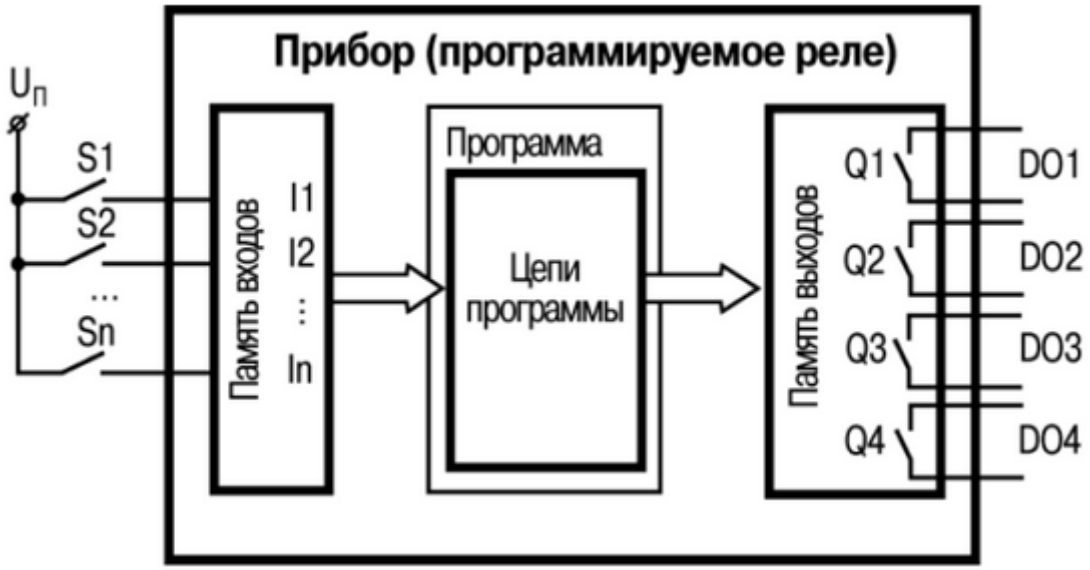


Рисунок 4.1

Работу прибора можно представить в виде последовательно выполняемых шагов (рабочий цикл):

1. Логическое состояние входов автоматически записывается в ячейки памяти входов (количество ячеек равно числу входов – I1…In).
2. Программа считывает значения из ячеек памяти входов и выполняет над ними логические операции в соответствии с алгоритмом работы.
3. После обработки всей программы результаты записываются на физические выходы прибора (для включения выходных элементов Q1…Q4).
4. Переход к Шагу 1 (после выполнения всех предыдущих шагов обработки программы цикл работы прибора повторяется с первого шага).

Время выполнения всех шагов зависит от сложности алгоритма программы (подробнее см. раздел Время цикла 8.2).

4.2 Разработка проекта и порядок работы

Создание и редактирование проекта

Для создания нового проекта:

1. Нажмите кнопку на панели инструментов или выберите в главном меню **Файл** → **Новый проект…**.
2. Откроется окно выбора модели со списком приборов. При выборе прибора в поле справа появится изображение прибора, его характеристики и модификации. Выберите модель и модификацию прибора и нажмите **Создать**.

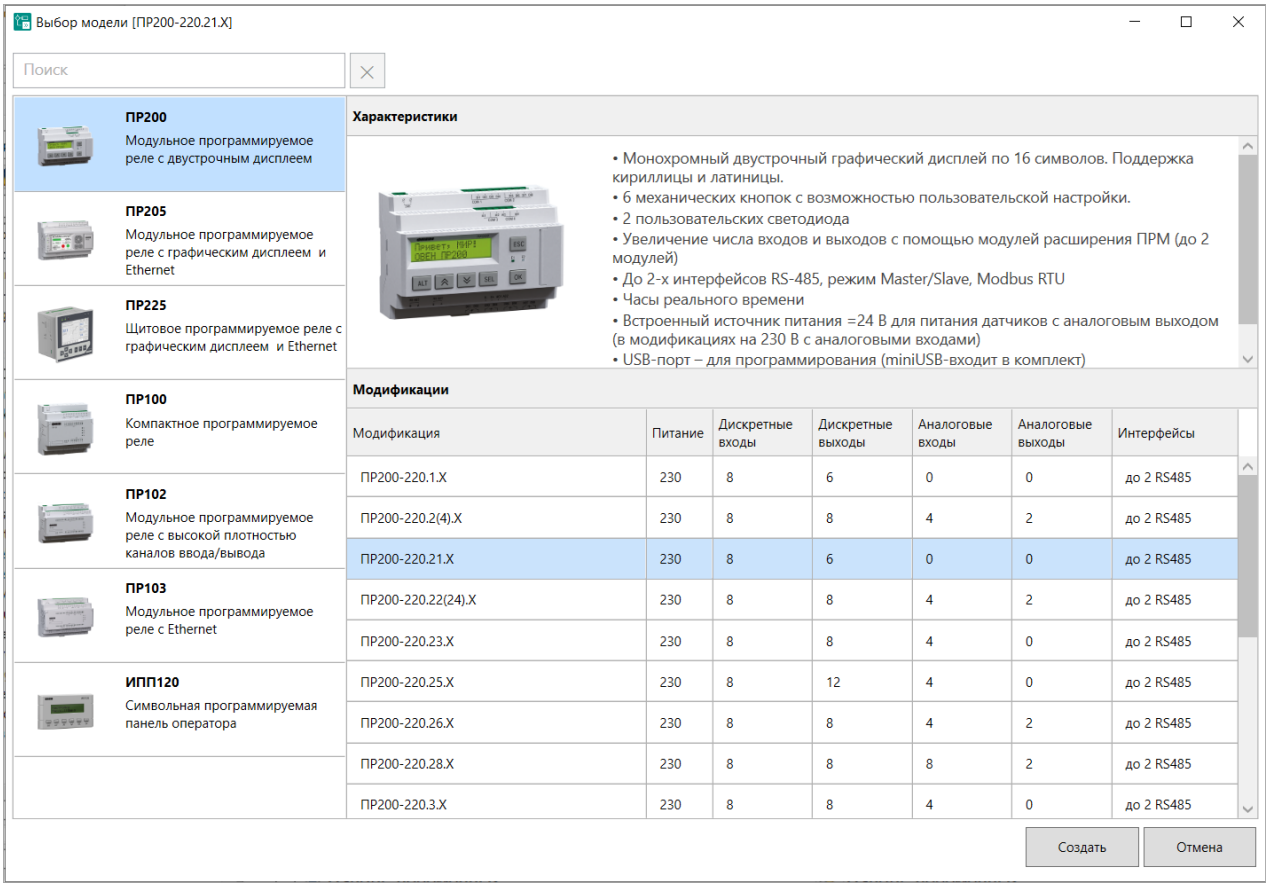


Рисунок 4.2 – Окно выбора прибора

ПРИМЕЧАНИЕ
Если к ПК подключен прибор, то Owen Logic предложит модель и модификацию подключенного прибора в окне выбора.

В главном окне Owen Logic отобразится холст проекта, станут активными строка состояния, панели Библиотека компонентов, Свойства и Переменные.

Если у выбранного прибора есть дисплей, то в главном окне отобразится вкладка Менеджер экранов, где можно настроить вывод информации на дисплей.

Сохранить текущий или открыть сохраненный проект можно с помощью соответствующих кнопок на панели инструментов или в главном меню **Файл**.

Порядок программирования прибора в проекте:

1. Разработка программы.
2. Отладка программы.
3. Подключение прибора к ПК.
4. Загрузка разработанной программы в прибор.

Разработка программы

На холсте проекта создается программа на языке FBD с помощью блоков и элементов вставки из Библиотеки компонентов.

Отладка программы

После создания можно смоделировать работу программы в режиме симулятора. Во время проверки правильности работы программы пользователь может изменять состояние входов, контролируя состояние выходов на соответствие нужным условиям. После выявления ошибок можно вернуться к правке программы.

Подключение прибора к ПК

Для подключения прибора к ПК следует настроить порт в Owen Logic и с помощью кабеля подсоединить прибор. Далее на прибор подается питание (если требуется), и производятся настройки прибора в Owen Logic.

Загрузка программы в прибор

После настройки соединения можно загрузить программу в прибор из Owen Logic.

Онлайн отладка

Загруженную программу можно проверить в режиме онлайн отладки с реальными значениями, поступающими из прибора.

4.2.1 Автосохранение проекта

Автосохранение автоматически создает резервные копии проекта через заданные интервалы. Это помогает сохранить результаты работы и быстро восстановить проект при неожиданном закрытии программы (например, из-за отключения питания), или если исходный файл проекта окажется поврежден.

Для каждого проекта создается один файл автосохранения: по истечении заданного периода файл с резервной копией перезаписывается на актуальный.

ПРИМЕЧАНИЕ
Функция автосохранения доступна в версиях Owen Logic 3.2 и выше.

При запуске Owen Logic автосохранение включено по умолчанию. Если отключить автосохранение в настройках, это состояние сохранится и применится при последующих запусках программы и при создании новых проектов. Параметры автосохранения можно изменить в настройках.

Настройка автосохранения

1. В главном меню доступен пункт **Настройки.**

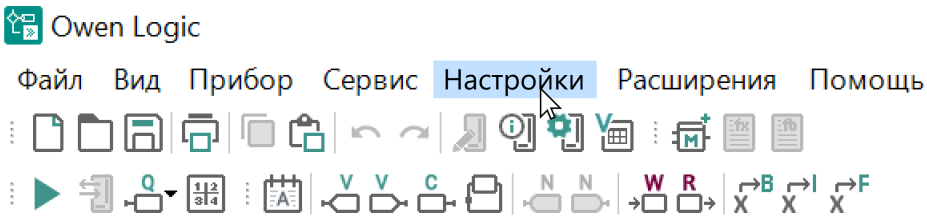


Рисунок 4.3

2. В окне настроек отображаются параметры автосохранения.

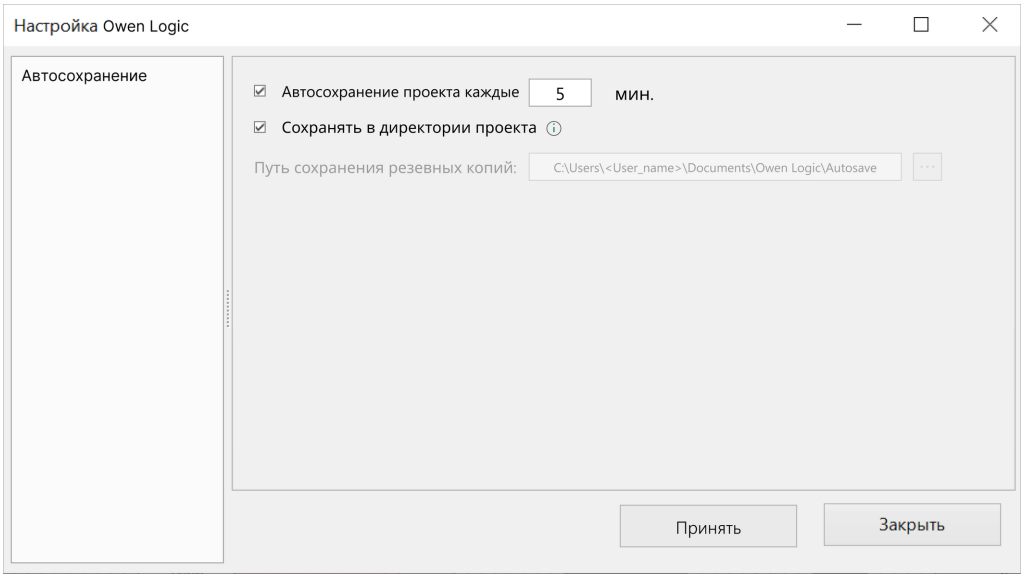


Рисунок 4.4

3. Для настройки периода автосохранения:

- при установленном чекбоксе **Автосохранение проекта каждые … мин** становится доступен ввод периода;
- значение периода вводится в диапазоне от 1 до 120 минут. При вводе недопустимого значения отображается маркер ошибки , при наведении курсора на который отображается всплывающая подсказка с допустимыми значениями параметра.

Чтобы **выключить** автосохранение проекта снимите чекбокс **Автосохранение проекта каждые ... мин**.

4. По умолчанию файлы сохраняются в директорию проекта (установлен чекбокс **Сохранять в директорию проекта**)

   Чтобы изменить папку хранения файлов автосохранения:
   - снимите чекбокс **Сохранять в директорию проекта**
   - в строке **Путь сохранения резервных копий** нажмите на кнопку "...";
   - выберите папку для сохранения файла и нажмите **Ок**.

   Если выбранная папка станет недоступна (например, выбранный для сохранения файлов USB-носитель будет отсоединен), при очередном автосохранении отобразится информационное сообщение, резервная копия автоматически сохранится в папку *C:\Users\<имя пользователя>\Documents\Owen Logic\Autosave*.

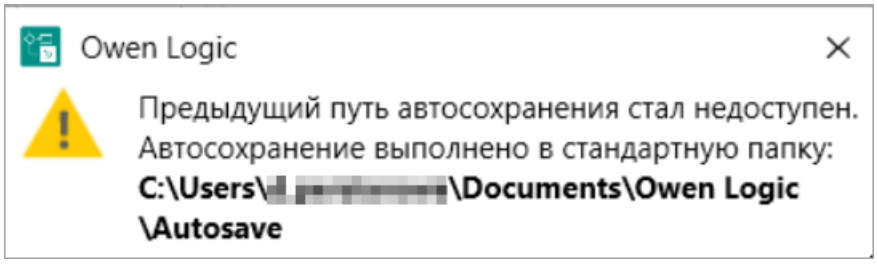


Рисунок 4.5

   Резервные копии новых проектов, которые еще не были сохранены пользователем, также сохраняются по пути *C:\Users<имя пользователя>\Documents\Owen Logic\Autosave*.

5. Кнопка **Принять** — сохраняет внесенные изменения параметров автосохранения.

   Кнопка **Закрыть** (Esc или крестик в правом верхнем углу окна Настройка Owen Logic) — закрывает окно настроек без сохранения внесенных изменений.

Имя файла с резервной копией формируется из имени файла проекта и постфикса "*_autosave*".

Пример
Файл резервной копии проекта с именем "ПР200-реле времени" будет именоваться "ПР200-реле времени_autosave.owle".

4.2.2 Защита проекта ключом

Создание файла с ключом

Для создания файла с ключом следует:

1. Выбрать в главном меню **Файл** → **Создать файл с ключом…**

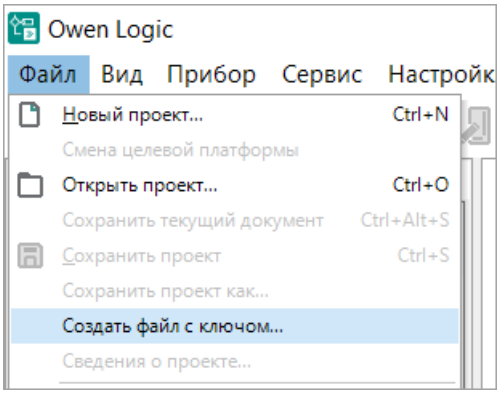


Рисунок 4.6

2. В открывшемся диалоговом окне ввести имя файла и сгенерировать ключ. Ключ можно изменить нажатием кнопки или вручную.

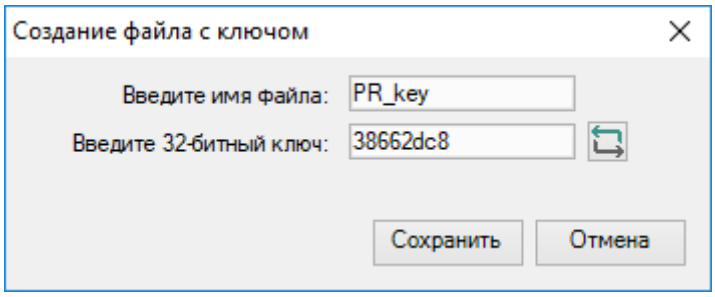


Рисунок 4.7

3. Сохранить файл. Файл с ключом по умолчанию сохраняется в папку *C:\Users\[имяпользователя]\Документы\Owen Logic\Keys*.

Привязка файла-ключа

Для привязки файла-ключа к проекту следует:

1. На панели свойств **Свойства Схема** в свойстве файла нажать икону «**…**».

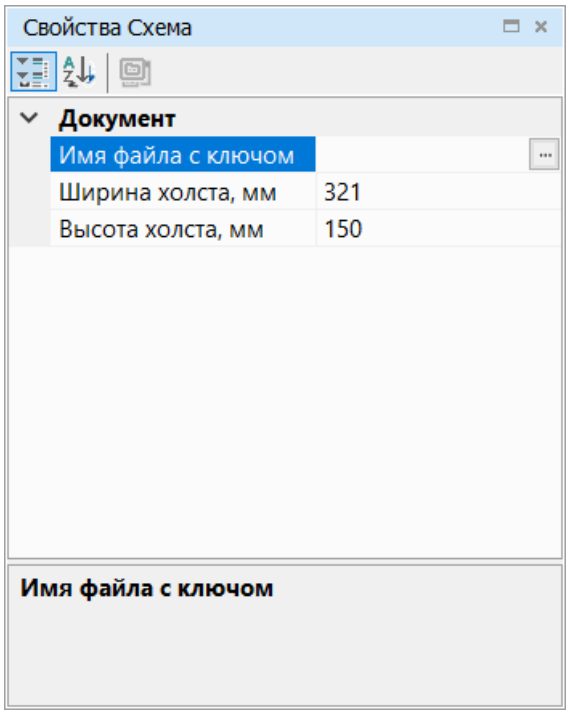


Рисунок 4.8

2. В открывшемся окне выбрать файл-ключ.

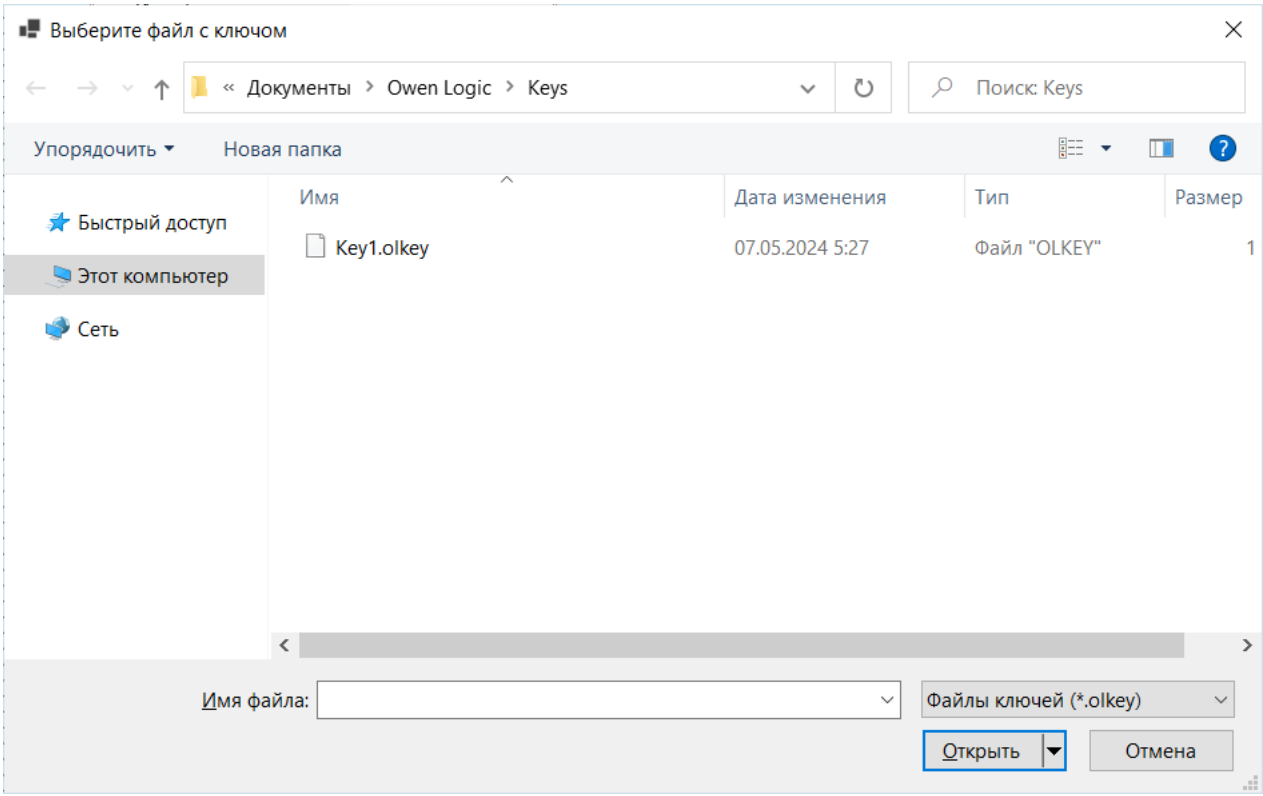


Рисунок 4.9

ПРИМЕЧАНИЕ
Если в папке *C:\Users\[имяпользователя]\Документы\Owen Logic\Keys* не найдено ни одного файла-ключа, Owen Logic предложит создать новый ключ.

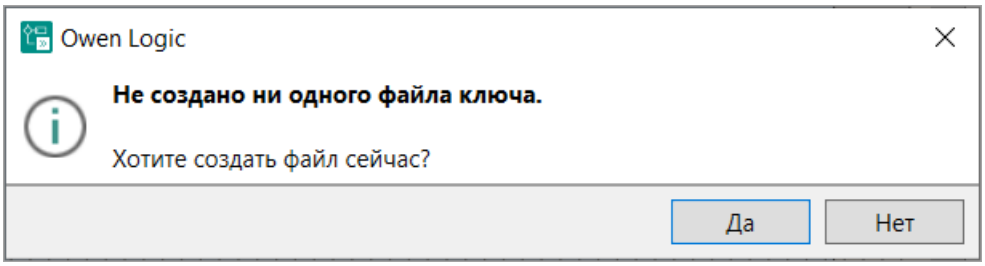


Рисунок 4.10

3. Нажать кнопку открыть.

4.3 Разработка программы

Разработку программы рекомендуется начать с планирования. План должен описывать все возможные состояния прибора во время функционирования (в виде диаграммы режимов, таблицы состояний, электрической или функциональной схемы и/или др.).

После того как продуманы все задачи, которые должны выполняться прибором, следует составить программу на холсте проекта с помощью элементов Библиотеки компонентов.

Размещение компонентов и создание связей

Компоненты размещаются на схеме путем выбора нужного компонента в панели Библиотека компонентов и перемещения на холст методом drag & drop.

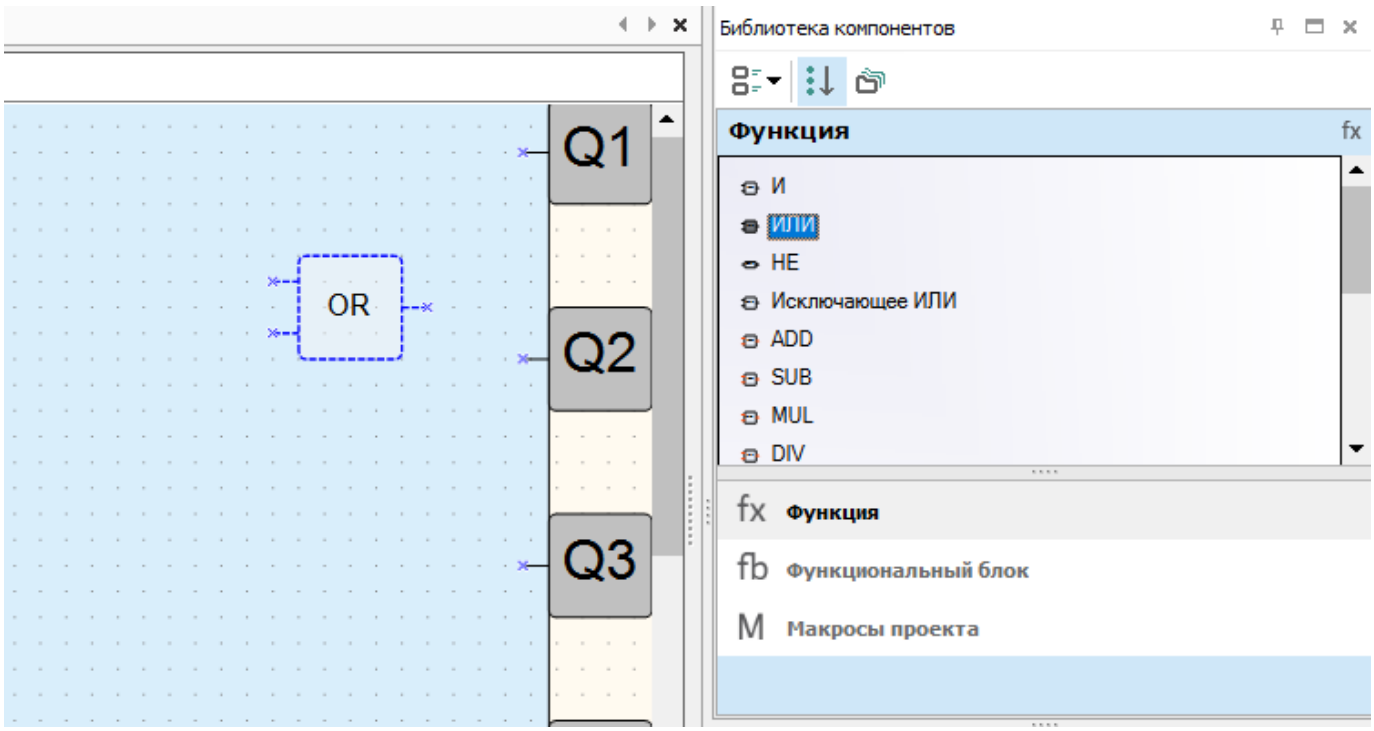


Рисунок 4.11

Входы и выходы прибора и компонентов следует соединить, проведя между ними линии связи. Для создания соединительной линии следует:

- нажать ЛКМ на выход первого компонента. Линия присоединится к нему и потянется вслед за курсором мыши;
- для создания изгиба линии следует нажать на холсте для отметки точки поворота линии;
- протянуть линию до входа второго компонента и нажать по нему. Линия связи создана.

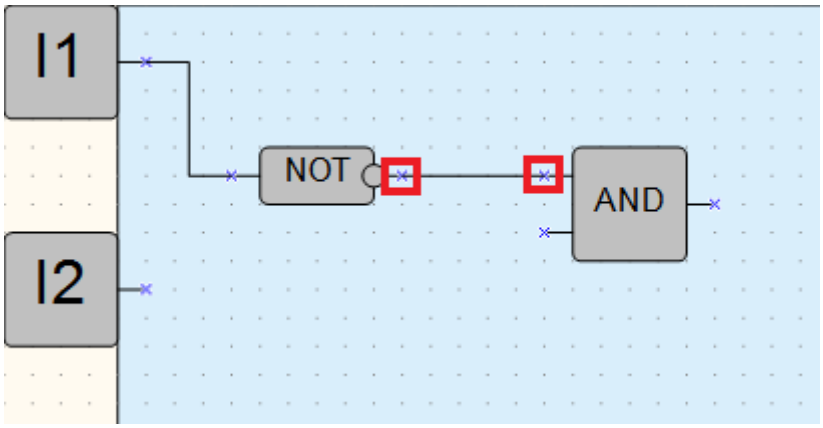


Рисунок 4.12

Если типы входа и выхода компонентов не совпадают, то линия не создастся. Подробнее см. раздел Типы переменных.

Для выделения блока или компонента на схеме следует нажать по нему ЛКМ. Чтобы выделить несколько элементов схему, можно воспользоваться «лассо» – заключить нужные элементы внутрь растягивающегося вслед за курсором мыши прямоугольника, удерживая ЛКМ.

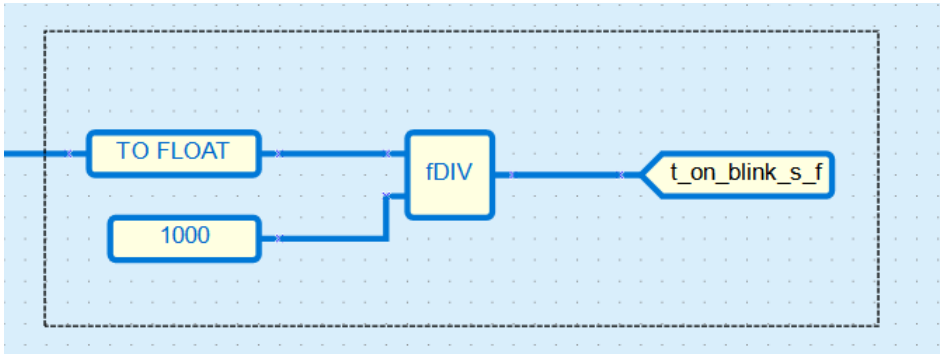


Рисунок 4.13

Незадействованные переменные не участвуют в логике программы и не занимают память устройства.

Настройка параметров компонентов

Чтобы отредактировать параметры, следует выделить нужный компонент или блок на схеме и настроить значения на панели Свойства.

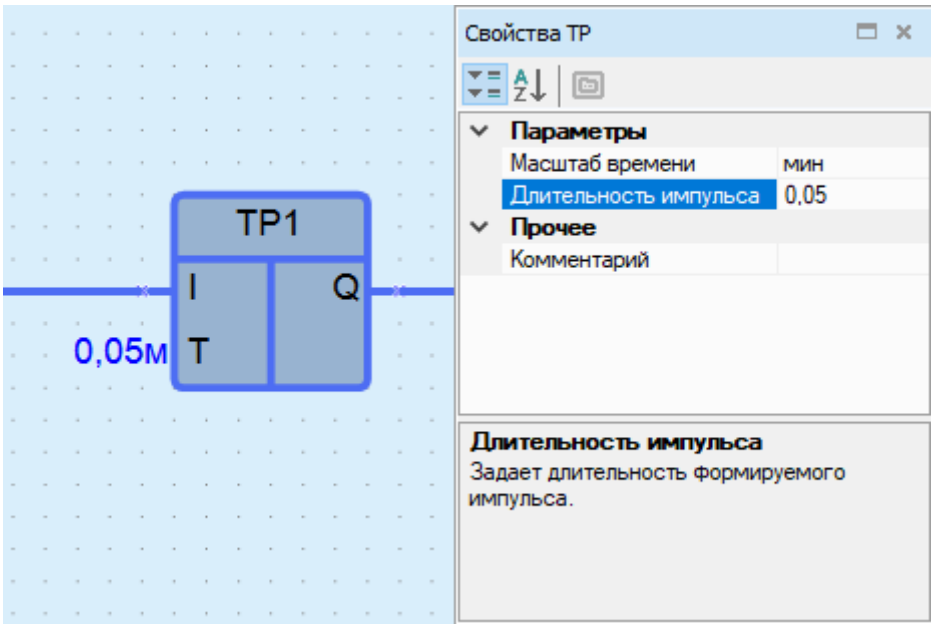


Рисунок 4.14

Свойства компонента или блока можно открыть с помощью контекстного меню.

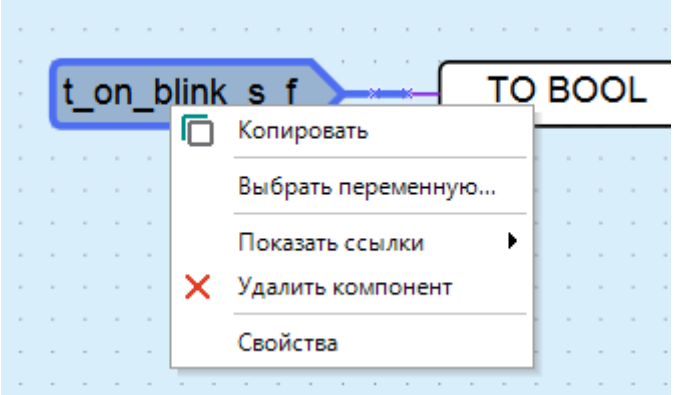


Рисунок 4.15

Для разработки программы используются следующие блоки и функции, вызываемые на панели вставки:

Блок комментариев	Размещение текстового комментария на схеме
Блок переменной	Размещение блока переменной для записи или чтения значений программы

Блок константы	Размещение блока с фиксированным числовым значением
Линии задержки	Создание задержки на один цикл передачи значения с выхода компонента на вход
Блок сетевой переменной	Размещение блоков обмена данными с другими устройствами, подключенными к прибору по сети
Блок чтение/записи в ФБ	Запись/чтение значений отдельных параметров из ФБ в переменную и наоборот
Блок преобразования	Конвертация значений различных типов для передачи
Перенумеровать компоненты	Переназначение порядковых номеров ФБ схемы
Порядок исполнения	Изменение порядка вычисления значений выходов программы

4.3.1 Блок комментариев

Блок комментариев позволяет оставлять записи и комментарии на схеме проекта.

Чтобы добавить текстовый комментарий на схему, следует нажать на кнопку на панели вставки. Затем удерживая ЛКМ начертить прямоугольную область блока комментариев на том месте схемы, где должен располагаться комментарий.

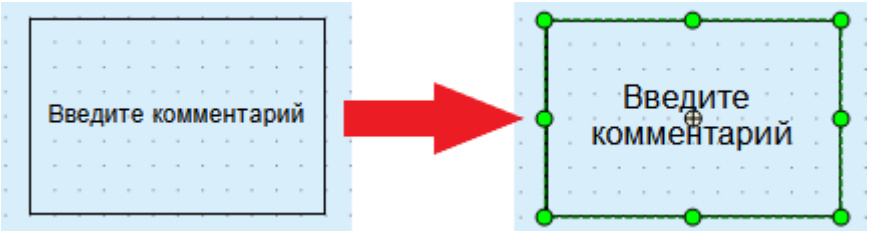


Рисунок 4.16

Для выделенного блока комментариев можно изменить настройки ввода текста на панели свойств.

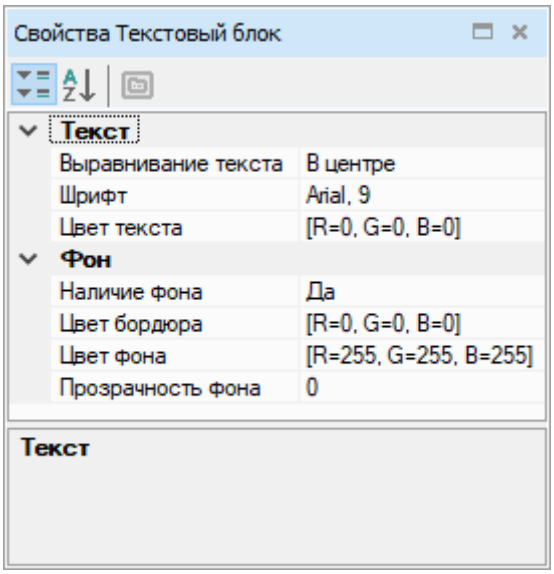


Рисунок 4.17

Чтобы цвет фона блока комментариев был виден, следует задать значение параметру **Прозрачность фона** более 20.

Для ввода текста следует дважды нажать ЛКМ на блок комментариев и ввести новый текст.

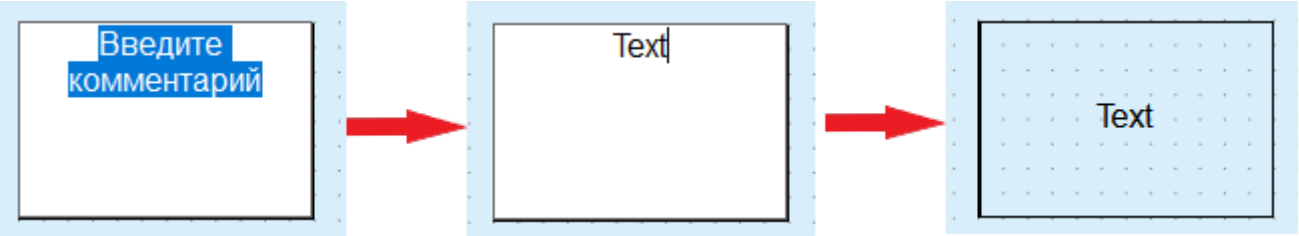


Рисунок 4.18

4.3.2 Блок переменной

Блок переменной служит для использования переменной на схеме проекта.

Чтобы добавить блок переменной на схему, следует нажать на его пиктограмму на панели вставки:

- **входная переменная** — для передачи значения в программу;
- **выходная переменная** — для записи в нее значения из программы.

Затем нажать на схеме в то место, куда нужно поместить блок.

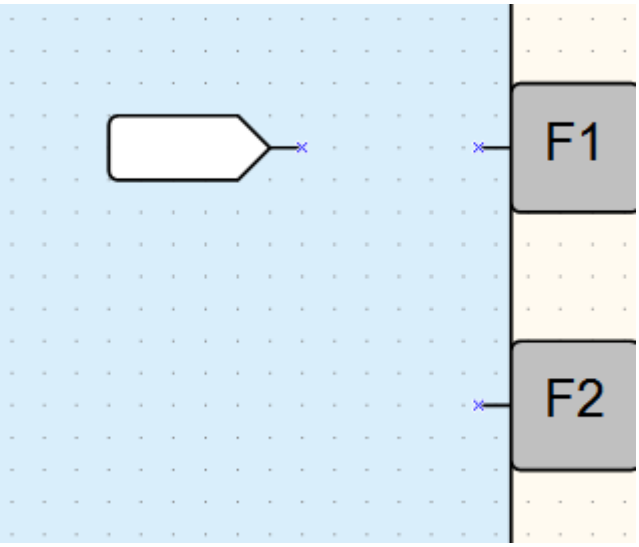


Рисунок 4.19

Блок переменной также можно добавить на схему из панели Переменные.

Для привязки переменной к блоку переменной следует:

1. Выделить блок переменной.
2. В свойствах блока переменной нажать кнопку «**...**» в поле **Переменная** или два раза нажать на блок переменной на схеме.

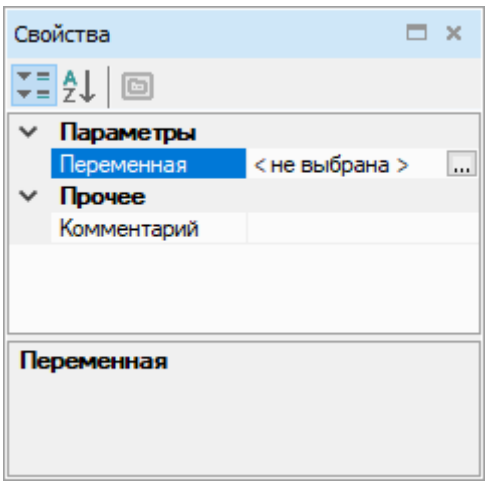


Рисунок 4.20

3. В открывшемся окне Таблицы переменных выбрать или создать новую переменную и нажать кнопку **ОК**. Выбранная переменная будет привязана к блоку.

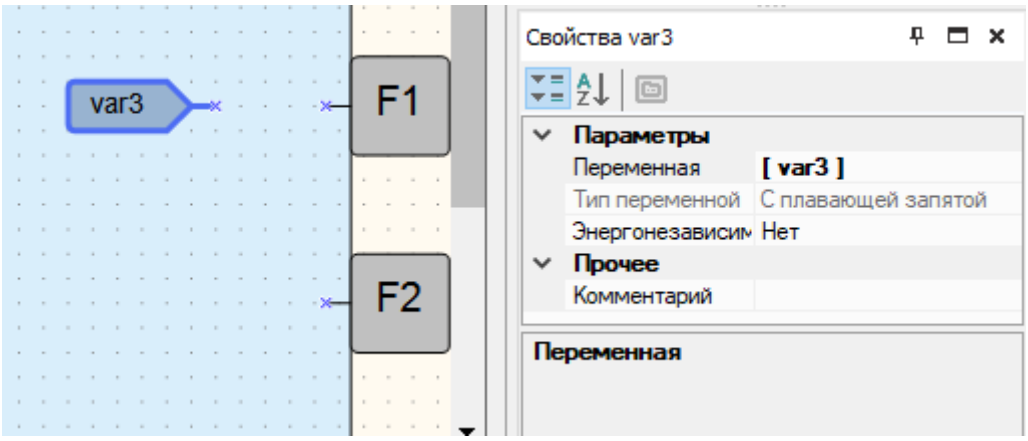


Рисунок 4.21

Если после привязки блок переменной выделяется красным цветом, значит, присутствует ошибка. Сообщение об ошибке отображается в строке состояния.

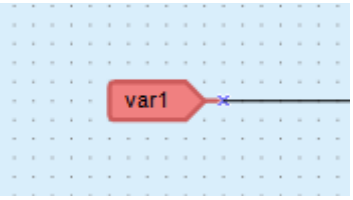


Рисунок 4.22

Для блока выходной переменной, имеющей привязку к параметру прибора, либо элементу визуализации, доступна настройка **Запись в конце цикла**, которая служит для записи значения в переменную после всех операций чтения в рабочем цикле.

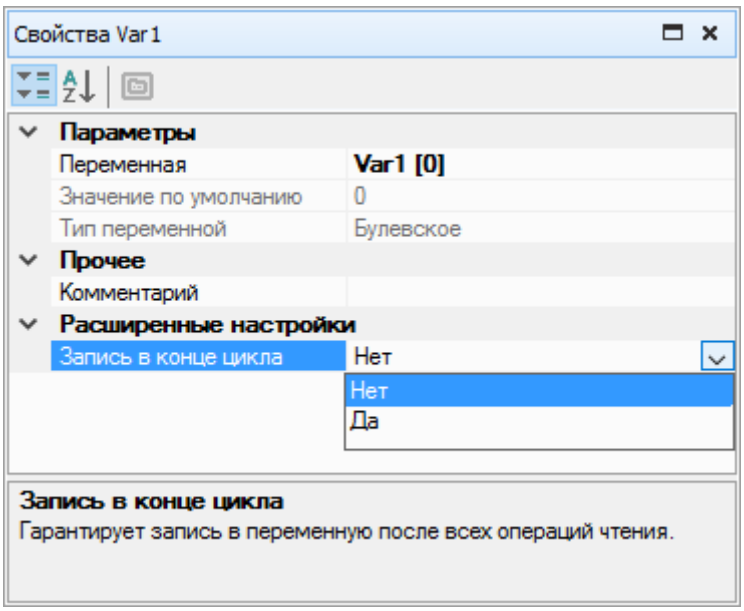


Рисунок 4.23

Значение по умолчанию в настройке **Запись в конце цикла** устанавливается в зависимости от свойства **Энергонезависимость** переменной, привязанной к выходному блоку:

- если чекбокс **Энергонезависимость** не отмечен (галочка отключена), тогда **Запись в конце цикла** принимает значение **"Нет"**;
- если чекбокс **Энергонезависимость** отмечен (галочка включена), тогда **Запись в конце цикла** принимает значение **"Да"**.

Значение по умолчанию в настройке **Запись в конце цикла** устанавливается при следующих изменениях:

- при каждом изменении привязки переменной к выходному блоку;
- при каждом изменении свойства **Энергонезависимость** привязанной переменной к выходному блоку.

Примеры использования настройки параметра **Запись в конце цикла** с рекомендуемыми значениями:

Пример 1

Создана переменная mode, привязанная к элементу визуализации. Переменная инициализируется значением с холста.

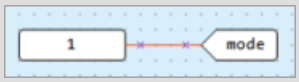
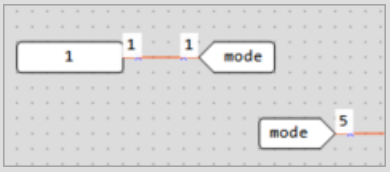
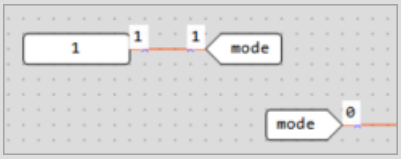


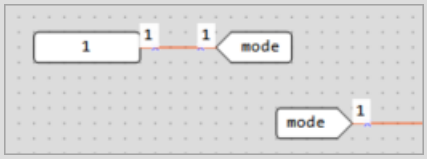
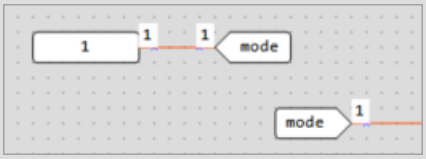
Рисунок 4.24

Необходимо произвести действие с этой переменной в первом рабочем цикле.

В случае настройки **Запись в конце цикла "Да"**:

Энергонезависимая переменная	Не энергонезависимая переменная
Переменная mode в блоке входной переменной будет проинициализирована значением по умолчанию (например, 5):  **Рисунок 4.25**	Переменная mode в блоке входной переменной будет проинициализирована нулевым значением:  **Рисунок 4.26**

В случае настройки **Запись в конце цикла "Нет"**:

Энергонезависимая переменная	Не энергонезависимая переменная
Переменная mode в блоке входной переменной будет проинициализирована значением с холста:  **Рисунок 4.27**	Переменная mode в блоке входной переменной будет проинициализирована значением с холста:  **Рисунок 4.28**

Рекомендуется использовать **Запись в конце цикла "Нет"**.

Пример 2
Созданы две стандартные переменные screen_var1 и screen_var2, привязанные к разным элементам визуализации, доступным для изменения с экрана прибора.
Необходимо произвести синхронизацию значений этих переменных. Для этого можно использовать макросы SelChg/SelChgF из менеджера компонентов.
Макрос SelChgF предназначен для изменения переменной тип int/float с помощью двух переменных. На выходе макроса записывается последнее изменение одной из двух переменных.
В случае настройки **Запись в конце цикла "Нет"** (по умолчанию для не энергонезависимых переменных):

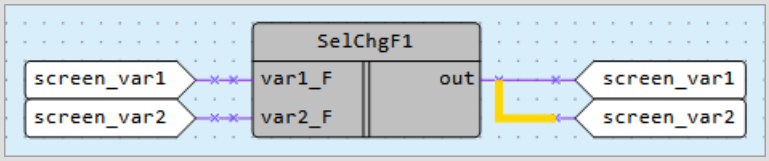


Рисунок 4.29

У одной из переменных (в данном примере screen_var2) появляется циклическая связь (подсвечивается желтым цветом), в результате чего недоступно изменение значения переменной с экрана, т.к. переменная screen_var2 будет принимать значение переменной screen_var1.
В случае настройки **Запись в конце цикла "Да"** (по умолчанию для энергонезависимых переменных) для переменной screen_var2 циклическая связь пропадет, обе переменные станут доступны для изменения с экрана:

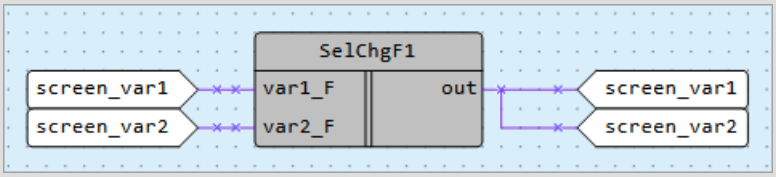


Рисунок 4.30

Рекомендуется использовать **Запись в конце цикла "Да"**.

Указанная в примерах информация имеет рекомендательный характер, у пользователя есть возможность установить именно то значение, которое обеспечит необходимое поведение в данной части схемы.

4.3.3 Блок константы

Для включения в программу жестко заданного числового значения используется блок константы. Для размещения блока константы на схеме следует нажать на кнопку C на панели вставки, затем – в место на схеме, где должен быть расположен блок.

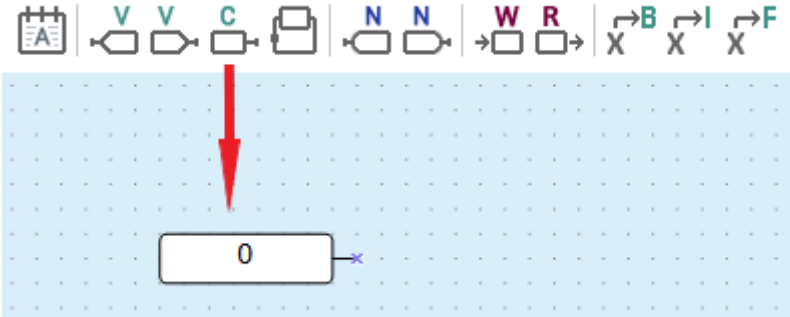


Рисунок 4.31

На панели свойств выбранного блока указывается тип данных для константы и вводится значение. Значение будет отображено в блоке на схеме.

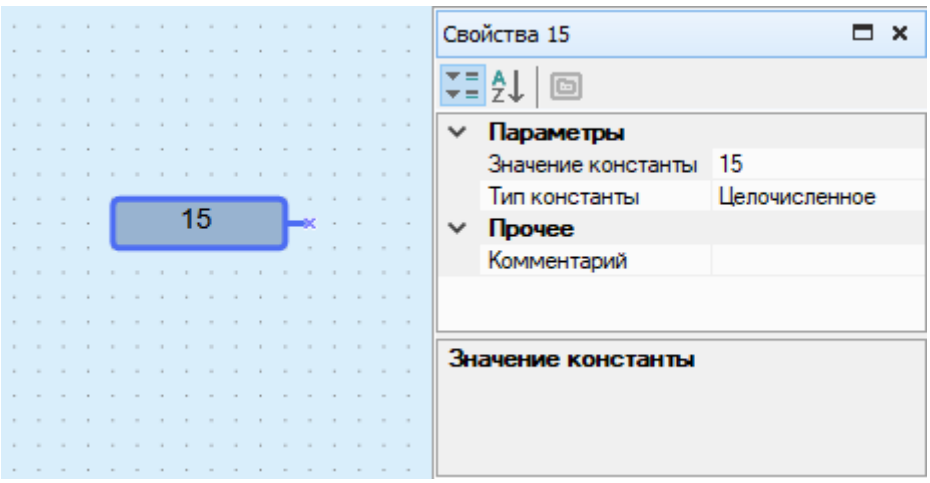


Рисунок 4.32

Значение константы можно отредактировать двойным нажатием на блок константы или выбором **Изменить значение** в контекстном меню блока.

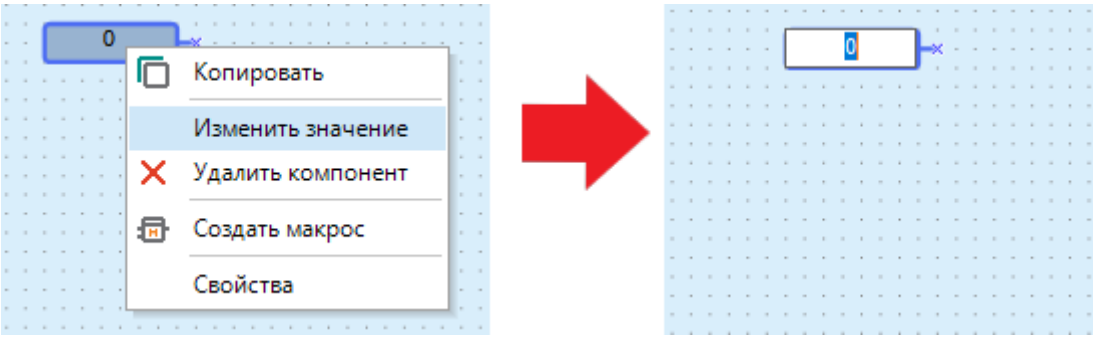


Рисунок 4.33

Допустимые значения для типов константы:

Булевское	от 0 до 1
Целочисленное	от 0 до 4294967295
С плавающей запятой	от –3,402823E+38 до 3,402823E+38

Значение константы не подвергается изменениям на всем протяжении работы программы.

4.3.4 Линия задержки

Линия задержки используется для создания задержки на один цикл передачи значения с выхода компонента на вход.

Для создания линии задержки следует нажать на кнопку на панели вставки и построить линию между выходом и входом компонента/компонентов проекта. Линия задержки отобразится на схеме пунктирной линией со стрелкой, цвета соответствующего типу данных переменной.

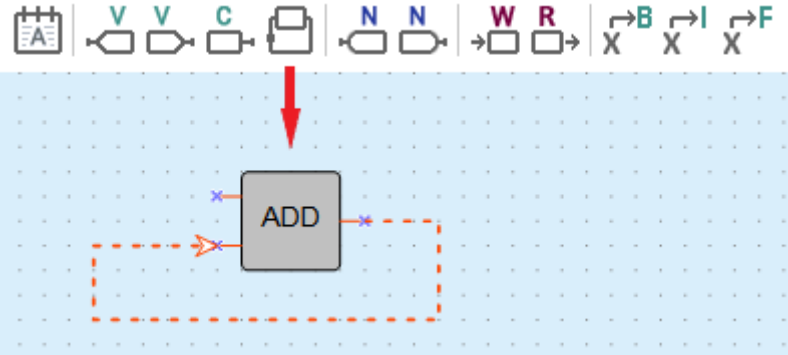


Рисунок 4.34

Пример

На вход **I1** блока арифметического сложения (целочисленный тип) подается константа, значение которой равно 1. На вход **I2** по линии задержки подается значение с выхода блока **Q**, которое было вычислено в предыдущем цикле.

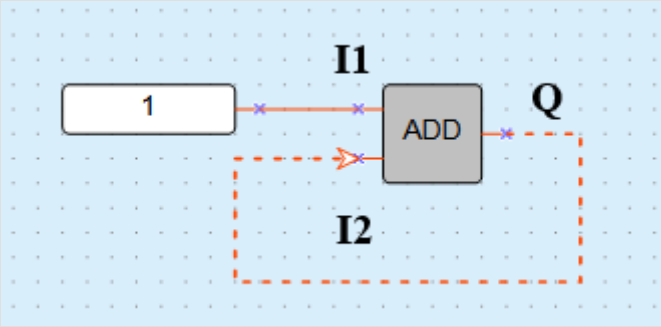


Рисунок 4.35

Значения на входах и выходах блока:

№ цикла	1	2	3	4	5	6	7	8	9	10
I2	0	1	2	3	4	5	6	7	8	9
Q	1	2	3	4	5	6	7	8	9	10

4.3.5 Блок сетевой переменной

Блоки сетевых переменных служат для передачи и получения значений по интерфейсам связи и использования их в программе.

Для добавления на схему блока сетевой переменной следует нажать на кнопку:

- N – передача значения из программы по сети;
- N – получение значения в программу по сети.

Привязка переменной к блоку сетевой переменной

Для привязки переменной к блоку сетевой переменной следует:

1. Выбрать на схеме блок сетевой переменной.
2. Нажать на кнопку «**...**» в поле **Переменная** в свойствах блока сетевой переменной или два раза нажать на блок сетевой переменной на схеме.

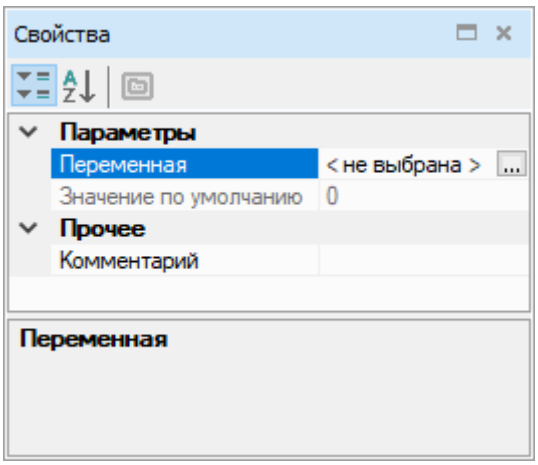


Рисунок 4.36

3. В открывшейся таблице переменных 6 выбрать или создать новую переменную для привязки и нажать кнопку **ОК**.

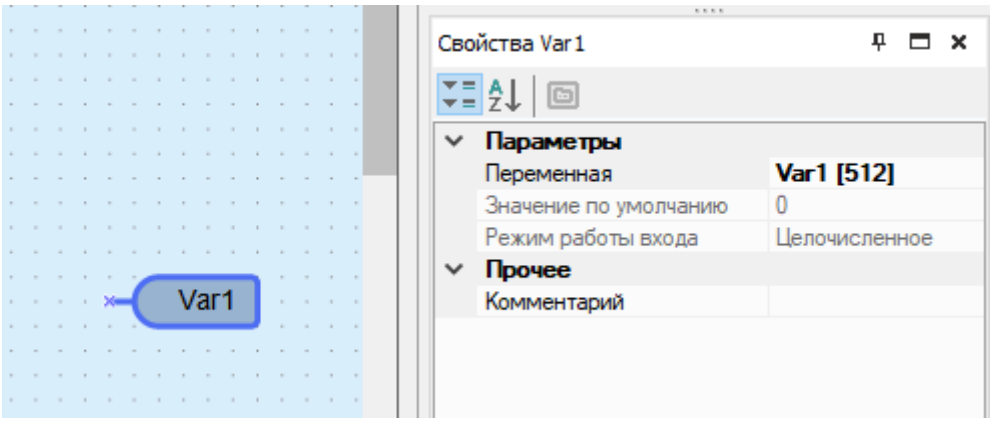


Рисунок 4.37

4. Присоединить выход/вход блока сетевой переменной к необходимому компоненту.

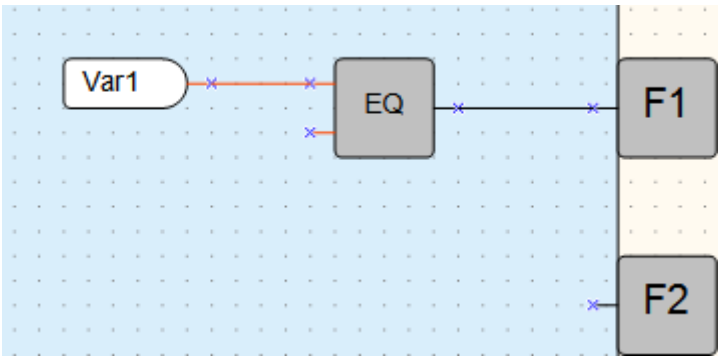


Рисунок 4.38

Вышеописанный алгоритм работы с переменными интерфейсов связи удобен, если пользователь начинает работу с проектом с настройки прибора. Если список параметров Мастера сети Modbus уже известен, работу удобнее начинать с определения переменных в таблице переменных 6. Подробнее см. раздел Работа по протоколу Modbus 5.3.2.

ПРИМЕЧАНИЕ
Нельзя привязать переменную к блоку, если в настройках прибора 5 нет интерфейсов связи.

Расширенные настройки

Для приборов второго поколения доступна настройка допустимого отклонения для переменных с плавающей точкой в блоке передачи значения из программы по сети. Если входящее в блок значение не превышает допустимое, то значение переменной не перезаписывается каждый цикл, что позволяет экономить аппаратные ресурсы прибора. Допустимое отклонение задается в свойствах блока сетевой переменной.

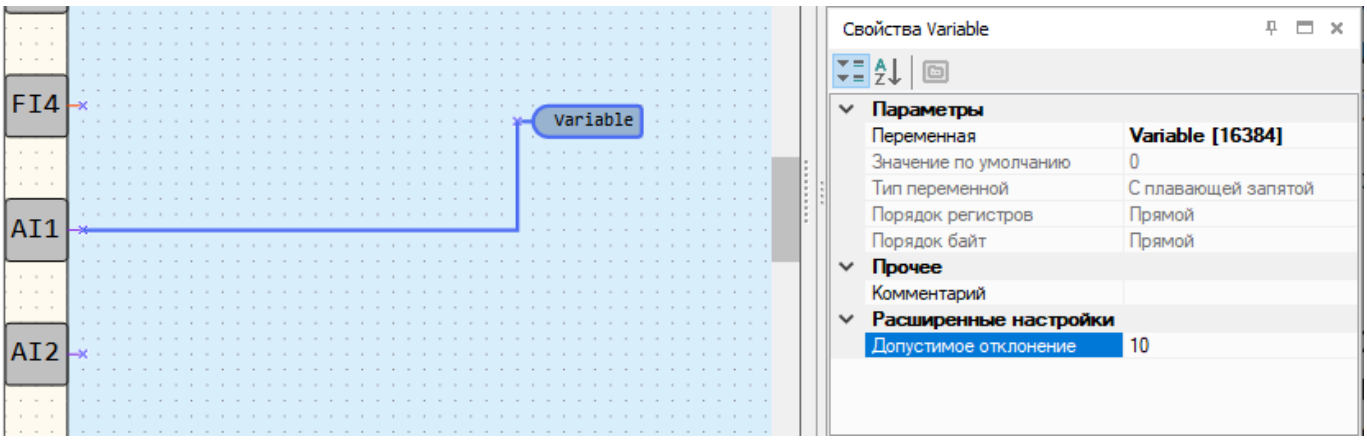


Рисунок 4.39

4.3.6 Блок чтения/записи в ФБ

Блок чтения/записи в ФБ используется для задачи или чтения значения параметра ФБ в процессе выполнения программы.

На схему могут быть добавлены блоки:

- W — записи значения в ФБ;
- R — чтения значения из ФБ.

Пример

В зависимости от значения на входе **I1** в функциональном блоке **BLINK1** значение параметра **Длительность вкл. состояния** будет иметь значение **2** или **10**.

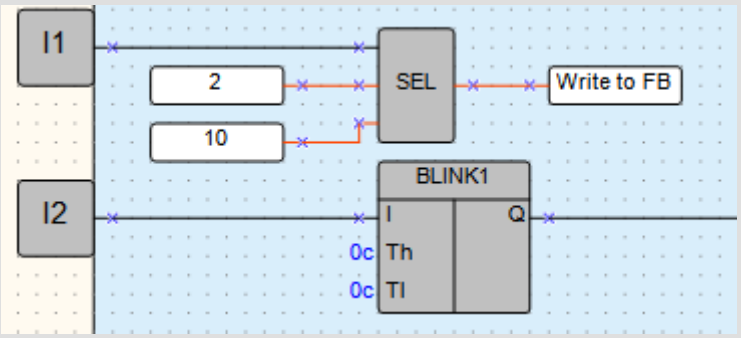


Рисунок 4.40

Для настройки блока записи значения в ФБ следует на панели свойств блока выбрать ФБ и имя переменной, в которую будет записываться значение.

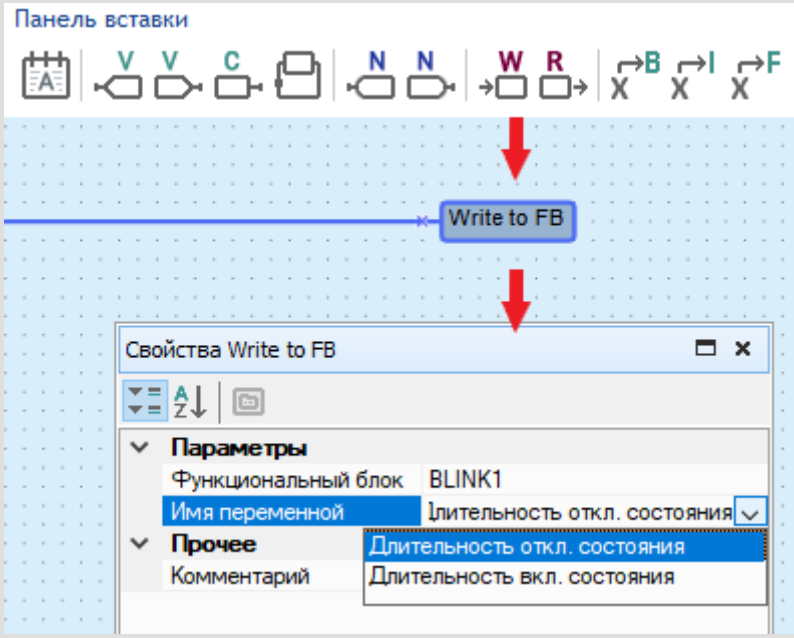


Рисунок 4.41

4.3.7 Блок преобразования

Линия связи между компонентами программы может быть создана только для входа и выхода одного и того же типа: булевского, целочисленного, с плавающей запятой или массива (с одинаковым количеством и типом элементов). Для создания линии связи между входом и выходом разных типов следует использовать блоки преобразования.

Использование блоков преобразования с переменными типа массив недоступно. Возможно преобразование элементов массива.

Чтобы добавить блок преобразования на схему, следует выбрать нужный тип блока на панели вставки и затем нажать в то место схемы, где он должен быть расположен.

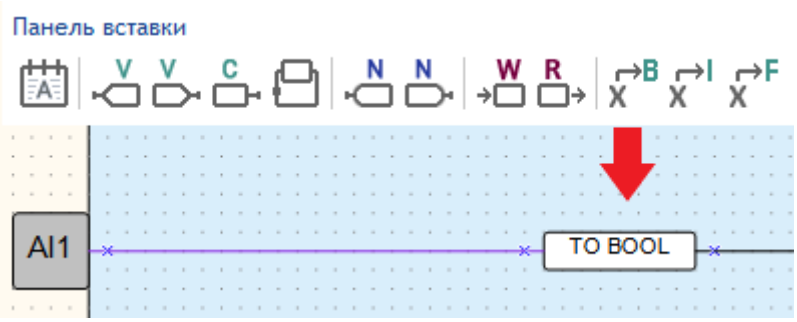


Рисунок 4.42

Типы блоков преобразования:

	Преобразователь в булевское значение	Конвертация значения сигнала в булевское значение. Если значение на входе блока больше либо равно 1, то на выходе будет 1 (**True**). В остальных случаях на выходе будет 0 (**False**).
	Преобразователь в целочисленное значение	Конвертация значения сигнала в целочисленное значение. Если на входе блока булевское значение **0 (False)**, то на выходе будет **0**, если на входе **1 (True)**, то на выходе будет **1**. Если на входе блока значение числа с плавающей запятой, то на выходе оно округляется с недостатком. Отрицательные значения с плавающей запятой конвертируются в **0**.
	Преобразователь в число с плавающей запятой	Конвертация значения сигнала в число с плавающей точкой. Для преобразования целочисленного в число с плавающей точкой добавляется дробная часть (int = 5 → float = 5,0000). Если на входе блока булевское значение **0 (False)**, то на выходе будет **0,0000**, если на входе блока **1 (True)**, то на выходе будет **1,0000**.

4.3.8 Перенумеровать компоненты

Для переназначения порядковых номеров компонентов на схеме следует нажать кнопку на панели инструментов или выбрать в главном меню **Сервис** → **Перенумеровать компоненты**. Однотипные компоненты нумеруются по порядку сверху вниз и слева направо.

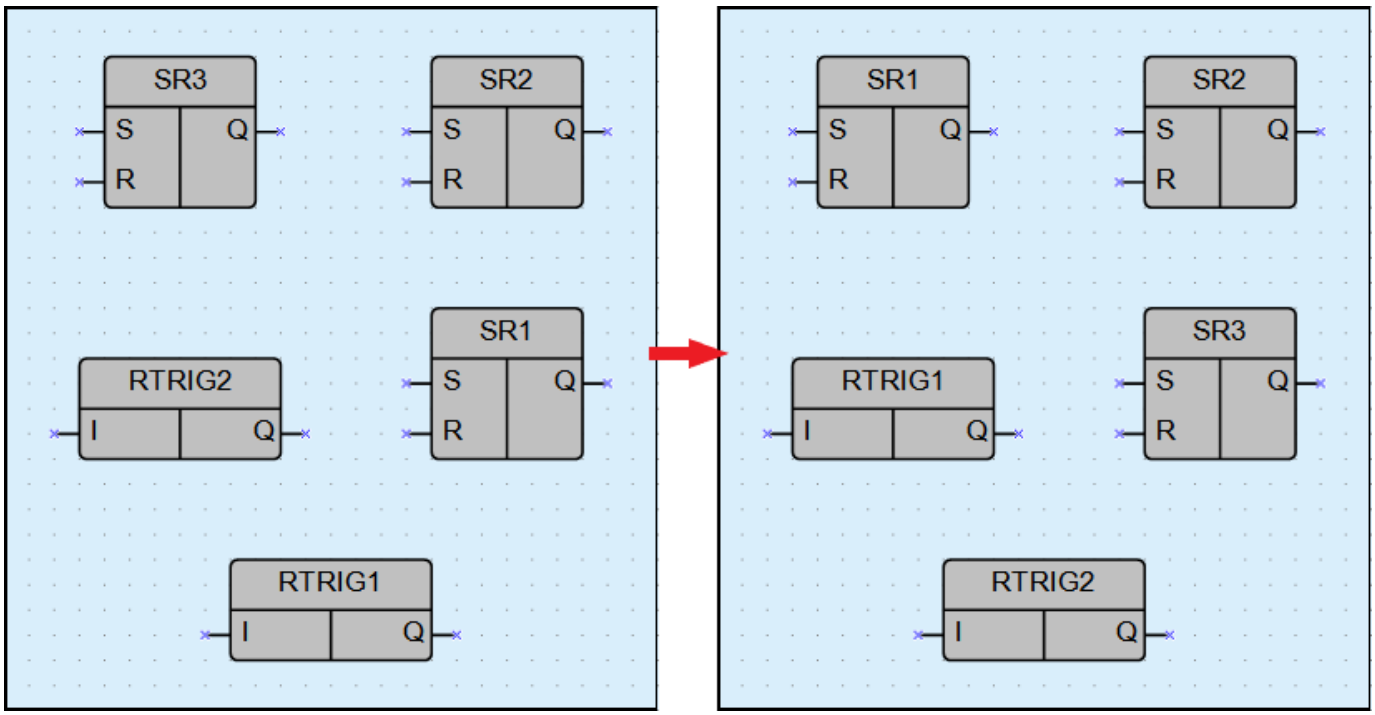


Рисунок 4.43

4.3.9 Порядок исполнения

Значения всех выходов и обратных связей в программе вычисляются в определенном порядке. Для изменения порядка вычисления следует нажать на кнопку Q на панели отладки и в выпадающем меню выбрать **Обратные связи** или **Выходы**.

Owen Logic перейдет в режим настройки порядка исполнения – на выходах и обратных связях отобразятся порядковые номера очереди исполнения.

Для изменения очередности исполнения следует дважды нажать ЛКМ на нужный выход или обратную связь и ввести число, какой по счету должна выполняться данная операция.

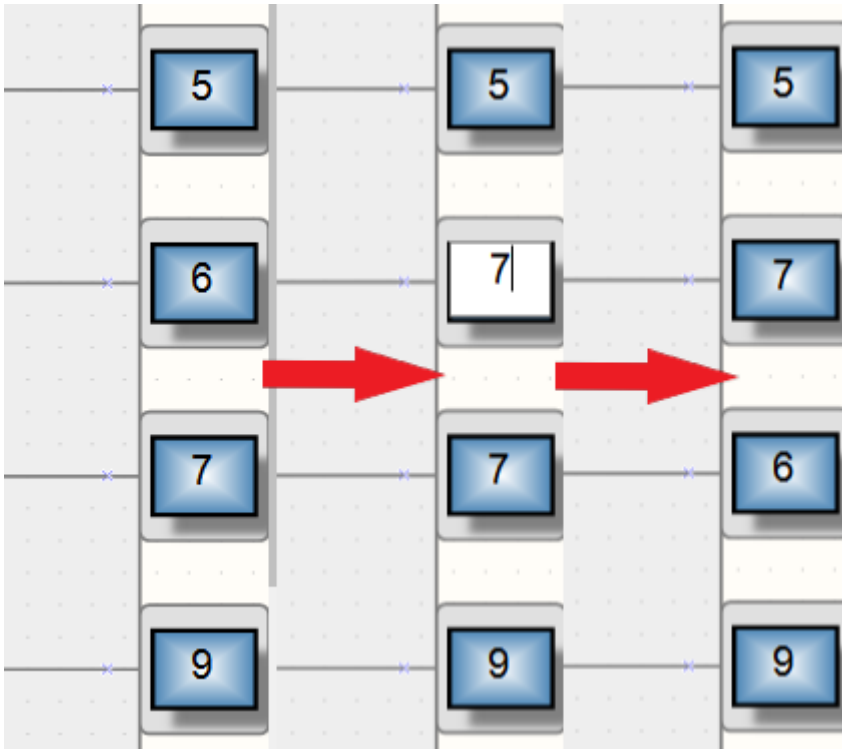


Рисунок 4.44

После установки необходимой очередности следует отключить режим настройки порядка исполнения повторным нажатием на кнопку .

4.4 Программирование дисплея прибора

Если прибор оснащен дисплеем, то дисплей может быть использован для отображения различной информации, согласно заданному алгоритму в программе. На экране может отображаться текст, графические компоненты и значения переменных программы. Для программирования экрана используется Менеджер экранов. Чтобы открыть менеджер экранов, следует нажать на заголовок вкладки в левой части рабочей области.

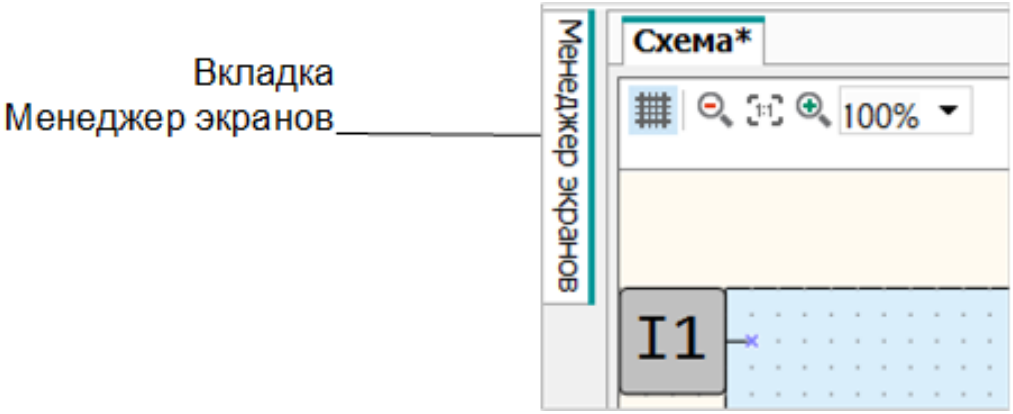


Рисунок 4.45

Менеджер экранов для приборов с графическим цветным и монохромным текстовым ЖКИ имеет следующие различия:

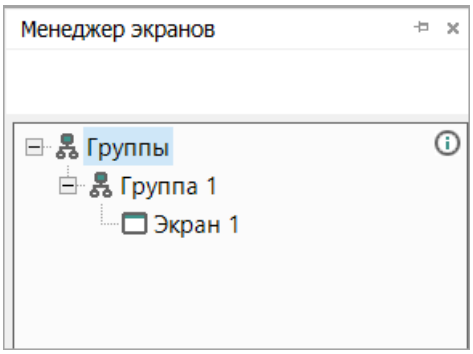


Рисунок 4.46 – Менеджер экранов для приборов с монохромным текстовым ЖКИ

По умолчанию в Группу 1 добавлен Экран 1:

- редактирование Группы 1 доступно;
- удаление Группы 1 не доступно.

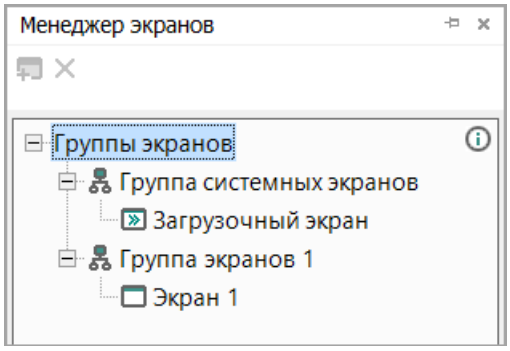


Рисунок 4.47 – Менеджер экранов для приборов с графическим цветным ЖКИ

По умолчанию в Группу системных экранов добавлен Загрузочный экран:

- Группа системных экранов и Загрузочный экран не доступны для редактирования и удаления.

По умолчанию в Группу экранов 1 добавлен Экран 1:

- редактирование Группы экранов 1 доступно;
- удаление Группы экранов 1 не доступно.

Добавление экранов

В менеджере экранов можно добавить несколько экранов. Возможно настроить переход между экранами на дисплее прибора по нажатию кнопки и по изменению переменной. Настройка переходов отличается для монохромного текстового ЖКИ и графического цветного ЖКИ.

Для добавления экрана следует нажать ПКМ на элемент **Группа 1** и в контекстном меню выбрать **Добавить экран**.

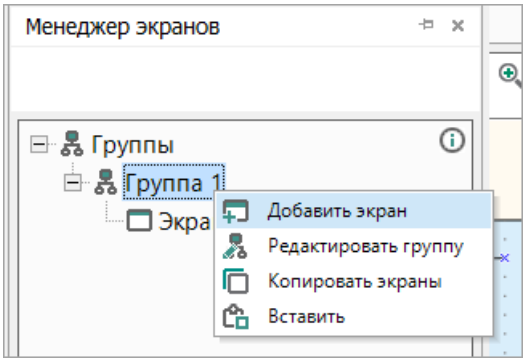


Рисунок 4.48

Для удаления экрана следует нажать ПКМ на нужный экран и в контекстном меню выбрать **Удалить**.

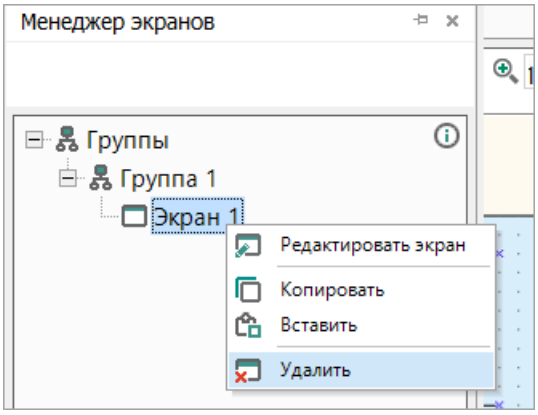


Рисунок 4.49

Для смены позиции экрана используйте метод drag&drop, удерживая клавишу **Shift**.

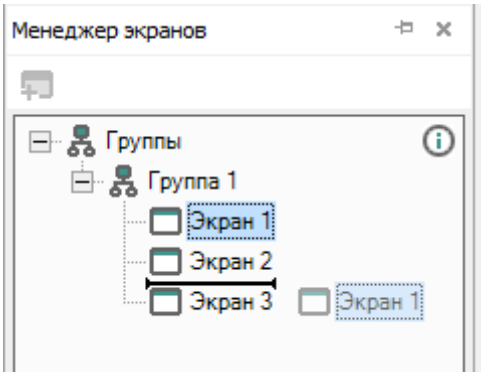


Рисунок 4.50

При перемещении новое положение будет отображаться в виде горизонтального маркера.

Редактор экрана

Функционал редактора экрана зависит от типа дисплея в приборе:

- монохромный текстовый ЖКИ;
- графический цветной ЖКИ.

4.4.1 Монохромный текстовый ЖКИ

Параметры экрана

Параметры экрана отображаются в поле свойств **Менеджера экранов**.

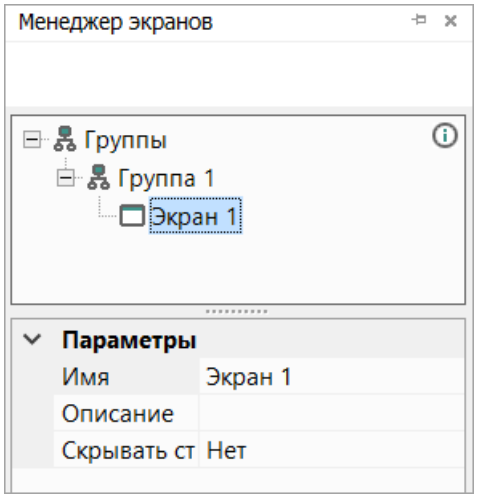


Рисунок 4.51

- **Имя** – для отображения в **Менеджере экранов** и в заголовке редактора экранов;
- **Описание** – текстовое описание экрана;
- **Скрывать строки** – скрытие/отображение пустых строк на экране прибора.

Для отображения **параметров** экрана на панели свойств следует дважды нажать на нужный экран в **Менеджере экранов**.

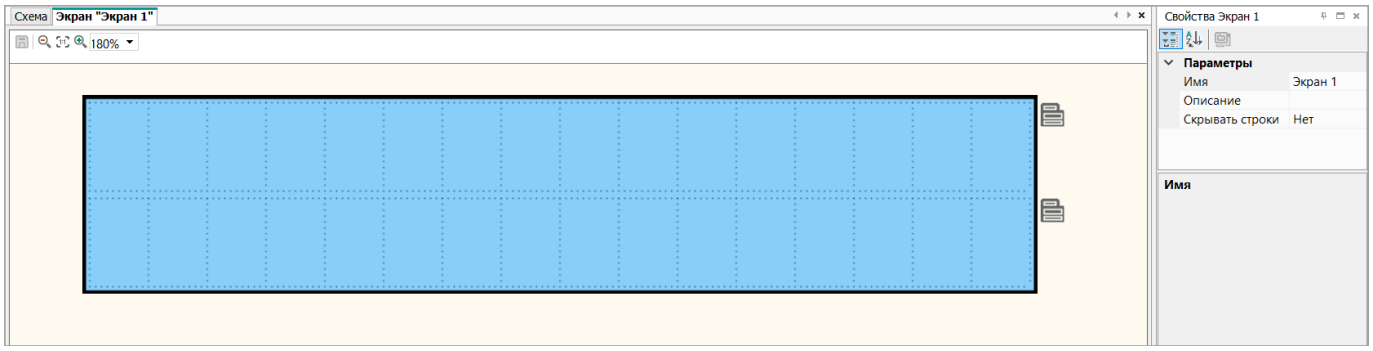


Рисунок 4.52

Редактор группы экранов

Для приборов с монохромным текстовым ЖКИ также доступен редактор группы экранов, который открывается выбором команды **Редактировать группу** в контекстном меню группы экранов в **Менеджере экранов**.

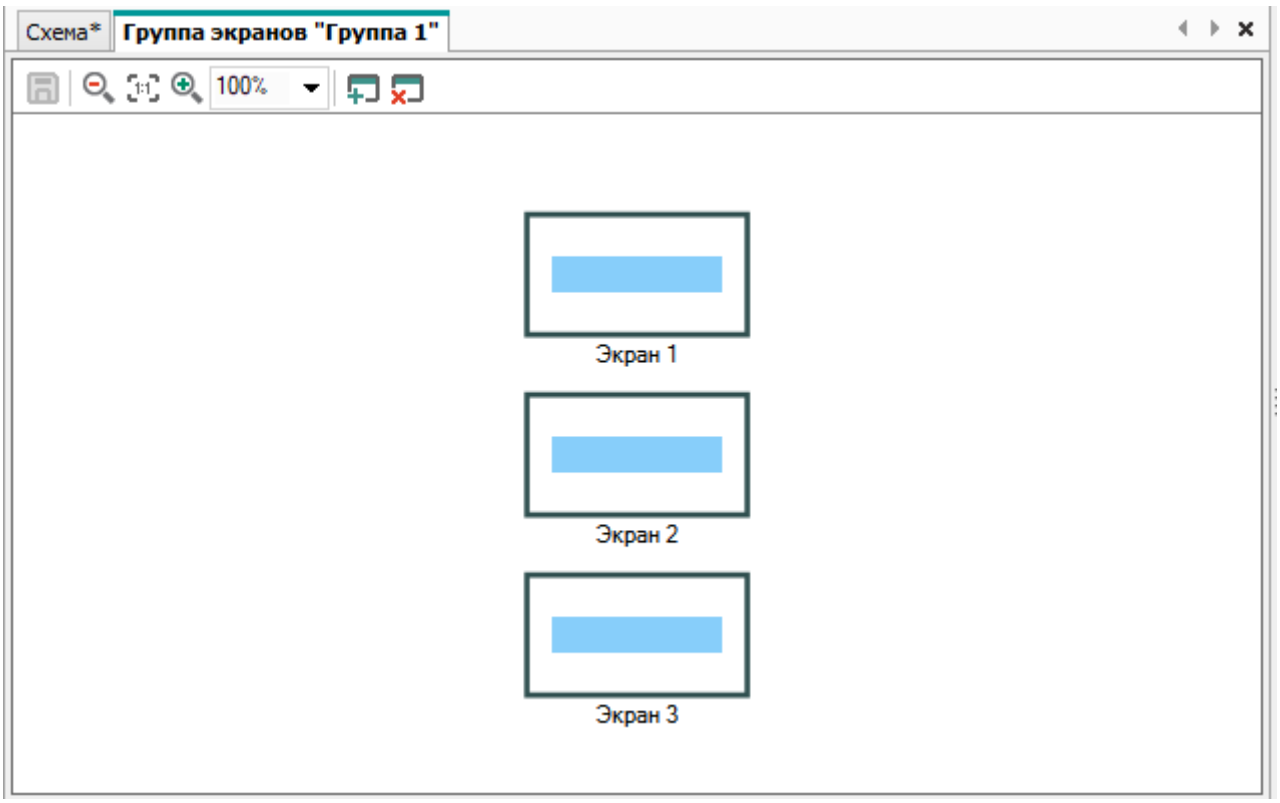


Рисунок 4.53

В верхней части редактора экранов и группы экранов расположены кнопки:

	Сохранить документ
	Уменьшить масштаб
	Оригинальный размер
	Увеличить масштаб

Масштаб может быть изменен с помощью выпадающего меню справа от кнопок.

Редактор экранов

Для настройки вывода информации на экран следует открыть редактор экрана двойным нажатием по экрану в менеджере экранов или нажатием на кнопку **Редактировать экран** из контекстного меню экрана.

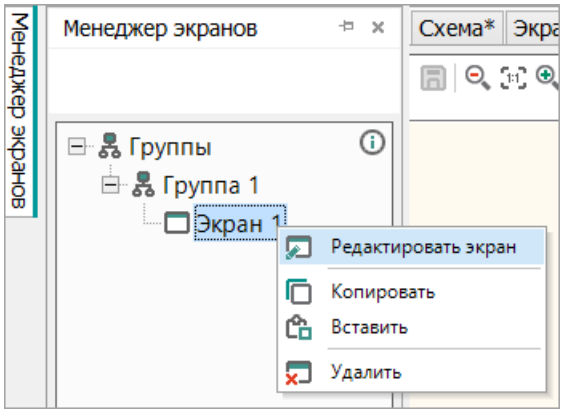


Рисунок 4.54

Экран может содержать несколько строк, переход к которым осуществляется с помощью кнопок прибора. Для добавления или удаления строк следует нажать на кнопку **Контекстное меню строки** и выбрать необходимую команду.

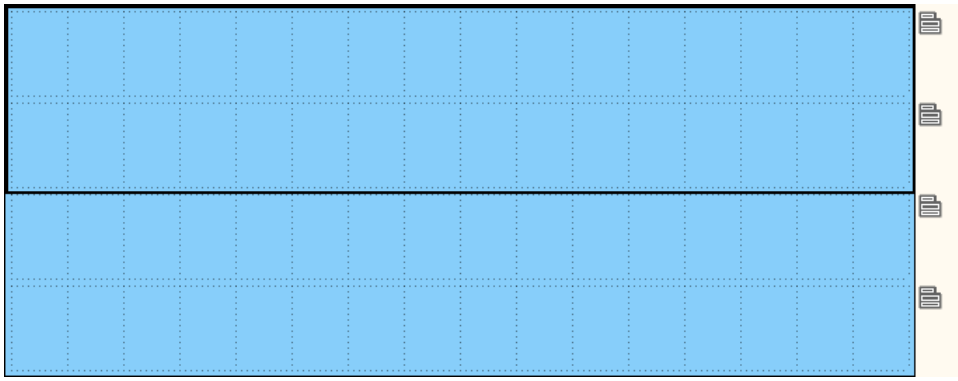


Рисунок 4.55

В строках располагаются элементы управления, которые перетаскиваются из панели **Библиотека компонентов** методом drag & drop. Описания блоков приведены в разделе Элементы управления.

ПРИМЕЧАНИЕ
Набор символов реализован в рамках кодировки Windows-1251.

Копирование экранов

В менеджере экранов можно копировать экраны для вставки в текущий или другой проект. Для копирования выделенных экранов следует выбрать команду **Копировать** в контекстном меню экрана или группы экранов, либо нажать сочетание клавиш **Ctrl + C**. Несколько экранов можно выделить с помощью зажатой клавиши **Ctrl** или **Shift**.

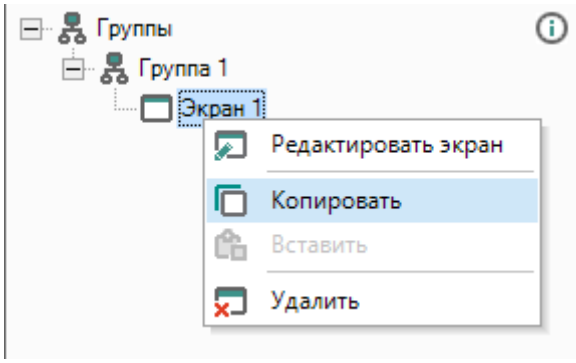


Рисунок 4.56

Для вставки скопированных экранов следует выбрать в контекстном меню экрана или группы экранов команду **Вставить**, либо нажать сочетание клавиш **Ctrl + V**.

Вместе с экраном копируются все размещенные на экране элементы управления и свойства экрана. Привязанные к экрану переменные копируются в другой проект согласно разделу Копирование переменных.

Если между выделенными экранами есть переходы, то они копируются вместе с экранами. Если у экрана есть переход за пределы выделенной области, то такой переход удаляется.

Переход к экрану

Если в проекте участвуют два или более экранов, то необходимо настроить переходы (условия переключения) между ними.

Для создания перехода следует:

1. Нажать ПКМ по элементу **Группа 1** в дереве менеджера экранов и выбрать **Редактировать группу** в контекстном меню. Откроется вкладка редактора группы экранов.
2. Выбрать стартовый экран в редакторе группы экранов.
3. В свойствах экрана нажать «**…**» в параметре **Переход к экрану**. Откроется меню создания перехода.
4. В выпадающем меню **Список экранов** выбрать другой экран для перехода к нему.

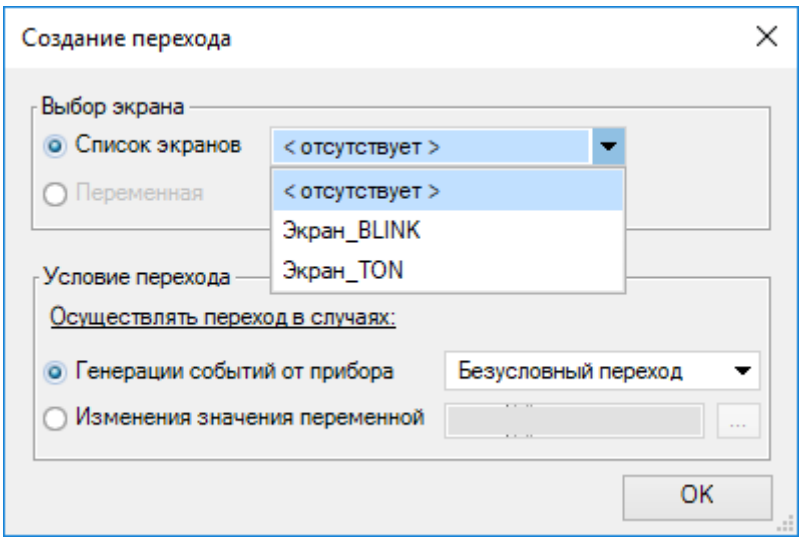


Рисунок 4.57

5. В поле **Условие перехода** выбрать пункт **Генерация событий от прибора** и в выпадающем меню указать кнопку прибора для выполнения перехода или выбрать пункт **Изменения значения переменной** для указания переменной булевского типа, в случае изменения значения которой будет выполнен переход.

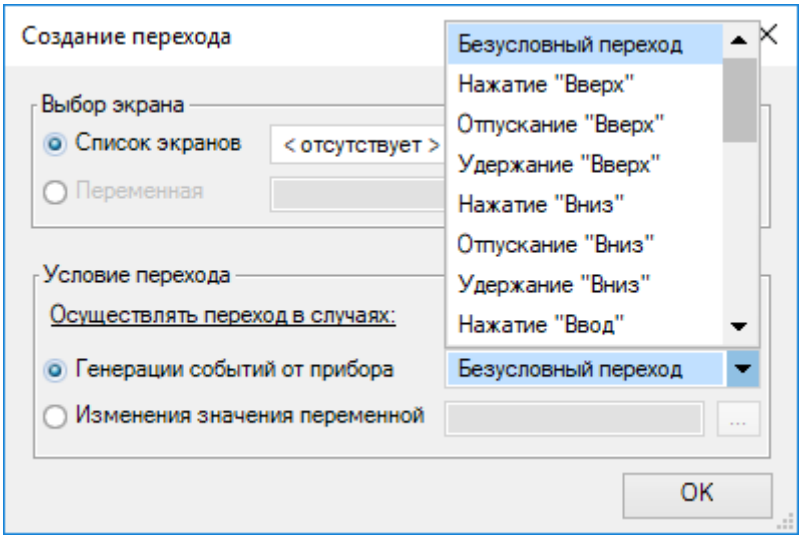


Рисунок 4.58

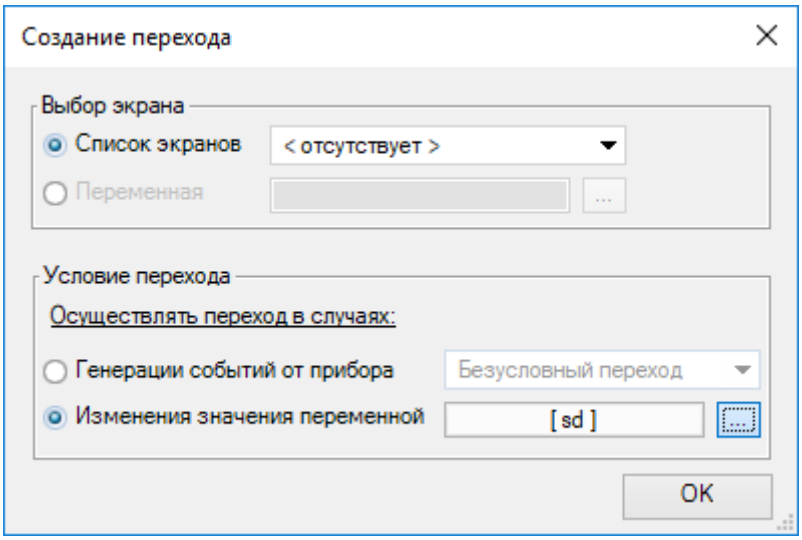


Рисунок 4.59

ПРИМЕЧАНИЕ
Безусловный переход служит для настройки перехода, если ни одно из событий или изменений значения переменной не происходит, например, для перехода на экран аварий.

6. После создания переход будет отображен на схеме в редакторе группы экранов и в свойствах экрана.

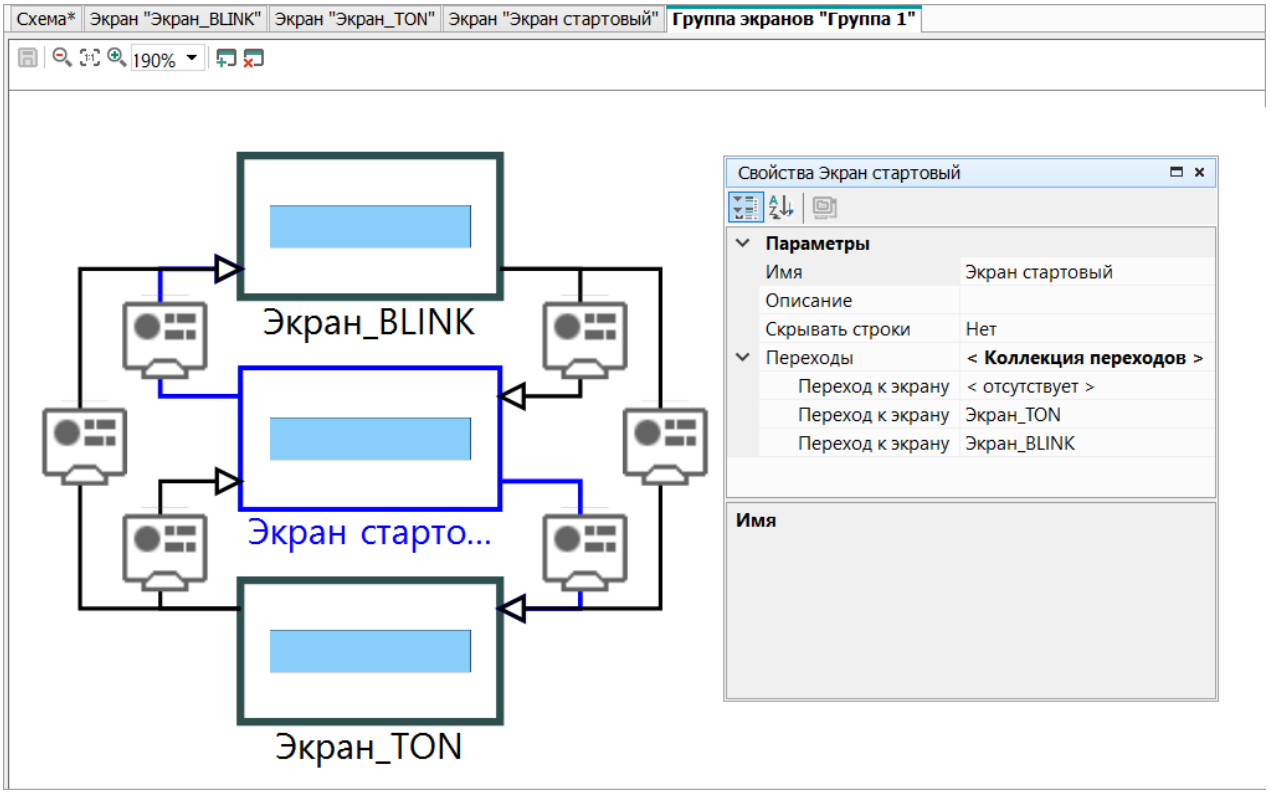


Рисунок 4.60

Переход от экрана к экрану может происходить по различным событиям, для этого можно настроить несколько (**коллекцию**) переходов. Переходы добавляются аналогично описанию выше.

Для редактирования или удаления перехода следует выбрать его в свойствах экрана и настроить соответствующие параметры. Переход будет удален, если в поле **Список экранов** меню создания перехода будет стоять значение **< отсутствует >**.

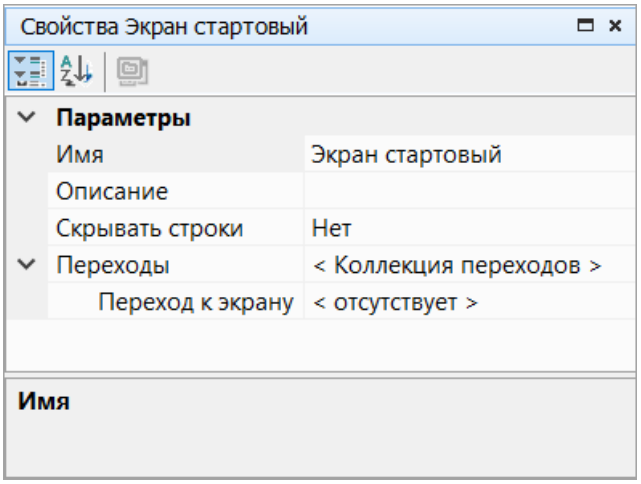


Рисунок 4.61

4.4.2 Графический цветной ЖКИ

Для приборов с графическим цветным ЖКИ вывод информации на экран программируется с помощью графических элементов.

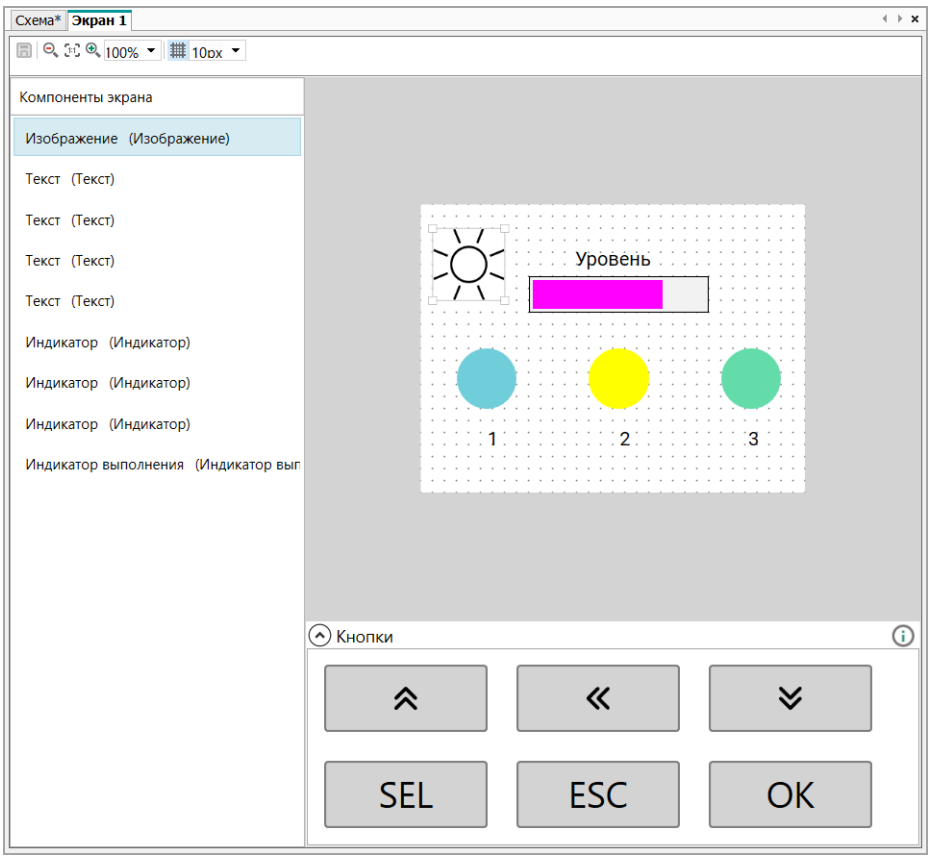


Рисунок 4.62

Области редактора экрана:

- **панель элементов экрана** служит отображения списка расположенных на поле компонентов. На панели элементов экрана также можно менять порядок графических элементов, задавать им имена;
- **панель инструментов экрана** содержит кнопки управления экраном: масштабирование, включение/выключение и настройка шага сетки, предназначенной для упрощения выравнивания элементов;
- **поле экрана** служит для расположения графических элементов;
- **кнопки** служат для программирования кнопок прибора.

Графические элементы для расположения в поле экрана выбираются из **Библиотеки компонентов**. Описания элементов приведены в разделе Базовые элементы.

В режиме симулятора доступно окно симуляции экрана. Сетка при симуляции не отображается.

Панель инструментов экрана

	Сохранить документ	Сохранение документа не поддерживается
	Уменьшить масштаб	Уменьшение масштаба на 10% от первоначального
1:1	Оригинальный размер	Возврат к исходному масштабу
	Увеличить масштаб	Увеличение масштаба на 10% от первоначального
100%	Выбор масштаба	Выбор масштаба. Возможен ввод значения в диапазоне от 10 до 2000%
	Включить/ Выключить сетку	При включенной сетке на экране отображается сетка из точек. Элементы визуализации при размещении привязываются к сетке
8px	Шаг сетки	Выбор шага сетки. Возможен ввод значения в диапазоне от 2 до 1000px

Поле экрана

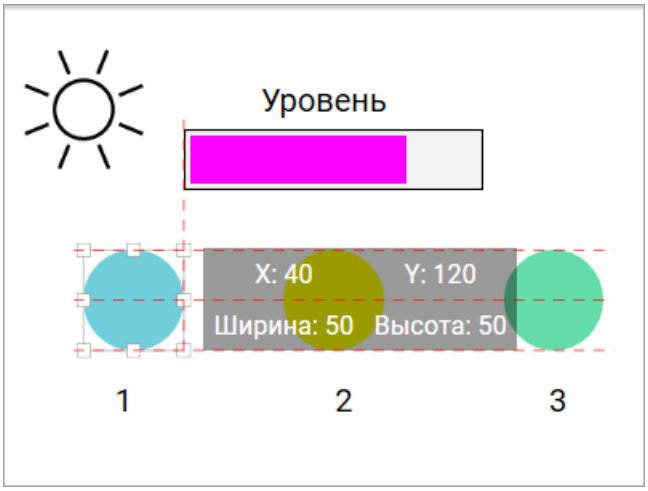


Рисунок 4.63

Перемещать элементы визуализации по полю экрана можно с помощью:

- мыши (методом drag&drop) — во время перемещения на поле экрана отображаются подсказки с координатами и размерами элемента. Если отображение сетки выключено, то при перемещении отображаются направляющие линии относительно других элементов. При включенной сетке графический элемент сдвигается на шаг сетки;
- стрелок на клавиатуре — при включенной сетке графический элемент сдвигается на шаг сетки;
- с помощью сочетания клавиш.

Для графических элементов и групп графических элементов доступны следующие действия в контекстном меню:

- копирование;
- вставка;
- удаление;
- перемещение на передний и задний план;
- выравнивание.

Двойное нажатие ЛКМ по графическим элементам:

- Текст и Динамический текст — активирует режим ввода текста;
- Линия — действие на двойное нажатие ЛКМ отсутствует;
- Кнопка — открывает окно настройки действий;
- Группа переключателей — открывает редактор списка переключателей;
- Изображение — открывает окно выбора пользовательского изображения;
- Меню — открывает окно редактирования строк;
- График — открывает окно Список линий График, для настройки отображения графика;
- для остальных элементов открывается окно привязки переменной.

Свойства графических элементов

Свойства графического элемента отображаются при:

- выделении графического элемента, расположенного на поле экрана;
- выделении графического элемента, расположенного на панели Компоненты экрана.

Общие свойства для всех графических элементов:

- **имя элемента**;
- **размер** – ширина и высота в пикселях;
- **координаты** – по осям X и Y, точкой отсчета является верхний левый край элемента.

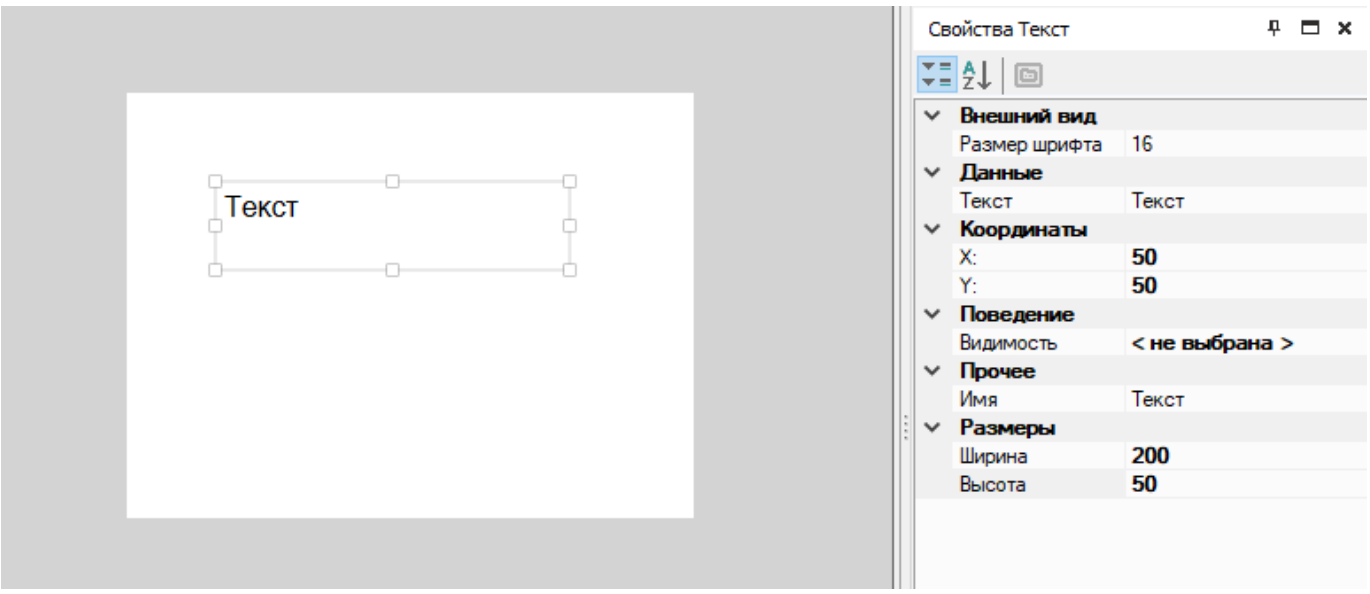


Рисунок 4.64

Группы экранов

Для папки **Группы экранов** в поле свойств **Менеджера экранов** доступна настройка параметров, которые будут применены ко всем элементам визуализации всех экранов, входящих в эту группу. Изменение настроек на уровне группы автоматически влияет на все графические элементы, расположенные на всех экранах группы.

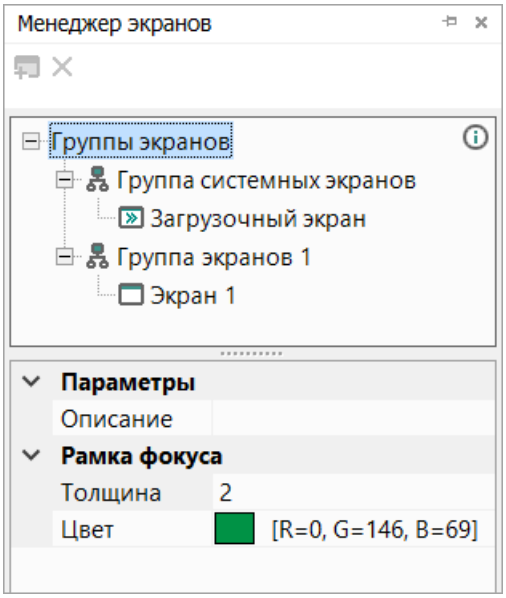


Рисунок 4.65

- **Параметры:**

  Описание — текстовое описание группы экранов.

- **Рамка фокуса:**

  Толщина — толщина рамки графического элемента в фокусе в режиме симуляции и на экране прибора;

  Цвет — цвет рамки графического элемента в фокусе в режиме симуляции и на экране прибора.

Загрузочный экран

Для приборов с графическим цветным ЖКИ доступна загрузка пользовательского изображения для экрана, отображающегося в момент инициализации устройства после подачи питания.

ПРИМЕЧАНИЕ
По умолчанию на Загрузочном экране отображается предустановленное изображение.

Для добавления пользовательского изображения дважды нажмите ЛКМ на элемент **Загрузочный экран** во вкладке **Менеджер экранов.** Откроется окно **Свойств загрузочного изображения**:

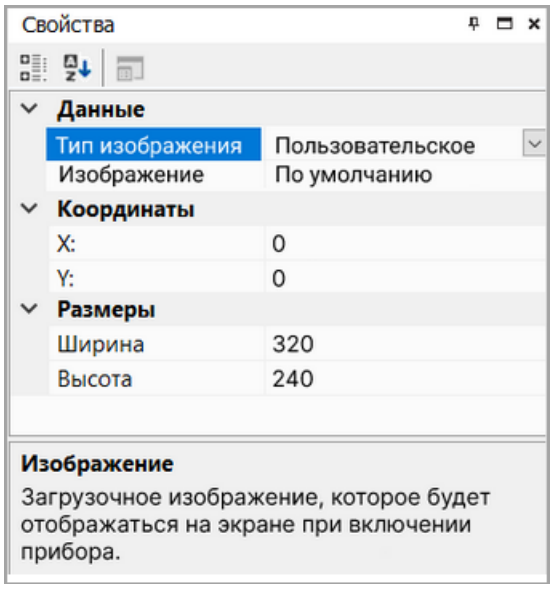


Рисунок 4.66 – Окно свойств загрузочного изображения

Группа	Параметр	Описание	Дополнительные возможности
Данные	**Тип изображения**	• при выборе значения "По умолчанию" дальнейшая настройка параметров не доступна - на загрузочном экране будет отображаться предустановленное изображение; • при выборе значения "Пользовательское" доступна настройка параметров изображения	
	Изображение	Выбор файла изображения **ПРЕДУПРЕЖДЕНИЕ** Для выбора доступны следующие типы файлов: *.jpg; *.jpeg; *.jpe, *.bmp. Максимальное разрешение изображения ограничено разрешением экрана устройства. Допускается уменьшение изображения в редакторе экрана Owen Logic	

Группа	Параметр	Описание	Дополнительные возможности
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Размеры	**Ширина**	Ширина элемента по оси X. Максимальное значение не более ширины загруженного изображения	Размеры элемента можно задать изменением границ объекта на поле экрана. Ширина и высота элемента изменяются пропорционально размерам загруженного изображения
	Высота	Высота элемента по оси Y. Максимальное значение не более высоты загруженного изображения	

После **загрузочного экрана** по умолчанию отображается первый экран из списка Группы экранов 1.

Параметры экрана

Параметры экрана отображаются в поле свойств **Менеджера экранов**.

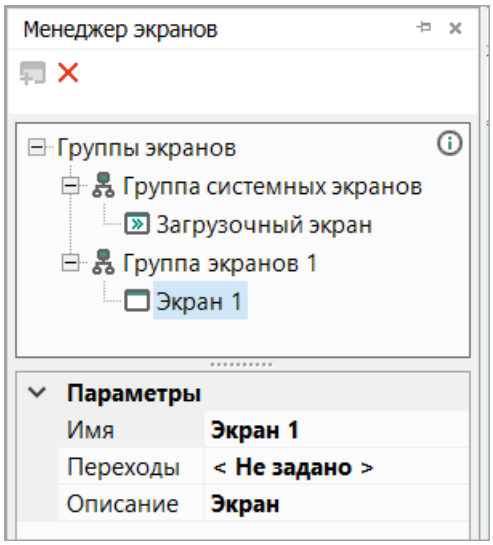


Рисунок 4.67

- **Имя** — для отображения в **Менеджере экранов** и в заголовке редактора экранов;
- **Переходы** — создание переходов между экранами;
- **Описание** — текстовое описание экрана;

Создание переходов между экранами

ПРИМЕЧАНИЕ
Настройка переходов к **Загрузочному экрану** недоступна.

Создание перехода экрана по изменению переменной

1. Нажмите ПКМ по элементу экран в дереве менеджера экранов и выберите **Настроить переходы** в контекстном меню или выделите ЛКМ свойство экрана Переходы на панели свойств экрана, и нажмите на кнопку **...**. Откроется окно редактора переходов. Выбранный экран будет отправной точкой для перехода.

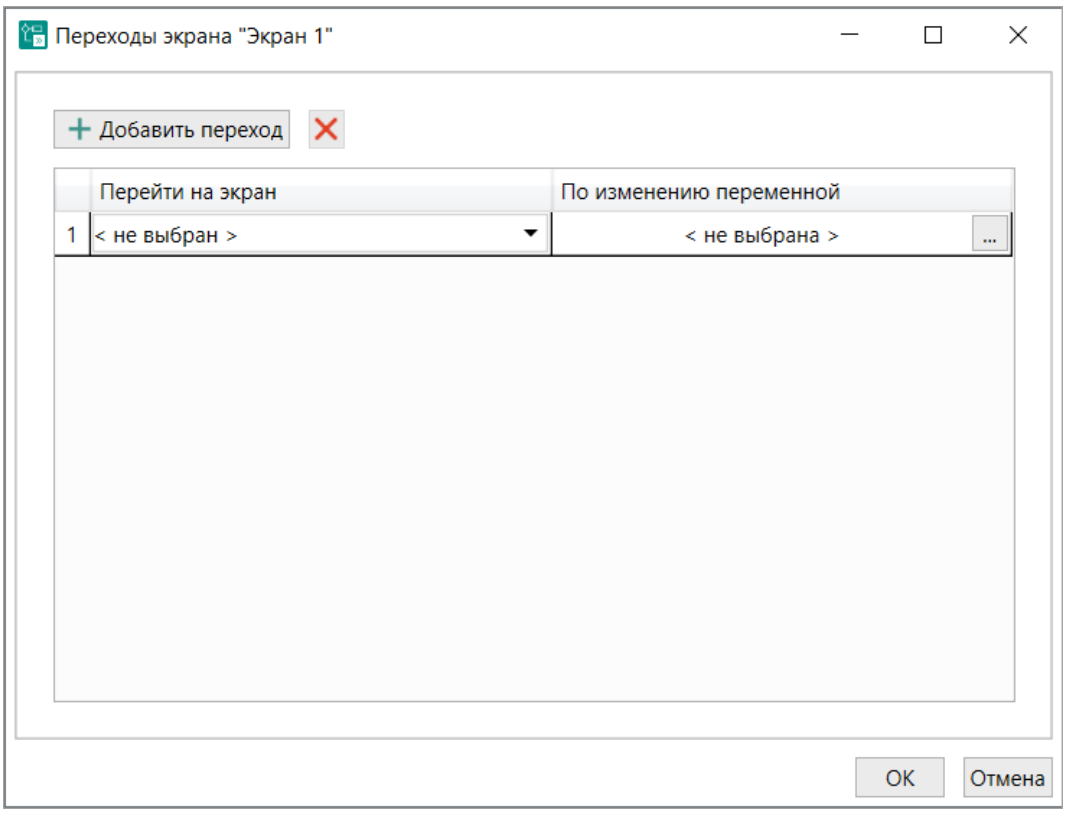


Рисунок 4.68

2. В выпадающем меню столбца **Перейти на экран** выберите другой экран для перехода к нему.
3. В столбце **По изменению переменной** нажмите на кнопку **...** и выберите из таблицы переменных переменную булевского типа, в случае изменения значения которой будет выполнен переход. После исполнения перехода значения привязанной переменной изменяется на «0».
4. Для добавления другого перехода нажмите **Добавить переход**, для удаления выделенного перехода нажмите ✕.
5. Нажмите **OK**. Созданные переходы будут отображаться на панели свойств экрана. Порядок выполнения переходов соответствует их порядку в свойствах экрана.

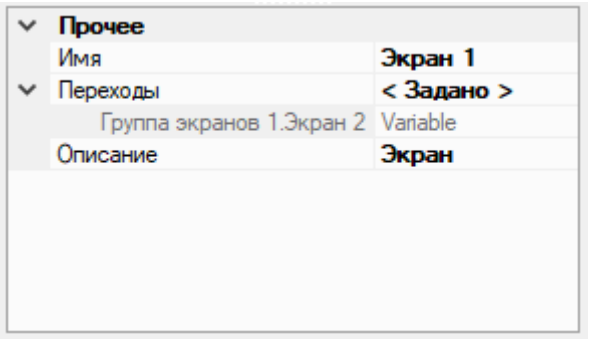


Рисунок 4.69

Создание переходов между экранами по кнопке

В нижней части редактора визуализации расположена панель "Кнопки", на которой отображаются все имеющиеся в приборе физические кнопки:

Рисунок 4.70

Двойное нажатие ЛКМ по кнопке открывает окно программирования событий для кнопки. Для каждого экрана кнопки программируются отдельно.

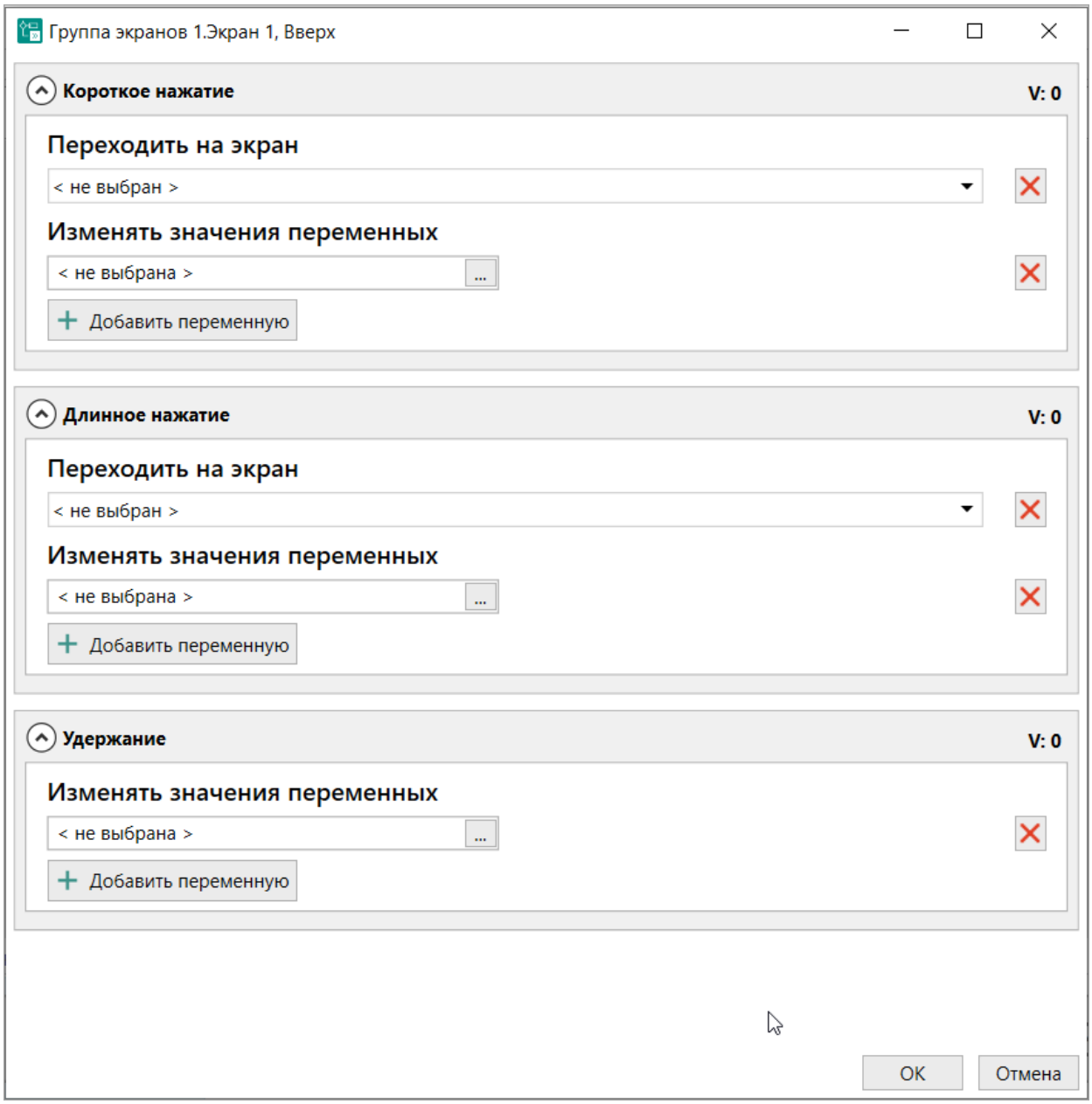


Рисунок 4.71

Для программирования кнопки доступны следующие события:

- короткое нажатие (менее 3 секунд);
- долгое нажатие (3 секунды и более);
- удержание (от 5 секунд).

Для короткого и долгого нажатия можно запрограммировать несколько действий (переходов на другие экраны и изменений значения булевских переменных). Для удержания можно запрограммировать только действие изменения значений булевских переменных. Действия выполняются в программе в порядке добавления.

Для добавления переменной нажмите + **Добавить переменную**. Для удаления запрограммированного действия нажмите ✕.

Настроенные события для кнопок являются приоритетными и переопределяют поведение кнопок по умолчанию. Для возврата к поведению кнопок по умолчанию удалите действия по событию и запишите программу в прибор.

Если для кнопок **Вверх** и **Вниз** события не настроены, то при нажатии на них прибор будет осуществлять переходы на другие экраны в соответствии с порядком в окне менеджера экранов.

Для каждой кнопки отображается количество настроенных событий:

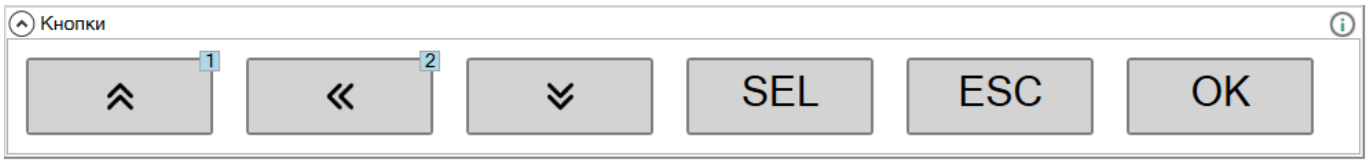


Рисунок 4.72

4.5 Режим симулятора

Корректность работы созданной программы проверяется с помощью режима симулятора, в котором моделируется изменение состояний выходов в зависимости от изменения состояний входов. Симуляция позволяет проанализировать состояние сигналов внутри программы.

Для перехода в режим симулятора нажмите кнопку на панели инструментов или выберите в главном меню **Сервис** → **Режим симулятора**. Откроется панель симуляции.

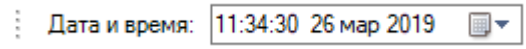
Панель симуляции

	Пуск	Запуск симуляции в реальном времени
	Шаг на один цикл	Пошаговая симуляция, выполняет 1 цикл программы
	Пауза	Пауза симуляции, повторное нажатие продолжает симуляцию
	Остановка симулятора	Остановка симуляции, переводит режим симуляции в первоначальное состояние
	Период, мс	Поле ввода для задания периода обновления информации на схеме в миллисекундах
	Время цикла	Поля ввода для задания времени цикла выполнения программы в режиме симулятора и выбора единиц измерения времени цикла
	Окно просмотра	Открыть/закрыть окно для просмотра значений переменных на каждом шаге выполнения программы
	Симуляция дисплея	Открыть окно симуляции дисплея

ВНИМАНИЕ
Время цикла в режиме симуляции не то же самое, что время цикла 8.2, рассчитываемое непосредственно в приборе. **Время цикла** может быть выбрано произвольно и требуется для симуляции работы временных ФБ – TON 7.2.2.2, TOF 7.2.2.3, BLINK 7.2.3.1. Все уставки временных ФБ будут отработаны в промежуточных шагах симулятора.

Симулирование работы блоков часов

Для приборов с функцией часов реального времени открывается дополнительная панель, с помощью которой можно симулировать поведение ФБ CLOCK 7.2.2.4 и CLOCK WEEK 7.2.2.5 во времени.



Последовательность и правила работы в режиме симуляции

Порядок симуляции программы:

1. Запуск симулятора в одном из режимов: реального времени ( ) или в пошаговом ( ).
2. Задание значений входных сигналов на блоках программы:

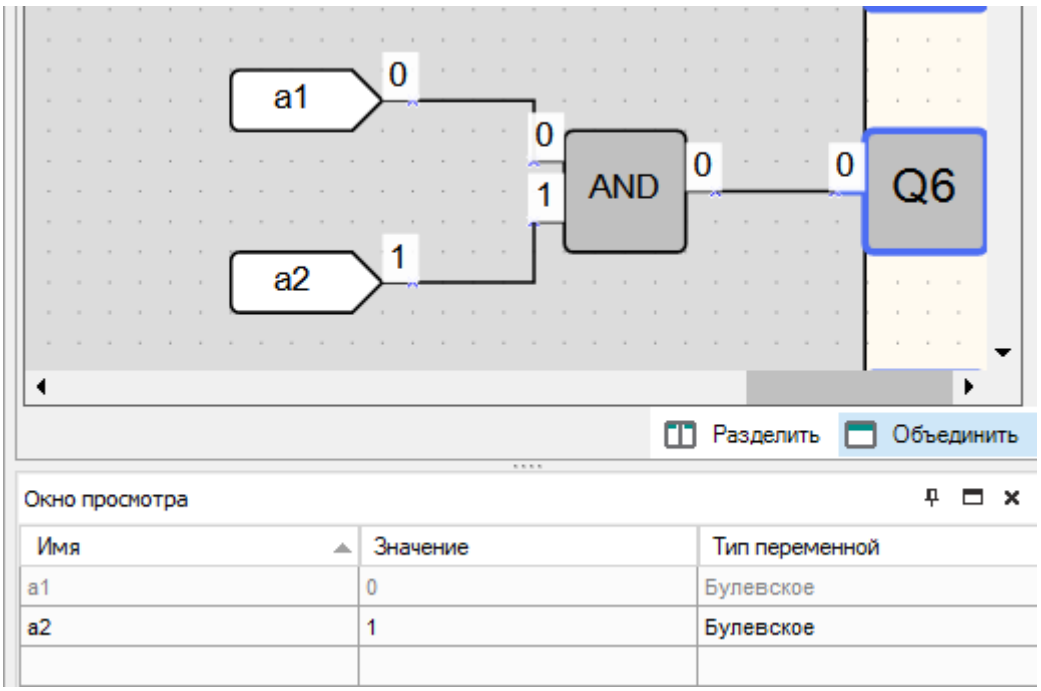


Рисунок 4.73

3. Подбор значений параметров **Период, мс** и **Время цикла** для удобства симуляции.
4. Выход из режима симуляции для корректировки программы.

Режим симуляции работает по следующим правилам:

- программа (вкладка **Схема**) и макросы моделируются по отдельности;
- работа блоков, не имеющих связи ни с одним из выходов прибора или выходным блоком сетевой переменной, не симулируется;
- симуляция не будет работать для энергозависимых и некорректно привязанных переменных.

Окно просмотра

Окно просмотра можно открыть или скрыть нажатием кнопки на панели симуляции. Окно предназначено для просмотра значений переменных или входов/выходов на каждом шаге выполнения программы.

До запуска симуляции окно пустое.

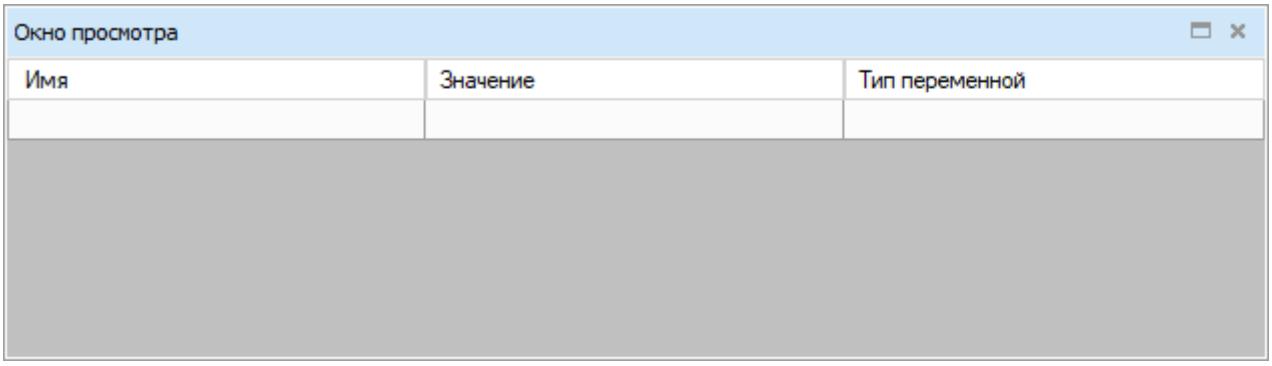


Рисунок 4.74

Для добавления переменной в **окно просмотра** нажмите ПКМ на переменную на схеме и выберите **Добавить в окно просмотра** в контекстном меню.

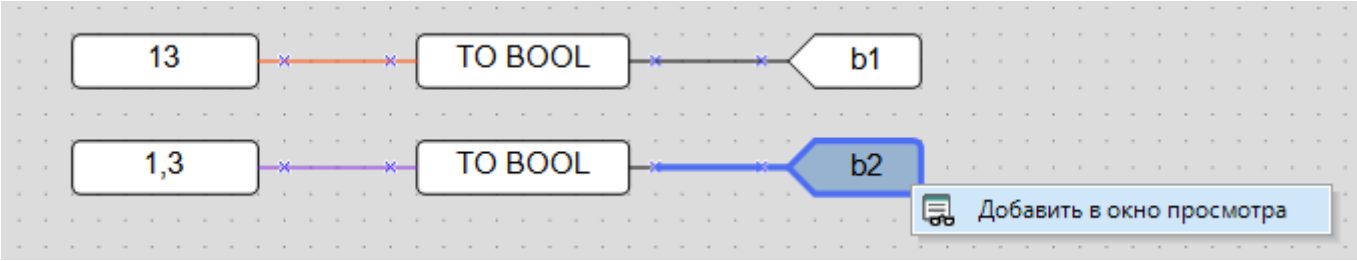


Рисунок 4.75

Также добавить переменную в **окно просмотра** можно нажав на пустое поле в колонке **Имя** в **Окне просмотра**, затем на кнопку «**...**».

Рисунок 4.76

Откроется таблица переменных. В ней могут быть выбраны переменные проекта, входы и выходы.

Ограничения при работе с таблицей переменных, открытой из окна просмотра переменных:

- создание, дублирование и удаление переменных недоступны;
- изменение имени, типа данных и параметров энергонезависимости невозможно;
- для стандартных переменных недоступна настройка параметров массива, а также функции импорта и экспорта переменных;
- для сетевых Slave переменных недоступна настройка параметров.

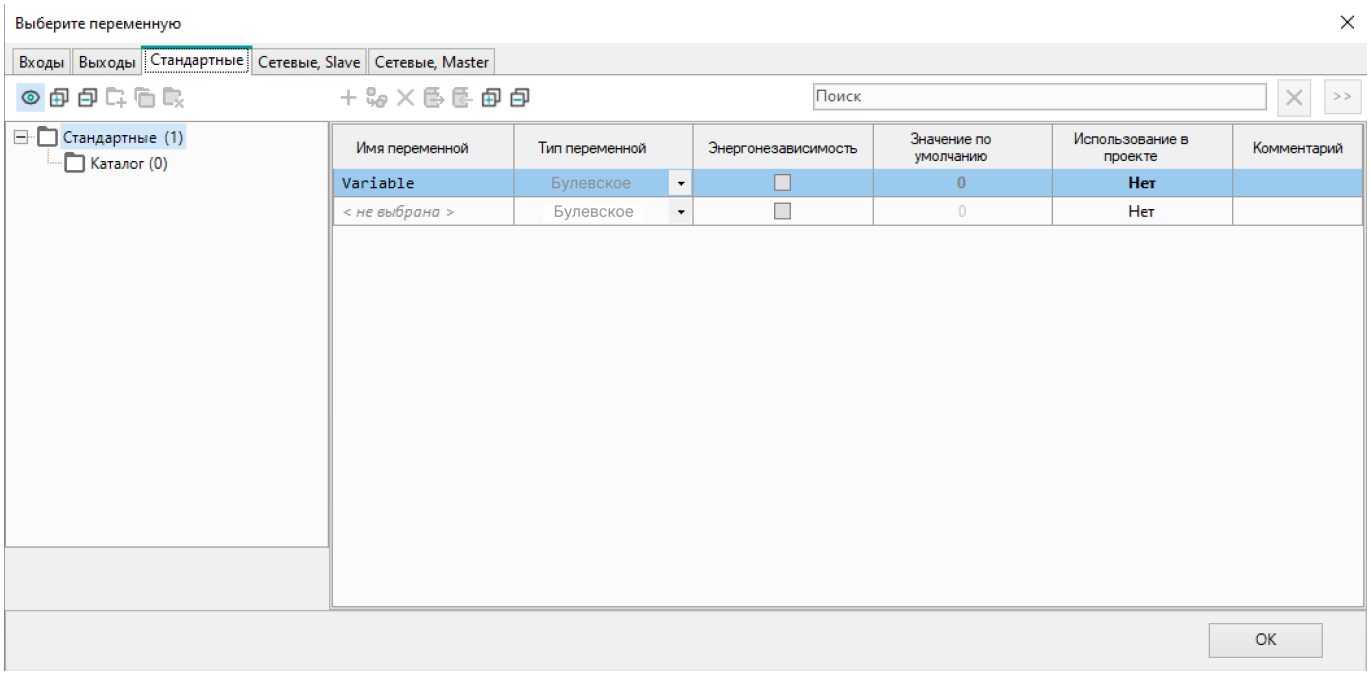


Рисунок 4.77

Выберите переменные и нажмите кнопку **OK**. Выбранные переменные добавятся в окно просмотра.

Переменная типа массив добавляется в таблицу окна просмотра переменных вместе с элементами массива. При этом допускается добавление отдельных элементов массива, даже если соответствующий массив уже присутствует в таблице окна просмотра целиком.

Окно просмотра

Имя	Значение	Тип переменной
Mass_1[3]	0	Целочисленное
Variable	0	Булевское
Variable-1	0	Булевское
Mass_1		[0..5] Целочисленное
Mass_1[0]	0	Целочисленное
Mass_1[1]	0	Целочисленное
Mass_1[2]	0	Целочисленное
Mass_1[3]	0	Целочисленное
Mass_1[4]	0	Целочисленное
Mass_1[5]	0	Целочисленное

Рисунок 4.78

Незадействованные переменные будут отображаться в окне просмотра серым цветом, значение всегда обозначено числом 0, ввод значения для таких переменных невозможен.

Для удаления переменной из окна просмотра выделите переменную и нажмите Delete, или воспользуйтесь контекстным меню переменной в **Окне просмотра**:

Окно просмотра

Имя	Значение	Тип переменной
Mass_1[3]	0	Целочисленное
Variable	0	Булевское
Variable-1		Булевское
Mass_1		[0..5] Целочисленное
Mass_1[0]		Целочисленное
Mass_1[1]	0	Целочисленное
Mass_1[2]	0	Целочисленное

Удалить переменную Delete
Показать ссылки

Рисунок 4.79

Удалить элемент массива, вложенный в массив, невозможно.

Ввод значений переменных

Во режиме симулятора значения переменных можно задавать в окне просмотра в колонке **Значения**.

Для переменных также можно указывать значения на схеме. Двойное нажатие на переменную вызывает окно с полем ввода нового значения.

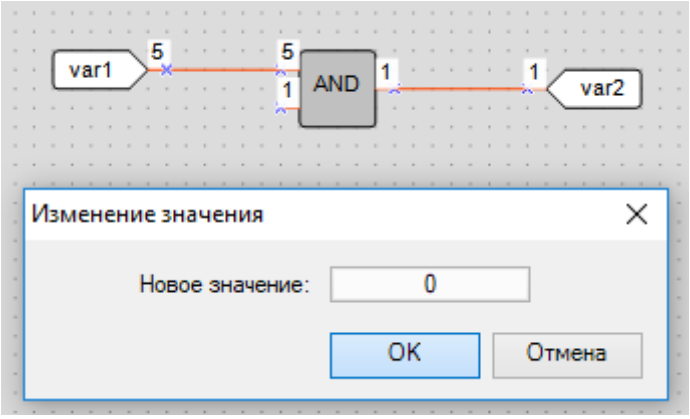


Рисунок 4.80

После ввода значения переменной, отличного от нуля, блок на схеме изменит цвет на красный, на один цикл выполнения программы. При остановленной симуляции блок сохранит красный цвет до запуска симуляции или выполнения команды **Шаг на один цикл:**

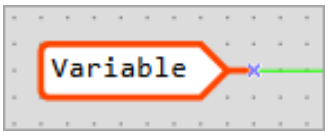


Рисунок 4.81

После ввода нулевого значения переменной (для переменной типа массив значение 0/False должно быть у всех элементов массива) блок на схеме изменит цвет на синий на один цикл выполнения программы. При

остановленной симуляции блок сохранит синий цвет до запуска симуляции или выполнения команды **Шаг на один цикл:**

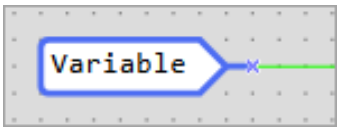


Рисунок 4.82

В режиме симулятора можно изменять значения входов прибора, нажимая на них. По нажатию дискретные входы меняют цвет и булевское значение. Для аналоговых входов значение задается аналогично переменным в окне с полем ввода.

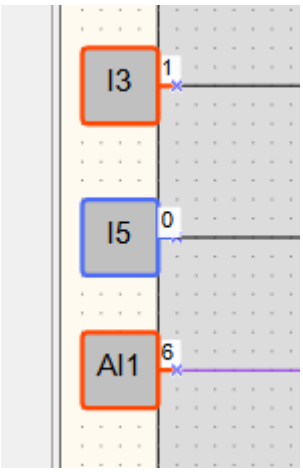


Рисунок 4.83

4.5.1 Симуляция визуализации

В режиме симулятора для приборов с графическим цветным ЖКИ доступна симуляция дисплея и кнопок прибора.

ПРИМЕЧАНИЕ
Работа элемента визуализации **График** в режиме симуляции не поддерживается.

После входа в режим откроется окно, отображающее стартовый экран прибора.

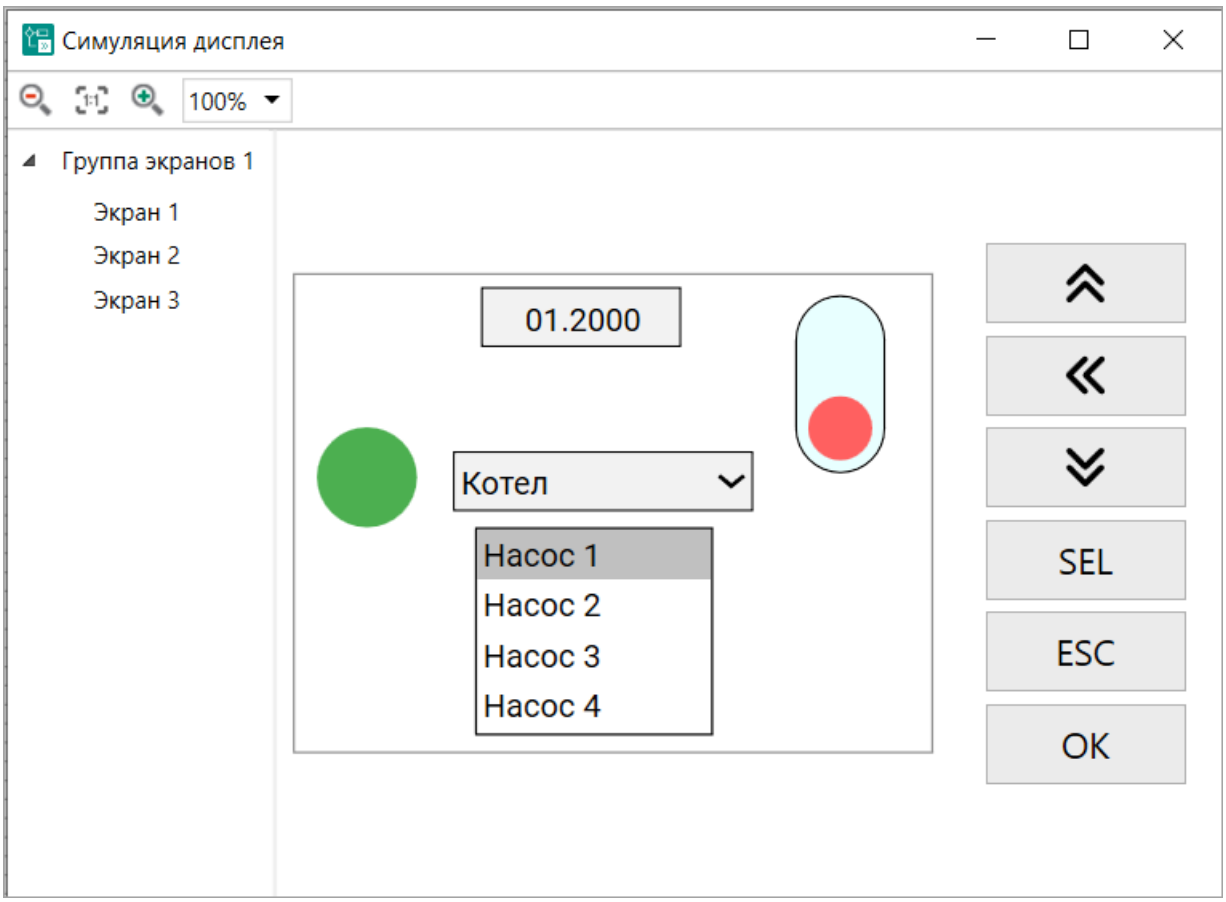


Рисунок 4.84

В левом верхнем углу окна расположены кнопки масштабирования.

В левой части окна расположен список экранов, доступных в проекте.

В центральной части окна отображается симуляция экрана приора.

В правой части окна находятся кнопки для симуляции действий с реальными кнопками прибора.

Работа с экраном

В режиме симуляции доступно редактирование значения переменных следующих элементов визуализации:

- Динамический текст;
- Формы ввода/вывода int/float;
- Формы ввода/вывода времени;
- Формы ввода/вывода IP;
- Переключатель;
- Группа переключателей;
- Круг, многоугольник, изображение, кнопка если у элемента настроено действие;
- Меню — возможно переключение экранов. Редактирование значений переменных не предусмотрено.

Работа с кнопками

Кнопки в правой части окна предназначены для симуляции действий с реальными кнопками прибора.

Действия кнопок задаются для каждого экрана. Действия кнопок обрабатываются для экрана, который в текущей момент отображается на основной области окна.

Управление кнопками симуляции может осуществляться клавишами ***Q***, ***W***, ***E***, ***A***, ***S***, ***D***, ***Z***, ***X***, ***C***. Нажатие кнопок на клавиатуре активно только если окно симуляции визуализации находится в фокусе.

При пошаговой симуляции удержание кнопок не работает.

4.5.2 Симуляция ST кода

В Owen Logic доступна симуляция кода функции или функционального блока, написанного на языке ST. Для этого перейдите в окно **Редактор функции** (или **Редактор функционального блока**) и нажмите кнопку ▶ на

панели инструментов, или в главном меню **Сервис** → **Режим симулятора**. Откроется панель симуляции. В окне **Редактора** значения переменных будут отображены в окнах просмотра, расположенных рядом с переменной. Если в программе объявлены массивы, то в окне просмотра отображается слово **Изменить**.

Схема | Редактор функционального блока "functionblock1"

```
1  function_block functionblock1 //имя функционального блока.
2
3      var_input //объявление входных переменных
4         U [FALSE] : bool; //входная переменная с типом данных bool
5         Res [FALSE] : bool; //входная переменная с типом данных bool
6         N [0] : udint; //входная переменная для значения счетчика после сброса
7      end_var
8
9      var_output //объявление выходных переменных
10        Q [0] : udint; //выходная переменная с типом данных udint
11     end_var
12
13     var //объявление локальных переменных
14        CounterValue [0] : udint; //переменная для текущего значения счетчика
15        RTrig [FALSE] : bool; //переменная для определения переднего фронта на входе «U»
16        fb2 : functionblock2;
17        arr1 [Изменить] : array [0..4] of udint;
18     end_var
19
20     CounterValue [0] := arr1[0] [0];
21     CounterValue [0] := arr1[N [0]] [0];
22     CounterValue [0] := arr1[N [0] + 1] [Изменить];
23     CounterValue [0] := arr1[fb2.Q [0]] [0];
24     CounterValue [0] := arr1[bool_to_udint(U [FALSE])] [Изменить];
25
26 end_function_block
```

Рисунок 4.85 – Окно Редактор функционального блока в режиме симулятора

Цветовые обозначения окон просмотра:

[13.3] — текущее значение переменной;

[12] — значение переменной изменено пользователем;

[13.3] — значение переменной зафиксировано пользователем;

[Изменить] — значение одного или нескольких элементов массива (не всех) зафиксировано пользователем;

[???] — ошибка: неверное значение переменной (например, когда с помощью переменной задается несуществующий элемент массива).

При нажатии ЛКМ на окно просмотра для ввода значения целочисленной переменной (udint) или переменной с плавающей запятой (real) в режиме симуляции появится окно, в котором можно ввести новое значение переменной и, если необходимо, зафиксировать значение.

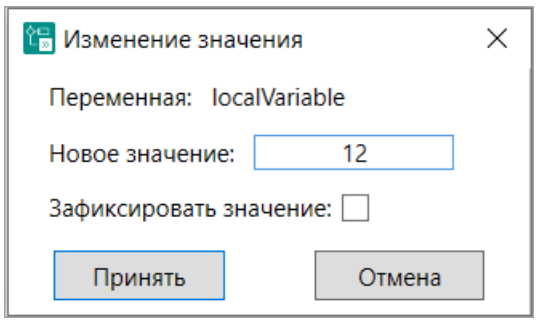


Рисунок 4.86 – Изменение значения переменной udint

ПРИМЕЧАНИЕ
Если значение переменной типа real превышает 8 знаков, то значение округляется.
Если значение переменной типа real превышает 9 знаков, то значение переводится в экспоненциальную форму. Например, значение 1234567891 будет отображаться, как 1.2345678E+09.

При нажатии ЛКМ на окно просмотра для ввода значения булевской переменной (bool) в режиме симуляции появится окно, в котором можно выбрать значение true или false и, если необходимо, зафиксировать значение.

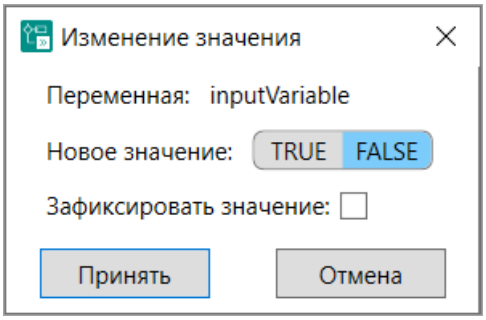


Рисунок 4.87 – Изменение значения переменной bool

При нажатии ЛКМ на окно просмотра для ввода значения временной переменной (time или DT) в режиме симуляции появится окно, в котором можно ввести значение в формате T#0d0h0m0s0ms (для time) или DT#2000-01-01-0:00:00 (для dt), и, если необходимо, зафиксировать значение.

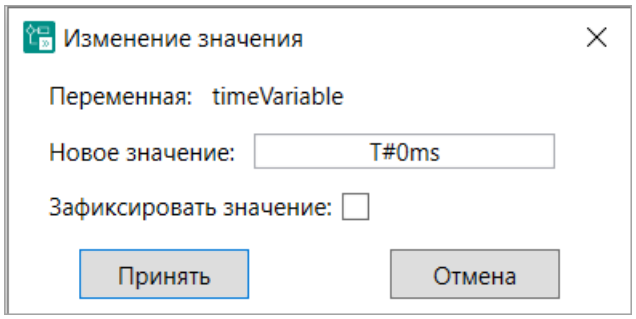


Рисунок 4.88 – Изменение значения переменной time

ПРИМЕЧАНИЕ
Время пересчитывается автоматически после нажатия кнопки **Принять**. Например, введенное значение переменной T#80m при повторном открытии окна **Изменение значения** примет вид T#1h20m.

В случае ввода некорректного значения или отсутствия значения появится сообщение об ошибке. Кнопка **Принять** будет неактивна.

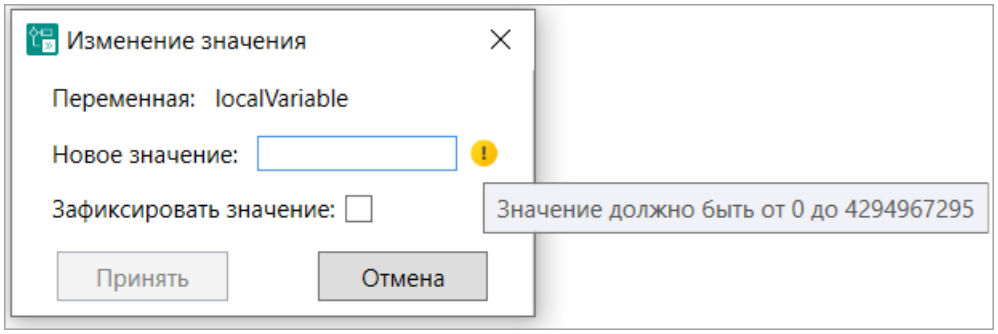


Рисунок 4.89 – Ошибка ввода значения

Для ввода значений в элементах массива нажмите ЛКМ на окно просмотра с надписью **Изменить**:

- если индекс элемента задается константой или с помощью переменной, то появится окно Изменение значения для конкретного элемента массива;
- если индекс элемента массива задается как результат математического выражения или как результат функции/функционального блока, появится окно со всеми элементами данного массива и значениями переменных, доступными для ввода:

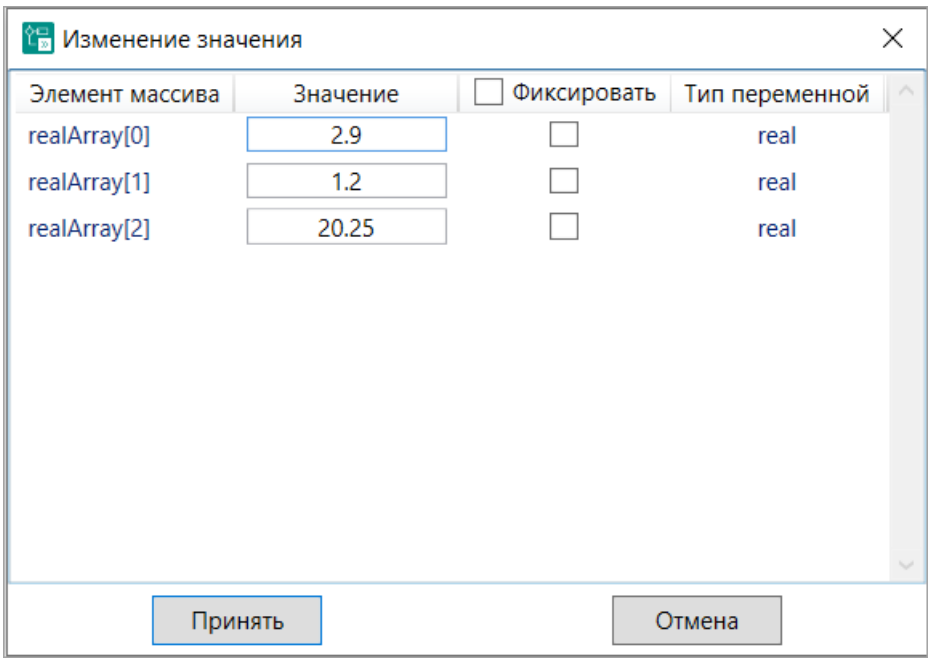


Рисунок 4.90 – Изменение значений элементов массива с переменными real

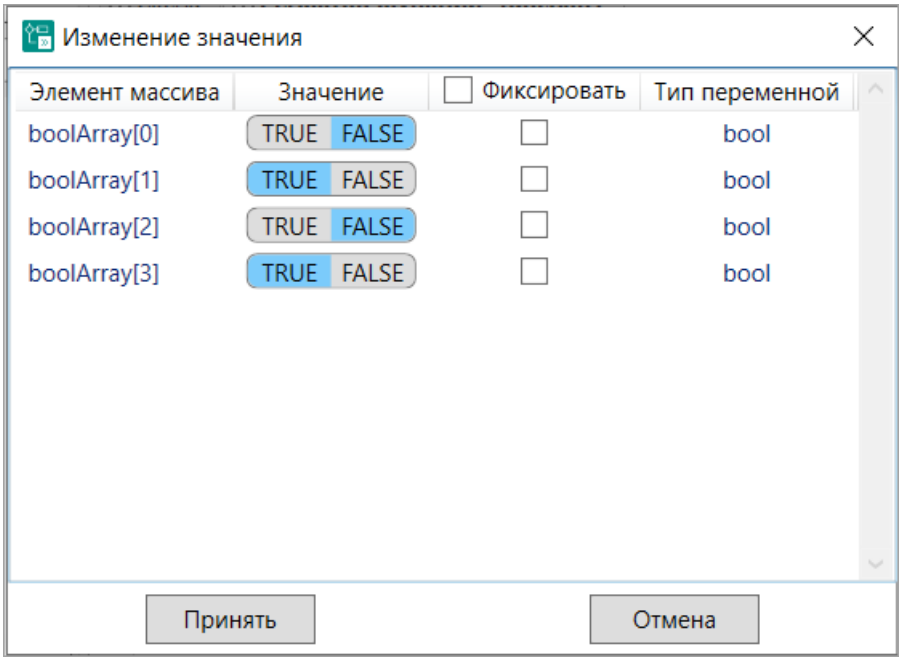


Рисунок 4.91 – Изменение значений элементов массива с переменными bool

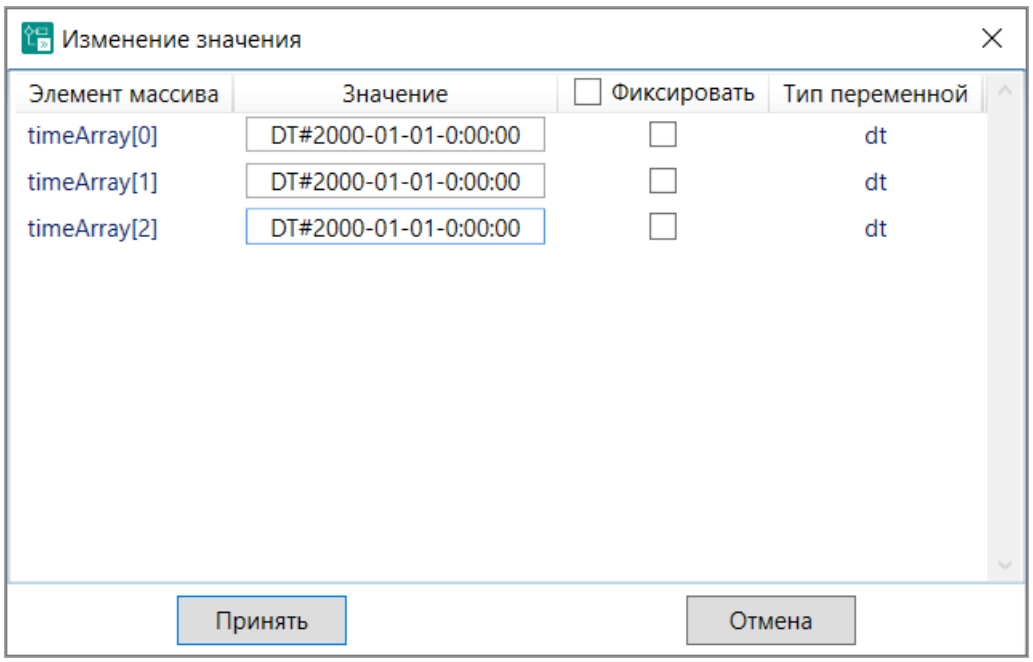


Рисунок 4.92 – Изменение значений элементов массива с переменными dt

В случае ввода некорректного значения или отсутствия значения появится сообщение об ошибке. Кнопка **Принять** будет неактивна.

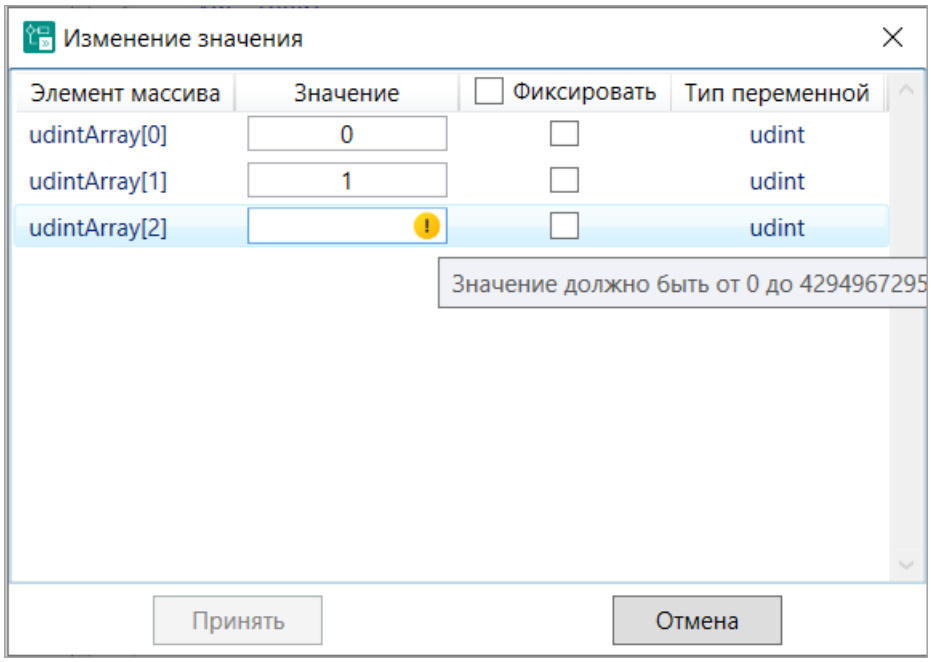


Рисунок 4.93 – Ошибка ввода значения

4.6 Настройка порта и подключение прибора

Чтобы открыть окно **Настройки порта**, следует выбрать в главном меню **Прибор** → **Настройки порта** или нажать на индикатор **COM** в строке состояния 3.7.

В окне **Настройка порта** выбирается номер используемого COM-порта и адрес подключаемого прибора, остальные настройки фиксированные и выводятся для справки.

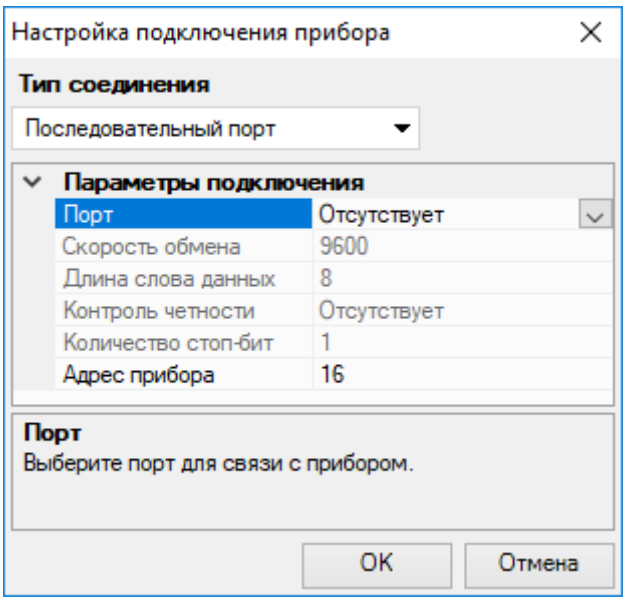


Рисунок 4.94

Подключение прибора по USB

Для подключения прибора к ПК через порт USB следует:

1. Установить драйвер USB, который можно скачать на сайте www.owen.ru.
2. Подключить прибор к ПК и включить питание (если требуется). В зависимости от модели прибора для соединения используется комплект для программирования ПР-КП20 или кабель для программирования из комплекта поставки прибора. Номер эмулируемого COM-порта можно узнать в Диспетчере устройств ОС Windows.

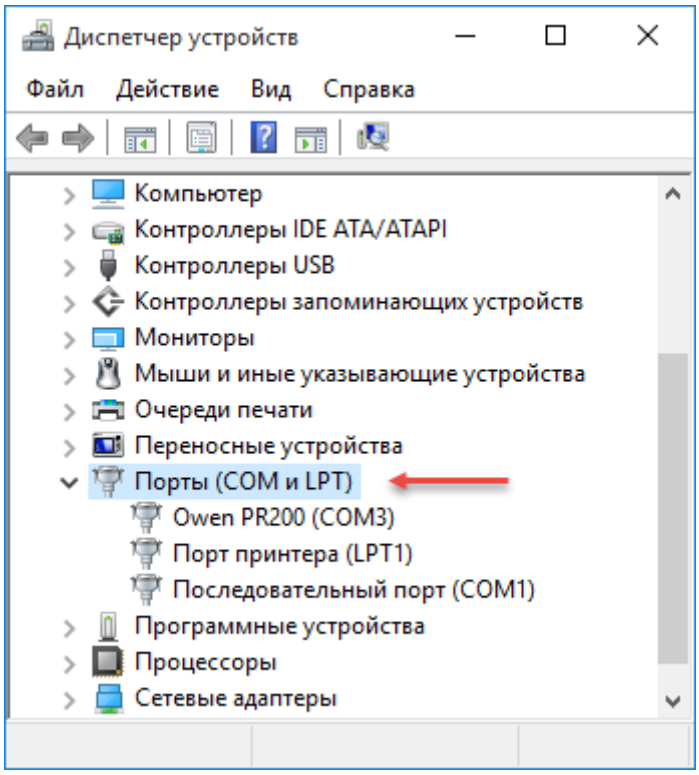


Рисунок 4.95

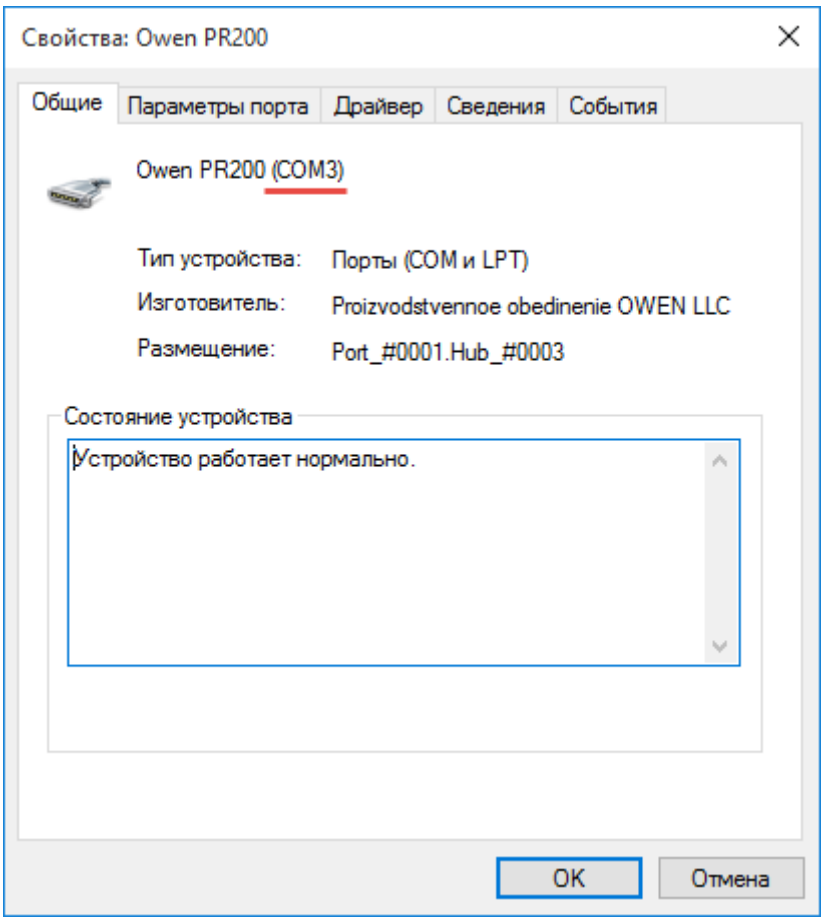


Рисунок 4.96

3. В окне **Настройка порта** в выпадающем списке **Тип подключения** выбрать **Последовательный порт**.
4. Ввести номер COM-порта и адрес прибора (по умолчанию – **16**) в окне **Настройка порта** и нажать **OK**. Если подключение корректно, то в строке состояния отобразится наименование подключенного прибора.

Подключение прибора по Ethernet/Wi-Fi

Для подключения прибора к ПК по интерфейсу Ethernet или с помощью Wi-Fi следует:

1. Подключить прибор к той же локальной сети, в которой находится ПК.
2. Узнать IP-адрес подключенного прибора. IP-адрес по умолчанию указан в *Руководстве по эксплуатации* прибора. Текущий IP-адрес прибора можно посмотреть с помощью Owen Configurator.
3. В окне **Настройка порта** в выпадающем списке **Тип подключения** выбрать **Ethernet/Wi-Fi**.

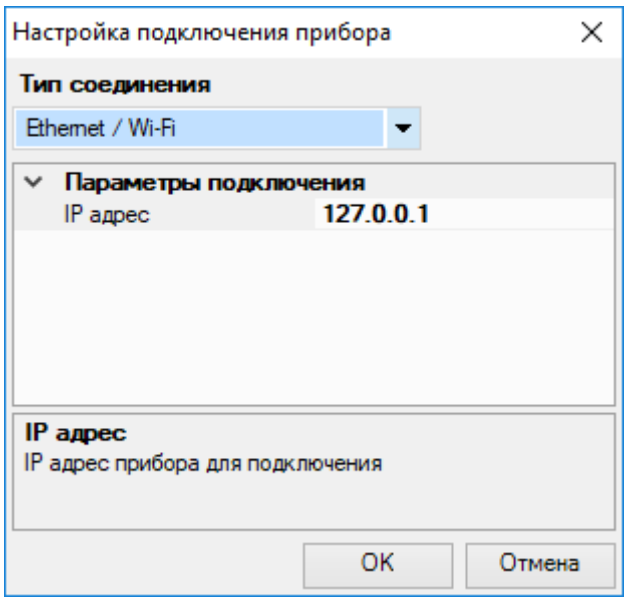


Рисунок 4.97

4. Ввести IP-адрес подключенного прибора в графе **IP адрес** и нажать **OK**. Если подключение корректно, то в строке состояния отобразится наименование подключенного прибора.

4.7 Запись программы в прибор

Запись программы в прибор

ВНИМАНИЕ
Если в подключенном приборе уже есть ранее записанная программа, то она заменяется новой.

Созданная программа записывается в ПЗУ прибора.

Для записи программы в прибор следует:

1. Подключить прибор к ПК.
2. Подать питание на прибор, если требуется.
3. Настроить параметры соединения, если в этом есть необходимость.
4. Записать программу в прибор, нажав кнопку на панели инструментов или выбрав в главном меню **Прибор** → **Записать программу в прибор**.

ПРИМЕЧАНИЕ
Перед записью программы Owen Logic проверяет совместимость с версией встроенного ПО прибора. Если они не совпадают, то запустится обновление встроенного ПО 8.3 прибора, затем программа запишется в прибор.

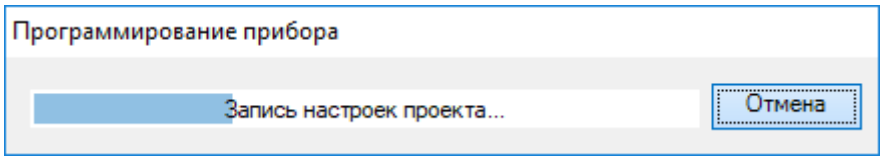


Рисунок 4.98

Если записывается программа с настройками для другой модели прибора, то появится следующее сообщение:

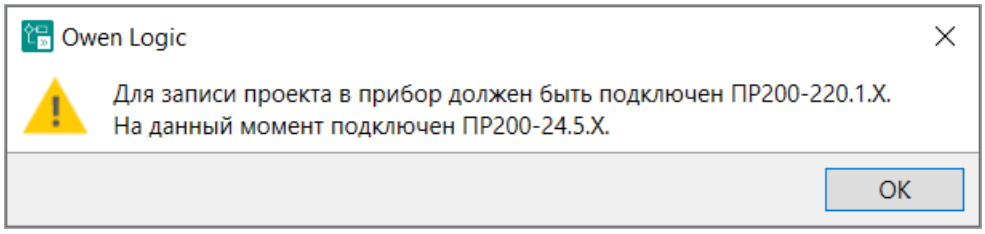


Рисунок 4.99

Если прибор защищен паролем (см. раздел Пароль 5.6), то для записи программы в прибор требуется ввести пароль.

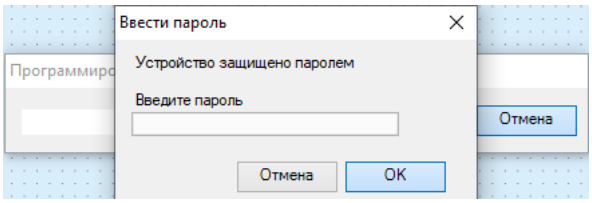


Рисунок 4.100

Соответствующее уведомление появится во время подключения прибора к ПК. Введение пароля требуется только один раз на все время сессии работы с Owen Logic. Для повторной записи программы (даже после подсоединения заново прибора к ПК) введение пароля не потребуется.

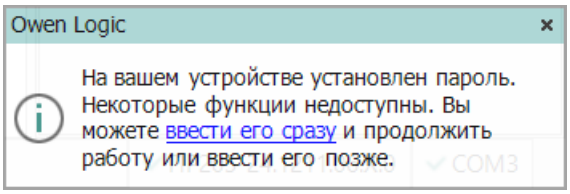


Рисунок 4.101

ПРИМЕЧАНИЕ
Сразу после записи прибор переходит в рабочий режим и программа запускается автоматически.

Переключение Owen Logic в режим OFFLINE

Режим OFFLINE следует использовать, если в двух или более приложениях Owen Logic на одном ПК настроена передача данных в один прибор. Приложения будут поочередно занимать порт, загрузка в прибор в таком случае прервется.

Приложения, которые не должны осуществлять загрузку, следует перевести в режим OFFLINE.

Режим OFFLINE включается/выключается кликом по имени прибора в строке состояния 3.7 или в главном меню **Сервис** → **Режим OFFLINE**.

4.8 Онлайн отладка

Режим онлайн отладки предназначен для тестирования программы в режиме реального времени с фактическими значениями со входов прибора. Онлайн отладка проводится только с подключенным к ПК прибором. В режиме онлайн отладки внесение изменений в проект невозможно.

Последовательность и правила работы в режиме онлайн отладки

Онлайн отладку можно проводить только после загрузки программы в прибор. Программа в приборе должна соответствовать программе на схеме проекта. В приборе должна быть соответствующая версия встроенного ПО.

Для проведения онлайн отладки следует:

1. Запустить онлайн отладку нажатием на кнопку на панели инструментов — откроется панель онлайн отладки.
2. Проверить логику работы программы с реальными значениями на входах прибора, которые отображаются на схеме.

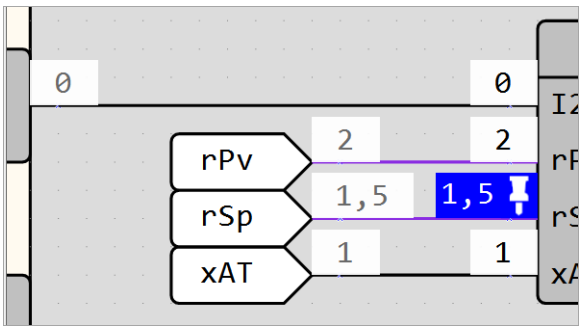


Рисунок 4.102

3. Для внесения корректировок выйти из режима онлайн отладки нажатием на кнопку .

Онлайн отладка доступна только для схемы проекта (для проверки работы макросов, функций и функциональных блоков на языке ST следует использовать режим симуляции).

Ручное задание значений

В режиме онлайн-отладки можно задавать необходимые значения для блоков выходной переменной, а также для входов функций, функциональных блоков и макросов. Чтобы задать значение в режиме онлайн отладки нажмите на плашку со значением переменной и, в открывшемся окне, введите новое значение.

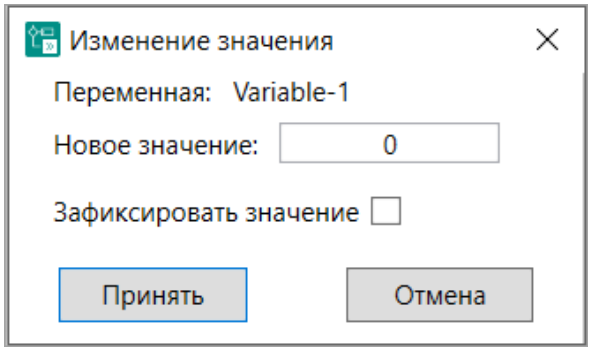


Рисунок 4.103

Для переменной типа массив возможен ввод значения всех элементов массива:

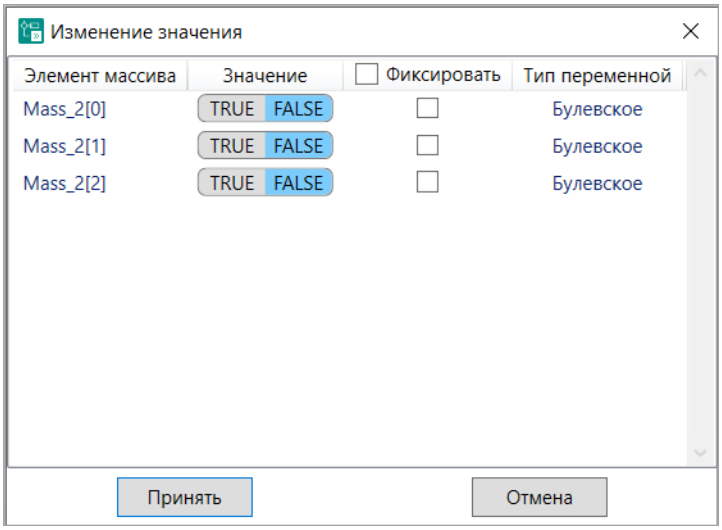


Рисунок 4.104

Введенные значения элемента переменной типа массив в окне "Изменение значения" выделяются оранжевым цветом:

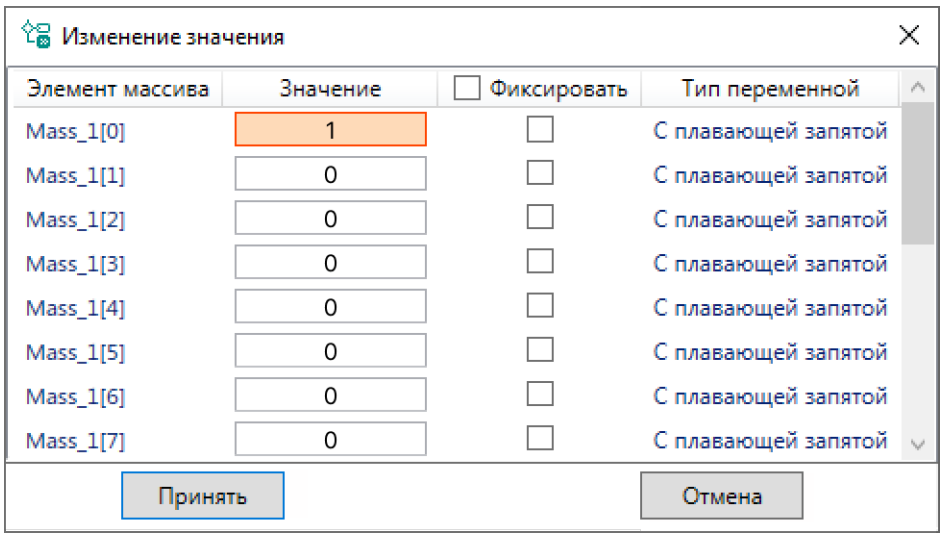


Рисунок 4.105

Отображение значений на схеме

Если не установить галочку **Зафиксировать значение**, то новое значение будет действовать в течение одного рабочего цикла. В конце текущего цикла или в начале следующего заданное значение может быть перезаписано. В течение действия заданное значение выделяется желтым цветом.

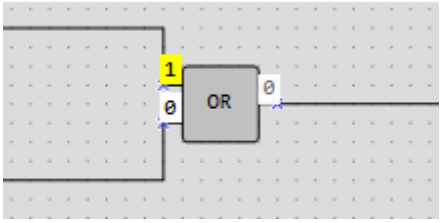


Рисунок 4.106

Если зафиксировать заданное значение, то оно будет действовать каждый рабочий цикл до изменения вручную на другое, или до отключения фиксации. Фиксация значений отключается в случае выхода из режима онлайн отладки. Для переменных типа массив доступна фиксация одного, нескольких или всех значений элементов массива. Зафиксированное значение (для массивов - фиксация значений всех элементов) отображается на схеме плашкой синего цвета:

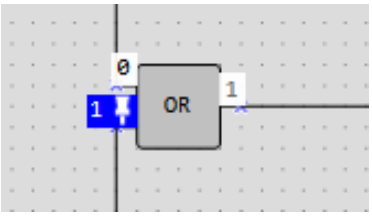


Рисунок 4.107

В случае, если зафиксировать значение одного или нескольких элементов массива (не всех), то плашка отображается белого цвета с синей рамкой:

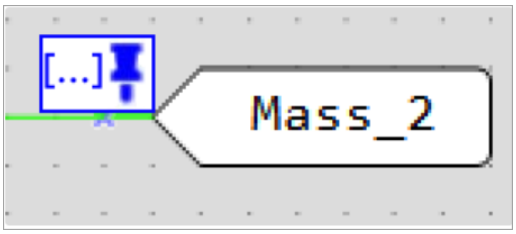


Рисунок 4.108

Значения блоков входной переменной задать невозможно, поэтому значение этих входных переменных отображаются на схеме серым цветом. Для входных блоков переменных типа массив значение не отображается.

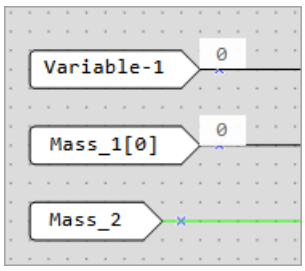


Рисунок 4.109

Устранение неполадок

В случае обрыва связи с прибором режим онлайн отладки отключится через 10 секунд, и прибор перейдет в рабочий режим. Если успеть восстановить связь, то онлайн отладка продолжится, но ранее зафиксированные значения будут отключены.

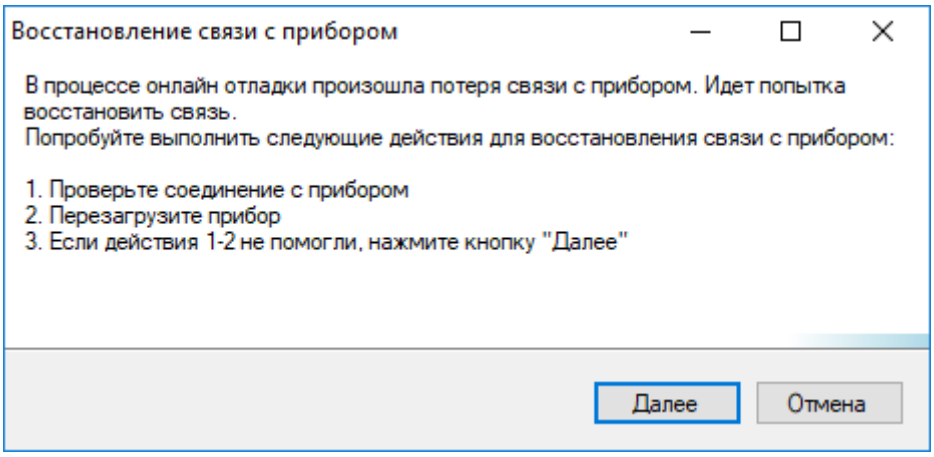


Рисунок 4.110

Предел передаваемых значений

Для каждой модификации прибора есть свой предел буфера памяти, используемого для передачи данных в режиме онлайн отладки. Если на схеме отображаются пустые плашки, значит срабатывает ограничение, и следует увеличить масштаб схемы, чтобы уменьшить количество отображаемых значений в окне схемы. Зафиксированные значения остаются зафиксированными, даже если не отображаются в окне схемы, и продолжают занимать буфер памяти. Если буфер памяти прибора будет полностью заполнен, то при нажатии на пустую плашку окно "Изменение значения" не откроется. Вместо него появится информационное сообщение о превышении лимита отображаемых значений:

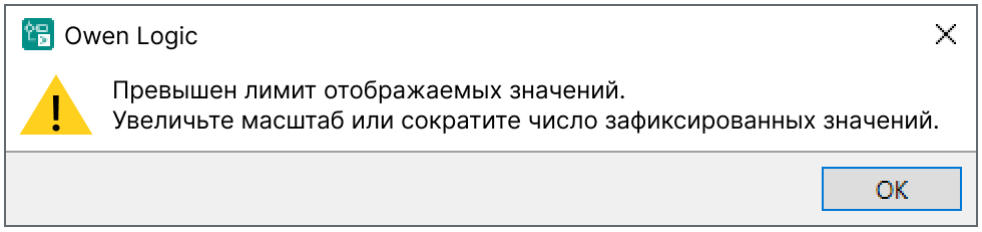


Рисунок 4.111

Если в окне Изменение значения присутствует хотя бы один зафиксированный элемент массива, а значения остальных элементов отображаются, но недоступны для ввода, то значит буфер памяти прибора переполнен. Значение зафиксированного элемента можно изменить или снять фиксацию.

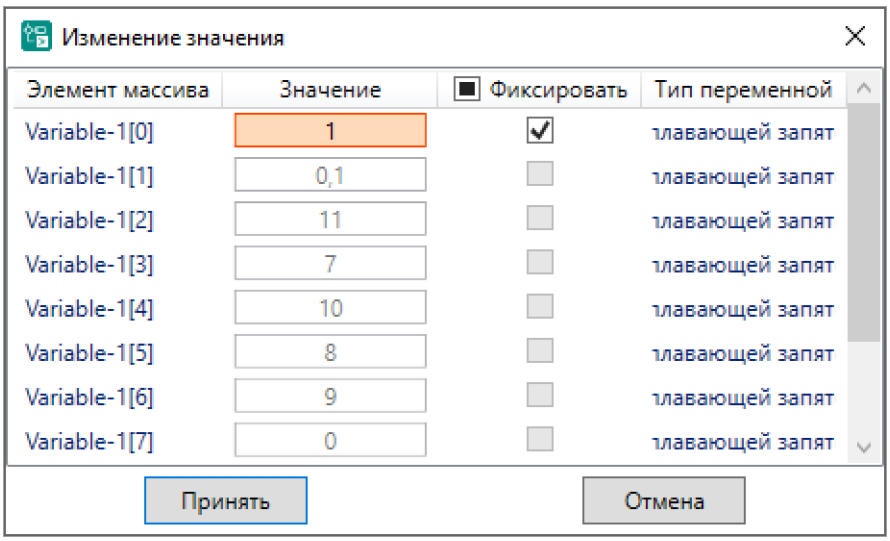


Рисунок 4.112

ПРИМЕЧАНИЕ
Отключение фиксации элементов массива не всегда уменьшает объем занимаемой памяти. В некоторых случаях (из-за особенностей внутреннего выделения буфера) снятие фиксации может привести к увеличению занимаемой памяти.

Если при попытке отключить фиксацию возникает риск переполнения буфера, кнопка **Принять** становится недоступной. В окне **Изменение значения** появляется предупреждающий маркер. При наведении на него курсора отображается сообщение с рекомендацией изменить количество зафиксированных элементов.

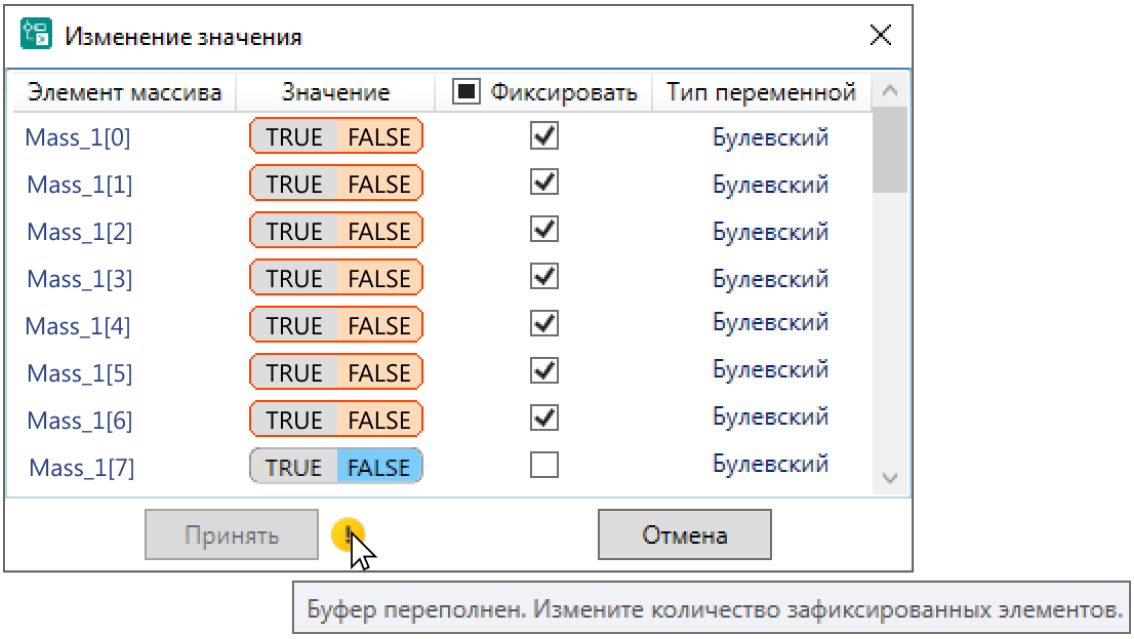


Рисунок 4.113

4.9 Сведения о проекте

Для запуска окна **Сведения о проекте** следует выбрать в главном меню **Файл** → **Сведения о проекте…**.

Вкладка «Общие»

На вкладке **Общие** содержится информация о версиях Owen Logic, в которых создавался и изменялся проект.

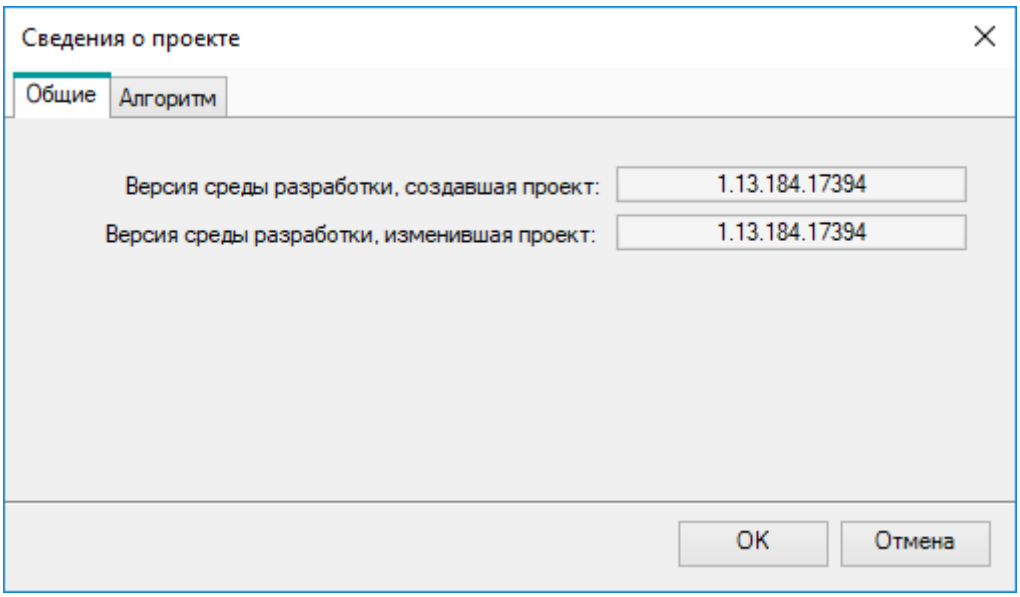


Рисунок 4.114

Вкладка «Алгоритм»

На вкладке **Алгоритм** можно указать и посмотреть информацию о группе, номере и версии программы для отображения в окне информации 8.1 о подключенном приборе после записи.

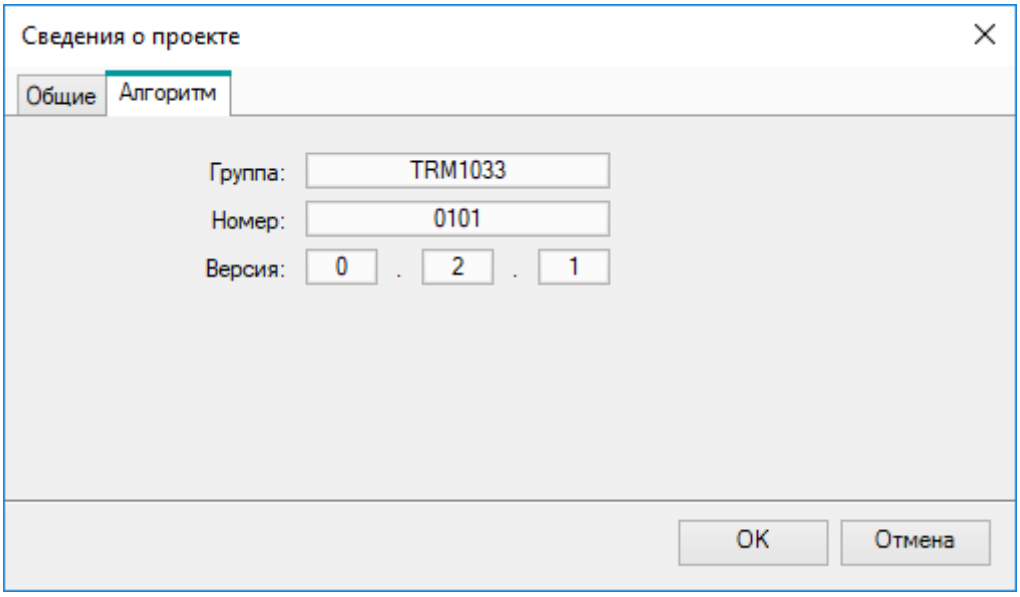


Рисунок 4.115

- **Группа** – определяет группу (семейство) алгоритмов;
- **Номер** – определяет номер и позволяет отличать схожие алгоритмы в рамках группы;
- **Версия** – определяет версию алгоритма. Рекомендуется для задания версии использовать формат Semantic Versioning.

Для приборов второго поколения доступен параметр **Имя**.

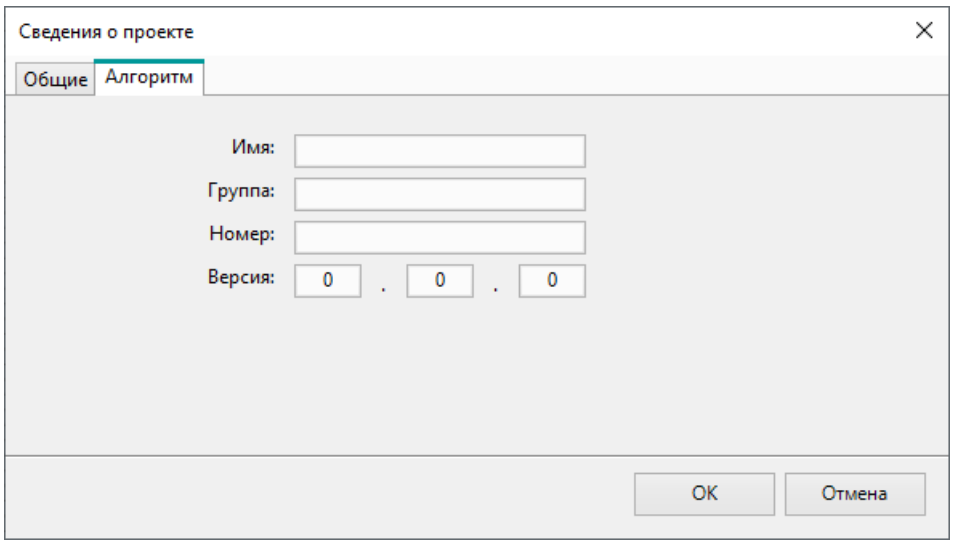


Рисунок 4.116

4.10 Менеджер компонентов

Менеджер компонентов служит для загрузки в проект компонентов из онлайн-базы компании ОВЕН (требует подключение к Интернету) и управления компонентами, которые хранятся на ПК.

Для запуска менеджера компонентов следует выбрать в главном меню **Файл** → **Менеджер компонентов**.

Вкладка «Онлайн-база»

На вкладке **Онлайн-база** отображается список компонентов из онлайн-базы компании ОВЕН.

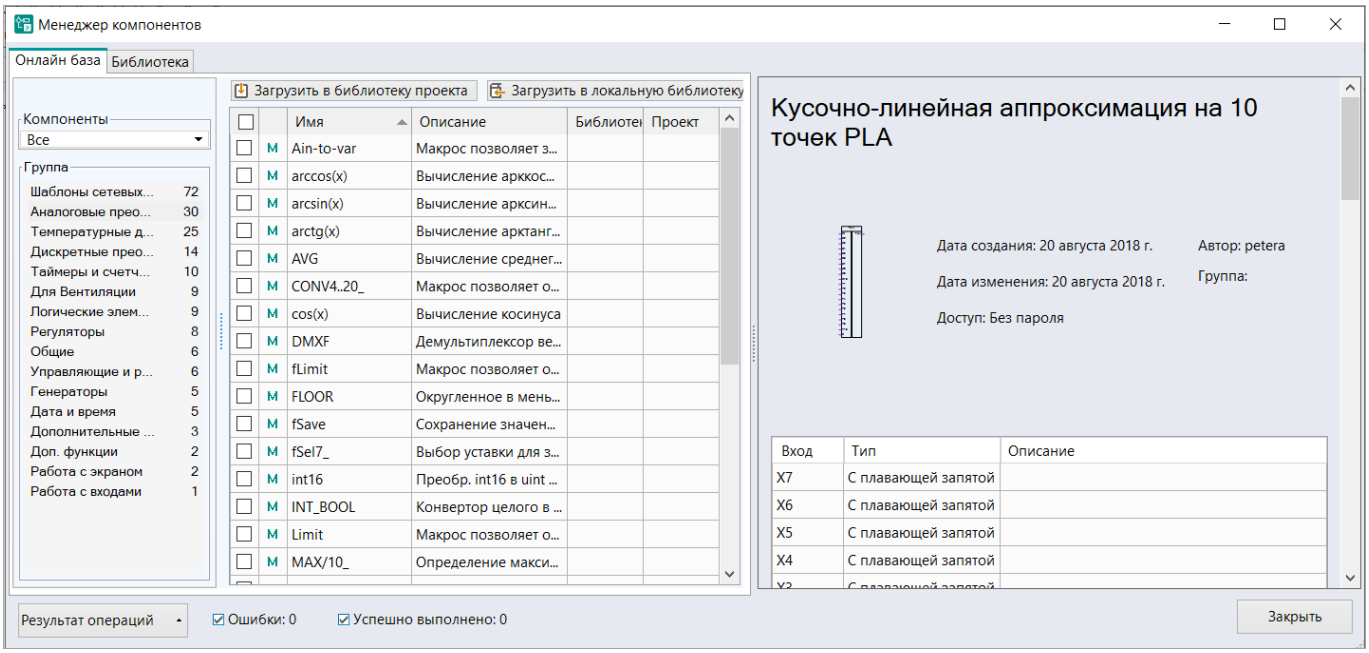


Рисунок 4.117

Выпадающее меню **Компоненты** позволяет отфильтровать список по типу:

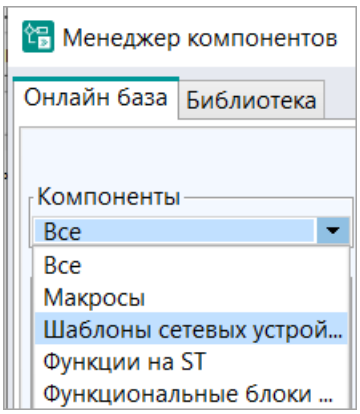


Рисунок 4.118

Макросы дополнительно разделены на категории в зависимости от назначения:

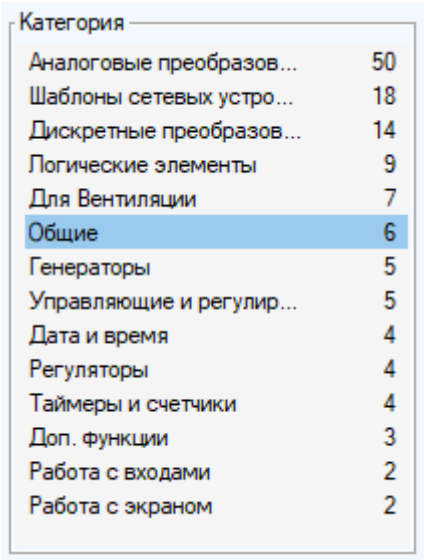


Рисунок 4.119

Полное описание компонента можно прочитать, выделив его в списке. Описание компонента отображается в правой части окна. Для отображения полной документации к компоненту требуется Acrobat Reader.

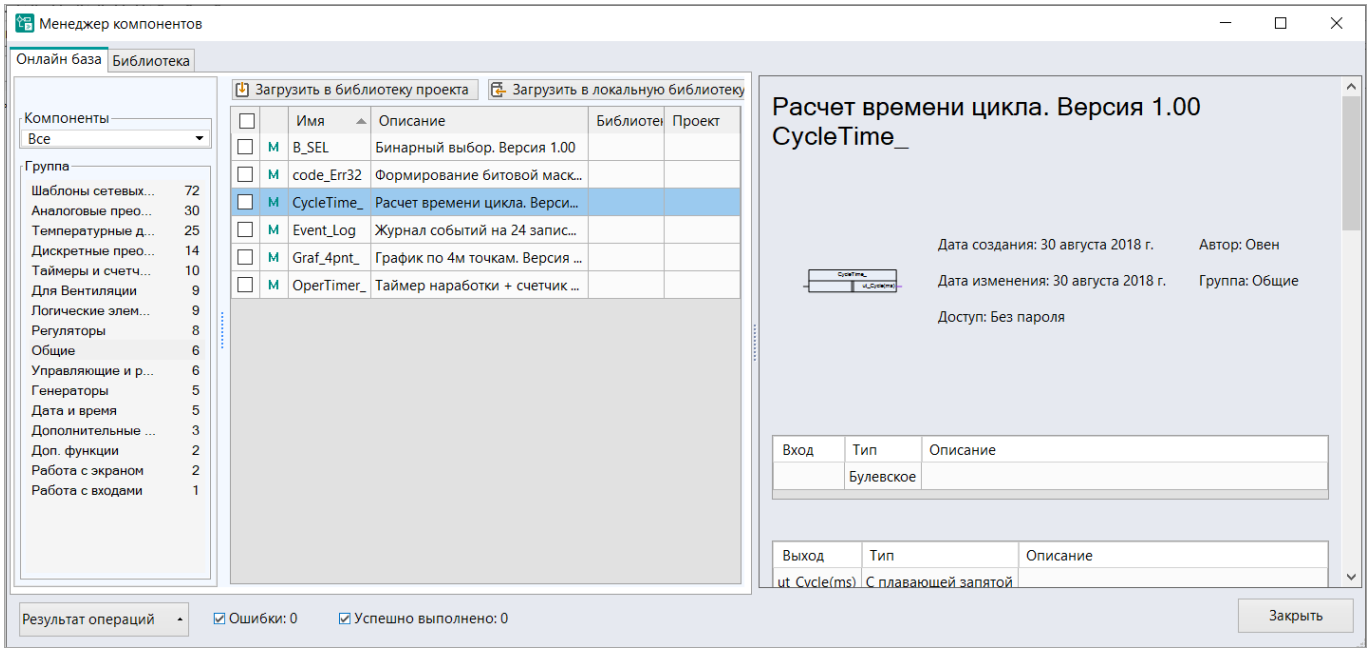


Рисунок 4.120

Чтобы компоненты стали доступны без подключения к Интернету, следует их отметить галочкой и нажать кнопку **Загрузить в локальную библиотеку** – компоненты загрузятся в локальное хранилище и будут отображаться во вкладке **Библиотека**.

Чтобы сразу загрузить компоненты в текущий проект, следует их отметить галочкой и нажать кнопку **Загрузить в библиотеку проекта** – компоненты будут добавлены в панель **Библиотека компонентов** в раздел **Макросы проекта**.

Успешная загрузка компонентов в локальное хранилище или текущий проект будет отображена в списке зеленой галочкой в графах **Библиотека** и **Проект** соответственно.

Для подключения сетевого шаблона к конфигурации прибора следует воспользоваться рекомендациями раздела Сетевые шаблоны устройств.

Вкладка «Библиотека»

На вкладке **Библиотека** отображается содержимое локального хранилища компонентов на ПК пользователя. Для загрузки компонентов в текущий проект следует выделить необходимые компоненты и нажать кнопку **Загрузить в библиотеку проекта** – компоненты будут добавлены в панель **Библиотека компонентов** в раздел **Макросы проекта**. Успешная загрузка компонентов в текущий проект будет отображена в списке зеленой галочкой в графе **Проект**.

ПРИМЕЧАНИЕ
Загрузить в библиотеку проекта компонент, содержащий массивы возможно начиная с версии Owen Logic3.0.

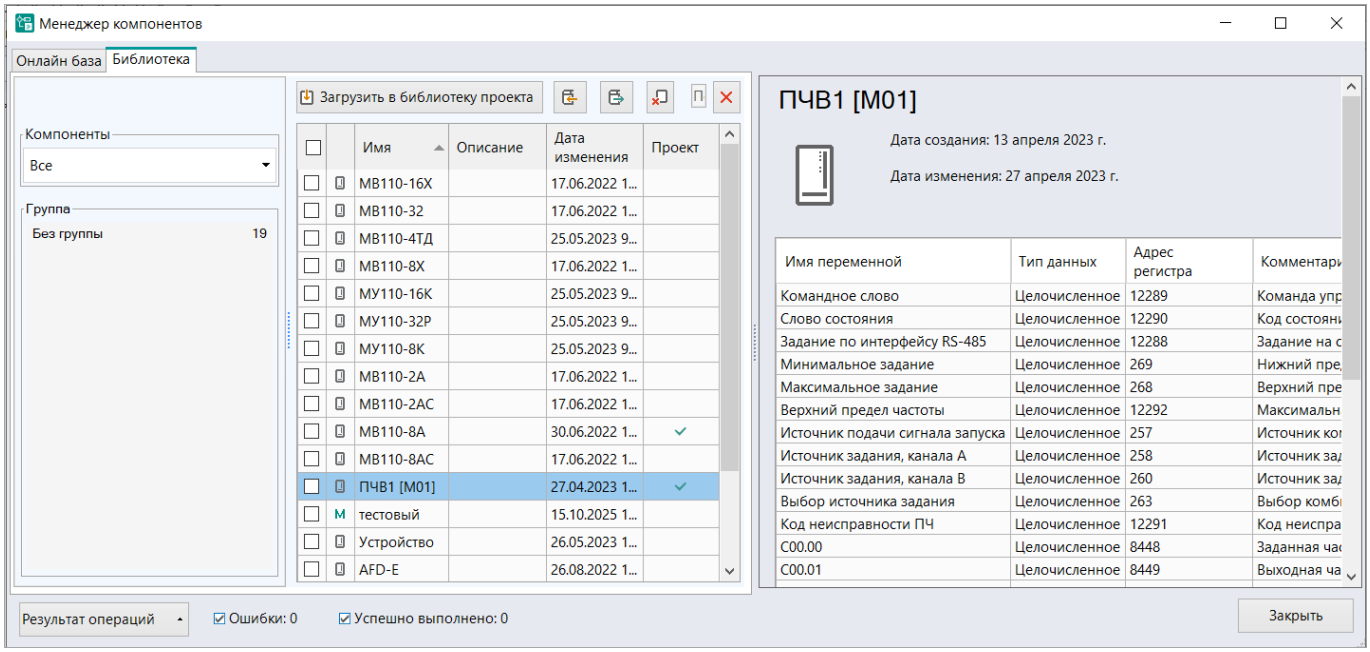


Рисунок 4.121

ПРИМЕЧАНИЕ
Локальное хранилище компонентов располагается по адресу: *C:\Users\[имя пользователя]\Documents\Owen Logic\Library*

Менеджер компонентов поддерживает следующие функции управления компонентами:

- – импорт файла в локальное хранилище;
- – экспорт файла из локального хранилища;
- – удаление выбранных компонентов.

Импорт компонента

С помощью импорта можно загрузить хранящиеся на ПК компоненты, но не входящие в состав локального хранилища.

Для импорта компонента следует:

1. Нажать кнопку .
2. В появившемся окне выбрать файлы для импорта.

3. Нажать кнопку **Открыть**. Файлы отобразятся на вкладке **Библиотека**.

Ограничения при импорте компонентов в более ранние версии Owen Logic

При импорте компонента, содержащего функционал, неподдерживаемый в версии, в которую осуществляется импорт:

- В версию Owen Logic 2.11.370 и выше — компонент будет загружен, но отобразится в **Менеджере компонентов** с ограничениями. В описании будет указана версия Owen Logic, начиная с которой поддерживается работа компонента.
- В версию Owen Logic ниже 2.11.370 — компонент будет загружен, однако при выборе его в библиотеке откроется окно с сообщением об ошибке.

При выполнении любых операций с такими компонентами будут отображаться ошибки. Использование в проекте компонентов, не поддерживаемых в текущей версии Owen Logic, невозможно.

Экспорт компонента

С помощью экспорта можно сохранить на ПК компоненты для дальнейшего изменения или передачи.

Для экспорта компонента следует:

1. Отметить галочкой необходимый компонент во вкладке **Библиотека**.
2. Нажать кнопку .
3. В появившемся окне выбрать путь для экспорта.
4. Нажать кнопку **ОК**. Компонент сохранится в виде файла по указанному пути.

Прочее

Для документации к компоненту доступны функции в соответствии с установленной версией Acrobat Reader, в том числе сохранение файла в формате pdf и печать.

Для отображения результатов операций в менеджере компонентов следует нажать кнопку Результат операций – в нижней части окна раскроется строка сообщений.

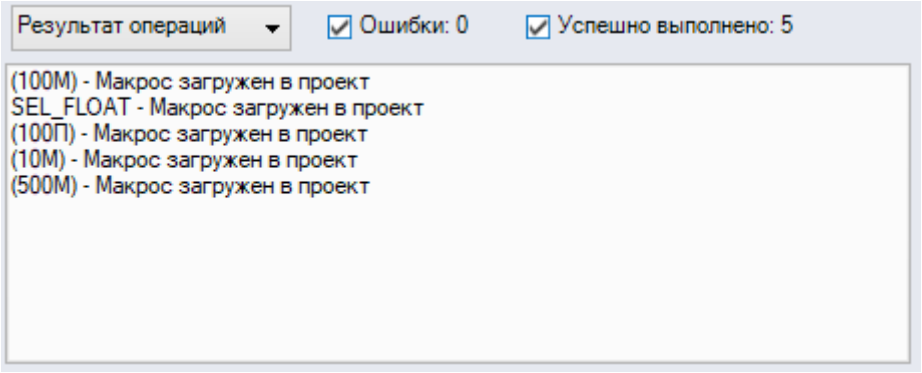


Рисунок 4.122

4.11 Работа с макросами

Макрос – это пользовательский алгоритм для использования в проекте.

Основные действия с макросами:

- **сохранение** в Библиотеке компонентов для дальнейшего использования в проекте;
- **экспортирование** в файл для использования в других проектах;
- **импортирование** созданных ранее макросов;
- **загрузка** из онлайн-базы в проект.

Создание макроса

Для создания макроса следует:

1. Выбрать в главном меню **Файл** → **Создать макрос**. В открывшемся диалоговом окне задать количество входов и выходов в макросе:

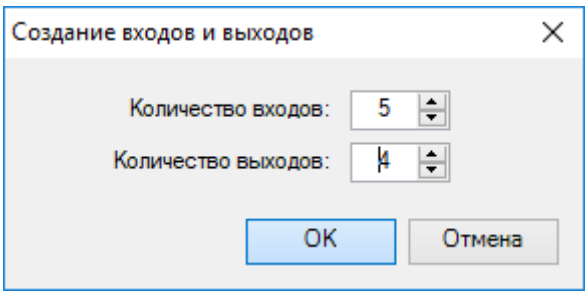


Рисунок 4.123

ПРИМЕЧАНИЕ
Количество входов и выходов можно изменить после создания макроса

2. Разработать алгоритм работы макроса во вкладке **Редактор макросов**, аналогично разработке программы на схеме.

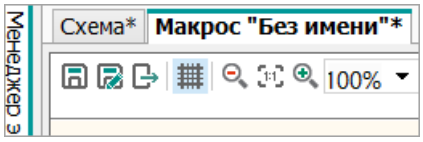


Рисунок 4.124

Количество входов и выходов можно увеличить с помощью контекстного меню редактора или с помощью кнопок I+ и Q+ .

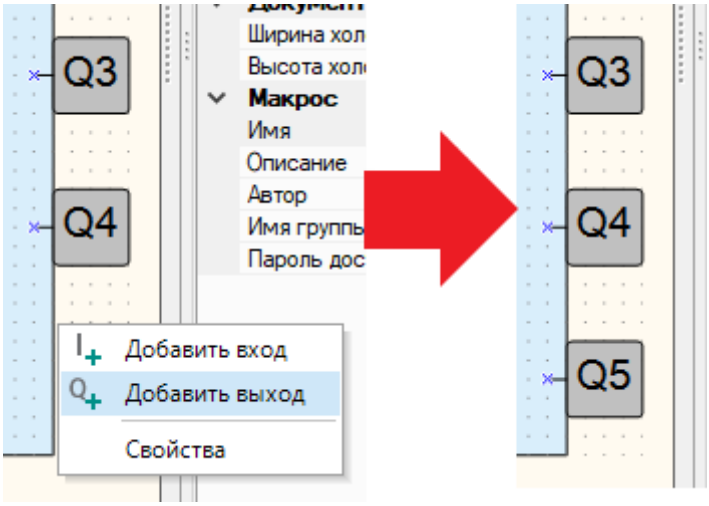


Рисунок 4.125

Чтобы удалить вход или выход, следует нажать по нему ПКМ и выбрать команду **Удалить**.

3. Задать имя и описание для макроса на панели свойств.

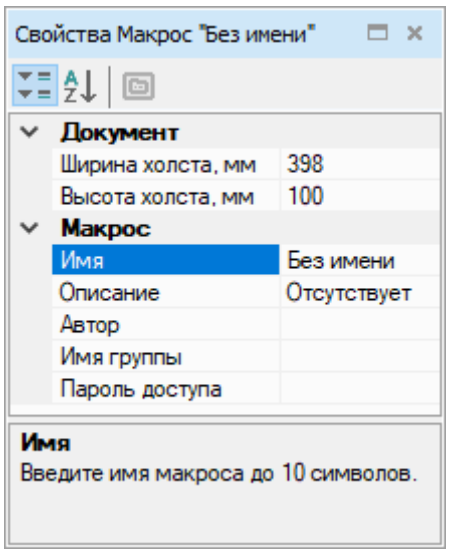


Рисунок 4.126

Введенное имя будет указано в заголовке редактора макросов и на схеме.

4. Для определения типа и имени входов/выходов выбрать вход или выход и настроить его на панели свойств.

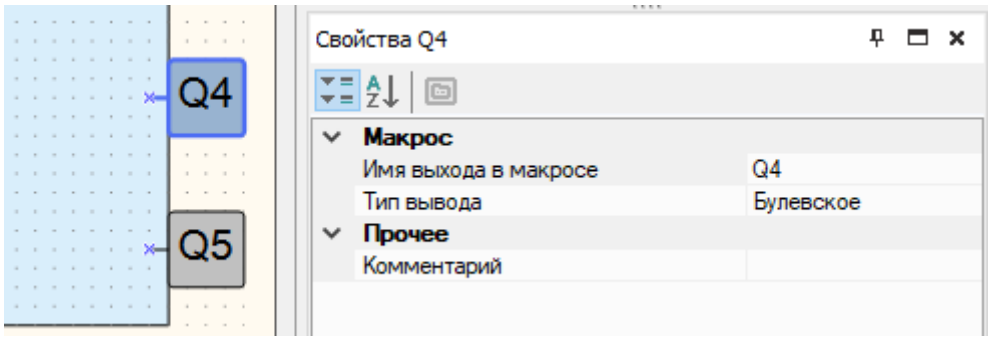


Рисунок 4.127

5. Далее можно задать «видимость» из программы уставок ФБ, использованных в макросе.

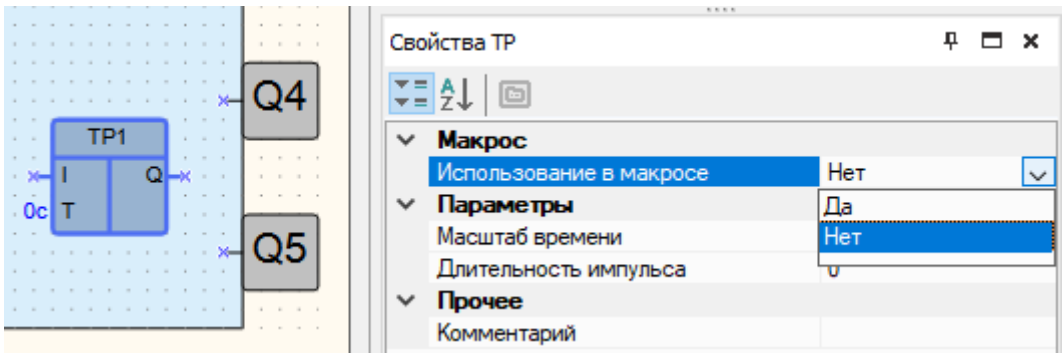


Рисунок 4.128

Чтобы уставки выбранного ФБ были доступны во время использования ФБ в программе, следует настроить на панели свойств блока параметр **Использование в макросе – Да**.

После выбора варианта **Да** появится выпадающий список **Переименование переменных** с перечислением доступных переменных. Например, для ФБ BLINK:

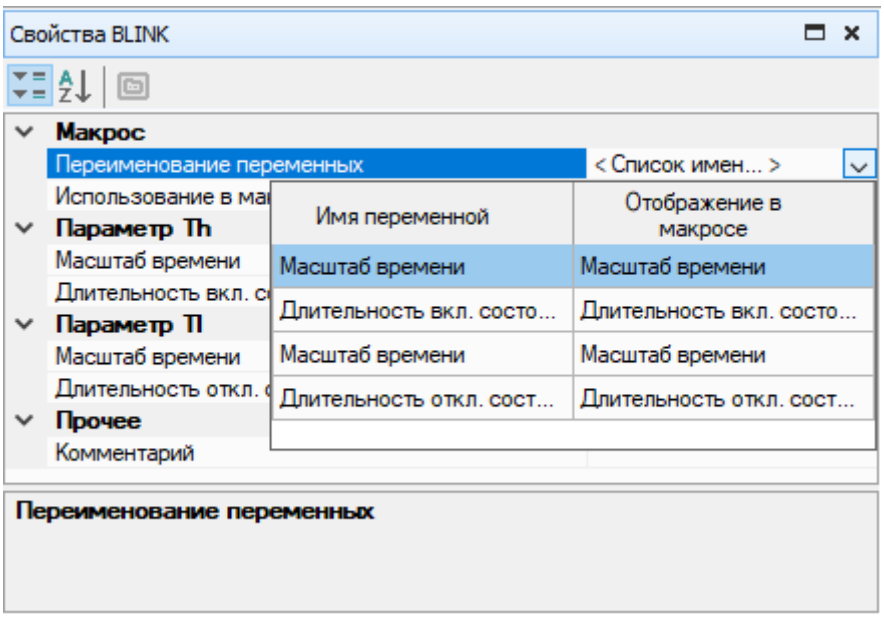


Рисунок 4.129

6. После создания макроса запустить режим симулятора 4.5 и проверить логику его работы.
7. Перед сохранением макроса можно заполнить поля: **Имя**, **Описание**, **Автор**, **Имя группы** и **Пароль доступа**.

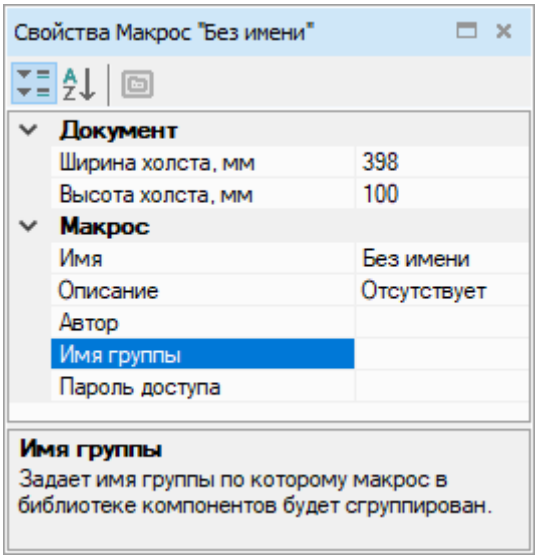


Рисунок 4.130

Рекомендуется задавать недлинное и понятное имя макроса. Содержимое поля **Описание** отображается под изображением макроса в панели **Библиотека компонентов**.

Если пароль не задан (поле **Пароль** не заполнено), то доступ к редактированию макроса открыт для всех.

Если не заполнено поле **Имя группы**, то макрос сохраняется в библиотеке проекта в группе **Другие**.

Макрос можно сохранить, выбрав в главном меню **Файл** → **Сохранить макрос как** или нажав на кнопку на панели редактора макросов.

Сохраненный макрос доступен только для открытого проекта, для открытия макроса в других проектах его следует экспортировать.

8. Выбрать раздел **Макросы проекта** на панели **Библиотека компонентов** и перенести макрос на холст проекта.

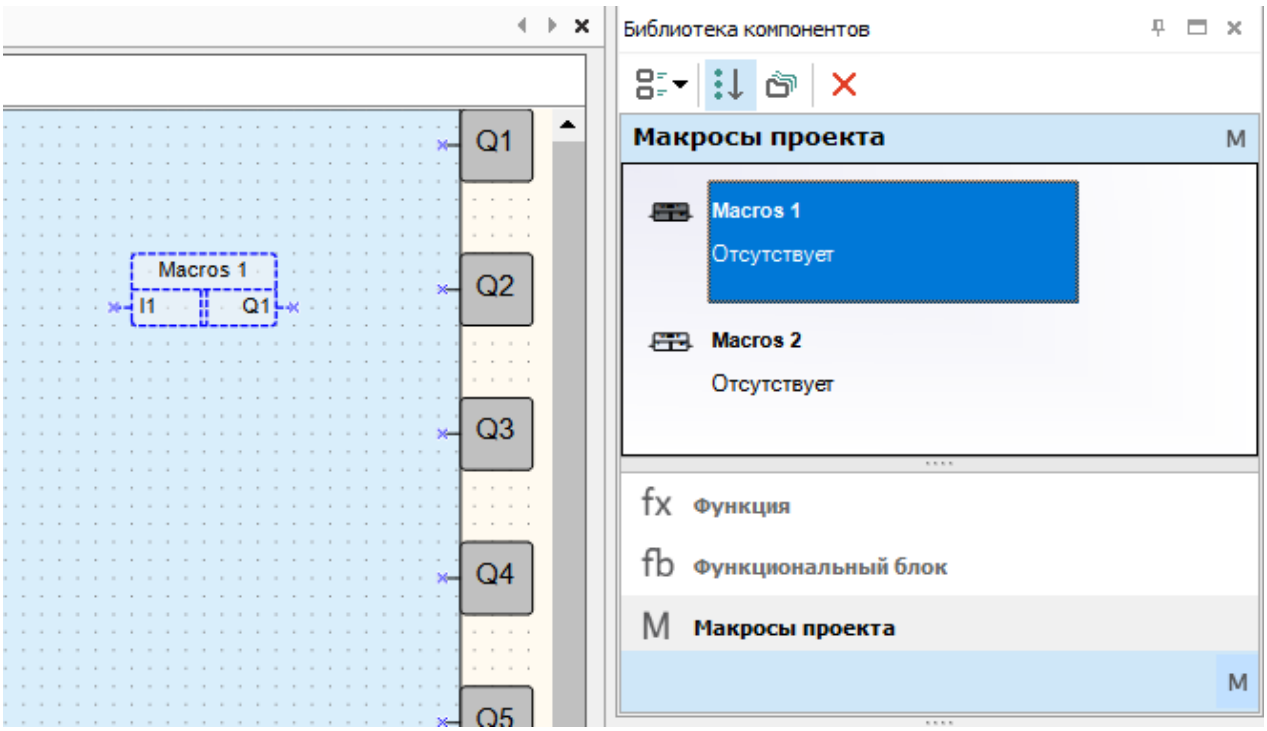


Рисунок 4.131

Создание макроса на схеме

Макрос можно создать, выделив элементы на схеме проекта и выбрав в контекстном меню **Создать макрос**. На месте элементов появится блок макроса.

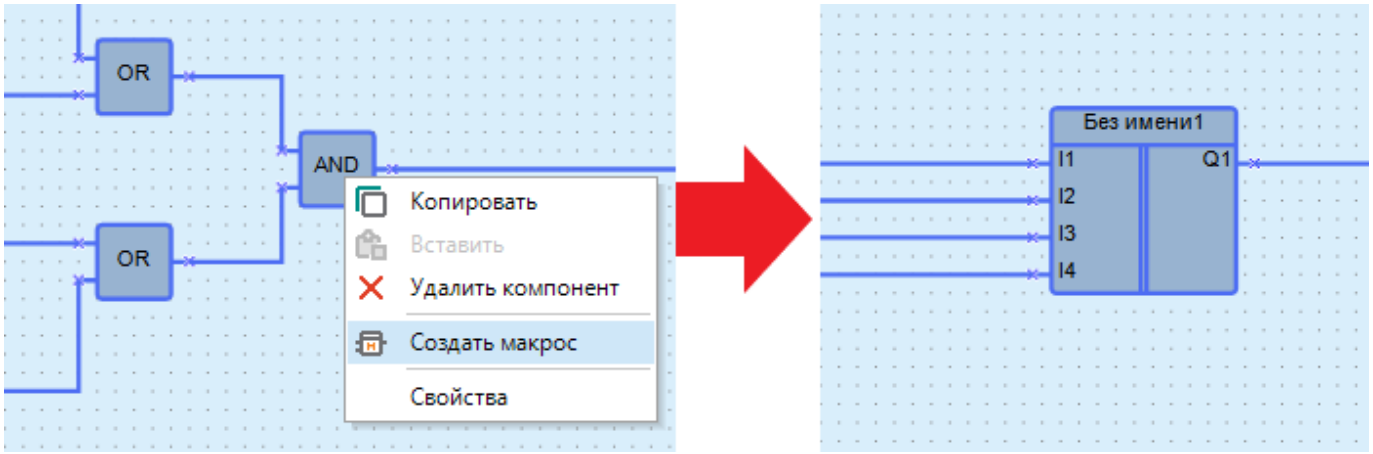


Рисунок 4.132

Входы и выходы макроса именуются согласно подсоединенным во время создания макроса элементам и блокам. Если к выходу макроса присоединено несколько элементов, то выход именуется по умолчанию (*Qn*, где *n* – порядковый номер выхода макроса).

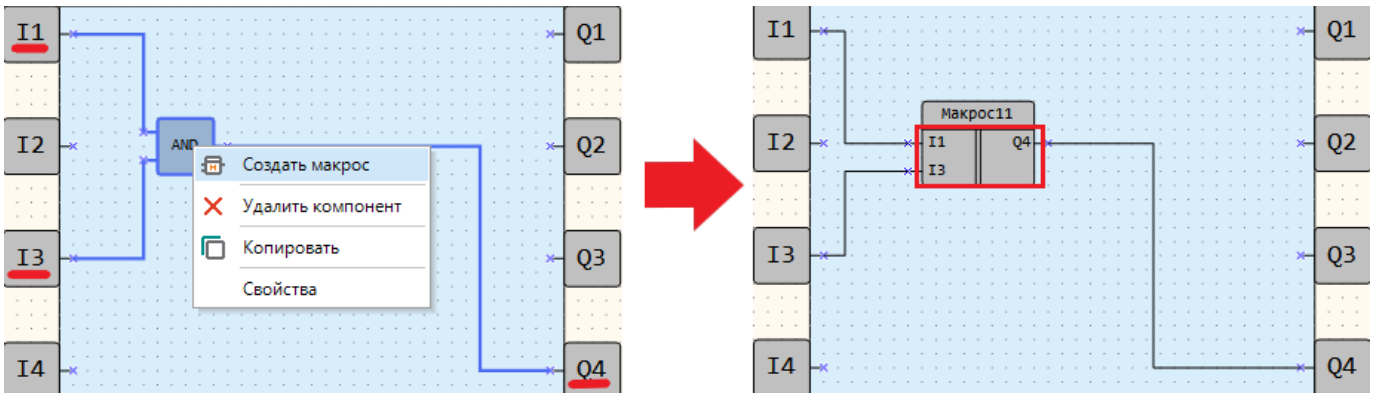


Рисунок 4.133

Созданный макрос и названия входов/выходов можно отредактировать в **редакторе макросов**, выбрав в контекстном меню блока **Редактировать макрос**.

Количество входов и выходов макроса будет соответствовать количеству соединенных входных и выходных связей в выделенном участке схемы. В случае выделения элементов без связей создается макрос с одним входом и выходом.

Элементы, не попадающие в макрос при создании на схеме:

- входы и выходы;
- сервисные переменные;
- сетевые переменные;
- ПИД-регулятор PID.

Если вышеуказанные элементы были выделены при создании макроса, то они будут соединены с соответствующим входом/выходом блока макроса. Если в выделенную область попал ФБ, в который записывается или считывается значение, то в макрос входит блок записи/чтения из ФБ, даже если он находится вне выделенной области.

В случае попадания стандартных переменных в область выделения, они копируются внутрь макроса под тем же именем. Переменные в макросе не тождественны переменным на схеме. Если все блоки одной переменной попадают внутрь макроса, то они исчезают со схемы.

Если при создании макроса в выделенную область попадает входной и выходной блок **задействованной** переменной, то внутри макроса будет создана переменная, наследующая все свойства выделенной переменной: тип, энергонезависимость, значение по умолчанию. Имя сформируется на основе имени исходной переменной, в конце имени добавится апостроф (').

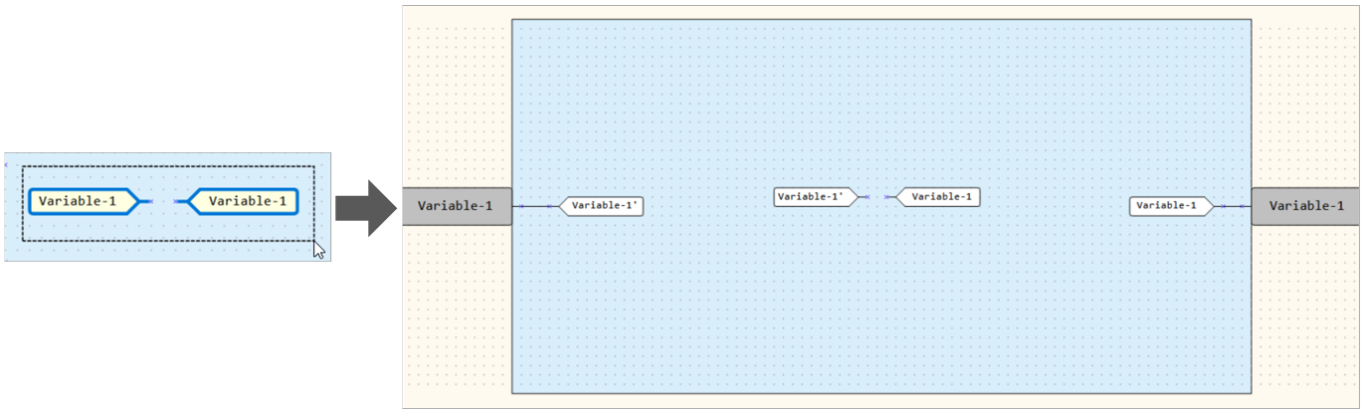


Рисунок 4.134

Те же правила применяются в случае, если при создании макроса в выделенную область попадает:

- входной и выходной блок **задействованного** элемента массива;
- входной и выходной блок массива, содержащего хотя бы один **задействованный** элемент;
- входной/выходной блок массива, содержащего хотя бы один **задействованный** элемент, и входной/выходной блок элемента **задействованного** массива.

Имя элемента массива сформируется с применением следующего форматирования: квадратные скобки ([ ]) будут заменены на фигурные ({ }), и в конце имение добавится апостроф (').

Пример
Mass_1[0] → Mass_1{0}'

Если при создании макроса в выделенную область попадает входной блок массива и входные блоки с элементами этого массива, то в макросе будет создан только один вход типа массив, с именем этого массива:

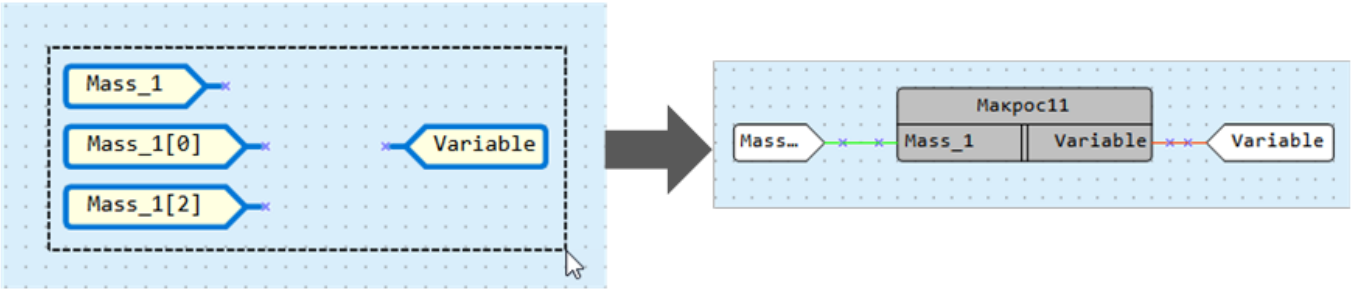


Рисунок 4.135

Если при создании макроса в выделенную область попадают входной и выходной блоки одной **незадействованной** переменной, которая присутствует за пределами выделенной области на схеме, то блоки копируются под тем же именем внутрь макроса и на схеме остается только блок выходной переменной.

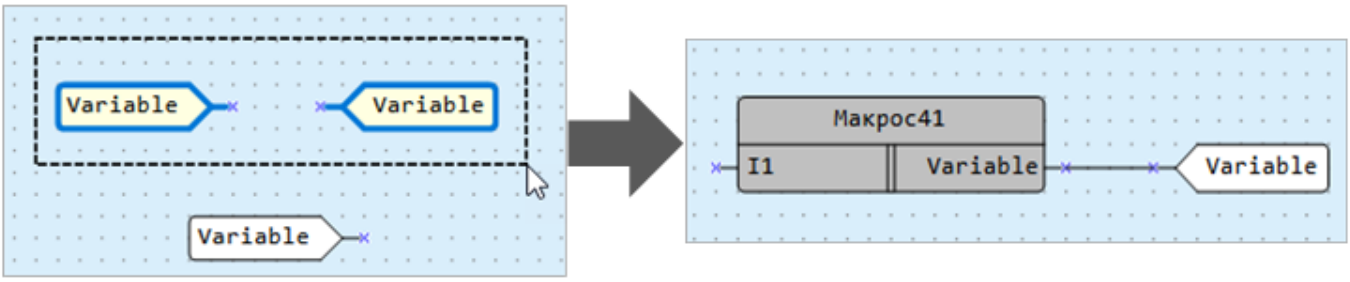


Рисунок 4.136

ПРИМЕЧАНИЕ

Обновление макроса в программе

Если у используемого в программе макроса были изменены имя, тип, количество входов/выходов или добавлен элемент (переменная, константа) со свойством **Использование в макросе – Да**, то Owen Logic потребует его обновить и отметит цветом на холсте. Макрос считается измененным после сохранения правок в **Редакторе макросов**.

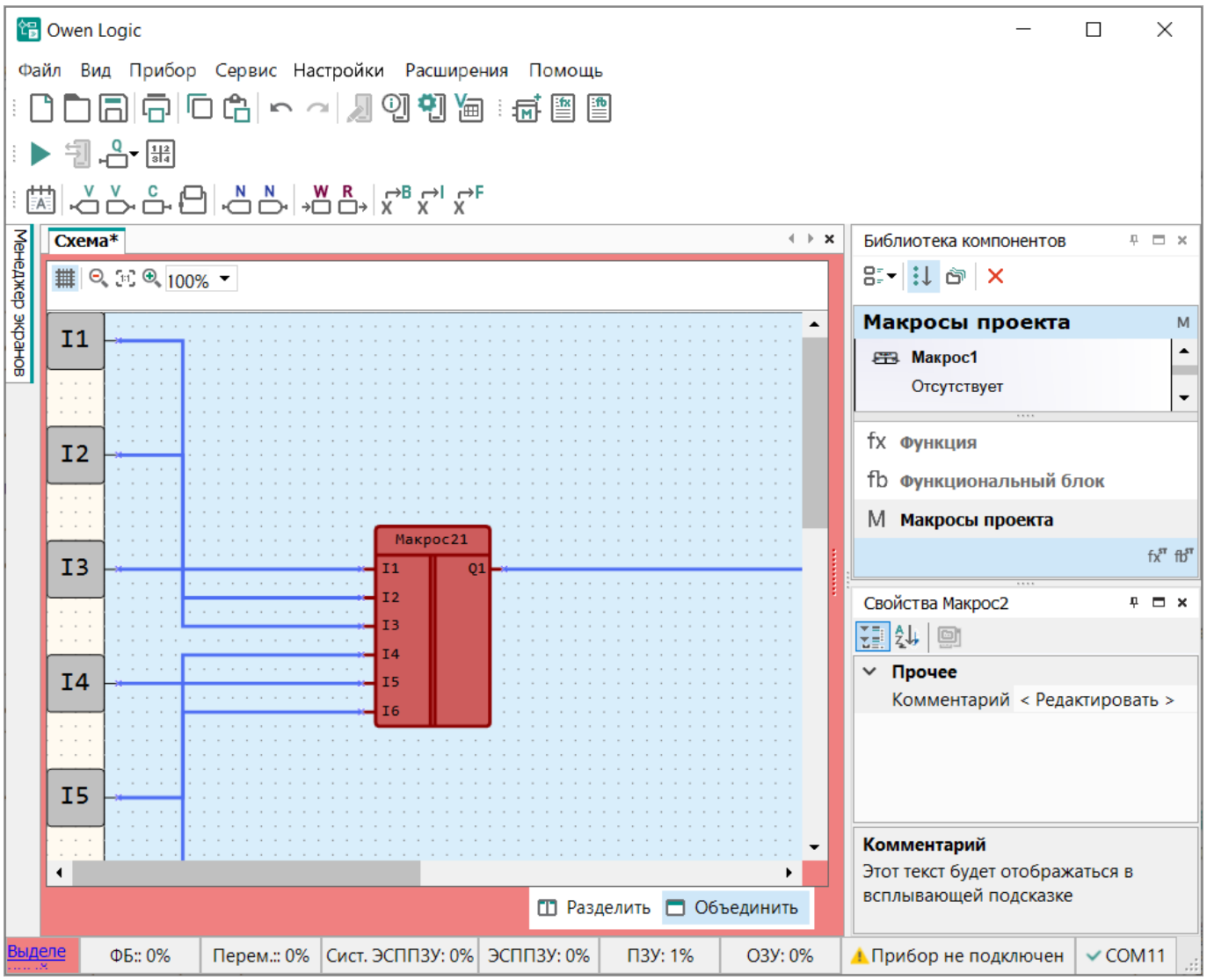


Рисунок 4.137

Для обновления макроса на холсте следует:

1. Нажать по отмеченному цветом макросу ПКМ и в появившемся меню выбрать **Обновить макрос**.

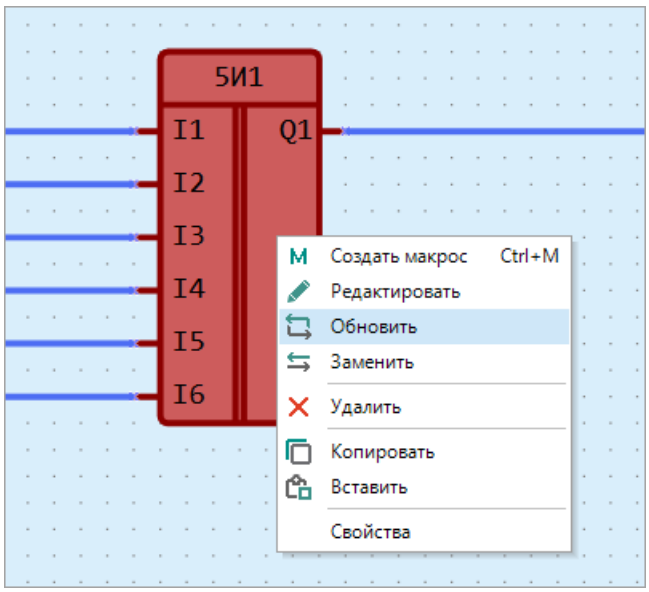


Рисунок 4.138

ПРИМЕЧАНИЕ
В случае вложенного макроса Owen Logic предложит перейти к нему. Действия с вложенным макросом аналогичны.

2. После обновления макроса Owen Logic автоматически перейдет к следующему макросу, требующему обновления.

Макрос обновляется в проекте по следующим правилам:

- если в макросе изменен тип или имя входа/выхода с присоединенной связью, при обновлении связь будет разорвана;
- если в макрос добавлены входы/выходы, при обновлении связи уже существующих входов/выходов не разрываются.

ПРИМЕЧАНИЕ
Owen Logic идентифицирует связи с входами/выходами макроса по имени и типу. Если в макросе изменить имя или тип входа/выхода с присоединенной связью и создать новый вход/выход с таким же именем и типом, то при обновлении макроса Owen Logic автоматически пересоединит связь к новому входу/выходу.

Быстрая замена макроса

Если необходимо заменить макрос на другой из макросов проекта, следует нажать кнопку **Заменить элемент** в контекстном меню макроса.

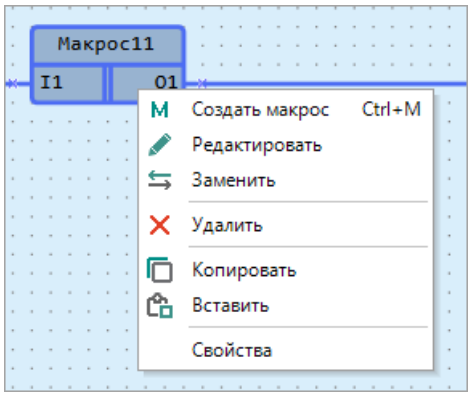


Рисунок 4.139

Откроется окно «Заменить макрос».

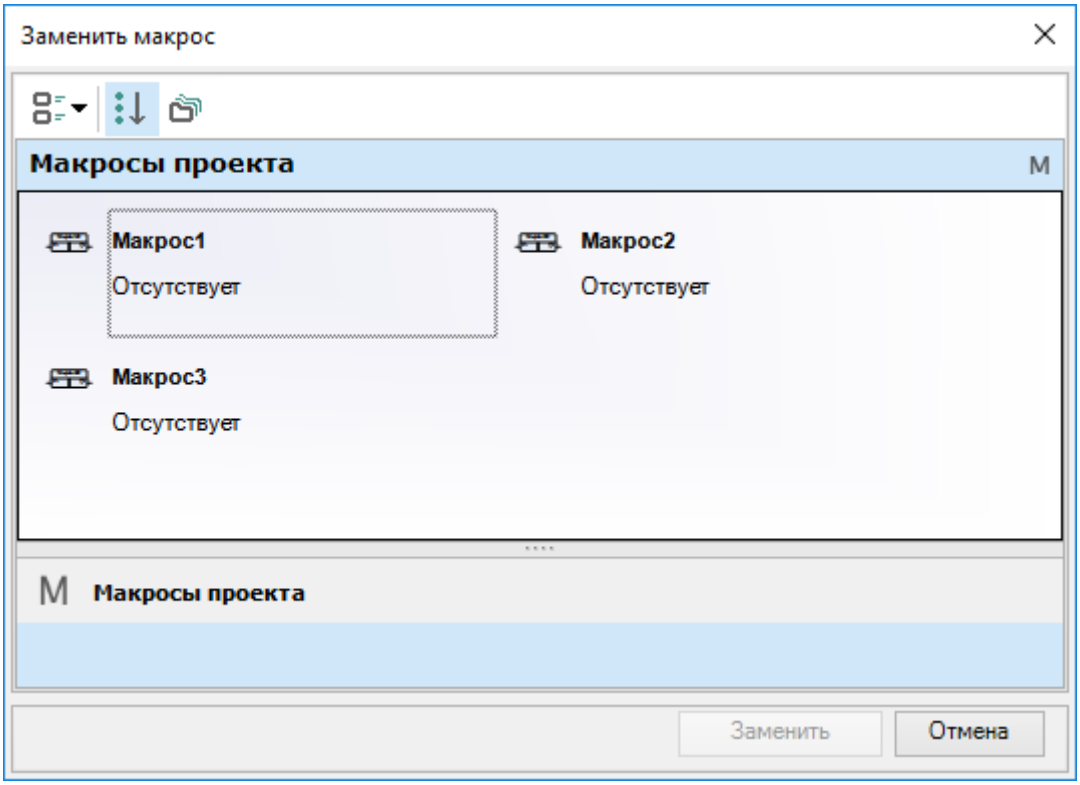


Рисунок 4.140

Если в макросе изменен тип или имя входа/выхода с присоединенной связью, то при замене связь будет разорвана, остальные связи будут сохранены.

Экспорт макроса

Экспортировать макрос в файл возможно только при открытом окне редактора макроса. Для экспорта макроса следует выбрать в главном меню **Файл** → **Экспорт в файл**.

Для экспорта макроса из панели библиотеки компонентов следует:

1. Открыть макрос в редакторе.

   Если перед сохранением макрос нужно отредактировать, то его следует перетащить на холст проекта и в контекстном меню макроса выбрать Редактировать и внести изменения.

2. Выбрать в главном меню **Файл** → **Экспорт в файл**.

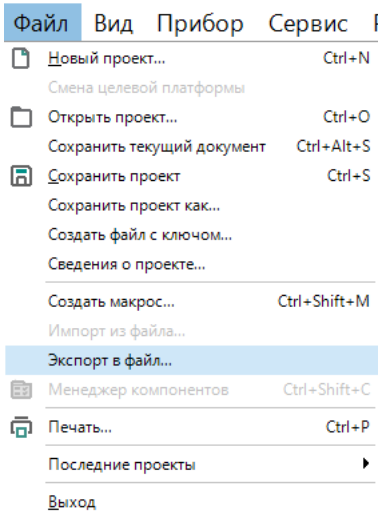


Рисунок 4.141

3. В открывшемся окне выбрать место расположения и сохранить файл макроса с расширением **.tpl*. После сохранения отобразится сообщение об успешном экспорте макроса.

Импорт макроса

Если для создания программы требуется использовать макрос, созданный в другом проекте, то необходимый макрос можно импортировать в проект.

Для импорта макроса следует выбрать в главном меню **Файл** → **Импорт из файла**.

ПРИМЕЧАНИЕ
Пункт **Импорт из файла** активен только при фокусе на редакторе схемы.

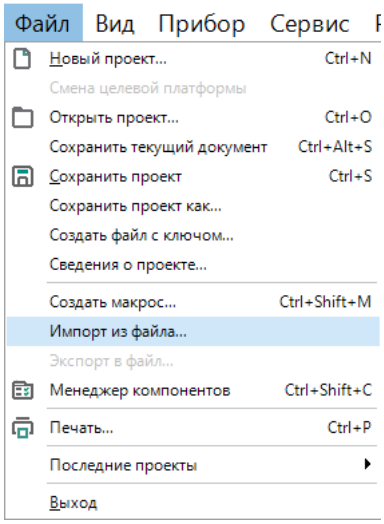


Рисунок 4.142

В открывшемся окне выбрать нужный файл и нажать кнопку **OK**. Макрос будет добавлен в панель **Библиотека компонентов** в раздел **Макросы проекта**, теперь его можно использовать в проекте.

ПРИМЕЧАНИЕ
Импорт макросов, содержащих переменные типа массив или элементы массивов возможен в версии Owen Logic 3.0 и выше.

Копирование макроса

Макрос можно скопировать из проекта в проект для повторного использования и сокращения времени на разработку. Для копирования макроса следует выделить блок макроса в исходном проекте и нажать кнопку на панели инструментов или выбрать команду **Копировать** в контекстном меню блока. Макрос вставляется в другой проект нажатием на кнопку на панели инструментов или с помощью выбора команды **Вставить** в контекстном меню холста. Для копирования и вставки также можно воспользоваться сочетаниями клавиш, см. раздел Сочетания клавиш.

После вставки, макрос будет доступен в разделе **Макросы проекта** на панели Библиотека компонентов.

ПРИМЕЧАНИЕ
Вставка макросов, содержащих переменные типа массив или элементы массивов возможна в версии Owen Logic 3.0 и выше.

4.12 Работа с элементами на языке ST

4.12.1 Работа с функциями на языке ST

В Owen Logic доступно создание пользовательских функций на языке ST (кроме приборов линейки ПР100 — исключая [M02]).

ПРЕДУПРЕЖДЕНИЕ
Функции на ST резервируют место в памяти ПЗУ после их добавления в библиотеку проекта, независимо от того используются они в проекте или нет.

Создание функции

Для создания функции следует:

1. Выбрать на панели инструментов кнопку **Создать функцию на ST**. Откроется вкладка “Редактор функции”.

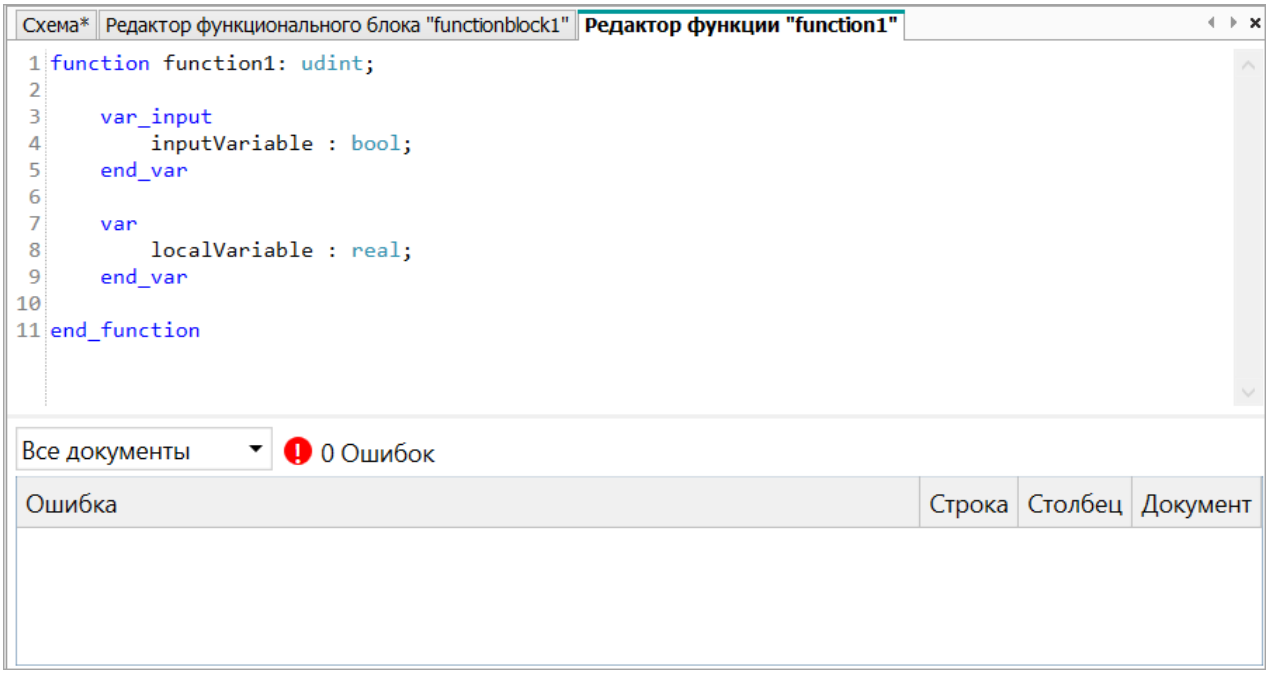


Рисунок 4.143

ПРИМЕЧАНИЕ
По умолчанию первые одиннадцать строк редактора заполнены шаблоном функции.

2. Задать имя функции и тип выходных данных в первой строке редактора.
3. Задать необходимое количество входных переменных в блоке объявления входных переменных (**var_input**).

4. Задать необходимое количество локальных переменных в блоке объявления локальных переменных (**var**).
5. Разработать алгоритм работы функции в соответствии с синтаксисом языка ST.

   ПРИМЕЧАНИЕ
   Подробнее см. Язык программирования ST.

6. Перейти во вкладку **Схема** или закрыть вкладку **Редактор функции**. Функция сохранится автоматически.
7. Выбрать раздел **Функции на ST** на панели **Библиотека компонентов** и перенести функцию на холст проекта.

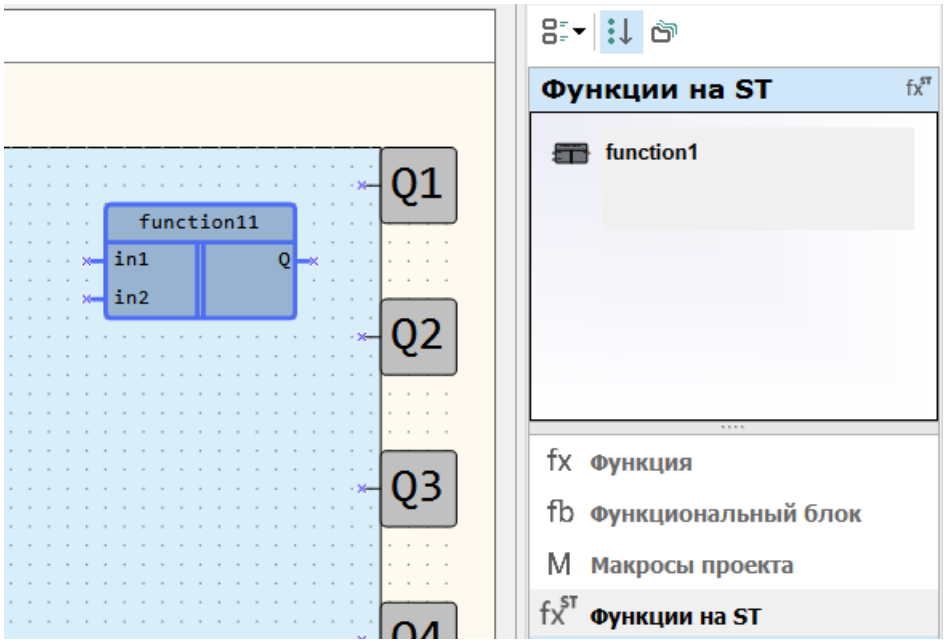


Рисунок 4.144

Редактор функции

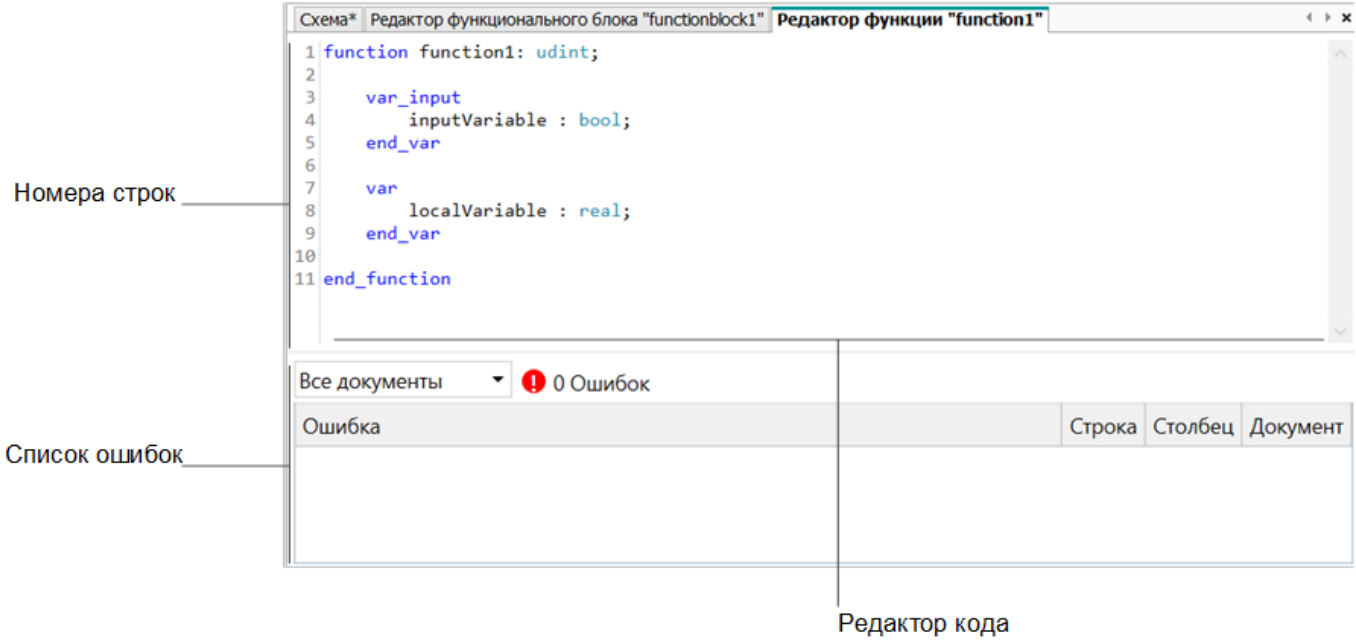


Рисунок 4.145

Элементы редактора функции:

- **Номера строк** — заполняется автоматически по количеству написанных строк кода.
- **Редактор кода** — область редактирования кода с функцией подсветки синтаксиса.
- **Список ошибок** — область отображения ошибок.

Сниппеты

Для удобства написания кода в редакторе функций реализованы сниппеты. Сниппет — фрагмент кода программы, который вставляется в код автоматически. В случае введения первого символа в редакторе откроется контекстное меню с фокусом на первой строке. Для выбора сниппета используются клавиши управления курсором. Чтобы вставить выбранный сниппет в код, следует нажать клавишу **Enter** или **Tab** или дважды нажать ЛКМ по пункту списка.

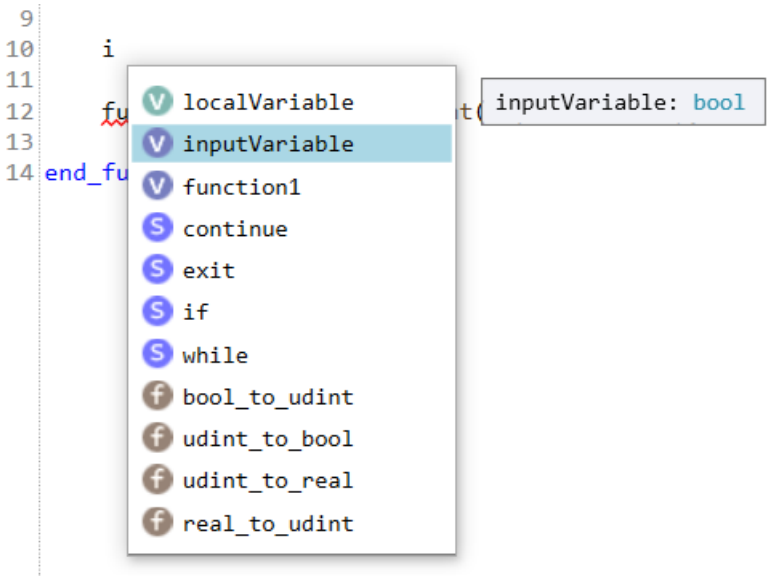


Рисунок 4.146

Список сниппетов формируется по группам:

- локальные переменные функции;
- конструкции (while, for и т. д.);
- выражения (true, false);
- встроенные функции;
- другие функции проекта.

Внутри групп сниппеты расположены по алфавиту.

Переход к месту объявления или использования

Для удобной работы с функциями и переменными доступен переход к местам их объявления или использования.

Для перехода к месту объявления:

1. Установить курсор на имя функции или переменной в коде программы.
2. Нажать ПКМ на имя функции или переменной.
3. В открывшемся контекстном меню выбрать пункт **Перейти к объявлению**.

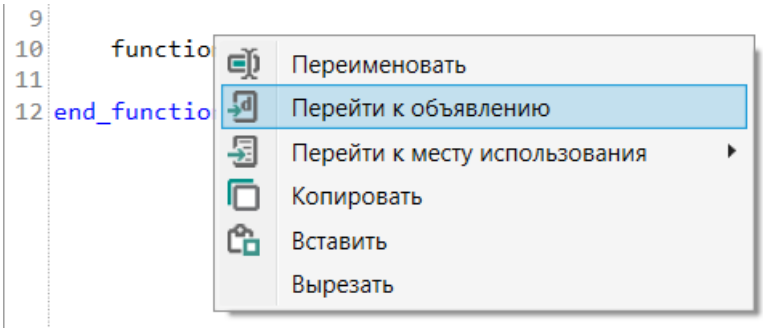


Рисунок 4.147

Для перехода к месту использования:

1. Установить курсор на имя функции или переменной в коде программы.
2. Нажать ПКМ на имя функции или переменной.

3. В открывшемся контекстном меню выбрать пукт **Перейти к месту использования**. Откроется список мест, в которых используется выбранная переменная или функция.

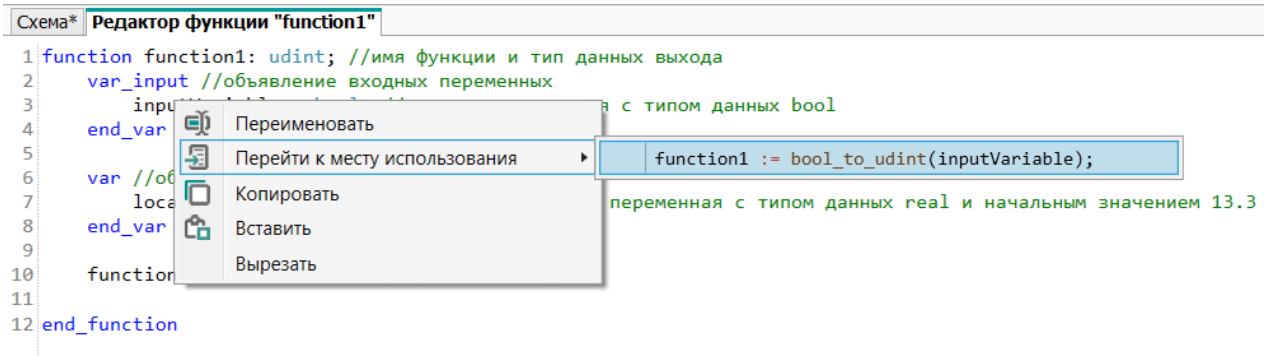


Рисунок 4.148

4. Нажать ЛКМ на выбранное место использования. Курсор переместится на строку, в которой используется функция или переменная.

Переименование переменной или функции

Для удобной работы с именами функций и переменных доступно централизованное переименование.

Для переименования следует:

1. Установить курсор на имя функции или переменной в коде программы.
2. Нажать ПКМ на имя функции или переменной.
3. В открывшемся контекстном меню выбрать пункт **Переименовать**. Имя выделится зеленым во всех местах использования.

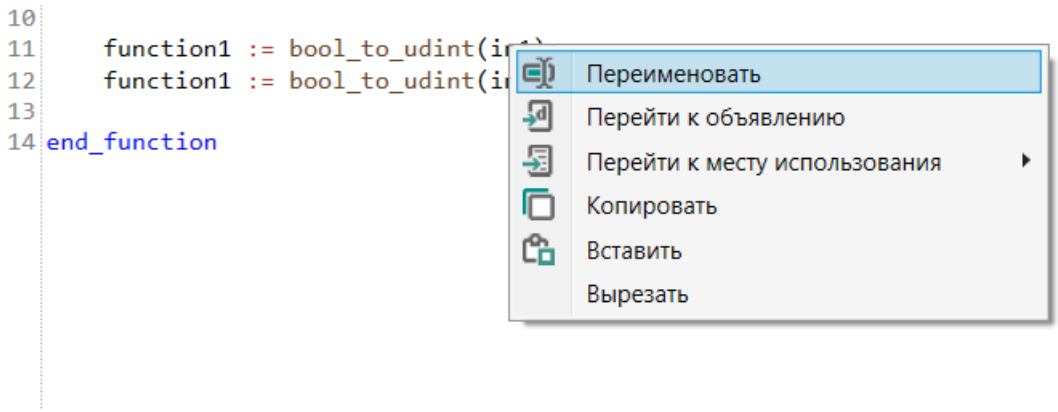


Рисунок 4.149

4. Ввести новое имя переменной в области объявления переменных.

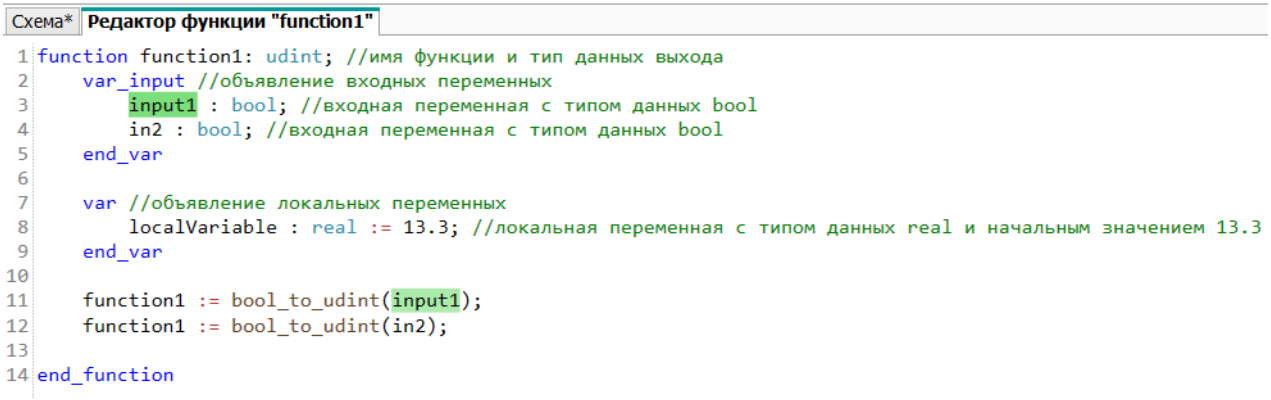


Рисунок 4.150

5. Переместить курсор на другую строку в редакторе функций. Переменная будет переименована во всех местах использования.

Список ошибок

В области отображения ошибок формируется таблица с описанием и расположением ошибок, допущенных в коде. Для перехода в место ошибки следует дважды нажать ЛКМ по строке с ошибкой.

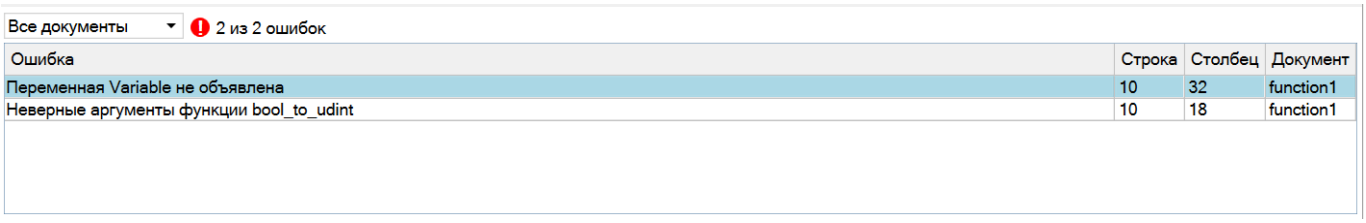


Рисунок 4.151

Экспорт функции на ST

Экспортировать функцию в файл возможно только при открытой вкладке редактора функции. Для экспорта функции следует выбрать в главном меню **Файл** → **Экспорт в файл**.

Для экспорта макроса из панели библиотеки компонентов следует:

1. Открыть функцию в редакторе.
2. Выбрать в главном меню **Файл** → **Экспорт в файл**.

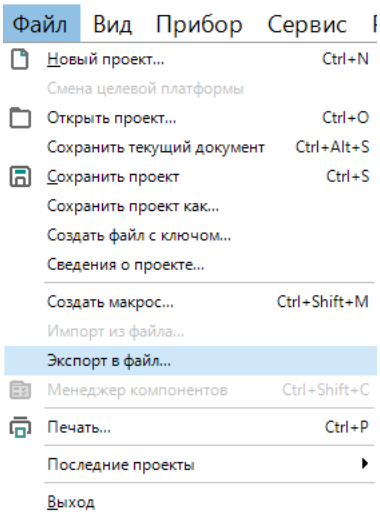


Рисунок 4.152

3. В открывшемся окне выбрать место расположения и сохранить файл функции с расширением **.fst*. После сохранения отобразится сообщение об успешном экспорте функции.

Импорт функции на ST

Если для создания программы требуется использовать функцию, созданную в другом проекте, то ее можно импортировать в проект.

Для импорта функции следует выбрать в главном меню **Файл** → **Импорт из файла**.

ПРИМЕЧАНИЕ
Пункт **Импорт из файла** активен только при фокусе на редакторе схемы.

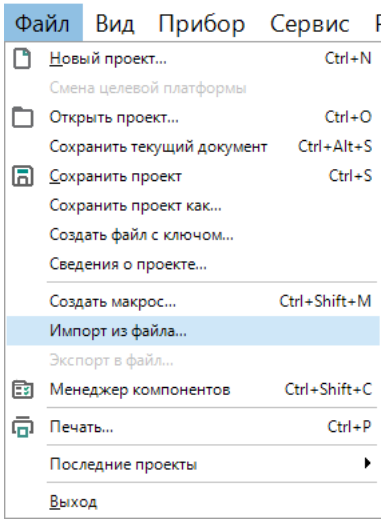


Рисунок 4.153

В открывшемся окне выбрать нужный файл и нажать кнопку **ОК**. Функция будет добавлен в панель **Библиотека компонентов** в раздел **Функции на ST**, теперь его можно использовать в проекте.

4.12.2 Работа с функциональными блоками на языке ST

Создание пользовательских функциональных блоков на языке ST не доступно для приборов линейки ПР100 (кроме [M02]).

Создание функционального блока

Для создания функционального блока следует:

1. Выбрать на панели инструментов кнопку **Создать функциональный блок на ST**. Откроется вкладка “Редактор функционального блока”.

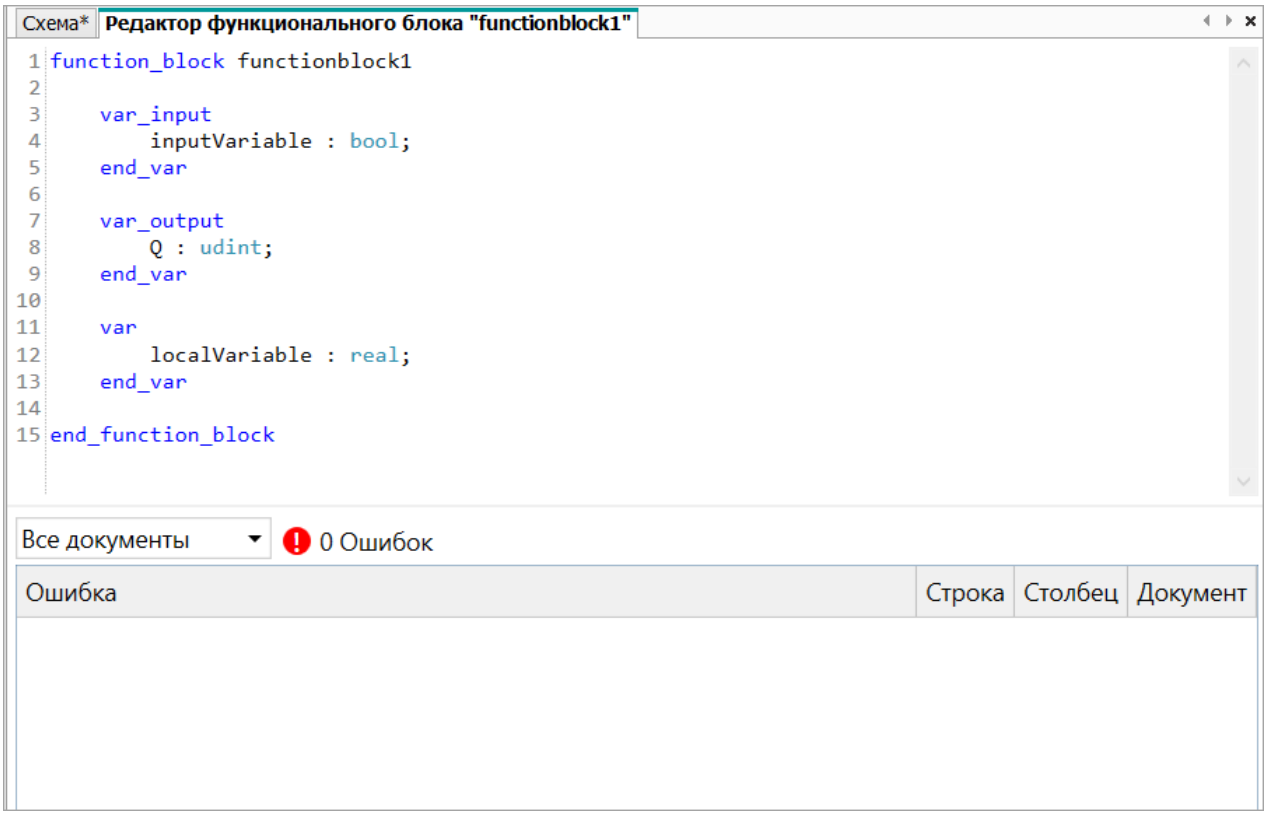


Рисунок 4.154

ПРИМЕЧАНИЕ
По умолчанию первые пятнадцать строк редактора заполнены шаблоном функции.

2. Задать имя функционального блока в первой строке редактора.
3. Задать необходимое количество входных переменных в блоке объявления входных переменных (**var_input**).
4. Задать необходимое количество выходных переменных в блоке объявления входных переменных (**var_output**).
5. Задать необходимое количество локальных переменных в блоке объявления локальных переменных (**var**).
6. Разработать алгоритм работы функционального блока и в соответствии с синтаксисом языка ST.

   ПРИМЕЧАНИЕ
   Подробнее см. Язык программирования ST.

7. Перейти во вкладку **Схема** или закрыть вкладку **Редактор функционального блока**. Функциональный блок сохранится автоматически.
8. Выбрать раздел **Функциональные блоки на ST** на панели **Библиотека компонентов** и перенести функцию на холст проекта.

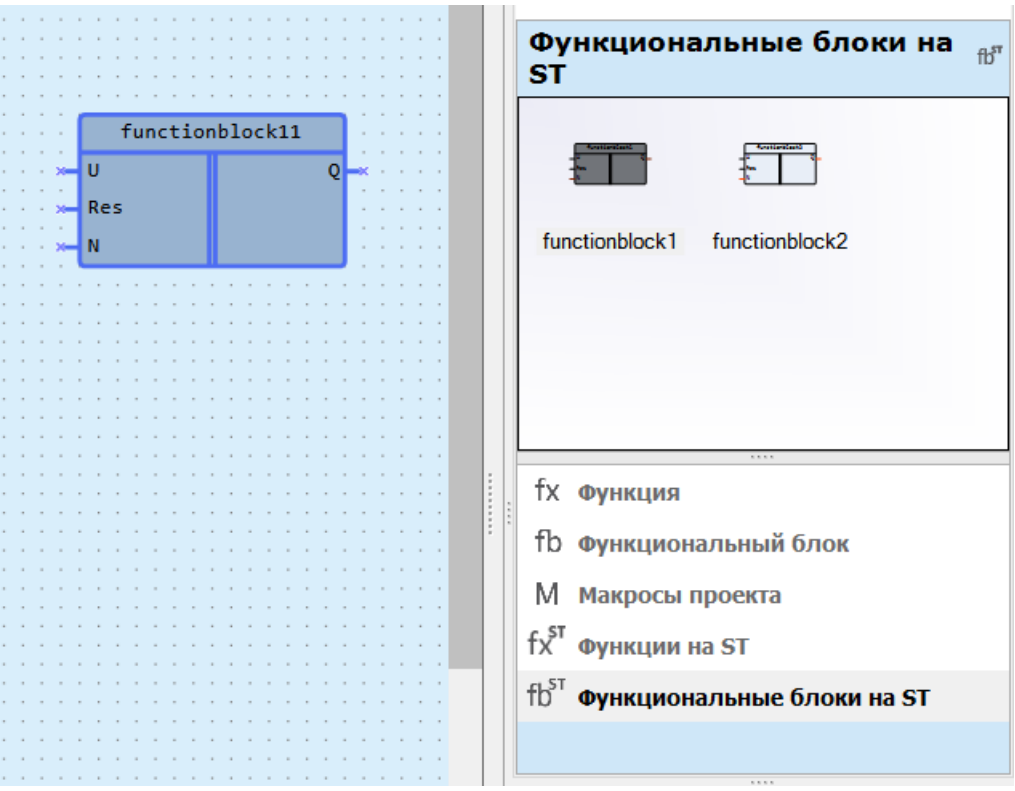


Рисунок 4.155

Редактор функционального блока

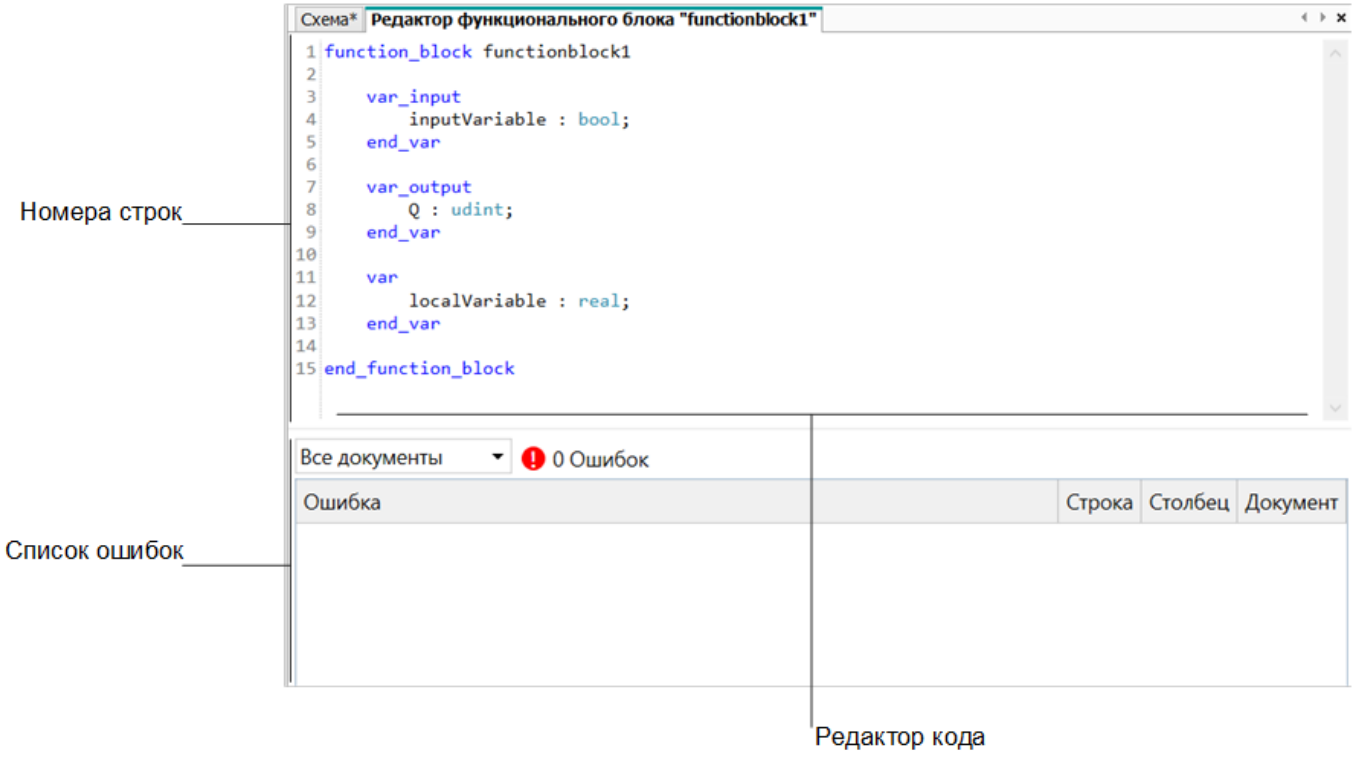


Рисунок 4.156

Элементы редактора функционального блока:

- **Номера строк** — заполняется автоматически по количеству написанных строк кода.
- **Редактор кода** — область редактирования кода с функцией подсветки синтаксиса.
- **Список ошибок** — область отображения ошибок.

ПРИМЕЧАНИЕ
Аналогично редактору функции редактор функционального блока поддерживает функции сниппетов , отслеживания места объявления и использования , переименования и отслеживания ошибок .

Экспорт функционального блока на ST

Экспортировать функциональный блоке в файл возможно только при открытой вкладке редактора функционального блока. Для экспорта функционального блока следует выбрать в главном меню **Файл** → **Экспорт в файл**.

Для экспорта функционального блока из панели библиотеки компонентов следует:

1. Открыть функциональный блок в редакторе.
2. Выбрать в главном меню **Файл** → **Экспорт в файл**.

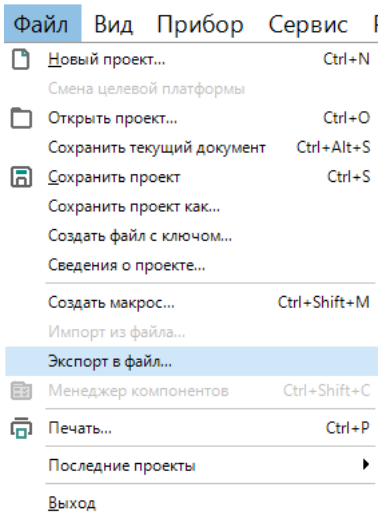


Рисунок 4.157

3. В открывшемся окне выбрать место расположения и сохранить файл функционального блока с расширением **.fbst*. После сохранения отобразится сообщение об успешном экспорте функционального блока.

Импорт функционального блока на ST

Если для создания программы требуется использовать функциональный блок, созданный в другом проекте, то его можно импортировать в проект.

Для импорта функционального блока следует выбрать в главном меню **Файл** → **Импорт из файла**.

ПРИМЕЧАНИЕ
Пункт **Импорт из файла** активен только при фокусе на редакторе схемы.

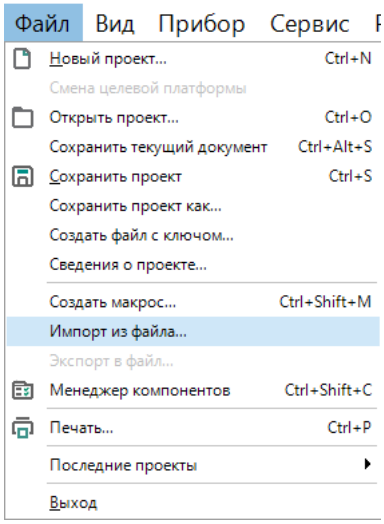


Рисунок 4.158

В открывшемся окне выбрать нужный файл и нажать кнопку **OK**. Функциональный блок будет добавлен в панель **Библиотека компонентов** в раздел **Функциональные блоки на ST**, теперь его можно использовать в проекте.

5 Настройка прибора

Чтобы открыть окно настроек прибора нажмите кнопку на панели инструментов или выберите в главном меню **Прибор** → **Настройка прибора**.

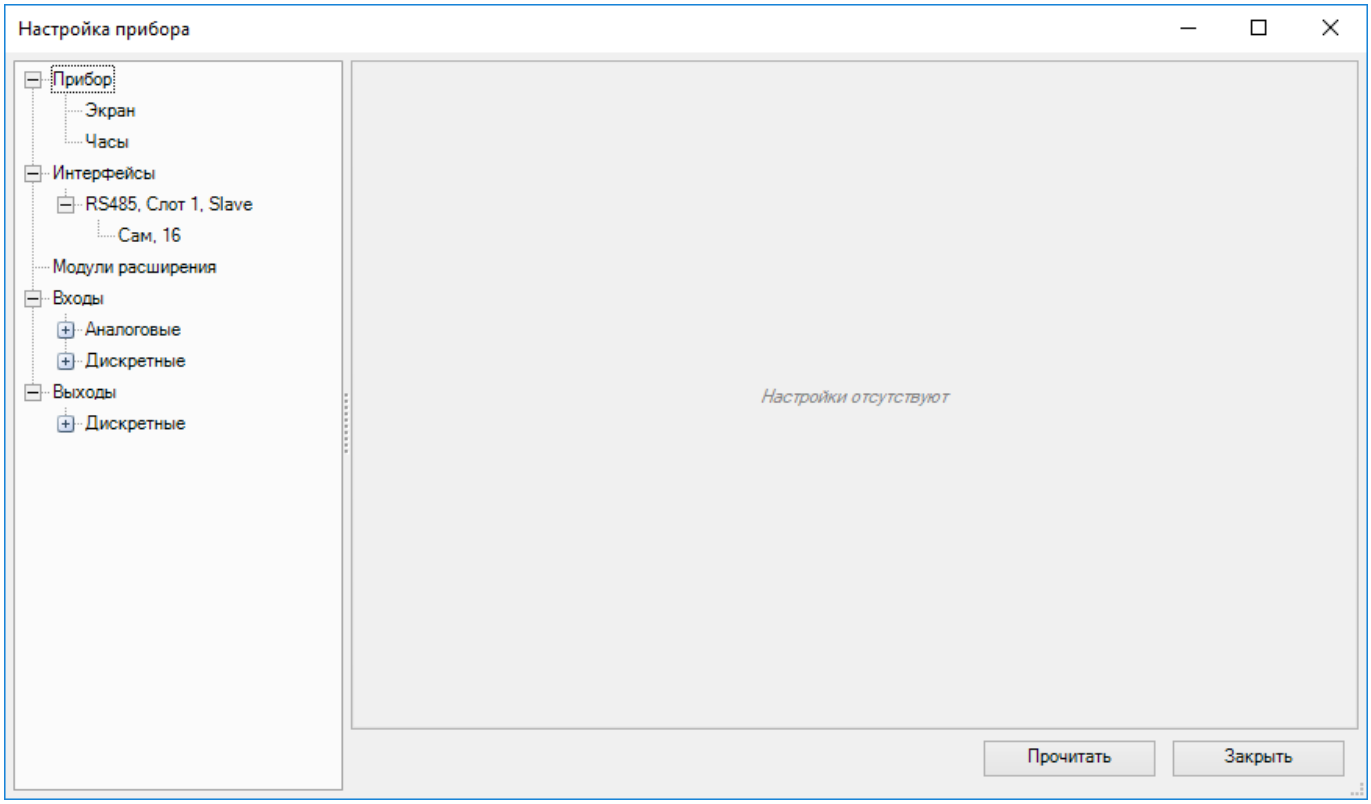


Рисунок 5.1

В левой части окна настроек прибора содержится дерево с разделами:

- Экран;
- Часы;
- Интерфейсы;
- Модули расширения;
- Входы и выходы.

ПРИМЕЧАНИЕ
Перечень разделов может отличаться для разных моделей приборов.

В правой части окна настроек прибора отображаются настройки выбранного раздела.

Все установленные настройки для прибора, за исключением часов, сохраняются вместе с проектом. Настройки прибора не требуют подключения прибора.

Настройки для приборов второго поколения

Для приборов второго поколения окно настроек прибора имеет другой интерфейс и содержит больше разделов. Параметры в правой части окна отображаются в табличном виде.

В левой части окна настроек прибора в дереве помимо основных групп доступны дополнительно настройки:

- Ethernet;
- Подключение к OwenCloud;
- Пароль;

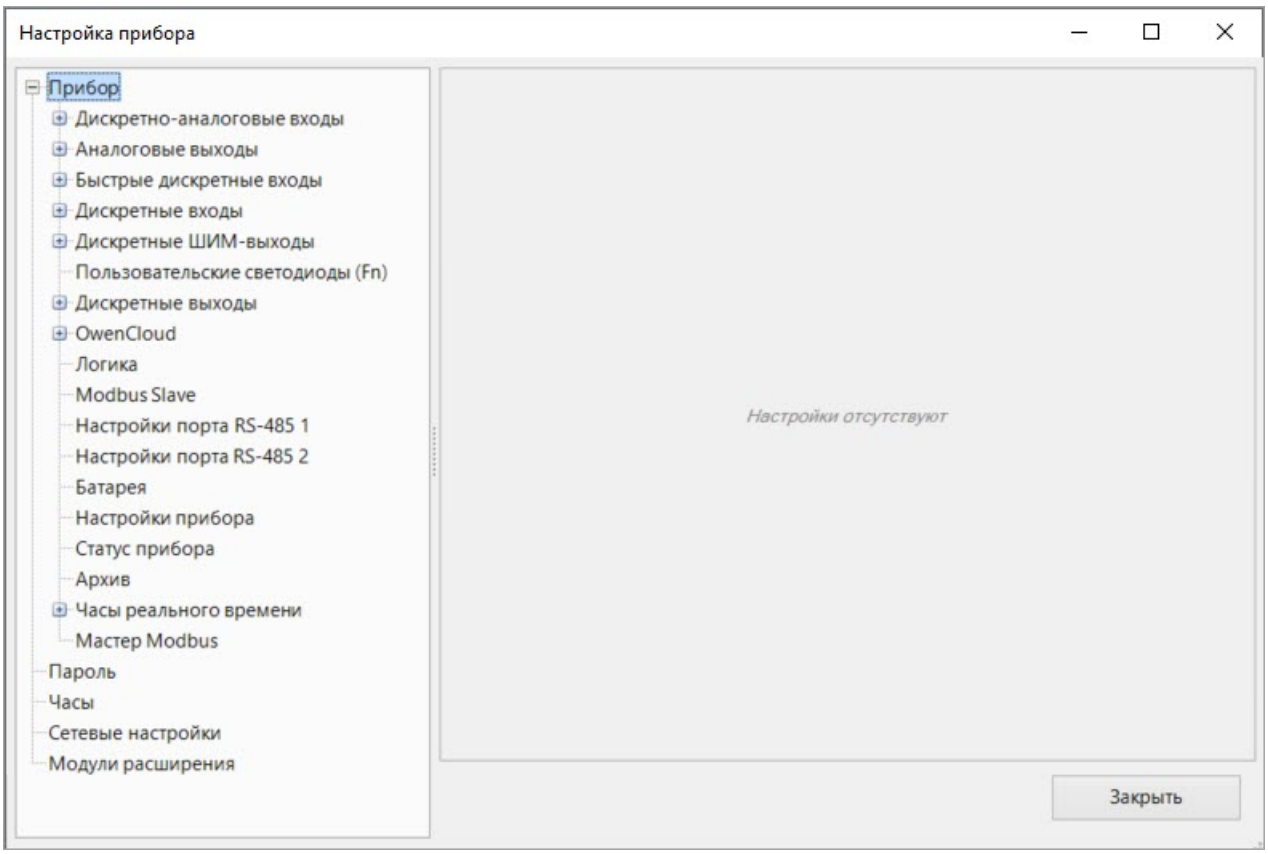


Рисунок 5.2

5.1 Экран

Настройки экрана доступны только для приборов с дисплеем.

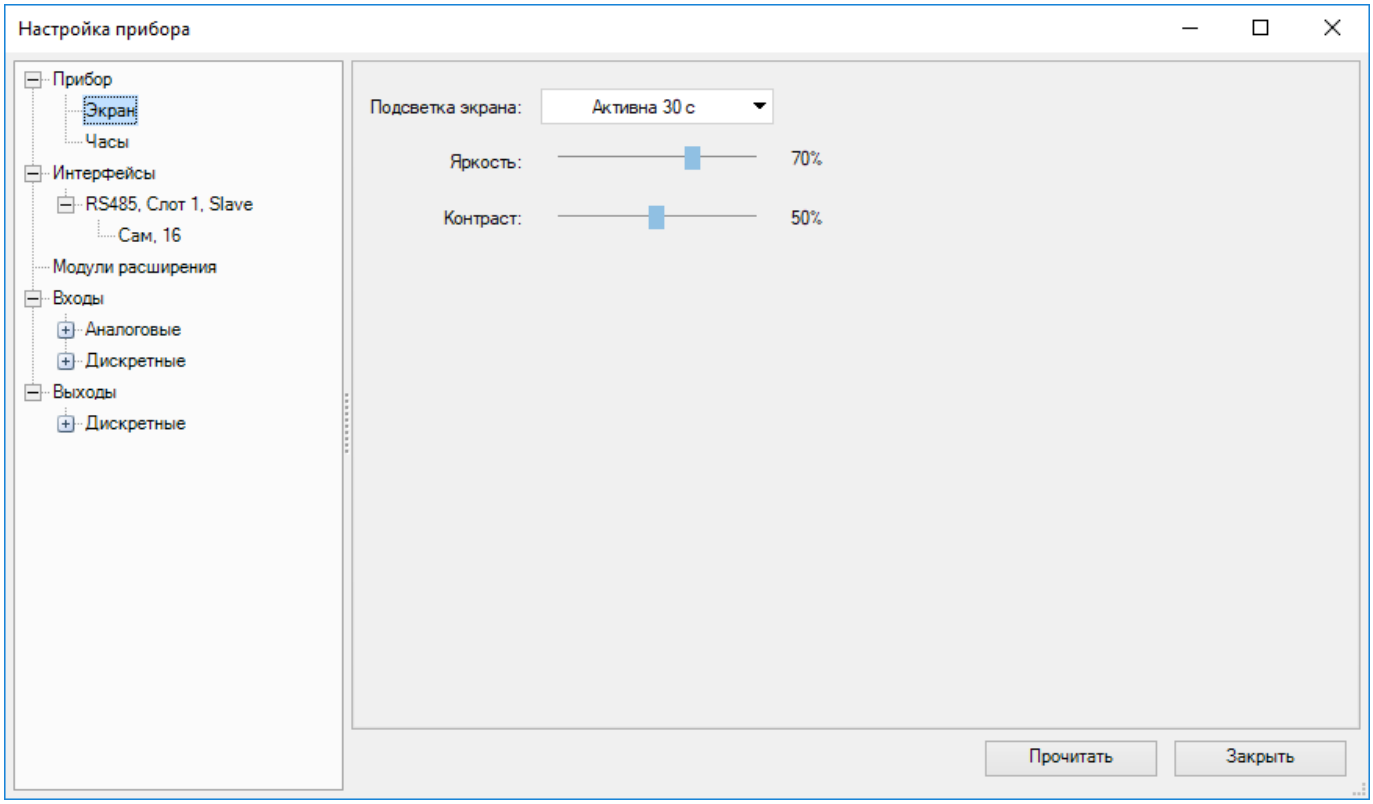


Рисунок 5.3

Перечень параметров зависит от типа прибора.

Подсветка экрана — в выпадающем меню можно выбрать длительность работы подсветки с момента последнего действия пользователя.

Яркость — настройка яркости экрана, от 0 до 100 %.

Контраст — настройка контрастности отображаемых символов, от 0 до 100 %.

Кнопка **Прочитать** используется для считывания текущих настроек экрана из подключенного прибора.

5.2 Часы

В разделе **Часы** настраиваются часы реального времени прибора. Настройки часов реального времени требуются для работы системных переменных и ФБ программы.

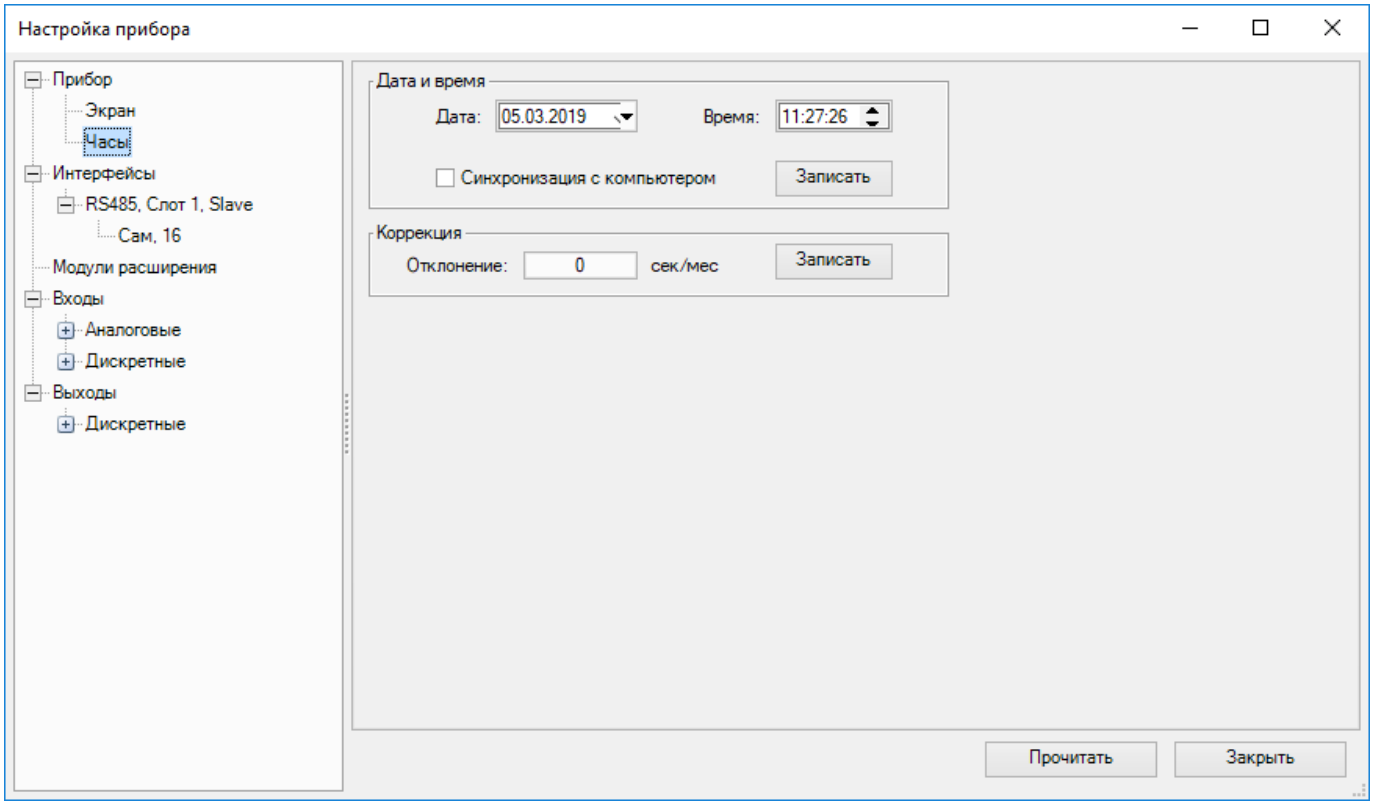


Рисунок 5.4

Дата и время

Дата и время задаются в соответствующих полях.

Для синхронизации часов реального времени прибора с часами ПК следует отметить галочку **Синхронизация с компьютером** — поля **Дата** и **Время** станут неактивны.

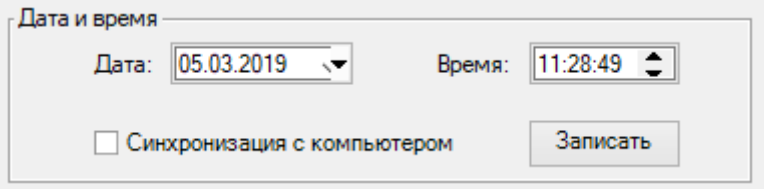


Рисунок 5.5

Чтобы задать новые значения для часов реального времени прибора, следует нажать кнопку **Записать**.

Коррекция

Поле **Отклонение** служит для указания погрешности хода часов реального времени прибора в секундах. Если часы реального времени прибора спешат, то значение указывается со знаком минус. Коррекция на указанную величину происходит равномерно в течение месяца.

Рисунок 5.6

Чтобы задать новое значение коррекции для часов реального времени прибора, следует нажать кнопку **Записать**.

Кнопка **Прочитать** используется для считывания текущих настроек часов реального времени из подключенного прибора.

Настройки для приборов второго поколения

ПРИМЕЧАНИЕ
Список приборов второго поколения см. в разделе О программе.

Окно настройки часов для приборов второго поколения имеет другой интерфейс и не имеет коррекции времени (приборы аппаратно точнее). Настройка часового пояса требуется для корректного отображения местного времени, так как в приборе значение времени хранится в виде среднего времени по Гринвичу (GMT). Включение параметра **Установить часовой пояс компьютера** синхронизирует часы реального времени прибора с часами ПК.

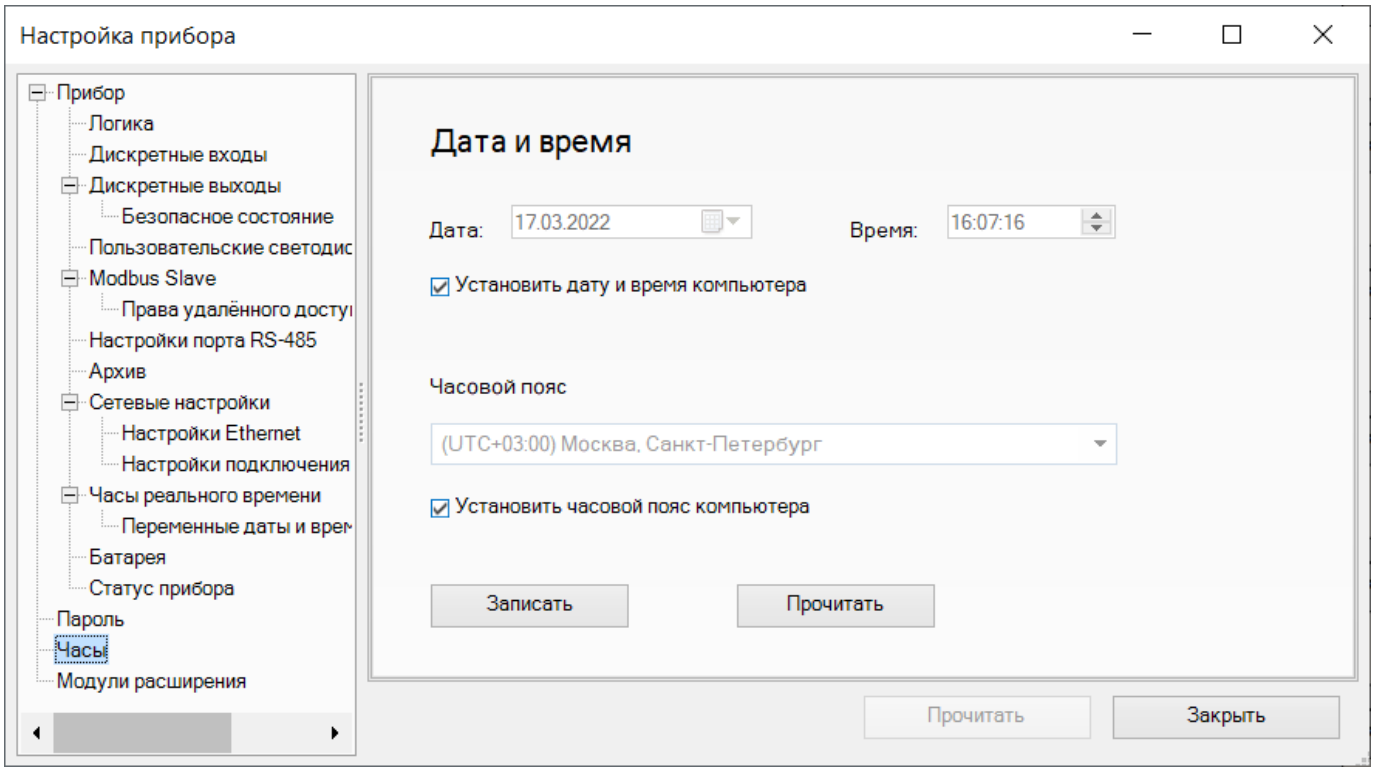


Рисунок 5.7

5.3 Обмен по сети

- Интерфейсы;
- Modbus

5.3.1 Интерфейсы

В разделе **Интерфейсы** настраиваются параметры интерфейсов связи прибора с другими устройствами в сети.

Доступны следующие интерфейсы:

- RS-485;
- Ethernet;
- LoRa.

Для интерфейсов связи RS-485 и Ethernet обмен данными осуществляется по протоколу Modbus.

Обмен данными по интерфейсу LoRa осуществляется по протоколу LoRaWAN.

Добавление, замена и удаление интерфейса доступно для приборов первого поколения. В приборах второго поколения все доступные интерфейсы находятся в дереве настроек.

Добавить интерфейс

Если у прибора доступно добавление интерфейса связи, то следует нажать ПКМ на раздел **Интерфейсы** в дереве настроек и выбрать **Добавить интерфейс**.

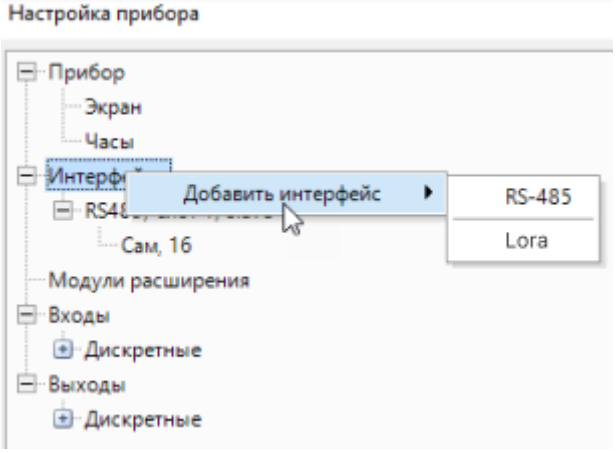


Рисунок 5.8

Интерфейс будет добавлен в дерево с параметрами по умолчанию. Имя интерфейса состоит из номера занимаемого в приборе слота и режима интерфейса.

Интерфейс LoRa в приборах ПР102 и ПР200 может быть расположен только во втором слоте.

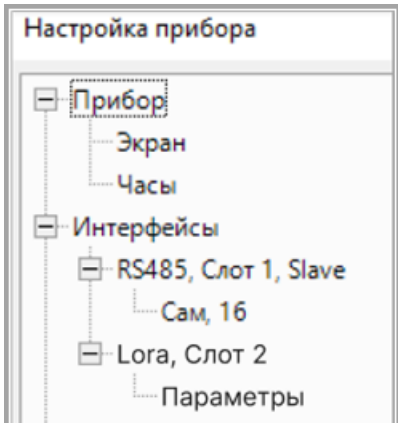


Рисунок 5.9

Количество интерфейсов связи для прибора не может превышать количество слотов в приборе.

Замена/удаление интерфейса

Для замены или удаления интерфейса следует нажать ПКМ на имя интерфейса в дереве настроек и выбрать необходимое действие.

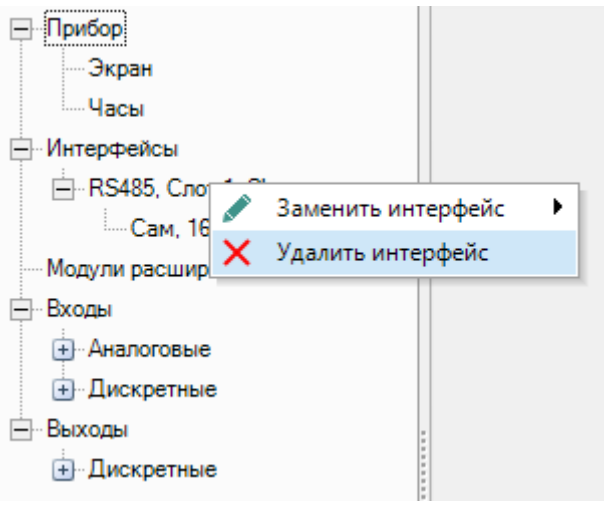


Рисунок 5.10

Если переменные устройств интерфейса задействованы в проекте, то заменить/удалить интерфейс не получится - необходимо сначала удалить привязки переменных.

ПРИМЕЧАНИЕ
Добавление, замена и удаление интерфейсов у ПР100 недоступно.

5.3.1.1 RS-485

Для изменения параметров интерфейса его следует выбрать в дереве настроек.

Параметры интерфейса для приборов первого поколения

В правой части окна настроек прибора отобразятся параметры интерфейса.

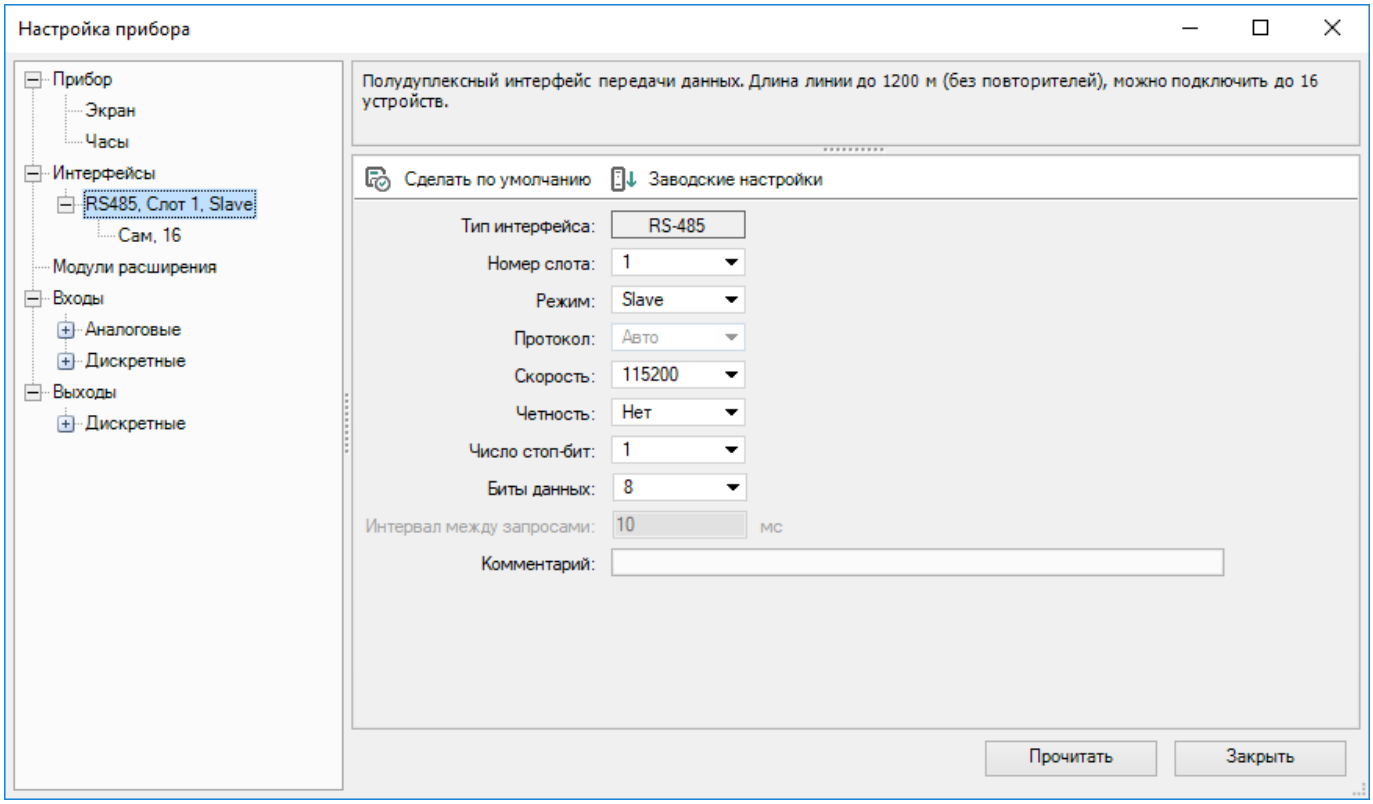


Рисунок 5.11

Номер слота определяет какой слот прибора занимает интерфейс. При наличии в дереве настроек прибора интерфейса LoRa выбор слота для RS–485 недоступен.

Режимы интерфейса:

- Slave – прибор является ведомым, значения которого опрашиваются другим устройством.

- Master – прибор считывает и записывает значения ведомых устройств, подключенных по интерфейсу связи.

Кнопка **Сделать по умолчанию** — сохраняет введенные в поля данные как параметры по умолчанию.

Кнопка **Заводские настройки** — отображает в полях ввода заводские настройки для модели подключенного прибора.

Остальные параметры вводятся в соответствии с задачами проекта и техническими характеристиками других устройств в сети.

ПРИМЕЧАНИЕ
Параметры «Протокол» и «Интервал между запросами» активны только в режиме Master.

Кнопка **Прочитать** считывает текущие параметры подключенного прибора.

После задания параметров интерфейса следует нажать кнопку **Закрыть**. Параметры вступают в силу после перезагрузки прибора по питанию или после загрузки программы в прибор.

Параметры интерфейса для приборов второго поколения

ПРИМЕЧАНИЕ
Список приборов второго поколения см. в разделе О программе.

Для приборов второго поколения параметры интерфейса находятся в разделе **Настройки порта RS-485** в дереве настроек. В правой части окна настроек прибора отображаются параметры интерфейса. Окно настроек имеет вид как на рисунке ниже.

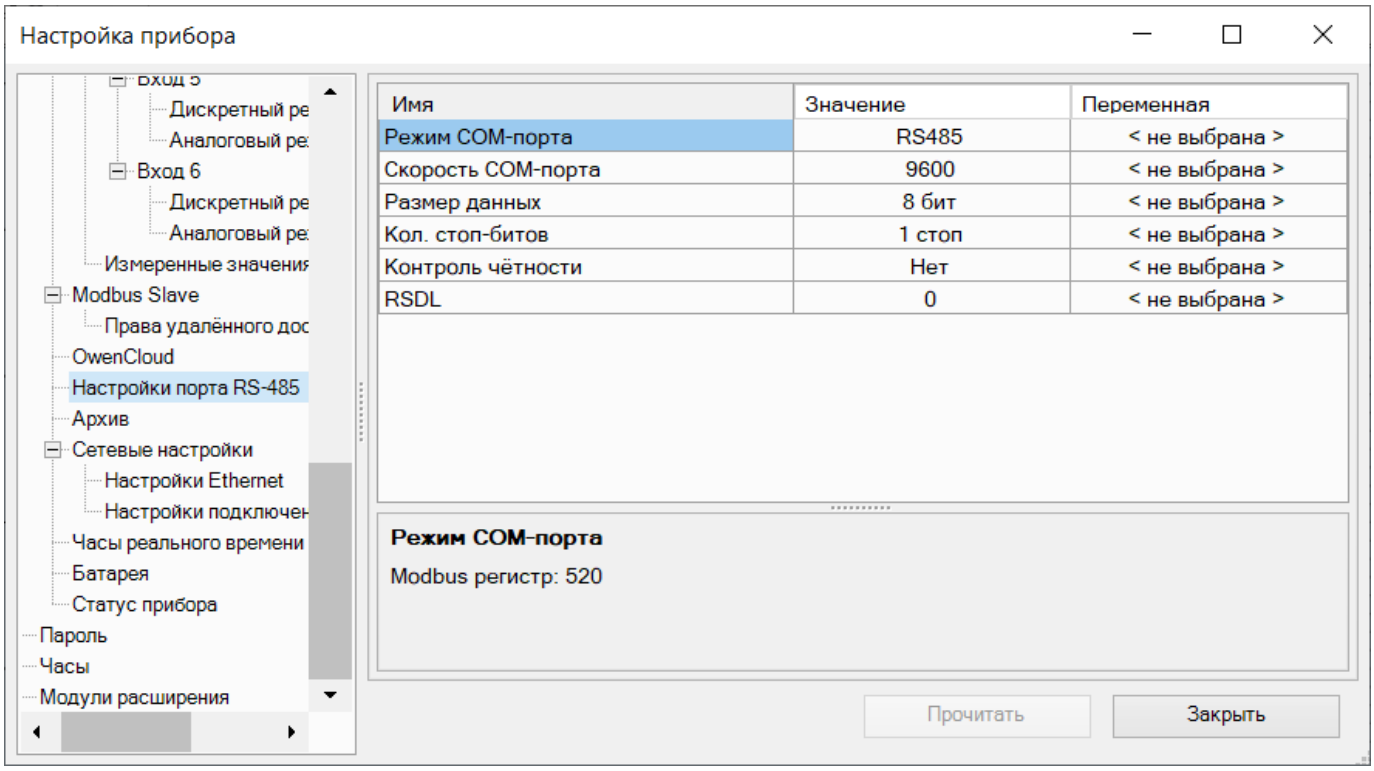


Рисунок 5.12

Список параметров интерфейса зависит от прибора.

Выберите из выпадающего списка или введите значения параметров в столбце **Значение**. В столбце **Переменная** возможна привязка переменной к параметру.

RSDL – задержка ответа прибора по сети RS-485.

5.3.1.2 Ethernet

Настройки Ethernet доступны для приборов второго поколения в меню **Настройки Ethernet** в разделе **Сетевые настройки** дерева настроек.

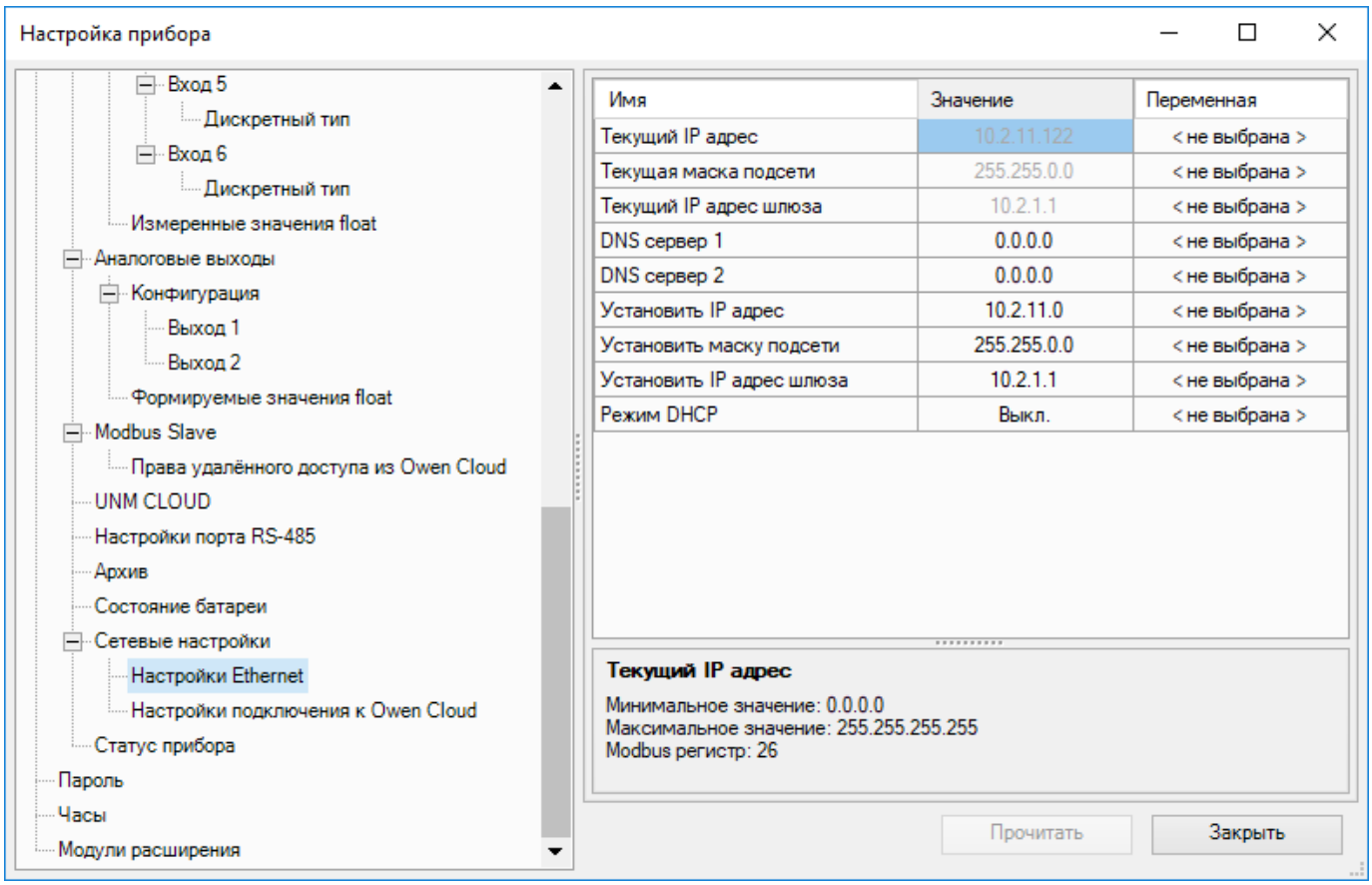


Рисунок 5.13

В окне настроек отображаются текущие сетевые параметра прибора, а также задаются новые.

После записи настроек с новым IP-адресом прибор следует перезагрузить.

ПРИМЕЧАНИЕ
После задания нового IP-адреса прибору потеряется соединение с ПК. Для нового соединения нужно указать новый IP-адрес (см. Настройка порта и подключение прибора).

5.3.1.3 LoRa

Настройки LoRa доступны только для модификаций приборов с модулем LoRa, и располагаются в дереве настроек: раздел **Интерфейсы** → подраздел **Lora**. Обмен данными по интерфейсу LoRa осуществляется по протоколу LoRaWan. Программируемое реле выступает в роли конечного узла сети LoRaWan.

Параметры радиосети

Настройки интерфейса LoRa и параметров подключения сети LoRaWAN задаются на вкладке **Параметры радиосети**.

Тип подключения — выбор типа подключения OTAA/ABP.

- **OTAA** — подключение к сети с настройками, получаемыми от сервера;
- **ABP** — подключение к сети с параметрами настройки, заданными вручную.

Регион — выбор частотного плана, в соответствии с региональными параметрами LoRaWAN.

Начальная скорость — коэффициент расширения спектра, определяет скорость передачи данных, дальность связи и помехоустойчивость. DR5 (SF7, 5.47кБ) - максимальная скорость, минимальная дальность; DR0 (SF12, 0.25кБ) - минимальная скорость, максимальная дальность. Перечень доступных скоростей зависит от выбранного региона.

Задержка на отправку, сек — время задержки (в секундах) перед отправкой пакетов. Позволяет отложить передачу после формирования посылки.

Адаптивный режим — при включении автоматически изменяет параметр **Начальная скорость** на оптимальный, в зависимости от качества связи. Указанная **Начальная скорость** используется как стартовое значение, с которого начинается выбор оптимального режима.

Наличие подтверждения передачи — при включении устройство ожидает подтверждение от сервера о получении пакета. При отсутствии подтверждения выполняет повторную передачу.

Формирование сообщения при пропадании питания — при включении параметра устройство будет отправлять сообщение на сервер при пропадании питания.

OTAA

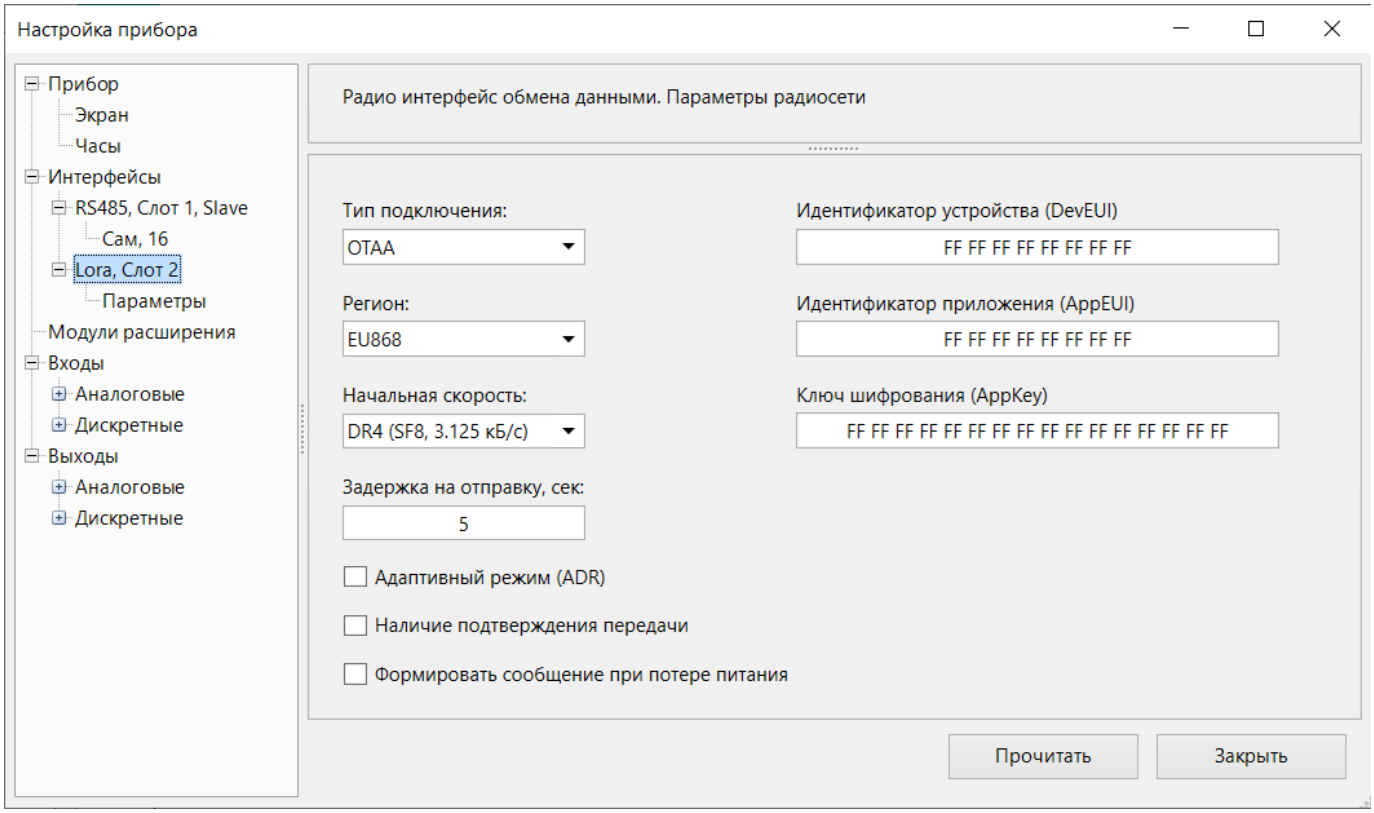


Рисунок 5.14

Идентификатор приложения (AppEUI) — уникальный идентификатор приложения (64 бита). Шестнадцатеричный код, по которому устройство идентифицирует приложение, принимающее и обрабатывающее данные с устройства.

Идентификатор устройства (DevEUI) — уникальный идентификатор конечного устройства (64 бита). Шестнадцатеричный код, по которому сеть идентифицирует устройство при подключении.

Ключ шифрования (AppKey) — уникальный ключ (128 бит). Предназначен для шифрования информации, которая будет передаваться по сети. Должен храниться конфиденциально и совпадать на устройстве и на стороне сервера.

АВР

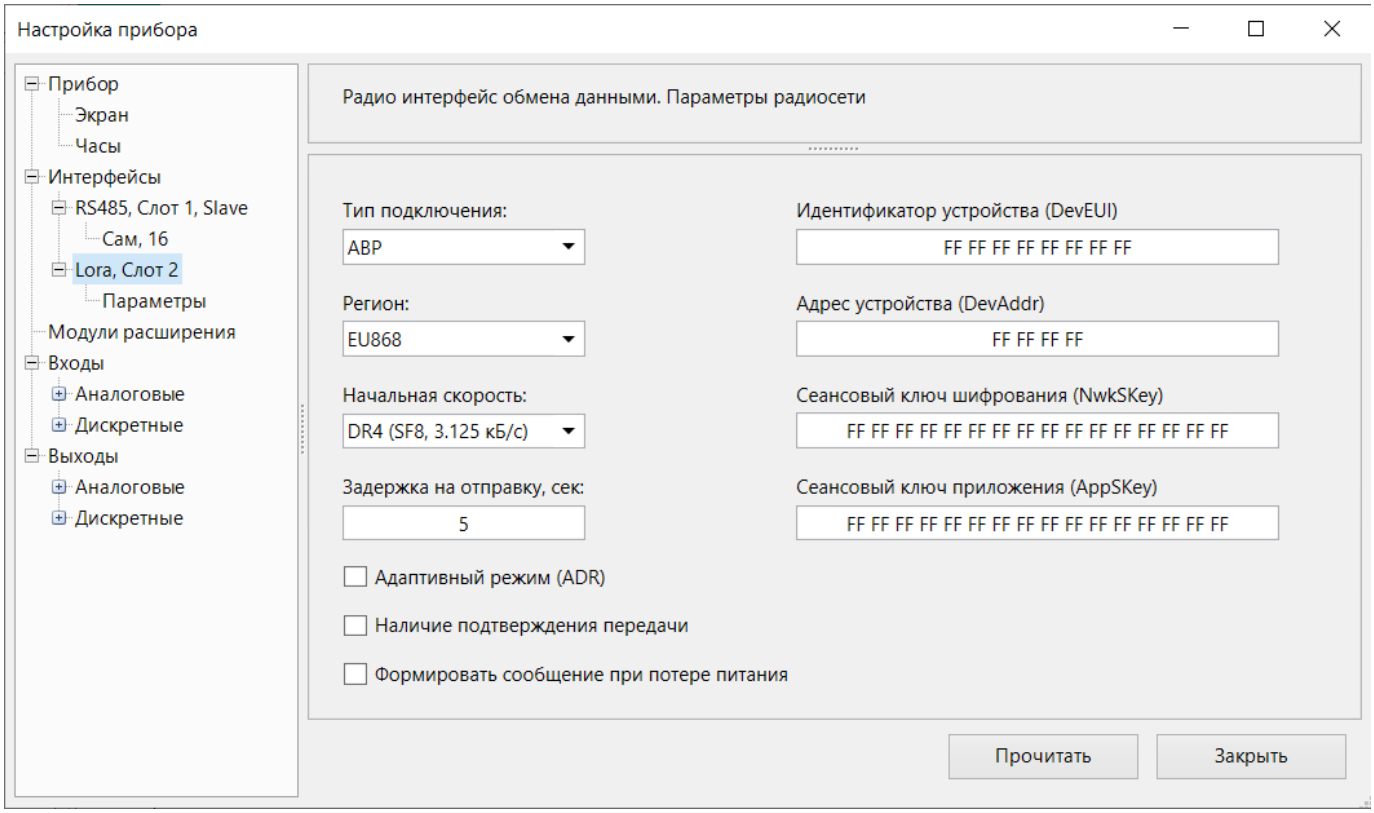


Рисунок 5.15

Идентификатор устройства (DevEUI) — уникальный идентификатор конечного устройства (64 бита). Шестнадцатеричный код, по которому приложение идентифицирует устройство при подключении.

Адрес устройства (DevAddr) — локальный сетевой адрес устройства. Используется для маршрутизации сообщений и идентификации устройства в ходе обмена данными.

Сеансовый ключ шифрования (NwkSKey) — сетевой сеансовый ключ (128 бит). Шестнадцатеричный код, применяемый для шифрования/расшифровки данных между устройством и сетевым сервером.

Сеансовый ключ приложения (AppSKey) — прикладной сеансовый ключ (128 бит). Шестнадцатеричный код, используемый для шифрования/расшифровки данных между устройством и сетевым сервером, в течение конкретной сессии связи.

Кнопка **Прочитать** — считывает все текущие параметры подключенного прибора.

Кнопка **Закрыть** — записывает введенные параметры в прибор и закрывает окно Настройки прибора. Параметры вступают в силу после перезагрузки прибора по питанию или после загрузки программы в прибор.

Сетевые (пользовательские) параметры

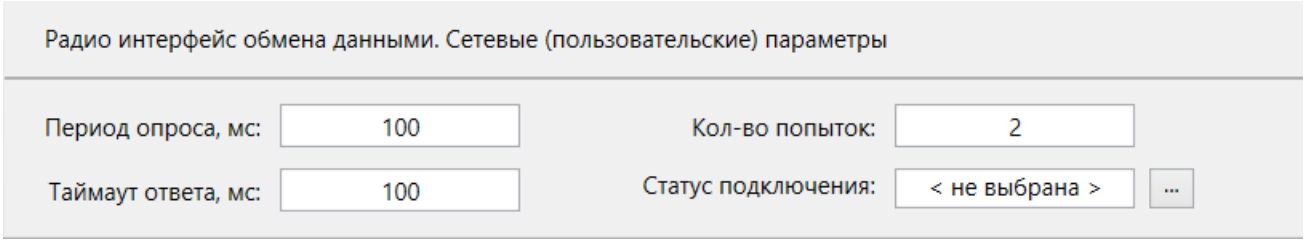


Рисунок 5.16

Период опроса, мс — временной интервал, через который повторяется опрос.

Таймаут ответа, мс — время, по истечении которого попытка опроса считается неудачной.

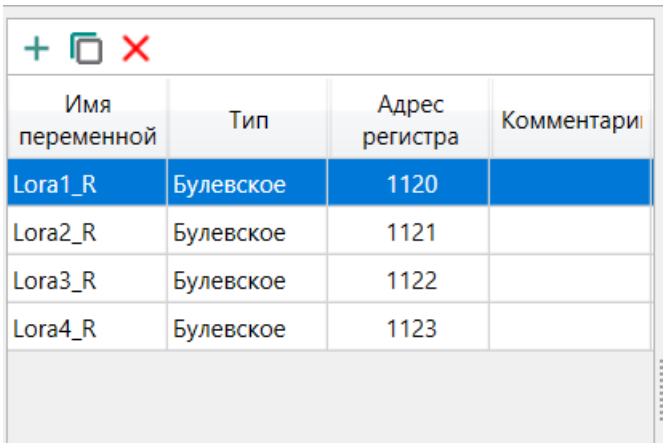
Количество попыток — количество повторов запросов, при отсутствии ответа от опрашиваемого устройства.

Статус подключения — указание булевской переменной, в которую записывается статус подключения прибора: 1 — связь с устройством установлена, 0 — связь с устройством потеряна. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «...»;

Для модуля LoRa групповой опрос не поддержан.

Пользовательские переменные

Чтобы добавить переменную, нажмите на кнопку . Переменная будет добавлена в таблицу, как на рисунке ниже.



Имя переменной	Тип	Адрес регистра	Комментари
Lora1_R	Булевское	1120	
Lora2_R	Булевское	1121	
Lora3_R	Булевское	1122	
Lora4_R	Булевское	1123	

Рисунок 5.17

Переменным, добавленным в интерфейс Lora, к имени автоматически добавляется постфикс:

- _R — если в свойствах переменной выбрана функция чтения;
- _W — если в свойствах переменной выбрана функция записи.

Чтобы создать несколько переменных со схожими настройками используйте тиражирование переменной. Подробнее **Тиражирование переменных** описано в разделе Режим Slave.

Для удаления из таблицы следует выделить необходимую переменную и нажать кнопку

Свойства переменной прибора

Свойства переменной прибора настраиваются справа от таблицы.

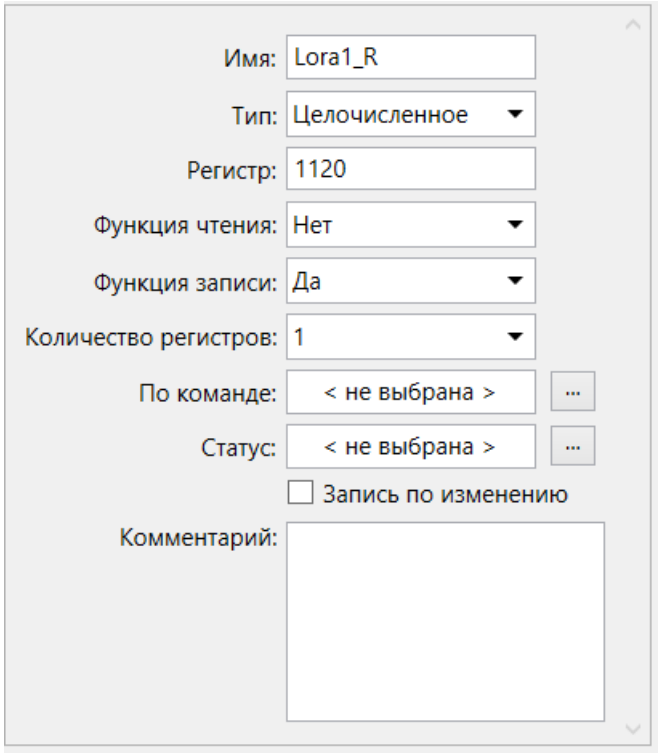


Рисунок 5.18

- **Имя переменной** — имя переменной прибора (задает пользователь);
- **Тип** — тип переменной прибора: целочисленный или с плавающей запятой;
- **Регистр** — адрес регистра переменной прибора (задает пользователь). Диапазон доступных адресов указан в руководстве по эксплуатации прибора;
- **Функция чтения/функция записи** — отключение или выбор функции чтения/записи;
- **Количество регистров** (только для целочисленных переменных) — количество занимаемых переменной регистров: 1 или 2;
- **По команде** — указание булевской переменной, по изменению которой происходит запуск чтения/записи;
- **Статус** — назначается целочисленная переменная, в которую будет записан код ошибки, в случае ее появления. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «...»;
- **Запись по изменению** (отображается только при выборе "Функция записи" = "Да") — если эта функция активирована, то в случае изменения значения переменной мастер инициирует запись значения переменной в Slave-устройство;
- **Комментарий** — текстовое описание переменной для отображения в таблице переменных.

5.3.2 Работа по протоколу Modbus

Общие сведения

Для устройств в режиме Master необходимо выбрать протокол в настройках.

Modbus Master	**ПР200, ПР100, ПР102, ИПП120** **Первое поколение**	**ПР225, ПР205, ПР103** **Второе поколение**
Поддерживаемые протоколы	Modbus RTU, Modbus ASCII (выбирается пользователем)	Modbus RTU, Modbus ASCII (выбирается пользователем) Modbus TCP
Максимальное количество slave-устройств	16 для каждого интерфейса	32 для всех интерфейсов

Modbus Master	**ПР200, ПР100, ПР102, ИПП120** **Первое поколение**	**ПР225, ПР205, ПР103** **Второе поколение**
Максимальное количество запросов	255 переменных для каждого устройства Одна переменная в зависимости от настроек может соответствовать запросу чтения, запросу записи или запросу чтения и записи одновременно	192, начиная с версии Owen Logic 2.7.349 и версий прошивок: ПР205: 1.5.0, ПР103: 1.7.0 64 (в более ранних версиях ПО) Одна переменная в зависимости от настроек может соответствовать либо запросу чтения, либо запросу записи
Поддержка групповых запросов	Поддерживается, начиная с версии Owen Logic 2.11 и версий прошивок: ПР100: 3.08, ПР100 [M02]: 2.70, ПР102: 2.70, ПР200: 2.73, ИПП120: 2.73. Количество регистров в запросе – до 48 (настраиваемо)	Поддерживается, начиная с версии Owen Logic 2.10 и версий прошивок: ПР225: 1.9.0, ПР205: 1.9.0, ПР103: 1.11.0 Количество регистров в запросе – до 16 (настраиваемо)
Поддерживаемые функции Modbus	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0F, 0x10	0x01, 0x02, 0x03, 0x04, 0x06, 0x0F, 0x10
Поддерживаемые типы данных	Bool, Uint16, Uint32, Float	Bool, Uint16, Uint32, Float
Поддержка изменения порядка байт/регистров	Поддерживается, влияет только на переменные типа Float и Uint32	Поддерживается, влияет только на переменные типа Float и Uint32
Поддержка широковещательного запроса (broadcast)	Отсутствует	Отсутствует
Поддержка шаблонов опроса	Поддерживается	Поддерживается начиная с версии Owen Logic 2.10
Поддержка управления опросом	На уровне устройства (опрашивать/не опрашивать) – см. параметр **Опрос** На уровне запроса – отправка по фронту булевской переменной (параметры **Запуск чтения**/**Запуск записи**) + галочка **Запись по изменению**	На уровне устройства (опрашивать/не опрашивать) – см. параметр **Опрос** На уровне запроса – отправка по фронту булевской переменной (параметры **Запуск чтения**/**Запуск записи**) + галочка **Запись по изменению**
Управление желаемым периодом опроса	На уровне устройства	На уровне запросов
Поддержка диагностики	На уровне устройства (есть связь/нет связи) – см. параметр **Статус** На уровне запроса – код ошибки Modbus -см. параметр **Статус**	На уровне устройства (есть связь/нет связи) – см. параметр **Статус**
Поддержка Modbus TCP	Отсутствует	Поддерживается Опрос всех slave-устройств происходит последовательно (активно только одно клиентское соединение) В рамках соединения запросы отправляются последовательно допустимый диапазон для адреса slave-устройства (Unit ID): 1…247

Устройства в режиме Slave определяют протокол автоматически.

Modbus Slave	**ПР200, ПР100, ПР102, ИПП120** **Первое поколение**	**ПР225, ПР205, ПР103** **Второе поколение**
Поддерживаемые протоколы	Modbus RTU, Modbus ASCII (автоопределение – ответ отправляется в формате запроса)	Modbus RTU, Modbus ASCII (автоопределение – ответ отправляется в формате запроса) Modbus TCP
Модель области памяти	Общая, с наложением input/holding регистров	Общая, с наложением input/holding регистров
Объём памяти	64 регистра (128 байт)	1020 регистров (2040 байт)
Поддержка групповых запросов	поддерживается без искусственных ограничений	
Поддерживаемые функции Modbus	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0F, 0x10	0x03, 0x04, 0x06, 0x0F, 0x10, 0x14, 0x15
Работа с битовыми масками	Параметры битовой маски могут читаться как функцией 0x03, так и 0x01 – в этом случае номер регистра нужно умножить на 16 и прибавить номер бита	Параметры битовой маски можно прочитать только функцией 0x03
Поддерживаемые типы данных	Uint16, Float	Uint16, Float
Поддержка изменения порядка байт/регистров	Поддерживается, но влияет только на системные переменные типа Float и Uint32	Не поддерживается. При опросе переменных, занимающими два и более регистра следует использовать следующие настройки: • порядок байт – старшим байтом вперед • порядок регистров – младшим регистром вперед
Поддержка широковещательного запроса (broadcast)	Поддерживается	Не поддерживается
Поддержка Modbus TCP	Не поддерживается	Поддерживается, до 4 одновременных клиентских подключений (пятое – независимое и зарезервировано под Owen Cloud) Unit ID в запросе не валидируется
Поддержка RETAIN	Все сетевые переменные всегда являются энергонезависимыми	Начиная с версии Owen Logic 2.9 – энергонезависимость каждой сетевой переменной настраивается пользователем В более ранних версиях: все сетевые переменные всегда являются энергонезависимыми
Поддержка экспорта карты регистров	Поддерживается экспорт в Owen OPC Server, OwenCloud и *.csv*-файл	Поддерживается экспорт в Owen OPC Server, OwenCloud и *.csv*-файл

Для организации обмена данными в сети через интерфейс связи необходим Мастер сети – устройство в режиме Master. Основная функция Мастера сети – инициировать обмен данными.

На линии связи допускается наличие только одного устройства в режиме Master.

Количество интерфейсов связи

Количество интерфейсов связи прибора зависит от количество слотов подключения. Если прибор поддерживает работу до двух интерфейсов связи, то их можно настроить на работу в двух независимых сетях.

Стандартные коды ошибок Modbus

- **01** — принятый код функции не может быть обработан;
- **02** — адрес данных, указанный в запросе, недоступен;
- **03** — значение в поле данных запроса, является недопустимой величиной.

Функции и области данных

При запросе Master обращается к одной из **областей памяти** Slave с помощью **функции**. **Область памяти** характеризуется типом хранящихся в ней значений (биты/регистры) и типом доступа (только чтение/чтение и запись).

Таблица 5.1 – Области данных протокола Modbus

Область данных	Обозначение	Тип данных	Тип доступа
Coils (Регистры флагов)	0x	Булевый	Чтение/запись
Discrete Inputs (Дискретные входы)	1x	Булевый	Только чтение
Input Registers (Регистры ввода)	3x	Целочисленный	Только чтение
Holding Registers (Регистры хранения)	4x	Целочисленный	Чтение/запись

Каждая область памяти состоит из определенного (зависящего от конкретного устройства) количества ячеек. Каждая ячейка имеет уникальный адрес. Для конфигурируемых устройств производитель предоставляет **карту регистров**, в которой содержится информация о соответствии параметров устройства и их адресов. Для программируемых устройств пользователь формирует такую карту самостоятельно с помощью среды программирования. Существуют устройства, в которых сочетаются оба рассмотренных случая – у их карты регистров есть фиксированная часть и часть, которую пользователь может дополнить в соответствии со своей задачей .

В некоторых устройствах области памяти наложены друг на друга (например, **0x** и **4x**) – т. е. пользователь сможет обращаться разными функциями к одним и тем же регистрам.

Функция определяет операцию (чтение/запись) и область памяти, с которой эта операция будет произведена.

Таблица 5.2 – Основные функции протокола Modbus

Код функции	Имя функции	Выполняемая команда
1 (0x01)	Read Coil Status	Чтение значений из нескольких регистров флагов
2 (0x02)	Read Discrete Inputs	Чтение значений из нескольких дискретных входов
3 (0x03)	Read Holding Registers	Чтение значений из нескольких регистров хранения
4 (0x04)	Read Input Registers	Чтение значений из нескольких регистров ввода
5 (0x05)	Force Single Coil	Запись значения в один регистр флага
6 (0x06)	Preset Single Register	Запись значения в один регистр хранения
15 (0x0F)	Force Multiple Coils	Запись значений в несколько регистров флагов
16 (0x10)	Preset Multiple Registers	Запись значений в несколько регистров хранения

В различных документах идентичные обозначения могут иметь разный смысл в зависимости от контекста. Например, префикс **0x** часто используют как указание на шестнадцатеричную систему счисления, поэтому в одном случае **0x30** может обозначать «30-й бит области памяти **coils**», а в другом – «адрес 30 в шестнадцатеричной (HEX) системе счисления» (при этом данный адрес может относиться к любой области памяти).

Опрос Slave может быть **одиночным** или **групповым**. При **одиночном опросе** Master считывает каждый из параметров Slave отдельной командой.

При **групповом опросе** Master считывает одной командой сразу несколько параметров, чьи адреса в карте регистров расположены строго последовательно и не имеют разрывов. Групповой опрос позволяет уменьшить трафик в сети и время, затрачиваемое на опрос устройства, но в некоторых случаях его применение невозможно (или возможно с ограничениями) из-за индивидуальных особенностей прибора.

Время опроса и тайм-аут ответа

Прибор формирует запросы с периодичностью, заданной в параметрах опрашиваемого устройства **Период опроса**. Время ожидания ответа на запрос задается в параметре **Таймаут ответа**. Если за это время ответ не поступает, то прибор повторит запрос N раз, где N – параметр **Кол-во попыток**. При отсутствии ответа переменная **Статус** примет значение 0 (False).

Типы сетевых переменных

Тип данных		Размер в режиме Slave	Размер в режиме Master
Целочисленный	UINT	16 бит (1 регистр)	16 бит (1 регистр)
	UDINT	-	32 бита (2 регистра)
С плавающей запятой	REAL	32 бита (2 регистра)	32 бита (2 регистра)
Булевский	BOOL	-	1 бит

Порядок регистров и порядок байт

При работе прибора в режиме Slave настройки порядка следования регистров/байт влияют только на считывание Мастером системных сетевых переменных с плавающей запятой (аналоговые входы/выходы).

Для пользовательских сетевых переменных в режиме Slave данные всегда передаются в формате:

- Старшим регистром вперед— нет;
- Старшим байтом вперед— да.

Автоподстройка времени цикла программы

Приборы могут подстраивать время работы цикла программы в зависимости от сложности алгоритма. Автоподстройка времени цикла программы влияет на работу интерфейса связи, так как запросы обрабатываются в оставшееся после выполнения программы время цикла.

Согласно алгоритму подстройки времени цикла, минимальное число вызовов Master составляет до 50 раз в секунду. Если Master не успевает опросить все устройства, то необходимо внести изменения в алгоритм для оптимизации количества запросов.

Период опроса и правила его назначения

Прибор все запросы ставит в очередь. Если очередь короткая, то прибор выполнит все циклы запроса-ответа и остановится в ожидании пока не подойдет к концу заданный период. Если очередь длинная и не укладывается в заданный период, то прибор будет опрашивать все необходимые параметры с максимально возможным периодом, но этот период будет больше заданного в настройках.

Временная диаграмма опроса:

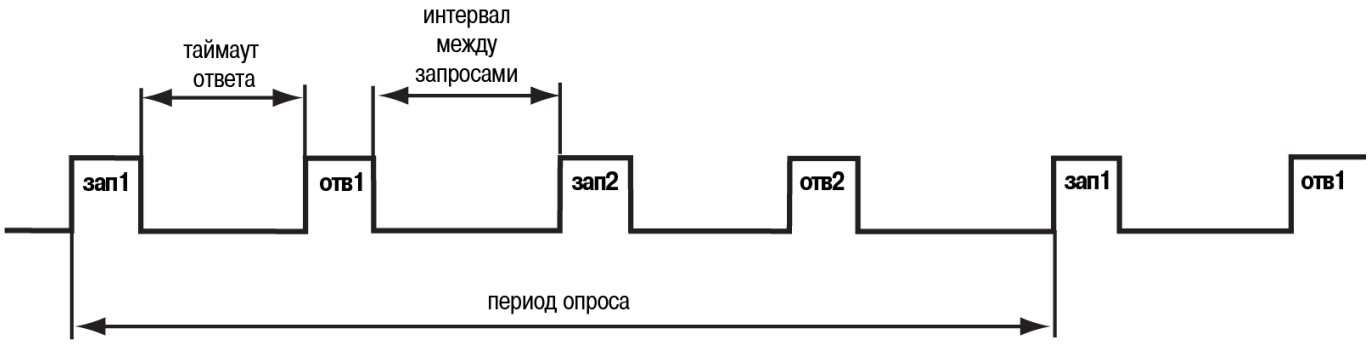


Рисунок 5.19

Для максимальной скорости 115200 бит/с опрос двух переменных типа FLOAT в «идеальных» условиях (короткая линия связи, отсутствие помех) у одного подчиненного устройства следует установить:

- время ответа от начала запроса – 2,64 мс;
- следующий запрос начинается не раньше, чем через 12 мс;
- общее время на запрос 1 регистра FLOAT – 3,4 мс.

В вышеперечисленных условиях, прибор сможет отправлять 83 запроса в секунду. Данное значение справедливо и для других приборов с подобными временными характеристиками. В процессе разработки алгоритма, когда логика усложняется, то увеличивается время цикла и количество запросов за секунду будет снижаться.

Значение периода опроса зависит от алгоритма, как часто и какие параметры надо опрашивать. Рекомендуется выставлять период опроса равным 1 с. В этом случае прибор сможет опросить до 50 переменных.

Порядок опроса нескольких устройств на шине

Устройства опрашиваются согласно сформированной очереди. Опрос происходит от наименьшего адреса к наибольшему. В примере на иллюстрации ниже первым будет опрошено устройство с адресом 8, последним – с адресом 32.

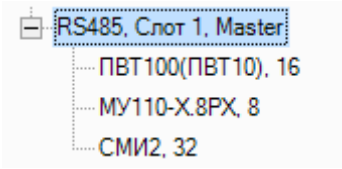


Рисунок 5.20

Можно задавать разный период опроса подчиненных устройств. Конкретное значение будет зависеть от решаемой задачи.

Расчет адреса и бита для считывания булевой переменной в режиме Master

В некоторых случаях требуется вычислить адрес регистра и номер бита подчиненного устройства. Для определения адреса регистра и номера бита из документации на опрашиваемый прибор берется расположение битовой переменной, например, *бит 1400*, далее следует:

1. Вычислить адрес регистра: 1400 ÷ 16 = 87.
2. Вычислить номер бита 1400 mod 16 = 8.

Полученные адрес и номер вводятся в поля «Регистр» и «Бит» во время настройки опроса устройства.

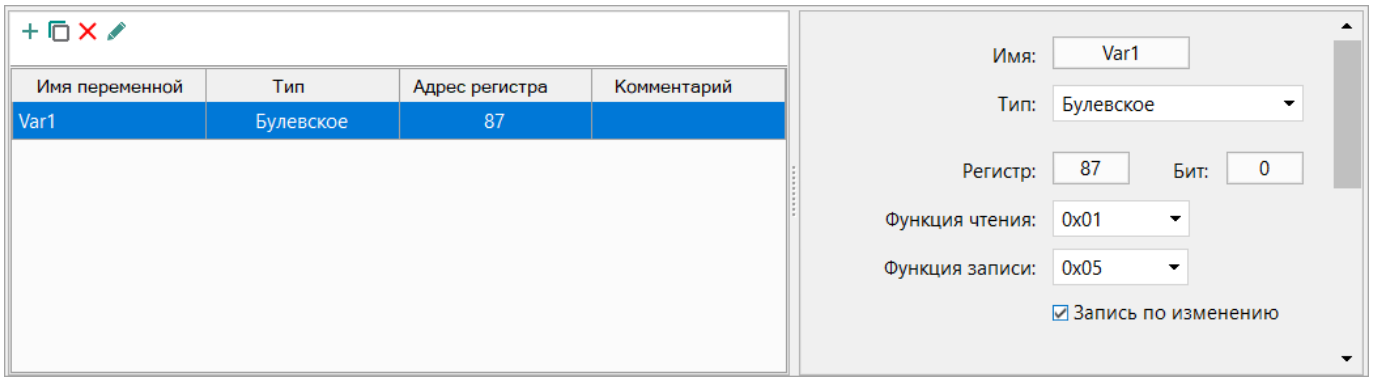


Рисунок 5.21

5.3.2.1 Режим Master

В режиме Master прибор опрашивает подключенные по интерфейсу другие подчиненные устройства. Для опроса однотипных параметров подключаемых Slave-устройств доступен **групповой опрос**.

Для опроса следует добавить и настроить устройства. Чтобы добавить устройство нажмите ПКМ на имя интерфейса Master в окне настройки прибора - для приборов первого поколения, или на пункт Мастер Modbus для приборов второго поколения, и выберите **Добавить устройство** в контекстном меню.

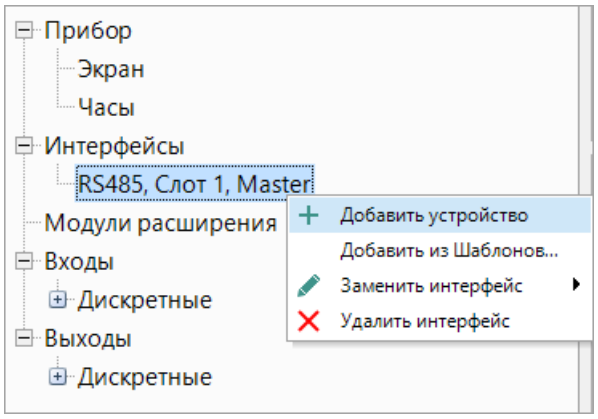


Рисунок 5.22 – Добавить устройство в интерфейс прибора первого поколения

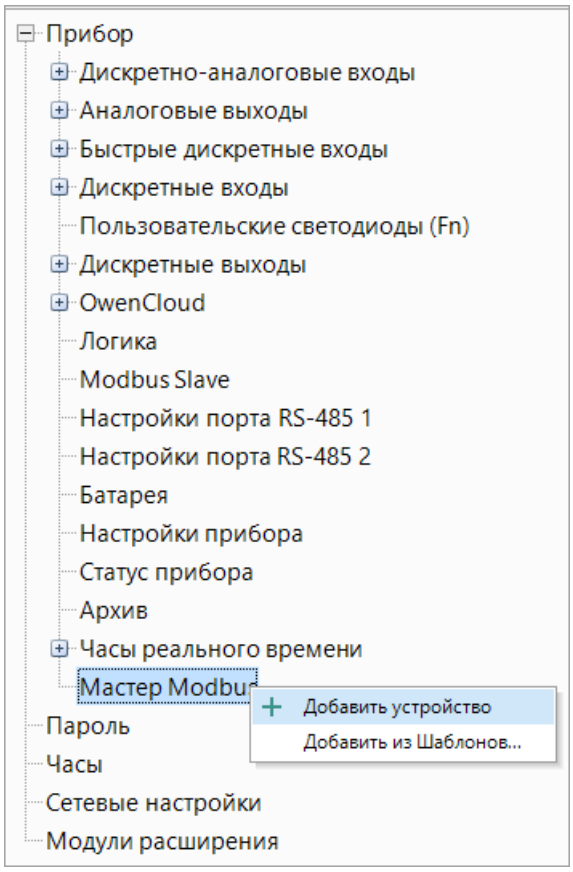


Рисунок 5.23 – Добавить устройство в интерфейс прибора второго поколения

Добавленное устройство будет отображено в дереве в виде новой ветки для соответствующего интерфейса. Наименование устройства состоит из имени, адреса устройства и интерфейса связи (для приборов второго поколения).

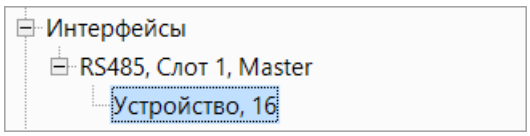


Рисунок 5.24 – Добавленное устройство в интерфейс прибора первого поколения

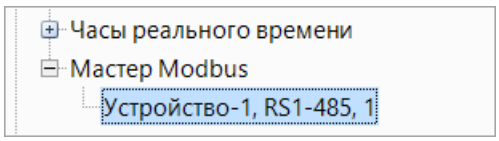


Рисунок 5.25 – Добавленное устройство в интерфейс прибора второго поколения

Групповой опрос

Групповой опрос позволяет считать сразу несколько регистров одной командой. Максимальное количество регистров в групповом опросе зависит от объема буфера устройства, типа протокола Modbus и функции чтения/записи. Для устройств первого поколения максимальный размер группового опроса составляет 48 регистров, для устройств второго поколения - 16 регистров. Формирование группового опроса подчиняется следующим правилам:

1. Переменные должны располагаться в карте регистров строго последовательно — без пропусков и незадействованных регистров/бит между ними.
2. У переменных в опросе должны быть указаны одинаковы коды функций чтения/записи.

   Исключение: функции **0x05** и **0x06** не допускаются - групповые запросы для них не поддерживаются.
3. Переменные объединяются в групповой опрос при одинаковом типе опроса по периоду, по команде или по изменению.
4. В групповой опрос объединяются только переменные совместимого размера:
   - **Булевские переменные** — только с булевскими.
   - **Целочисленные (1 регистр)** — только с целочисленными (1 регистр).

- **Целочисленные (2 регистра)** — с целочисленными (2 регистра) и переменными с плавающей запятой.
- **С плавающей запятой** — с целочисленными (2 регистра) и с переменными с плавающей запятой.

Для минимизации времени получения данных со всех устройств рекомендуется:

- если одно или несколько Slave-устройств не подключены или недоступны, то рекомендуется предусмотреть в программе блокировку опроса этих устройств или сократить до минимума параметр «Таймаут ответа» для этих устройств. В противном случае время опроса будет расти пропорционально количеству неподключенных устройств и величине установленного параметра «Таймаут ответа» для этих устройств;
- во время установки параметра «Интервал между запросами» следует учитывать количество Slave-устройств и общее количество запросов. Если время обработки всех запросов занимает больше времени, чем установлено параметром «Интервал между запросами», то данный параметр будет игнорироваться.

Допускается использование одинаковых адресов и имен переменных для каждого устройства.

Режим Master для приборов первого поколения

ПРИМЕЧАНИЕ
Список приборов первого и второго поколения см. в разделе О программе.

Параметры опрашиваемого устройства

Для изменения параметров устройства следует нажать на его наименование в дереве настроек – в правой части окна настроек прибора отобразятся параметры.

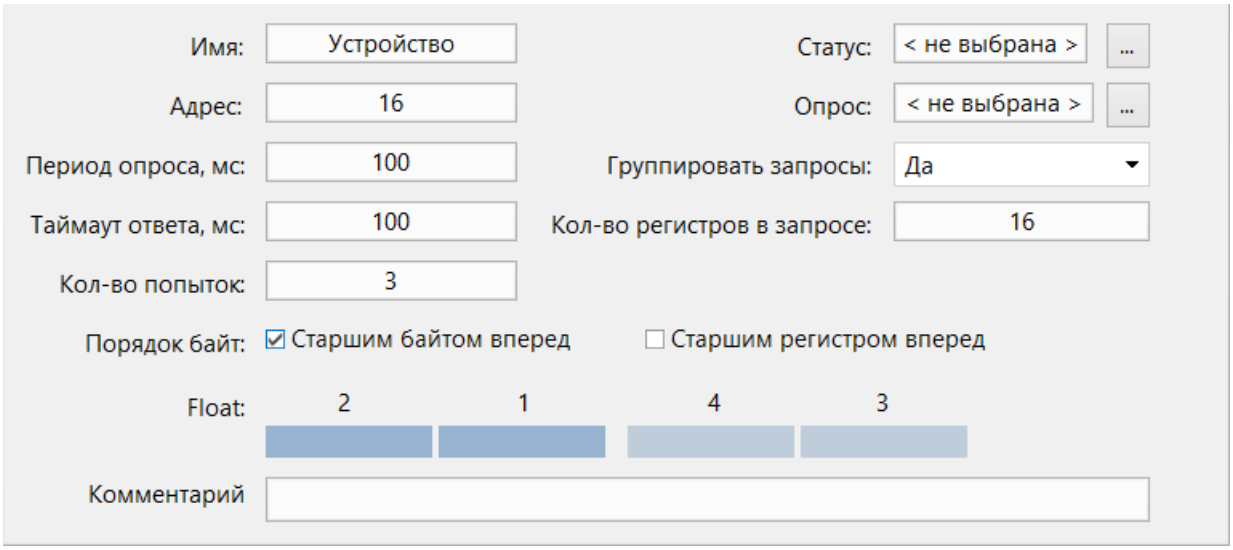


Рисунок 5.26

- **Имя** — имя устройства для отображения в дереве настроек;
- **Адрес** — сетевой адрес устройства;
- **Период опроса, мс** — временной интервал, через который повторяется опрос. Допустимый диапазон от 0 до 65535 мс;
- **Таймаут ответа, мс** — время, по истечении которого попытка опроса считается неудачной. Допустимый диапазон от 0 до 65535 мс. Таймаут ответа рекомендуется задавать в диапазоне 300–500 мс;
- **Кол-во попыток** — количество неудачных попыток опроса, при достижении которого изменяется Статус устройства. Допустимый диапазон от 0 до 255;
- **Статус** — указание булевской переменной, в которую записывается статус прибора: 1 — устройство функционирует нормально, 0 — связь с устройством потеряна. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «**...**»;
- **Опрос** — указание булевской переменной, которая определяет доступ к опросу: 0 — запрет опроса, 1 — разрешение опроса. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «**...**»;

  ПРИМЕЧАНИЕ
  Привязанные к параметрам **Статус** и **Опрос** переменные можно использовать в логике работы программы.

- **Группировать запросы** — опрос более одной переменной;

- **Количество регистров в запросе** — от 2 до 48;
- **Порядок байт** — определяет очередность следования байтов в пакете;
- **Старшим регистром вперед** — определяет очередность посылки регистров во время работы с двухрегистровыми переменными;
- **Старшим байтом вперед** — определяет очередность следования байтов в посылке;
- **Комментарий** — текстовое описание устройства.

Пример

Требуется считать число «–250.123».

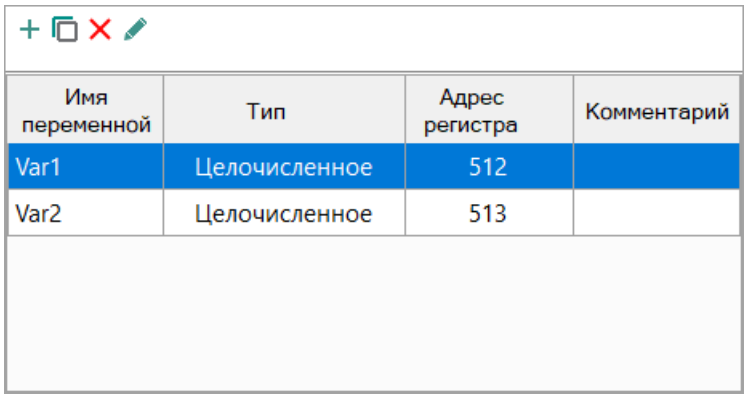
В зависимости от настройки порядка следования регистров/байт оно будет передаваться различным образом.

Настройка порядка следования регистров/байт	Передаваемое число
Порядок байт: ☐ Старшим байтом вперед ☐ Старшим регистром вперед Float: 1 2 3 4 **Рисунок 5.27**	Младший байт Старший байт Младший байт Старший байт 01111101 00011111 01111010 11000011 Младший регистр Старший регистр **Рисунок 5.28**
Порядок байт: ☑ Старшим байтом вперед ☐ Старшим регистром вперед Float: 2 1 4 3 **Рисунок 5.29**	Старший байт Младший байт Старший байт Младший байт 00011111 01111101 11000011 01111010 Младший регистр Старший регистр **Рисунок 5.30**
Порядок байт: ☐ Старшим байтом вперед ☑ Старшим регистром вперед Float: 3 4 1 2 **Рисунок 5.31**	Младший байт Старший байт Младший байт Старший байт 01111010 11000011 01111101 00011111 Старший регистр Младший регистр **Рисунок 5.32**
Порядок байт: ☑ Старшим байтом вперед ☑ Старшим регистром вперед Float: 4 3 2 1 **Рисунок 5.33**	Старший байт Младший байт Старший байт Младший байт 11000011 01111010 00011111 01111101 Старший регистр Младший регистр **Рисунок 5.34**

Переменные опрашиваемого устройства

В нижней части окна располагается таблица переменных опрашиваемого устройства.

Чтобы добавить переменную, следует нажать кнопку . Переменная будет помещена в таблицу, как на рисунке ниже.



Имя переменной	Тип	Адрес регистра	Комментарий
Var1	Целочисленное	512	
Var2	Целочисленное	513	

Рисунок 5.35

Для удаления из таблицы следует выделить необходимую переменную и нажать кнопку ✕.

Чтобы создать несколько переменных со схожими настройками, следует выбрать переменную в таблице и нажать кнопку **Тиражировать**.

ПРИМЕЧАНИЕ
Подробнее **Тиражирование переменных** описано в разделе Режим Slave.

Свойства сетевых переменных опрашиваемого устройства

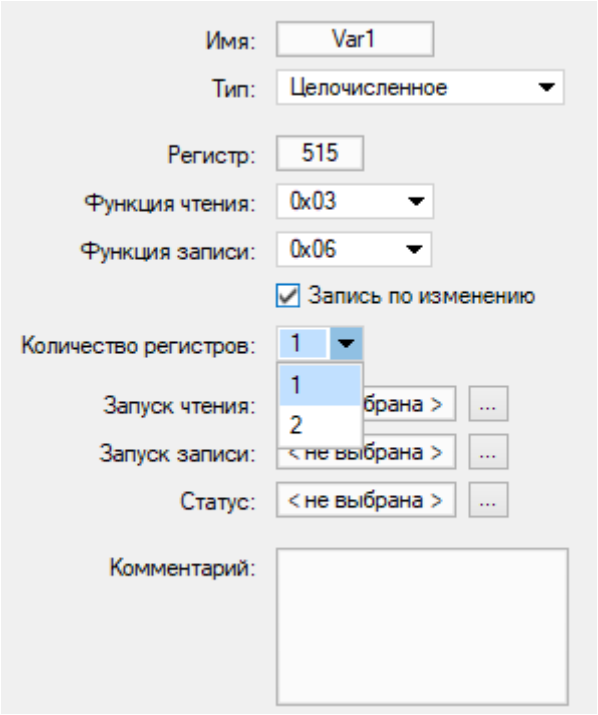


Рисунок 5.36

Свойства переменной опрашиваемого устройства настраиваются справа от таблицы:

- **Имя переменной** — наименование для отображения в таблице переменных;

  ПРИМЕЧАНИЕ
  Создание переменных с одинаковыми именами не рекомендуется.

- **Тип** — тип переменной: булевский, целочисленный или с плавающей запятой;
- **Бит (только для переменных булевского типа)** — номер бита для чтения;
- **Регистр/Адрес регистра** — значение регистра, к которому обращается прибор, отображается в таблице;
- **Функция чтения/функция записи** — отключение или выбор функции записи/чтения;

  ПРИМЕЧАНИЕ
  Запись по изменению — если эта функция активирована, то в случае изменения значения переменной мастер инициирует запись значения переменной в Slave-устройство.

- **Количество (только для целочисленных переменных)** — количество занимаемых переменной регистров: 1 или 2;
- **Запуск чтения** — назначается переменная булевского типа (командная переменная) для принудительного чтения настраиваемой переменной: если значение командной переменной меняется с 0 на 1, мастер инициирует запрос на чтение, после успешного выполнения запроса значение командной переменной автоматически меняется с 1 на 0;
- **Запуск записи** — назначается командная переменная булевского типа (командная переменная) для принудительной записи настраиваемой переменной: если значение командной переменной меняется с 0 на 1, мастер инициирует запрос на запись, после успешного выполнения запроса значение командной переменной автоматически меняется с 1 на 0;
- **Статус** — назначается целочисленная переменная, в которую будет записан код ошибки Modbus, в случае ее появления;
- **Комментарий** — текстовое описание переменной для отображения в таблице переменных;

ПРИМЕЧАНИЕ
Чтение и запись по команде позволяет принудительно читать или записывать значения переменной и более гибко настраивать чтение и запись.
Возможно комбинировать различные варианты чтения и записи:

- чтение по таймеру;
- чтение/запись по событию;
- запись по изменению (используется по умолчанию).

Режим Master для приборов второго поколения

Для настройки параметров опроса подключенных устройств следует выбрать в дереве параметров прибора узел **Modbus Master**.

ПРИМЕЧАНИЕ
Максимальное количество устройств на всех интерфейсах — 32.

Для изменения параметров опрашиваемого устройства следует нажать на его название в дереве параметров. В правой части окна отобразятся доступные параметры: в верхней части — параметры устройства, в нижней части — сетевые переменные устройства.

ПРИМЕЧАНИЕ
Доступно изменение сетевых настроек опрашиваемых устройств из внешнего ПО (например, OwenConfigurator).

Параметры опрашиваемого устройства

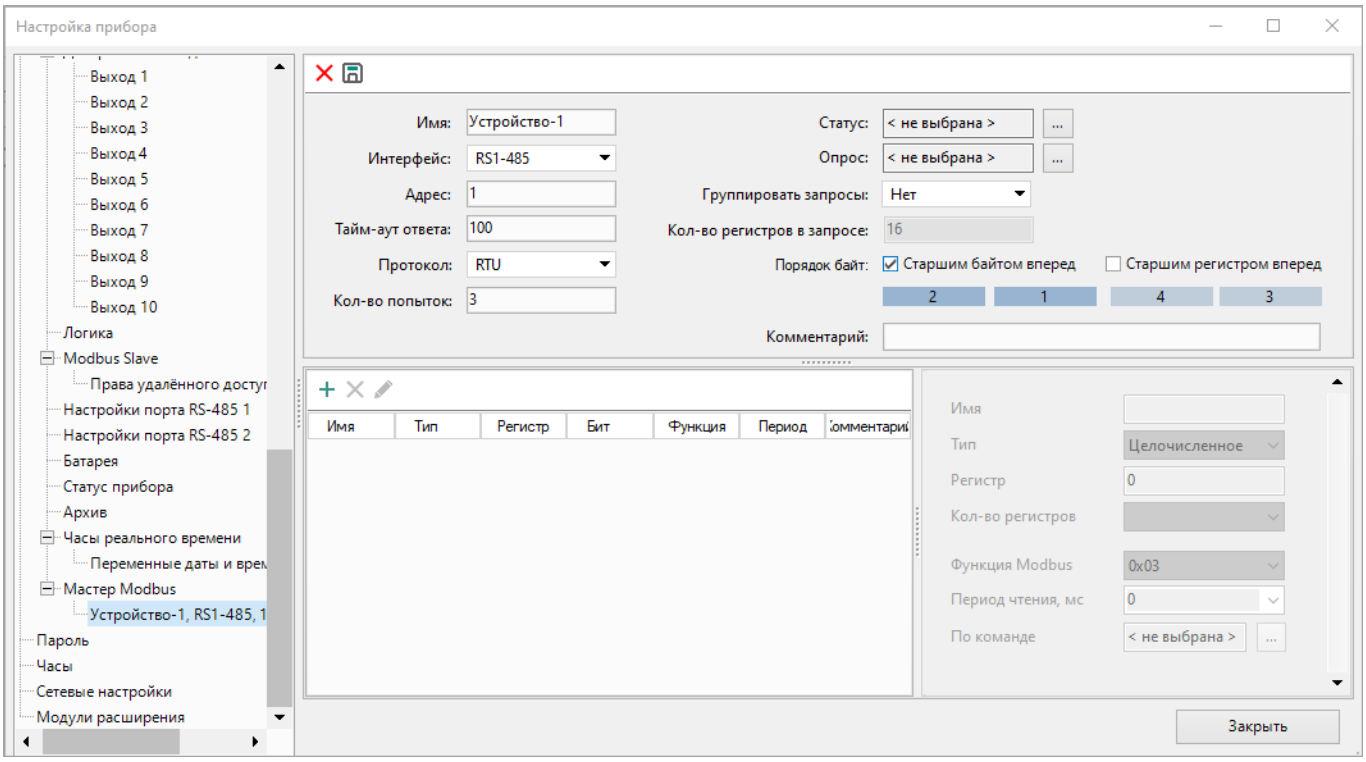


Рисунок 5.37

- **Имя** — имя устройства для отображения в дереве настроек;
- **Интерфейс** — интерфейс, по которому подключено опрашиваемое устройство. От выбранного интерфейса зависит список доступных параметров.
- **Адрес** — сетевой адрес устройства;
- **Таймаут ответа, мс** — время, по истечении которого попытка опроса считается неудачной. Допустимый диапазон от 10 до 10 000 мс;
- **Протокол**
- **Количество попыток** — количество неудачных попыток опроса. Допустимый диапазон от 0 до 3;
- **Статус** — указание булевской переменной, в которую записывается статус прибора: 1 — устройство функционирует нормально, 0 — связь с устройством потеряна. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «**...**»;
- **Опрос** — указание булевской переменной, которая определяет доступ к опросу: 0 — запрет опроса, 1 — разрешение опроса. Переменная выбирается в таблице переменных, которая вызывается нажатием кнопки «**...**»;
- **Группировать запросы** — опрос более одной переменной;
- **Количество регистров в запросе** — от 2 до 16;

- **Порядок байт** — определяет очередность следования байтов в пакете;
- **Комментарий** — текстовое описание устройства.

Специфические параметры опрашиваемого устройства, подключенного по интерфейсу Ethernet:

- **IP адрес** — уникальный сетевой адрес устройства, допустимый диапазон от 0.0.0.0 до 255.255.255.255.;
- **Порт** — номер порта, допустимый диапазон от 0 до 65535.

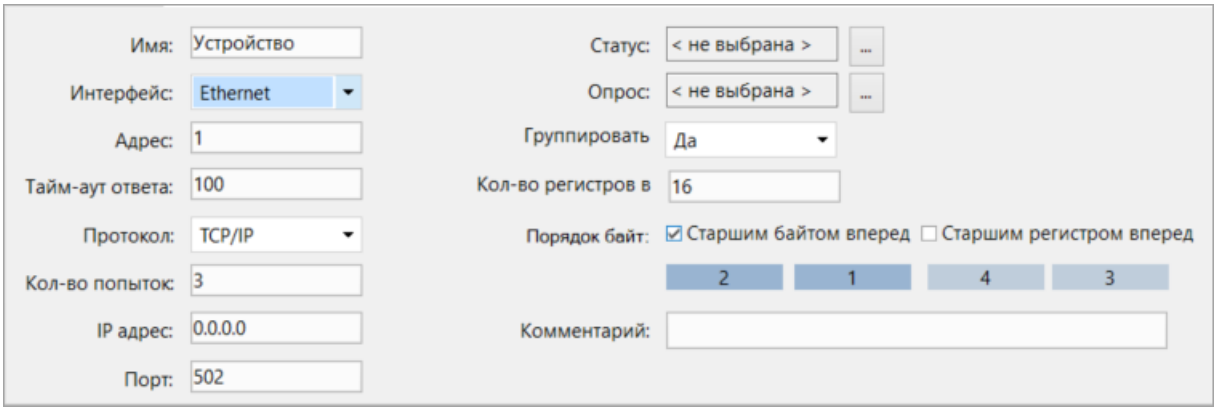


Рисунок 5.38

Свойства сетевых переменных опрашиваемого устройства

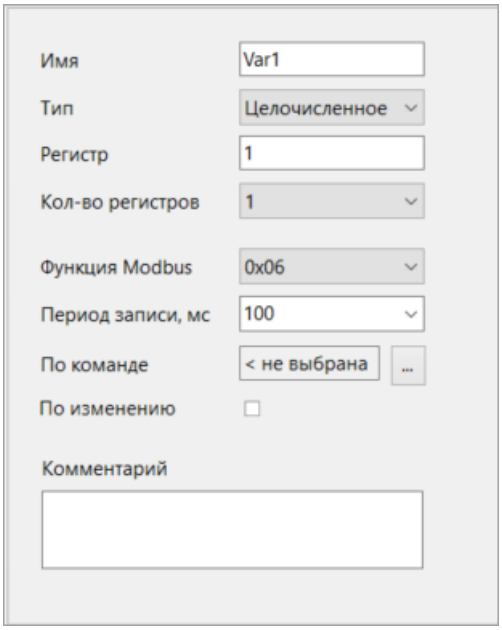


Рисунок 5.39

- **Имя** — наименование для отображения в таблице переменных ;

  ПРИМЕЧАНИЕ
  Создание переменных с одинаковыми именами не рекомендуется.

- **Тип** — тип переменной: булевский, целочисленный или с плавающей запятой;
- **Регистр** — значение регистра, к которому обращается прибор, отображается в таблице;
- **Бит (только для переменных булевского типа)** — номер бита для чтения;
- **Количество регистров (только для целочисленных переменных)** — количество занимаемых переменной регистров: 1 или 2;
- **Функция** — отключение или выбор функции записи/чтения.

От выбора функции записи/чтения зависит список параметров для настройки.

Параметры функций чтения:

- **Период чтения** — временной интервал между запросами;
- **Команда на чтение** — переменная булевского типа, по изменению которой происходит чтение параметра.

Параметры функций записи:

- **Период записи** — временной интервал между операциями перезаписи;
- **Команда на запись** — переменная булевского типа, по изменению которой происходит запись параметра.
- **Запись по изменению** — если эта функция активирована, то в случае изменения значения переменной мастер инициирует запись значения переменной в Slave-устройство.
- **Комментарий** — текстовое описание переменной для отображения в таблице переменных.

5.3.2.2 Режим Slave

В режиме Slave прибор предоставляет данные для считывания другим устройством в сети, самостоятельный опрос не ведет.

Режим Slave для приборов первого поколения

Чтобы настроить параметры передачи данных, следует нажать на имя прибора (по умолчанию — **Сам, 16**) под наименованием интерфейса в дереве настроек.

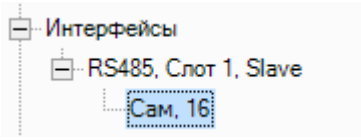


Рисунок 5.40

Параметры прибора

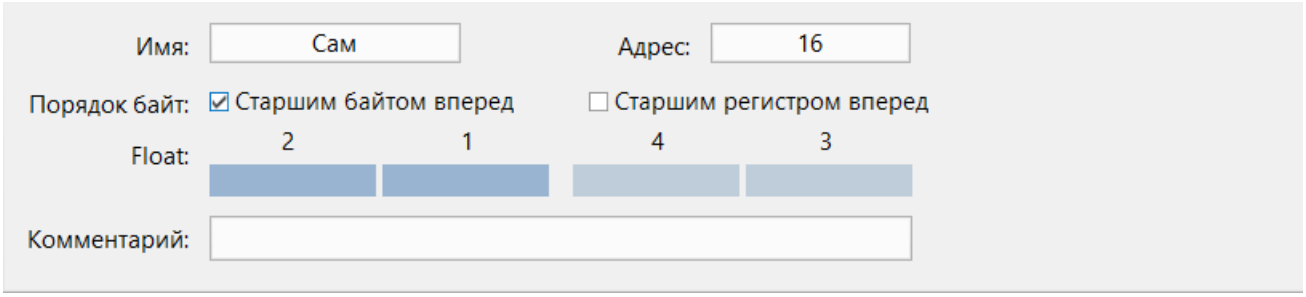


Рисунок 5.41

В правой части окна настроек расположены параметры прибора общие для всех переменных:

- **Имя** — имя прибора для отображения в дереве настроек;
- **Адрес** — сетевой адрес прибора;
- **Комментарий** — текстовое описание прибора.

ПРИМЕЧАНИЕ
В режиме Slave настройки **Старшим регистром вперед** и **Старшим байтом вперед** влияют на порядок упаковки регистров/байт только для системных сетевых переменных устройства.
Для пользовательских сетевых переменных проекта в режиме Slave данные всегда передаются в формате:

- **Старшим регистром вперед**— нет;
- **Старшим байтом вперед**— да.

Для прибора в режиме Slave допускается опрос до 64 регистров при групповом опросе.

Переменные прибора

В нижней части окна располагается таблица переменных прибора, значения которых будут считываться другим устройством.

Чтобы добавить переменную, следует нажать кнопку + . Переменная будет добавлена в таблицу, как на рисунке ниже.

Имя переменной	Тип	Адрес регистра	Комментарий
Var1	Целочисленное	512	
Var2	Целочисленное	513	

Рисунок 5.42

Для удаления из таблицы следует выделить необходимую переменную и нажать кнопку ✕.

Тиражирование переменных

ПРИМЕЧАНИЕ
Тиражирование переменной доступно только для приборов первого поколения.

Чтобы создать несколько переменных со схожими настройками, следует выбрать переменную в таблице и нажать кнопку **Тиражировать**.

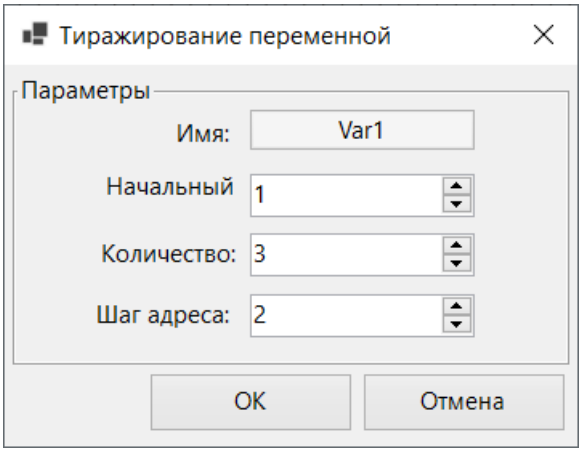


Рисунок 5.43

Настройки тиражирования переменной:

- **Имя** — имя переменной, которая будет тиражирована;
- **Начальный номер** — начальное число, которое будет добавлено к имени тиражируемой переменной;
- **Количество** — количество переменных, которое нужно добавить;
- **Шаг адреса** — шаг, с которым будет увеличиваться значение параметра **Регистр**.

Переменные будут добавлены в таблицу.

Имя переменной	Тип	Адрес регистра	Комментарий
Var1	Целочисленное	512	
Var2	Целочисленное	513	
Var3	Целочисленное	514	
Var11	Целочисленное	516	
Var12	Целочисленное	518	
Var13	Целочисленное	520	

Рисунок 5.44

При тиражировании переменных, добавленных в интерфейс Lora и имеющих в имени постфикс _R или _W, начальный номер устанавливается перед постфиксом.

Свойства переменной прибора

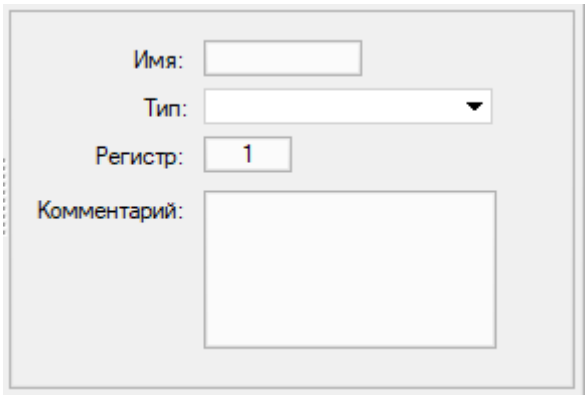


Рисунок 5.45

Свойства переменной прибора настраиваются справа от таблицы:

- **Имя переменной** — имя переменной прибора (задает пользователь);
- **Тип** — тип переменной прибора: целочисленный или с плавающей запятой;

  ПРИМЕЧАНИЕ
  Булевские переменные можно считать в целочисленную переменную. Состояние дискретного входа можно извлечь с помощью блока EXTRACT или подходящего макроса из Менеджера компонентов.

- **Адрес регистра** — адрес регистра переменной прибора (задает пользователь). Диапазон доступных адресов указан в *руководстве по эксплуатации* прибора;
- **Комментарий** — текстовое описание значения переменной для отображения в таблице переменных.

Режим Slave для приборов второго поколения

Приборы второго поколения работают в режиме Slave по умолчанию. Настройка переменных прибора происходит через таблицу переменных, вкладка **Сетевые, Slave**.

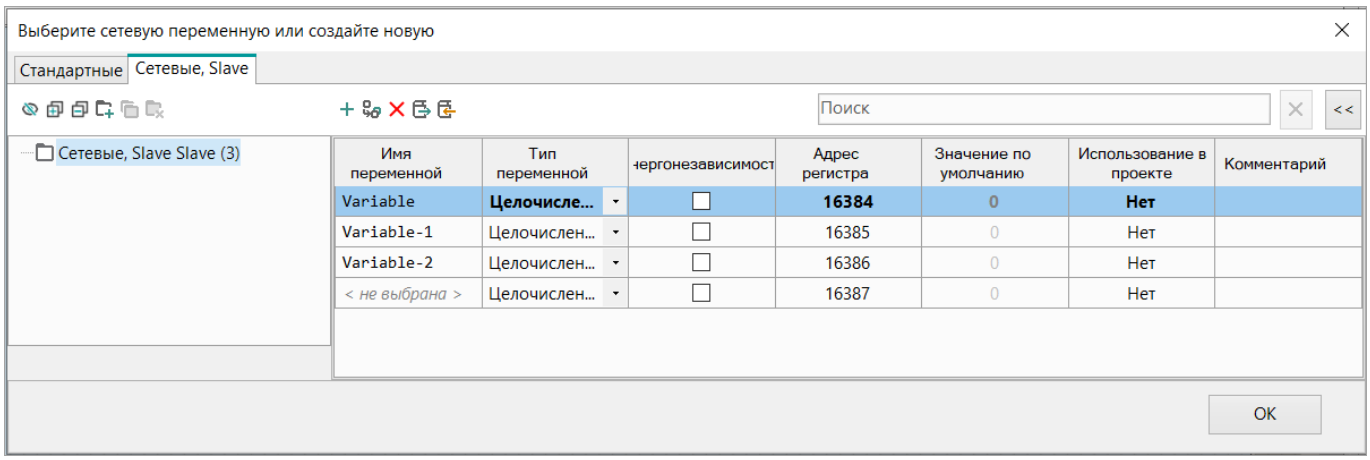


Рисунок 5.46 – Таблица переменных

- **Имя переменной** — имя переменной прибора (задает пользователь);
- **Тип переменной** — целочисленный или с плавающей запятой;

> **ПРИМЕЧАНИЕ**
> Булевские переменные можно считать в целочисленную переменную. Состояние дискретного входа можно извлечь с помощью блока EXTRACT или подходящего макроса из Менеджера компонентов.

- **Энергонезависимость** — при включении чекбокса Энергонезависимость переменная размещается в области системного ЭСППЗУ. При отключении питания значение переменной сохранится в энергонезависимой памяти прибора;
- **Адрес регистра** — адреса сетевых переменных в диапазоне от 16384 до 20479;
- **Значение по умолчанию** — значение переменной, которое будет инициироваться во время начала работы программы;
- **Использование в проекте** — индикация привязки к блокам в программе. Если переменная привязана, то значение — **Да**;
- **Комментарий** — текстовое описание переменной для отображения в таблице переменных.

После подключения прибора к Owen Configurator или OwenCloud сетевые переменные будут отображаться в дереве параметров.

Для настройки отображения следует выбрать переменную в таблице и нажать на кнопку >> **Настройки параметра** в правом верхнем углу окна. Появится панель **Настройка параметра.**

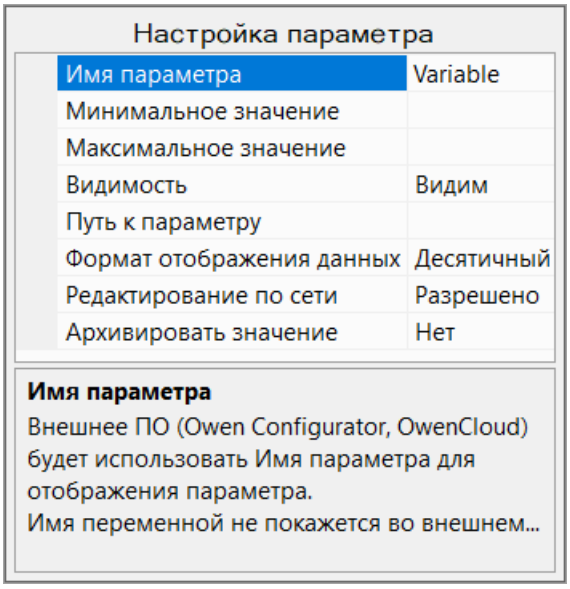


Рисунок 5.47 – Настройка параметра

- **Имя параметра** - название параметра, которое будет отображаться в дереве параметров при работе с Owen Configurator и OwenCloud;
- **Минимальное значение** - минимальное значение параметра для отображения в Owen Configurator и OwenCloud, не влияет на логику исполнения программы;

- **Максимальное значение** - максимальное значение параметра для отображения в Owen Configurator и OwenCloud, не влияет на логику исполнения программы;
- **Видимость** - видимость параметра при работе с Owen Configurator и OwenCloud: **Видим** – параметр отображается в дереве параметров, **Скрыт** – параметр не отображается в дереве параметров;
- **Путь к параметру** - во внешнем ПО параметр будет отображаться по указанному пути. По умолчанию отображается путь, в соответствии с расположением переменной в дереве каталогов таблицы переменных. Иерархия папок при указании пути к параметру оформляется с использованием обратного слэша (\). Если значение не задано, параметр будет отображаться в каталоге «Логика»;
- **Формат отображения данных** - формат, в котором будет отображаться значение параметра, актуален для целочисленных переменных, доступные форматы: десятичный, двоичный (отображается в Owen Configurator как битовая маска), **hex** (шестнадцатиричный) и **enum** (перечисление, только для uint16);
- **Редактирование по сети** - возможность редактирования значения параметра из Owen Configurator и OwenCloud;
- **Архивировать значение** - необходимость архивирования значения параметра: **Да** – значение параметра записывается в архив прибора, **Нет** – значение параметра не записывается в архив прибора.

5.3.2.3 Шаблоны сетевых устройств

Шаблоны сетевых устройств

Для интерфейсов, работающих в режиме Master, доступно создание и использование шаблонов сетевых устройств. Параметры и переменные настроенного устройства могут быть сохранены в виде файла шаблона.

Создание шаблона

Для создания шаблона выберите **Сохранить устройство как шаблон** в контекстном меню устройства или в верхней части окна настройки прибора.

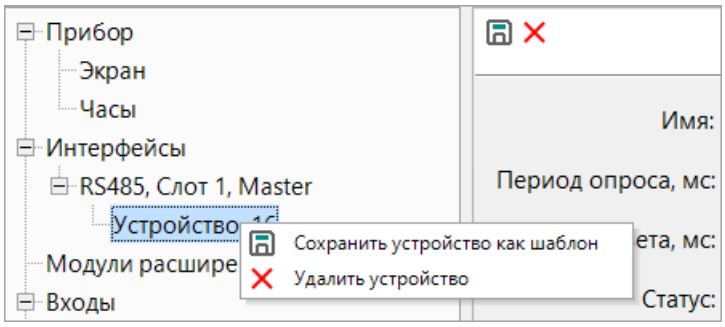


Рисунок 5.48

В открывшемся окне проводника выберите расположение и введите имя файла. Файл будет сохранен с расширением **.dvtpe*. Сохраненный шаблон можно использовать для других интерфейсов и проектов.

Для использования сохраненного шаблона нажмите ПКМ на наименование интерфейса в дереве настроек и выберите **Добавить из Шаблонов…** В открывшемся окне проводника выберите файл содержащий шаблон.

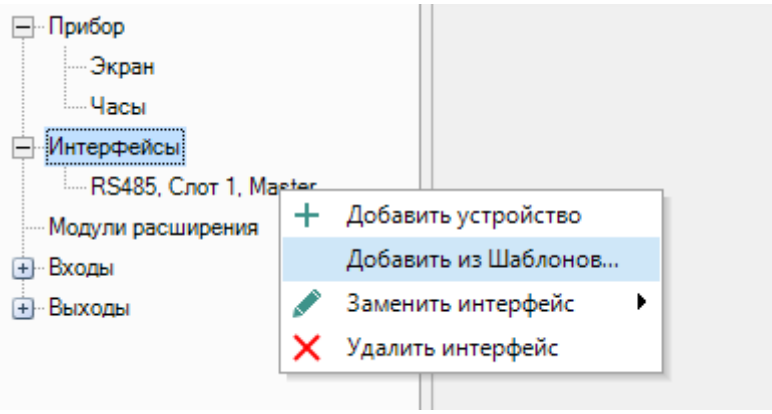


Рисунок 5.49

К интерфейсу добавится устройство из шаблона вместе со всеми параметрами.

Для ряда устройств производства компании ОВЕН есть готовые шаблоны, которые можно загрузить из онлайн-базы с помощью Менеджера компонентов.

При использовании шаблонов сетевых устройств, созданных в проектах для приборов первого поколения, в проектах с приборами второго поколения:

- при выборе **группового опроса** количество регистров в запросе либо сохранит значение (при введенном значении от 2 до 16), либо примет значение 16, если сохраненное в шаблоне значение больше 16:

Рисунок 5.50

- переменная с функцией чтения и записи разделяется на две переменные: с функцией чтения _read и с функцией записи _write:

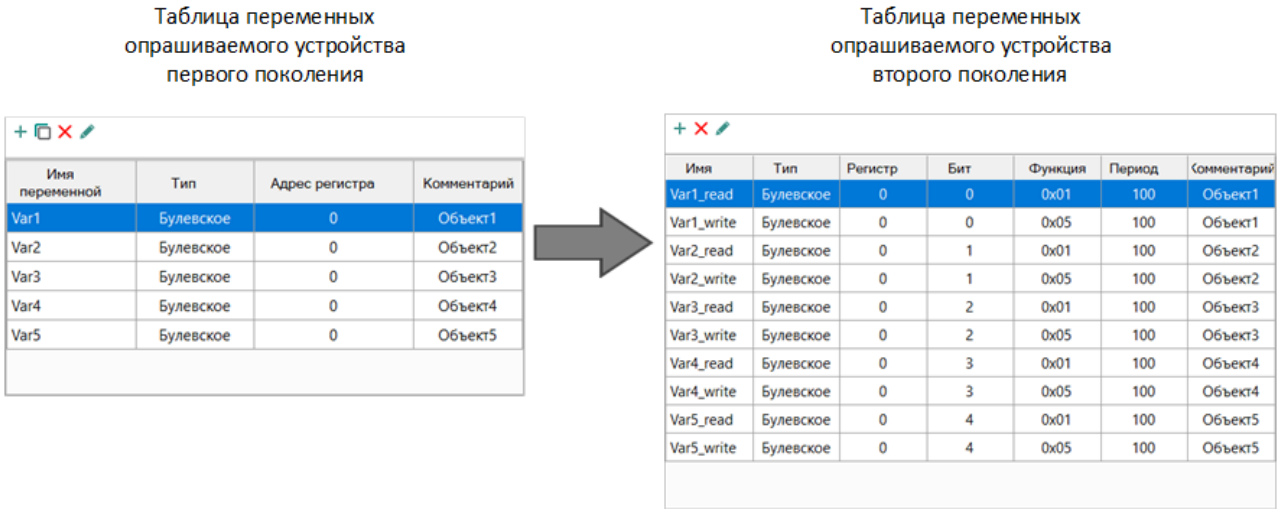


Рисунок 5.51

При использовании шаблонов сетевых устройств, созданных в проектах для приборов второго поколения, в проектах с приборами первого поколения параметр период опроса примет значение по умолчанию (100 мс).

Групповое переименование переменных

Для группового переименования переменных в таблице переменных опрашиваемого устройства окна настройки прибора нажмите кнопку .

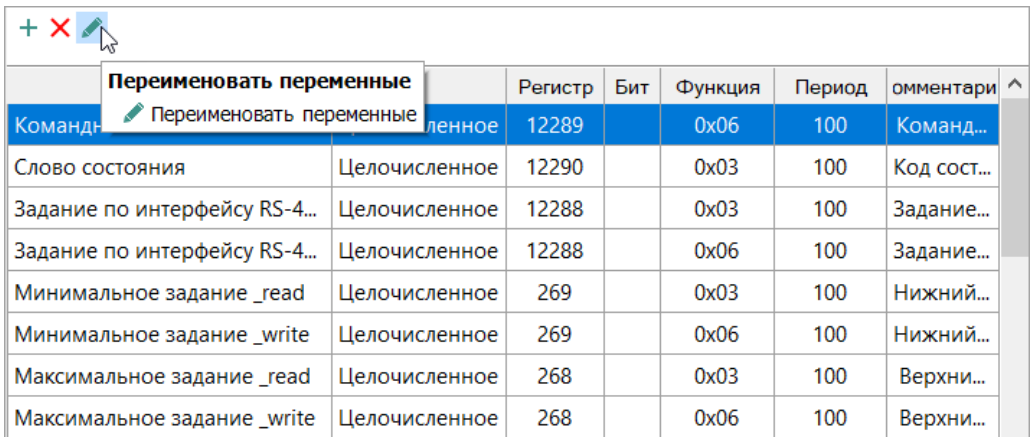


Рисунок 5.52

В появившемся окне введите текст для отображения перед именем, и после имени переменной, и нажмите кнопку **Применить**.

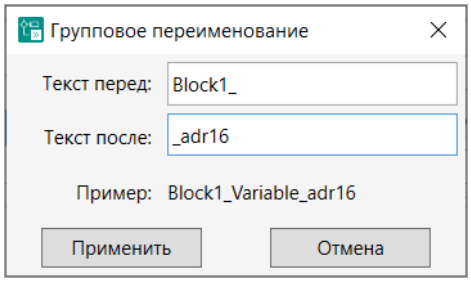
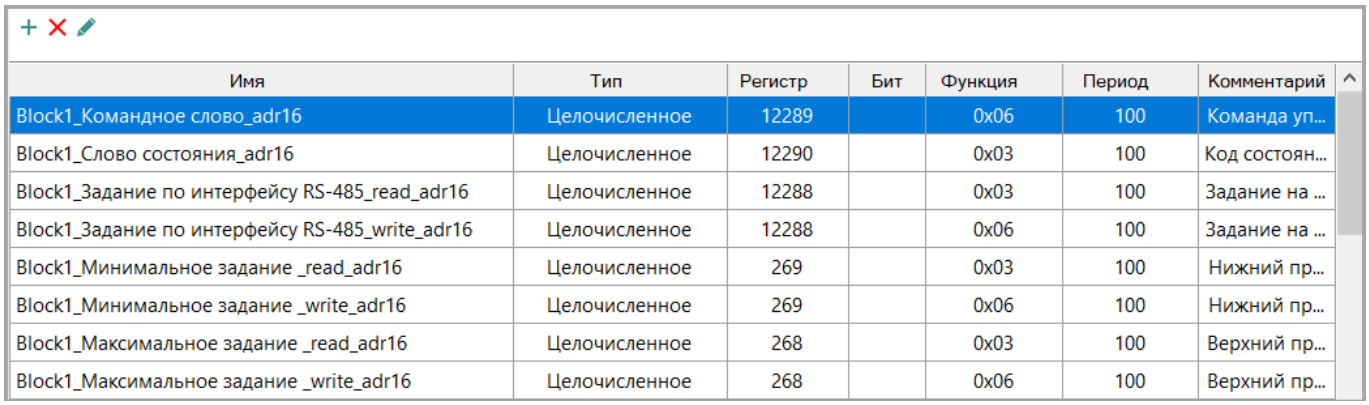


Рисунок 5.53

В результате все переменные переименуются согласно введенным данным:



Имя	Тип	Регистр	Бит	Функция	Период	Комментарий
Block1_Командное слово_adr16	Целочисленное	12289		0x06	100	Команда уп...
Block1_Слово состояния_adr16	Целочисленное	12290		0x03	100	Код состоян...
Block1_Задание по интерфейсу RS-485_read_adr16	Целочисленное	12288		0x03	100	Задание на ...
Block1_Задание по интерфейсу RS-485_write_adr16	Целочисленное	12288		0x06	100	Задание на ...
Block1_Минимальное задание _read_adr16	Целочисленное	269		0x03	100	Нижний пр...
Block1_Минимальное задание _write_adr16	Целочисленное	269		0x06	100	Нижний пр...
Block1_Максимальное задание _read_adr16	Целочисленное	268		0x03	100	Верхний пр...
Block1_Максимальное задание _write_adr16	Целочисленное	268		0x06	100	Верхний пр...

Рисунок 5.54

При тиражировании переменной, переименованной с помощью группового переименования, "Текст после" не изменится, начальный и последующие номера добавятся к имени тиражируемой переменной.

Настройки группового переименования сохраняются в шаблоне, и доступны для редактирования при последующем использовании шаблона.

Переименование каждой переменной по отдельности доступно в таблице переменных и в коде программы.

5.4 Модули расширения

В разделе **Модули расширения** настраиваются подключенные к прибору модули расширения ввода/вывода.

Добавить модуль расширения

Чтобы добавить модуль расширения, следует нажать ПКМ на раздел **Модули расширения** в дереве настроек и выбрать **Добавить модуль расширения** → **< Модель модуля расширения >**.

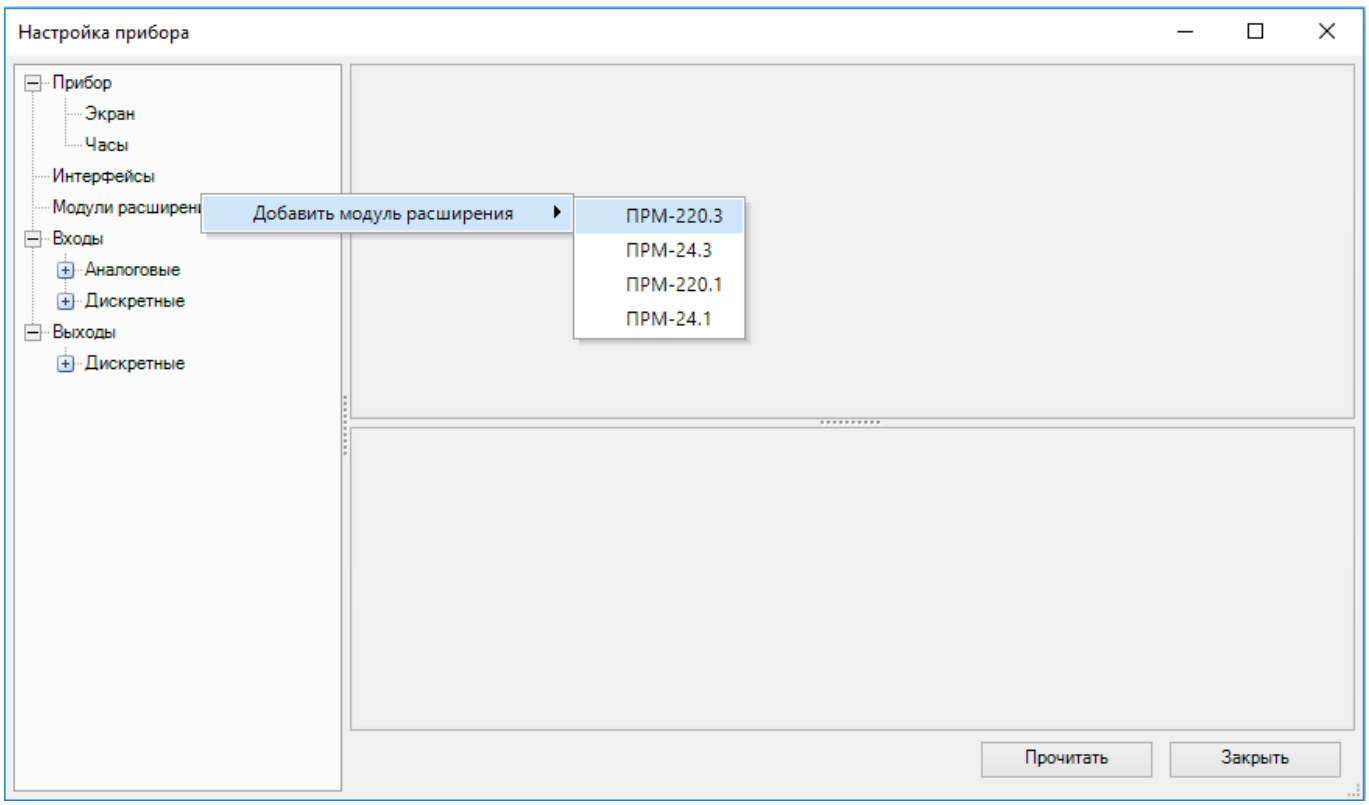


Рисунок 5.55

Модуль расширения отобразится как ответвление раздела в дереве настроек. Наименование модуля расширения состоит из модели модуля и номера занимаемого слота.

Модуль расширения можно удалить, нажав ПКМ на его наименование в дереве настроек и выбрав соответствующую функцию.

Параметры модуля расширения

Параметры выделенного модуля расширения отображаются в правой части окна настроек.

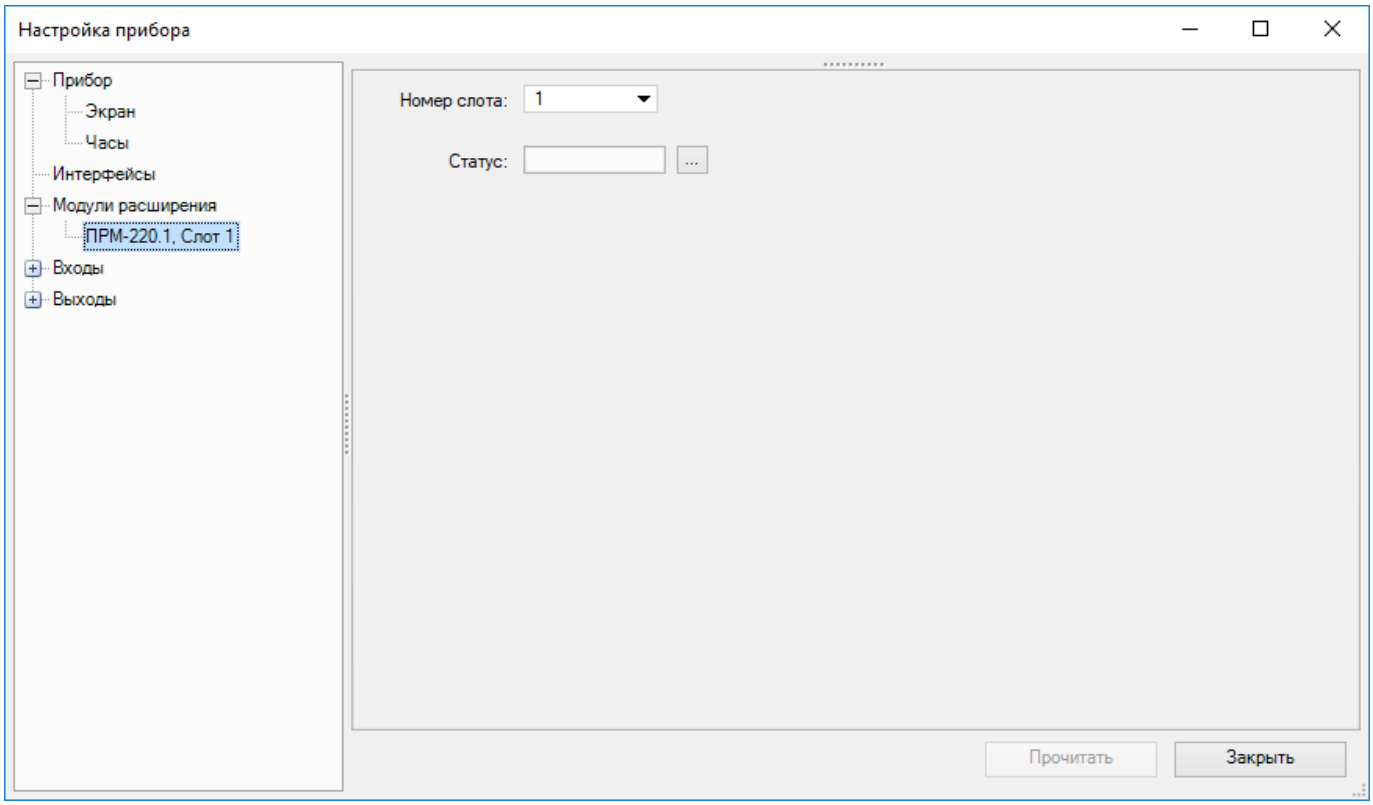


Рисунок 5.56

В выпадающем меню **Номер слота** указывается номер слота в приборе, занимаемый модулем расширения. Количество модулей расширения не может превышать количество слотов в приборе.

Статус подключения модуля расширения может сохраняться в переменную булевского типа. Если связь с модулем установлена, то переменная хранит значение **True**, если связь не установлена — **False**. Для выбора переменной, в которой будет храниться статус, нажмите кнопку «**...**».

Откроется таблица переменных, в которой можно выбрать переменную из проекта или создать новую.

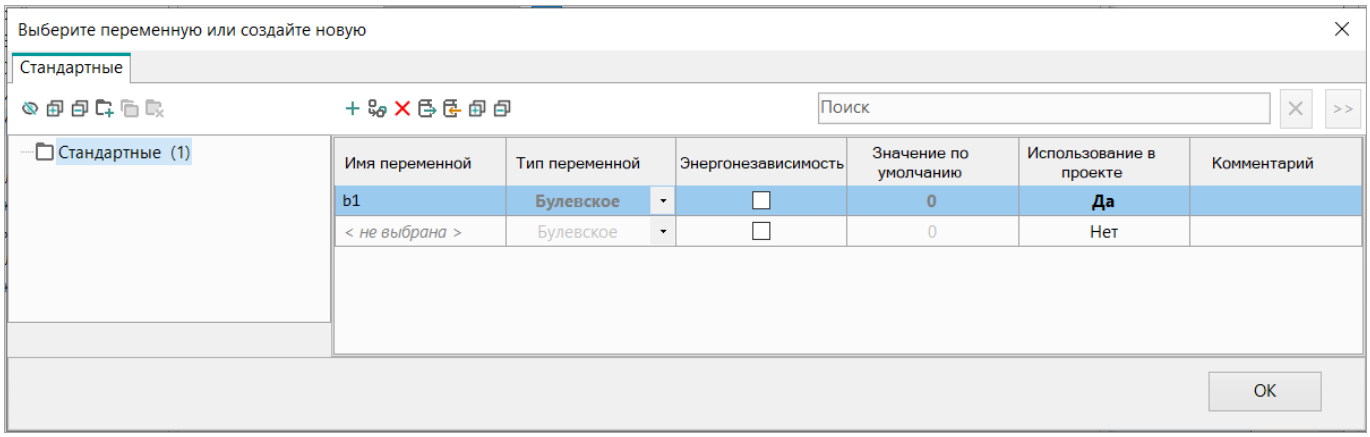


Рисунок 5.57

Выбранная переменная будет отображаться в поле ввода **Статус**.

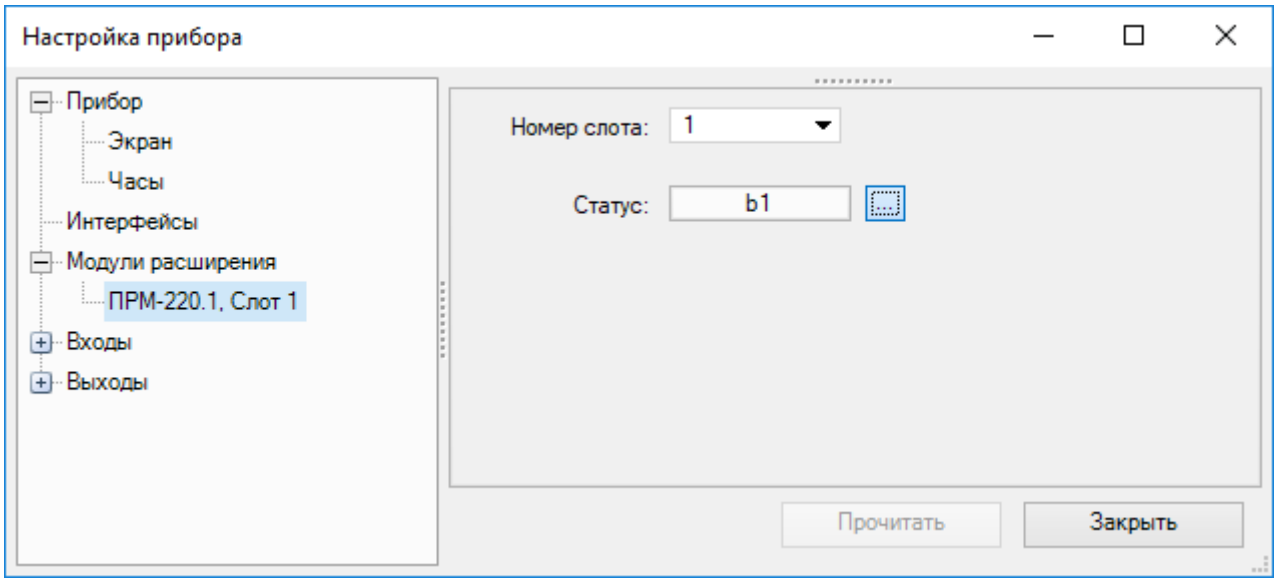


Рисунок 5.58

Для выходов модуля расширения может быть настроено безопасное состояние.

Настройки для приборов второго поколения

ПРИМЕЧАНИЕ
Список приборов второго поколения см. в разделе О программе.

Окно настройки для приборов второго поколения имеет другой интерфейс и параметры оформлены в табличном виде.

ПРИМЕЧАНИЕ
Информация о подключенных к прибору модулях расширения отображается в окне Информация о приборе.

Чтобы добавить модуль расширения, следует нажать ПКМ на раздел **Модули расширения** в дереве настроек и выбрать **Добавить модуль расширения** → **< Модель модуля расширения >**.

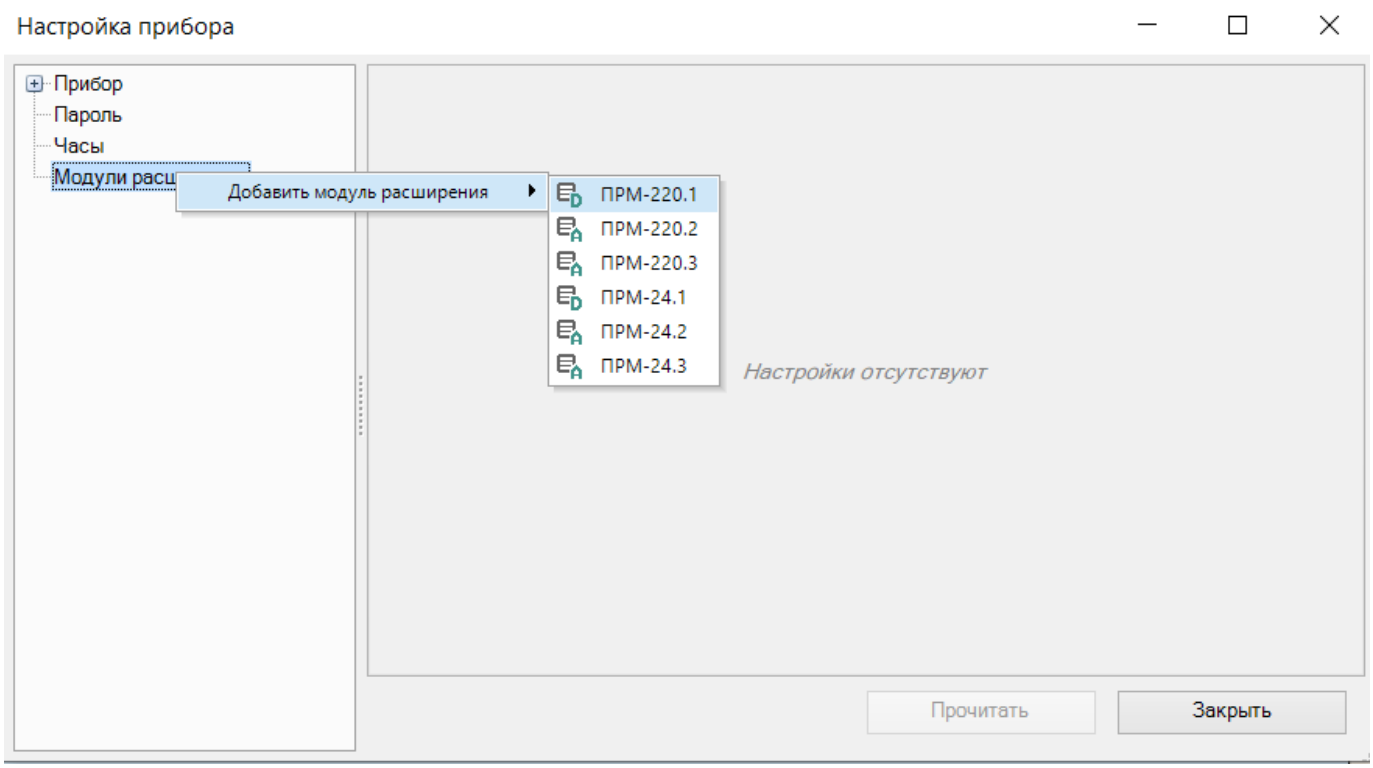


Рисунок 5.59

Модули расширения настраиваются аналогично другим приборам, только меню настроек безопасного состояния выходов находится в ответвлении добавленного модуля расширения, см. безопасное состояние.

5.5 Входы и выходы

В разделах **Входы** и **Выходы** настраиваются параметры и комментарии для входов/выходов прибора и модулей расширения.

Параметры зависят от типа входов/выходов выбранного прибора.

Входы/выходы модулей расширения отображаются в дереве настроек с номером занимаемого слота в скобках.

Для правильной настройки входов/выходов прибора следует обратиться к *руководству по эксплуатации* прибора.

Добавление комментариев

Комментарий для входа/выхода можно написать в поле **Комментарий** или на панели свойств.

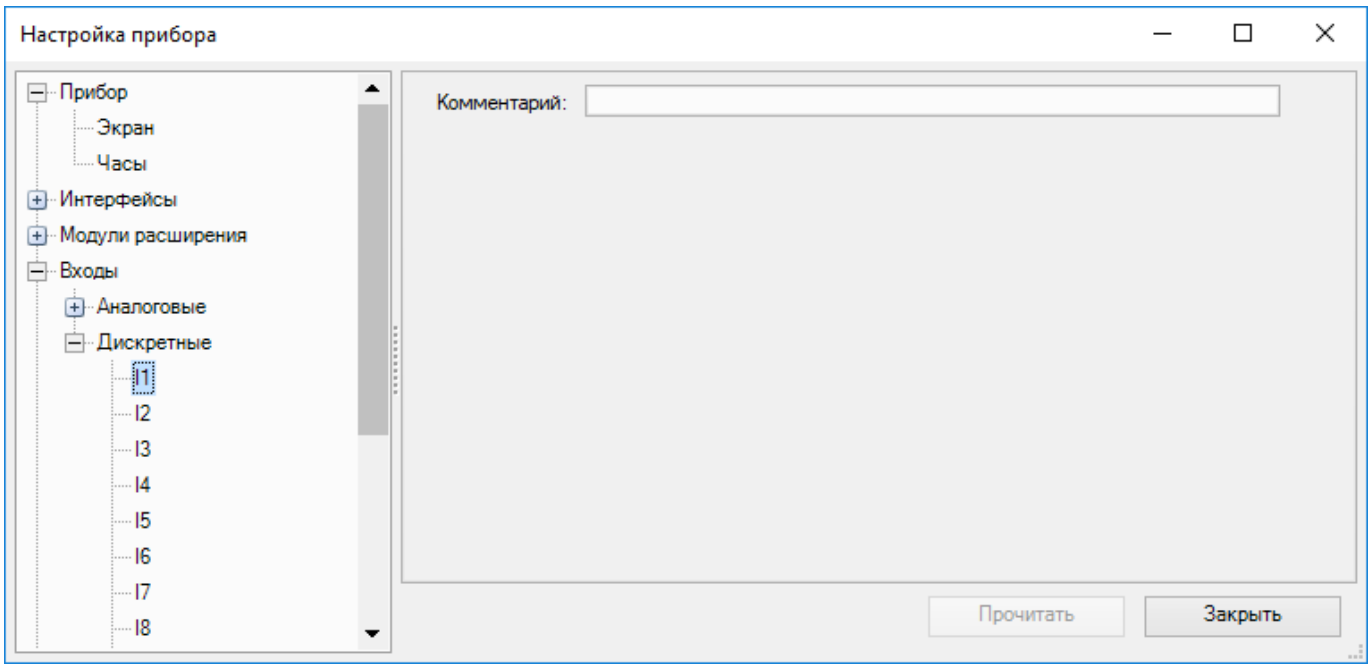


Рисунок 5.60

Текст введенного комментария будет отображаться во всплывающей подсказке при наведении курсора на вход/выход на схеме:

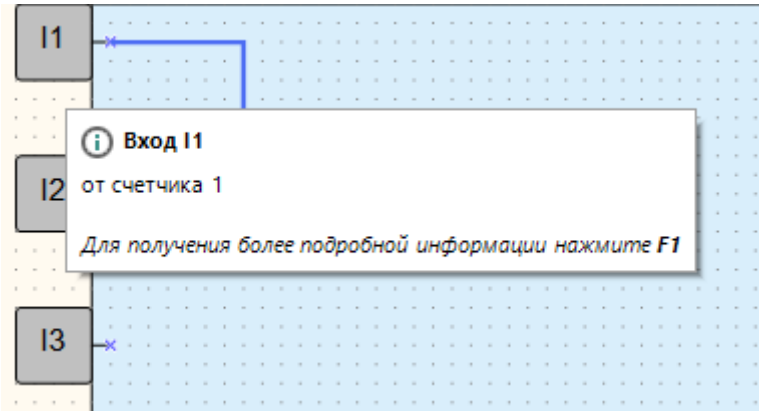


Рисунок 5.61

Безопасное состояние выходов модулей расширения

Для выходов модулей расширения можно настроить безопасное состояние. Выход будет принимать безопасное состояние в случае потери связи с прибором.

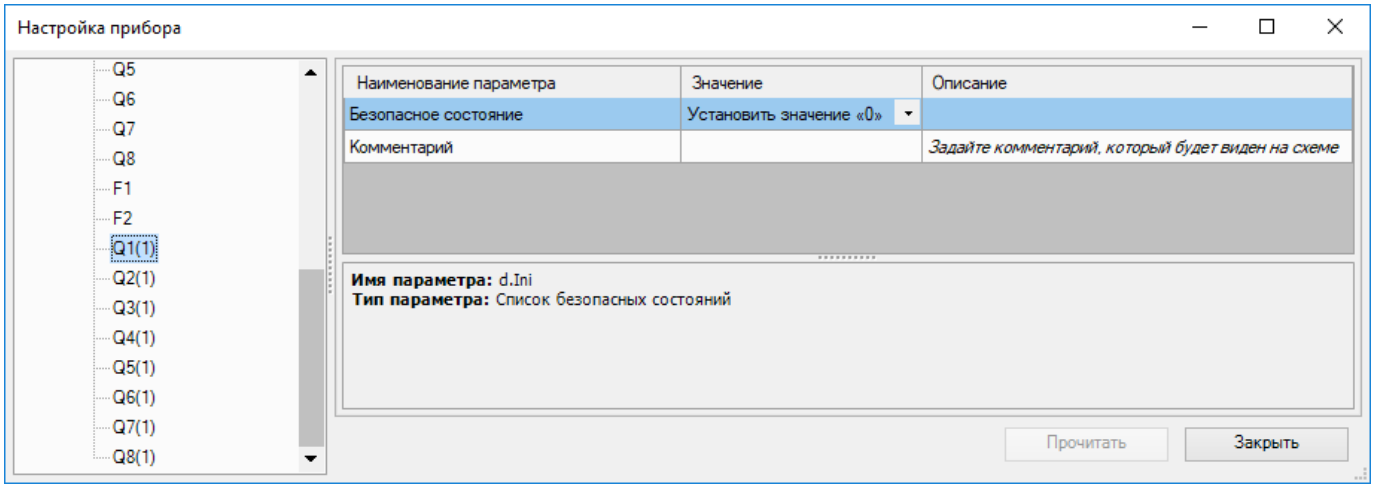


Рисунок 5.62

Настройки для приборов второго поколения

Окно настройки входов и выходов для приборов второго поколения имеет другой интерфейс, и параметры в правой части окна оформлены в табличном виде.

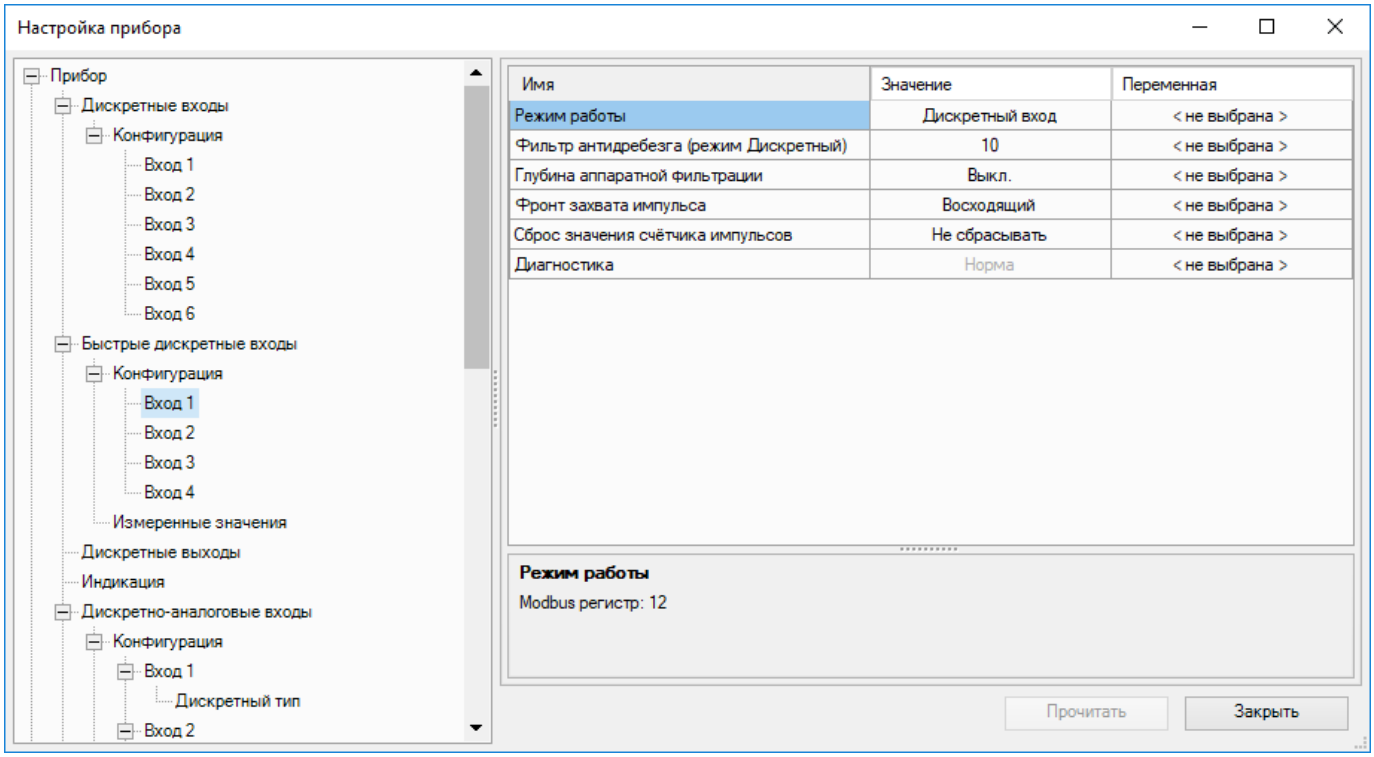


Рисунок 5.63

Для приборов второго поколения меню настроек безопасного состояния выходов находится в ответвлении добавленного модуля расширения.

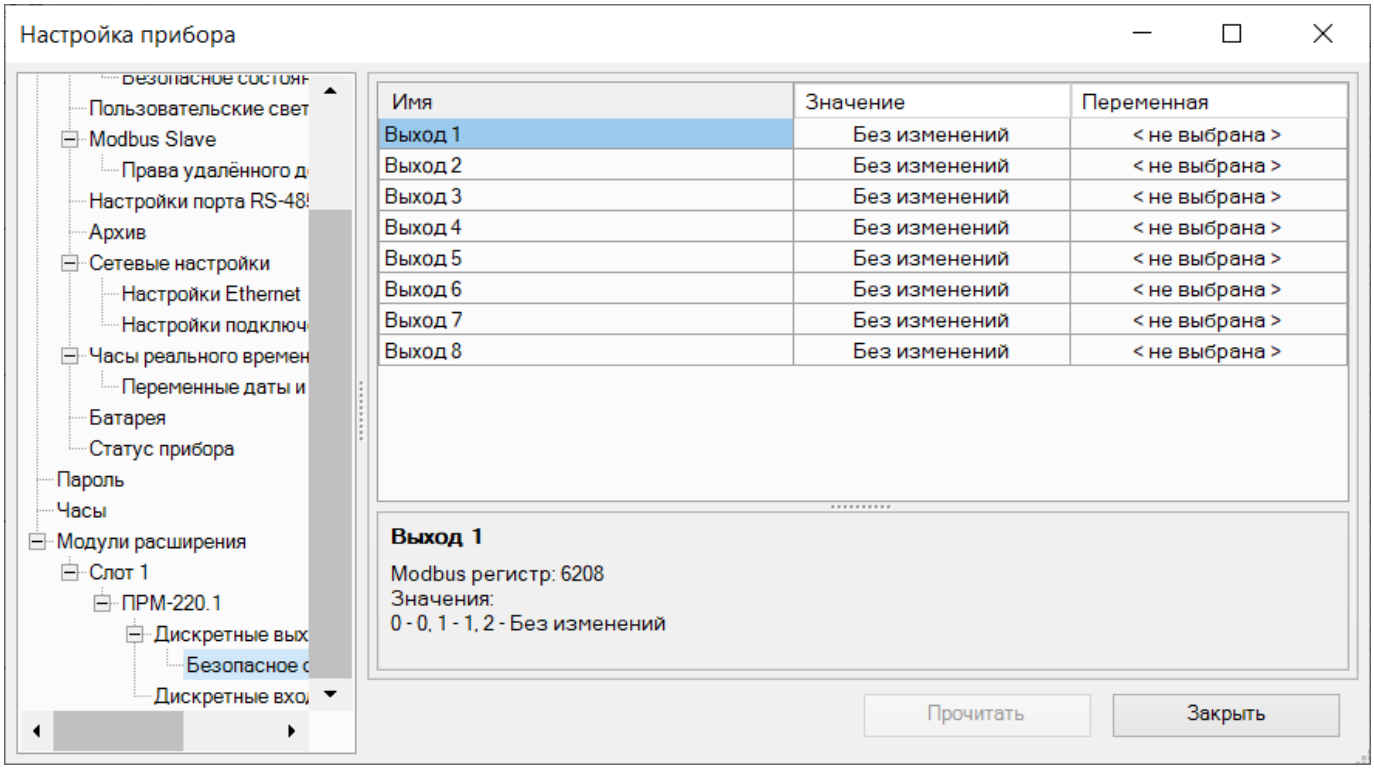


Рисунок 5.64

5.6 Пароль

Для приборов второго поколения можно установить пароль для защиты устройства.

ПРИМЕЧАНИЕ
Список приборов второго поколения см. в разделе О программе.

Пароль задается в разделе **Пароль** в дереве настроек только для подключенного к ПК устройству.

Создание пароля

Если в приборе не установлен пароль, то в окне настроек будет активно создание пароля. Для установки следует ввести и подтвердить пароль.

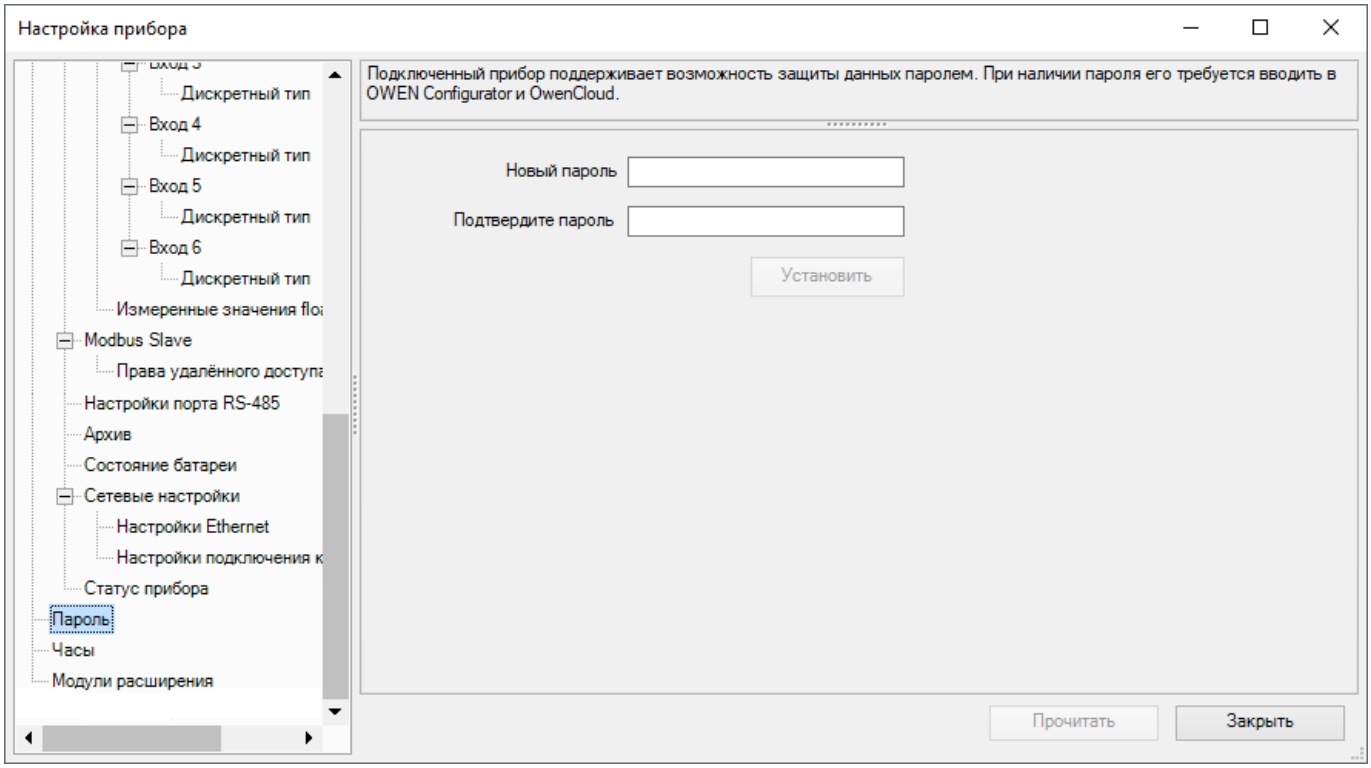


Рисунок 5.65

Изменение и сброс пароля

Если в приборе уже установлен пароль, то его можно изменить или сбросить.

Для изменения пароля следует ввести текущий пароль и новый пароль в графах **Изменение пароля**.

Для сброса пароля следует ввести текущий пароль в графе **Сброс пароля**.

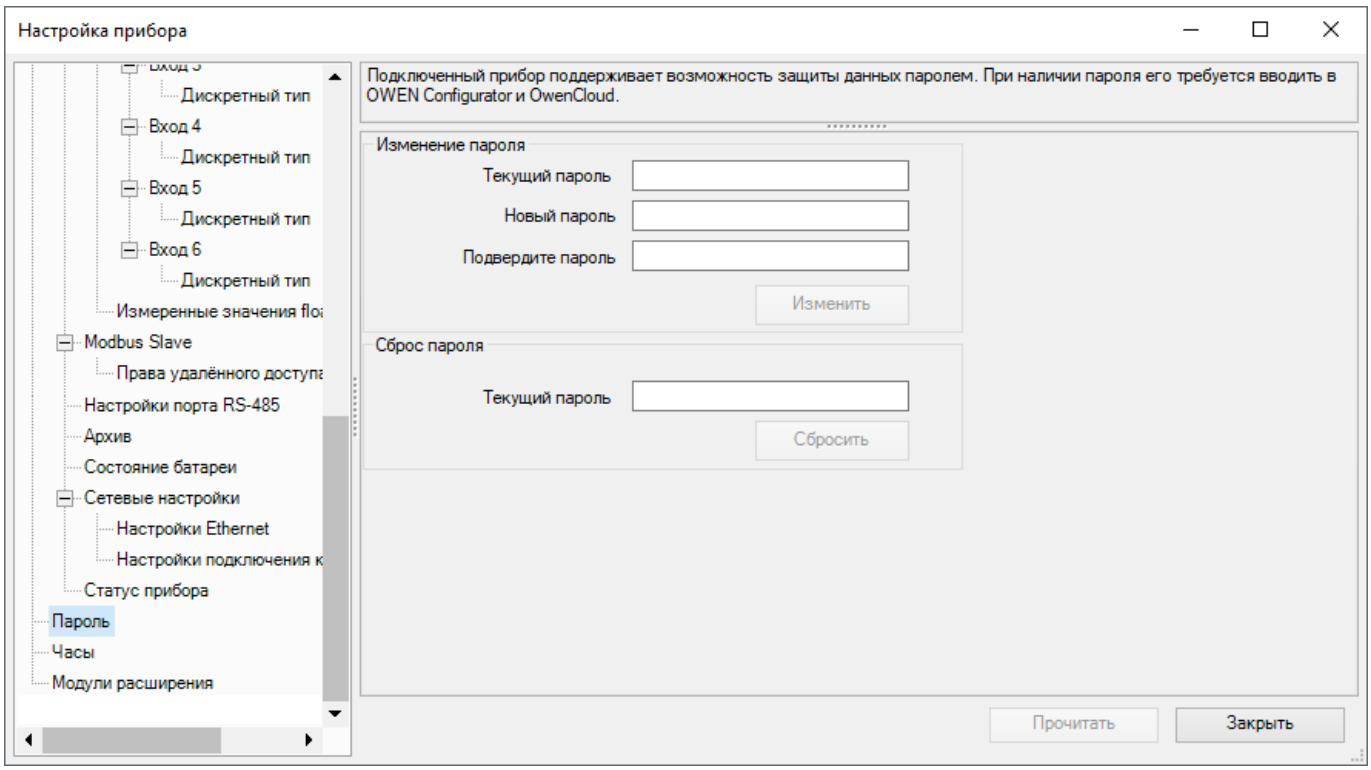


Рисунок 5.66

В случае утери пароля для его сброса см. *Руководство по эксплуатации* прибора.

Для защищенного паролем прибора во время записи программы требуется ввод пароля, см. Запись программы в прибор.

ПРИМЕЧАНИЕ
Пароль обязательно требуется для подключения прибора к OwenCloud, подробнее см. Подключение к OwenCloud.

5.7 Подключение к OwenCloud

С помощью шлюзов линейки Пх210 к OwenCloud можно подключать приборы первого и второго поколения.

Прямое подключение к OwenCloud доступно только для приборов второго поколения.

ПРИМЕЧАНИЕ
Для подключения к OwenCloud прибор обязательно должен быть защищен паролем. Без пароля прибор не подключится к OwenCloud. Об установке пароля см. раздел Пароль.

Перед настройкой подключения к OwenCloud следует настроить авторизацию прибора в OwenCloud, подробнее см. Руководство пользователя OwenCloud и *Руководство по эксплуатации* прибора. Во время работы прибор должен быть подключен к Интернету для передачи данных.

В разделе **Настройки подключения к OwenCloud** дерева настроек задается подключение и статус подключения к OwenCloud.

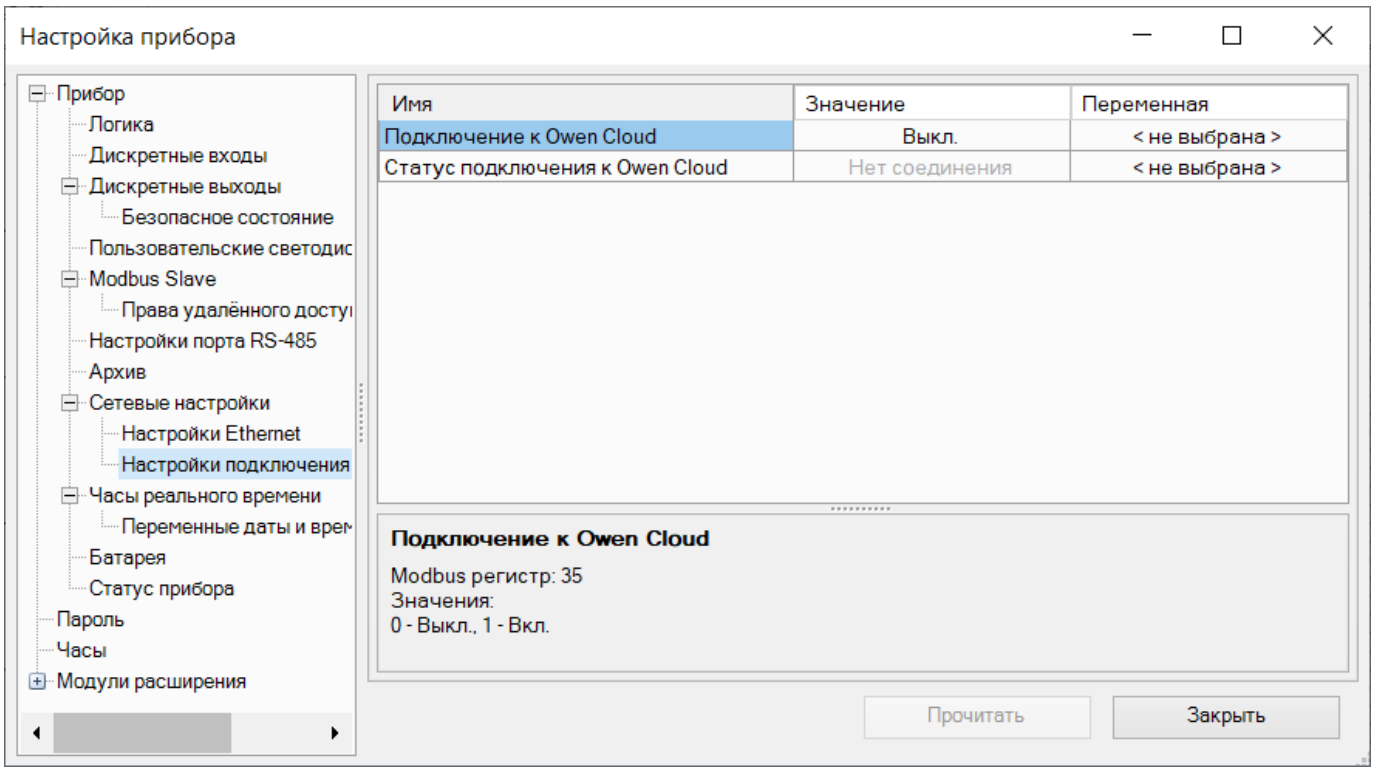


Рисунок 5.67

В разделе **Права удаленного доступа из OwenCloud** дерева настроек задаются уровни доступа OwenCloud к прибору.

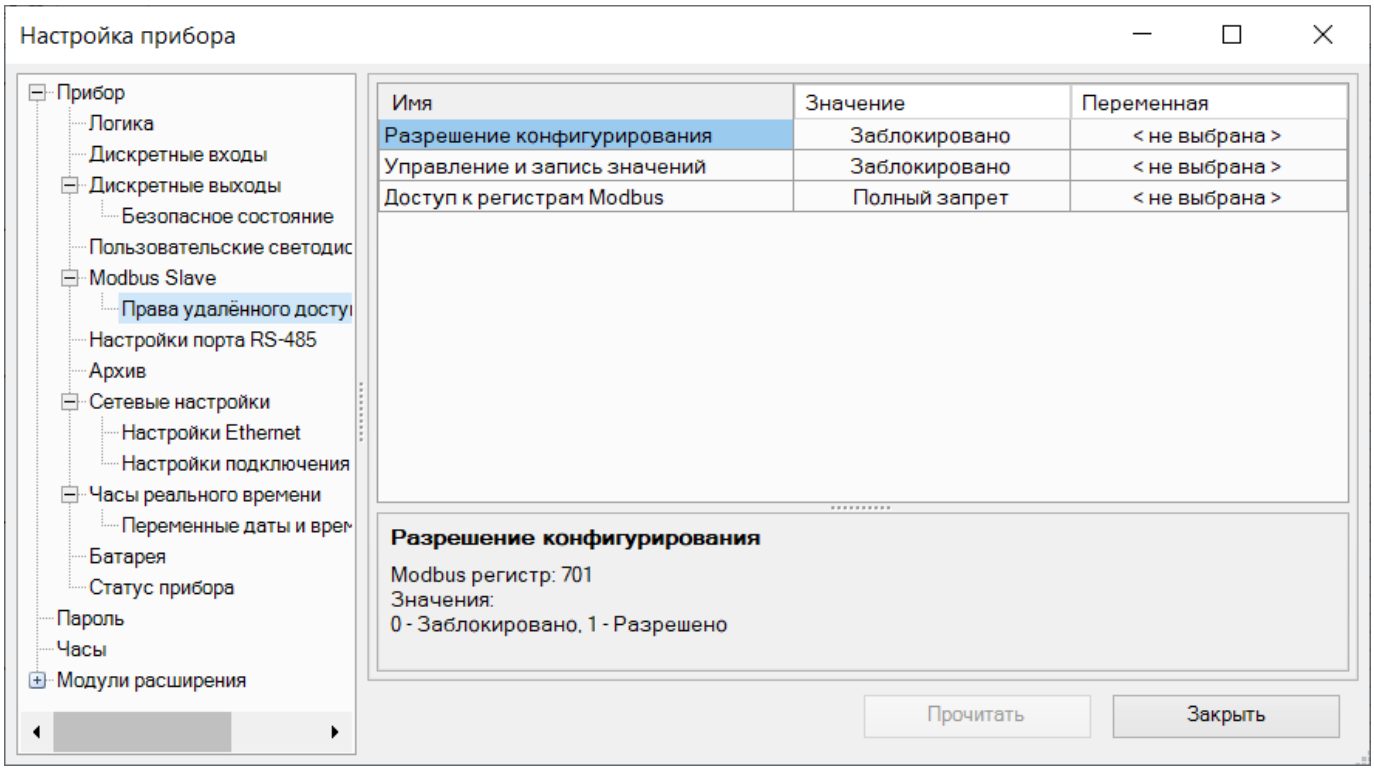


Рисунок 5.68

Пример работы с OwenCloud см. в разделе Прямое подключение приборов второго поколения к OwenCloud.

6 Переменные

Переменные служат для записи и считывания значений на схеме проекта и при программировании экранов.

Задействованные переменные

Переменная является **задействованной**, если выполняется **хотя бы одно** из следующих условий:

- Общие условия для всех типов переменных:
  - имеет привязку к элементу визуализации;
  - имеет привязку к параметру прибора;
  - имеет привязку к физическим кнопкам;
  - используется в логических переходах между экранами;
  - для переменной на схеме присутствует выходной блок энергонезависимой переменной (Энергонезависимость = Вкл).
- Дополнительные условия, для переменных типа массив:
  - Для массива с **элементами булевского типа** — при энергонезависимости "Выкл" на схеме присутствует выходной блок массива или элементов массива;
  - Для массива с **элементами целочисленного и вещественного типа** — при энергонезависимости "Выкл" на схеме присутствует входной блок массива или элементов массива.

Если ни одно из условий не выполняется, то такая переменная является **незадействованной**.

Незадействованные переменные:

- не участвуют в логике программы;
- не участвуют в симуляции программы;
- не занимают память прибора.

Таблица переменных

Переменные проекта отображаются в Таблице переменных.

Для запуска Таблицы переменных нажмите кнопку на панели инструментов или выберите в главном меню **Прибор** → **Таблица переменных**.

Таблица переменных содержит **Дерево каталогов переменных**, а также вкладки:

- Стандартные переменные;
- Сервисные переменные;
- Сетевые переменные (переменные интерфейса связи) — для каждого интерфейса связи отдельная вкладка.

Дерево каталогов переменных позволяет создавать разделы каталогов для размещения переменных. Для каждой категории переменных отображается своё дерево каталогов. Для стандартных и сетевых переменных возможно создавать дополнительные разделы каталогов (максимальная вложенность — 3). Каталоги сервисных переменных неизменяемые.

Рядом с названием каждого каталога в скобках указывается количество переменных в каталоге - с учётом переменных, находящихся во вложенных каталогах, либо без них, в зависимости от настройки "Отображение вложенных переменных".

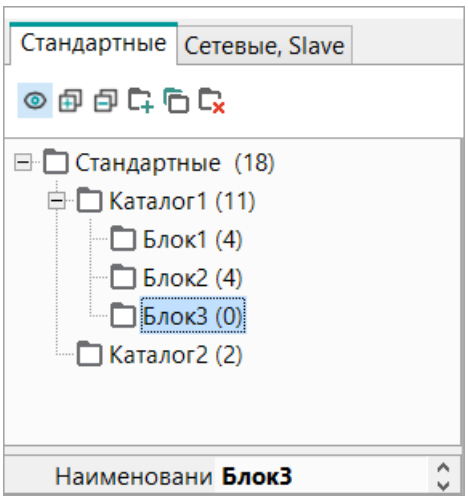


Рисунок 6.1 – Дерево каталогов переменных

	Включить/выключить отображение вложенных переменных • Кнопка **Включить** отображение вложенных переменных — в скобках отображается общее количество переменных, находящихся в текущем каталоге и во всех его вложенных подкаталогах. • Кнопка — в скобках отображается количество переменных, находящихся непосредственно в выбранном каталоге, без учета вложенных подкаталогов. По умолчанию отображение вложенных переменных включено.
	Раскрыть подкаталоги
	Свернуть подкаталоги
	Создать новый подкаталог. Также создание подкаталога доступно через контекстное меню каталога.
	Дублировать каталог — выбранная папка продублируется в родительский каталог со всеми вложенными разделами и добавленными переменными. Также копирование раздела доступно через контекстное меню каталога. Выберите **Копировать**, затем выберите каталог, в который необходимо добавить скопированный раздел, нажмите ПКМ и выберите **Вставить** **ПРИМЕЧАНИЕ** Каталог для добавления информации должен быть на уровень выше копируемого.
	Удалить каталог. Также удаление раздела каталога доступно через контекстное меню каталога. **ПРИМЕЧАНИЕ** Раздел каталога, содержащий переменные, удалить невозможно.

Для перемещения раздела каталога нажмите и удерживайте клавишу Shift, и с помощью drag&drop переместите раздел на новое место.

ПРИМЕЧАНИЕ
Перемещение разделов доступно только в пределах родительского каталога.

Для переименования раздела каталога введите новой имя в строке **Наименование**, расположенной в нижней части **Дерева каталогов переменных**.

ПРИМЕЧАНИЕ
Поле Наименование не может быть пустым или дублировать имя другого раздела в пределах родительского каталога. В случае ввода некорректного имени появится окно с сообщением об ошибке.

Чтобы определить раздел каталога, в котором находится переменная, выделите переменную в таблице и нажмите ***Ctrl+H***. Раздел, в котором находится переменная, в дереве каталогов выделится синим цветом, наименование раздела отобразится в строке снизу, в таблице переменных откроется тот раздел, где расположена переменная. Или наведите мышку на переменную — во всплывающем окне отобразится имя переменной и ее расположение.

Чтобы использовать переменную в проекте, сначала следует создать ее в таблице переменных.

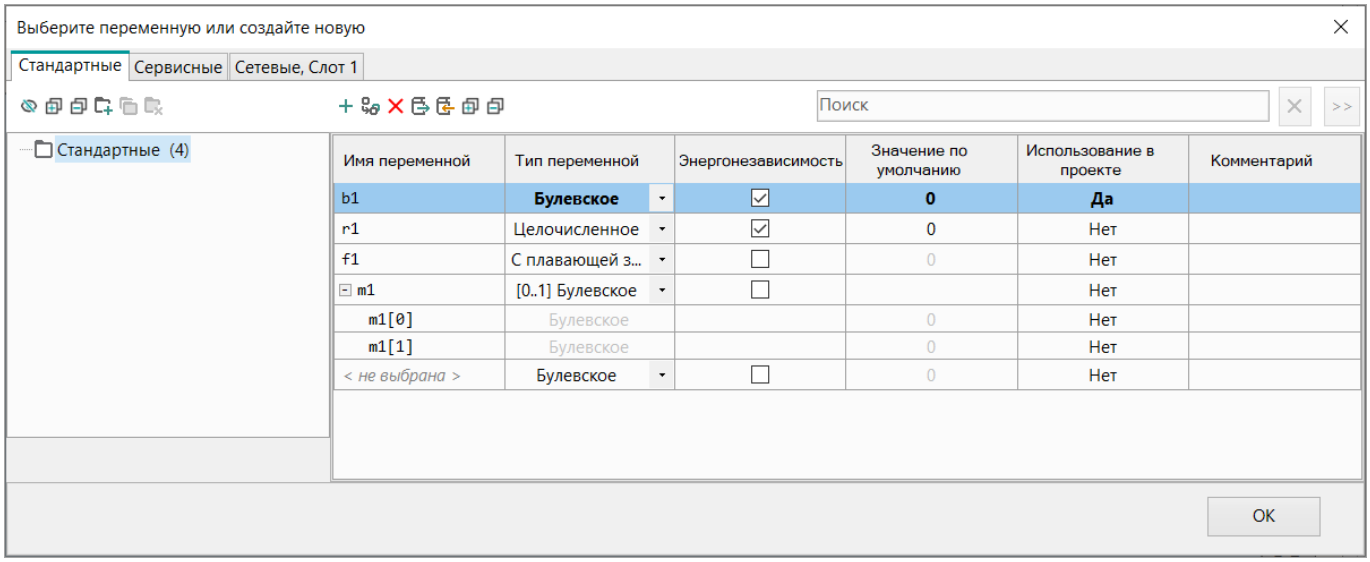


Рисунок 6.2 – Таблица переменных

+	Создать переменную
	Дублировать переменную
×	Удалить переменную
	Экспортировать стандартные переменные в CSV файл
	Импортировать стандартные переменные из CSV файла
	Раскрыть все массивы
	Свернуть все массивы

Для создания переменной нажмите кнопку + в левой верхней части таблицы, или введите имя переменной в столбце “Имя переменной”. Задайте тип, заполните остальные параметры в зависимости от эксплуатационной потребности.

- **Имя переменной** – имя для отображения переменной в проекте. При создании массива элементы массива автоматически получают имена, содержащие имя массива с добавлением индекса в квадратных скобках;
- **Тип переменной** – может быть назначен один из трех типов: булевский, целочисленный, с плавающей запятой, массив. См. подробнее раздел Типы переменных;

  При выборе переменной типа массив, в появившемся окне **Настройка массива** введите:

  – **Размер массива** — количество элементов массива, не более 256. Значения, задаются в формате [x..y], где x - индекс первого элемента, y - последнего;

  – **Тип элементов** — тип элементов массива: булевский, целочисленный, с плавающей запятой.

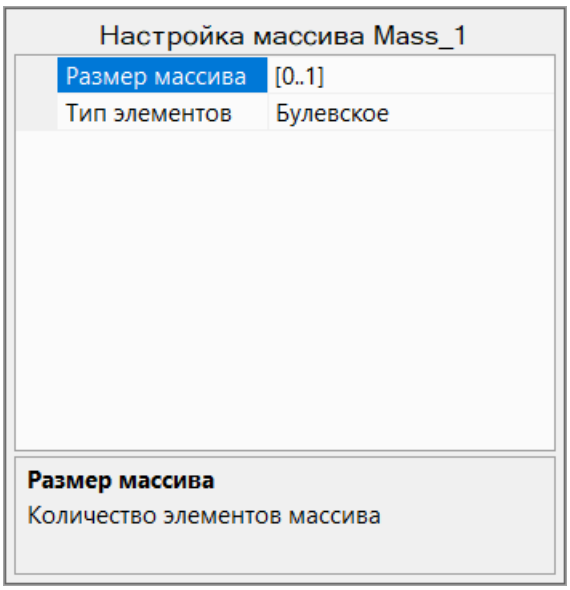


Рисунок 6.3 – Окно Настройка массива

Изменение размера, диапазона индексов и/или типа элементов массива возможно в любой момент.

- **Энергонезависимость** — для сохранения значения переменной в ПЗУ прибора в случае отключения электропитания следует включить чекбокс в поле настройки. Время хранения переменной в ПЗУ зависит от типа прибора, см. *руководство по эксплуатации* прибора;

  ПРИМЕЧАНИЕ
  Для приборов второго поколения, при включении в таблице переменных чекбокса Энергонезависимость, размещение значения переменной происходит:

  – в системное ЭСППЗУ — для стандартной переменной, привязанной к параметру прибора или к элементу визуализации. Для сетевой Slave переменной;

  – в ЭСППЗУ — для стандартной переменной, не привязанной к параметру прибора или к элементу визуализации.

- **Значение по умолчанию** — значение, которое будет хранить переменная, пока в нее не будет записано новое значение;
- **Использование в проекте** — индикация использования переменной в проекте. Значение"Да" устанавливается если переменная является **задействованной**, и/или имеет привязку к входному или выходному блоку на схеме;
- **Комментарий** — текстовое описание для отображения во всплывающей подсказке на схеме проекта при наведении курсора на блок переменной.

Для дублирования переменной нажмите ПКМ на строку переменной и в контекстном меню выберите **Дублировать переменную** или нажмите кнопку в верхней части таблицы.

ПРИМЕЧАНИЕ
Дублирование элемента массива невозможно.

Чтобы удалить переменную из таблицы, нажмите ПКМ на строку переменной и в контекстном меню выберите **Удалить переменную** или нажмите кнопку в верхней части таблицы. В случае, если переменная используется в проекте, отобразится информационное окно, с предложением удалить ссылки на переменную. При выборе **Удалить** переменная будет удалена из таблицы переменных и блока на схеме.

ПРИМЕЧАНИЕ
Удаление элемента массива невозможно.

Экспорт переменных в файл

Для экспорта вкладки переменных в виде таблицы в формате ***.csv***:

1. Нажмите **Экспортировать переменные в CSV файл** в верхней части таблицы.
2. В открывшемся окне укажите место выгрузки файла.
3. Нажмите кнопку **Сохранить**.

ПРИМЕЧАНИЕ
Имя файла формируется в зависимости от экспортируемой вкладки по схеме ***ИмяПроекта_Вкладка_Переменные***.

ПРИМЕЧАНИЕ
Для приборов второго поколения вкладка Slave сетевых переменных экспортируется вместе с настройками параметров переменных.

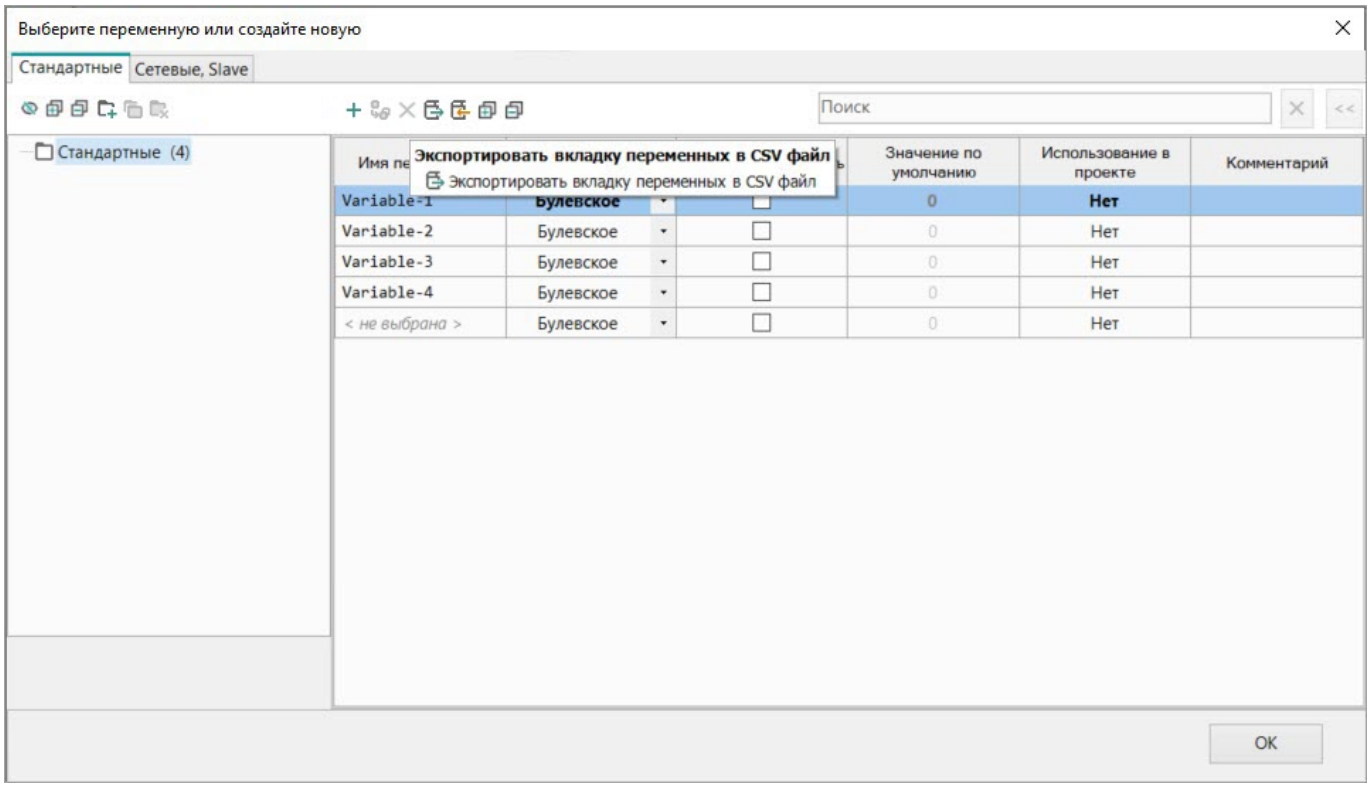


Рисунок 6.4

Импорт переменных из файла

Для импорта таблицы переменных из файла в формате ***.csv*** нажмите кнопку **Импортировать переменные из CSV файла** в верхней части таблицы. Выберите файл из которого необходимо импортировать данные и нажмите **Открыть**. Каталоги, при наличии, также импортируются из файла с сохранением вложенности переменных. В случае присутствия в файле некорректных данных, например при отсутствии столбца с именем, типом переменной или регистром, будет отображено сообщение об ошибке. После устранения ошибки выполните импорт заново. При совпадении имен и/или регистров переменных в проекте и импортируемом файле, появится информационное окно с сообщением обо всех пересечениях имен и/или регистров и предложением продолжить либо отменить импорт

Раскрыть/свернуть массивы

Если в проекте присутствует переменная типа массив можно настроить отображение элементов массива в таблице переменных. Массив отображается в таблице переменных в виде дерева массива. Для раскрытия дерева массива у всех переменных типа массив в таблице переменных нажмите кнопку **Раскрыть все массивы** в верхней части таблицы. Данная настройка сохраняется при следующем открытии таблицы переменных, в рамках одной сессии. Для отображения элементов массива у одной переменной типа массив нажмите на частную кнопку "+", расположенную перед именем массива.

Чтобы свернуть дерево массива у всех переменных типа массив в таблице переменных нажмите кнопку **Свернуть все массивы** в верхней части таблицы. Данная настройка сохраняется при следующем открытии таблицы переменных, в рамках одной сессии. Для скрытия отображения элементов массива у одной переменной типа массив нажмите на частную кнопку "-", расположенную перед именем массива.

Поиск переменных

Таблица переменных поддерживает поиск и фильтрацию по имени. Символ «Пробел» выполняет роль логического ИЛИ и позволяет фильтровать переменные по нескольким критериям.

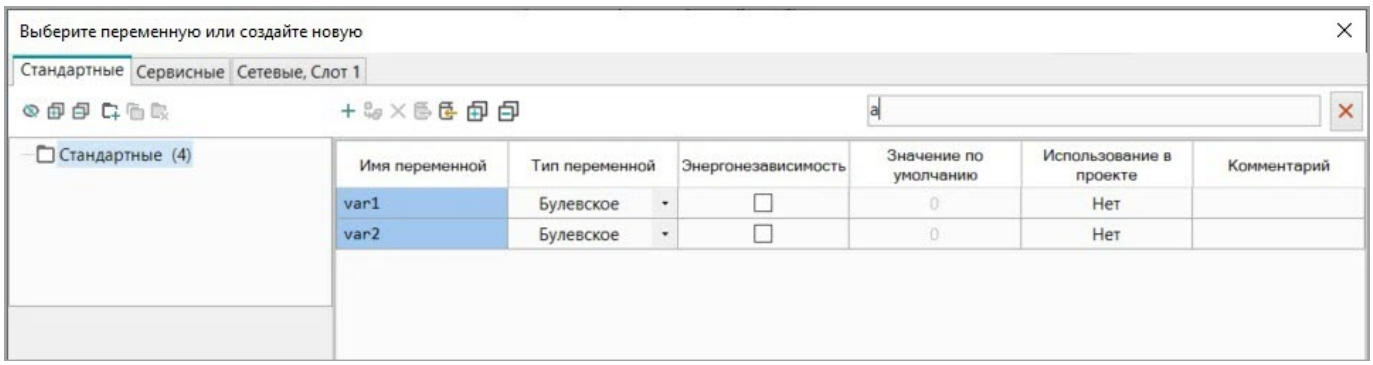


Рисунок 6.5 – Поиск переменных

Если введенной в строке поиска переменной не существует, то её можно создать нажатием на кнопку + в верхней части таблицы. Переменной автоматически присвоится введенное в строке поиска имя.

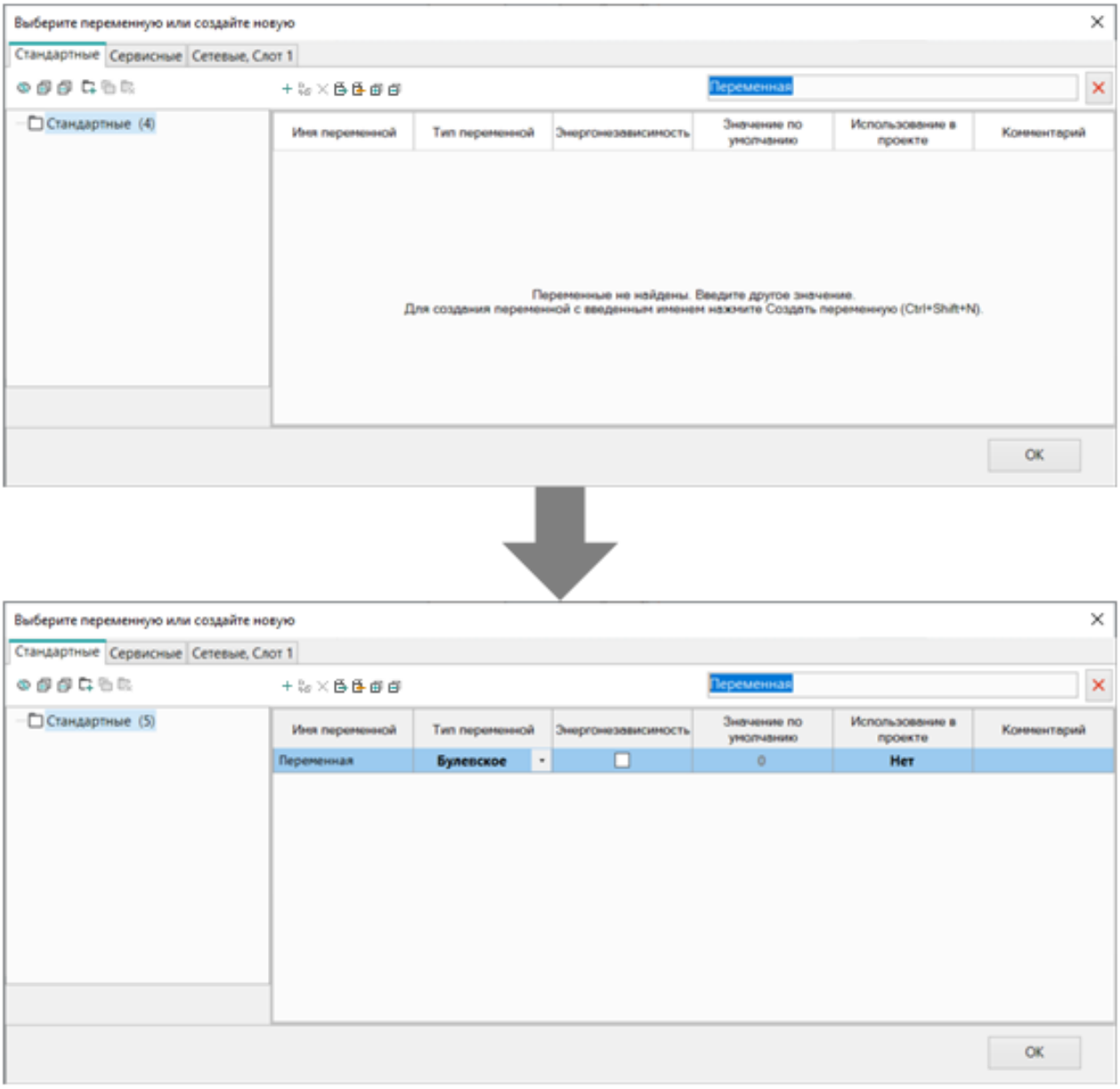


Рисунок 6.6 – Создание переменной с заданным именем

Контекстное меню таблицы переменных

Чтобы найти места использования переменной в проекте нажмите ПКМ на строку переменной в таблице и выберите в контекстном меню переменной **Показать ссылки**.

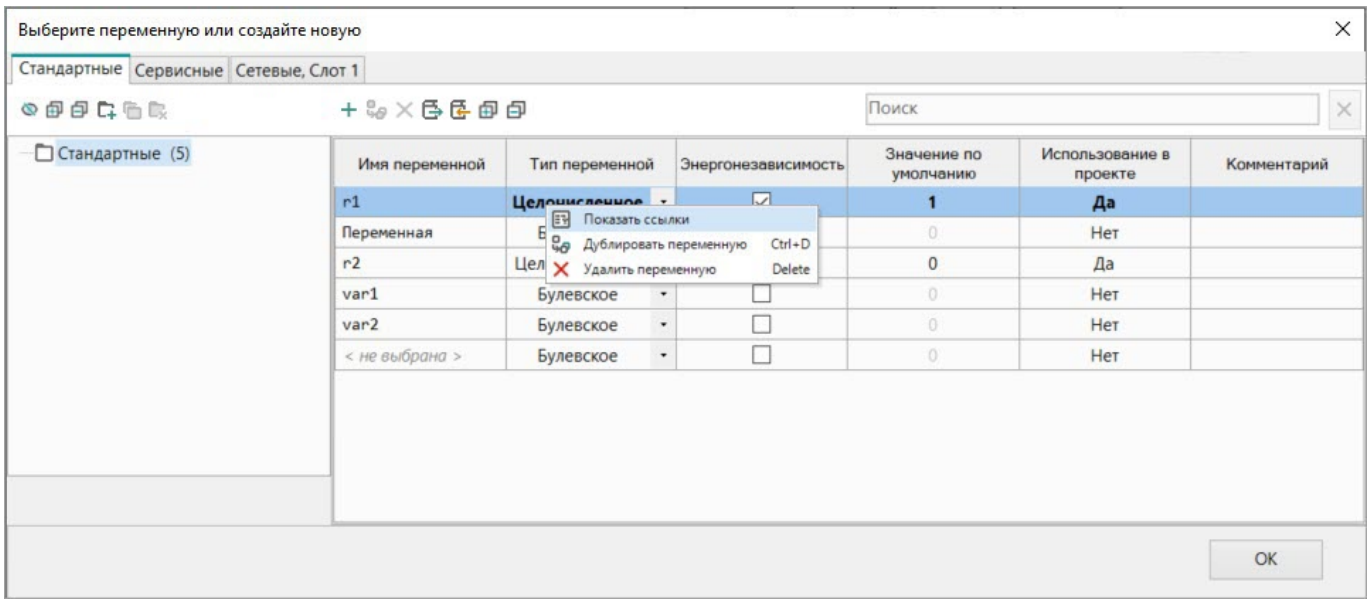


Рисунок 6.7

В открывшемся окне отобразится информация о выбранной переменной.

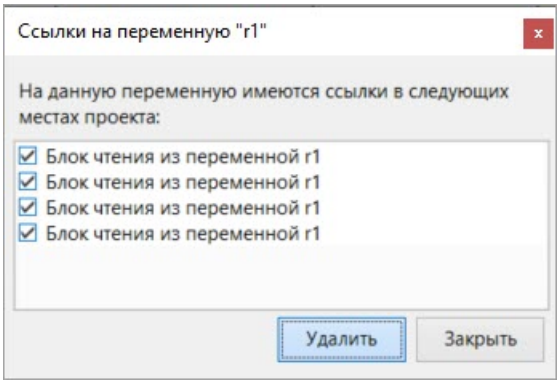


Рисунок 6.8 – Ссылки на переменную

Для переменной типа массив указываются ссылки как на сам массив, так и на его элементы.

Чтобы удалить привязку к блоку переменной в проекте отметьте нужные переменные галочками и нажмите кнопку **Удалить**. Переменная останется в таблице, но не будет использоваться в проекте.

6.1 Типы переменных

В Owen Logic используются три типа переменных:

- булевский (двоичный);
- целочисленный;
- с плавающей запятой (вещественный);
- массив.

Значения от одной переменной к другой могут передаваться только при совпадающих типах переменных.

Булевский тип

Переменная булевского типа может принимать одно из двух значений: 1 (True) или 0 (False).

На схеме переменные булевского типа соединяются черными линиями:

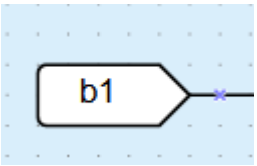


Рисунок 6.9

Целочисленный тип

Переменная целочисленного типа может принимать значение целого числа в диапазоне от 0 до 4294967295.

На схеме переменные целочисленного типа соединяются красными линиями:

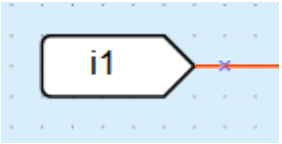


Рисунок 6.10

Тип с плавающей запятой

Переменная типа с плавающей запятой может принимать значение вещественного числа в диапазоне от –3,402823E+38 до 3,402823E+38.

Числа одинарной точности с плавающей запятой обеспечивают относительную точность 7–8 десятичных цифр в диапазоне от 10^{-38} до примерно 10^{38}.

На схеме переменные типа с плавающей запятой соединяются фиолетовыми линиями:

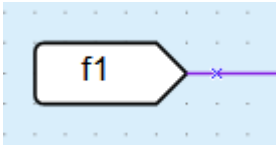


Рисунок 6.11

Массив

Переменная типа массив хранит упорядоченный набор элементов одного типа: булевского, целочисленного или с плавающей запятой. Максимальное количество элементов в одном массиве - 256.

ПРИМЕЧАНИЕ
Работа с массивами доступна в Owen Logic начиная с версии 3.0. В текущей реализации поддерживается работа только с одномерными массивами.

На схеме переменные типа массив соединяются зелеными линиями:

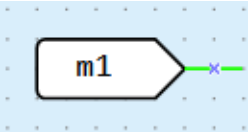


Рисунок 6.12

6.2 Сервисные переменные

Сервисные переменные связаны с настройкой прибора и могут только считываться. Запись в сервисные переменные недоступна. Набор сервисных переменных может отличаться в зависимости от прибора.

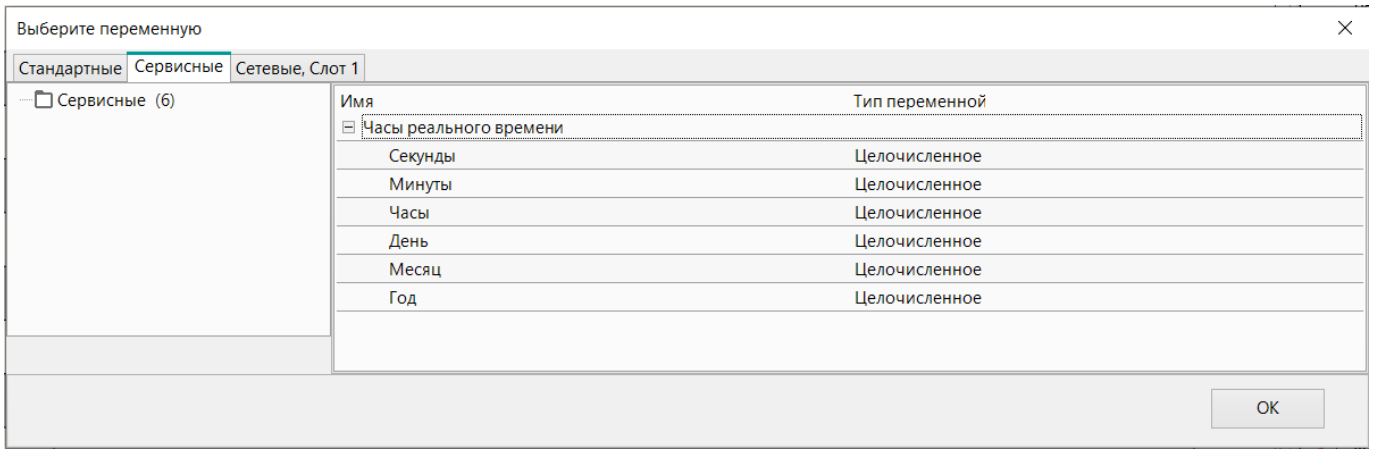


Рисунок 6.13 – Вкладка Сервисные переменные

На схеме сервисные переменные обозначаются серым цветом:

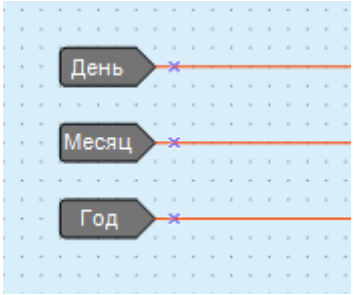


Рисунок 6.14

Сервисные переменный для приборов второго поколения

В приборах второго поколения нет сервисных переменных. Вместо сервисных переменных используется привязка переменных к параметрам прибора (см. раздел Привязка переменных к параметрам).

6.3 Переменные интерфейса связи

В таблице переменных для каждого интерфейса связи создается отдельная вкладка с описанием сетевых переменных. В заголовке вкладки указывается тип интерфейса и номер занимаемого слота.

Режим Master

Для интерфейса связи в режиме Master таблица содержит вкладки с сетевыми переменными для каждого опрашиваемого устройства.

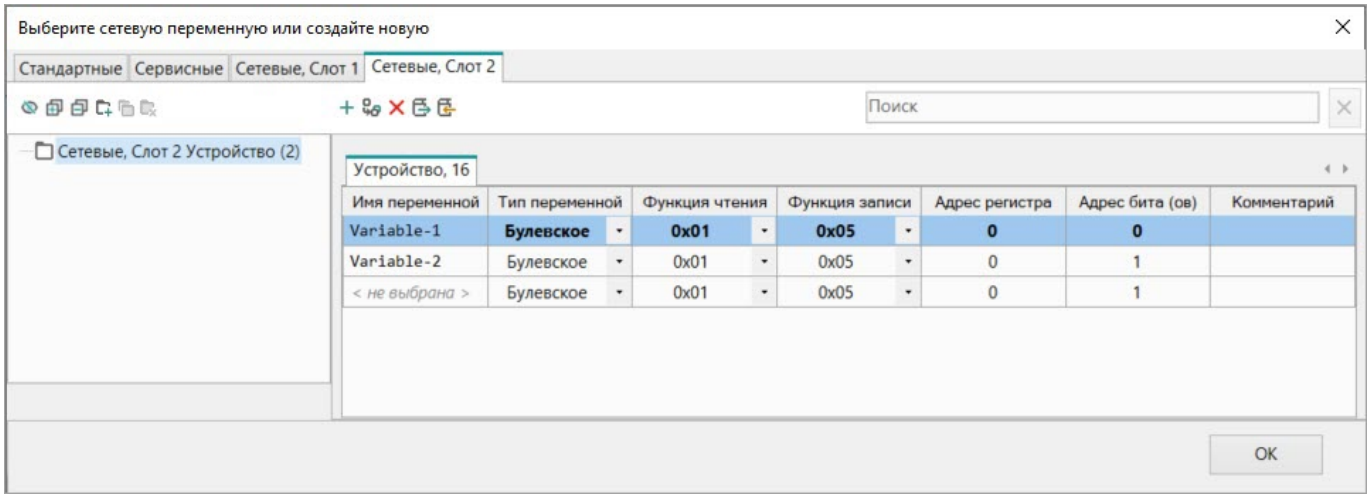


Рисунок 6.15

О настройках переменных см. подробнее в разделе Режим Master.

Режим Slave

Для интерфейса связи в режиме Slave таблица отображает считываемые сетевые переменные.

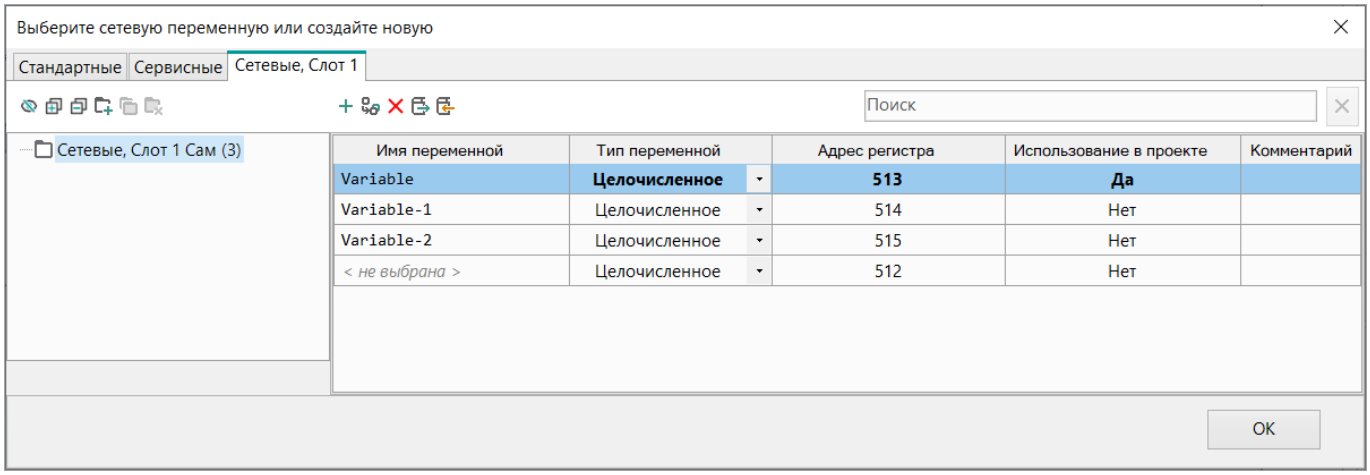


Рисунок 6.16

О настройках переменных см. подробнее в разделе Режим Slave.

6.4 Привязка переменных к параметрам

ПРИМЕЧАНИЕ
Привязка переменных к параметрам доступна только для **приборов второго поколения**. Список приборов второго поколения см. в разделе О программе.

В приборах второго поколения можно привязать переменные к параметрам прибора и использовать в программе. Переменные, привязанные к параметрам, доступны для чтения и изменения.

Для привязки переменной к параметру прибора следует:

1. Открыть **Настройки прибора**.

   ПРИМЕЧАНИЕ
   Привязка переменных доступна не для всех параметров.

2. Выбрать параметр для привязки и нажать на кнопку «…» в графе **Переменная** окна настроек.

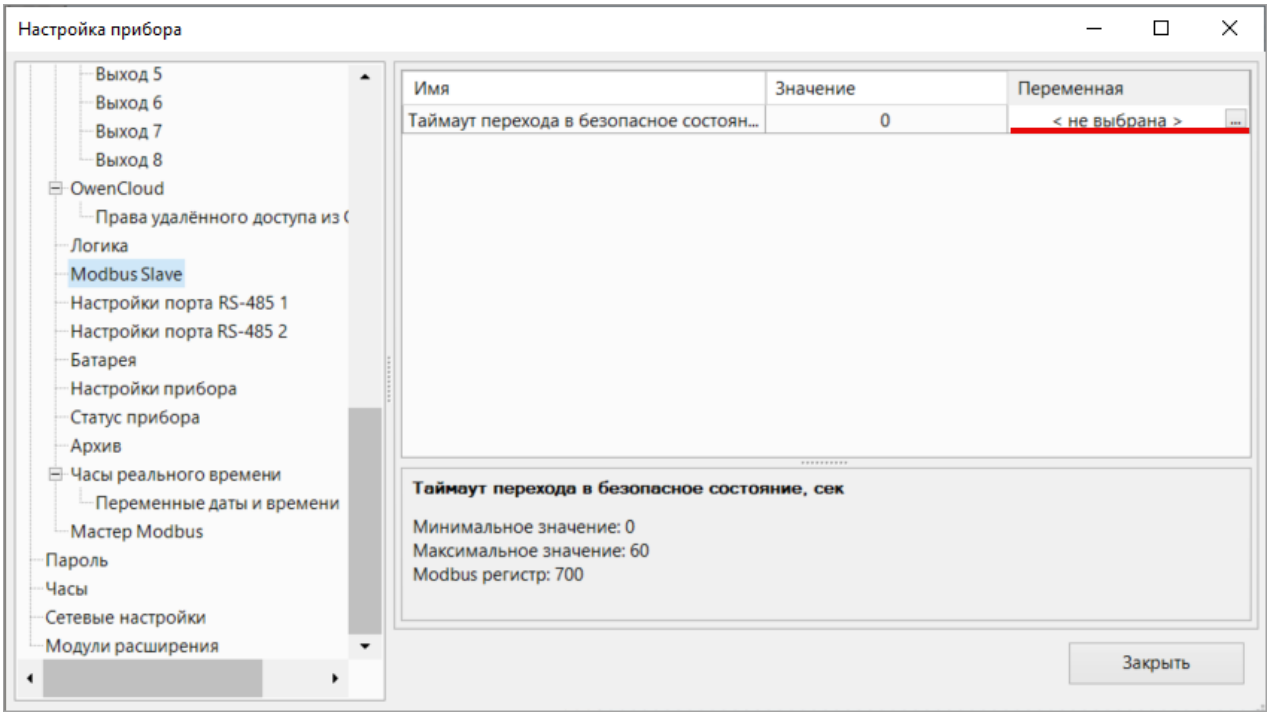


Рисунок 6.17

3. В открывшемся окне Переменных выбрать или создать переменную для привязки. Выделенные серым переменные доступны только для чтения.

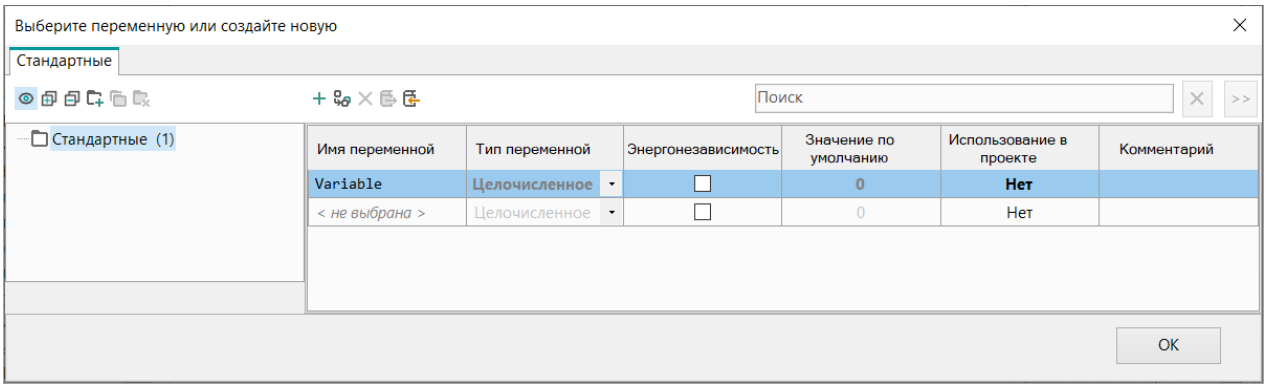


Рисунок 6.18

Имя привязанной переменной будет отображаться в графе «Переменная» напротив параметра.

После привязки к параметрам переменные можно использовать в программе. Таким образом, значения параметров можно читать и изменять в программе.

ПРИМЕЧАНИЕ
Если привязанная к параметру переменная используется для записи с помощью блока, а значение параметра не изменилось, то записанное через блок значение будет храниться в переменной один цикл. В следующем цикле значение переменной будет прочитано из параметра.

6.5 Копирование переменных

Переменные можно копировать из проекта в проект для повторного использования и сокращения времени на разработку. Для копирования переменной следует выделить блок с переменной в исходном проекте и нажать кнопку на панели инструментов или выбрать команду **Копировать** в контекстном меню блока. Переменная вставляется в другой проект нажатием на кнопку на панели инструментов или с помощью выбора команды **Вставить** в контекстном меню холста.

Переменные

Во время вставки скопированного блока переменной в проект проверяются все свойства переменных. Если в проекте есть переменная с полностью совпадающими свойствами, то она дублируется в блок. Переменные с отличающимися свойствами автоматически создаются в проекте.

Копирование элемента массива из одного проекта в другой происходит по следующим правилам:

- Если в проекте, в который выполняется вставка, отсутствует массив со свойствами (тип, размер) идентичными свойствам массива, которому принадлежит скопированный элемент, то вместе с элементом создаётся новый массив, полностью повторяющий свойства исходного.
- Если в проекте, в который выполняется вставка, уже существует массив с идентичными свойствами, то создание нового массива не происходит.

Правила копирования переменных типа массив не отличаются от правил копирования стандартных переменных.

Если имена переменных совпадают, но другие свойства отличаются, то создается новая переменная, но конфликт имен отображен в таблице переменных, и требуется сменить имя одной из переменных.

ПРИМЕЧАНИЕ
В проект для прибора, который не поддерживает тип данных с плавающей запятой, невозможно вставить переменные типа с плавающей запятой.

ПРИМЕЧАНИЕ
Энергонезависимые переменные невозможно вставить в проект для прибора, который их не поддерживает.

Сервисные переменные

В проект для прибора, который не поддерживает часы реального времени, невозможно скопировать сервисные переменные.

Переменные интерфейса связи

В другой проект можно скопировать только переменные интерфейса связи в режиме Slave для идентичного номера слота. Переменные интерфейса связи в режиме Master следует создавать вручную. Во время вставки скопированного блока сетевой переменной в проект проверяются все свойства переменных. Если в проекте есть переменная с полностью совпадающими свойствами, то она дублируется в блок. Переменные с отличающимися свойствами автоматически создаются в проекте. Если имя или регистры переменных совпадают, но другие свойства отличаются, то создается новая переменная, но возникает конфликт регистров или имен. Конфликт можно устранить в таблице переменных, задав другое имя или регистры.

7 Библиотека

ВНИМАНИЕ
Для разных моделей приборов список доступных элементов библиотеки компонентов может отличаться.

В открытом проекте на панели Библиотека компонентов 3.3 доступны разделы с компонентами для разработки программы 4.3.

Разделы библиотеки компонентов:

- Функции 7.1 служат для выполнения логических, арифметических операций и др. Функции не сохраняют состояния своих внутренних переменных. Число используемых функций ограничено только объемом памяти прибора;
- Функциональные блоки (ФБ) 7.2 служат для выполнения операций, требующих сохранения состояния своих внутренних переменных. К ФБ относятся триггеры, счетчики, таймеры и другие. Каждая модель прибора имеет собственное ограничение на количество используемых ФБ (см. *руководство по эксплуатации* прибора);
- раздел Макросы проекта 7.3 содержит созданные пользователем или загруженные с помощью менеджера компонентов 4.10 макросы;
- Элементы управления 7.6 служат для программирования дисплея прибора 4.4 и отображаются на панели во время работы с прибором с монохромным текстовым ЖКИ 4.4.1.

7.1 Функции

- Логические функции 7.1.1;
- Арифметические функции 7.1.2;
- Функции сравнения 7.1.3;
- Сдвиговые функции 7.1.4;
- Битовые функции 7.1.4.

7.1.1 Логические функции

- И (AND) 7.1.1.1
- ИЛИ (OR) 7.1.1.2
- НЕ (NOT) 7.1.1.3
- Исключающее ИЛИ (XOR) 7.1.1.4

Особенностью работы блоков логических функций является их самонастройка на тип данных. Если к входу блока была подсоединена целочисленная переменная, то блок автоматически перестраивается на работу с целочисленными значениями.

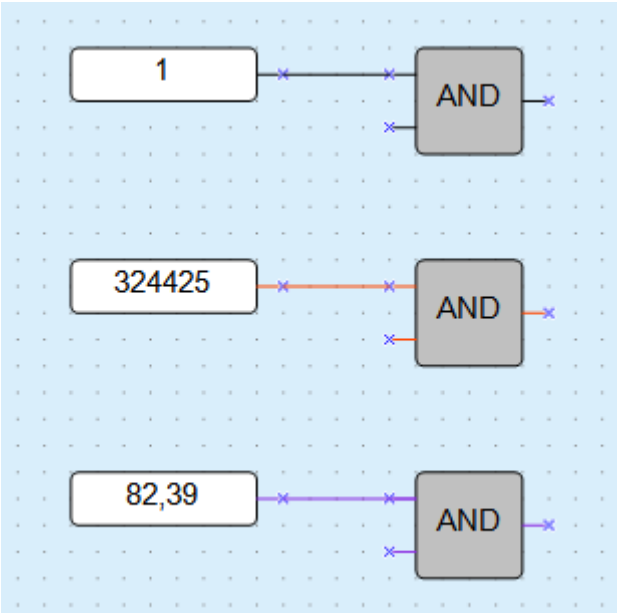


Рисунок 7.1

Для функций **И** и **ИЛИ** следует учитывать, что неподключенные входы блоков будут иметь следующие состояния:

- для функции **И** – логическая «1»;

- для функции **ИЛИ** – логический «0».

В этом случае блоки выполняют функцию повторителя сигнала. Для увеличения числа входов у логических функций используется их каскадное включение:

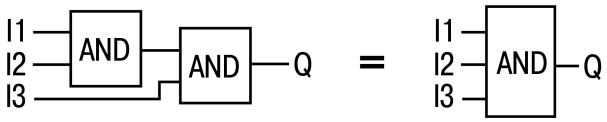


Рисунок 7.2

7.1.1.1 И (AND)

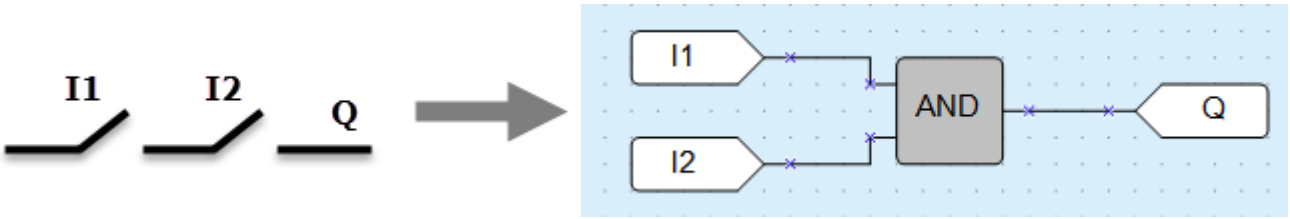


Рисунок 7.3

Если на оба входа блока функции поступают сигналы логической «1» (все входы включены – контакты замкнуты), то на выходе появляется сигнал логической «1» (выход включен).

Работе функции соответствует таблица состояний:

I1	I2	Q
0	0	0
0	1	0
1	0	0
1	1	1

Если на входы блока функции поступают целочисленные значения, то операция будет произведена над каждым битом значения в отдельности.

И	0011
	0101
	0001

7.1.1.2 ИЛИ (OR)

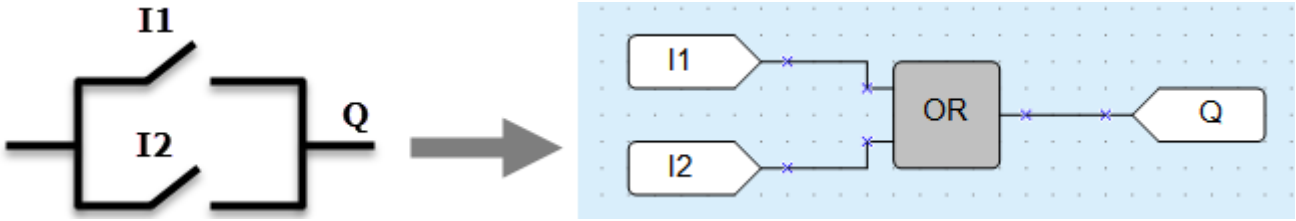


Рисунок 7.4

Если на один из входов блока функции поступает сигнал логической «1» (контакты замкнуты), то на выходе элемента появляется логическая «1» (выход включен).

Работе функции соответствует таблица состояний:

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Если на входы блока функции поступают целочисленные значения, то операция будет произведена над каждым битом значения в отдельности.

ИЛИ	0011
	0101
	0111

7.1.1.3 НЕ (NOT)

Функция используется для инвертирования сигнала.

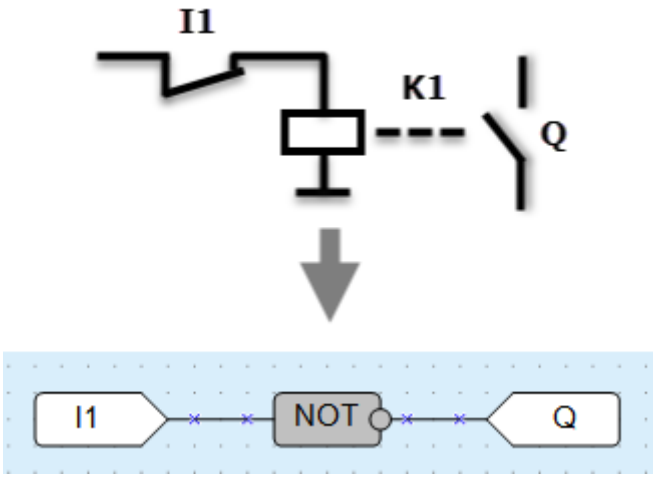


Рисунок 7.5

Если на вход блока функции поступает сигнал логического «0» (контакты разомкнуты), то на выходе элемента появляется логическая «1» (выход включен), и наоборот: сигнал инвертируется.

Работе функции соответствует таблица состояний:

I1	Q
0	1
1	0

Если на вход блока функции поступает целочисленное значение, то операция будет произведена над каждым битом значения в отдельности.

НЕ	01
	10

7.1.1.4 Исключающее ИЛИ (XOR)

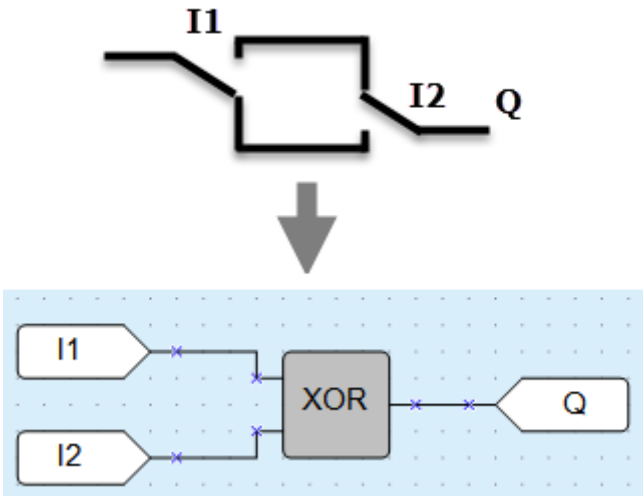


Рисунок 7.6

Если только на один из входов блока функции поступает логическая «1», то на выходе элемента появляется логическая «1» (выход включен).

Работе функции соответствует таблица состояний:

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	0

Если на входы функции поступают целочисленные значения, то операция будет произведена над каждым битом значения в отдельности.

XOR	0011
	0101
	0110

7.1.2 Арифметические функции

Операции **арифметических функций** могут производиться только над значениями целочисленного типа и типа с плавающей запятой.

Название	Целочисленный тип	Тип с плавающей запятой
Сложение	ADD 7.1.2.1	fADD 7.1.2.1
Вычитание	SUB 7.1.2.2	fSUB 7.1.2.2
Умножение	MUL 7.1.2.3	fMUL 7.1.2.3
Деление	DIV 7.1.2.4	fDIV 7.1.2.4
Деление с остатком	MOD 7.1.2.5	–
Возведение в степень	–	fPOW 7.1.2.6
Модуль от числа	–	fABS 7.1.2.7

7.1.2.1 Сложение (ADD, fADD)

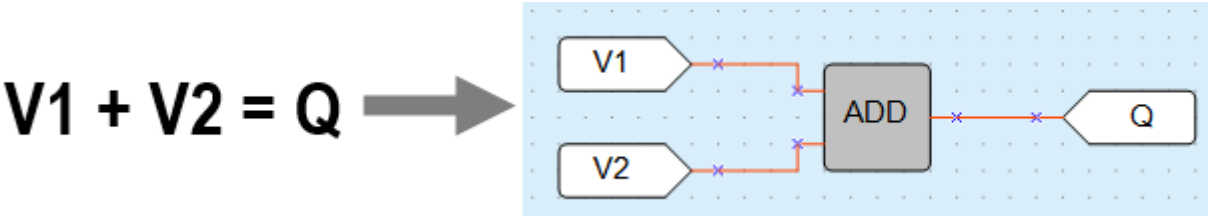


Рисунок 7.7

Для операций над целочисленными значениями используется блок **ADD**, над значениями с плавающей запятой — **fADD**.

Результатом операции функции на выходе является сумма входных значений.

Если во время выполнения операции значение числа получается больше 4294967295 (32 бита), то биты, выходящие за разрядность 32 бита, отсекаются.

Пример

Применение блоков функции для сложения чисел 24 + 2 + 2 + 7 = Q = 35.

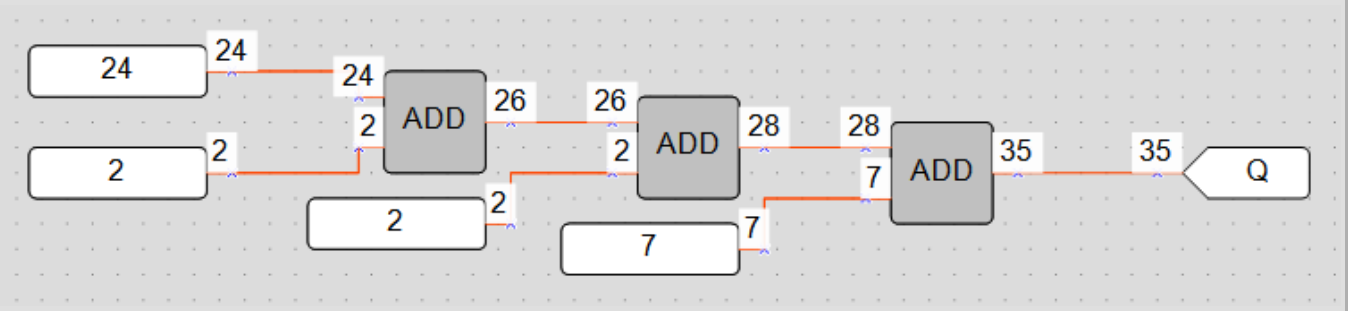


Рисунок 7.8

7.1.2.2 Вычитание (SUB, fSUB)

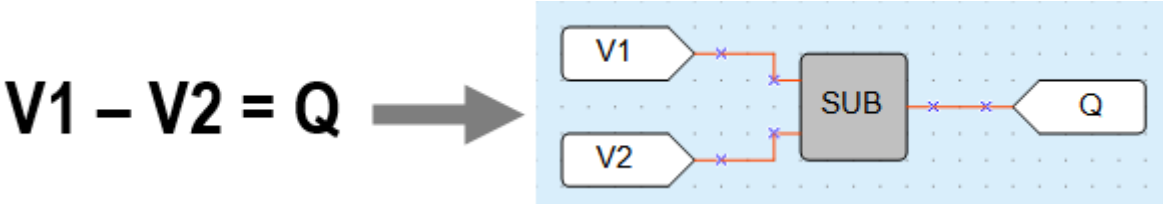


Рисунок 7.9

Для операций над целочисленными значениями используется блок **SUB**, над значениями с плавающей запятой — **fSUB**.

Результатом операции функции на выходе является разность входных значений.

Если во время выполнения операции функции на входе **V1** значение числа меньше значения на входе **V2**, то результатом будет число, полученное сложением младшего числа плюс 0x100000000 (4294967296) минус значение числа большего: **(V1 + 0x100000000) – V2 = Q**.

Пример

Применение блока функции для вычитания чисел 24 – 2 = 22:

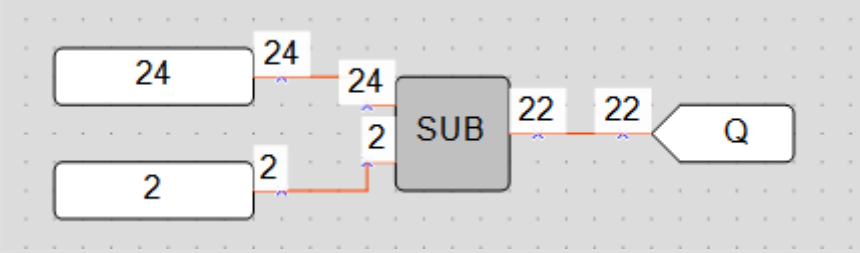


Рисунок 7.10

Применения блока функции для вычитания чисел, где V1 меньше V2: 2 – 24 = Q = 4294967274.

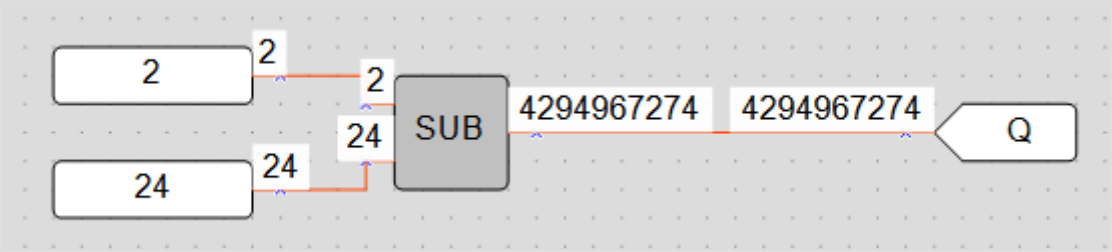


Рисунок 7.11

7.1.2.3 Умножение (MUL, fMUL)

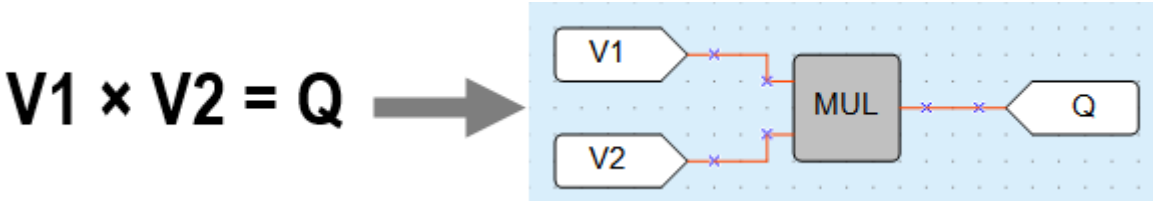


Рисунок 7.12

Для операций над целочисленными значениями используется блок **MUL**, над значениями с плавающей запятой — **fMUL**.

Результатом операции функции на выходе является произведение входных значений.

Если во время выполнения операции функции значение числа получается больше 4294967295 (32 бита), то биты, выходящие за разрядность 32 бита, отсекаются.

Пример

Применение блоков функции для перемножения чисел 24 · 2 · 2· 7 = Q = 672.

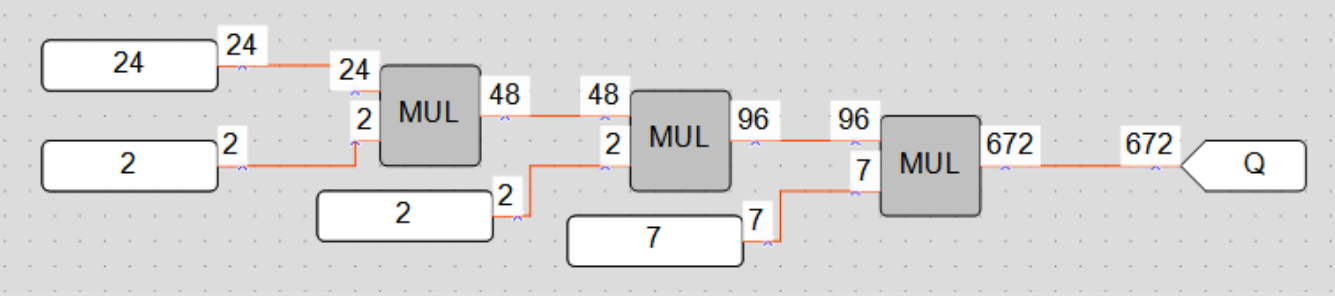


Рисунок 7.13

7.1.2.4 Деление (DIV, fDIV)

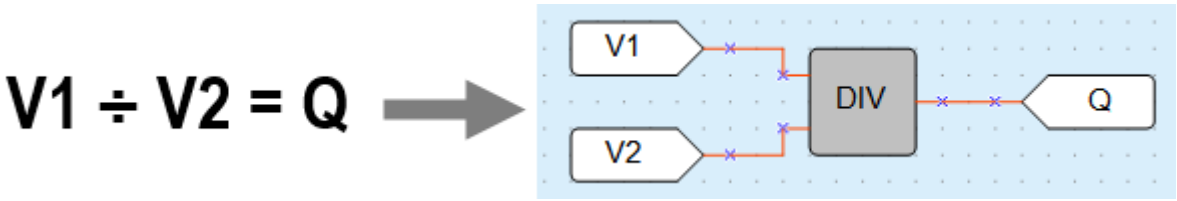


Рисунок 7.14

Для операций над целочисленными значениями используется блок **DIV**, над значениями с плавающей запятой — **fDIV**.

Результатом операции функции на выходе является частное от деления входных значений.

Если во время использования блока DIV в результате деления получаются доли целого числа, то на выходе производится округление значения до целого числа в меньшую сторону.

В случае деления на 0 на выходе элемента будет значение 0.

Пример

Применение блока функции для деления чисел 24 ÷ 2 = Q = 12

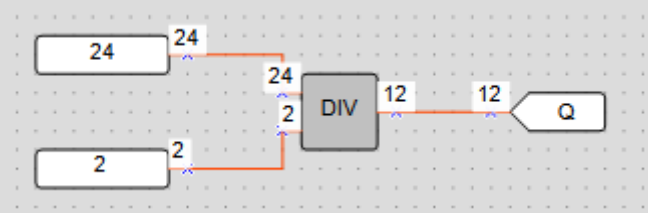


Рисунок 7.15

7.1.2.5 Деление с остатком (MOD)

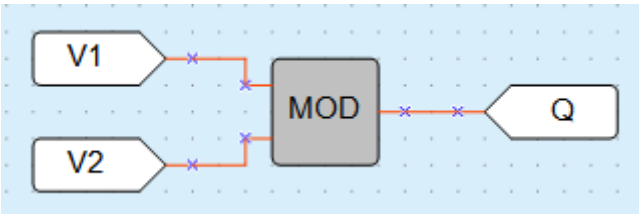


Рисунок 7.16

Результатом операции функции на выходе является остаток от деления входных целочисленных значений.

В случае деления на 0 на выходе блока функции будет значение 0.

Пример

Применение блока функции для выделения целого остатка от деления числа 22 на 3.

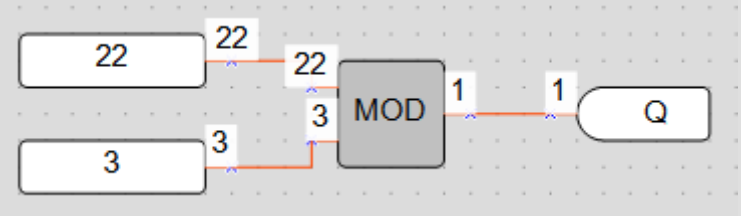


Рисунок 7.17

7.1.2.6 Возведение числа в степень (fPOW)

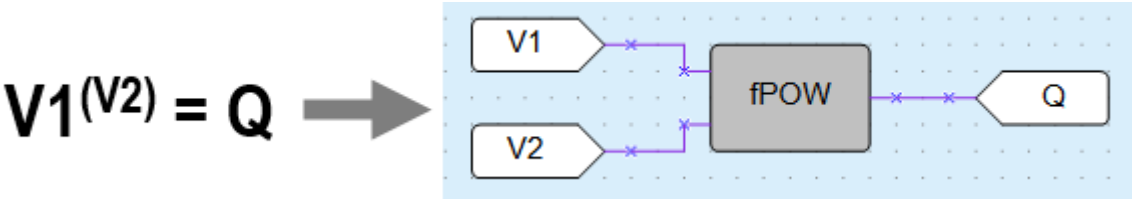


Рисунок 7.18

Результатом операции функции на выходе является переменная типа float, равная числу V1, возведенному в степень V2.

Блок работает только со значениями с плавающей запятой.

Пример

Применение блока функции для возведения числа 3 в степень числа 4.

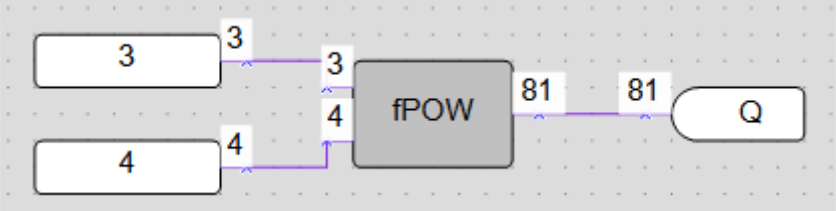


Рисунок 7.19

7.1.2.7 Взятие модуля от числа (fABS)

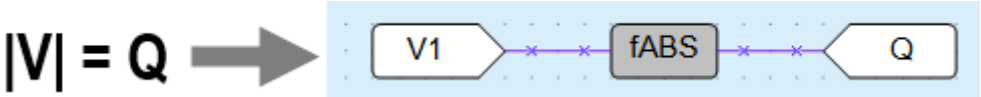


Рисунок 7.20

Блок работает только со значениями с плавающей запятой.

Результатом операции функции на выходе является модуль подаваемого на вход значения.

Пример

Применение блоков функции для определения модулей чисел: |–3,4| = Q = |–3,4|, |6,7| = Q = 6,7

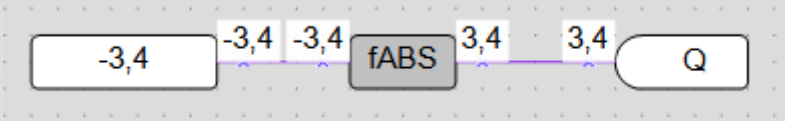


Рисунок 7.21

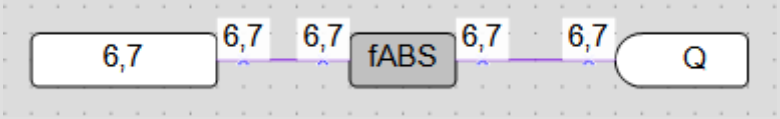


Рисунок 7.22

7.1.3 Функции сравнения

- Равно (EQ) 7.1.3.1;
- Больше (GT, fGT) 7.1.3.2;
- Выбор (SEL) 7.1.3.3.

7.1.3.1 Равно (EQ)

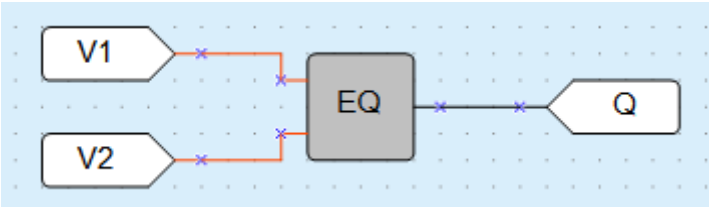


Рисунок 7.23

Результаты выполнения операции блока на выходе сравнения входных значений:

- V1 = V2 → Q = 1;
- V1 > V2 → Q = 0;
- V1 < V2 → Q = 0.

Пример

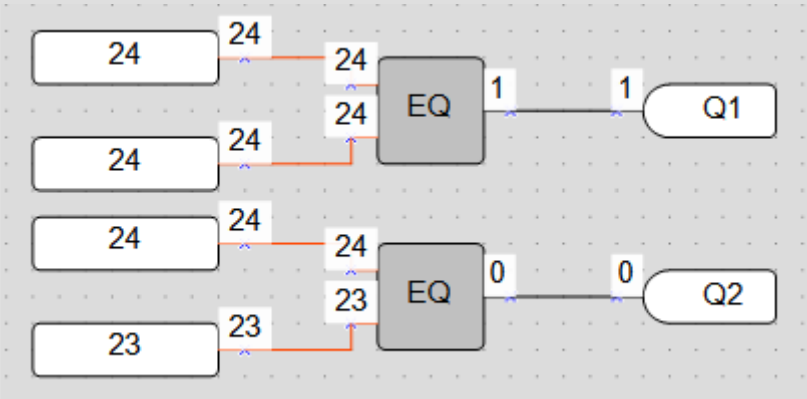


Рисунок 7.24

7.1.3.2 Больше (GT, fGT)

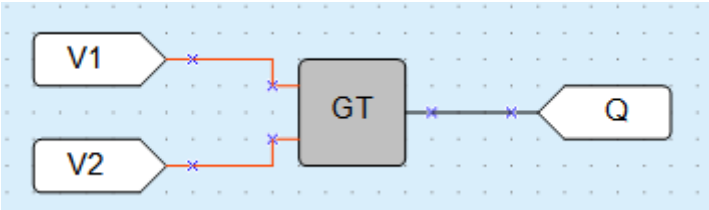


Рисунок 7.25

Результаты выполнения операции блока на выходе при сравнении входных значений:

- V1 = V2 → Q = 0;
- V1 > V2 → Q = 1;
- V1 < V2 → Q = 0.

Для сравнения целочисленных значений используется блок **GT**, для значений с плавающей запятой – блок **fGT**.

Пример

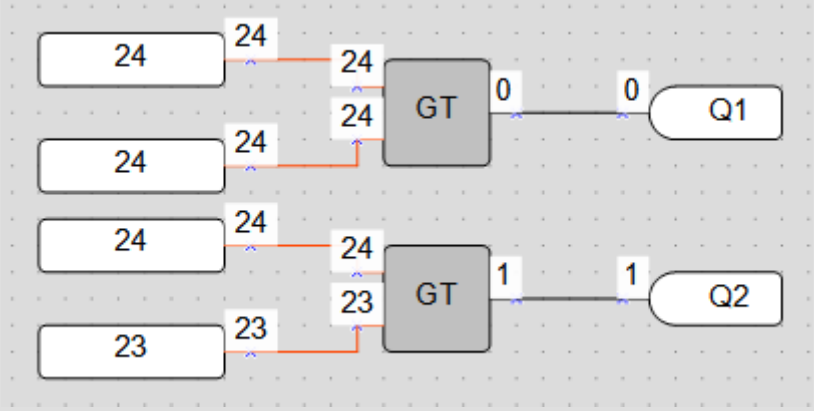


Рисунок 7.26

7.1.3.3 Выбор (SEL, fSEL)

ПРИМЕЧАНИЕ
Функция **Выбор (SEL, fSEL)** в Owen Logic называется **Тернарная условная операция сравнения**.

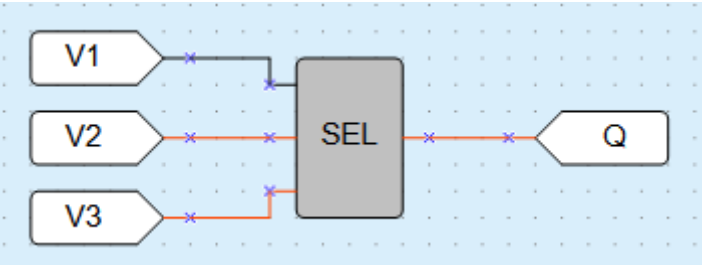


Рисунок 7.27

Если **V1** равен логическому «0», то результатом выполнения операции блока на выходе является входной сигнал **V2**. Если **V1** равен логической «1», то результатом выполнения операции блока на выходе является входной сигнал **V3**.

- V1 = 0 → Q = V2;
- V1 = 1 → Q = V3.

Для выбора целочисленных значений используется блок **SEL**, для значений с плавающей запятой — блок **fSEL**.

Пример

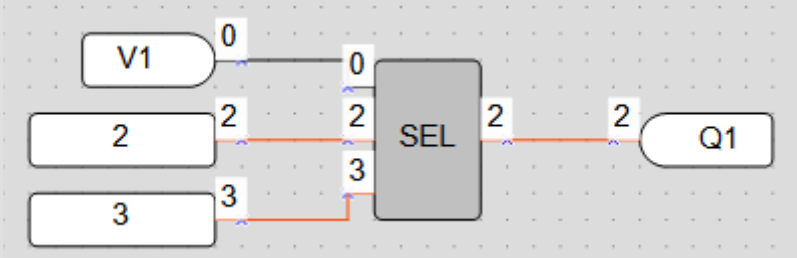


Рисунок 7.28

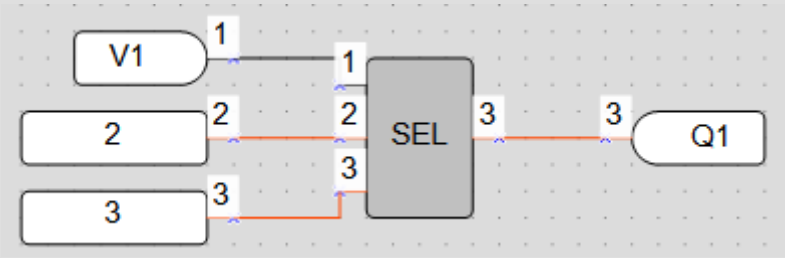


Рисунок 7.29

7.1.4 Сдвиговые функции

- Логический сдвиг влево (SHL) 7.1.4.1;
- Логический сдвиг вправо (SHR) 7.1.4.2.

7.1.4.1 Побитовый логический сдвиг влево (SHL)

Побитовый логический сдвиг влево (SHL) используется для выполнения операции побитового логического сдвига операнда **X** влево на **N** бит с дополнением нулями справа.

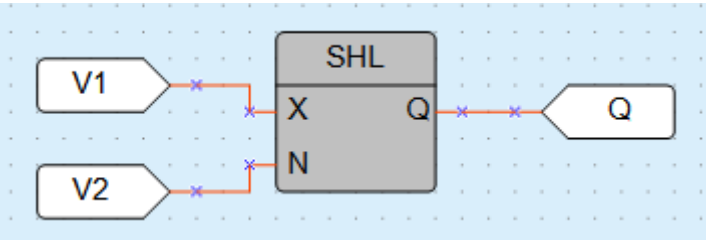


Рисунок 7.30

Пример

Применение для сдвига числа 38 (десятичное) = 00100110 (двоичное).

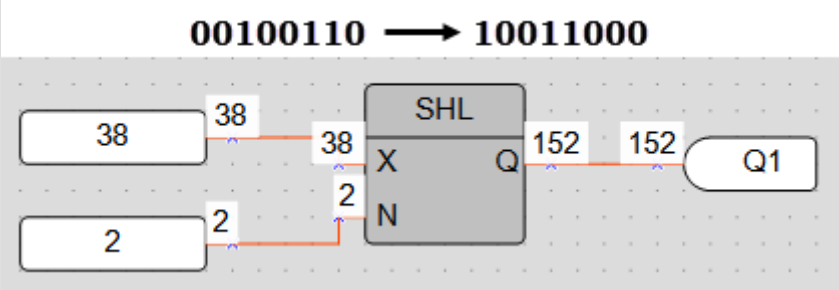


Рисунок 7.31

7.1.4.2 Побитовый логический сдвиг вправо (SHR)

Побитовый логический сдвиг вправо (SHR) используется для выполнения операции побитового логического сдвига операнда **X** вправо на **N** бит с дополнением нулями слева.

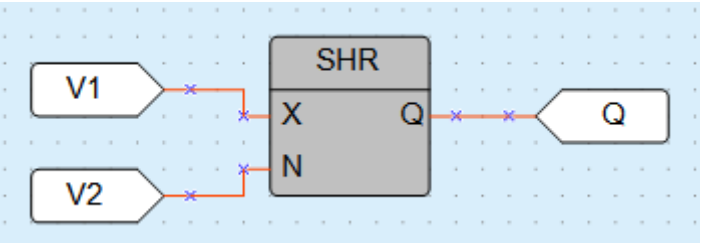


Рисунок 7.32

Пример

Применение для сдвига числа 152 (десятичное) = 10011000 (двоичное).

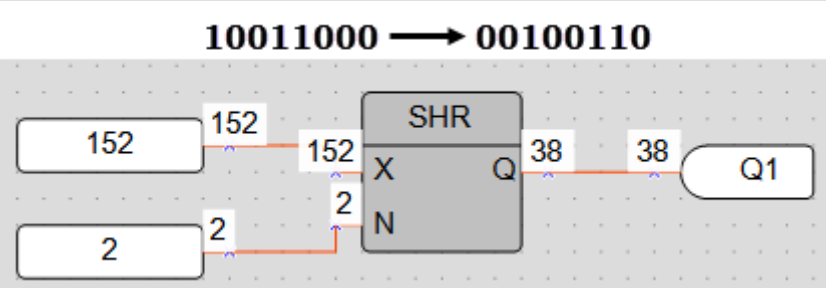


Рисунок 7.33

7.1.5 Битовые функции

- Чтение бита (EXTRACT) 7.1.5.1;
- Запись бита (PUTBIT) 7.1.5.2;
- Дешифратор (DC32) 7.1.5.3;
- Шифратор (CD32) 7.1.5.4.

7.1.5.1 Чтение бита (EXTRACT)

Чтение бита (EXTRACT) используется для выполнения операции чтения значения бита **N** в числе на входе **X**. Биты нумеруются с конца. Число на входе **X** задается в десятичной системе. Выходное значение **Q** всегда двоичное (0 или 1).

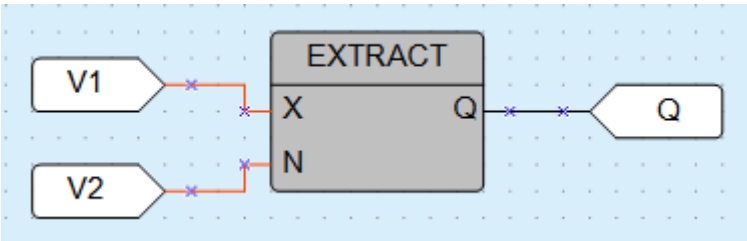


Рисунок 7.34

Пример

Применения блока для чтения пятого бита из числа 81 (десятичное) = 1010001 (двоичное).

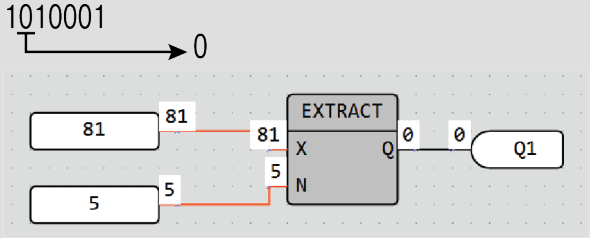


Рисунок 7.35

7.1.5.2 Запись бита (PUTBIT)

Запись бита (PUTBIT) используется для выполнения операции записи в числе **X** значения бита **N** в состояние, указанное на входе **B** (логические «0» или «1»). Число на входе X задается в десятичной системе. Выходное значение **Q** – целочисленное.

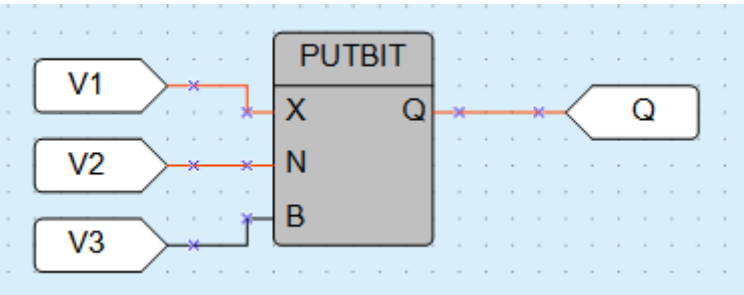


Рисунок 7.36

Пример

Применения записи четвертого бита в сигнале логической «1» для числа 38 (десятичное) = 100110 (двоичное).

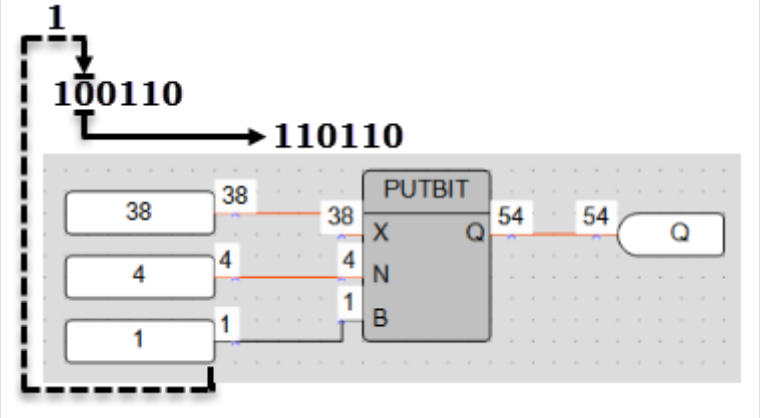


Рисунок 7.37

7.1.5.3 Дешифратор (DC32)

Дешифратор (DC32) используется для выполнения операции преобразования двоичного кода на входе в позиционный код на выходе. Перед выполнением операции над значением на входе предварительно выполняется побитовая логическая операция «И» с операндом 0x1F (11111b).

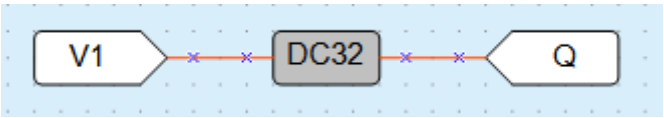


Рисунок 7.38

Работе дешифратора соответствует приведенная таблица состояний:

Двоичный код					Позиционный код								
5	4	3	2	1	32	31		6	5	4	3	2	1
0	0	0	0	0	0	0		0	0	0	0	0	1
0	0	0	0	1	0	0		0	0	0	0	1	0
0	0	0	1	0	0	0		0	0	0	1	0	0
0	0	0	1	1	0	0		0	0	1	0	0	0
		1	0	0	0	0	...		1				
...					...			...					
1	1	1	0	1	0	0		0	0	0	0	0	0
1	1	1	1	0	0	1		0	0	0	0	0	0
1	1	1	1	1	1	0		0	0	0	0	0	0

Рисунок 7.39

Пример

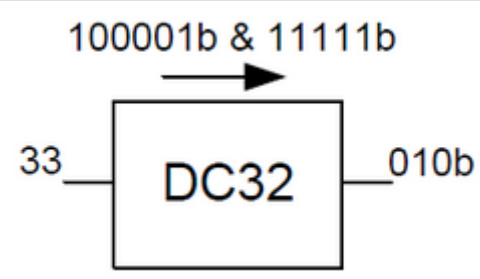


Рисунок 7.40

7.1.5.4 Шифратор (CD32)

Шифратор (CD32) используется для выполнения операции преобразования позиционного кода на входе в двоичный код на выходе.

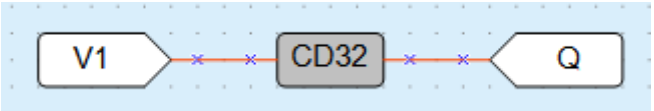


Рисунок 7.41

Если входное двоичное значение имеет более одной логической «1» в разрядах, то работа шифратора ведется только со старшим единичным разрядом.

Работе шифратора соответствует таблица состояний.

Двоичный код					Позиционный код								
5	4	3	2	1	32	31		6	5	4	3	2	1
0	0	0	0	0	0	0		0	0	0	0	0	1
0	0	0	0	1	0	0		0	0	0	0	1	0
0	0	0	1	0	0	0		0	0	0	1	0	0
0	0	0	1	1	0	0		0	0	1	0	0	0
		1	0	0	0	0	...		1				
...					...			...					
1	1	1	0	1	0	0		0	0	0	0	0	0
1	1	1	1	0	0	1		0	0	0	0	0	0
1	1	1	1	1	1	0		0	0	0	0	0	0

Рисунок 7.42

7.2 Функциональные блоки

- Триггеры;
- Таймеры;
- Генераторы;
- Счетчики;
- Регуляторы.

7.2.1 Триггеры

- RS-триггер с приоритетом выключения (RS) 7.2.1.1;
- SR-триггер с приоритетом включения (SR) 7.2.1.2;
- Детектор переднего фронта импульса (RTRIG) 7.2.1.3;
- Детектор заднего фронта импульса (FTRIG) 7.2.1.4;
- D-триггер (DTRIG) 7.2.1.5.

7.2.1.1 RS-триггер с приоритетом выключения

RS-триггер с приоритетом выключения используется для переключения с фиксацией состояния во время поступления коротких импульсов на соответствующий вход. На выходе **Q** появится логическая «1» по фронту сигнала на входе **S**.

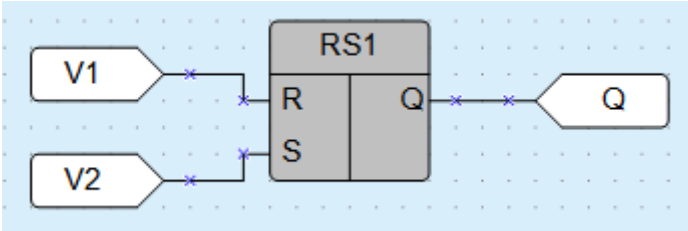


Рисунок 7.43

Работу триггера поясняет приведенная на рисунке ниже диаграмма.

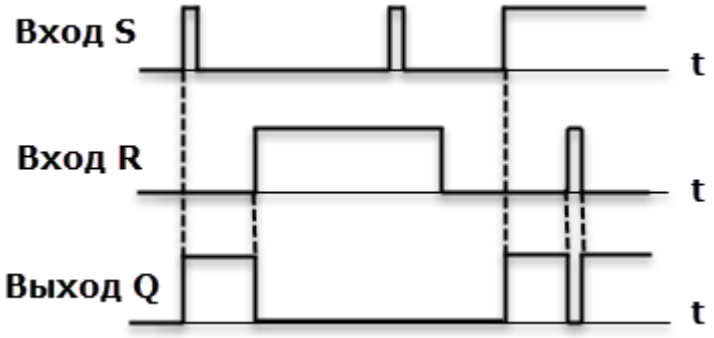


Рисунок 7.44

В случае одновременного поступления сигналов на оба входа приоритетным является сигнал входа **R**.

7.2.1.2 SR-триггер с приоритетом включения

SR-триггер с приоритетом включения используется для переключения с фиксацией состояния в случае поступления коротких импульсов на соответствующий вход. На выходе **Q** появится логическая «1» по фронту сигнала на входе **S**.

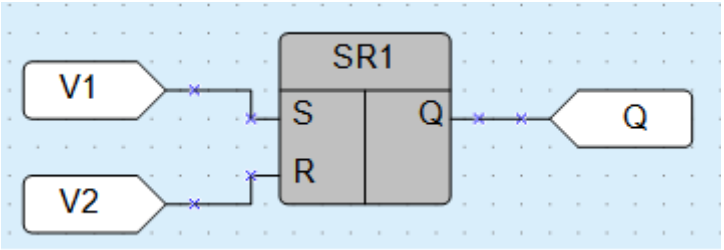


Рисунок 7.45

Работу триггера поясняет приведенная на рисунке ниже диаграмма.

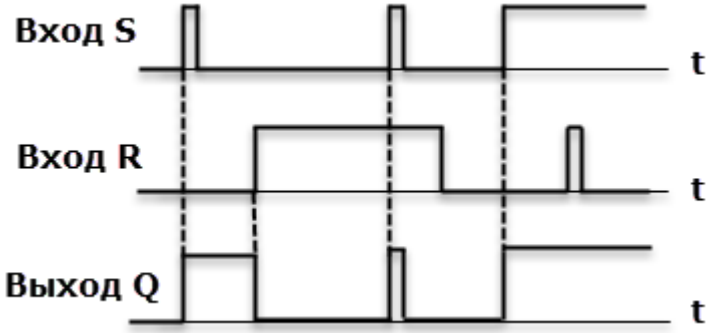


Рисунок 7.46

В случае одновременного поступления сигналов на оба входа приоритетным является сигнал входа **S**.

7.2.1.3 Детектор переднего фронта импульса (RTRIG)

Детектор переднего фронта импульса (RTRIG) используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе **Q** генерируется единичный импульс по переднему фронту входа **I**.

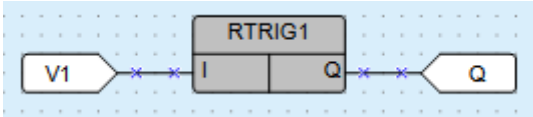


Рисунок 7.47

Работу детектора объясняет приведенная на рисунке ниже диаграмма.

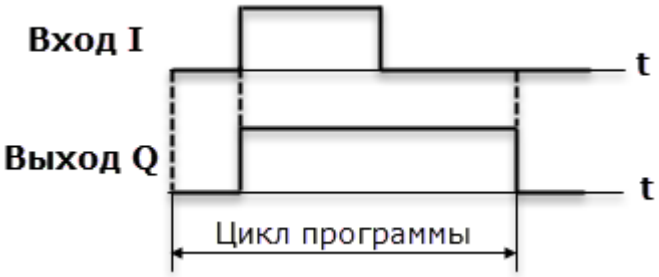


Рисунок 7.48

7.2.1.4 Детектор заднего фронта импульса (FTRIG)

Детектор заднего фронта импульса (FTRIG) используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе **Q** генерируется единичный импульс по заднему фронту входа **I**.

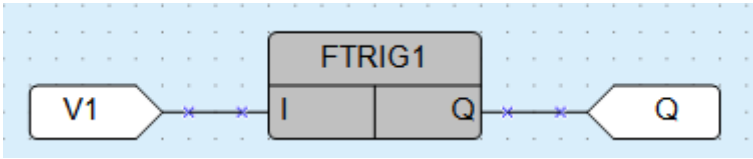


Рисунок 7.49

Работу детектора объясняет приведенная на рисунке ниже диаграмма.

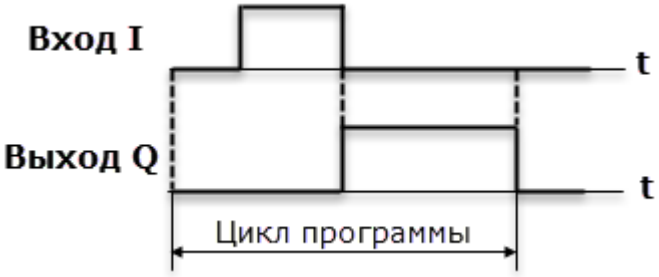


Рисунок 7.50

7.2.1.5 D-триггер (DTRIG)

D-триггер (DTRIG) используется для формирования импульса включения выхода на интервал времени импульса на входе **D**, выходной интервал будет синхронизирован с тактовой частотой на входе **C**.

На выходе **Q** триггера появится сигнал логической «1» по фронту тактовых импульсов на входе **C** при наличии сигнала логической «1» на входе **D**. Возврат выхода **Q** в сигнал логического «0» произойдет по фронту тактовых импульсов на входе **C** при наличии сигнала логического «0» на входе **D**.

Вход **S** принудительно устанавливает выход **Q** в состояние логической «1».

Вход **R** является приоритетным и устанавливает выход **Q** в состояние логического «0».

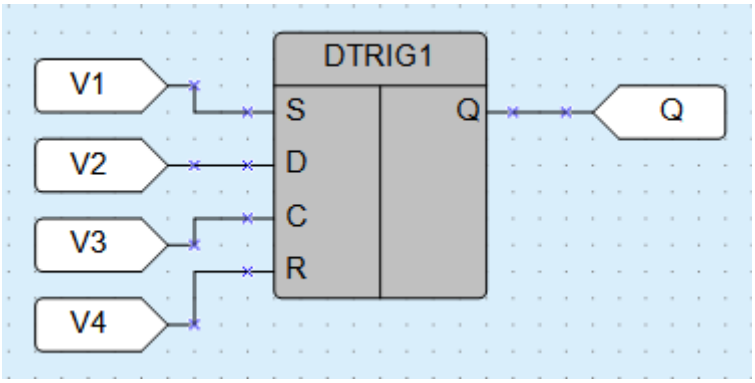


Рисунок 7.51

Работу триггера объясняет приведенная на рисунке ниже диаграмма.

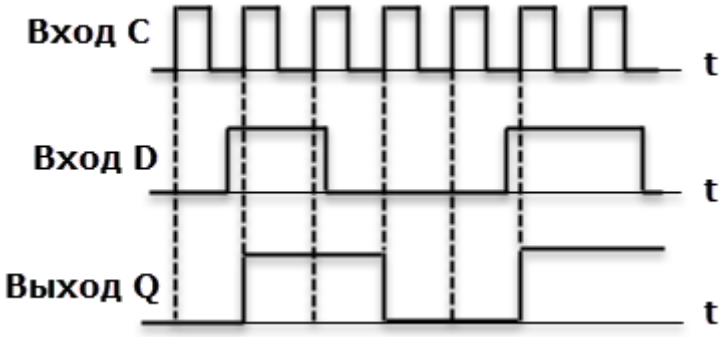


Рисунок 7.52

7.2.2 Таймеры

- Импульс включения заданной длительности (TP);
- Таймер с задержкой включения (TON);
- Таймер с задержкой отключения (TOF);
- Интервальный таймер (CLOCK);
- Интервальный таймер с недельным циклом (CLOCKWEEK).

7.2.2.1 Импульс включения заданной длительности (TP)

Импульс включения заданной длительности (TP) используется для формирования импульса включения выхода на заданный интервал времени. На выходе **Q** блока появляется сигнал логической «1» по фронту входного сигнала **I**. После запуска выход **Q** не реагирует на изменение значения входного сигнала в течение интервала $T_{имп}$. По истечение интервала $T_{имп}$ выходной сигнал сбрасывается в логический «0».

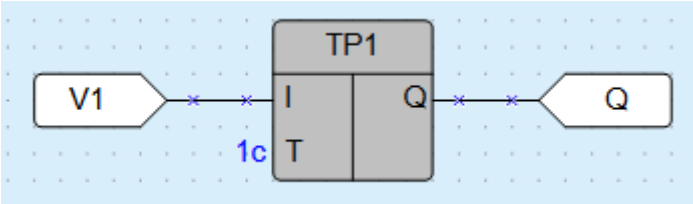


Рисунок 7.53

Работу импульса объясняет приведенная на рисунке ниже диаграмма.

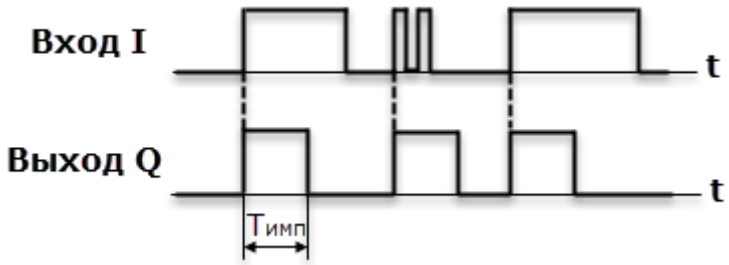


Рисунок 7.54

Допустимый диапазон значений $T^{имп}$ от 0 до 4147200000 мс или 48 дней.

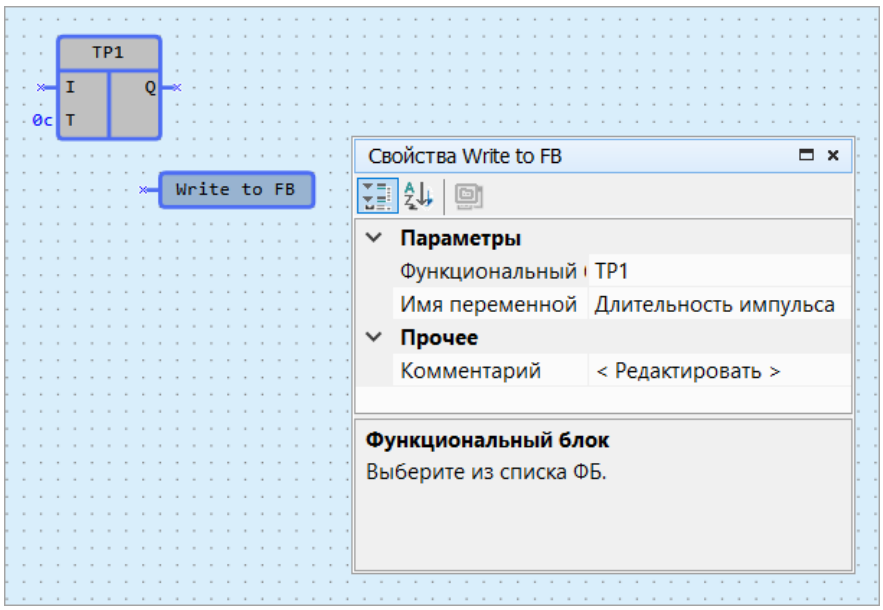


Рисунок 7.55

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

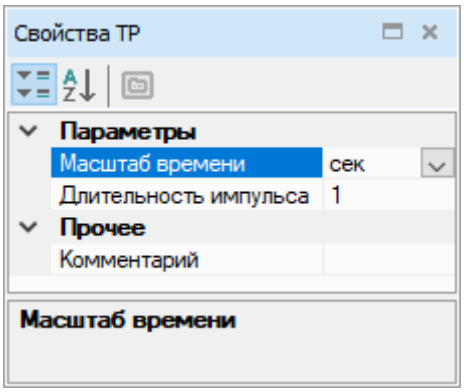


Рисунок 7.56

7.2.2.2 Таймер с задержкой включения (TON)

Таймер с задержкой включения (TON) используется для операции задержки передачи сигнала. На выходе **Q** таймера появится сигнал логической «1» с задержкой относительно фронта входного сигнала **I** продолжительностью не менее длительности **Ton** и выключится по спаду входного сигнала.

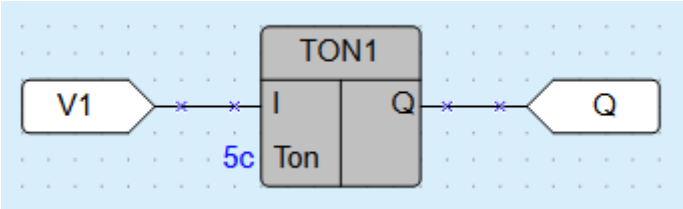


Рисунок 7.57

Работу таймера объясняет приведенная на рисунке ниже диаграмма.

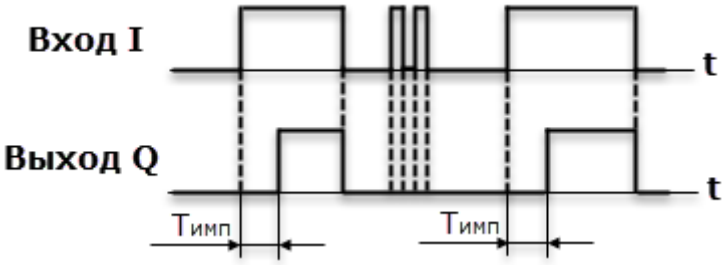


Рисунок 7.58

Допустимый диапазон значений **T** от 0 до 4147200000 мс или 48 дней.

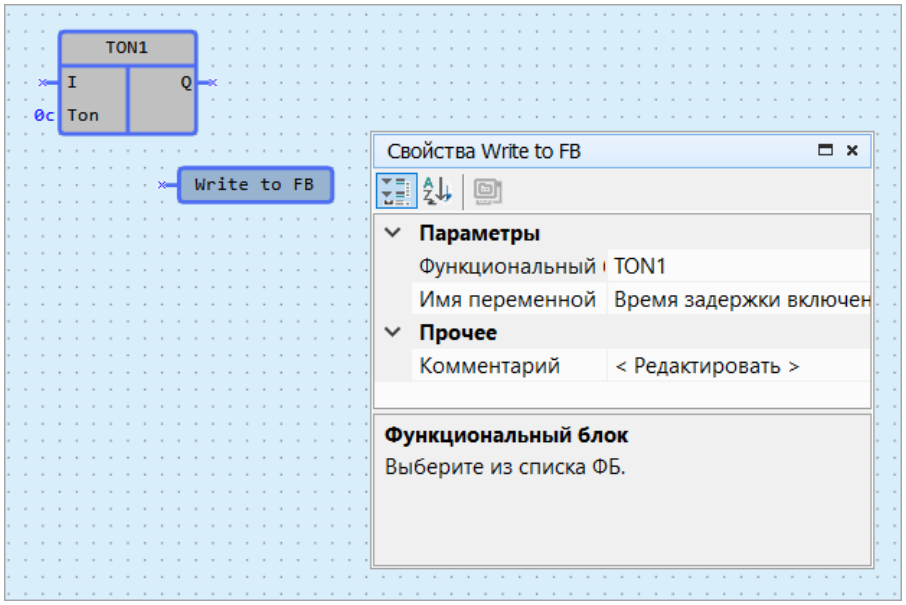


Рисунок 7.59

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

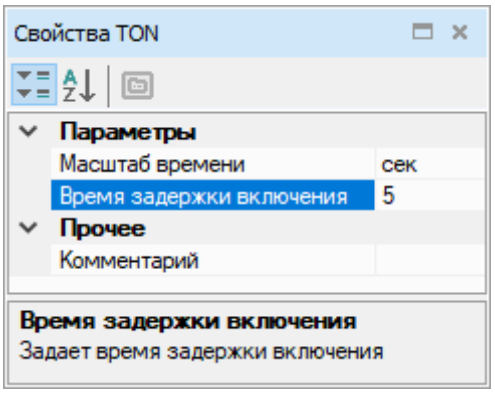


Рисунок 7.60

7.2.2.3 Таймер с задержкой отключения (TOF)

Таймер с задержкой отключения (TOF) используется для задержки отключения выхода. На выходе **Q** таймера появится сигнал логической «1» по фронту сигнала на входе **I**, отсчет времени задержки отключения **Toff** начнется по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится сигнал логического «0» с задержкой **Toff**.

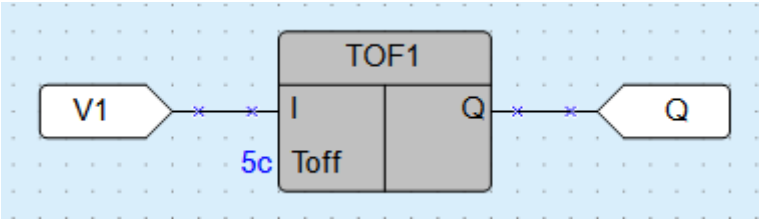


Рисунок 7.61

Работу таймера объясняет приведенная на рисунке ниже диаграмма.

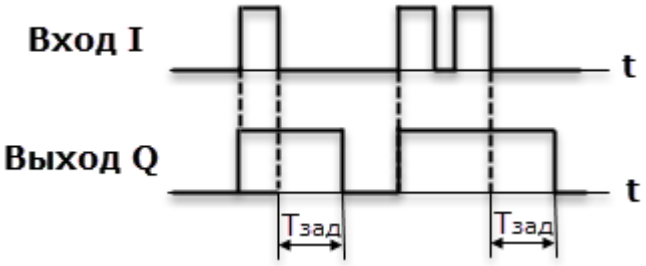


Рисунок 7.62

Допустимый диапазон значений Тзад от 0 до 4147200000 мс или 48 дней.

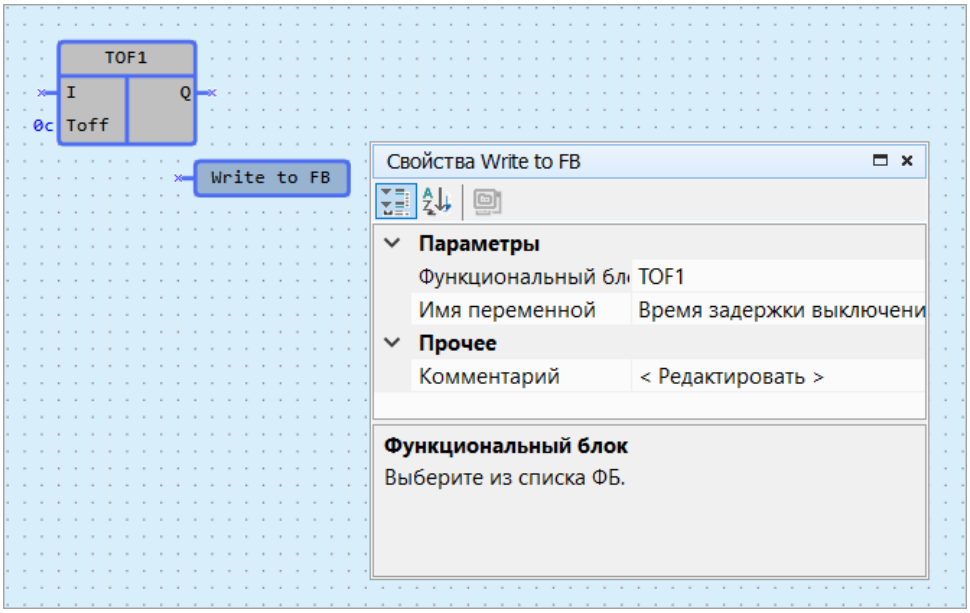


Рисунок 7.63

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

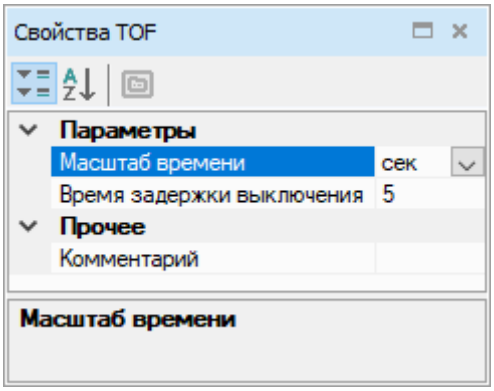


Рисунок 7.64

7.2.2.4 Интервальный таймер (CLOCK)

ПРИМЕЧАНИЕ
Интервальный таймер (CLOCK) доступен только для приборов с часами реального времени.

Интервальный таймер (CLOCK) используется для формирования импульса включения выхода **Q** по часам реального времени. Время включения **Th** и отключения **Tl** выхода устанавливают в качестве параметров таймера.

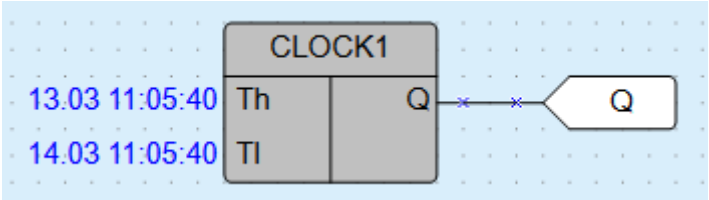


Рисунок 7.65

Работу таймера объясняет приведенная на рисунке ниже диаграмма.

Рисунок 7.66

Если заданное значение времени отключения меньше времени включения, то диаграмма переключений будет иметь вид:

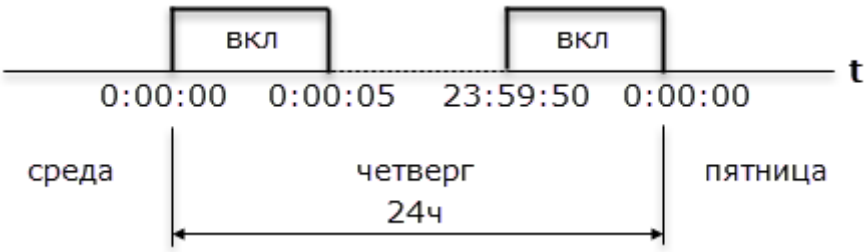


Рисунок 7.67

Допустимый диапазон значений $T_{вкл.}$ и $T_{откл.}$ от 0,00 с до 24 ч.

Дата и время включения и отключения настраивается на панели свойств ФБ.

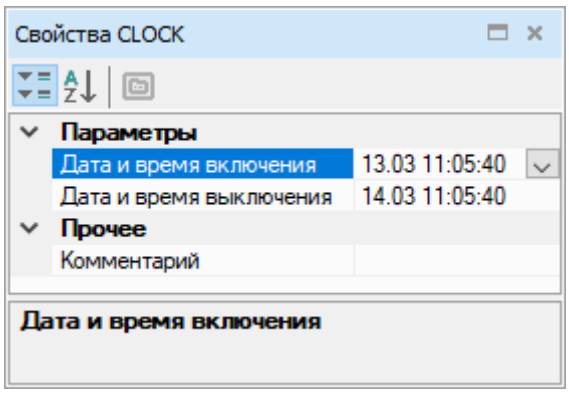


Рисунок 7.68

7.2.2.5 Интервальный таймер с недельным циклом (CLOCKWEEK)

ПРИМЕЧАНИЕ
Интервальный таймер с недельным циклом (CLOCKWEEK) доступен только для приборов с часами реального времени.

Интервальный таймер с недельным циклом (CLOCKWEEK) используется для формирования импульса включения выхода **Q** по часам реального времени с учетом дней недели. Время включения **Th** и отключения **Tl** выхода **Q** и дни недели работы устанавливают в качестве параметров таймера.

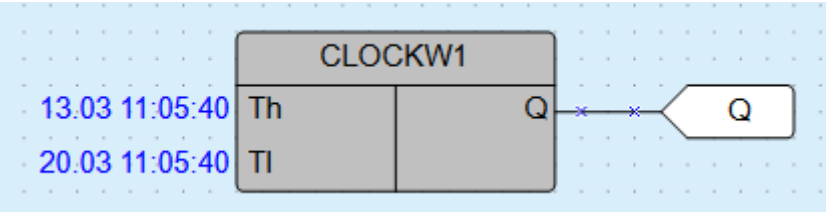


Рисунок 7.69

Работу таймера объясняет диаграмма на рисунке ниже.

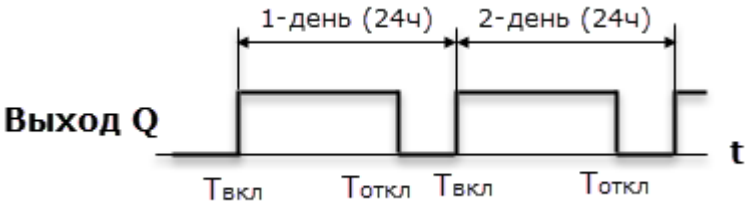


Рисунок 7.70

Внутренняя структура таймера имеет следующий вид.

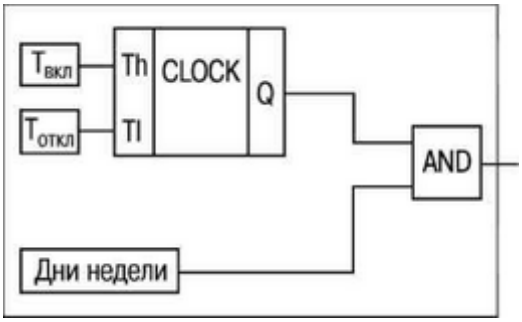


Рисунок 7.71

Допустимый диапазон значений $T_{вкл}$ и $T_{откл}$ от 0,00 с до 24 ч.

Настройка времени включения и отключения и указание дней недели производится на панели свойств ФБ.

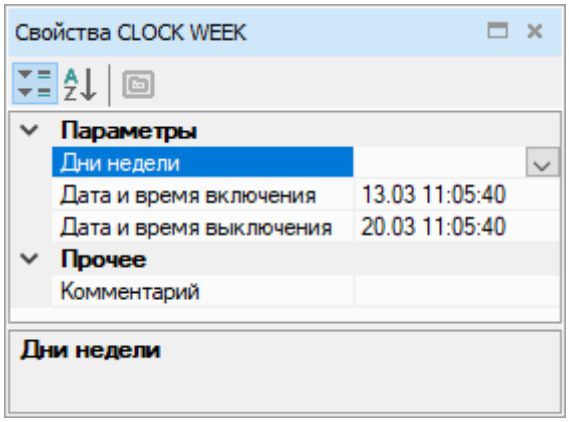


Рисунок 7.72

7.2.3 Генераторы

- Генератор прямоугольных импульсов (BLINK).

7.2.3.1 Генератор прямоугольных импульсов (BLINK)

Генератор прямоугольных импульсов (BLINK) используется для формирования прямоугольных импульсов пульсации. На выходе **Q** генератора формируются импульсы с заданными параметрами длительности включенного ($T_{вкл}$ – сигнал логической «1») и отключенного ($T_{откл}$ – сигнал логического «0») состояния на время действия управляющего сигнала на входе **I** (сигнал логической «1»).

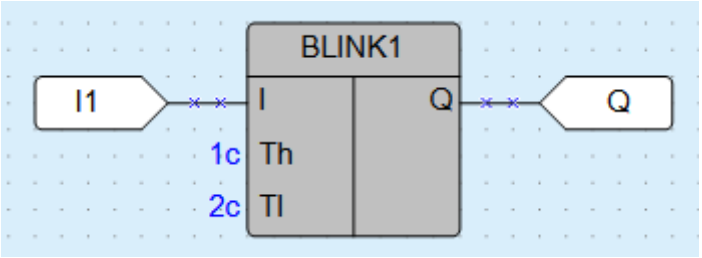


Рисунок 7.73

Работу генератора поясняет приведенная на рисунке ниже диаграмма.

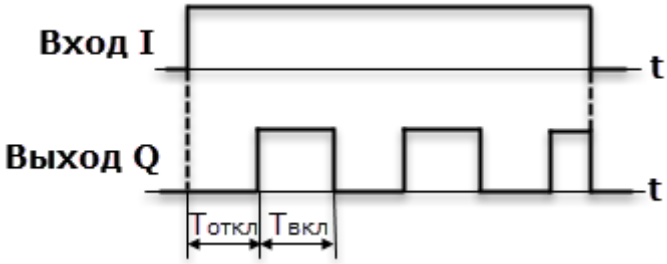


Рисунок 7.74

Допустимый диапазон значений $T_{вкл}$ и $T_{откл}$: от 0 до 4233600000 мс или 49 дней.

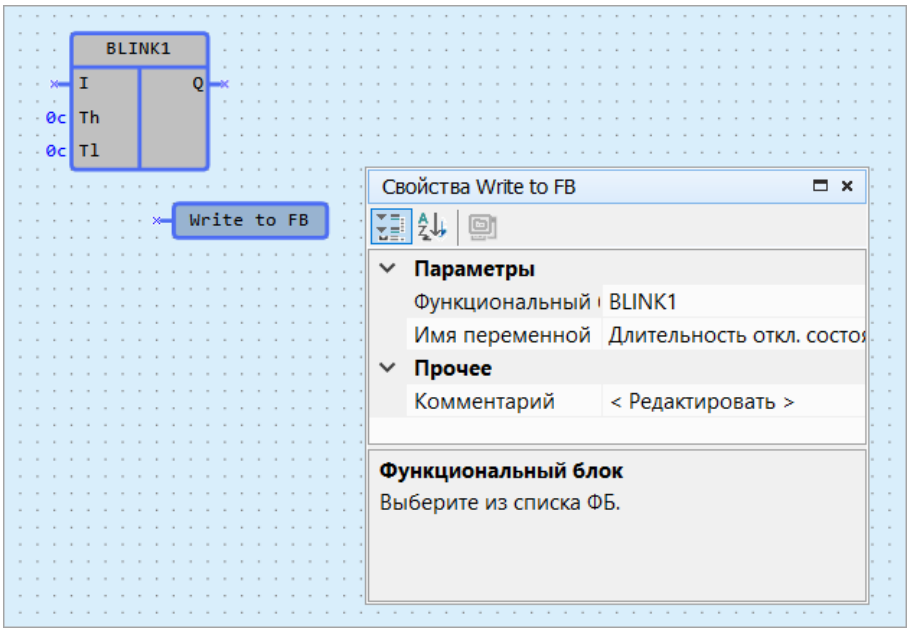


Рисунок 7.75

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

Параметр Th соответствует $T_{вкл}$.

Параметр Tl соответствует $T_{откл}$.

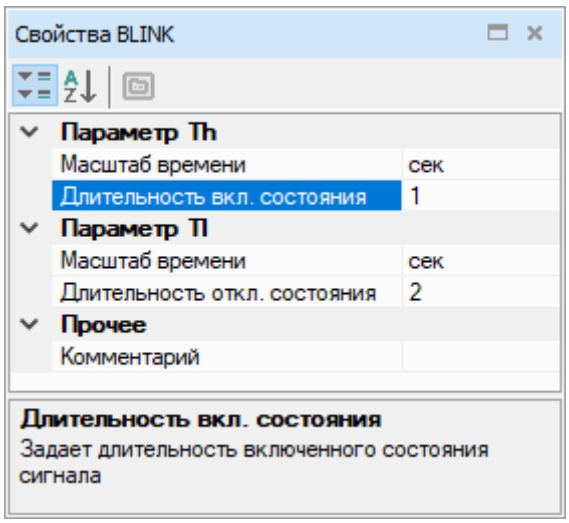


Рисунок 7.76

7.2.4 Счетчики

- Инкрементный счетчик с автосбросом (CT) 7.2.4.1;
- Универсальный счетчик (CTN) 7.2.4.2;
- Инкрементный счетчик (CTU) 7.2.4.3.

7.2.4.1 Инкрементный счетчик с автосбросом (CT)

Инкрементный счетчик с автосбросом (CT) используется для подсчета заданного числа импульсов **N** (вход **N** – уставка числа импульсов). На выходе **Q** счетчика появится импульс сигнала логической «1» с длительностью рабочего цикла прибора (Тцикл), если число приходящих на вход **C** импульсов достигнет установленного значения **N**.

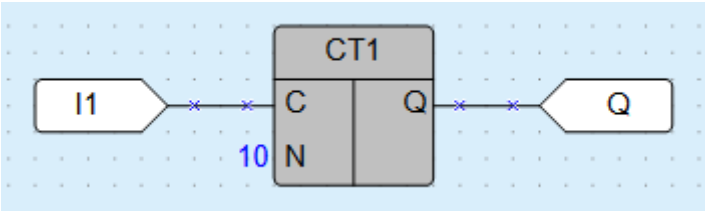


Рисунок 7.77

Работу счетчика поясняет приведенная на рисунке ниже диаграмма.

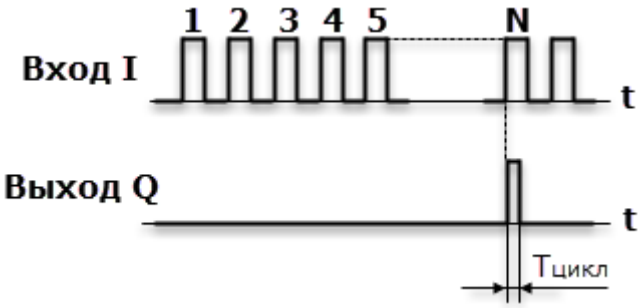


Рисунок 7.78

Допустимый диапазон значений числа импульсов **N**: от 0 до 65535.

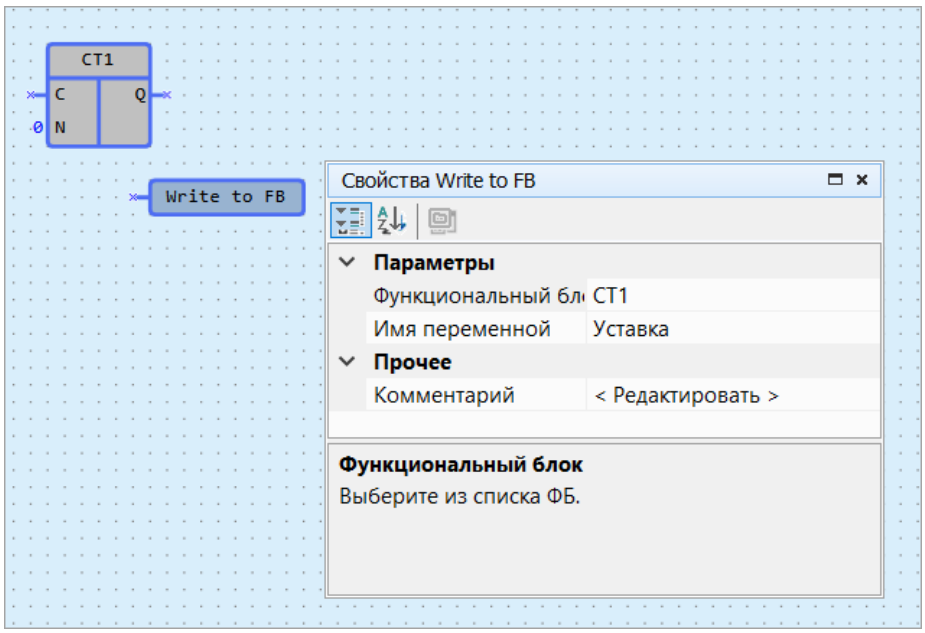


Рисунок 7.79

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

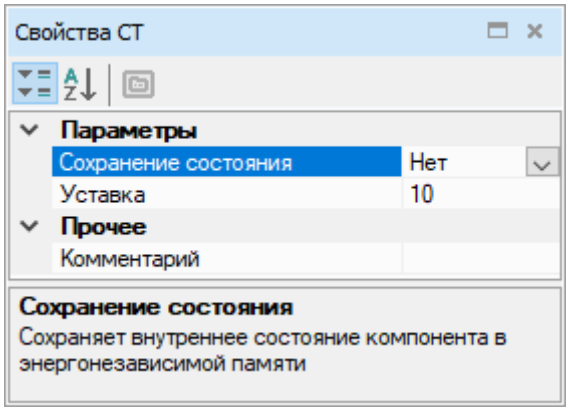


Рисунок 7.80

7.2.4.2 Универсальный счетчик (CTN)

Универсальный счетчик (CTN) используется для прямого и обратного счета. Операция «прямой счет» выполняется по переднему фронту импульса на входе прямого счета **U**, что увеличивает значение выходного сигнала **Q**. Импульсы, приходящие на вход **D** («обратный счет»), уменьшают значение выхода **Q**. В случае поступления на вход **R** сигнала логической «1», выход счетчика **Q** устанавливается в значение входа **N**.

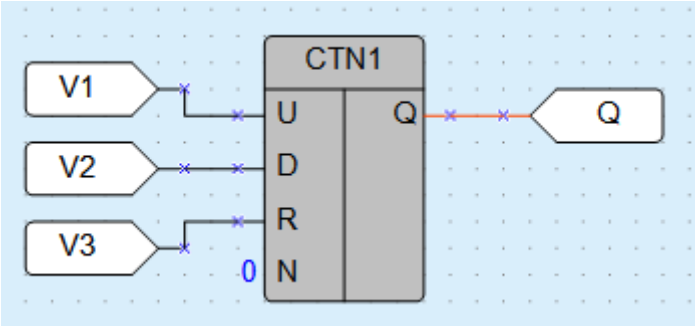


Рисунок 7.81

Работу счетчика поясняет приведенная на рисунке ниже диаграмма.

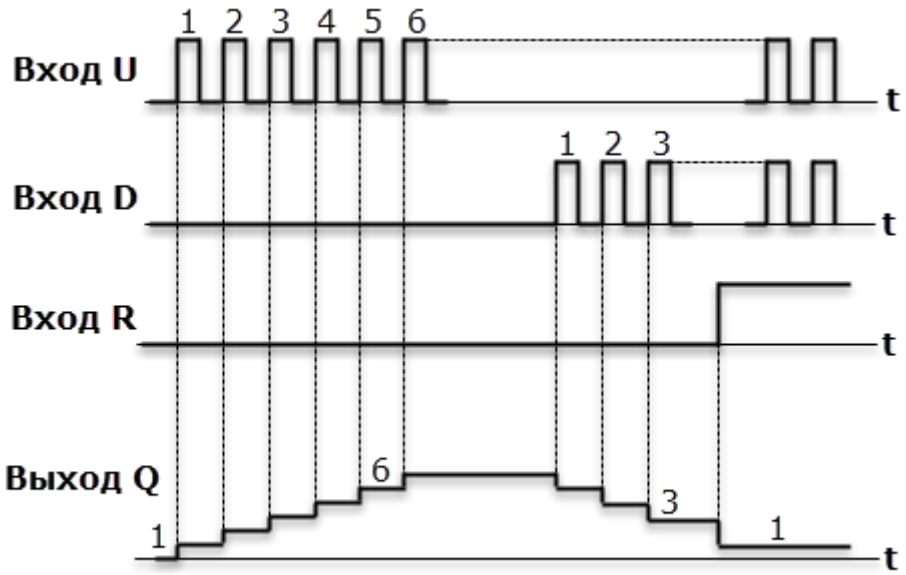


Рисунок 7.82

В случае одновременного поступления сигналов на входы **U** и **D** приоритетным является сигнал входа **U**.

Допустимый диапазон значений числа импульсов N: от 0 до 65535.

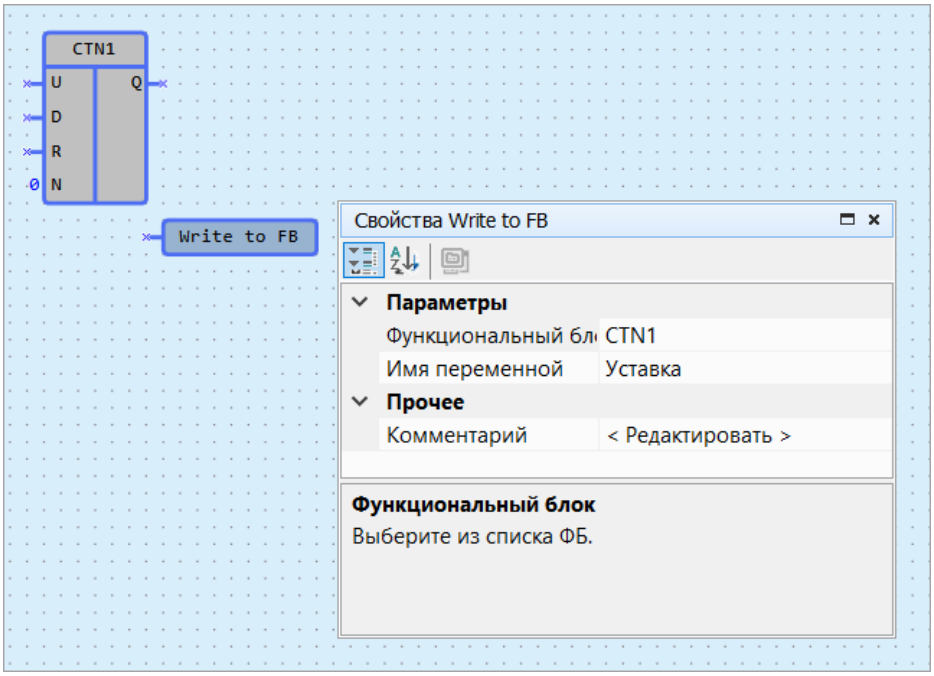


Рисунок 7.83

Значения параметров ФБ можно задавать с помощью блока Write to FB, как показано на рисунке выше, либо настраивать на панели свойств:

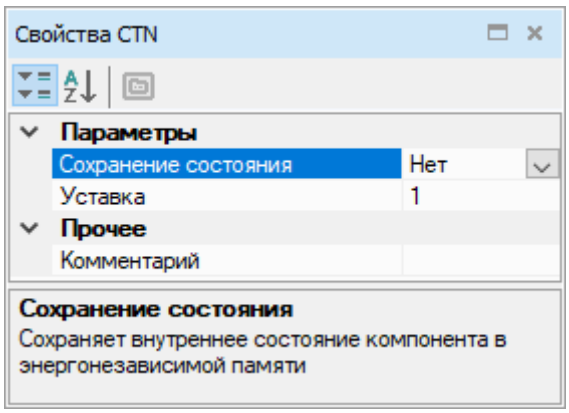


Рисунок 7.84

7.2.4.3 Инкрементный счетчик (CTU)

Инкрементный счетчик (CTU) используется для подсчета числа импульсов, приходящих на вход **C**. На выходе **Q** счетчика появится импульс сигнала логической «1», если число приходящих на вход импульсов достигнет установленного значения на входе **N** (**N** – уставка).

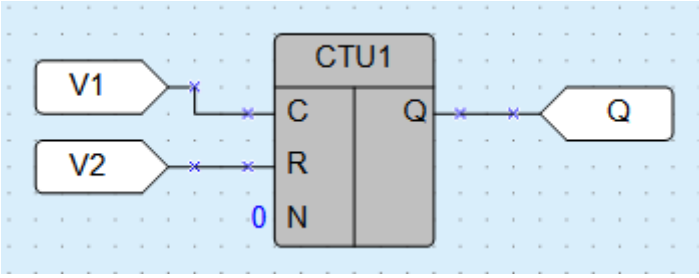


Рисунок 7.85

Работу счетчика поясняет приведенная на рисунке ниже диаграмма.

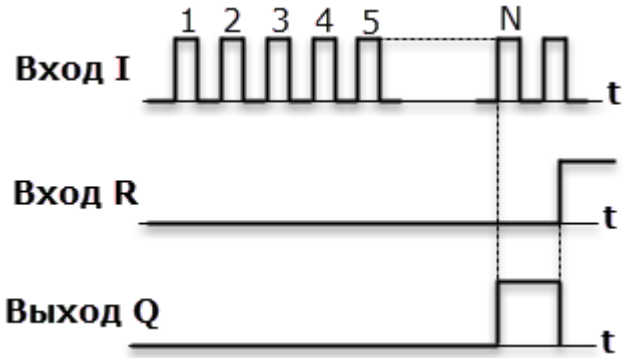


Рисунок 7.86

ПРИМЕЧАНИЕ
Блок **Запись в ФБ** для CTU не работает. Вместо CTU рекомендуется использовать универсальный счетчик CTN 7.2.4.2.

Допустимый диапазон значений числа импульсов **N**: от 0 до 65535.

Счетчик сбрасывается в 0 по переднему фронту импульса на входе **R**. В случае одновременного поступления сигналов на входы приоритетным является сигнал входа **R**.

Число импульсов N задается на панели свойств ФБ.

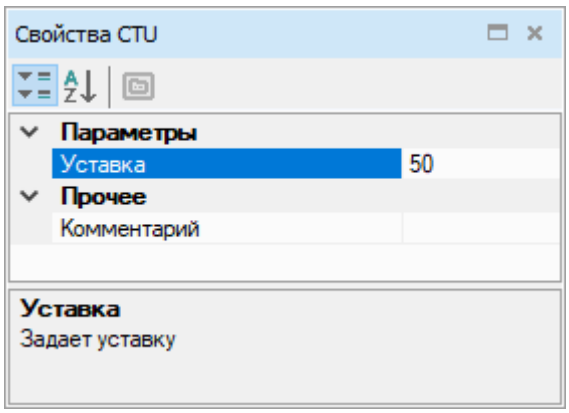


Рисунок 7.87

7.2.5 Регуляторы

- ПИД-регулятор для приборов первого поколения (PID)
- ПИД-регулятор для приборов второго поколения (PID REG)

7.2.5.1 ПИД-регулятор для приборов первого поколения (PID)

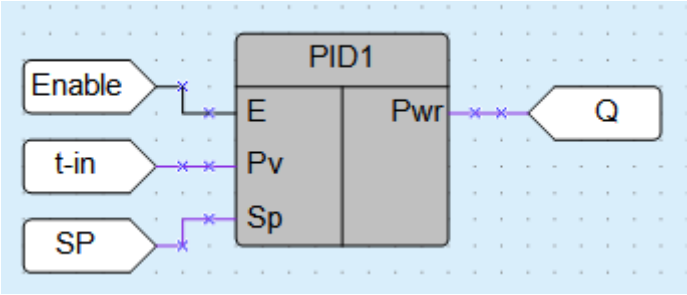


Рисунок 7.88

ПИД-регулятор используется для реализации пропорционально-интегрально-дифференциального (ПИД) закона регулирования.

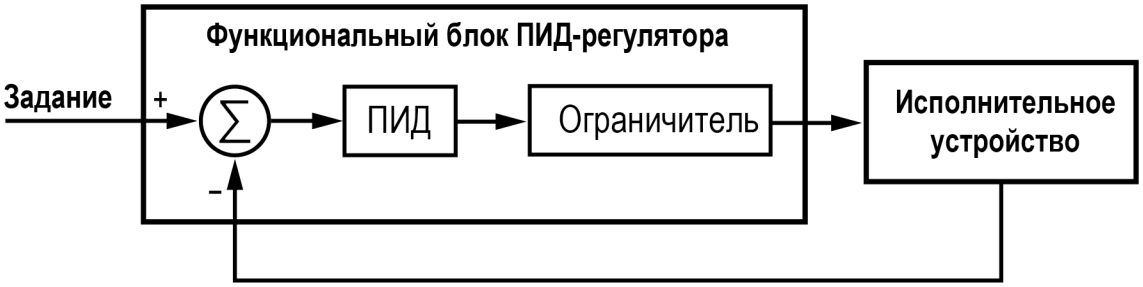


Рисунок 7.89

Входы блока

Таблица 7.1 – Входы блока

Название	Тип	Описание	Значения
E	Булевский	Разрешение на работу блока. В выключенном состоянии на выход блока подается значение, заданное в свойстве «выходная мощность»	0 – Выключен; 1 – Включен
Pv	С плавающей запятой	Текущее значение регулируемой величины	
Sp	С плавающей запятой	Заданное значение регулируемой величины	

Выходы блока

Таблица 7.2 – Выходы блока

Название	Тип	Описание	Значения
Pwr	С плавающей запятой	Выходная мощность, %	0…100

Свойства

Таблица 7.3 – Описание свойств

Название	Тип	Описание	Значения
Режим работы регулятора	Булевский	Режимы работы регулятора: • режим «нагреватель», используется для управления исполнительными механизмами, воздействие которых приводит к росту значения регулируемой величины; • режим «холодильник», используется для управления исполнительными механизмами, воздействие которых приводит к снижению значения регулируемой величины	0 – Нагреватель; 1 – Холодильник
Выходная мощность	С плавающей запятой	Выходная мощность в выключенном состоянии, %	0…100
Дифферен-циальный коэффициент	С плавающей запятой	Дифференциальный коэффициент с которым работает ПИД	–3,402823E+38…3,402823E+38
Интегральный коэффициент	С плавающей запятой	Интегральный коэффициент с которым работает ПИД	–3,402823E+38…3,402823E+38
Пропорцио-нальный коэффициент	С плавающей запятой	Пропорциональный коэффициент с которым работает ПИД	0…100
Минимальная мощность (значение по умолчанию)	С плавающей запятой	Нижнее ограничение выдаваемой мощности, %	0…100 (20)
Максимальная мощность (значение по умолчанию)	С плавающей запятой	Верхнее ограничение выдаваемой мощности, %	0…100 (80)
Запуск АНР	Булевский	Переменная, которая генерирует запуск автонастройки регулятора в случае подачи значения «1». Значение данной переменной задается с помощью блока «Запись в ФБ»	0 – Стоп; 1 – Запуск

Автонастройка

Автонастройка ПИД-регулятора проводится с помощью блоков чтения и записи ФБ (W, R).

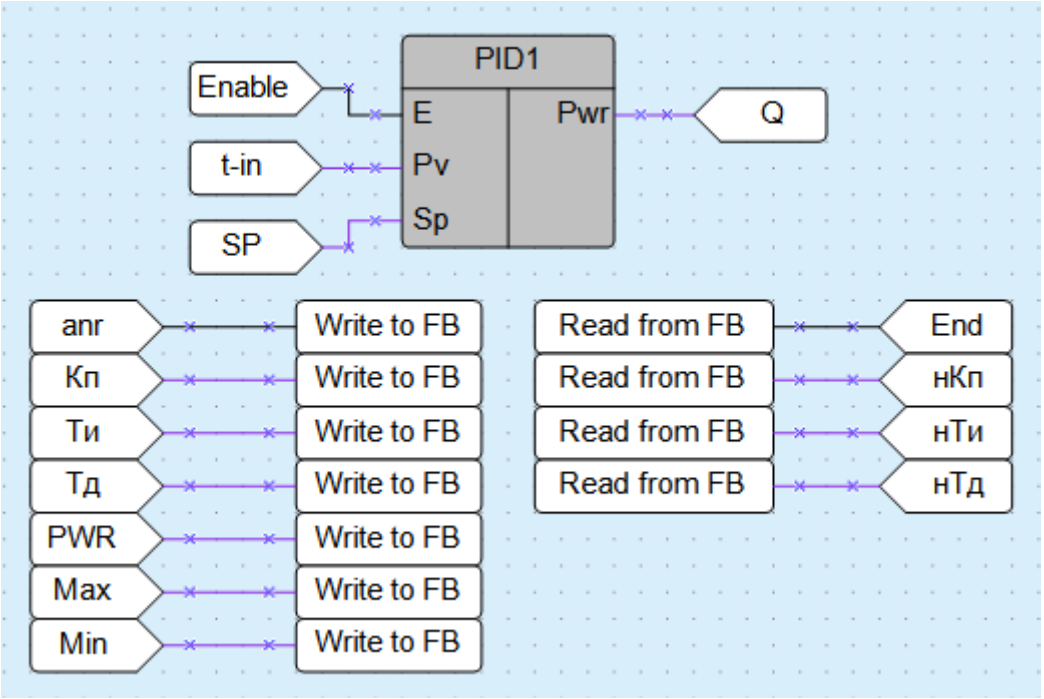


Рисунок 7.90

Для запуска автонастройки следует добавить блок **Запись в ФБ** и привязать его к переменной **Запуск АНР** ПИД-регулятора.

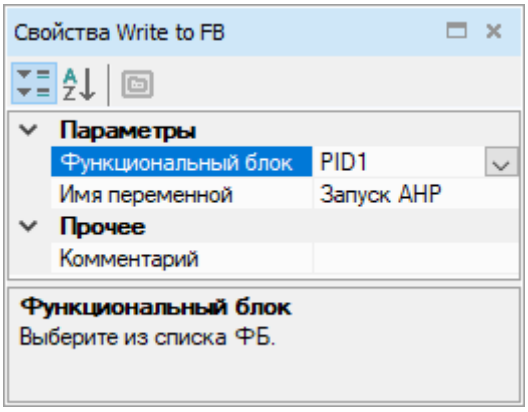


Рисунок 7.91

Значения остальных параметров ФБ можно задавать с помощью блока **Запись в ФБ**, как показано на рисунке выше, либо настраивать на панели свойств.

С помощью блоков **Чтение из ФБ** считываются значения параметров **Рассчитанный пропорциональный коэффициент**, **Рассчитанный интегральный коэффициент**, **Рассчитанный дифференциальный коэффициент** и **Флаг окончания автонастройки**.

Для старта автонастройки следует подать сигнал логической «1» на вход **E**.

По завершении процесса автонастройки для чтения доступны новые значения коэффициентов: **Рассчитанный пропорциональный коэффициент**, **Рассчитанное время интегрирования** и **Рассчитанное время дифференцирования**. Параметр **Флаг окончания автонастройки** выставляется в значение логической единицы. Если после автонастройки сбросить вход **Запуск АНР** в значение логического нуля, то сбросится и флаг окончания.

ВНИМАНИЕ
Флаг окончания автонастройки находится в состоянии логической единицы один цикл.

Если сбросить вход **Запуск АНР** в значение логического нуля до окончания настройки, то процесс останавливается, флаг окончания не выставляется, новые значения коэффициентов не вычисляются.

Во время процесса настройки на выход ПИД-регулятора подается тестовая мощность, ограниченная значениями параметров **Максимальная мощность** и **Минимальная мощность**.

Последовательность автонастройки для режима «Нагреватель»:

1. Текущее значение меньше уставки, на выход блока подается максимальная мощность (в соответствии с настройками).
2. Как только текущее значение станет больше уставки, на выход блока подается минимальная мощность.
3. Повторение шагов 1 и 2 еще один раз.
4. Рассчитанные параметры ПИД-регулятора подаются на соответствующие выходы, и выставляется флаг окончания.

Если в настройках указано значение максимальной мощности, при которой невозможно достичь уставки, процесс автонастройки не закончится, пока он не будет сброшен вручную.

7.2.5.2 ПИД-регулятор для приборов второго поколения (PID REG)

ПРИМЕЧАНИЕ
В линейке приборов ПР103 PID REG доступен только для модификации ПР103[M02].

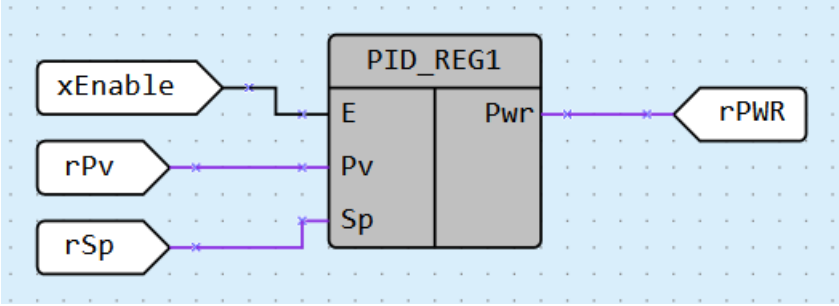


Рисунок 7.92

ПИД-регулятор используется для реализации пропорционально-интегрально-дифференциального (ПИД) закона регулирования.

Выходная мощность ПИД-регулятора высчитывается по формуле:

$$MV(t) = \frac{1}{X_p}\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right)$$

, где

Xp — полоса пропорциональности (единицы измерения);

Ti — интегральный коэффициент [с];

Td — дифференциальный коэффициент [с];

e (t) —рассогласование;

MV (t) — выход ПИД.

Входы блока

Таблица 7.4 – Входы блока

Название	Тип	Описание	Значения
E	Булевский	Разрешение на работу регулятора. В выключенном состоянии на выход подается значение, заданное в свойстве «Выходная мощность в выключенном состоянии»	0 – Выключен; 1 – Включен
Pv	С плавающей запятой	Текущее значение регулируемой величины	–9999...9999*
Sp	С плавающей запятой	Задаваемое значение регулируемой величины	

*Если на вход будет подано значение, выходящее за диапазон, то выход Pwr регулятора будет сброшен на значение мощности, указанное в параметре «Выходная мощность в выключенном состоянии».

Выходы блока

Таблица 7.5 – Выходы блока

Название	Тип	Описание	Значения
Pwr	С плавающей запятой	Выходная мощность регулятора	

Входы, доступные через Write to FB

Таблица 7.6 – Входы, доступные через Write to FB

Название	Тип	Описание	Значения
Режим работы регулятора	Булевский	Режимы работы регулятора: • режим «нагреватель», используется для управления исполнительными механизмами, воздействие которых приводит к росту значения регулируемой величины; • режим «холодильник», используется для управления исполнительными механизмами, воздействие которых приводит к снижению значения регулируемой величины	0 – Нагреватель; 1 – Холодильник
Запуск АНР	Булевский	Управление автонастройкой регулятора	0 – Автонастройка выключена 1 – Автонастройка запущена
Полоса пропорциональности	С плавающей запятой	Полоса пропорциональности ПИД регулятора, ед.изм.	0..9999
Интегральная постоянная времени	С плавающей запятой	Интегральная постоянная времени, сек	0..4000
Дифференциальная постоянная времени	С плавающей запятой	Дифференциальная постоянная времени, сек	0..4000
Выходная мощность в выключенном состоянии	С плавающей запятой	Состояние выхода Pwr при значении 0 на входе E	-9999..9999
Максимальная мощность	С плавающей запятой	Верхнее ограничение выдаваемой мощности	-9999..9999
Минимальная мощность	С плавающей запятой	Нижнее ограничение выдаваемой мощности	-9999..9999
Зона нечувствительности	С плавающей запятой	Дельта от уставки/2, в которой выходная мощность регулятора остается неизменной, ед.изм	
Тип выхода	Булевский	Выбор выходного сигнала регулятора	0 – Мощность 1 – КЗР дискретный
Полное время хода	С плавающей запятой	Время перемещения штока клапана из полностью закрытого в полностью открытое состояние, сек	5..999**

Продолжение таблицы 7.6

Название	Тип	Описание	Значения
Минимальное время импульса	С плавающей запятой	Минимальная длительность импульса управления задвижкой, сек	0,001..100
Время выборки люфта	С плавающей запятой	Время выборки люфта клапана, сек	0..10
Скорость изменения уставки	С плавающей запятой	Скорость изменения уставки, ед. изм/сек	0 – Функция отключена 0...9999
Ручной режим работы регулятора	Булевский	Флаг перевода регулятора в ручной режим	0 – Автоматический режим 1 – Ручной режим
Ручное управление выходом «Больше»	Булевский	В ручном режиме команда подачи сигнала «Больше» на выход блока	
Ручное управление выходом «Меньше»	Булевский	В ручном режиме команда подачи сигнала «Меньше» на выход блока	
Начальное значение регулируемой величины для АНР (Pv_0)	С плавающей запятой	Значение регулируемой величины, которое будет на входе Pv при подаче нулевой мощности (объект в состоянии покоя). Параметр необходим для корректного проведения автонастройки	-9999..9999

**Для применения нового полного времени хода задвижки нужно перезапустить блок при помощи переключения сигнала на входе E.

Выходы, доступные через Read from FB

Таблица 7.7 – Выходы, доступные через Read from FB

Название	Тип	Описание	Значения
Флаг окончания автонастройки	Булевский	Флаг окончания автонастройки	0 – АНР не завершена 1 – АНР завершена
Рассчитанная полоса пропорцио-нальности	С плавающей запятой	Рассчитанная после автонастройки полоса пропорциональности ПИД регулятора, ед.изм.	0..9999
Рассчитанная интегральная постоянная времени	С плавающей запятой	Рассчитанная после автонастройки интегральная постоянная времени, сек	0..4000
Рассчитанная дифферен-циальная постоянная времени	С плавающей запятой	Рассчитанная после автонастройки дифференциальная постоянная времени, сек	0..4000
Сигнал «Больше»	Булевский	Сигнал на открытие клапана	
Сигнал «Меньше»	Булевский	Сигнал на закрытие клапана	

Автонастройка

Автонастройка ПИД-регулятора проводится с помощью блоков чтения и записи в ФБ.

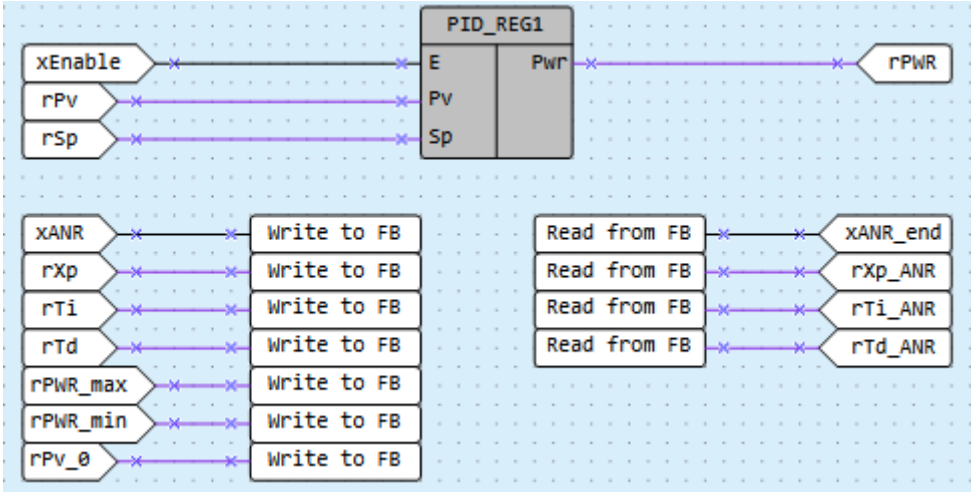


Рисунок 7.93

Для запуска автонастройки следует добавить блок **Write to FB** и привязать его к переменной **Запуск АНР** ПИД-регулятора.

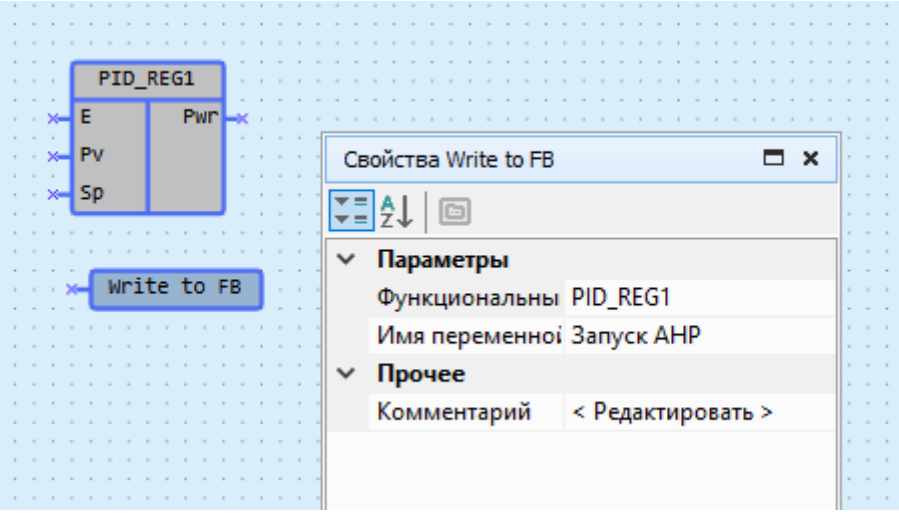


Рисунок 7.94

Значения остальных параметров ФБ можно задавать с помощью блока **Write to FB**, как показано на рисунке выше, либо настраивать на панели свойств.

С помощью блоков **Read from FB** считываются значения параметров **Рассчитанная полоса пропорциональности**, **Рассчитанная интегральная постоянная времени**, **Рассчитанная дифференциальная постоянная времени** и **Флаг окончания автонастройки**.

Для старта автонастройки следует подать сигнал логической «1» на вход **E**.

По завершении процесса автонастройки для чтения доступны новые значения коэффициентов: **Рассчитанный пропорциональный коэффициент**, **Рассчитанное время интегрирования** и **Рассчитанное время дифференцирования**. Параметр **Флаг окончания автонастройки** выставляется в значение логической единицы.

ВНИМАНИЕ
Флаг окончания автонастройки находится в состоянии логической единицы один цикл.

Если сбросить вход **Запуск АНР** в значение логического нуля до окончания настройки, то процесс останавливается, флаг окончания не выставляется, новые значения коэффициентов не вычисляются.

ВНИМАНИЕ
Для корректного проведения автонастройки необходимо задать параметр **«начальное значение регулируемой величины для АНР»**. Он должен быть равен регулируемой величине, которое соответствует нулевому воздействию (объект находится в состоянии покоя).

Во время процесса настройки на выход ПИД-регулятора подается тестовая мощность, ограниченная значениями параметров **Максимальная мощность** и **Минимальная мощность**.

Последовательность автонастройки для режима «Нагреватель»:

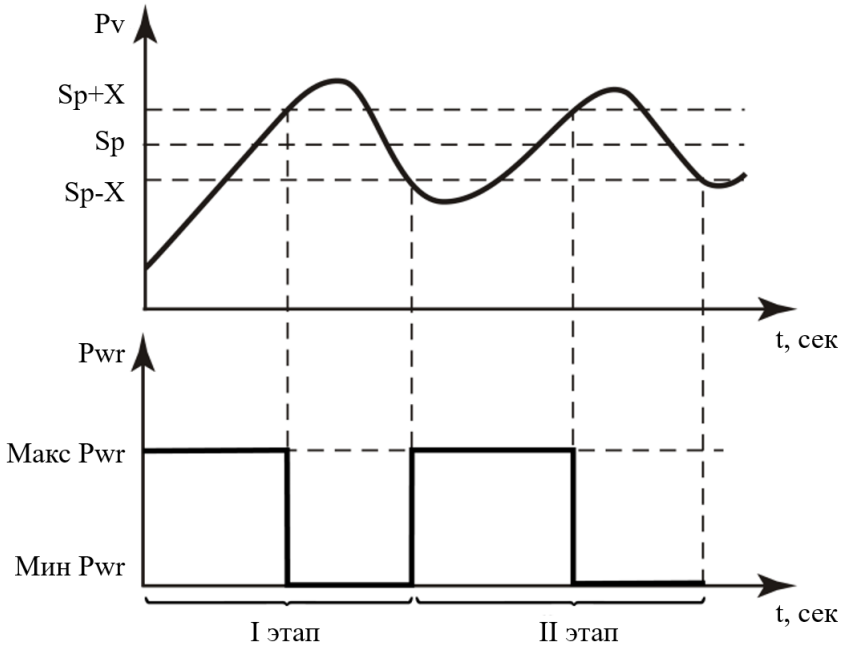


Рисунок 7.95

1. Текущее значение меньше уставки на величину, большую чем *X = 0,04*(Sp-Pv_0)*, на выход блока подается максимальная мощность (в соответствии с настройками).
2. Как только текущее значение станет больше уставки на величину, большую чем *X = 0,04*(Sp-Pv_0)*, на выход блока подается минимальная мощность.
3. Повторение шагов 1 и 2 еще один раз.
4. Рассчитанные параметры ПИД-регулятора подаются на соответствующие выходы, и выставляется флаг окончания (на один цикл).

Если в настройках указано значение максимальной мощности, при которой невозможно достичь уставки, процесс автонастройки не закончится, пока он не будет сброшен вручную.

Работа с дискретным КЗР

Для работы с дискретной задвижкой необходимо в соответствующем параметре (Тип выхода) указать значение **«КЗР дискретный»**, либо, при работе с данным параметром через **Write to FB** записать в него значение «1».

Для корректной работы необходимо задать параметры задвижки: **Время полного хода**, **Минимальное время импульса** и **Время выборки люфта**.

Доступ к управляющим сигналам (Больше/Меньше) осуществляется через соответствующие выходы **Read from FB** блока регулятора.

На выходе **Pwr** в данном режиме отображается виртуальное положение задвижки (оно может отличаться от фактического).

При снятии логической «1» со входа E задвижка остается в текущем положении.

Для быстрой настройки рекомендуем воспользоваться готовыми макросами из Менеджера компонентов:

- Для управления мощностью — блок **PID AT_**;
- Макрос с полным функционалом в режиме управления мощностью — **PID AT_F_**;
- Для управления дискретным КЗР – блок **PID Valve_**;
- Макрос с полным функционалом в режиме управления дискретным КЗР — **PID Valve_F_**.

Ручной режим

Работа в режиме управления мощностью: при переводе регулятора в ручной режим (вход **Ручной режим работы регулятора** устанавливается в значение 1) происходит фиксация вычисленного значения мощности.

Работа в режиме управления дискретным КЗР: при переводе регулятора в ручной режим (вход **Ручной режим работы регулятора** устанавливается в значение 1) происходит фиксация вычисленного положения задвижки и становятся доступными входы по ручному управлению открытием/закрытием клапана.

7.3 Макросы проекта

Раздел **Макросы проекта** содержит созданные пользователем или загруженные с помощью менеджера компонентов 4.10 макросы.

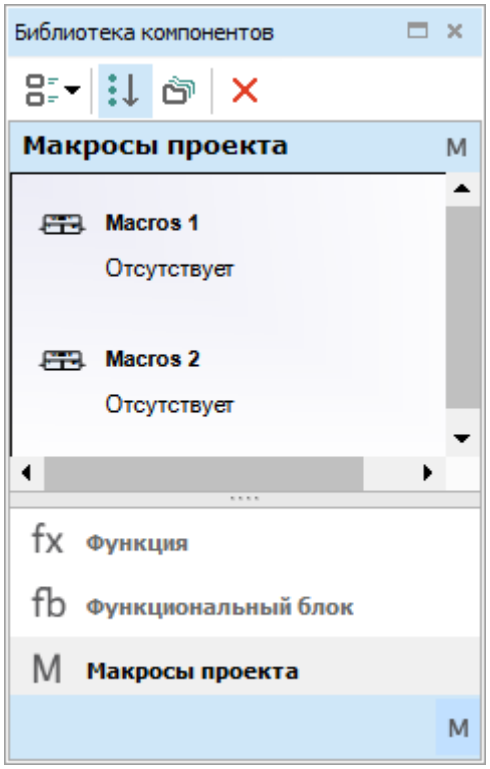


Рисунок 7.96

Для добавления макроса в проект следует перетащить макрос из панели Библиотека компонентов 3.3 на холст.

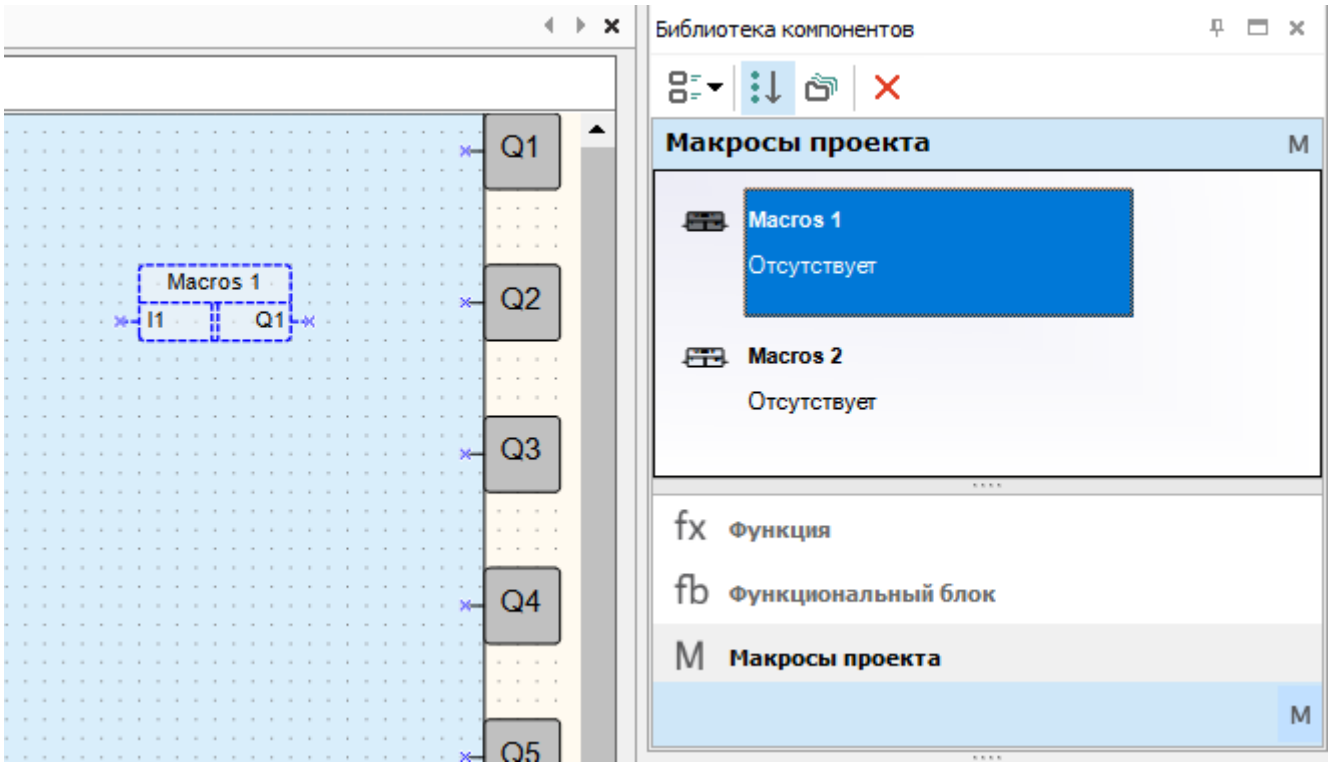


Рисунок 7.97

Чтобы удалить макрос из панели, следует выделить нужный макрос и нажать на кнопку ✕.

Рекомендации по созданию макросов приведены в разделе Работа с макросами 4.11.

7.4 Функция на ST

Если в проекте созданы функции на языке ST 4.12.1, они будут доступны в библиотеке компонентов.

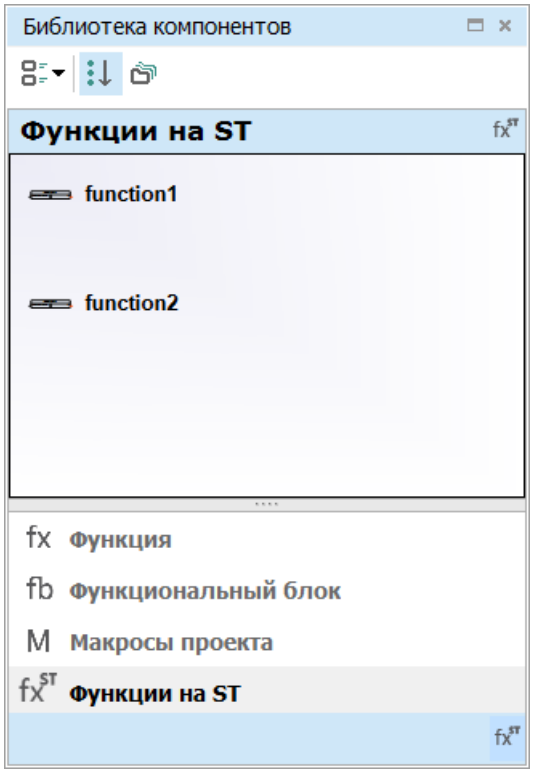


Рисунок 7.98

Места использования функции

Для просмотра все мест использования функции следует:

1. Нажать ПКМ на имя функции.
2. Выбрать пункт **Показать места использования…**.

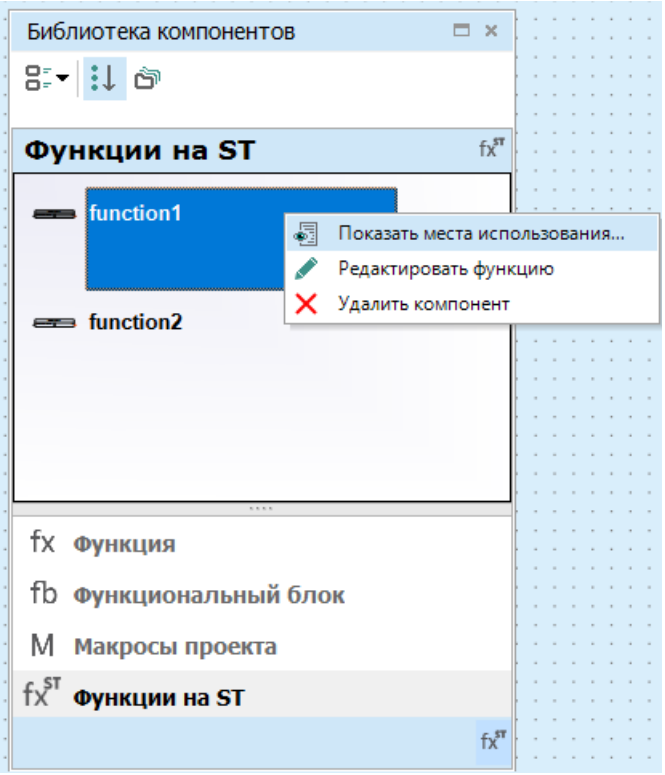


Рисунок 7.99

В нижней части окна откроется панель **Места использования функции**, на которой отобразятся места использования функции на схеме и в редакторе функций.

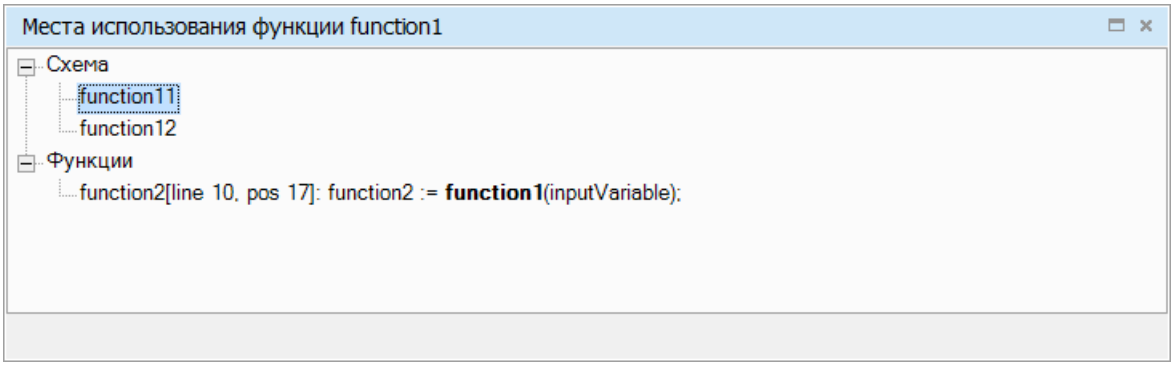


Рисунок 7.100

3. Нажать ПКМ на строку с местом использования функции.
4. Выбрать пункт **Перейти к месту использования**.

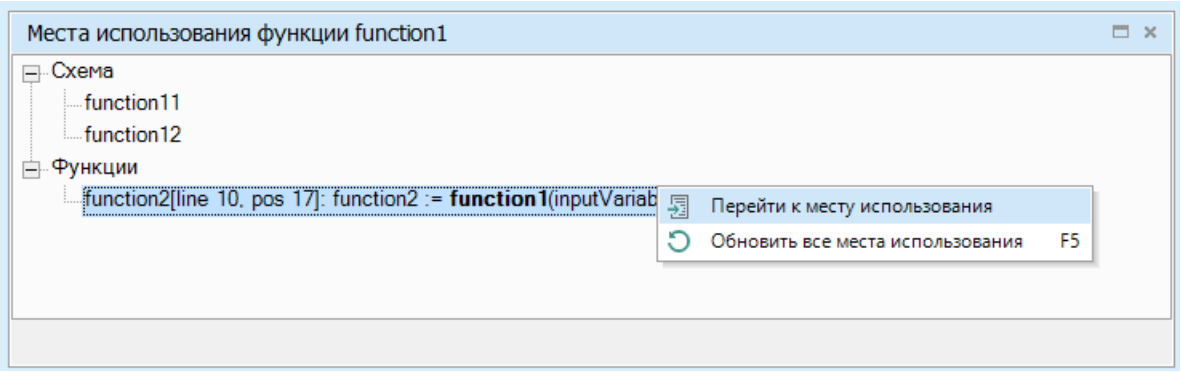


Рисунок 7.101

Фокус сместится в места использования функции на схеме или в редакторе функции.

ПРИМЕЧАНИЕ
Двойной клик ЛКМ приводит к тому же результату.

Если в ходе работы с программой места использования функции изменились, следует обновит панель **Места использования функции**:

1. Нажать ПКМ на любую строку панели.
2. Выбрать пункт **Обновить все места использования**.

Переход к редактору функций

Для перехода к Редактору функций 4.12.1 следует:

1. Нажать ПКМ на имя функции.
2. Выбрать пункт **Редактировать функцию**.

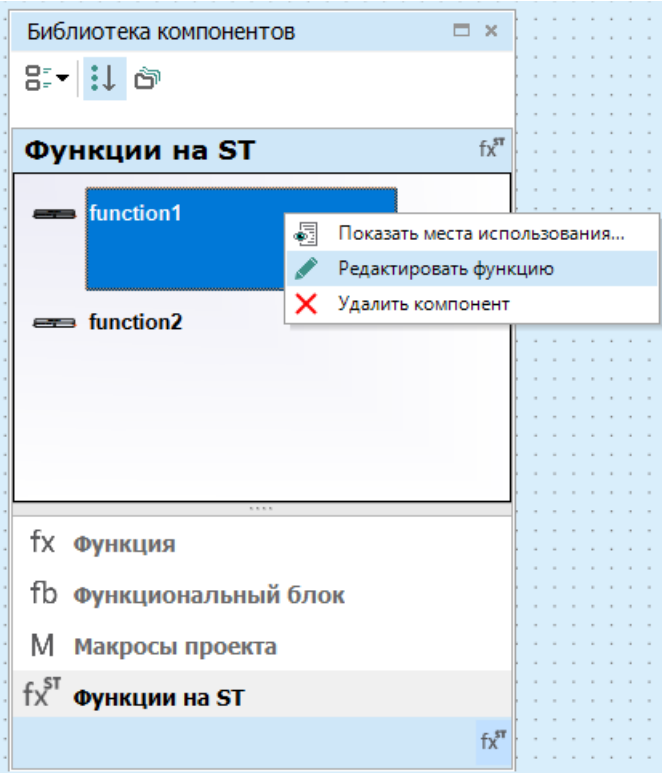


Рисунок 7.102

Откроется редактор функции 4.12.1.

Удаление функции

Для удаления функции из проекта следует:

1. Нажать ПКМ на имя функции.
2. Выбрать пункт **Удалить компонент**.

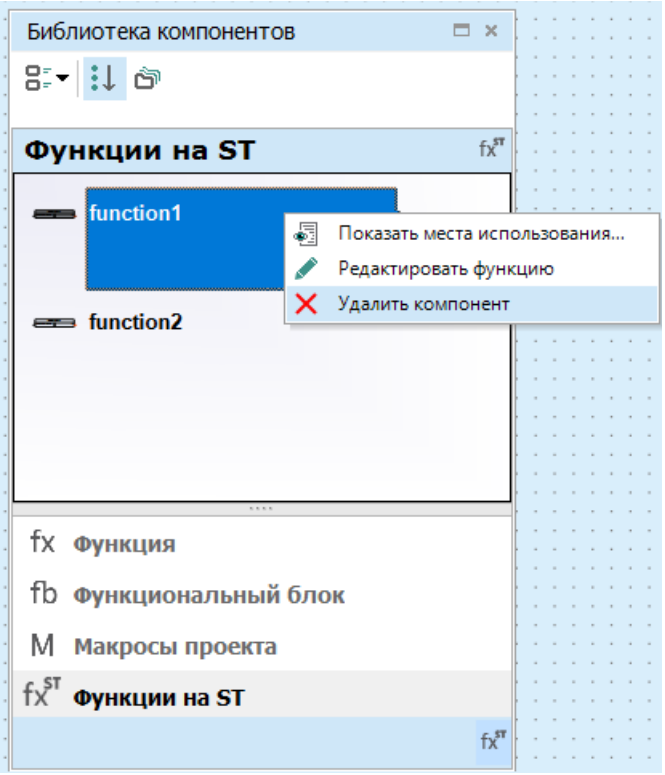


Рисунок 7.103

ПРЕДУПРЕЖДЕНИЕ
Если функция используется на схеме и/или в других функциях, удаление может привести к ошибкам компиляции.

7.5 Функциональный блок на ST

Если в проекте созданы функциональные блоки на языке ST 4.12.2, они будут доступны в библиотеке компонентов.

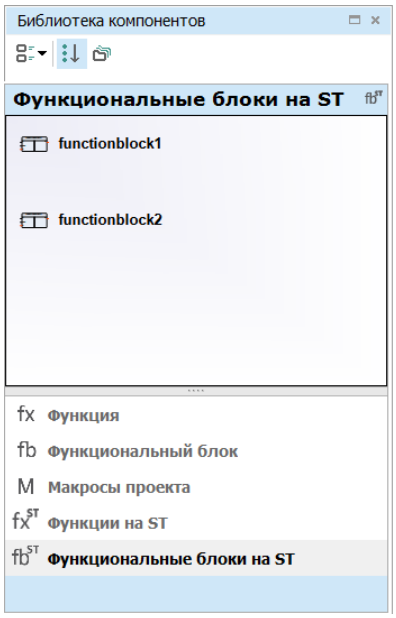


Рисунок 7.104

Места использования функционального блока

Для просмотра всех мест использования функции следует:

1. Нажать ПКМ на имя функционального блока.
2. Выбрать пункт **Показать места использования…**.

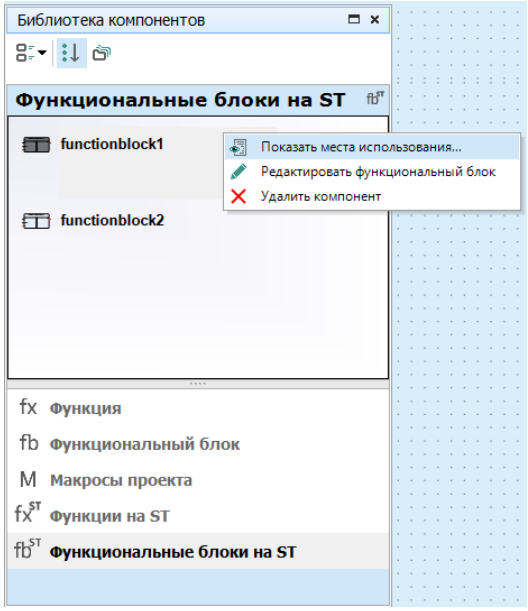


Рисунок 7.105

В нижней части окна откроется панель **Места использования функции**, на которой отобразятся места использования функции на схеме и в редакторе.

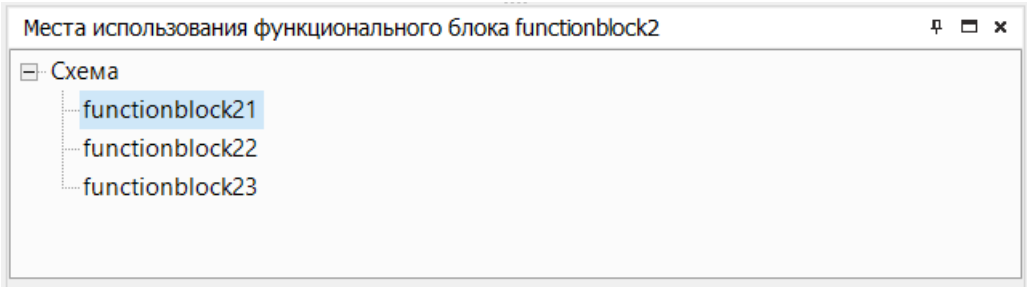


Рисунок 7.106

3. Нажать ПКМ на строку с местом использования функционального блока.

4. Выбрать пункт **Перейти к месту использования**.

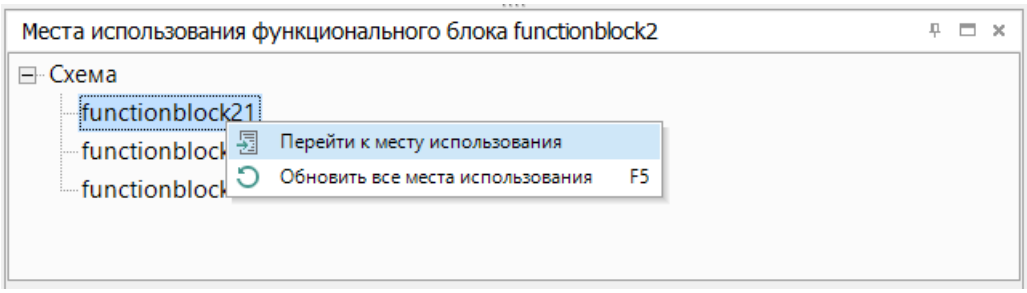


Рисунок 7.107

Фокус сместится в места использования функционального блока на схеме или в редакторе.

ПРИМЕЧАНИЕ
Двойной клик ЛКМ приводит к тому же результату.

Если в ходе работы с программой места использования функциональных блоков изменились, следует обновит панель **Места использования функционального блока**:

1. Нажать ПКМ на любую строку панели.

2. Выбрать пункт **Обновить все места использования**.

Переход к редактору функционального блока

Для перехода к Редактору функционального блока 4.12.2 следует:

1. Нажать ПКМ на имя функции.
2. Выбрать пункт **Редактировать функциональный блок**.

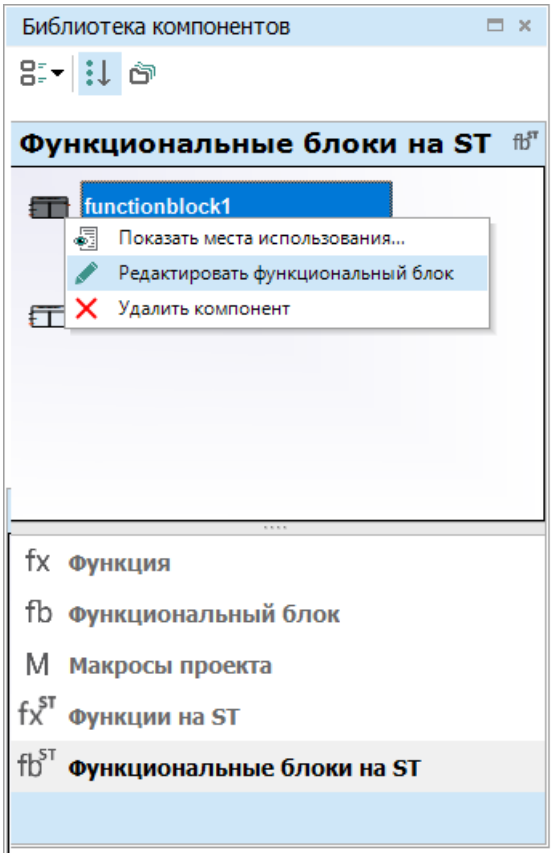


Рисунок 7.108

Откроется редактор функционального блока 4.12.2.

Удаление функции

Для удаления функции из проекта следует:

1. Нажать ПКМ на имя функции.
2. Выбрать пункт **Удалить компонент**.

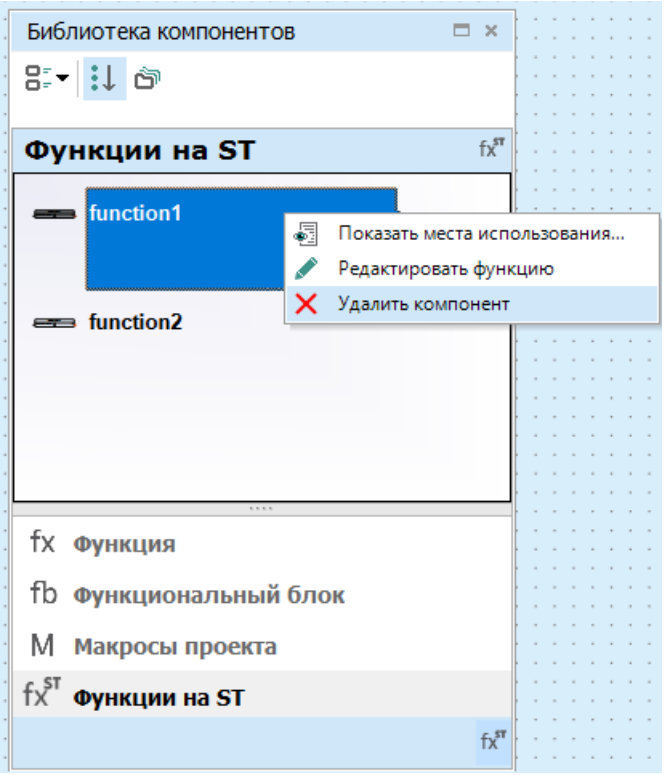


Рисунок 7.109

ПРЕДУПРЕЖДЕНИЕ
Если функция используется на схеме и/или в других функциях, удаление может привести к ошибкам компиляции.

7.6 Элементы управления

- Метка;
- Ввод/вывод (int/float);
- Ввод/вывод (Boolean);
- Динамический текст;
- Комбинированный список (ComboBox).

Элементы используются для оформления дисплея прибора линейки приборов ПР200 и ИПП120.

7.6.1 Метка

Элемент **Метка** предназначен для размещения текстового блока в строке экрана.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения и ввести текст на панели свойств.

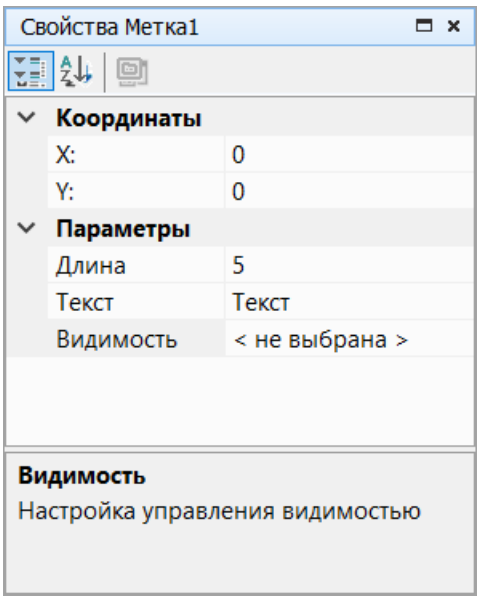


Рисунок 7.110

Координаты

Для расположения первого символа элемента по осям X и Y следует указать координаты. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо от 0 до 15 (в зависимости от свойства **Длина**);
- по оси Y — сверху вниз (в зависимости от количества строк экрана, установленного в редакторе экранов).

Значение координат (целое число) можно задать в выпадающем меню в строке координаты (X и Y настраиваются одинаково):

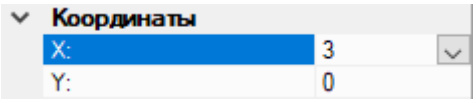


Рисунок 7.111

В выпадающем меню можно выбрать способ задания координат: константа или переменная. Для выбора неизменного расположения элемента следует указать значение координаты в поле ввода.

Для создания бегущей строки следует выбрать вариант **Переменная** и нажать кнопку **Выбор**. Откроется таблица переменных 6, в ней следует выбрать переменную целочисленного типа, которая будет определять значение координаты, и нажать кнопку **ОК**. Выбранная переменная будет отображаться в поле ввода:

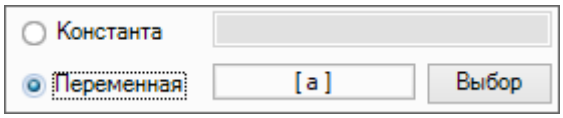


Рисунок 7.112

Параметры

- **Текст** — в поле вводится текст, который будет отображаться на экране прибора. Длина текста не должна превышать значение, указанное для параметра **Длина**.
- **Длина** — максимальное количество отображаемых символов блока.
- **Видимость** — к параметру привязывается булевская переменная, которая определяет видимость элемента.

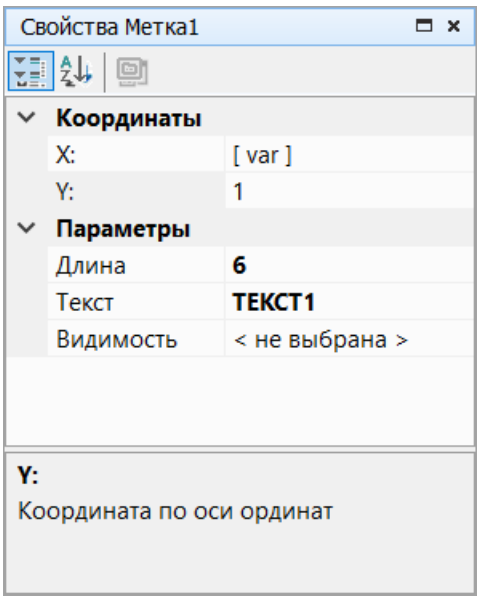


Рисунок 7.113

Отображение

В редакторе экранов отображается, как будет выглядеть элемент **Метка** на экране прибора.

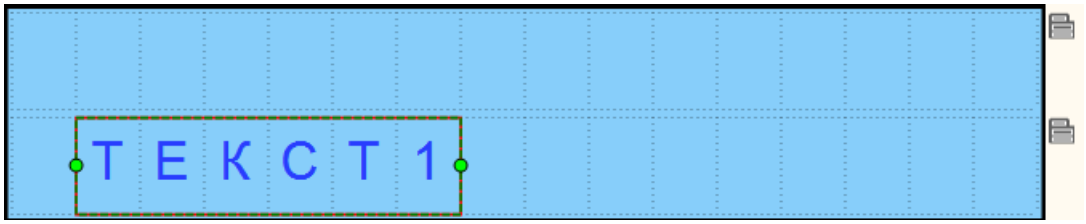


Рисунок 7.114

7.6.2 Ввод/вывод (int/float)

Элемент **Ввод/вывод (int/float)** предназначен для вывода на экран прибора или чтения с экрана значения целочисленной или вещественной переменной.

Настройка свойств

Для корректной работы элемента следует указать координаты расположения и настроить параметры ввода/вывода на панели свойства.

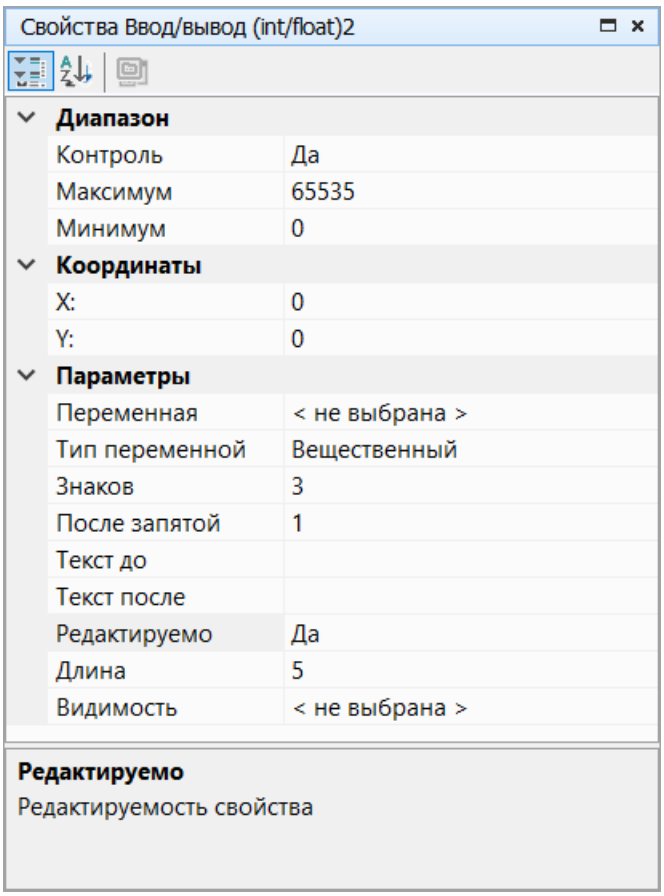


Рисунок 7.115

Диапазон

- **Контроль** — ограничение значения, которое пользователь вводит с лицевой панели прибора. Ограничение не распространяется на значение, полученное в ходе работы программы;
- **Максимум** — максимальное значение параметра для ввода;
- **Минимум** — минимальное значение параметра для ввода.

Координаты

Для расположения первого символа элемента по осям X и Y следует указать координаты. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо от 0 до 15 (в зависимости от свойства **Длина**);
- по оси Y — сверху вниз (в зависимости от количества строк экрана, установленного в редакторе экранов).

Значение координат (целое число) можно задать в выпадающем меню в строке координаты (X и Y настраиваются одинаково):

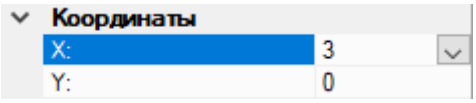


Рисунок 7.116

В выпадающем меню можно выбрать способ задания координат: константа или переменная. Для выбора неизменного расположения элемента следует указать значение координаты в поле ввода.

Для создания бегущей строки следует выбрать вариант **Переменная** и нажать кнопку **Выбор**. Откроется таблица переменных 6, в ней следует выбрать переменную целочисленного типа, которая будет определять значение координаты, и нажать кнопку **OK**. Выбранная переменная будет отображаться в поле ввода:

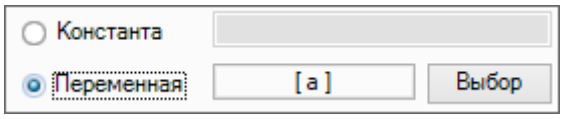


Рисунок 7.117

Параметры

- **Переменная** — привязка к переменной проекта (целочисленной или с плавающей точкой). Для выбора переменной следует нажать кнопку «**…**» и выбрать в таблице переменных 6;
- **Тип переменной** — выбор типа переменной: целочисленный или вещественный. Если переменная еще не привязана, то необходимо задать тип;
- **Знаков** — суммарное количество отображаемых знаков до и после запятой;
- **После запятой** — количество знаков после запятой: до 6 знаков или определяется автоматически (Авто). Более подробную информацию читайте в *руководстве по эксплуатации* прибора;
- **Текст до** — текст слева от числового значения;
- **Текст после** — текст справа от числового значения;
- **Редактируемо** — если выбрано **Да**, то значение привязанной переменной можно изменять с помощью кнопок на лицевой панели прибора;
- **Длина** — общая максимальная длина поля, включающая текст до и после значения переменной и само значение переменной.
- **Видимость** — к параметру привязывается булевская переменная, которая определяет видимость элемента.

Диапазон	
Контроль	Да
Максимум	65535
Минимум	0
Координаты	
X:	0
Y:	0
Параметры	
Переменная	**[var1]**
Тип переменной	Вещественный
Знаков	3
После запятой	1
Текст до	**T=**
Текст после	**C**
Редактируемо	Да
Длина	**8**
Видимость	< не выбрана >

Рисунок 7.118

Отображение

В редакторе экранов отображается, как будет выглядеть элемент **Ввод/вывод (int/float)** на экране прибора.

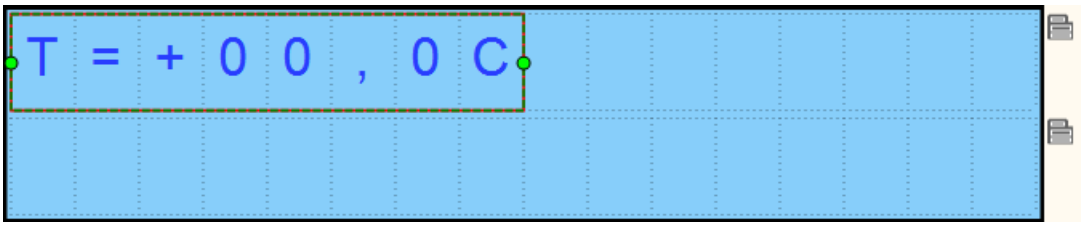


Рисунок 7.119

7.6.3 Ввод/вывод (boolean)

Элемент **Ввод/вывод (boolean)** предназначен для вывода на экран или чтения с экрана значения булевской переменной.

Настройка свойств

Для корректной работы элемента следует указать координаты расположения и настроить параметры ввода/вывода на панели свойства.

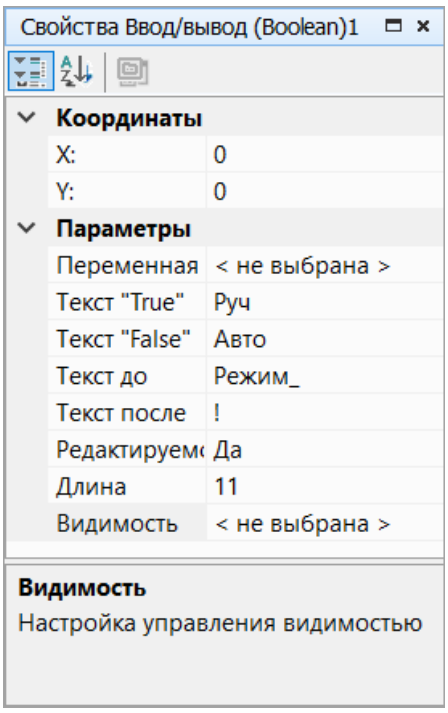


Рисунок 7.120

Координаты

Для расположения первого символа элемента по осям X и Y следует указать координаты. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо от 0 до 15 (в зависимости от свойства **Длина**);
- по оси Y — сверху вниз (в зависимости от количества строк экрана, установленного в редакторе экранов).

Значение координат (целое число) можно задать в выпадающем меню в строке координаты (X и Y настраиваются одинаково):

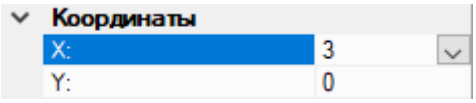


Рисунок 7.121

В выпадающем меню можно выбрать способ задания координат: константа или переменная. Для выбора неизменного расположения элемента следует указать значение координаты в поле ввода.

Для создания бегущей строки следует выбрать вариант **Переменная** и нажать кнопку **Выбор**. Откроется таблица переменных 6, в ней следует выбрать переменную целочисленного типа, которая будет определять значение координаты, и нажать кнопку **ОК**. Выбранная переменная будет отображаться в поле ввода:

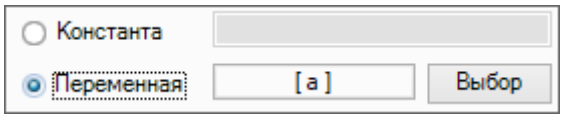


Рисунок 7.122

Параметры

- **Переменная** — привязка к булевской переменной проекта. Для выбора переменной следует нажать кнопку «**…**» и выбрать в таблице переменных 6;
- **Текст «True»** — текст, соответствующий значению переменной, равной «1»;
- **Текст «False»** — текст, соответствующий значению переменной, равной «0»;
- **Текст до** — текст слева от выводимого значения;
- **Текст после** — текст справа от выводимого значения;
- **Редактируемо** — если выбрано значение **Да**, то значение привязанной переменной можно изменять с помощью кнопок на лицевой панели прибора;
- **Длина** — общая максимальная длина поля, включающая текст до и после значения переменной и само значение переменной.

- **Видимость** — к параметру привязывается булевская переменная, которая определяет видимость элемента.

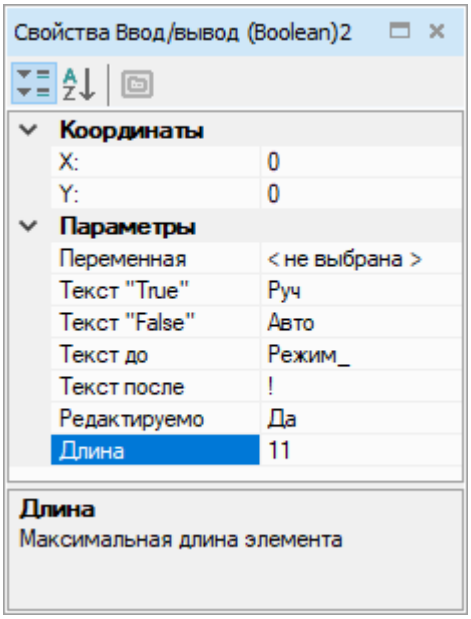


Рисунок 7.123

Отображение

В редакторе экранов отображается, как будет выглядеть элемент **Ввод/вывод (boolean)** на экране прибора.

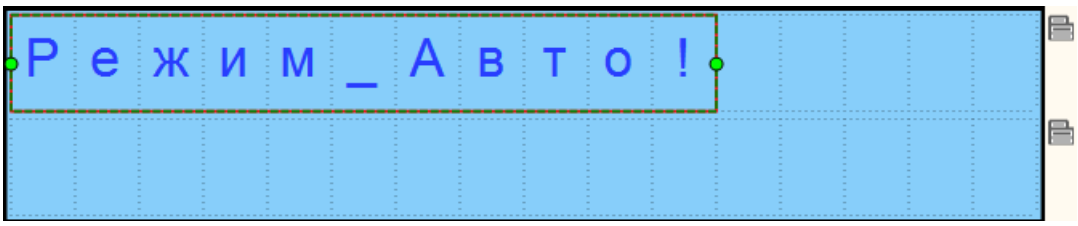


Рисунок 7.124

7.6.4 Динамический текст

Элемент **Динамический текст** предназначен для вывода на экран прибора одной из нескольких текстовых строк в зависимости от значения привязанной переменной.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения и ввести текст на панели свойств.

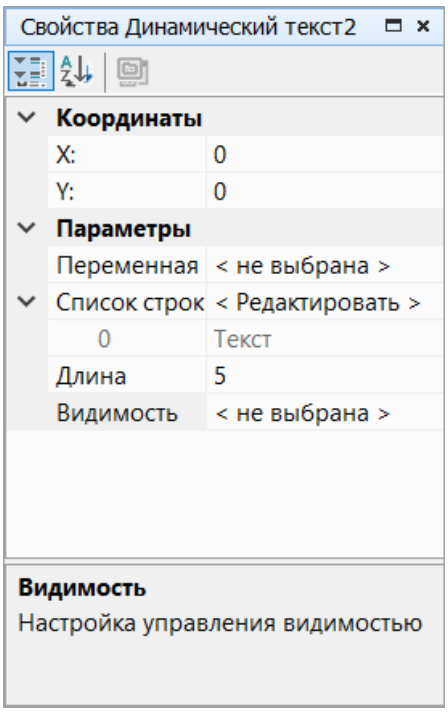


Рисунок 7.125

Координаты

Для расположения первого символа элемента по осям X и Y следует указать координаты. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо от 0 до 15 (в зависимости от свойства **Длина**);
- по оси Y — сверху вниз (в зависимости от количества строк экрана, установленного в редакторе экранов).

Значение координат (целое число) можно задать в выпадающем меню в строке координаты (X и Y настраиваются одинаково):

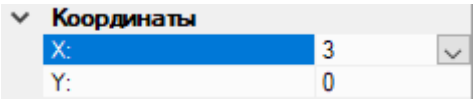


Рисунок 7.126

В выпадающем меню можно выбрать способ задания координат: константа или переменная. Для выбора неизменного расположения элемента следует указать значение координаты в поле ввода.

Для создания бегущей строки следует выбрать вариант **Переменная** и нажать кнопку **Выбор**. Откроется таблица переменных, в ней следует выбрать переменную целочисленного типа, которая будет определять значение координаты, и нажать кнопку **OK**. Выбранная переменная будет отображаться в поле ввода:

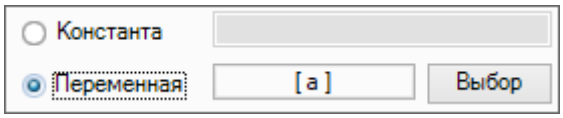


Рисунок 7.127

Параметры

- **Переменная** — привязка к целочисленной переменной проекта. Для выбора переменной следует нажать кнопку «**…**» и выбрать в таблице переменных;
- **Список строк** — таблица, **Значение** строки которой выводится на экран при соответствии номера (**№**) строки значению привязанной целочисленной переменной (**Переменная**). В столбце **Символов** указывается количество символов в строке. Пиктограмма ⊗ выводится в случае превышения значения параметра **Длина**, при наведении курсора на пиктограмму отобразится подсказка с описанием ошибки.

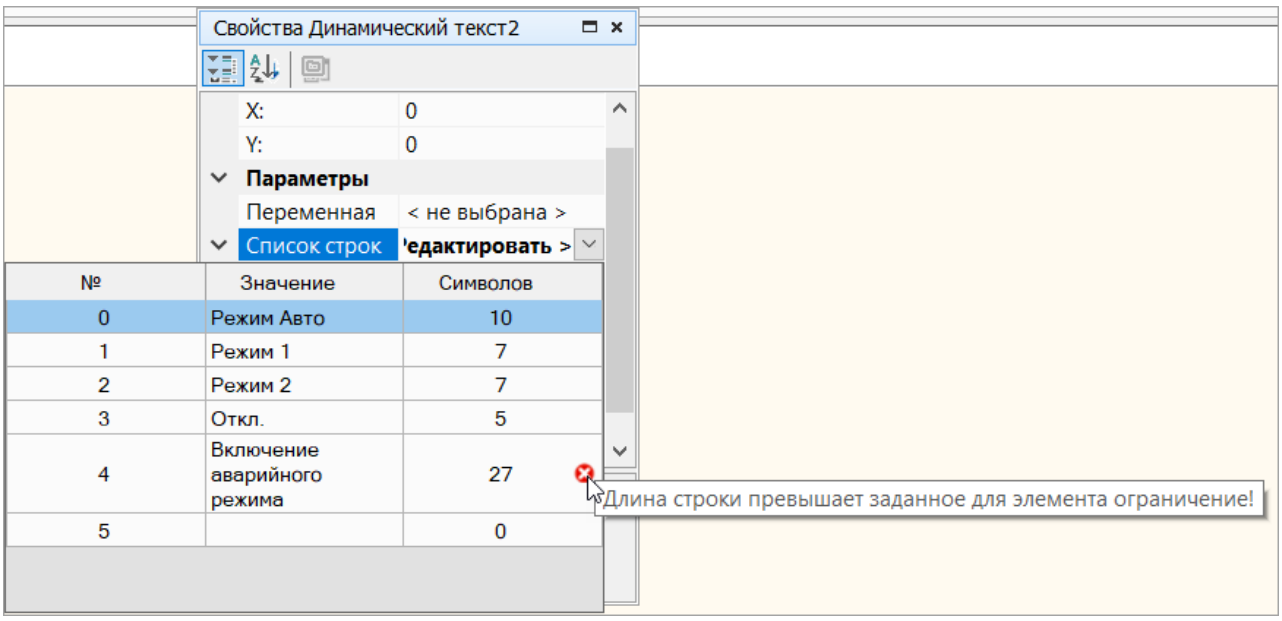


Рисунок 7.128

- **Длина** — максимальная длина поля. В случае превышения длины текст будет отображаться не полностью.
- **Видимость** — к параметру привязывается булевская переменная, которая определяет видимость элемента.

Отображение

В редакторе экранов отображается, как будет выглядеть элемент **Динамический текст** на экране прибора.

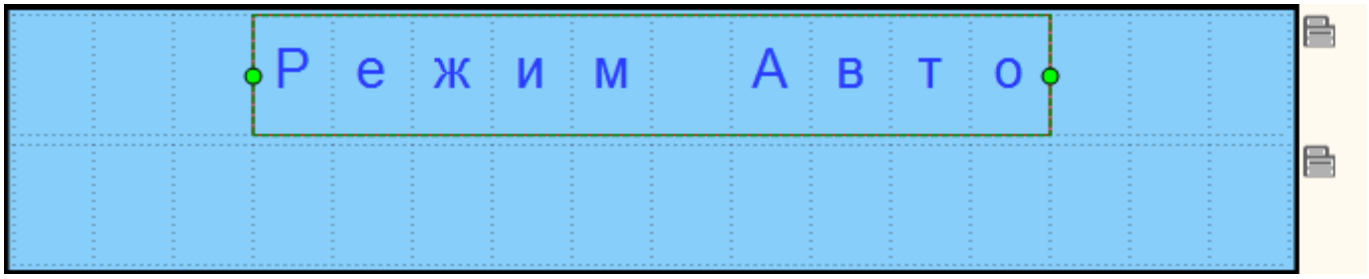


Рисунок 7.129

7.6.5 Комбинированный список (ComboBox)

Элемент **Комбинированный список (ComboBox)** используется для выбора одной строки из нескольких текстовых строк кнопками прибора и записи в привязанную переменную значения, соответствующего выбору.

Настройка свойств

Для корректной работы элемента следует указать координаты расположения и настроить параметры переменных и строк на панели свойства.

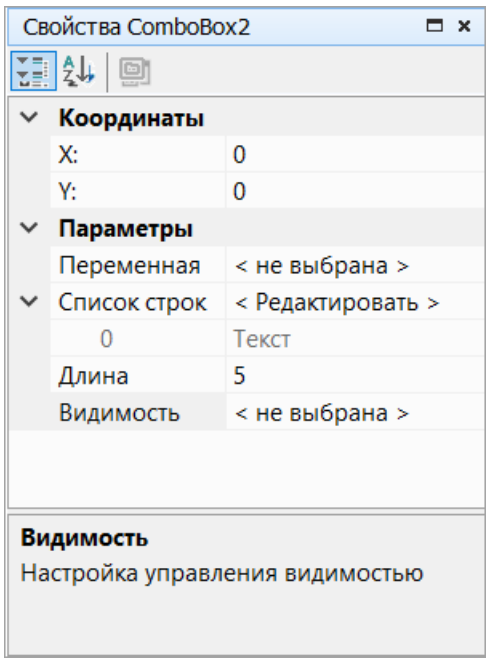


Рисунок 7.130

Координаты

Для расположения первого символа элемента по осям X и Y следует указать координаты. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо от 0 до 15 (в зависимости от свойства **Длина**);
- по оси Y — сверху вниз (в зависимости от количества строк экрана, установленного в редакторе экранов).

Значение координат (целое число) можно задать в выпадающем меню в строке координаты (X и Y настраиваются одинаково):

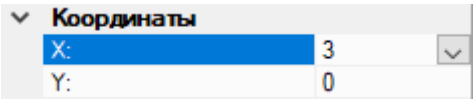


Рисунок 7.131

В выпадающем меню можно выбрать способ задания координат: константа или переменная. Для выбора неизменного расположения элемента следует указать значение координаты в поле ввода.

Для создания бегущей строки следует выбрать вариант **Переменная** и нажать кнопку **Выбор**. Откроется таблица переменных, в ней следует выбрать переменную целочисленного типа, которая будет определять значение координаты, и нажать кнопку **OK**. Выбранная переменная будет отображаться в поле ввода:

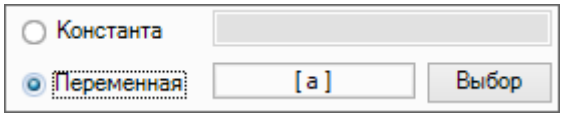


Рисунок 7.132

Параметры

- **Переменная** — привязка к целочисленной переменной проекта. Для выбора переменной следует нажать кнопку «**...**» и выбрать в таблице переменных.
- **Список строк** — таблица, номер (**№**) каждой строки которой служит для записи в привязанную целочисленную переменную при выборе **Значения** на дисплее прибора. В столбце **Символов** указывается количество символов в строке. Пиктограмма выводится в случае превышения значения параметра **Длина**, при наведении курсора на пиктограмму отобразится подсказка с описанием ошибки.

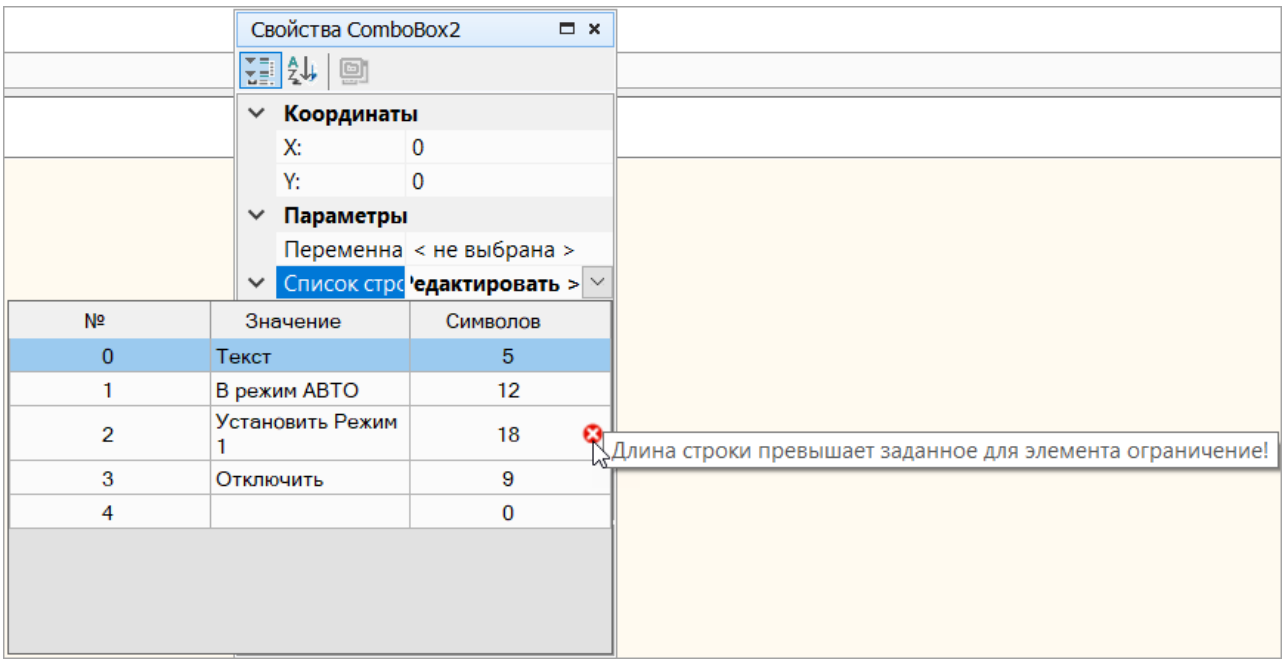


Рисунок 7.133

- **Длина** — максимальная длина поля. В случае превышения длины текст будет отображаться не полностью.
- **Видимость** — к параметру привязывается булевская переменная, которая определяет видимость элемента.

Отображение

В редакторе экранов отображается, как будет выглядеть элемент **Комбинированный список (ComboBox)** на экране прибора.

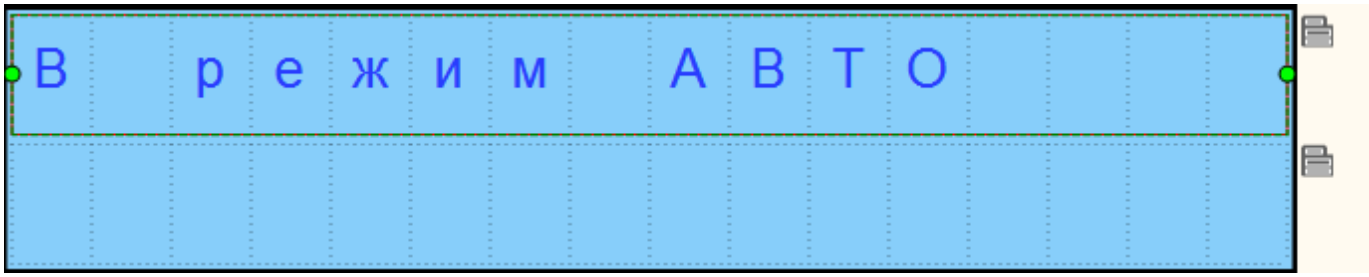


Рисунок 7.134

7.7 Базовые элементы

- Текст;
- Индикатор;
- Индикатор выполнения;
- Динамический текст;
- Ввод/вывод int/float;
- Ввод/вывод времени;
- Ввод/вывод IP;
- Линия;
- Многоугольник;
- Круг;
- Кнопка;
- Переключатель;
- Группа переключателей;
- Изображение;
- Меню;
- График;
- Стрелочный индикатор.

Данные элементы используются для создания графического интерфейса взаимодействия пользователя с логикой выполняемой программы на приборах с графическим цветным ЖКИ. Если на экран добавлено

несколько доступных для редактирования элементов визуализации, то порядок их выбора при помощи кнопки SEL будет определяться порядком в списке «Компоненты экрана» снизу вверх (первыми будут выделяться нижние элементы):

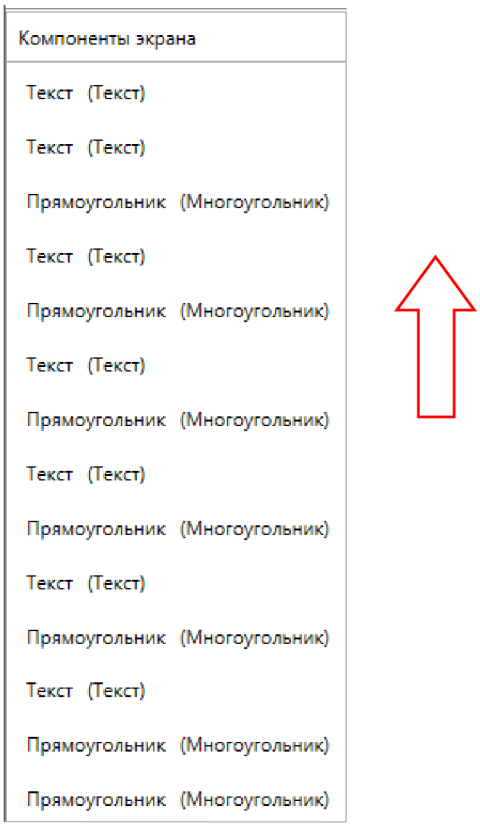


Рисунок 7.135

Изменить порядок элементов можно перетащив (drag & drop) элемент в списке Компоненты экрана или при помощи контекстного меню, нажав ПКМ на элемент на экране:

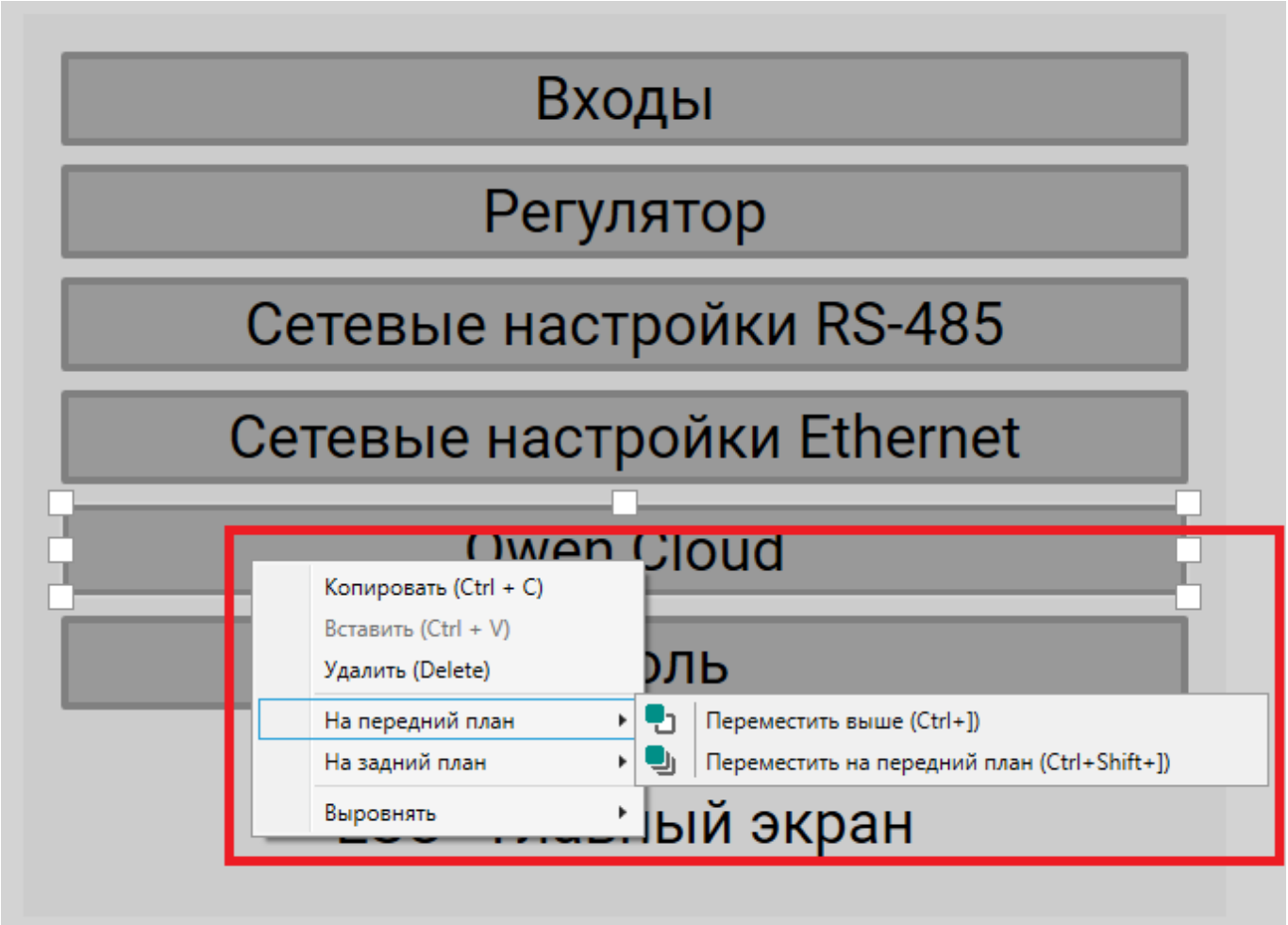


Рисунок 7.136

Ниже приведены данные о размере оперативной памяти (ОЗУ визуал.), требуемой для отображения элементов визуализации, для приборов с графическим цветным ЖКИ:

Элемент визуализации	**Занимаемое ОЗУ визуал., байт**	**Дополнительно**	**Примечание**
Пустой экран	252		
Физические кнопки	80	Расчет ОЗУ: 80 байт (каждая кнопка) + 20 байт * количество переменных + 10 байт*количество действий + 10 байт (обязательный запас) Примеры расчёта: 1. Изменение переменной по короткому нажатию = 80 +20*1+10*1+10 = 120 байт 2. Переход на экран по короткому нажатию кнопки = 80+20*0+10*1+10 = 100 байт 3. Изменение переменной по короткому и длинному нажатию и переход на экран по короткому = 80 +20*2+10*3+10 = 160 байт	
Текст	464	Если задана видимость по переменной: +20 байт за привязку переменной видимости	

Элемент визуализации	Занимаемое ОЗУ визуал., байт	Дополнительно	Примечание
Индикатор	Размер по умолчанию, без рамки – 630	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости	Объем занимаемой ОЗУ напрямую зависит от размеров элемента и толщины рамки: • Чем больше размеры элемента, тем больший объем памяти требуется для его отображения; • Рамка 0 px (отключена) - минимальный объем занимаемой ОЗУ для элемента данного размера; • Рамка 1 px - объем занимаемой ОЗУ увеличивается по сравнению с элементом без рамки; • Рамка 2 px и более - объем занимаемой ОЗУ увеличивается по сравнению с элементом с рамкой 1 px. При толщине рамки более 2 px объем занимаемой ОЗУ не увеличивается и соответствует объему, занимаемому элементом с рамкой толщиной 2 px
Индикатор выполнения	408	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Видимость значения: +16 байт за отображение показаний (Значение, видимость = Да)	

Элемент визуализации		Занимаемое ОЗУ визуал., байт	Дополнительно	Примечание
Динамический текст		836 (для одной строки)	+4 байта за каждую дополнительную строку Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Контроль ввода: +12 байт за доступ к редактированию (Запретить редактирование = Нет)	
Ввод/вывод int/float		776	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Контроль ввода: +24 байт за доступ к редактированию (Запретить редактирование = Нет)	
Ввод вывод времени		688 только дата **или** только время 704 дата **и** время	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Контроль ввода: +28 байт за доступ к редактированию (Запретить редактирование = Нет)	
Ввод/вывод IP адреса		676	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Контроль ввода: +24 байт за доступ к редактированию (Запретить редактирование = Нет)	
Линия	прямая	182	Если задана видимость по переменной: +20 байт за привязку переменной видимости	
	ломаная	190		

Элемент визуализации		Занимаемое ОЗУ визуал., байт	Дополнительно	Примечание
Многоугольник	Прямоугольник	338	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Заливка: +132 байта (Внешний вид, заливка = Да) Настройка действий (переход на экран или изменение переменной) = 10 байт * количество действий + 26 байт	
	Треугольник	334		
	Ромб	338		
Круг	Круг целиком	Размер по умолчанию, без заливки — 978 Размер по умолчанию, с заливкой — 1346	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Настройка действий (переход на экран или изменение переменной) = 10 байт * количество действий + 26 байт	Объем занимаемой ОЗУ напрямую зависит от размеров элемента и наличия заливки: • Чем больше размеры элемента, тем больший объем памяти требуется для его отображения; • Объем занимаемой ОЗУ элемента с заливкой больше, чем у элемента без заливки. При определенных комбинациях настроек параметр "Толщина" у рамки фокуса может незначительно влиять на занимаемое ОЗУ визуализации
	Сектор круга	Размер по умолчанию, без заливки — 1502 Размер по умолчанию, с заливкой — 1878		
Кнопка		Прямоугольная без скруглений, без рамки — 894	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости Настройка действий (переход на экран или изменение переменной) = 10 байт * количество действий + 26 байт +6 байт за добавление рамки толщиной 1px (для прямоугольной кнопки без скруглений) +20 байт за добавление рамки толщиной 2 px и более (для прямоугольной кнопки без скруглений)	Для круглой кнопки и прямоугольной со скруглением объем занимаемой ОЗУ напрямую зависит от размеров элемента: • Чем больше размеры элемента, тем больший объем памяти требуется для его отображения. При определенных комбинациях настроек параметр "Толщина" у рамки фокуса может незначительно влиять на занимаемое ОЗУ визуализации

Элемент визуализации	Занимаемое ОЗУ визуал., байт	Дополнительно	Примечание
Переключатель	Размер по умолчанию, без рамки — 1192	Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости	Объем занимаемой ОЗУ напрямую зависит от размеров элемента: • Чем больше размеры элемента, тем больший объем памяти требуется для его отображения. При определенных комбинациях настроек параметр "Толщина" у рамки фокуса может незначительно влиять на занимаемое ОЗУ визуализации
Группа переключателей	1688 (для группы с одной ячейкой)	+ 208 байт за каждую ячейку Если используется переменная: +20 байт за переменную Если задана видимость по переменной: +20 байт за привязку переменной видимости	
Изображение	372	Если задана видимость по переменной: +20 байт за привязку переменной видимости Настройка действий (переход на экран или изменение переменной) = 10 байт * количество действий + 26 байт + 32 байта за добавление рамки	Расчет заполнения пользовательского ПЗУ визуализации = Ширина*Высота*2. В расчете участвует ширина и высота исходного изображения (в пикселях), загруженного в редактор визуализации В случае копирования и привязки одного и того же изображения к нескольким элементам визуализации, дополнительный объем пользовательского ПЗУ визуализации не занимается
Меню	1870 (для меню с одним пунктом)	+ 298 байт за каждую строку Если задана видимость по переменной: +20 байт за привязку переменной видимости элемента или строки	

Элемент визуализации		Занимаемое ОЗУ визуал., байт	Дополнительно	Примечание
График	без работы в фоне	Без перьев, масштаб времени 2 сек — 982	Дополнительно за увеличение времени отображения: От 2 байт на 3 секундах до 916 байт на 23:59:59 Дополнительно за каждое перо: + 20 байт (первое перо) + каждое последующее в зависимости от времени добавления (от 74 байт на 2 секундах до 990 байт на 23:59:59) Если задана видимость по переменной: +20 байт за привязку переменной видимости элемента или пера	Объем занимаемой ОЗУ напрямую зависит от ширины элемента: • Чем шире элемент, тем больший объем памяти требуется для его отображения
	работа в фоне	Без перьев, масштаб времени 2 сек — 1002	Дополнительно за увеличение времени отображения: От 6 байт на 3 секундах до 2748 байт на 23:59:59 Дополнительно за каждое перо: + 20 байт (первое перо) + каждое последующее в зависимости от времени добавления (от 94 байт на 2 секундах до 2842 байт на 23:59:59) Если задана видимость по переменной: +20 байт за привязку переменной видимости элемента или пера	

7.7.1 Текст

Элемент **Текст** предназначен для размещения текстового блока в экране визуализации.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента, ввести текст и задать другие параметры на панели свойств.

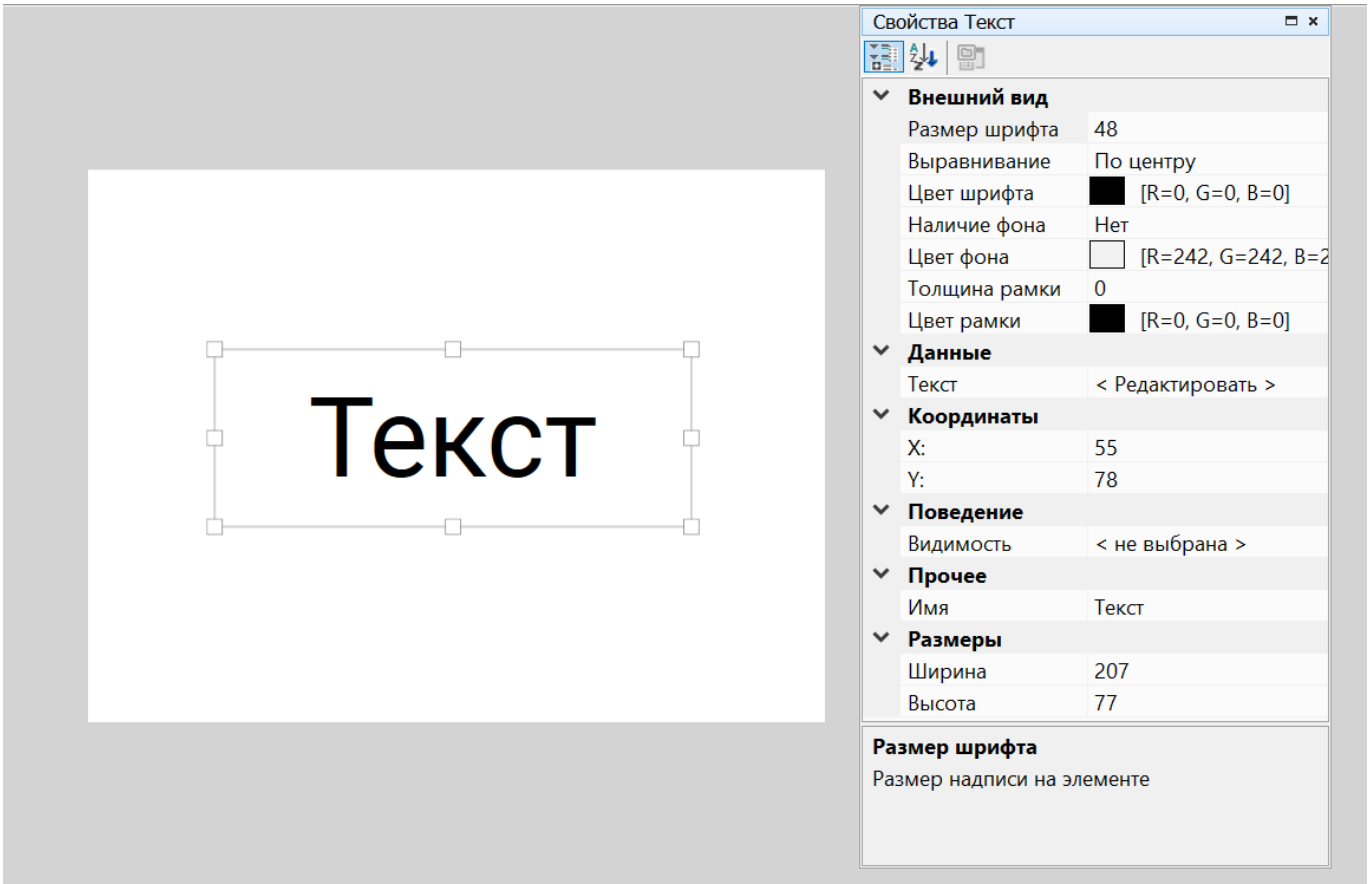


Рисунок 7.137

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Размер шрифта**	Размер отображаемого в элементе текста: 16, 32, 48
	Выравнивание	Выравнивание отображаемого в элементе текста: по левому краю, по центру, по правому краю
	Цвет шрифта	Цвет отображаемого в элементе текста
	Наличие фона	Фон отображаемого в элементе текста
	Цвет фона	Цвет фона отображаемого в элементе текста
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента
Данные	**Текст**	В поле вводится текст, который будет отображаться в элементе
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.2 Индикатор

Элемент **Индикатор** предназначен для отображения значения булевской переменной в экране визуализации. Если значение переменной равно **1**, то индикатор принимает цвет «True». Если значение переменной равно **0**, то индикатор принимает цвет «False».

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры на панели свойств.

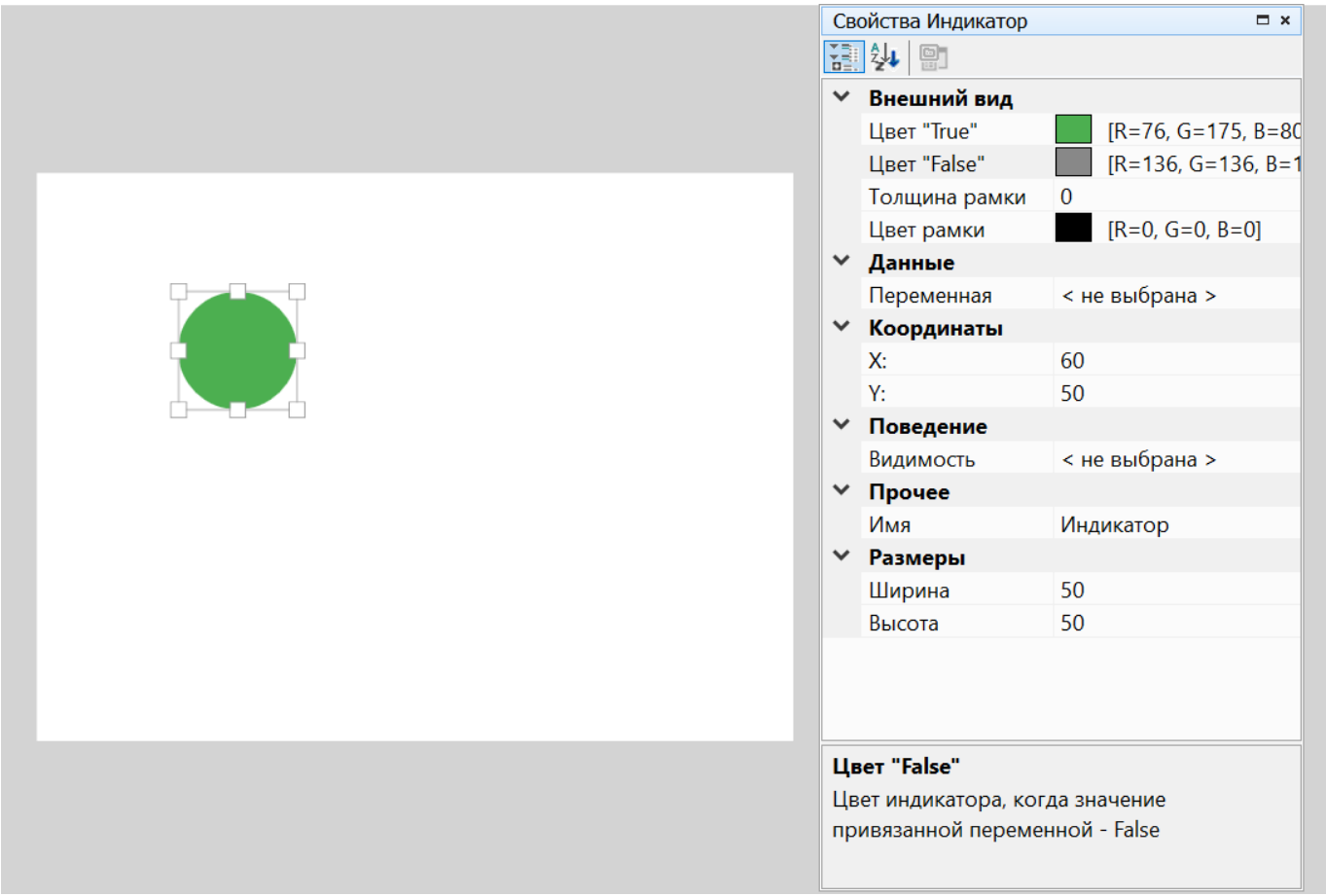


Рисунок 7.138

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Цвет “True”**	Цвет элемента, если значение привязанной переменной равно TRUE
	Цвет “False”	Цвет элемента, если значение привязанной переменной равно FALSE
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента
Данные	**Переменная**	К параметру привязывается булевская переменная для отображения

Группа	Параметр	Описание
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

ПРИМЕЧАНИЕ
Если к параметру **Переменная** не привязана переменная, то цвет элемента всегда будет **True**.

7.7.3 Индикатор выполнения

Элемент **Индикатор выполнения** предназначен для отображения значения целочисленной переменной или переменной с плавающей точкой в виде шкалы на экране визуализации.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры на панели свойств.

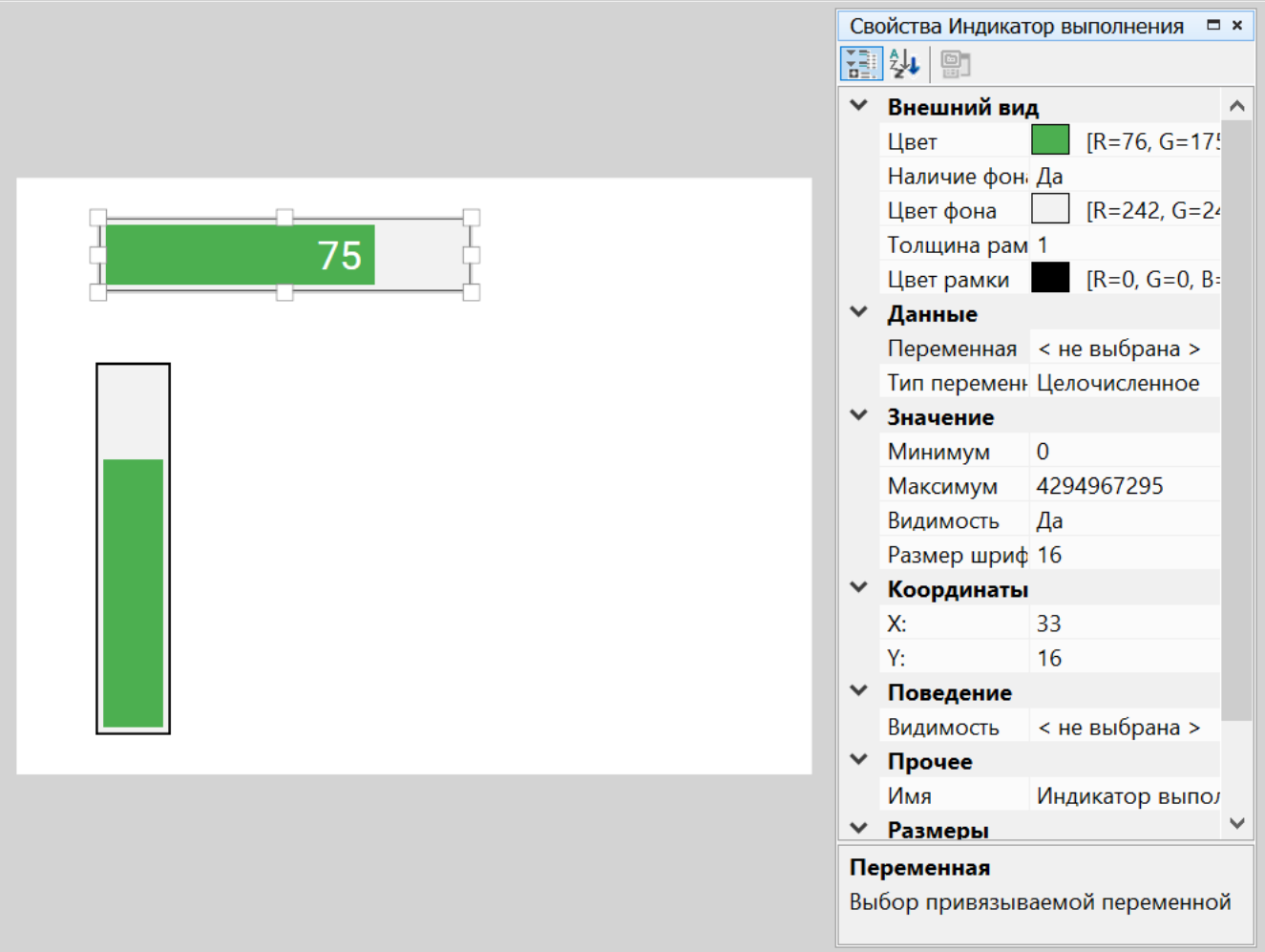


Рисунок 7.139

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана. Для вертикальной ориентации элемента значение Y должно быть больше значения X.

Параметры

Группа	Параметр	Описание
Внешний вид	Цвет	Цвет шкалы элемента
	Наличие фона	Фон отображаемого элемента
	Цвет фона	Цвет фона отображаемого элемента
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента
Данные	Переменная	К параметру привязывается целочисленная переменная или переменная с плавающей точкой для отображения
	Тип переменной	Тип привязанной к элементу переменной: с плавающей точкой, целочисленное
Значение	Минимум	0 % шкалы элемента
	Максимум	100 % шкалы измерения
	Видимость	Отображение значения шкалы в элементе
	Размер шрифта	Размер шрифта отображаемого в элементе текста: 16, 32, 48
Поведение	Видимость	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	Имя	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.4 Динамический текст

Элемент **Динамический текст** предназначен для выбора и отображения значения целочисленной переменной из выпадающего списка на экране визуализации. Целочисленному значению можно задать текстовое обозначение, которое будет отображаться в выпадающем списке.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры и данные на панели свойств.

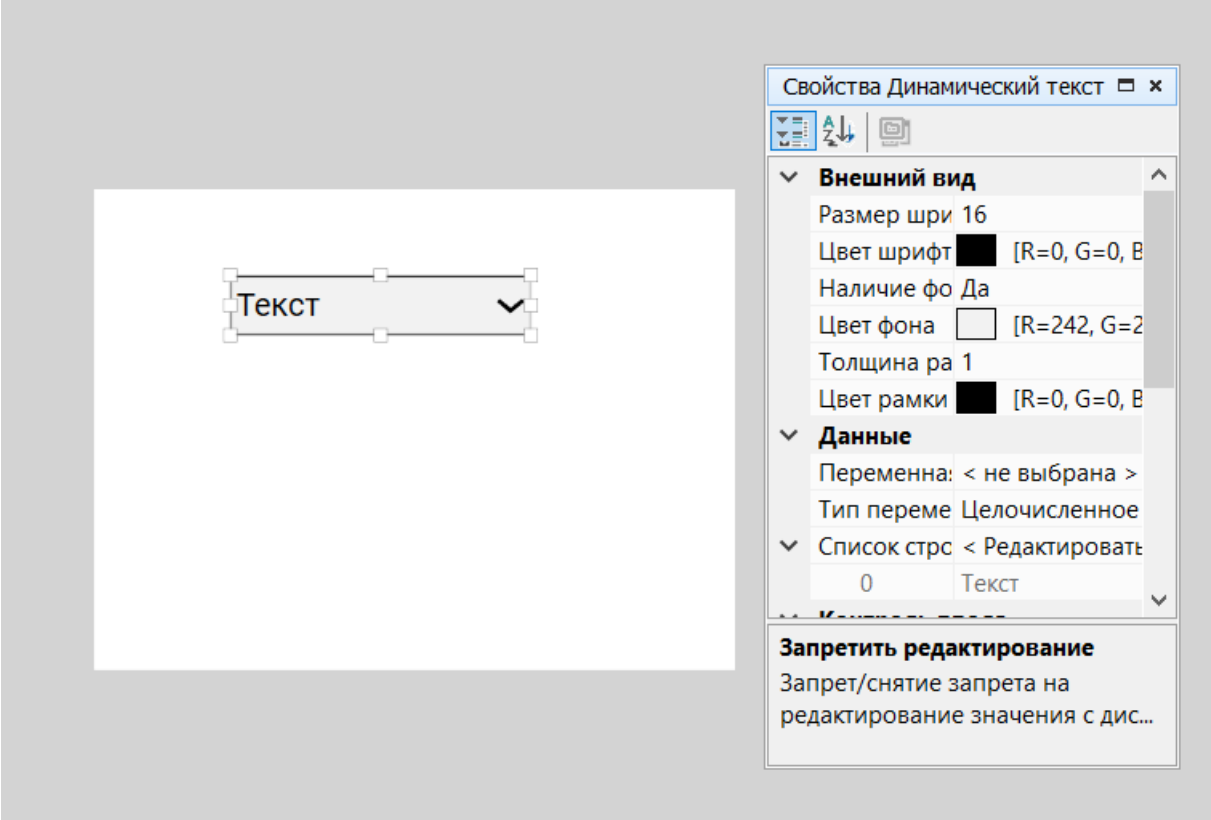


Рисунок 7.140

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Размер шрифта**	Размер отображаемого в элементе текста: 16, 32, 48
	Цвет шрифта	Цвет отображаемого в элементе текста
	Наличие фона	Фон отображаемого в элементе текста
	Цвет фона	Цвет фона отображаемого в элементе текста
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента
Данные	**Переменная**	К параметру привязывается переменная для хранения значений
	Тип переменной	Выбор типа данных привязанной переменной: **целочисленное** и **булевское** **ПРИМЕЧАНИЕ** Если выбран булевский тип переменной, то выпадающий список элемента может содержать только две строки. Если выбран целочисленный тип переменной, то выпадающий список может содержать максимум 128 строк.
	Список строк	В табличном виде значения переменной и соответствующие им наименования, которые будут отображаться в выпадающем списке элемента
Контроль ввода	**Запретить редактирование**	Установка/снятие запрета на изменение значения привязанной переменной с экрана прибора: • если установлено значение **Да**, то значение привязанной переменной можно только просматривать, опция ввода будет недоступна; • если установлено значение **Нет**, то ввод значения в элементе перезаписывает значение переменной
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

Для добавления новой строки в выпадающий список элемента следует:

1. Дважды кликнуть ЛКМ по элементу **Динамический текст** в редакторе экрана.
2. В открывшемся окне кликнуть ПКМ по одной из строк таблице.
3. Выбрать пункт **Добавить строку снизу**.

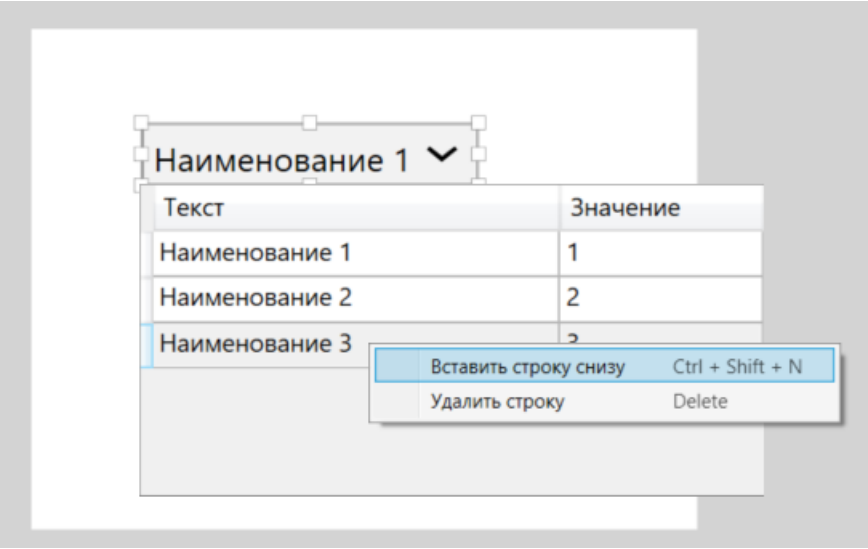


Рисунок 7.141

Пример

Таблица 7.8 – Список строк

0	Стоп
10	Старт
500	Пауза

7.7.5 Ввод/вывод int/float

Элемент **Ввод/вывод int/float** предназначен для ввода и отображения значения целочисленной переменной или переменной с плавающей запятой на экране визуализации.

Настройка свойств

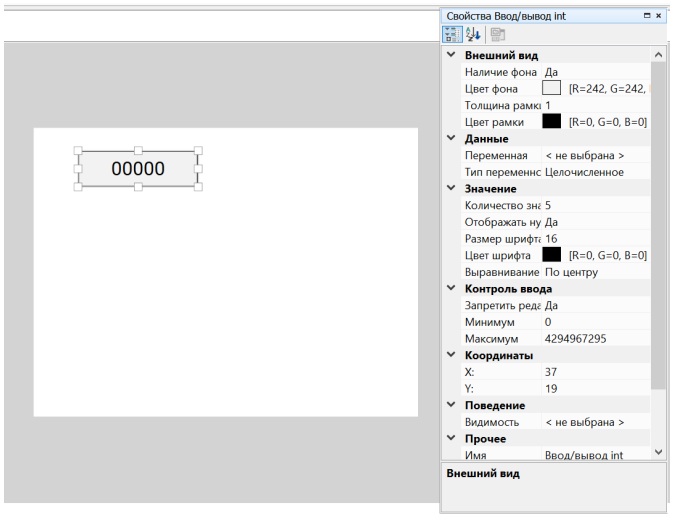


Рисунок 7.142 – Ввод/вывод int

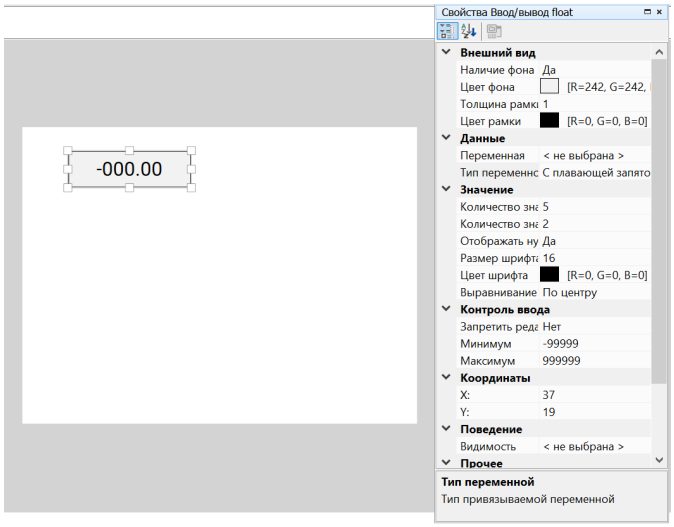


Рисунок 7.143 – Ввод/вывод float

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Наличие фона**	Фон отображаемого в элементе текста	
	Цвет фона	Цвет фона отображаемого в элементе текста	

Группа	Параметр	Описание	Дополнительные возможности
	Толщина рамки	Толщина рамки элемента	
	Цвет рамки	Цвет рамки элемента	
Данные	**Переменная**	К параметру привязывается переменная	
	Тип переменной	Для ввода/вывода целого значения следует указать тип **Целочисленное** Для ввода/вывода значения с плавающей запятой следует указать тип **С плавающей запятой**	
Значение	**Количество знаков**	Количество отображаемых в элементе знаков	При выборе переменной float: не может быть меньше или равно значению **Количество знаков после запятой**
	Количество знаков после запятой **(настройка доступна только при выборе переменной float)**	Количество знаков в дробной части отображаемого в элементе значения	При изменении значения может потребоваться изменение значений **Максимум** и **Минимум** (см. пример после таблицы)
	Отображать нули	Настройка отображения незначащих нулей	
	Размер шрифта	Размер отображаемого в элементе текста: 16, 32, 48	
	Цвет шрифта	Цвет отображаемого в элементе текста	
	Выравнивание	Выравнивание отображаемого в элементе текста	
Контроль ввода	**Запретить редактирование**	Если установлено значение **Да**, то значение привязанной переменной можно только просматривать, опция ввода не будет доступна. В противном случае ввод значения в элементе перезаписывает значение переменной	
	Минимум	Минимальное значение для ввода в элементе	При выборе переменной float: ограничение для значения рассчитывается по формуле: 2147483648 / 10^значение свойства Количество знаков после запятой При изменении значения может потребоваться изменение значения свойства **Количество знаков после запятой** (см. пример после таблицы)

Группа	Параметр	Описание	Дополнительные возможности
	Максимум	Максимальное значение для ввода в элементе	При выборе переменной float: ограничение для значения рассчитывается по формуле: -2147483647 / 10^значение свойства Количество знаков после запятой При изменении значения может потребоваться изменение значения свойства **Количество знаков после запятой** (см. пример после таблицы)
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана
	Высота	Высота элемента по оси Y	

Пример
Для элемента визуализации Ввод/вывод float при 8 отображаемых в элементе знаках необходимо установить отображение 6 знаков после запятой:

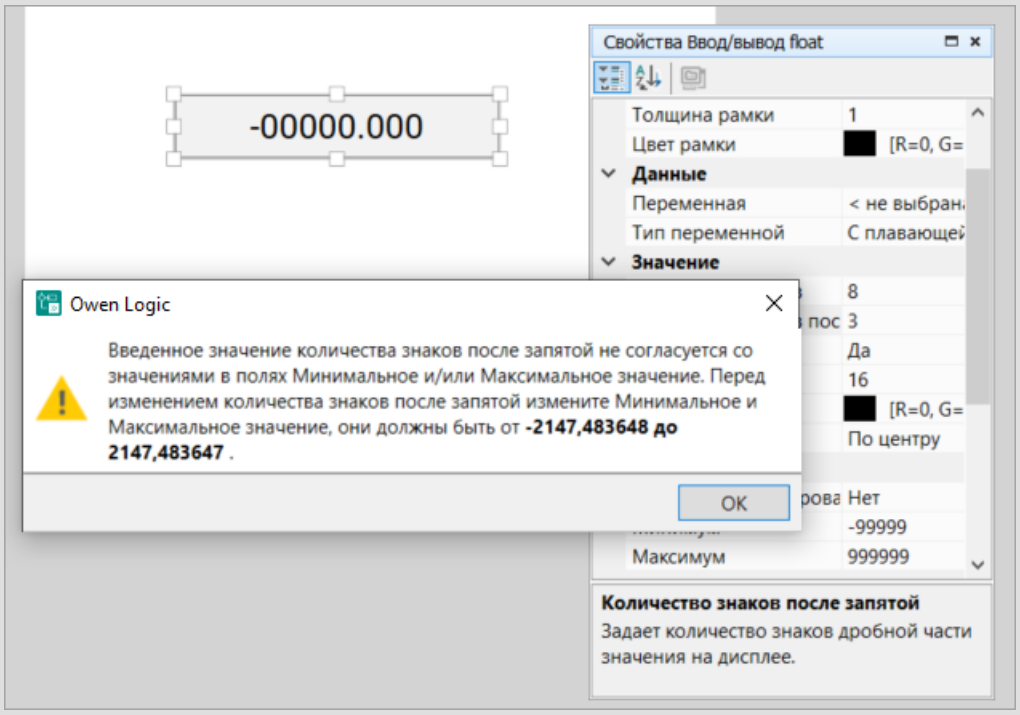


Рисунок 7.144

При выборе значения 6 свойства **Количество знаков после запятой** будет отображена ошибка, с указанием пределов **Минимального** и **Максимального** значения, рассчитанных по формуле:

Значение **Минимум**= -2147483648 / 10^6 = -2147,483648

Значение **Максимум** = 2147483647 / 10^6 = 2147,483647

Введем значение **Минимум** и **Максимум** в рамках указанного диапазона, и выберем значение 6 свойства **Количество знаков после запятой**:

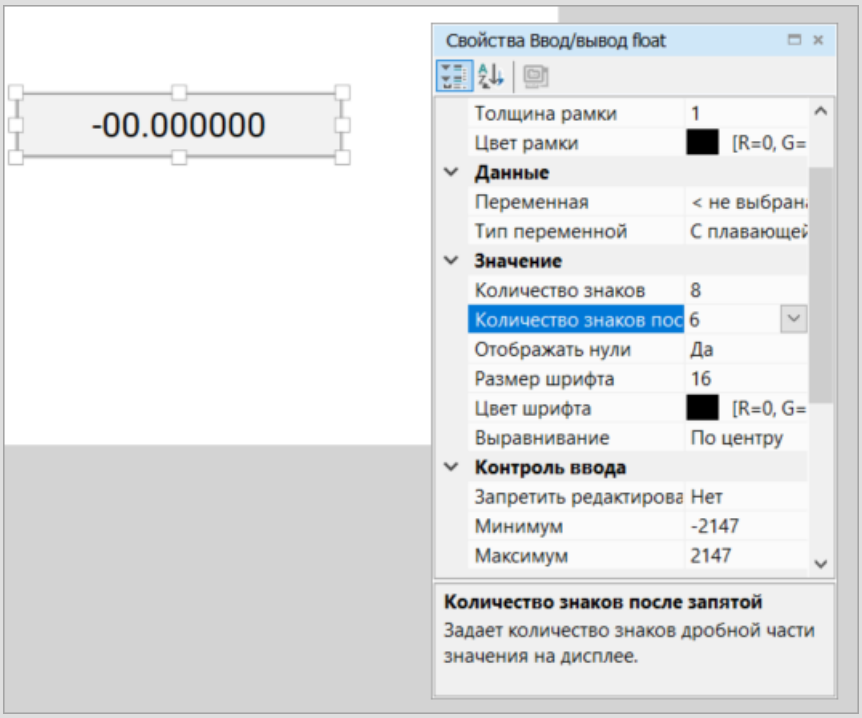


Рисунок 7.145

Значение сменилось, на элементе визуализации отображается 6 знаков после запятой.

При попытке сменить значение **Минимум** или **Максимум**, на значение выходящее за пределы рассчитанного по формуле выше, отобразится сообщение об ошибке:

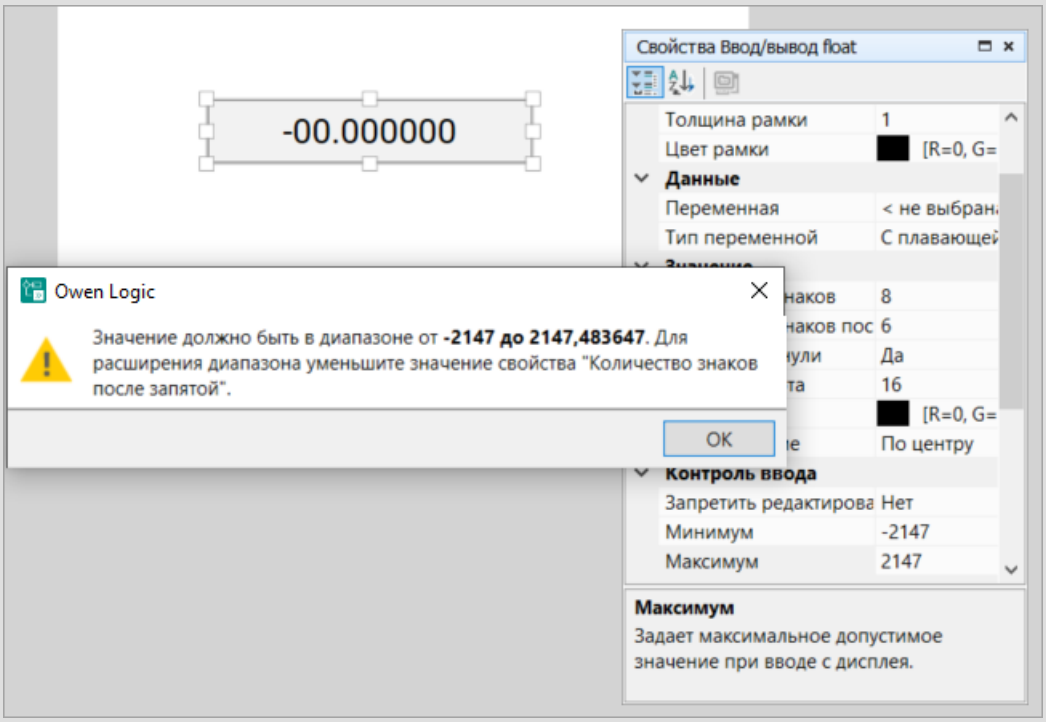


Рисунок 7.146

Свойства **Количество знаков после запятой** и значение **Минимум** и **Максимум** взаимосвязаны.

ПРИМЕЧАНИЕ
Если переменная для вывода выходит за границы, указанные в параметрах **Минимум** / **Максимум**, то на приборе элемент отобразит текущее состояние переменной и выделит красным рамку этого элемента.

7.7.6 Ввод/вывод времени

Элемент **Ввод/вывод времени** предназначен для ввода и отображения значения даты и времени на экране визуализации. Для хранения значения даты и времени используется целочисленная переменная, которая хранит количество секунд с 00:00:00 01 января 2000 г.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры и данные на панели свойств.

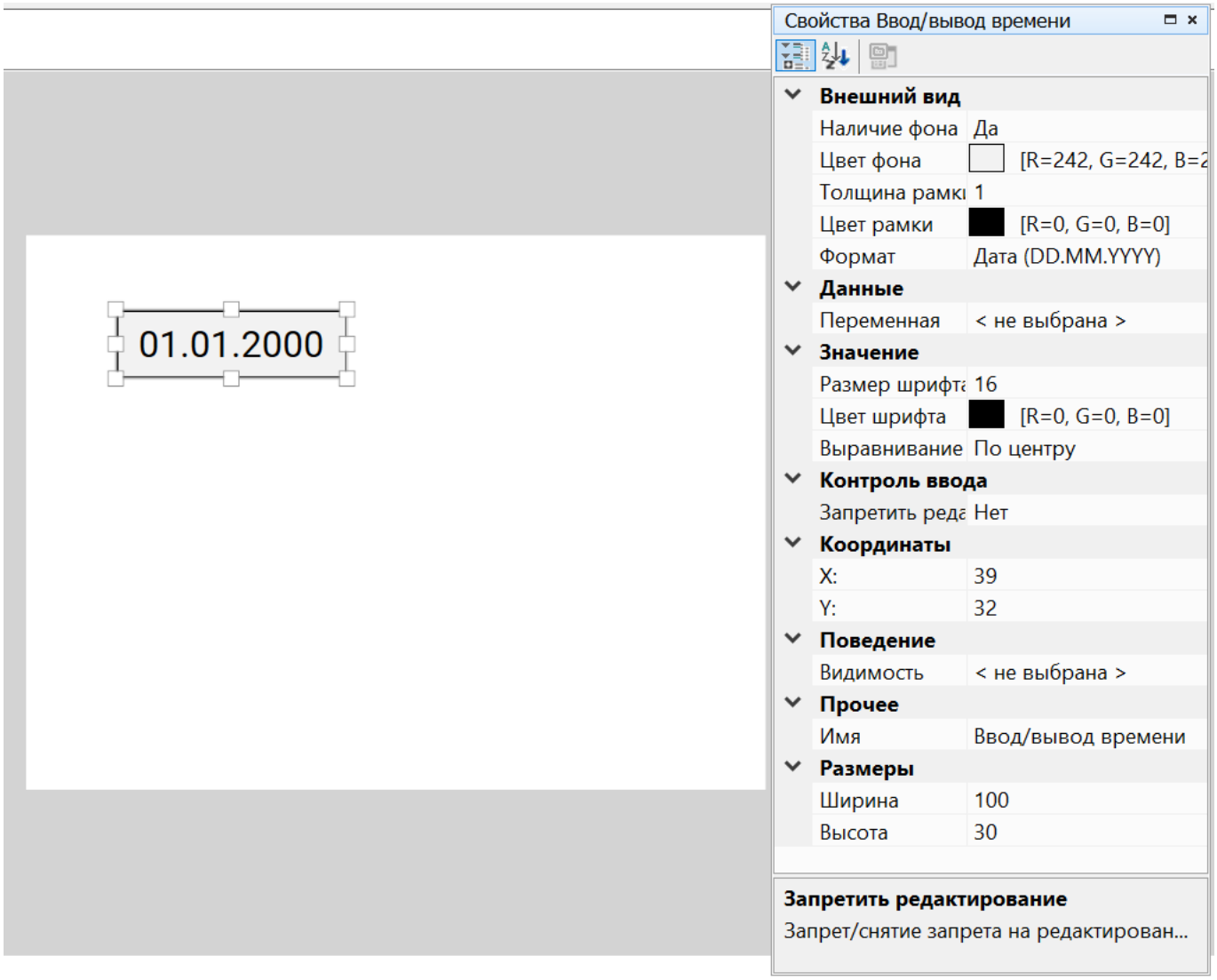


Рисунок 7.147

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Наличие фона**	Фон отображаемого в элементе текста
	Цвет фона	Цвет фона отображаемого в элементе текста
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента

Группа	Параметр	Описание
	Формат	Выбор из выпадающего списка формата отображения даты и времени. Обозначения для формата: • **ss** – секунды; • **mm** – минуты; • **hh** – часы; • **DD** – день; • **MM** – месяц; • **YYYY** – год.
Данные	**Переменная**	К параметру привязывается переменная
Значение	**Размер шрифта**	Размер отображаемого в элементе текста: 16, 32, 48
	Цвет шрифта	Цвет отображаемого в элементе текста
	Выравнивание	Выравнивание отображаемого в элементе текста
Контроль ввода	**Запретить редактирование**	Если установлено значение **Да**, то значение привязанной переменной можно только просматривать, опция ввода не будет доступна. В противном случае ввод значения в элементе перезаписывает значение переменной
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.7 Ввод/вывод IP

Элемент **Ввод/вывод IP** предназначен для ввода и отображения значения IP-адреса (IPv4) на экране визуализации. Для хранения значения IP-адреса используется целочисленная переменная, значение которой рассчитывается по формуле:

$$int = o1 * 256^3 + o2 * 256^2 + o3 * 256^1 + o4 * 256^0,$$

Рисунок 7.148

где *int* – значение целочисленной переменной;

o1, o2, o3, o4 – октеты IP-адреса.

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры и данные на панели свойств.

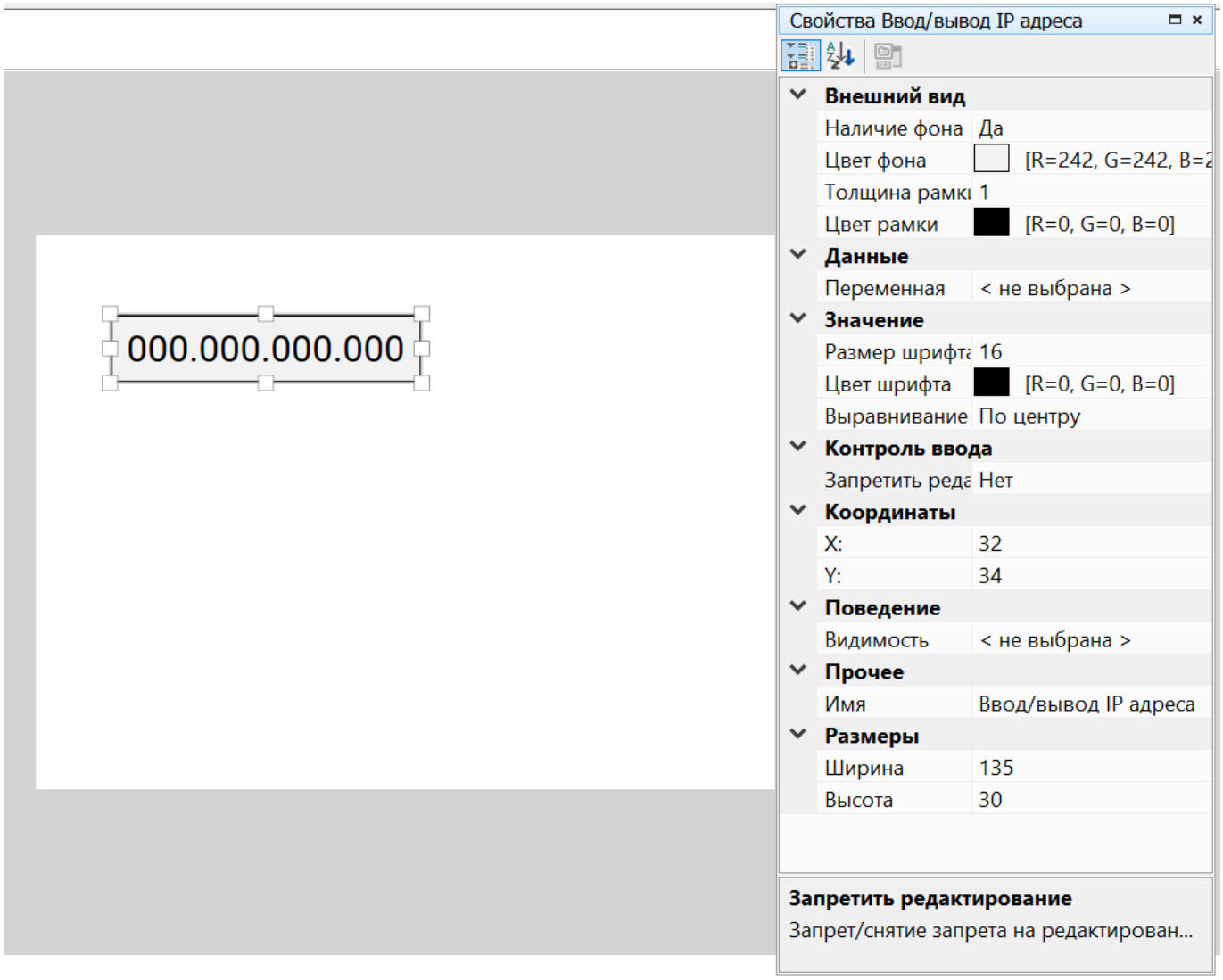


Рисунок 7.149

Координаты

Расположение элемента определяется по осям X и Y. Координаты задаются на панели свойств или перемещением элемента по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Наличие фона**	Фон отображаемого в элементе текста
	Цвет фона	Цвет фона отображаемого в элементе текста
	Толщина рамки	Толщина рамки элемента
	Цвет рамки	Цвет рамки элемента
Данные	**Переменная**	К параметру привязывается переменная
	Тип переменной	Для ввода/вывода целого значения следует указать тип **Целочисленное**
	Размер шрифта	Размер отображаемого в элементе текста: 16, 32, 48
	Цвет шрифта	Цвет отображаемого в элементе текста
	Выравнивание	Выравнивание отображаемого в элементе текста

Группа	Параметр	Описание
Контроль ввода	**Запретить редактирование**	Если установлено значение **Да**, то значение привязанной переменной можно только просматривать, опция ввода не будет доступна. В противном случае ввод значения в элементе перезаписывает значение переменной
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.8 Линия

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры на панели свойств.

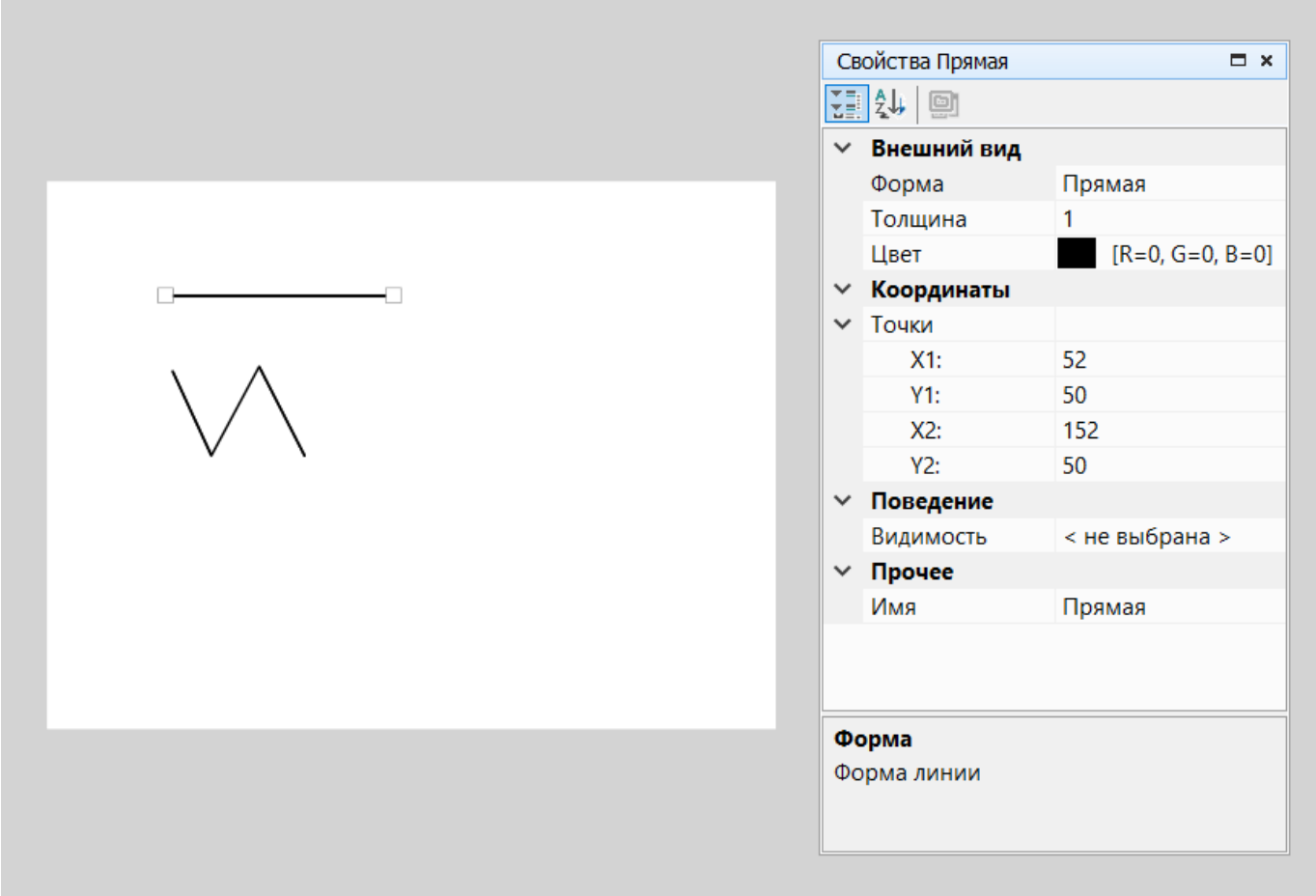


Рисунок 7.150

Координаты

Расположение прямой линии определяется двумя точками, координаты которых по осям X и Y задаются на панели свойств или перемещением точек по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Расположение ломаной линии определяется четырьмя точками.

Параметры

Группа	Параметр	Описание
Внешний вид	**Форма**	Форма отображаемой линии: **прямая**, **ломаная**
	Толщина	Толщина линии

Группа	Параметр	Описание
	Цвет	Цвет элемента
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.9 Многоугольник

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры на панели свойств.

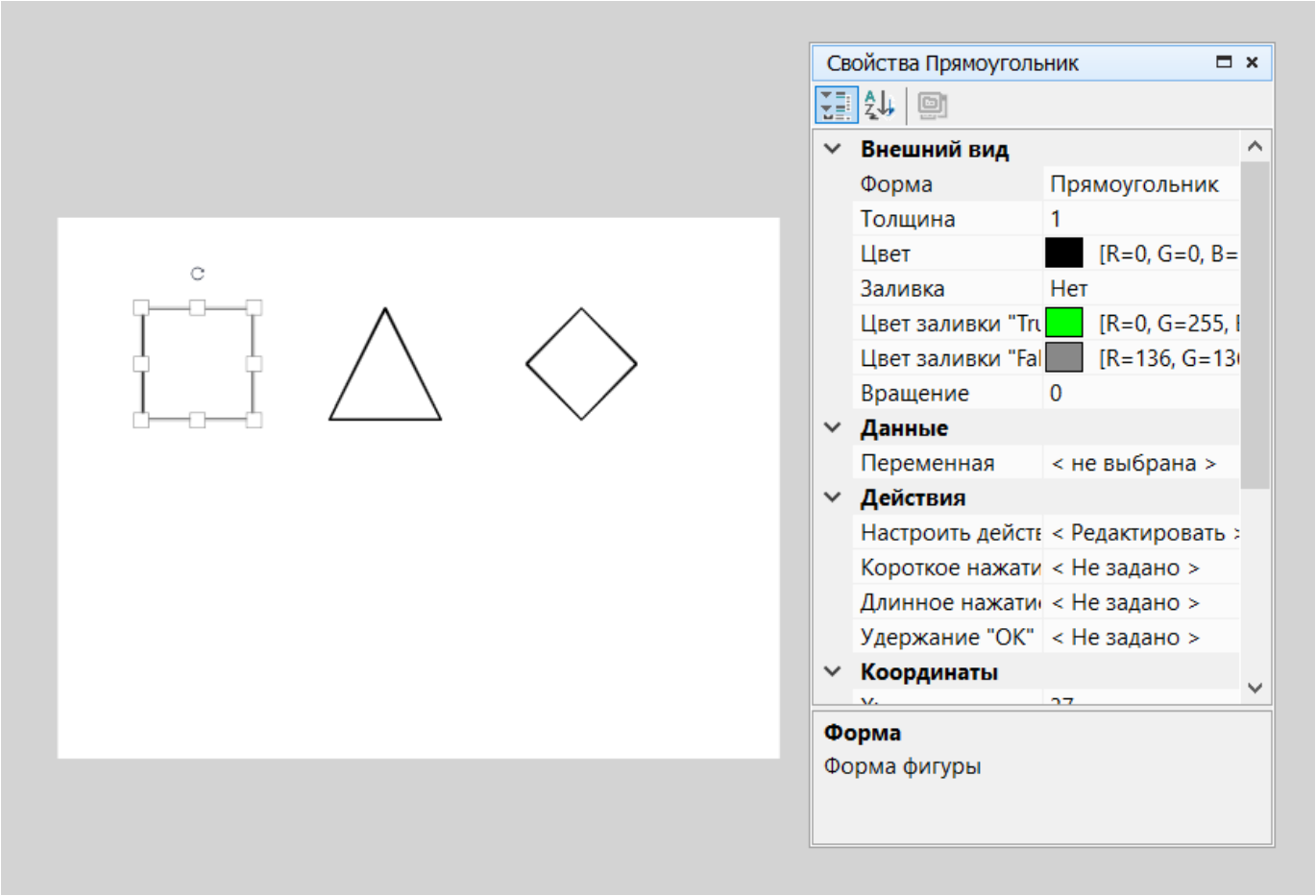


Рисунок 7.151

Координаты

Расположение многоугольника определяется координатами его левого верхнего угла по осям X и Y. Координаты задаются на панели свойств или перемещением объекта по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Форма**	Форма отображаемого многоугольника: **треугольник**, **прямоугольник**, **ромб**

Группа	Параметр	Описание
	Толщина	Толщина контура многоугольника
	Цвет	Цвет контура многоугольника
	Заливка	Наличие заливки фигуры
	Цвет заливки «True»	Цвет заливки фигуры, если значение привязанной переменной «True»
	Цвет заливки «False»	Цвет заливки фигуры, если значение привязанной переменой «False»
	Вращение	Вращение элемента вокруг центра в градусах
Данные	**Переменная**	К параметру привязывается переменная
Действия	**Настроить действия**	Выбор действия для перехода между экранами
	Короткое нажатие “ОК”	Список действий, выполняемых по короткому нажатию на кнопку “ОК”
	Длинное нажатие “ОК”	Список действий, выполняемых по длинному нажатию на кнопку “ОК”
	Удержание “ОК”	Список действий, выполняемых при удержании кнопки “ОК”
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.10 Круг

Настройка свойств

Для корректного отображения на экране прибора следует указать координаты расположения, размеры элемента и задать параметры на панели свойств.

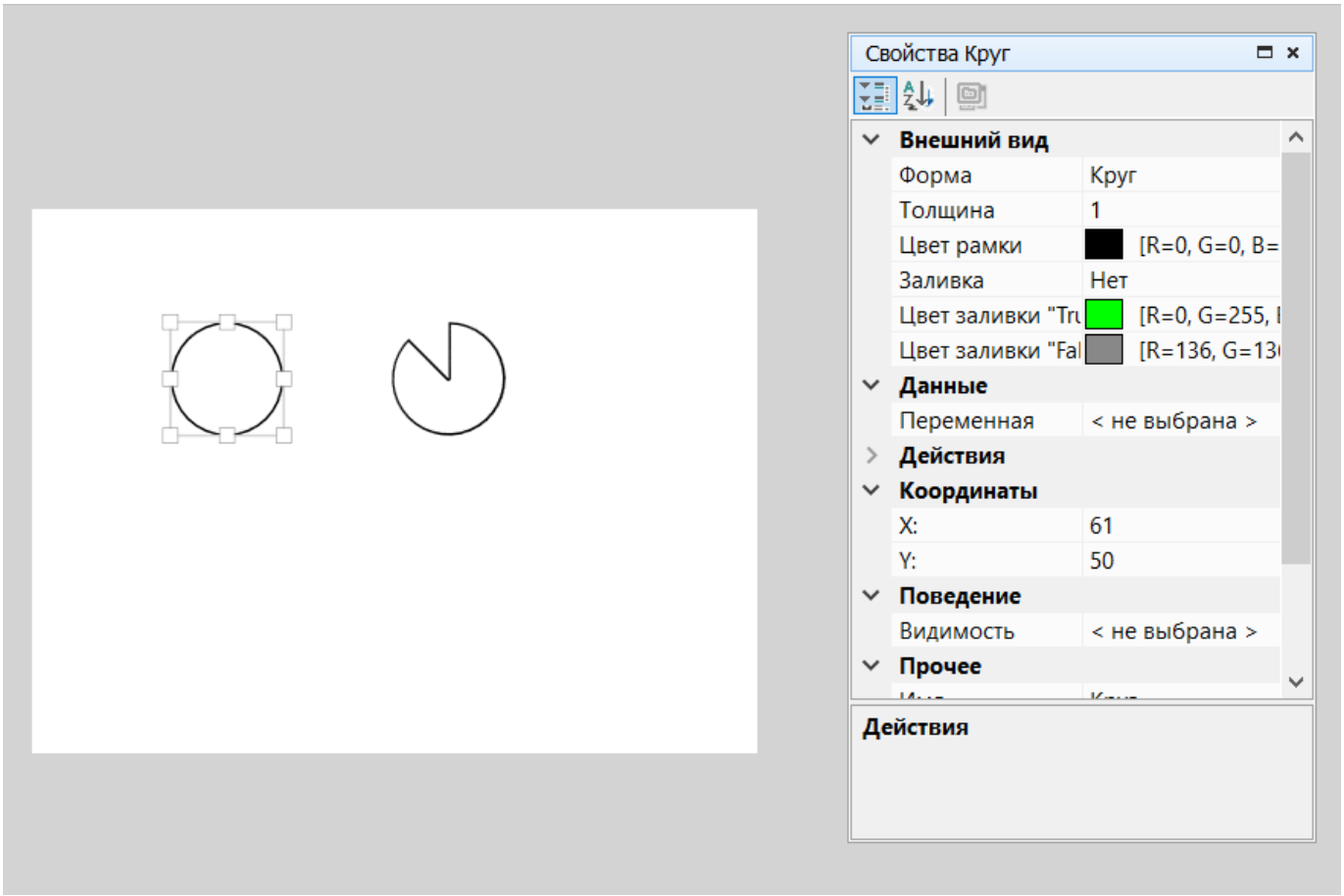


Рисунок 7.152

Координаты

Расположение круга определяется координатами верхнего левого углового маркера по осям X и Y. Координаты задаются на панели свойств или перемещением объекта по полю экрана. Отсчет координат по обеим осям начинается с 0:

- по оси X — слева направо, конечное значение определяется размерами элемента и шириной экрана;
- по оси Y — сверху вниз, конечное значение определяется размерами элемента и высотой экрана.

Размеры

Размеры элемента определяются по осям X и Y и задаются на панели свойств или изменением границ элемента на поле экрана.

Параметры

Группа	Параметр	Описание
Внешний вид	**Форма**	Форма отображаемого круга: **круг**, **сектор круга**
	Толщина	Толщина контура фигуры
	Цвет рамки	Цвет контура фигуры
	Заливка	Наличие заливки фигуры
	Цвет заливки «True»	Цвет заливки фигуры, если значение привязанной переменной «True»
	Цвет заливки «False»	Цвет заливки фигуры, если значение привязанной переменой «False»
Данные	**Переменная**	К параметру привязывается переменная
Действия	**Настроить действия**	Выбор действия для перехода между экранами
	Короткое нажатие “OK”	Список действий, выполняемых по короткому нажатию на кнопку “OK”

Группа	Параметр	Описание
	Длинное нажатие “ОК”	Список действий, выполняемых по длинному нажатию на кнопку “ОК”
	Удержание “ОК”	Список действий, выполняемых при удержании кнопки “ОК”
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации

7.7.11 Кнопка

Элемент **Кнопка** предназначен для переключения состояния переменной типа Bool.

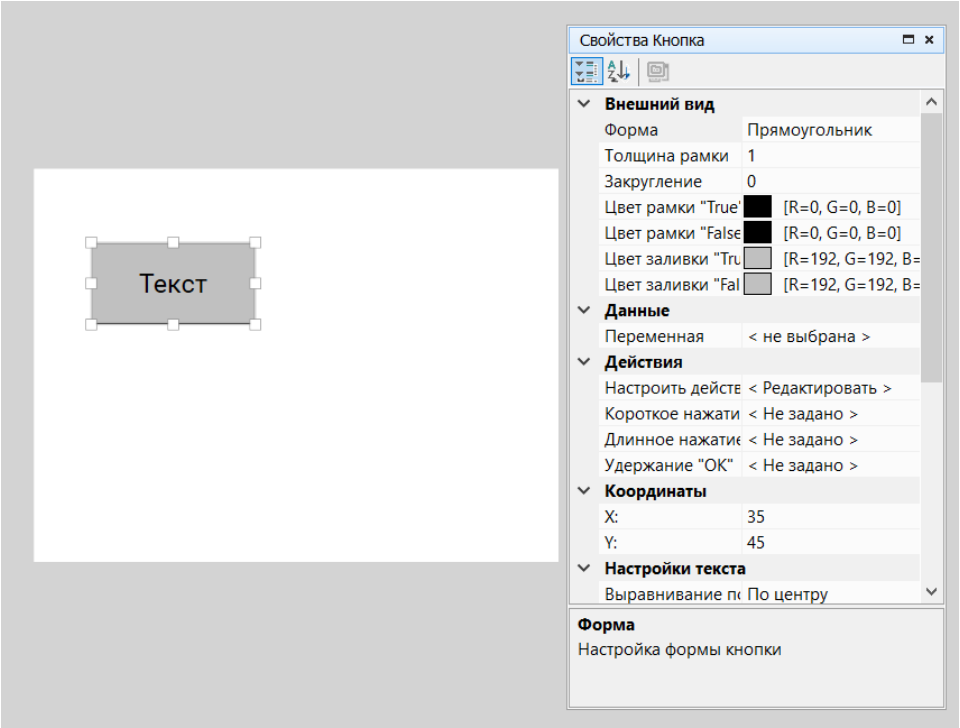


Рисунок 7.153

Настройка свойств

Для настройки необходимого вида элемента **Кнопка** на экране прибора доступны следующие свойства:

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Форма**	Форма отображаемой кнопки: **прямоугольник** или **круг**	
	Толщина рамки	Толщина контура кнопки	
	Закругление	**ПРИМЕЧАНИЕ** Доступно только при выборе формы «Прямоугольник» Скругление углов рамки	
	Цвет рамки «True»	Цвет контура фигуры, если кнопка нажата	
	Цвет рамки «False»	Цвет контура фигуры, если кнопка отжата	
	Цвет заливки «True»	Цвет заливки фигуры, если кнопка нажата	
	Цвет заливки «False»	Цвет заливки фигуры, если кнопка отжата	

Группа	Параметр	Описание	Дополнительные возможности
Данные	**Переменная**	К параметру привязывается переменная типа Bool	
Действия	**Настроить действия**	Выбор действия для перехода между экранами	
	Короткое нажатие “ОК”	Список действий, выполняемых по короткому нажатию на кнопку “ОК”	
	Длинное нажатие “ОК”	Список действий, выполняемых по длинному нажатию на кнопку “ОК”	
	Удержание “ОК”	Список действий, выполняемых при удержании кнопки “ОК”	
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Настройки текста	**Выравнивание по горизонтали**	Выравнивание отображаемого в элементе текста: по левому краю, по центру, по правому краю	
	Выравнивание по вертикали	Выравнивание отображаемого в элементе текста: по верхнему краю, по центру, по нижнему краю	
	Размер шрифта	Размер отображаемого в элементе текста: 16, 32, 48	
	Текст «True»	Отображаемый текст, если кнопка нажата	
	Текст «False»	Отображаемый текст, если кнопка отжата	
	Цвет текста «True»	Цвет отображаемого текста, если кнопка нажата	
	Цвет текста «False»	Цвет отображаемого текста, если кнопка отжата	
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана
	Высота	Высота элемента по оси Y	

7.7.12 Переключатель

Элемент **Переключатель** предназначен для управления состоянием переменной типа Bool, а также для отображения состояния этой переменной.

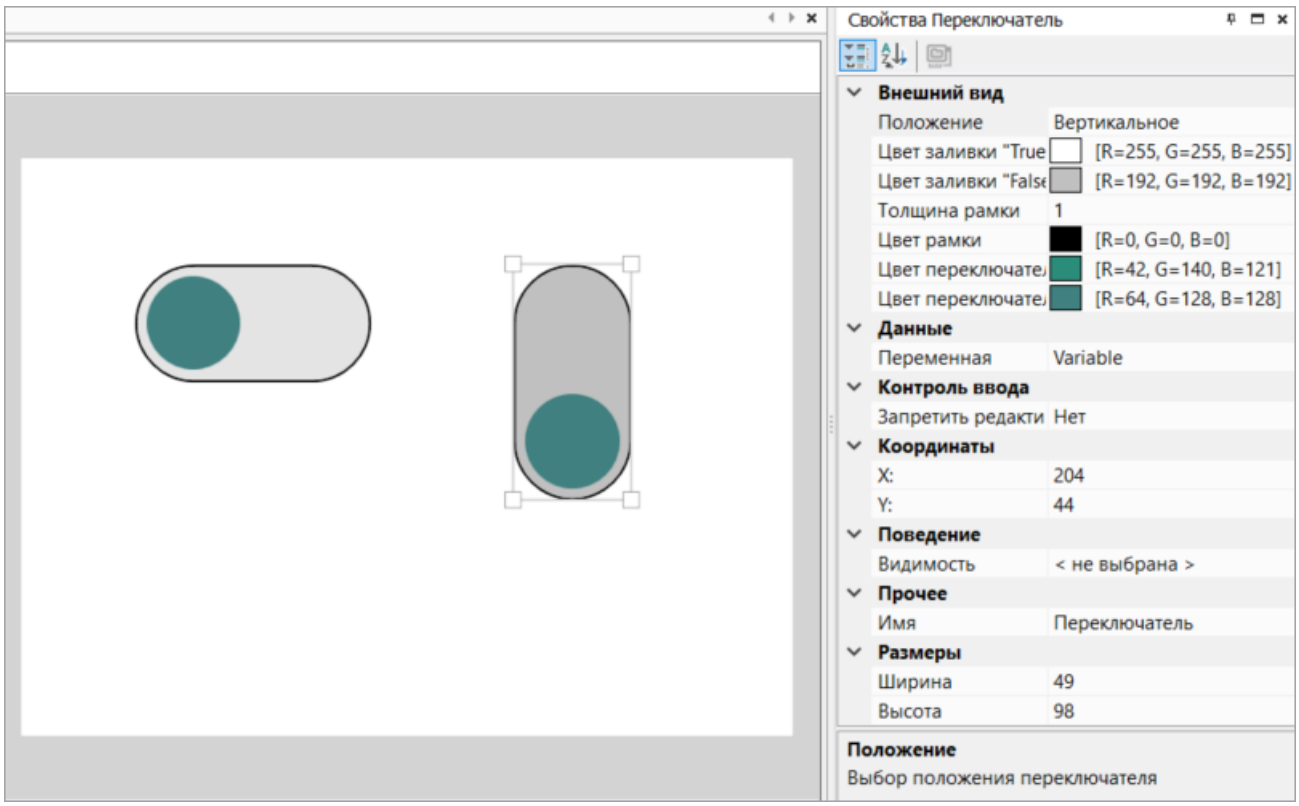


Рисунок 7.154 – Внешний вид и панель Свойства элемента Переключатель

Настройка свойств

Для настройки необходимого вида элемента **Переключатель** на экране прибора доступны следующие свойства:

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Положение**	Вертикальное или горизонтальное расположение переключателя	
	Цвет заливки «True»	Цвет заливки элемента, если переключатель включен	
	Цвет заливки «False»	Цвет заливки элемента, если переключатель выключен	
	Толщина рамки	Толщина контура элемента	
	Цвет рамки	Цвет контура элемента	
	Цвет переключателя «True»	Цвет переключателя в состоянии "включен"	
	Цвет переключателя «False»	Цвет переключателя в состоянии "выключен"	
Данные	**Переменная**	К параметру привязывается переменная типа Bool	
Контроль ввода	**Запретить редактирование**	Установка/снятие запрета на изменение значения переключателя с экрана прибора	
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана.

Группа	Параметр	Описание	Дополнительные возможности
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана. Пропорции переключателя сохраняются с соблюдением следующих правил: – горизонтальное положение: Длина = Ширина x 2 – вертикальное положение: Длина = Ширина / 2
	Высота	Высота элемента по оси Y	

7.7.13 Группа переключателей

Элемент **Группа переключателей** предназначен для управления списками параметров и отображения состояния этих списков:

- **радиокнопка** — допускает выбор только одного параметра из списка;
- **чекбокс** — допускает выбор одного или нескольких параметров из списка.

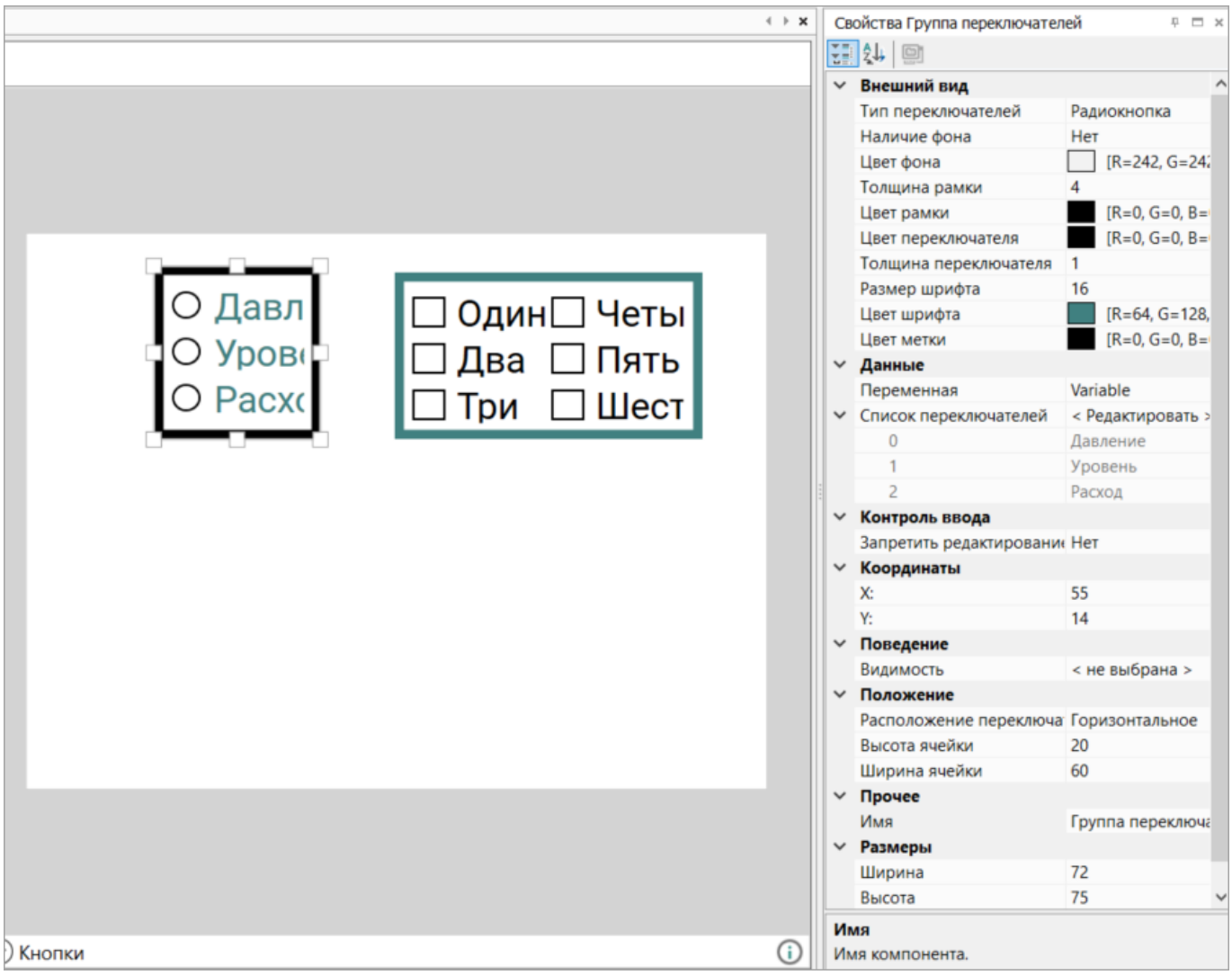
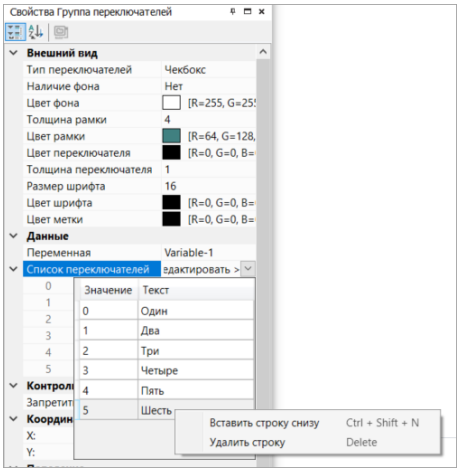
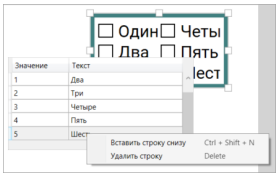


Рисунок 7.155 – Внешний вид и панель Свойства элемента Группа переключателей

Настройка свойств

Для настройки необходимого вида элемента **Группа переключателей** на экране прибора доступны следующие свойства:

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Тип переключателей**	Выбор типа переключателя: радиокнопка или чекбокс	
	Наличие фона	Наличие/отсутствие фона отображаемого в элементе текста	
	Цвет фона	Цвет фона отображаемого в элементе текста	
	Толщина рамки	Толщина рамки элемента	
	Цвет рамки	Цвет рамки элемента	
	Цвет переключателя	Цвет рамки поля для установки метки	
	Толщина переключателя	Толщина рамки поля для установки метки	
	Размер шрифта	Размер отображаемого в элементе текста: 16, 32, 48	
	Цвет шрифта	Цвет отображаемого в элементе текста	

Группа	Параметр	Описание	Дополнительные возможности
	Цвет метки	Цвет устанавливаемой метки	
Данные	**Переменная**	К параметру привязывается целочисленная переменная	
	Список переключателей (максимальное количество строк - 32)	Значения переменной и соответствующий им текст, который будет отображаться в элементе. Для добавления строк нажмите "Редактировать" и воспользуйтесь контекстным меню открывшейся таблицы:  **Рисунок 7.156** Укажите значение переменной, а также текст для отображения в элементе	Добавление строк в список параметров также доступно на поле экрана: 1. Дважды кликните ЛКМ по элементу, размещенному на экране 2. Для добавления строк воспользуйтесь контекстным меню открывшейся таблицы:  **Рисунок 7.157** 3. Укажите значение переменной, а также текст для отображения в элементе
Контроль ввода	**Запретить редактирование**	Установка/снятие запрета на действие с экрана прибора: • переключение между параметрами списка (для элемента радиокнопка); • включение/отключение параметров списка (для элемента чекбокс)	
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	

Группа	Параметр	Описание	Дополнительные возможности
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
Положение	**Расположение переключателей**	Выбор расположения пунктов списка: горизонтальный или вертикальный	
	Высота ячейки	Высота ячейки для расположения пунктов списка	
	Ширина ячейки	Ширина ячейки для расположения пунктов списка	
	Количество активных чекбоксов	**ПРИМЕЧАНИЕ** Доступно только при выборе типа переключателя "Чекбокс" Ограничение одновременного выбора нескольких параметров из списка. Ограничение не может быть больше количества добавленных строк в Список переключателей	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана
	Высота	Высота элемента по оси Y	

7.7.14 Изображение

Добавление пользовательского изображения в проект

Для добавления изображения:

1. Разместите в рабочей области объект **Изображение**.

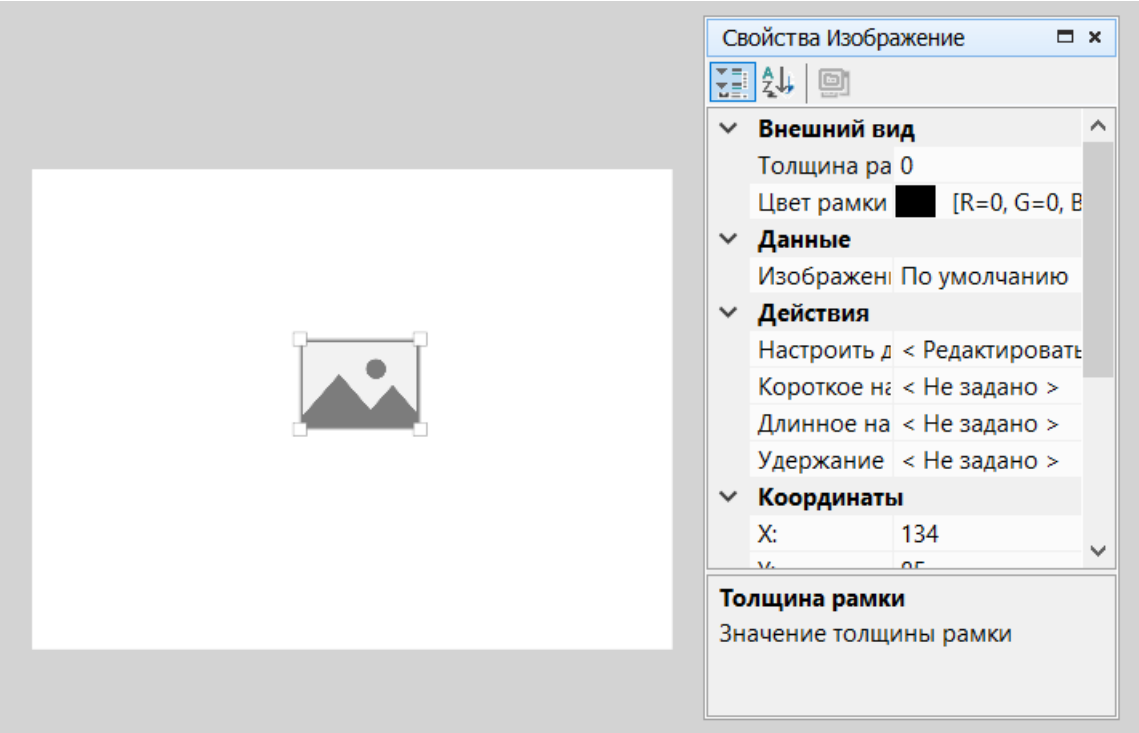


Рисунок 7.158

2. Дважды кликите ЛКМ по размещенной фигуре. Откроется окно выбора изображения.

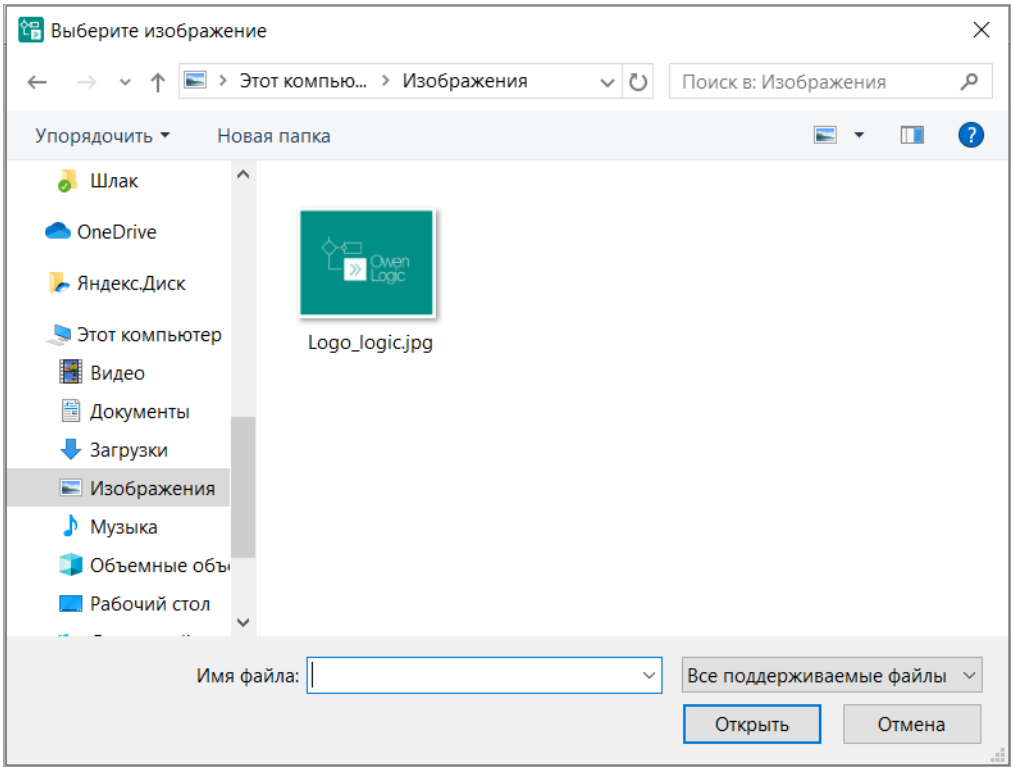


Рисунок 7.159

ПРИМЕЧАНИЕ
Для выбора доступны следующие типы файлов: (*.jpg; *.jpeg; *.jpe, *.bmp).
Максимальное разрешение изображения ограничено разрешением экрана устройства.
Допускается уменьшение изображения в редакторе экрана Owen Logic.

3. Выберите изображение и нажмите **Открыть**.

Настройка свойств

Для настройки необходимого вида элемента **Изображение** на экране прибора доступны следующие свойства:

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Толщина рамки**	Толщина рамки изображения	
	Цвет рамки	Цвет рамки изображения	
Данные	**Изображение**	Выбор файла изображения	
Действия	**Настроить действия**	Выбор действия для перехода между экранами	
	Короткое нажатие “ОК”	Список действий, выполняемых по короткому нажатию на кнопку “ОК”	
	Длинное нажатие “ОК”	Список действий, выполняемых по длинному нажатию на кнопку “ОК”	
	Удержание “ОК”	Список действий, выполняемых при удержании кнопки “ОК”	
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и шириной экрана	
Поведение	**Видимость**	К параметру привязывается булевская переменная, которая будет определять видимость элемента	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X. Максимальное значение не более ширины загруженного изображения.	Размеры элемента можно задать изменением границ объекта на поле экрана. Ширина и высота элемента изменяются пропорционально размерам загруженного изображения.
	Высота	Высота элемента по оси Y. Максимальное значение не более высоты загруженного изображения.	

ПРИМЕЧАНИЕ
В случае копирования и привязки одного и того же изображения к нескольким элементам визуализации, не занимается дополнительный объем пользовательского ПЗУ визуализации (индикатор **Польз. ПЗУ. визуал.** в строке состояния).

7.7.15 Меню

Элемент **Меню** предназначен для перехода между экранами визуализации. К каждому пункту меню привязывается экран визуализации.

Для добавления компонента **Меню** на экран визуализации:

1. Разместите в рабочей области элемент **Меню**.

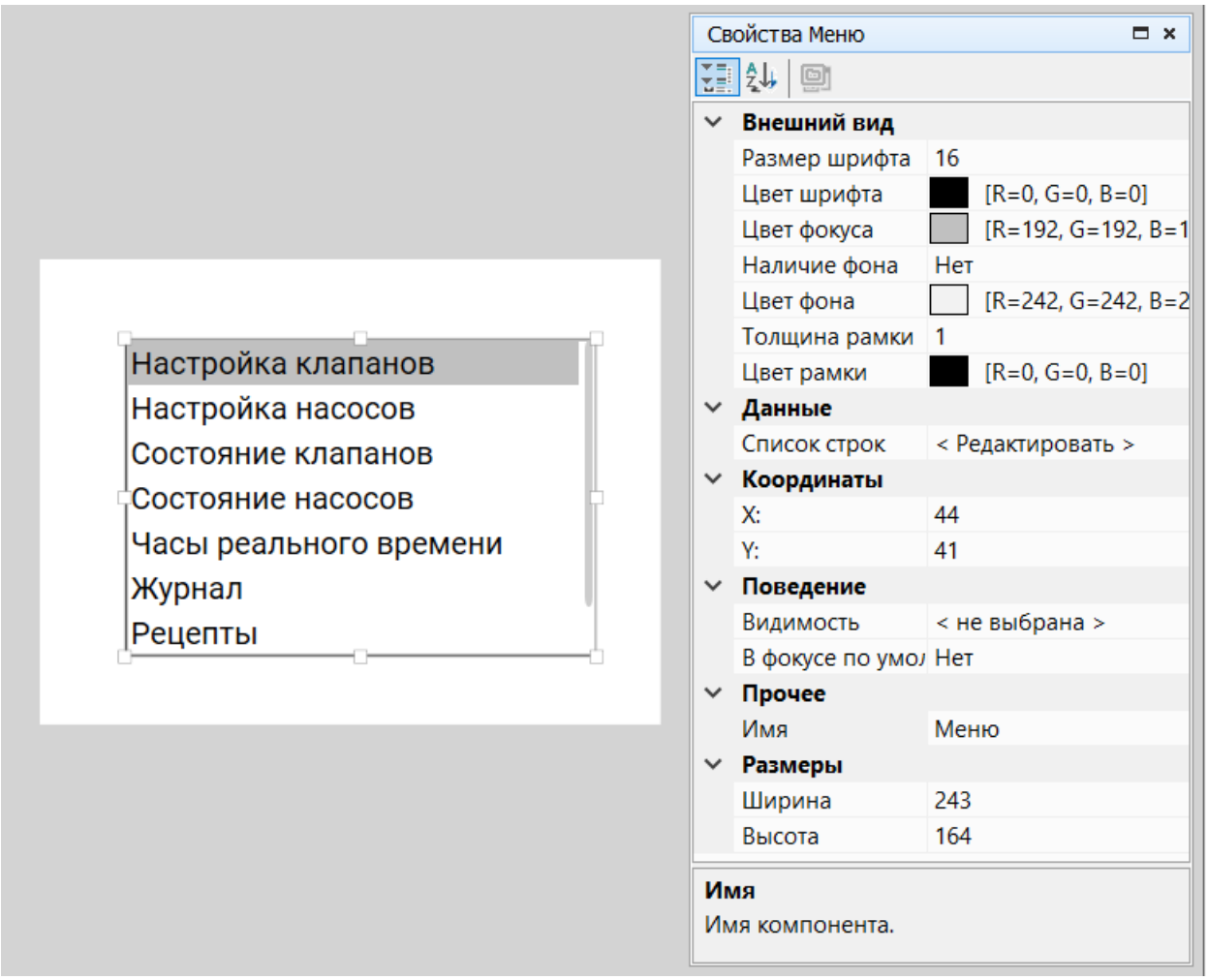


Рисунок 7.160

2. Дважды кликните ЛКМ по размещенному элементу. Откроется окно Список строк Меню, в котором доступно редактирование строк.

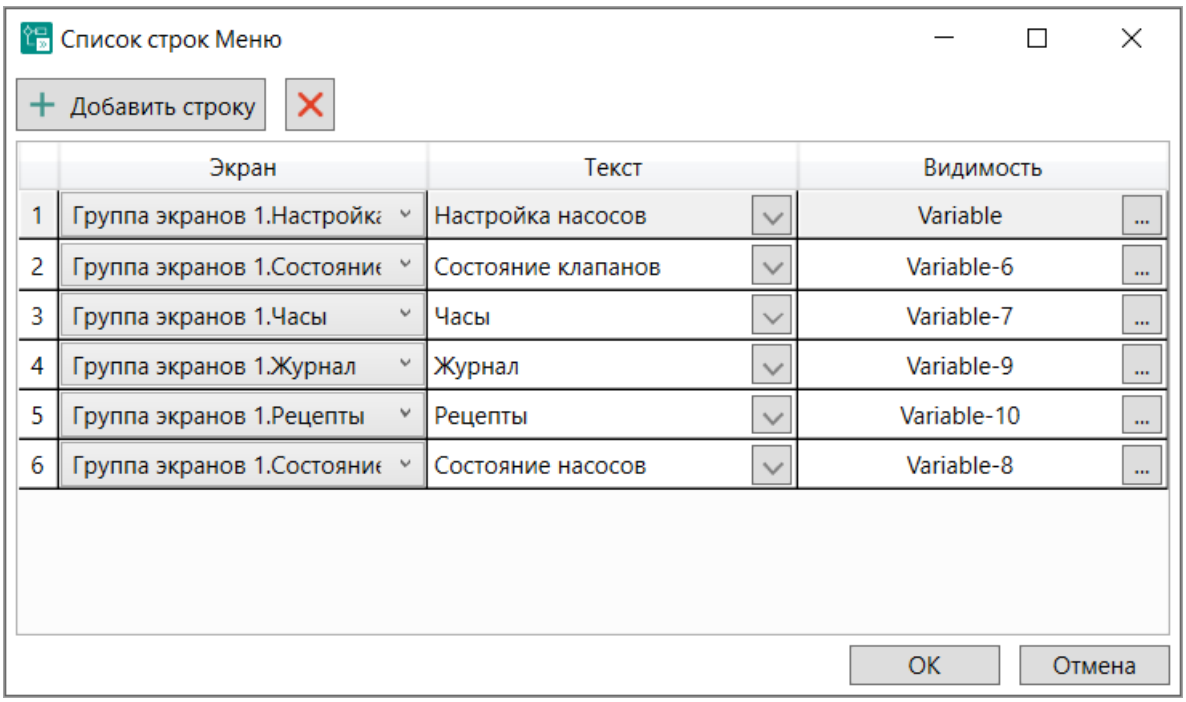


Рисунок 7.161

3. Добавьте необходимое количество строк.

   ПРИМЕЧАНИЕ
   Максимальное количество строк — 128.

4. В столбце «Экран» укажите экран, на который будет выполняться переход в случае выбора данного пункта меню.

   ПРИМЕЧАНИЕ
   Для выбора доступны все экраны проекта, кроме текущего.

5. В столбце «Текст» автоматически отображается название выбранного экрана. Для изменения текста нажмите на маркер , расположенный в конце строки, и в открывшемся поле введите текст для отображения в пункте меню.
6. Настройте видимость добавленных строк, привязав переменные в столбце «Видимость».
7. Нажмите кнопку **OK**.

Список свойств

Для настройки элемента **Меню** на экране прибора доступны следующие свойства:

Группа	Параметр	Описание	Дополнительные возможности
Внешний вид	**Размер шрифта**	Размер отображаемого в элементе текста: 16, 32, 48	
	Цвет шрифта	Цвет отображаемого в элементе текста	
	Цвет фокуса	Цвет строки в фокусе	
	Наличие фона	Фон отображаемого в элементе текста	
	Цвет фона	Цвет фона отображаемого в элементе текста	
	Толщина рамки	Толщина рамки элемента	
	Цвет рамки	Цвет рамки элемента	
Данные	**Список строк**	Открывает окно редактирования строк	

Группа	Параметр	Описание	Дополнительные возможности
Координаты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
	В фокусе по умолчанию	При выборе значения **Да** после перехода на экран элемент **Меню** будет находиться в фокусе. **ПРИМЕЧАНИЕ** На одном экране допускается только один элемент со значением **В фокусе по умолчанию** - **Да**	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана
	Высота	Высота элемента по оси Y	

7.7.16 График

Элемент **График реального времени** предназначен для отображения изменения значений переменных в графическом виде на экране прибора:

- позволяет визуализировать до 4 переменных типа udint или float одновременно;
- доступна настройка интервала времени опроса для построения графика (максимально допустимое значение 23ч59м59с);
- период опроса рассчитывается автоматически, зависит от максимального значения оси X, ширины элемента График и толщины рамки, и не может быть менее 1 секунды;
- количество срезов определяется по формуле: *Количество срезов = max X / t опроса*, где:
  - max X — максимальное значение оси X;
  - t опроса — период опроса.

ПРИМЕЧАНИЕ
Функционал построения графиков доступен только для приборов с графическим цветным ЖКИ. Работа **Графика** в режиме симуляции не поддерживается.

Чтобы добавить график на экран выберите в окне Библиотека компонентов элемент **График** и разметите в рабочей области.

Появятся оси координат с отображением шкалы и сетки. Ось X – ось времени, ось Y - ось значений переменных (udint или float).

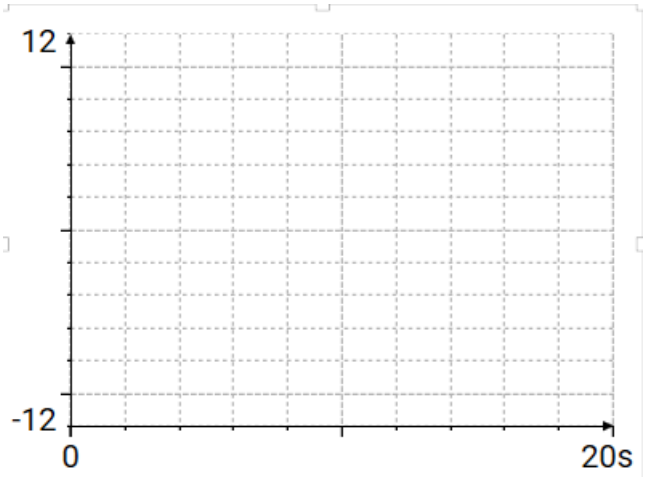


Рисунок 7.162 – Оси координат со значениями по умолчанию

Для настройки параметров и добавления переменных используйте окно **Свойства График**:

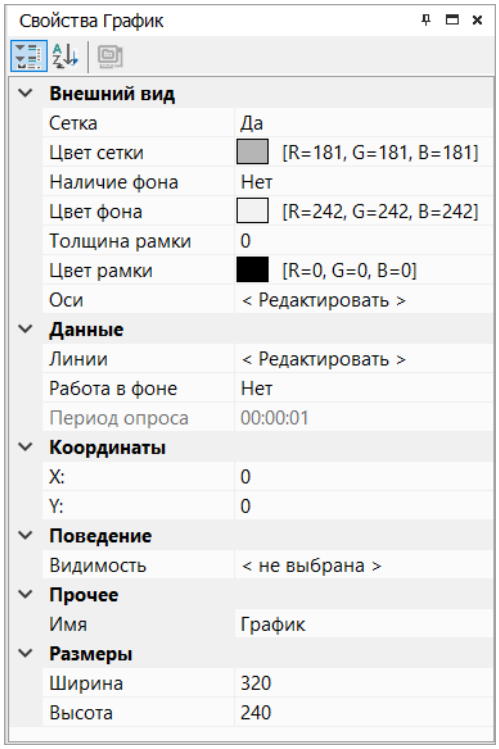


Рисунок 7.163 – Окно Свойства График

Настройте **Внешний вид** графика:

- наличие и цвет сетки;
- наличие и цвет фона;
- наличие, толщина и цвет рамки;
- оси:

  Выберите строчку Оси и нажмите ЛКМ на <Редактировать> и затем на появившуюся кнопку ... . Откроется окно Оси График:

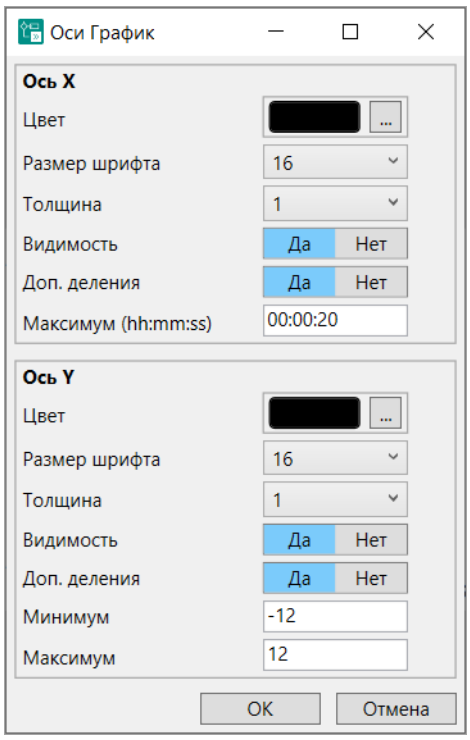


Рисунок 7.164 – Окно Оси График

Если необходимо, настройте цвет и толщину осей, размер шрифта, видимость осей и дополнительных делений. Введите необходимое максимальное значение оси X в формате hh:mm:ss. Максимально допустимое значение 23:59:59. Значение по умолчанию 20 секунд.

ПРИМЕЧАНИЕ
Минимальное значение оси X = 0 – не редактируемый параметр.

Введите минимальное и максимальное значение оси Y. Значение по умолчанию: -12, 12.

В случае ввода некорректного значения возникнет сообщение об ошибке, кнопка OK станет неактивна:

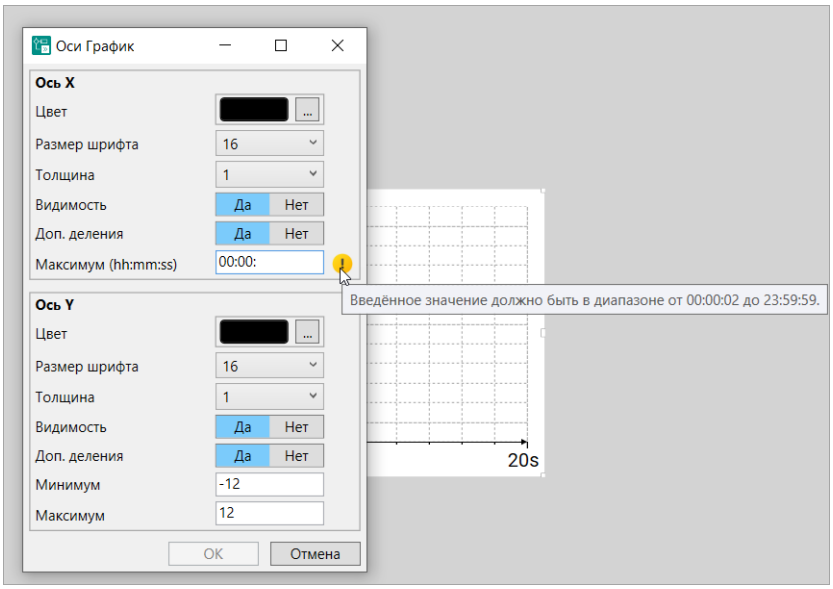


Рисунок 7.165 – Ошибка ввода данных

Для добавления переменных на график в разделе **Данные** окна Свойства График, выберите строчку Линии и нажмите ЛКМ на <Редактировать> и затем на появившуюся кнопку ..., или дважды нажмите ЛКМ на поле графика. Откроется окно Список линий График:

Рисунок 7.166 – Окно Список линий График

В открывшемся окне выберите:

- тип переменной – целочисленное / с плавающей запятой;
- переменную;
- цвет и толщину линии;
- видимость - возможен выбор переменной булевского типа из категории Стандартные таблицы переменных.

ПРИМЕЧАНИЕ
Максимальное количество переменных на графике - 4.

Для удобства понимания количества добавленных переменных, а также настроек внешнего вида линий на графике, линии отображаются со смещением по оси Y относительно друг друга.

Параметр Работа в фоне (Да/Нет) раздела **Данные** влияет на сохранение данных при переходе на другой экран.

Период опроса – не редактируемый параметр, зависит от максимального значения оси X, ширины элемента График и толщины рамки.

ПРИМЕЧАНИЕ
Период опроса не может быть менее 1 секунды.

Расположение графика определяется координатами левого верхнего угла по осям X и Y. Координаты задаются перемещением объекта по полю экрана. В случае необходимости точного положения графика на экране прибора настройте координаты окна Свойства График в разделе **Координаты**.

При необходимости в разделе **Поведение** настройте видимость всего графика в зависимости от переменной булевского типа.

Изменение имени графика возможно в разделе **Прочее** окна Свойства График.

Для масштабирования графика наведите мышь на угловой или боковой маркер, удерживая ЛКМ задайте необходимый размер графика на экране, также изменение размера возможно в разделе **Размеры**: ширина и высота задаются в пикселях.

После записи программы на экране прибора отобразится график с добавленными переменными:

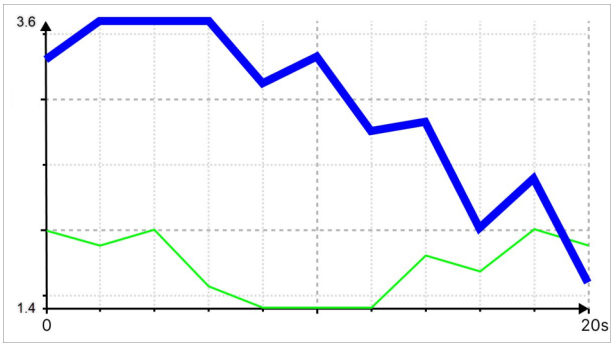


Рисунок 7.167 – Пример отображения графика на экране прибора

ПРИМЕЧАНИЕ
Если при отображении графика на экране прибора значение параметра выходит за границы минимального или максимального значения оси Y, то линия на графике продолжает движение по соответствующей границе.

В случае удаления из проекта переменной, добавленной на график, появится информационное окно:

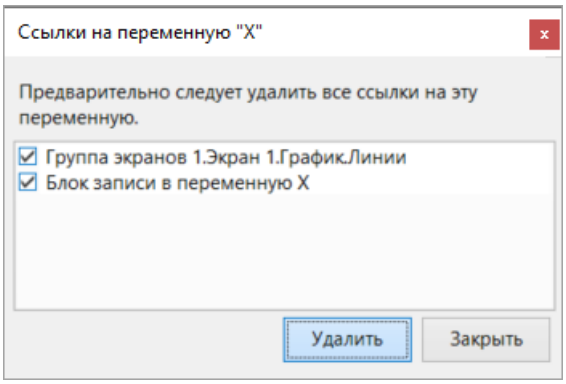


Рисунок 7.168 – Информационное окно

В случае выбора **Удалить** привязка переменной удаляется.

7.7.17 Стрелочный индикатор

Элемент **Круговой индикатор со стрелочным указателем** предназначен для отображения текущего значения параметра на дуговой шкале. Положение стрелки позволяет визуально оценивать отклонение значения параметра от заданного диапазона в ту или иную сторону. Если значение параметра меньше значения начала шкалы — стрелка фиксируется на минимальном значении, если больше значения конца шкалы — на максимальном.

Диапазоны шкалы кругового индикатора:

- **Аварийно низкий** — от начала шкалы до границы аварийно низкого диапазона.
- **Предупреждающе низкий** — от границы аварийно низкого до границы предупреждающе низкого диапазона.
- **Базовый** — между низкими и высокими предупреждающими границами.
- **Предупреждающе высокий** — от границы предупреждающе высокого до границы аварийно высокого диапазона.
- **Аварийно высокий** — от границы аварийно высокого диапазона до конца шкалы.

Добавление элемента на экран

Выберите из Библиотеки компонентов элемент **Стрелочный индикатор** и разместите в рабочей области.

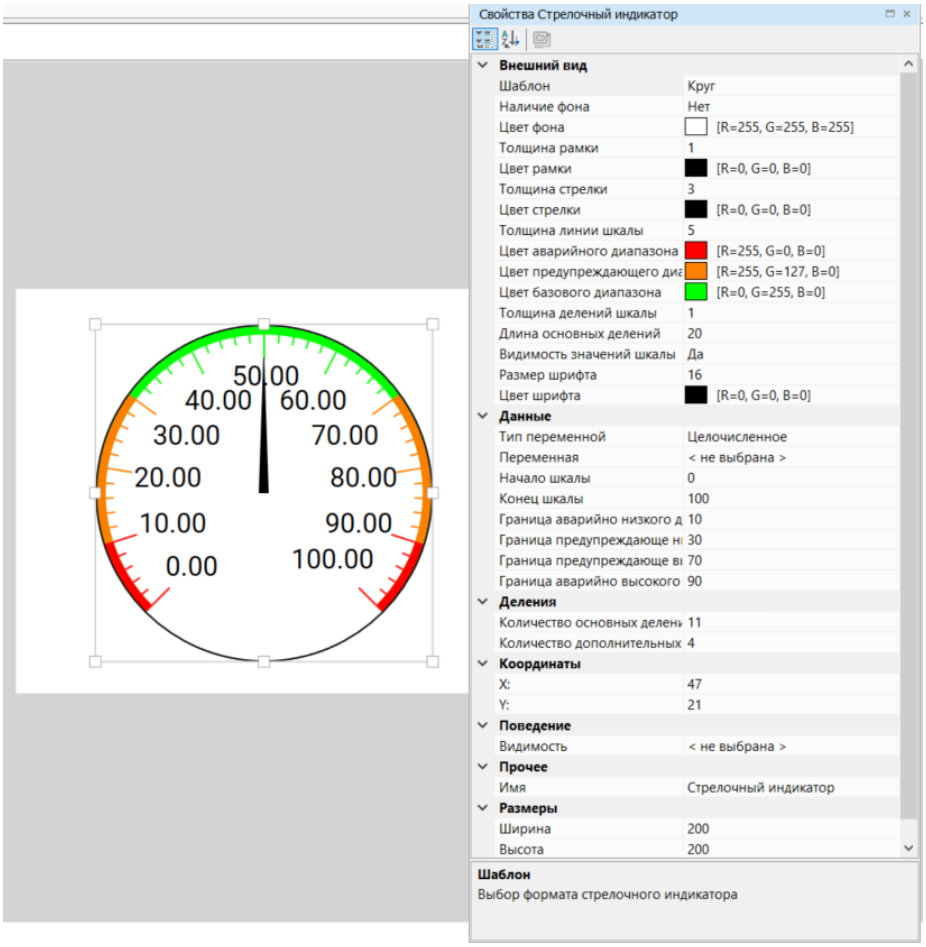


Рисунок 7.169 – Элемент визуализации Стрелочный индикатор

Настройка свойств

Для настройки отображения элемента **Стрелочный индикатор** на экране прибора доступны следующие свойства:

Группа	Параметр		Описание	Дополнительные возможности
Внешний вид	**Шаблон**		Выбор формата элемента: • круг; • сектор круга	
	При выборе шаблона Сектор круга	**Начальный угол**	Настройка начального угла элемента	Начальный и конечный углы можно изменить перемещением маркеров углов на поле экрана
		Конечный угол	Настройка конечного угла элемента	
	Наличие фона		Настройка отображения фона элемента	
	Цвет фона		Цвет заливки элемента	
	Толщина рамки		Толщина контура элемента	
	Цвет рамки		Цвет контура элемента	
	Толщина стрелки		Настройка толщины стрелочного индикатора элемента	
	Цвет стрелки		Цвет стрелочного индикатора элемента	
	Толщина линии шкалы		Настройка толщины дуги шкалы	
	Цвет аварийного диапазона		Цвет аварийно высокого и низкого диапазонов элемента	

Группа	Параметр	Описание	Дополнительные возможности
	Цвет предупреждающего диапазона	Цвет предупреждающе высокого и низкого диапазонов элемента	
	Цвет базового диапазона	Цвет нормального диапазона элемента	
	Толщина делений шкалы	Значение толщины делений шкалы элемента	
	Длина основных делений	Значение длины основных делений. Задается в процентах от радиуса окружности. Длина дополнительных делений в два раза меньше длины основных делений	
	Видимость значений шкалы	Настройка отображения значений основных делений шкалы элемента	
	Размер шрифта	Размер отображаемых в элементе значений шкалы: 16, 32, 48	
	Цвет шрифта	Цвет отображаемых в элементе значений шкалы	
Данные	**Тип переменной**	Выбор типа данных привязанной переменной: • целочисленное; • с плавающей запятой - поддерживает числа до двух знаков после запятой (без округления)	
	Переменная	К параметру привязывается переменная для отображения	
	Начало шкалы	Выбор значения начала шкалы	Доступные способы ввода: • константа - ввод числового значения; • переменная - привязка переменной
	Конец шкалы	Выбор значения конца шкалы	
	Граница аварийно низкого диапазона	Выбор значения границы аварийно низкого диапазона шкалы	
	Граница предупреждающе низкого диапазона	Выбор значения границы предупреждающе низкого диапазона шкалы	
	Граница предупреждающе высокого диапазона	Выбор значения границы предупреждающе высокого диапазона шкалы	
	Граница аварийно высокого диапазона	Выбор значения границы аварийно высокого диапазона шкалы	
Деления	**Количество основных делений**	Ввод количества основных делений на шкале	
	Количество дополнительных делений	Ввод количества дополнительных делений на шкале	
Координа-ты	**X**	Значение положения элемента по оси X. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет слева направо, конечное значение определяется размерами элемента и шириной экрана	Координаты можно задать перемещением объекта по полю экрана

Группа	Параметр	Описание	Дополнительные возможности
	Y	Значение положения элемента по оси Y. Расположение элемента определяется координатами верхнего левого угла. Отсчет координат начинается с 0 и идет сверху вниз, конечное значение определяется размерами элемента и высотой экрана	
Поведение	**Видимость**	К параметру привязывается переменная типа Bool, которая определяет видимость элемента	
Прочее	**Имя**	Отображается в списке используемых элементов в левой части редактора визуализации	
Размеры	**Ширина**	Ширина элемента по оси X	Размеры элемента можно задать изменением границ объекта на поле экрана. Размеры изменяются с сохранением пропорций 1:1
	Высота	Высота элемента по оси Y	

Особые состояния элемента Стрелочный индикатор

При определенном сочетании настроек элемент может перейти в невалидное состояние, при котором корректная работа элемента и достоверное отображение данных невозможны.

Чаще всего такие условия могут возникать при привязке переменных к параметрам Начало/Конец шкалы, Граница аварийно/предупреждающе низкого/высокого диапазона, когда значение переменной в ходе исполнения логики программы меняется.

Сочетание настроек	**Поведение элемента Стрелочный индикатор**
Начало шкалы **≥** Конец шкалы	• отображение стрелки скрыто; • шкала, деления и подписи на шкале скрыты; • элемент отображается с прозрачностью 50%
Граница аварийно низкого диапазона **<** Начало шкалы	аварийно низкий диапазон скрыт
Граница аварийно низкого диапазона **>** Конец шкалы	аварийно низкий диапазон отображается на всей шкале
Граница предупреждающе низкого диапазона **<** Начало шкалы	предупреждающе низкий диапазон скрыт
Граница предупреждающе низкого диапазона **>** Конец шкалы	предупреждающе низкий диапазон отображается на всей шкале
Граница предупреждающе высокого диапазона **<** Начало шкалы	предупреждающе высокий диапазон отображается на всей шкале
Граница предупреждающе высокого диапазона **>** Конец шкалы	отображение предупреждающе высокого диапазона скрыто
Граница аварийно высокого диапазона **<** Начало шкалы	аварийно высокий диапазон отображается на всей шкале

Сочетание настроек	Поведение элемента Стрелочный индикатор
Граница аварийно высокого диапазона **>** Конец шкалы	аварийно высокий диапазон скрыт
Пересечение (наложение) диапазонов	Отображение происходит слоями, верхние слои перекрывают нижние: • аварийно высокий (перекрывает все остальные); • аварийно низкий; • предупреждающе высокий; • предупреждающе низкий (перекрывается всеми остальными)

В случае, если значение привязанных переменных перестает соответствовать условиям, описанным в таблице выше, **Стрелочный индикатор** автоматически восстанавливает корректное отображение данных.

8 Работа с прибором

В настоящем разделе описываются функции работы и конфигурации прибора:

- Информация о приборе;
- Время цикла;
- Обновление встроенного ПО;
- Юстировка;
- Смена целевой платформы.

8.1 Информация о приборе

Для получения информации о подключенном приборе следует нажать кнопку на панели инструментов или выбрать в главном меню **Прибор** → **Информация...**.

Откроется окно с информацией о подключенном приборе:

Рисунок 8.1

Информация о проекте:

- **Выбранная модель прибора** – модель и модификация прибора, выбранные при создании проекта.

Информация о подключенном приборе:

- **Название подключенного прибора** – модель и модификация подключенного прибора;
- **Версия ПО прибора** – версия встроенного ПО подключенного прибора;
- **Среднее значение времени цикла** – средний временной промежуток выполнения операции в программе, зависит от сложности загруженной программы (см. раздел Время цикла);
- **Тип заказного элемента №** – тип выходного элемента подключенного прибора (дискретный, аналоговый);
- **Группа, номер, версия алгоритма** – значения, заданные пользователем для контроля версий загруженной программы в окне Сведения о проекте.

Информация о приборе второго поколения

Для приборов второго поколения выводимая в окне информация имеет отличия.

ПРИМЕЧАНИЕ
Список приборов на новой платформе см. в разделе О программе 1.

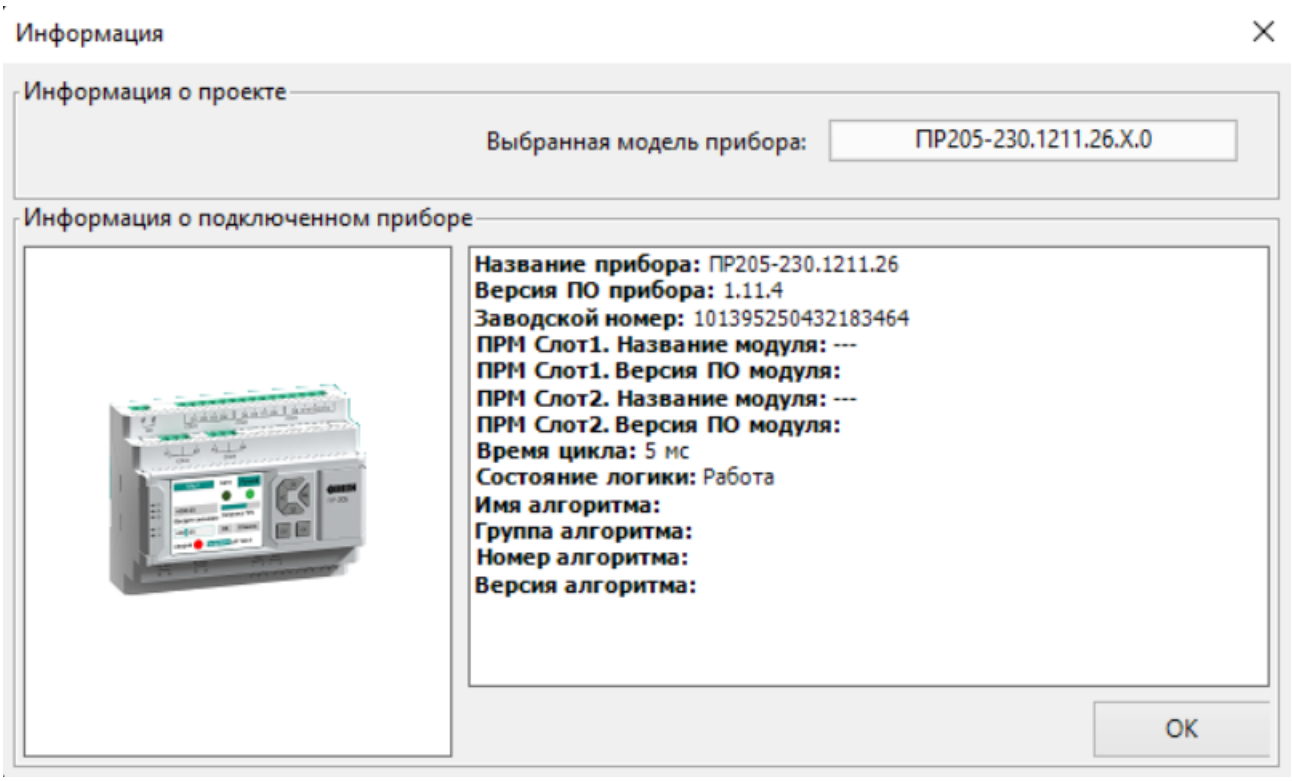


Рисунок 8.2

Информация о проекте:

- **Выбранная модель прибора** — модель и модификация прибора, выбранные при создании проекта.

Информация о подключенном приборе:

- **Название прибора** – модель и модификация подключенного прибора;
- **Версия ПО прибора** – версия встроенного ПО подключенного прибора;
- **Заводской номер** – уникальный идентификатор прибора, требуется для подключения к OwenCloud;
- **ПРМ Слот. Название модуля** – модель и модификация модуля расширения, подключенного к прибору;
- **ПРМ Слот. Версия ПО модуля** – версия встроенного ПО модуля расширения, подключенного к прибору;
- **Время цикла** — временной промежуток выполнения операции в программе, зависит от сложности загруженной программы (см. раздел Время цикла);
- **Состояние логики** — отображает текущий статус выполнения пользовательской программы;
- **Имя, группа, номер, версия алгоритма** – значения, заданные пользователем для контроля версий загруженной программы в окне Сведения о проекте.

ПРИМЕЧАНИЕ
Информация о приборах второго поколения может быть расширена разработчиками приборов.

8.2 Время цикла

Время цикла – это время выполнения рабочего цикла прибора, а именно:

- опрос состояния физических входов прибора и копирование их значений в ячейки памяти;
- обработка программы;
- чтение/запись сетевых переменных программы;
- запись результатов работы программы в физические выходы прибора.

По умолчанию время цикла равно **1 мс**. Прибор подстраивает время цикла в зависимости от сложности программы.

Условия увеличения времени цикла:

- возрастает сложность алгоритма (задействовано большое количество ФБ и макросов);
- в программе используется большое количество сетевых переменных;
- используется большое количество элементов управления данными с помощью дисплея прибора.

Пользователь не может задавать время цикла. Если прибор оснащен дисплеем, то текущее время цикла можно посмотреть в системном меню прибора. Если прибор подключен к ПК, то время цикла можно посмотреть в окне Информация о приборе.

8.3 Обновление встроенного ПО

Обновление встроенного ПО в приборе

Если для подключенного прибора выпущена новая версия встроенного ПО, то во время загрузки программы в прибор будет предложено обновить встроенное ПО:

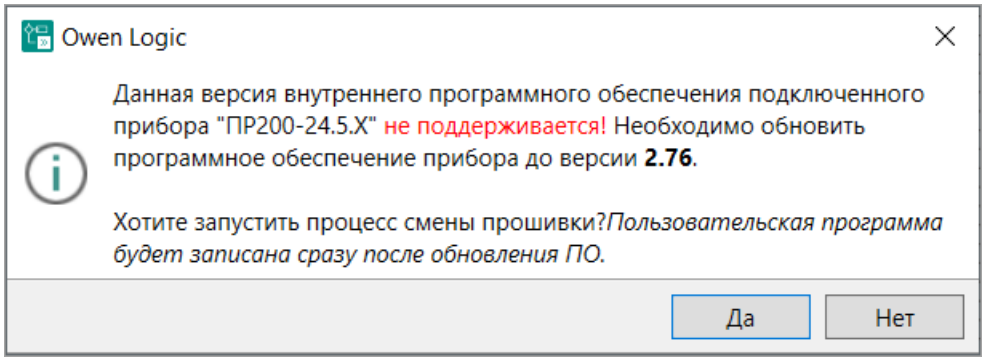


Рисунок 8.3

Для запуска обновления встроенного ПО следует нажать кнопку **Да**.

Без обновления встроенного ПО программа не будет загружена в прибор.

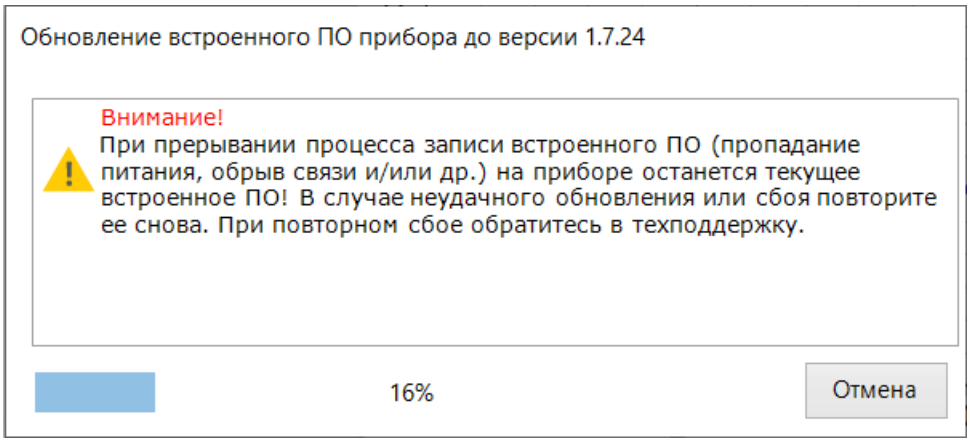


Рисунок 8.4

Разрыв связи между ПК и прибором во время обновления приведет к повреждению встроенного ПО и неработоспособности прибора. Для восстановления работоспособности прибора следует повторить операцию.

Восстановление поврежденного встроенного ПО прибора

Для восстановления поврежденного встроенного ПО следует:

1. Если прибор исправен, то перевести его в режим загрузчика (см. *руководство по эксплуатации* прибора).
2. В главном меню выбрать **Прибор** → **Обновление встроенного ПО**.

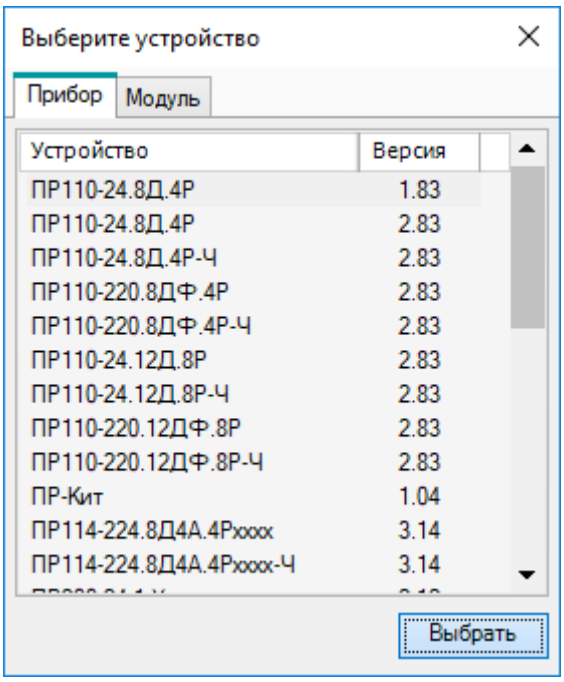


Рисунок 8.5

3. В открывшемся окне на вкладке **Прибор** выбрать модель и модификацию.
4. Нажать кнопку **Выбрать**. Начнется процесс загрузки, по окончании система выведет сообщение о результатах загрузки.

Если вышеуказанный способ не срабатывает, то необходимо ознакомиться с п. 10 FAQ или обратиться в сервисный центр компании ОВЕН.

Обновление встроенного ПО модуля расширения

Если к прибору подключен модуль расширения и для него выпущена новая версия встроенного ПО, то во время загрузки программы в прибор будет предложено обновить встроенное ПО модуля расширения.

Встроенное ПО для модуля расширения обновляется по аналогии с прибором.

Для самостоятельного обновления встроенного ПО модуля расширения следует:

1. В главном меню выбрать **Прибор** → **Обновление встроенного ПО**.
2. В открывшемся окне на вкладке **Модуль** следует выбрать модификацию и номер слота.

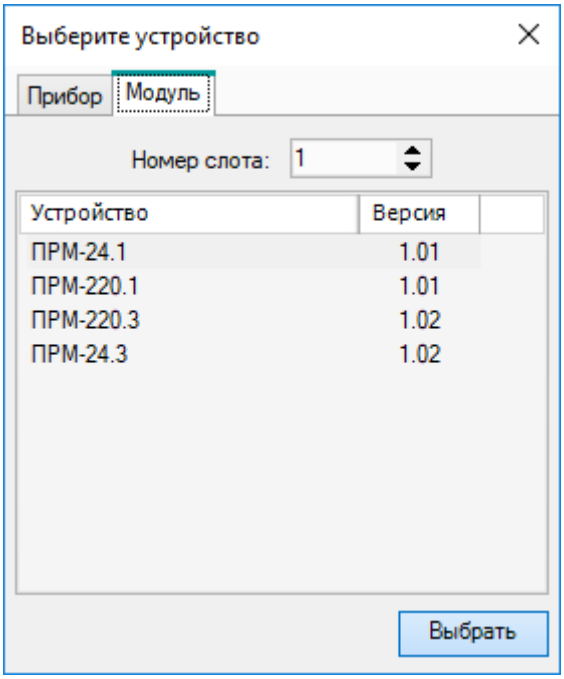


Рисунок 8.6

3. Нажать кнопку **Выбрать**. Начнется процесс загрузки, по окончании система выведет сообщение о результатах загрузки.

Во время обновления следует не прерывать подачу питания на прибор и модули расширения.

Если в прибор загружается программа без подключенного модуля расширения, возможна ситуация, когда внутреннее ПО прибора и модуля расширения будут иметь несовместимые версии. Это приведет к ситуации, что на модуле расширения будет индицирована **Авария**. Для устранения **Аварии** следует повторно обновить встроенное ПО модуля расширения.

8.4 Юстировка

ПРЕДУПРЕЖДЕНИЕ
Необходимость проведения юстировки определяется по результатам поверки прибора и должна производиться только квалифицированными специалистами метрологических служб, осуществляющих эту поверку.

ВНИМАНИЕ
Юстировку прибора следует выполнять только в соответствии с требованиями *руководства по эксплуатации* прибора. Рекомендуется не проводить юстировку самостоятельно без должных знаний и квалификации и обратиться в сервисный центр ОВЕН. Неправильная юстировка приведет к некорректной работе прибора.

Для запуска юстировки следует выбрать в главном меню **Прибор** → **Юстировка входов/выходов…**.

Перед началом юстировки появится меню выбора модели и модификации прибора.

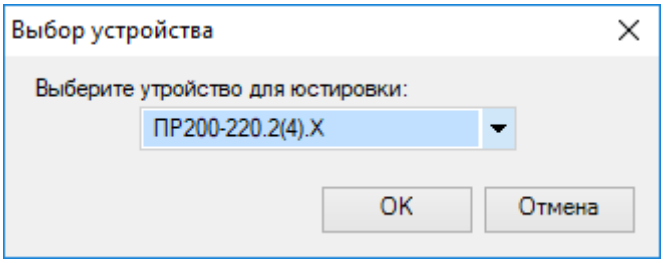


Рисунок 8.7

Далее в открывшемся диалоговом окне следует выбрать юстировку:

- аналоговых входов 8.4.1;
- аналоговых выходов 8.4.2.

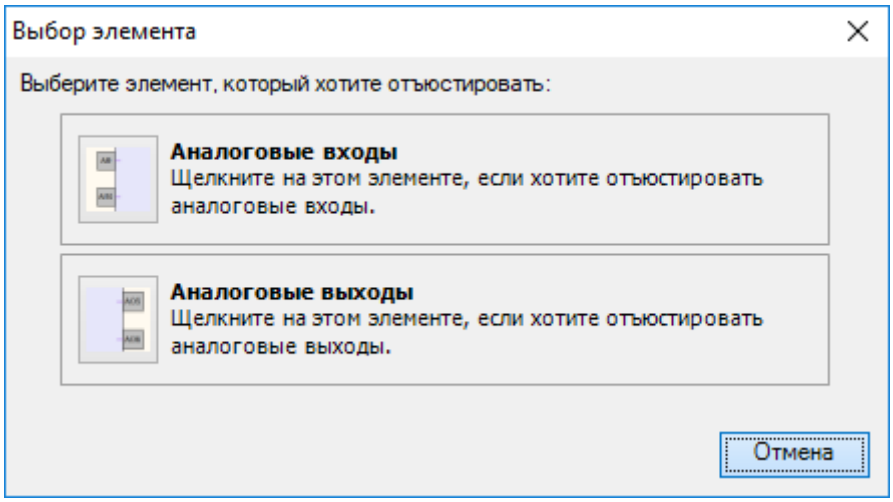


Рисунок 8.8

Если в приборе запущена программа, то во время юстировки она останавливается – на дисплее прибора отображается надпись **Программа LOGIC: Остановлена**.

8.4.1 Юстировка входов

Юстировка аналоговых входов

Для юстировки входов следует:

1. В диалоговом окне юстировки выбрать **Аналоговые входы**. Откроется окно настройки.
2. В окне настройки выбрать **Тип датчика**.

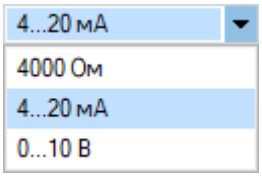


Рисунок 8.9

3. Ввести параметры подключенных к входам датчиков.

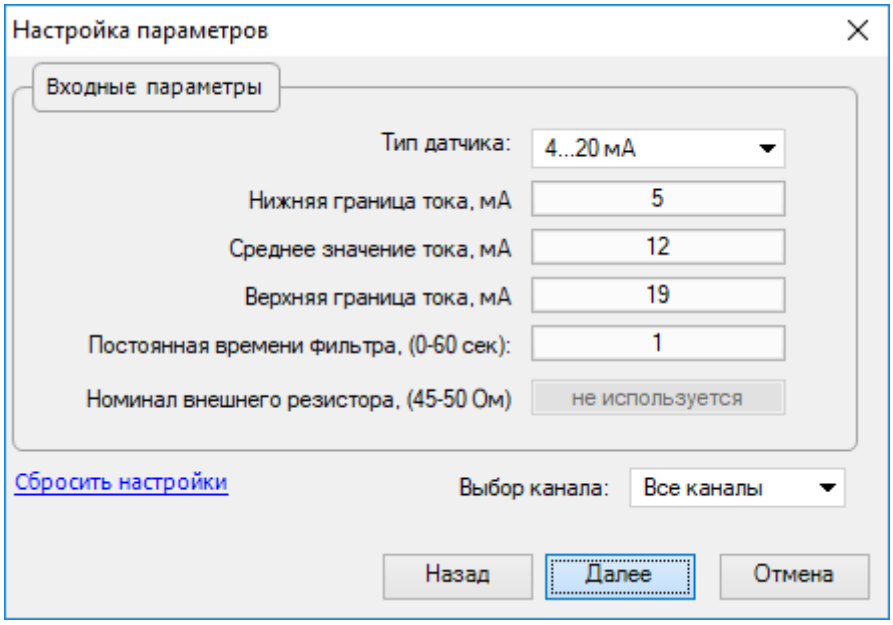


Рисунок 8.10

4. Если необходимо восстановить значения по умолчанию в полях ввода, то следует нажать **Сбросить настройки**.
5. В выпадающем меню **Выбор канала** выбрать номер входа для настройки в соответствии с введенными параметрами и нажать **Далее**.

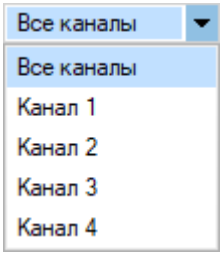


Рисунок 8.11

6. Выполнить дальнейшие указания по установке положения перемычек в приборе.

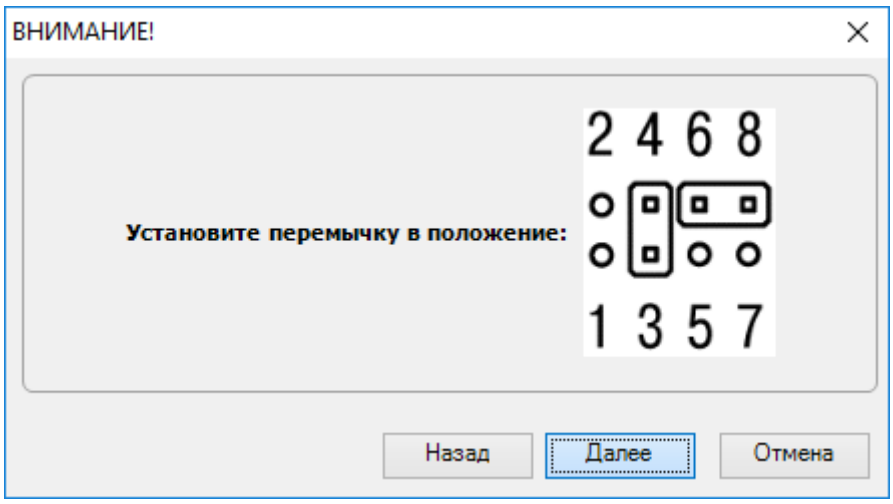


Рисунок 8.12

ПРИМЕЧАНИЕ
Установка перемычек описывается в *руководстве по эксплуатации* прибора.

7. На юстируемые входы подать сигналы с требуемыми характеристиками.

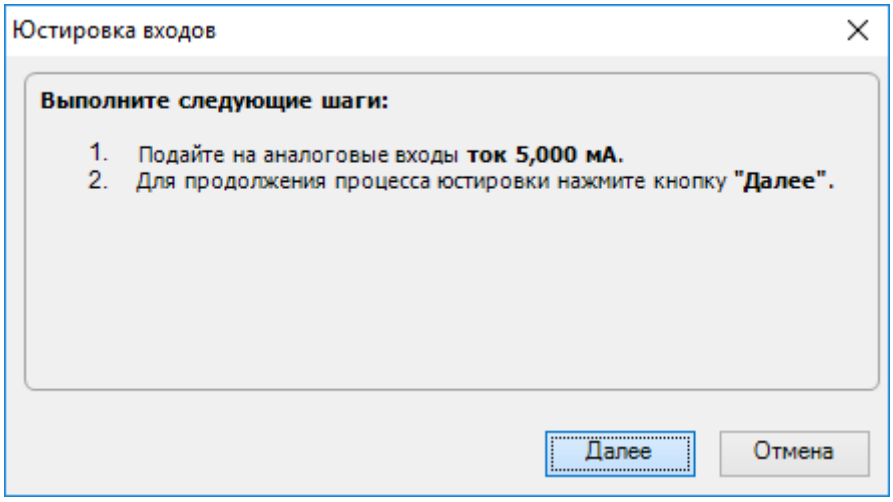


Рисунок 8.13

8. После юстировки появится сообщение об успешном завершении. Программа прибора будет снова запущена.

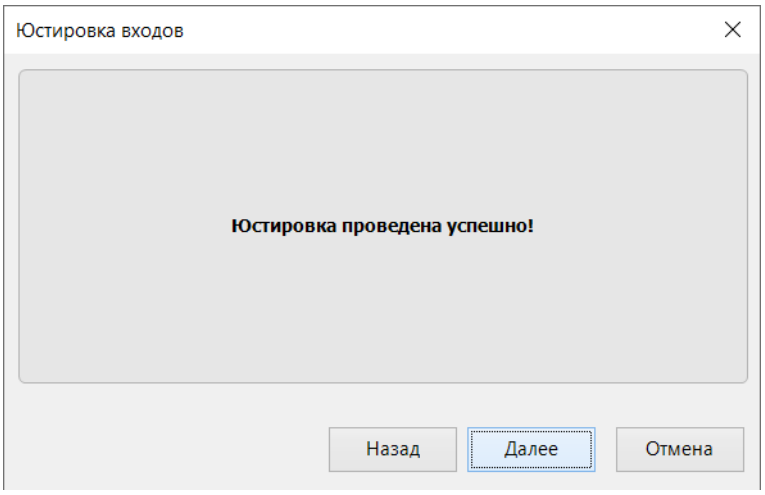


Рисунок 8.14

Юстировка кондуктометрических входов

Для юстировки входов следует:

1. В диалоговом окне юстировки выбрать **Кондуктометрические входы**. Откроется окно настройки.
2. В окне настройки выбрать **Измеряемый сигнал**.
3. Ввести значение **Постоянной времени фильтра**.

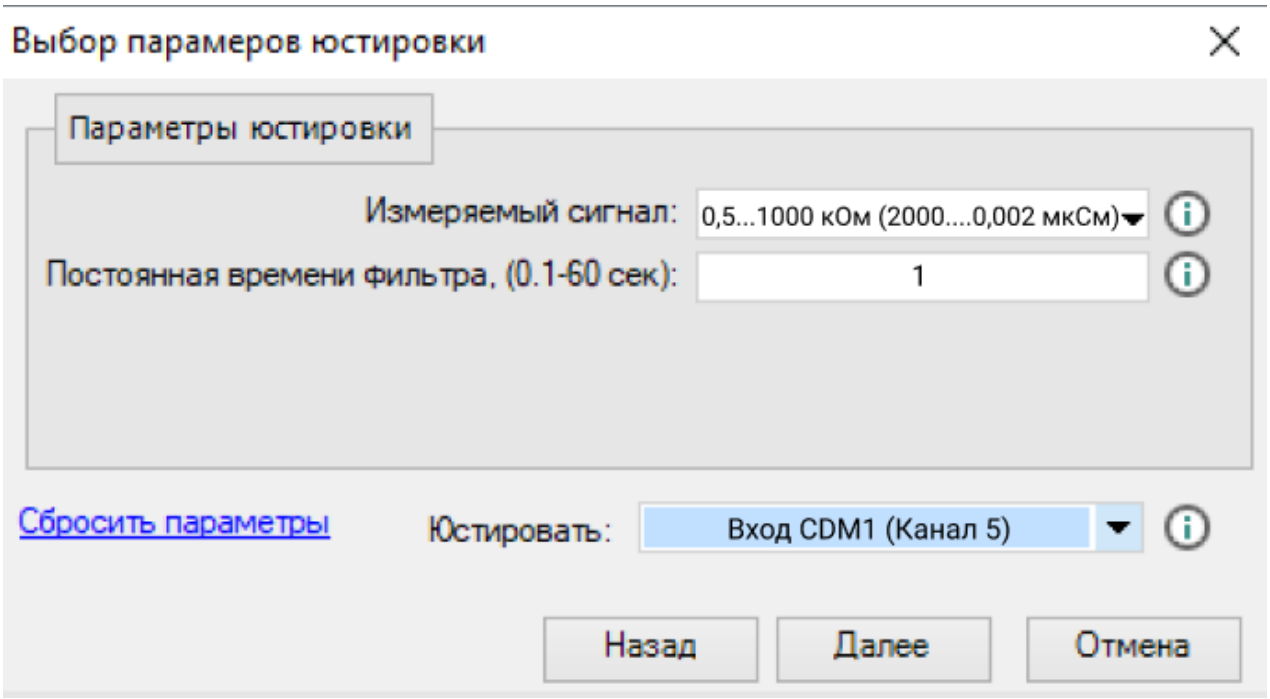


Рисунок 8.15

4. Если необходимо восстановить значения по умолчанию в полях ввода, то следует нажать **Сбросить настройки**.
5. В выпадающем меню **Юстировать** выбрать номер входа для настройки в соответствии с введенными параметрами и нажать **Далее**.
6. В окне выбора параметров юстировки задать точки юстировки.

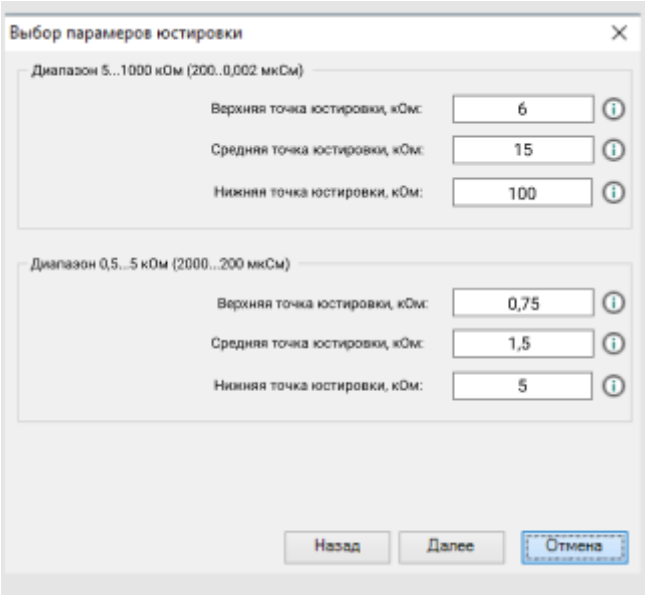


Рисунок 8.16

7. К юстируемым входам подключить сигналы сопротивления с требуемыми характеристиками.

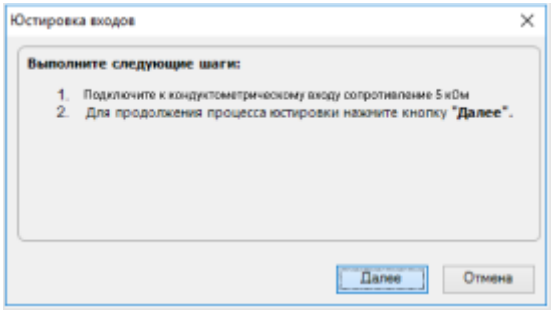


Рисунок 8.17

8. После юстировки появится сообщение об успешном завершении. Программа прибора будет снова запущена.

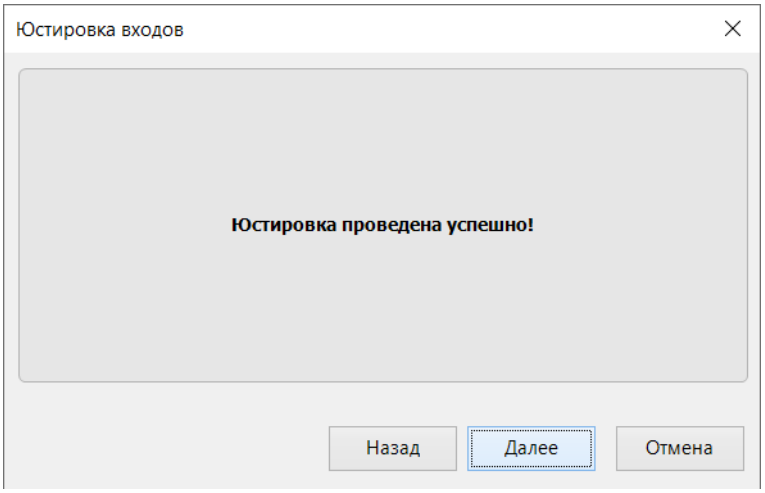


Рисунок 8.18

8.4.2 Юстировка выходов

Для юстировки выходов следует:

1. В диалоговом окне юстировки выбрать **Аналоговые выходы**. Откроется окно настройки.
2. В окне настройки выбрать тип юстируемого выходного сигнала:
   - Дискретный;
   - Аналоговый, напряжение;
   - Аналоговый, ток.

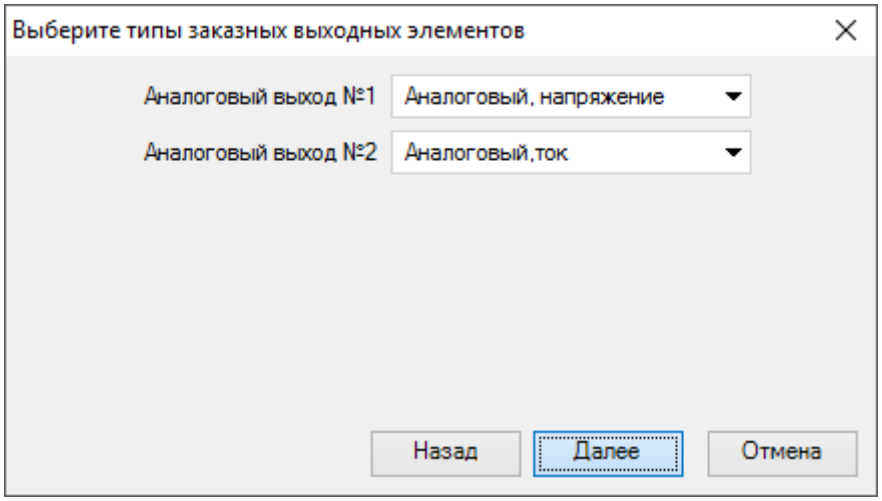


Рисунок 8.19

3. Измерить величину поданного на выход сигнала. Номер выхода указан в правом верхнем углу окна. Ввести значение в поле ввода и нажать **Далее**.

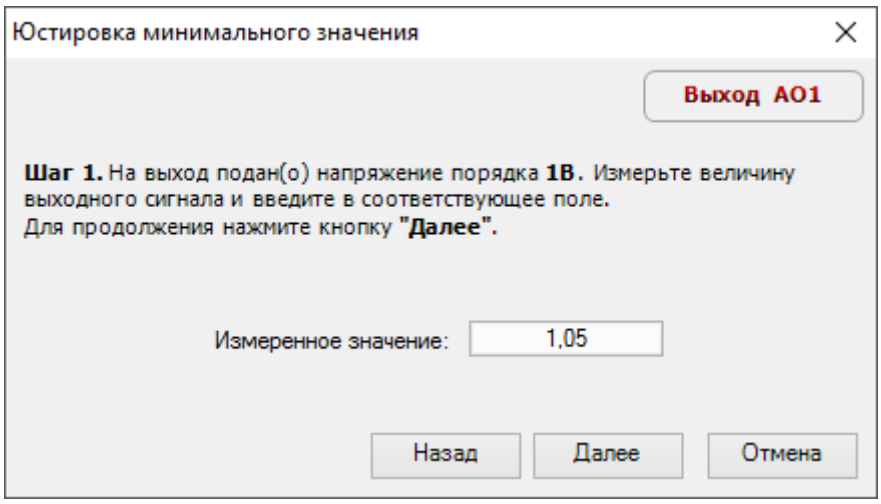


Рисунок 8.20

4. Аналогично измерить и ввести значения для всех типов сигналов.
5. После юстировки появится сообщение об успешном завершении.

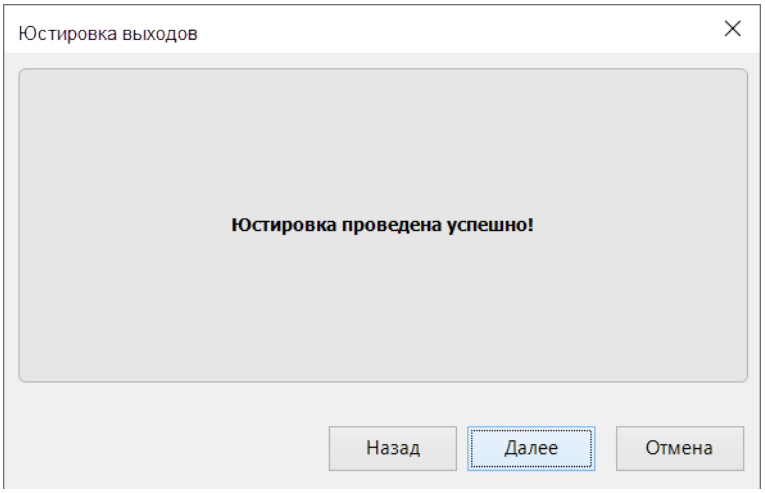


Рисунок 8.21

8.5 Смена целевой платформы

Функция **Смена целевой платформы** предназначена для переноса проекта на другую модификацию прибора, или прибор другой линейки, между приборами одного поколения.

Смена целевой платформы происходит по следующим правилам:

- холст автоматически подстраивается под изменение количества входов/выходов;
- настроенное пользователем расположение входов/выходов остается. Новые входы и выходы добавляются после уже существующих входов/выходов исходного проекта;
- связи входов/выходов, у которых изменился тип данных, разрываются;
- если количество входов/выходов увеличивается, но тип данных исходного набора входов/выходов не изменяется, то связи остаются;
- если количество входов/выходов уменьшается по сравнению с исходным, то связи удаленных входов/выходов разрываются;
- если в исходной платформе были добавлены модули расширения, то они переносятся на новую платформу, связи для них остаются. В случае отсутствия возможности подключения модуля расширения к прибору, на который идет перенос проекта, откроется окно с информацией об удалении связей с переменными, привязанными к модулям расширения;
- настройки аналоговых входов/выходов переносятся (при наличии аналоговых входов/выходов на приборе, на который идет перенос проекта);
- интерфейсы связи переносятся без изменений. В случае отсутствия поддержки интерфейса у прибора, на который идет перенос проекта, откроется окно с информацией об удалении интерфейса, сетевых переменных и связей с ними;

- все экраны с настройками переносятся из одной целевой платформы на другую. В случае отсутствия экрана у прибора, на который идет перенос проекта, откроется окно с информацией об удалении связей с переменными, привязанными к элементам визуализации;

ПРИМЕЧАНИЕ
При смене целевой платформы с ПР225 на ПР205 будет выполнено масштабирование элементов визуализации пропорционально изменению размеров экрана.
При смене целевой платформы с ПР205 на ПР225 элементы визуализации переносятся без масштабирования.

- все переменные переносятся из одной целевой платформы на другую. Если в проекте есть привязка переменных к параметрам, не существующим в приборе на который идет перенос проекта, то откроется окно с информацией об удалении связей с переменными, привязанными к данным параметрам;
- если целевая платформа, на которую осуществляется перенос проекта, не поддерживает функциональные блоки, содержащиеся в исходном проекте, появится информационное окно с уведомлением о необходимости удаления функциональных блоков.

Для смены целевой платформы следует:

1. Открыть проект, который нужно перенести на другой прибор или другую модификацию прибора.
2. Выбрать в меню **Файл** → **Смена целевой платформы**. Owen Logic предложит сохранить проект перед сменой платформы. Далее появится меню со списком доступных для смены платформы приборов и модификаций:

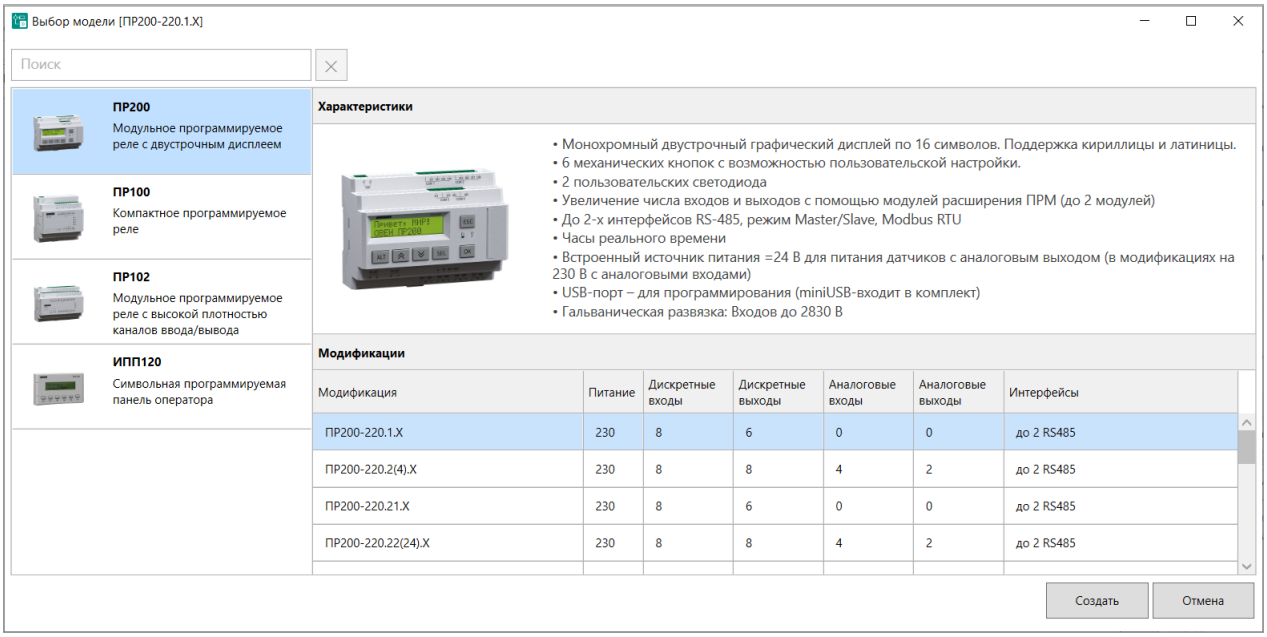


Рисунок 8.22 – Окно выбора модели

3. Выбрать прибор и модификацию, на который будет произведена смена платформы. Нажать кнопку **ОК**.
4. Проверить и восстановить разорванные связи, если они есть. Работу программы можно проверить в Симуляторе.
5. Сохранить измененный проект.

9 Расширения

Расширения используются для создания сложных проектов и их интеграции с другими сервисами компании ОВЕН.

Расширения в Owen Logic:

- Мастер тиражирования 9.1. С помощью **Мастера тиражирования** можно загрузить программу в прибор без участия Owen Logic и защитить ее от копирования;
- Экспорт устройства в OwenCloud 9.3. Позволяет экспортировать конфигурацию сетевых переменных для опроса в облачном сервисе OwenCloud;
- Экспорт устройства в OPC 9.2. Позволяет экспортировать конфигурацию сетевых переменных для опроса в Owen OPC Server.

Установка расширений

Для установки расширений следует:

1. В главном меню выбрать **Расширения** → **Управление расширениями**.

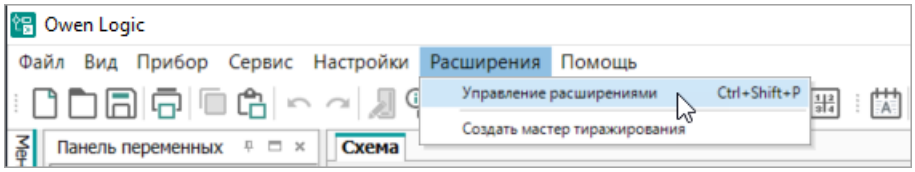


Рисунок 9.1

.

2. В окне **Расширения** отметить галочкой необходимые расширения.

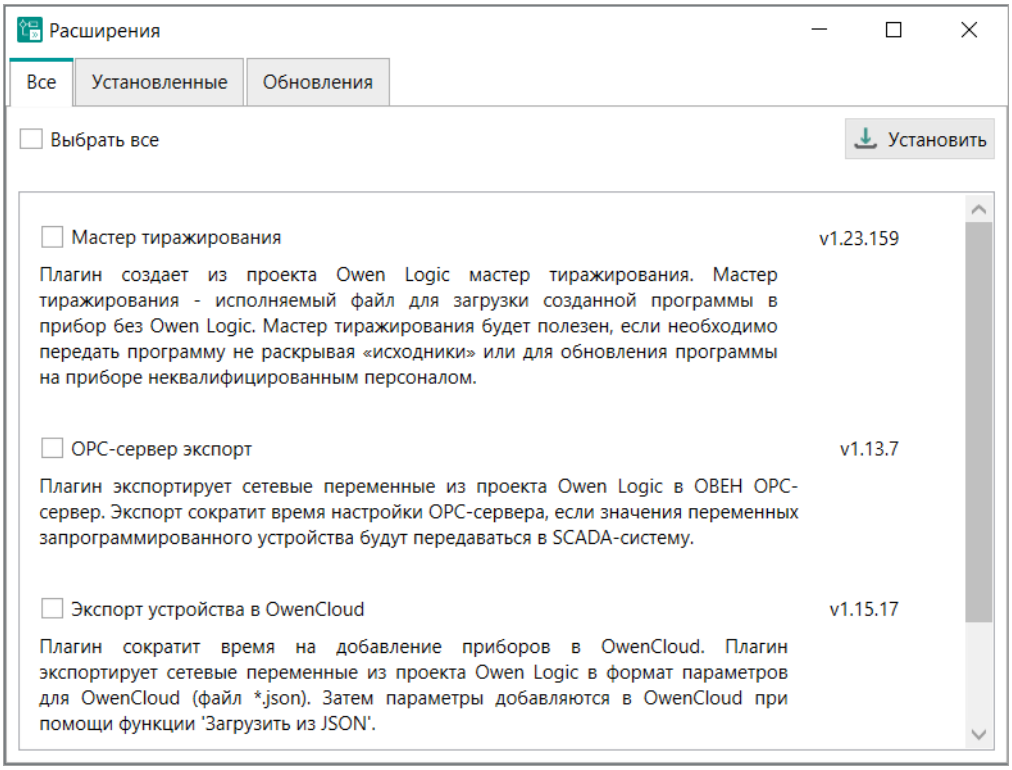


Рисунок 9.2

3. Нажать кнопку **Установить**.
4. Через некоторое время появится сигнал об успешной установке в виде зеленой галочки напротив выбранных расширений .
5. После закрытия окна **Расширения** появится диалоговое окно перезапуска Owen Logic.

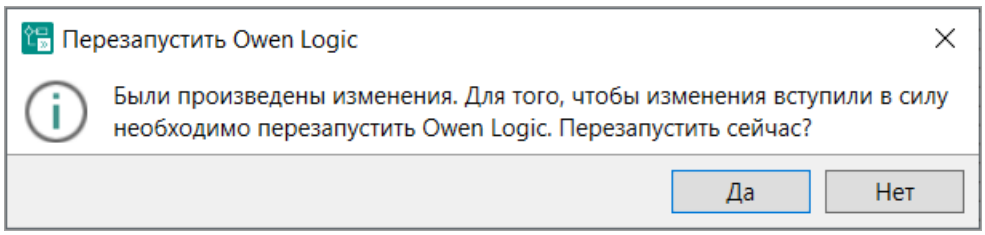


Рисунок 9.3

После перезапуска Owen Logic расширения будут готовы к использованию.

ПРИМЕЧАНИЕ
Если расширения не устанавливаются, следует проверить открыт ли порт 8084.

Удаление расширений

Для удаления расширения следует:

1. В главном меню выбрать **Расширения** → **Управление расширениями**.
2. В окне **Расширения** нажать на кнопку ✕ напротив необходимого расширения.

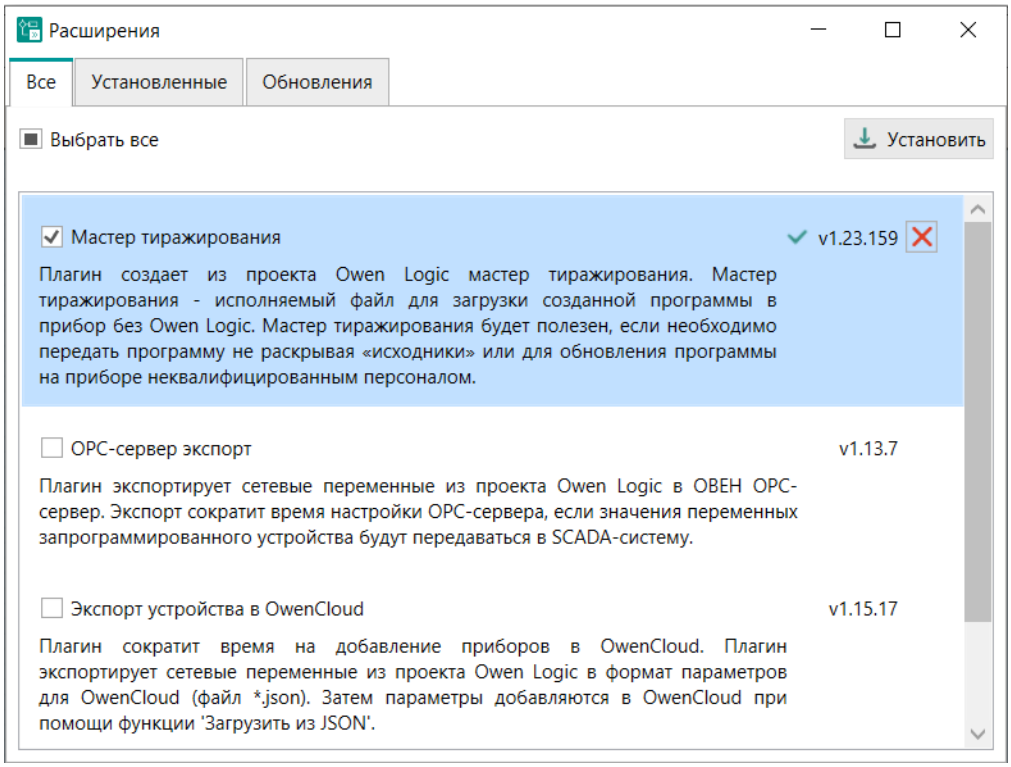


Рисунок 9.4

3. Пиктограмма удаления сменится на пиктограмму установки.
4. После закрытия окна **Расширения** появится диалоговое окно перезапуска Owen Logic. После перезапуска Owen Logic удаленные расширения перестанут быть доступными в главном меню.

9.1 Мастер тиражирования

Мастер тиражирования позволяет создавать из проекта Owen Logic файл, содержащий внутри себя программу, с расширением:

**.exe* — для ОС Windows;

**.dll* — для ОС Linux.

С помощью файла можно загрузить программу в прибор без участия Owen Logic.

ВНИМАНИЕ
Тиражировать программу можно только между одинаковыми модификациями прибора.

Результат записи программы в прибор с помощью мастера тиражирования зависит от наличия ключа в приборе и мастере тиражирования. Ключ в прибор можно записать с помощью Owen Logic.

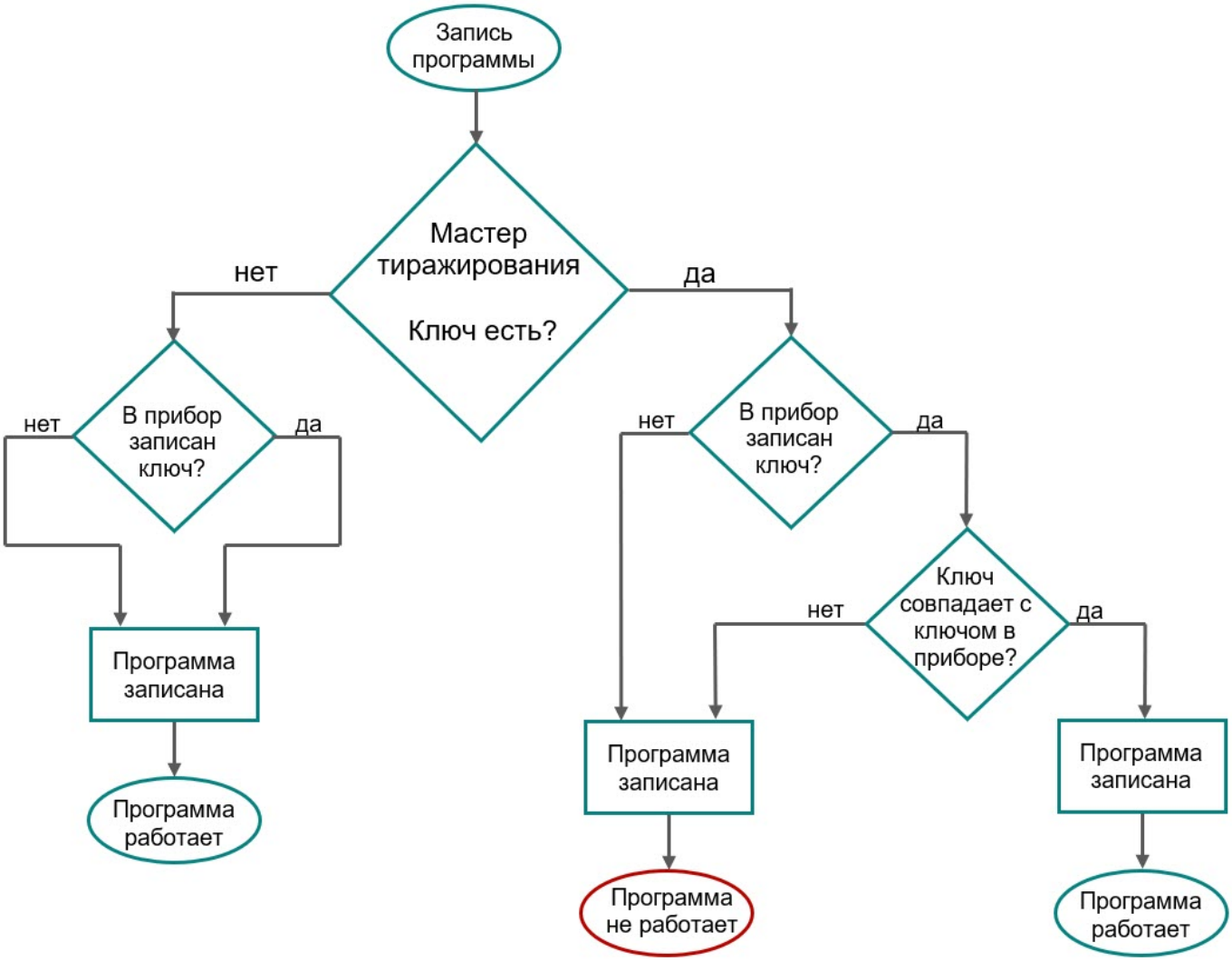


Рисунок 9.5 – Запись программы с помощью мастера тиражирования

Создание мастера тиражирования

Перед созданием мастера тиражирования для ОС Windows рекомендуется создать текстовый файл с расширением **.rtf* с описанием проекта. Описание проекта будет отображаться во время запуска мастера тиражирования перед загрузкой программы в прибор.

Для создания мастера тиражирования для ОС Linux необходимо установить пакет .NET 6 SDK. Это возможно сделать при установке Owen Logic. Для этого поставьте чекбокс **Установить .NET 6 SDK** в окне установки:

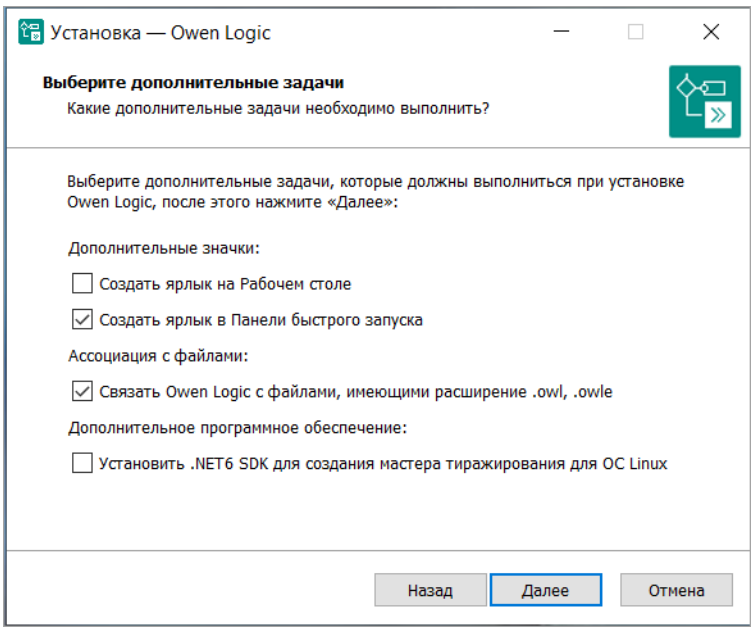


Рисунок 9.6 – Окно установки Owen Logic

В случае, если .NET 6 SDK не был установлен, при создании мастера тиражирования для ОС Linux возникнет ошибка, файл не будет создан. В нижней части окна отобразится информация о необходимости установки и ссылка для скачивания пакета:

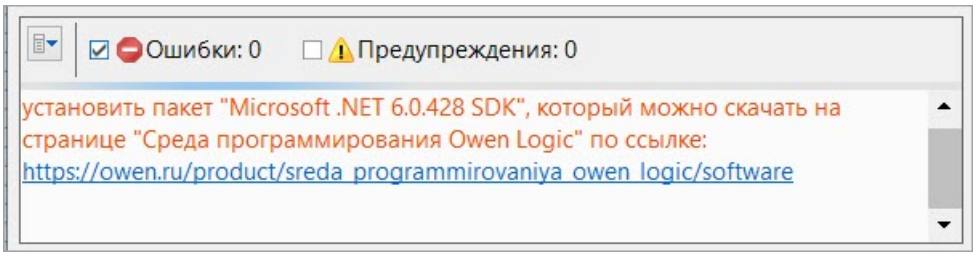


Рисунок 9.7 – Ошибка создания файла

Для создания мастера тиражирования:

1. Укажите на панели свойств проекта в поле **Имя файла с ключом** файл, содержащий ключ, хранящийся на ПК. Для загрузки программы без ключа оставьте это поле пустым.

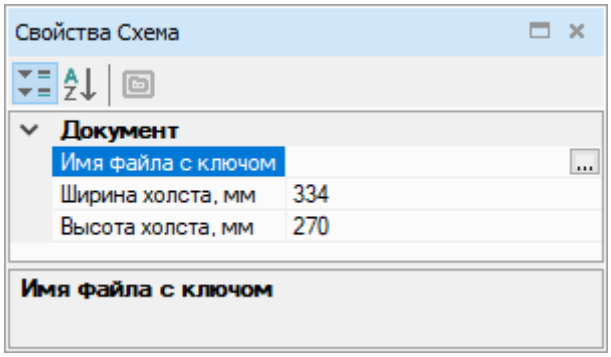


Рисунок 9.8

В случае, если после указания пути к файлу с ключом, ключ будет удален, мастер тиражирования не будет создан, в нижней части окна отобразится информация об ошибке:

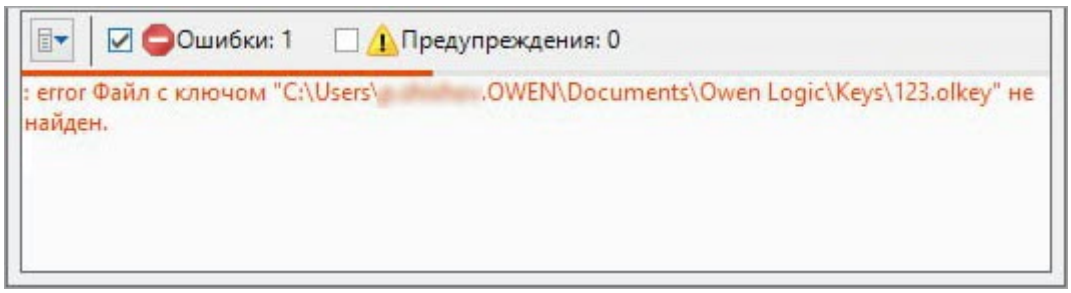


Рисунок 9.9 – Ошибка создания файла

2. Выберите в главном меню **Расширения** → **Создать мастер тиражирования**.
3. Откроется окно для создания мастера тиражирования. Если к проекту подключен ключ, то будет выведена надпись **Пользовательская программа защищена ключом!** В ином случае будет выведена надпись **Устройство поддерживает защиту пользовательской программы, но для проекта не выбран ключ**.

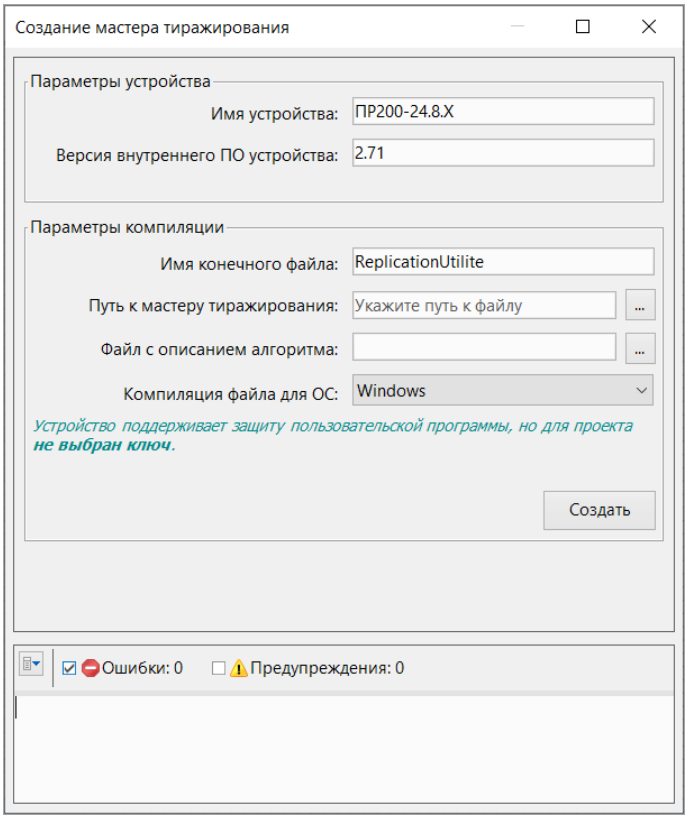


Рисунок 9.10 – Создание мастера тиражирования

Параметры устройства считываются из прибора и заполняются автоматически.

В окне параметров компиляции заполните поля:

- *Имя конечного файла* — имя файла мастера тиражирования, который будет создан;
- *Путь к мастеру тиражирования* — расположение создаваемого файла;
- *Файл с описанием алгоритма (для ОС Windows)* — путь к текстовому файлу с описанием проекта в формате **.rtf* (не обязательно);
- *Компиляция файла для ОС* — ОС для которой создается файл.

4. Нажмите кнопку **Создать**. В случае отсутствия ошибок будет создан файл с указанным именем и расширением **.exe* или **.dll*.

ПРИМЕЧАНИЕ
Создание мастера тиражирования для ОС Linux может занять довольно продолжительное время.

ПРИМЕЧАНИЕ
Расширение файла **.dll* может не отображаться в ОС Windows, если не включена опция отображения системных файлов:

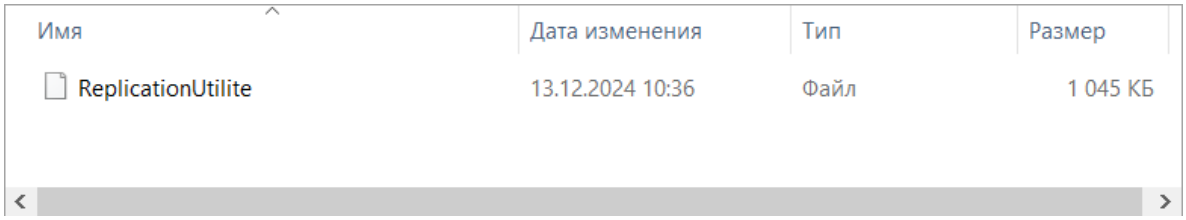


Рисунок 9.11 – Отображение файла с расширением **.dll*

Отчет о создании файла выводится в нижней части окна.

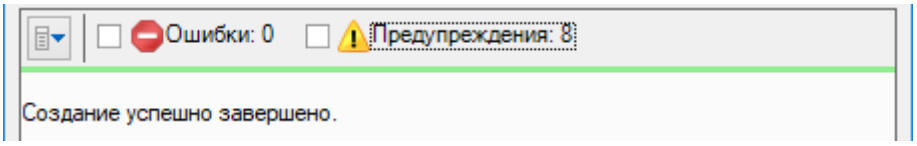


Рисунок 9.12 – Отчет о создании файла

Чтобы скрыть или открыть область отображения отчета, следует нажать кнопку . Информация о предупреждениях и ошибках выводится, если отмечены соответствующие флажки.

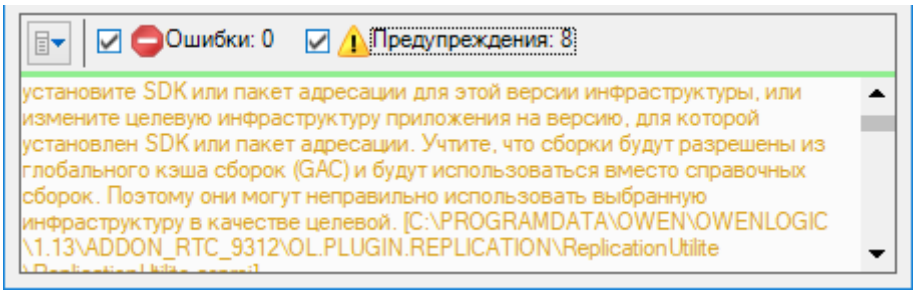


Рисунок 9.13 – Примеры предупреждений

9.1.1 Запись программы в прибор через мастер тиражирования на ОС Windows

Для записи программы в прибор с помощью мастера тиражирования следует:

1. Запустить созданный файл мастера тиражирования.

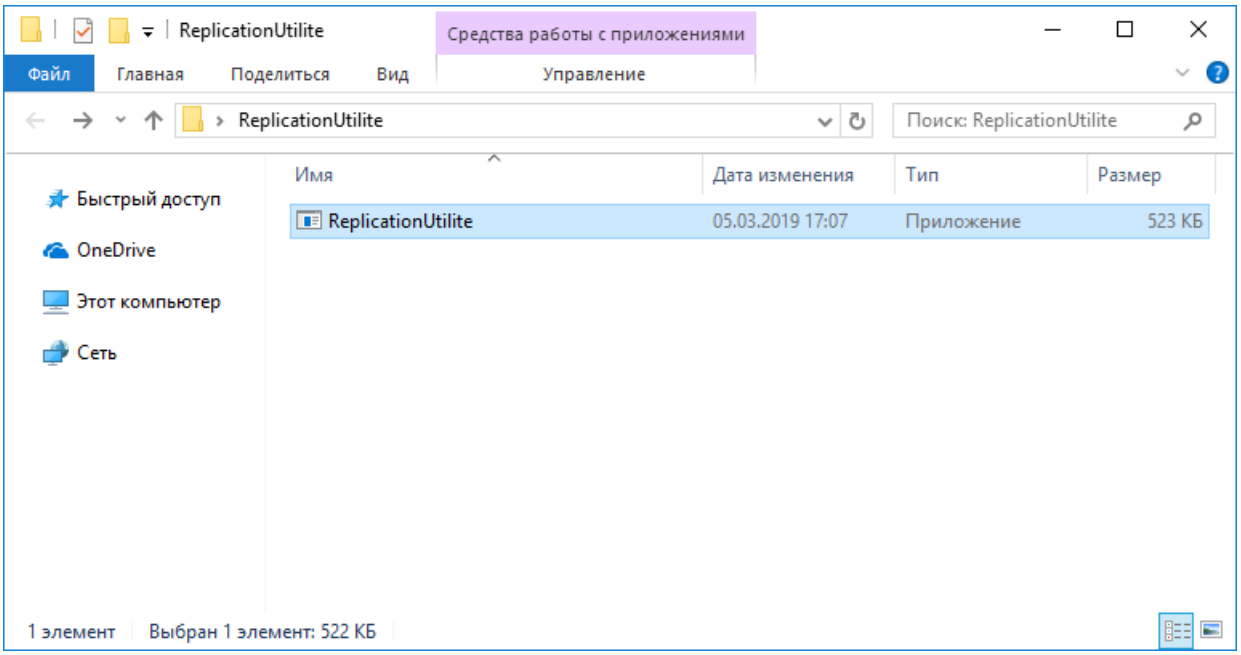


Рисунок 9.14 – Файл мастера тиражирования

ПРИМЕЧАНИЕ
Во время открытия мастера отобразится текст описания проекта из прикрепленного файла с расширением **.rtf*. Если файл не был прикреплен во время создания мастера, то описание проекта будет пустым.

2. Следовать указаниям мастера тиражирования.

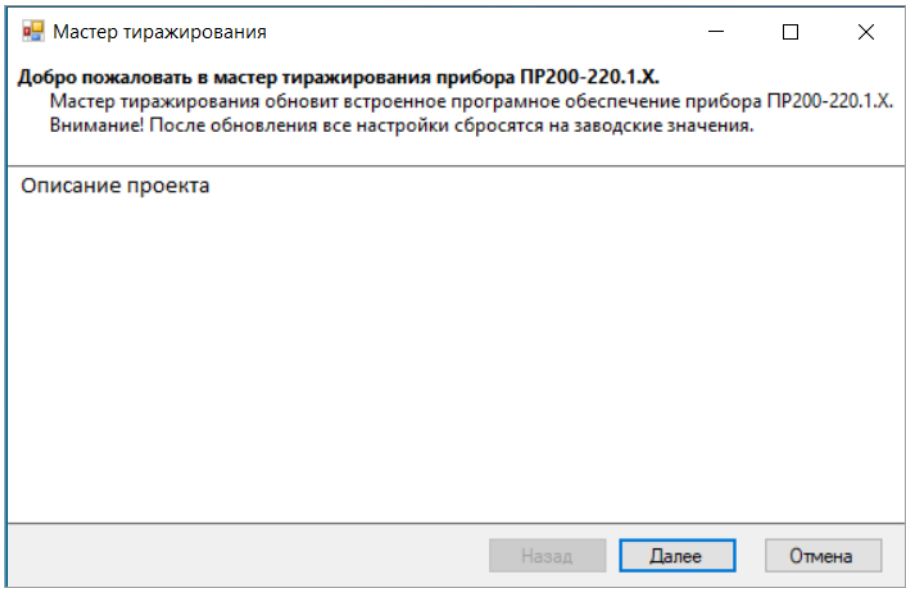


Рисунок 9.15 – Окно Мастер тиражирования

3. В разделе **Настройки порта** указывается номер COM-порта, к которому подключен прибор. Как узнать номер порта описано в разделе Настройка порта 4.6.
4. Если устройство защищено паролем введите пароль в окне мастера тиражирования. В случае отсутствия или ввода неверного пароля кнопка **Далее** будет неактивна. После выбора порта в правом нижнем углу будет отображаться состояние подключения к прибору.

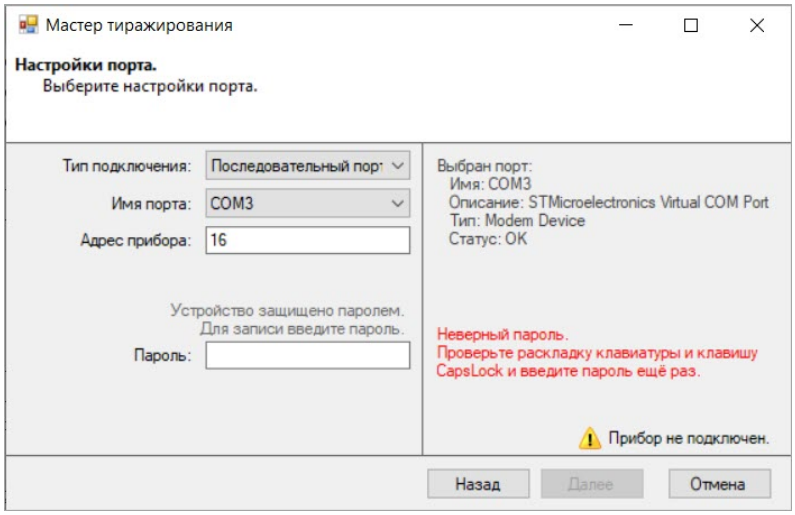


Рисунок 9.16 – Ошибка подключения прибора

В случае ошибки подключения будет отображен знак и текст ошибки. При наведении на знак отобразится подсказка с описанием ошибки.

Ошибка	Возможные причины
Прибор не подключен	• неправильно выбран COM-порт; • не установлен драйвер USB; • разрыв соединения.
Требуется другой прибор	Подключенный прибор не соответствует проекту. Нужно заменить прибор в проекте и создать новый мастер тиражирования, либо подключить подходящий прибор

Если никаких ошибок нет, то отображается зеленая галочка с названием прибора.

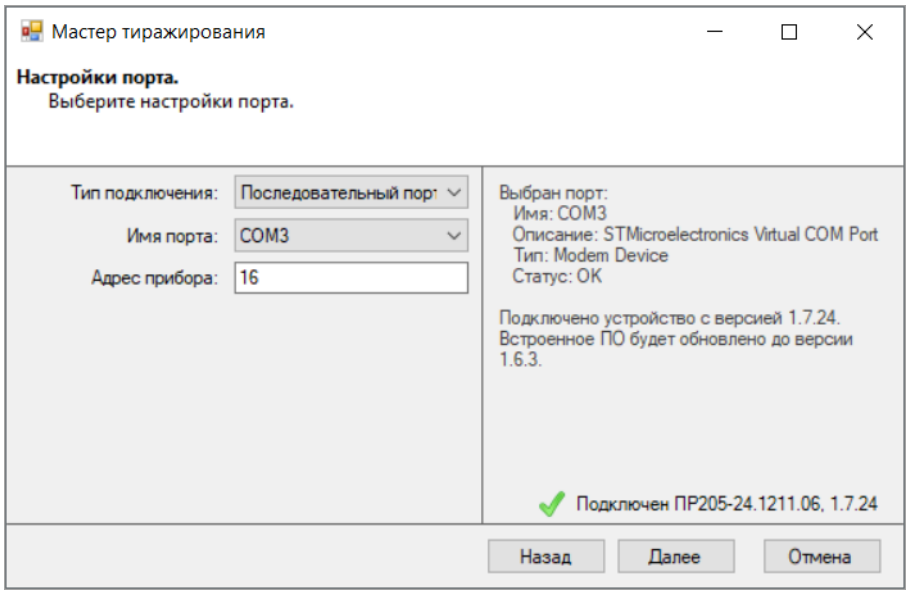


Рисунок 9.17

Если связь с прибором постоянно прерывается (галочка мигает), то, возможно, другое приложение занимает порт USB. Это может быть, если запущена программа Owen Logic и в ней настроено подключение к этому же порту. В таком случае для загрузки проекта в прибор другое приложение следует закрыть или перевести в режим OFFLINE .

5. Если соединение стабильно, то следует нажать кнопку **Далее**.

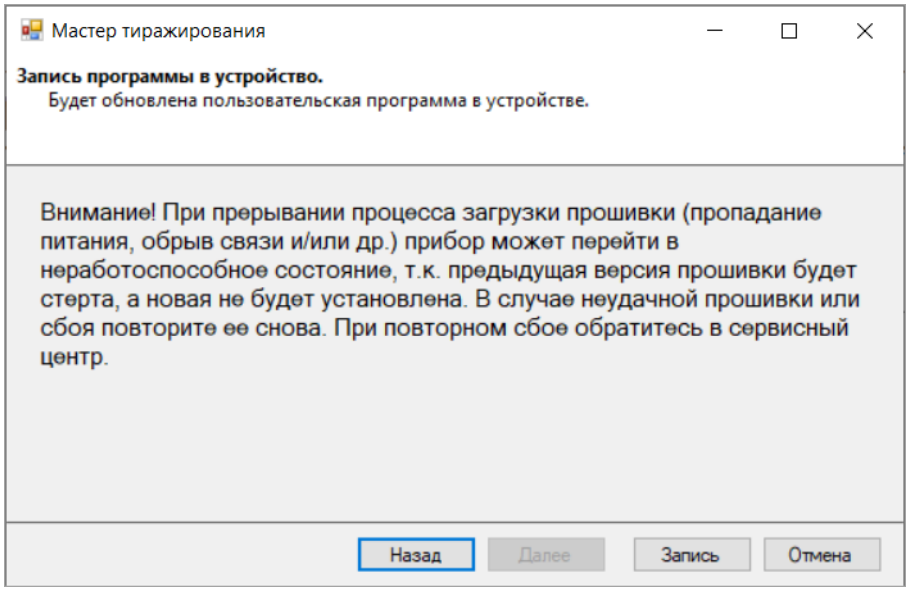


Рисунок 9.18 – Запись программы

6. Для записи программы в устройство следует нажать кнопку **Запись**. В нижней части окна отобразится строка состояния записи. Во время загрузки программы нельзя прерывать связь ПК с прибором. По завершении загрузки будет выведено сообщение.

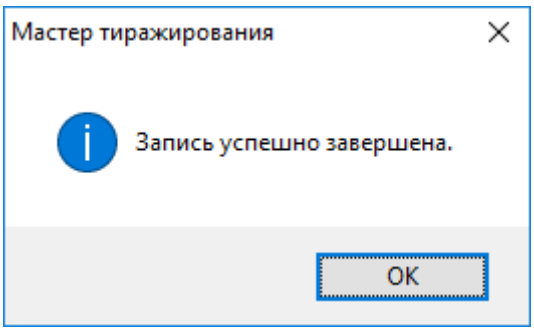


Рисунок 9.19 – Завершение загрузки

Режим «тихой» установки

ПРИМЕЧАНИЕ
Режим «тихой» установки доступен при работе с мастером тиражирования версии 1.21.14 и старше.

Мастер тиражирования поддерживает режим «тихой» прошивки приборов. Для его запуска следует использовать следующие аргументы командной строки:

- ***/silent*** – аргумент «тихой» установки;
- ***/password:*** – пароль, необходим, если прибор был защищён паролем;
- **/ip:** – IP адрес подключенного прибора (если прибор подключен по интерфейсу Ethernet);
- **/portname:** – номер COM-порта (если прибор подключен по интерфейсу USB);
- **1>> File_Name.txt** – текстовый файл с отображением процесса прошивки в процентах;
- **2>> File_Name.txt** – текстовый файл с отображением ошибок в процессе прошивки.

Пример команды запуска файла мастера тиражирования ReplicationUtilite.exe, расположенного на рабочем столе, при подключении по интерфейсу Ethernet:

C:\Users\<Имя пользователя\Desktop>ReplicationUtilite.exe /silent /IP:10.2.3.204

Пример команды запуска файла мастера тиражирования ReplicationUtilite.exe, расположенного на рабочем столе, при подключении по интерфейсу USB к прибору с паролем *1234*:

C:\Users\<Имя пользователя\Desktop>ReplicationUtilite.exe /silent /portname:COM7 /password:1234

Пример команды запуска файла мастера тиражирования ReplicationUtilite.exe, расположенного на рабочем столе, при подключении по интерфейсу USB, для создания текстового файла с отображением ошибок в процессе прошивки:

C:\Users\<Имя пользователя\Desktop>ReplicationUtilite.exe /silent /portname:COM7 2>> File_Name.txt

9.1.2 Запись программы в прибор через мастер тиражирования на ОС Linux

Требования к ПК:

- Ubuntu 22.04 или 24.04, архитектура процессора x86/32 или x64;
- .NET 6.

Для записи программы в прибор на ОС Linux откройте командную строку и введите команду:

- **./ReplicationUtilite -ip:192.168.1.100** — при подключении прибора по Ethernet, где 192.168.1.100 — IP - адрес прибора;
- **./ReplicationUtilite -portname:/dev/ttyACM0** — при подключении по последовательному порту, где ACM0 — адрес порта.

ПРИМЕЧАНИЕ
Для определения порта устройства используйте команды:

- **lsusb** — для USB;
- **dmesg | grep tty** — для последовательного порта.

Пример команды запуска файла мастера тиражирования при подключении прибора по последовательному порту с адресом ACM0:

```
:~$ ./ReplicationUtilite -portname:/dev/ttyACM0
```

Рисунок 9.20 – Команда для записи программы

В случае верно выполненных действий в командной строке отобразится строка состояния загрузки программы в прибор:

```
[#######################################-----------]  77% /
```

Рисунок 9.21 – Строка состояния

Таблица 9.1 – Возможные ошибки

Ошибка	Решение
Отсутствуют права на запуск файла `Permission denied`	• обратитесь к администратору; • используйте команду **sudo ./ReplicationUtilite -portname:/dev/ttyACM0**, где ACM0 — адрес порта.
Ввод некорректной команды `Указаны неверные аргументы командной строки`	Введите допустимую команду для запуска файла.
Устройство защищено паролем `One or more errors occurred. (Error occurred during a cryptographic operation.)`	Введите команду **-password:**.

9.2 Экспорт устройства в OPC

Расширение **Экспорт устройства в OPC** экспортирует сетевые переменные из проекта Owen Logic для использования в ПО Owen OPC Server.

Для экспорта следует:

1. Экспортировать устройство из Owen Logic – выбрать в главном меню **Расширения** → **Экспорт устройства в OPC**.

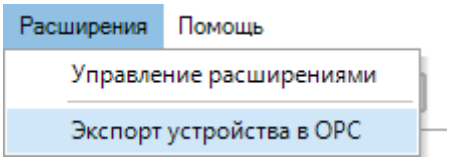


Рисунок 9.22

   Появится окно проводника, где следует выбрать место сохранения и имя файла с расширением **.dev*.

2. В Owen OPC Server добавить узел в дерево объектов.
3. В контекстном меню узла выбрать **Добавить** → **Устройство из файла**. Откроется окно проводника.
4. В окне выбрать файл, который был экспортирован из Owen Logic. Нажать кнопку **Открыть**.
5. Выбрать протокол обмена с устройством. Нажать **Выбрать**.

Устройство и теги будут добавлены в дерево объектов.

Если необходимо отредактировать устройство, то следует изменить свойства устройства и тегов.

Для проверки правильности настройки следует подключить прибор к ПК и запустить режим опроса.

Работа с Owen OPC Server описывается в справке к ПО.

9.3 Экспорт устройства в OwenCloud

Экспорт устройства в OwenCloud сокращает время добавления прибора, настроенного в режим Slave, в облачный сервис компании ОВЕН.

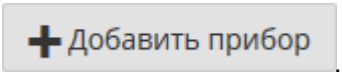
Работа с OwenCloud требует подключения специального устройства, шлюза, к прибору. Будем считать, что шлюз уже настроен и подключен к прибору, настройки приведены в *Руководстве по эксплуатации* на шлюз.

Для экспорта конфигурации параметров Modbus прибора следует:

1. Создать проект с настройками в режиме Slave и сетевыми переменными.
2. Выбрать в главном меню **Расширения** → **Экспорт устройства в OwenCloud**.
3. В открывшемся меню выбрать расположение и задать имя файла в формате **.json*.

Дальнейшие действия требуют наличия личного кабинета в OwenCloud.

После входа в личный кабинет следует:

1. Нажать на кнопку .
2. На открывшейся странице нажать кнопку 

.
3. В появившемся меню следует задать **Идентификатор** — IMEI/заводской номер сетевого шлюза (указан на корпусе шлюза). В выпадающем списке **Тип прибора** выбрать **Произвольный прибор Modbus**. Ввести адрес прибора, который был задан в Owen Logic. Заполнить остальные поля. Далее следует нажать кнопку **Добавить**.

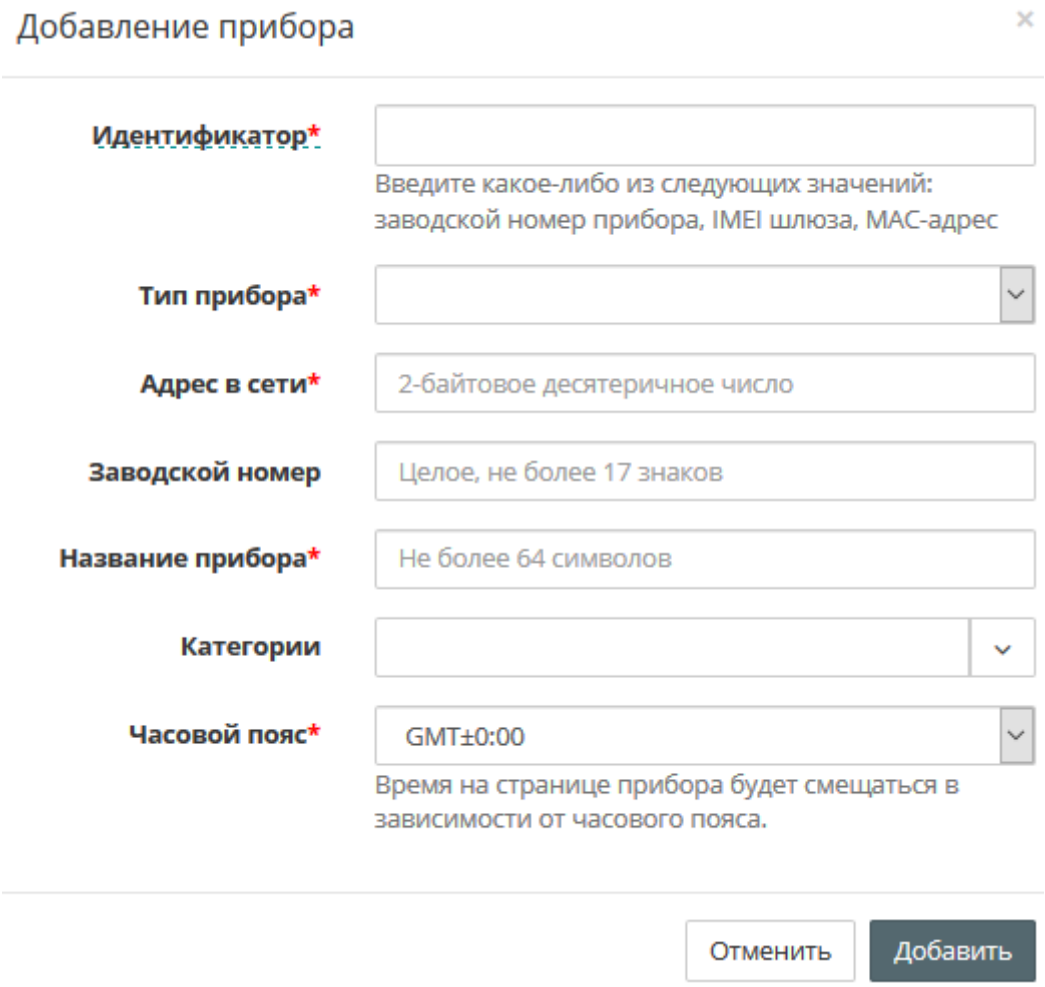


Рисунок 9.23

4. На вкладке **Общее/Общие настройки** указать скорость опроса и настройки COM-порта прибора в соответствии с настройками в Owen Logic. Нажать кнопку **Сохранить** для применения новых настроек.

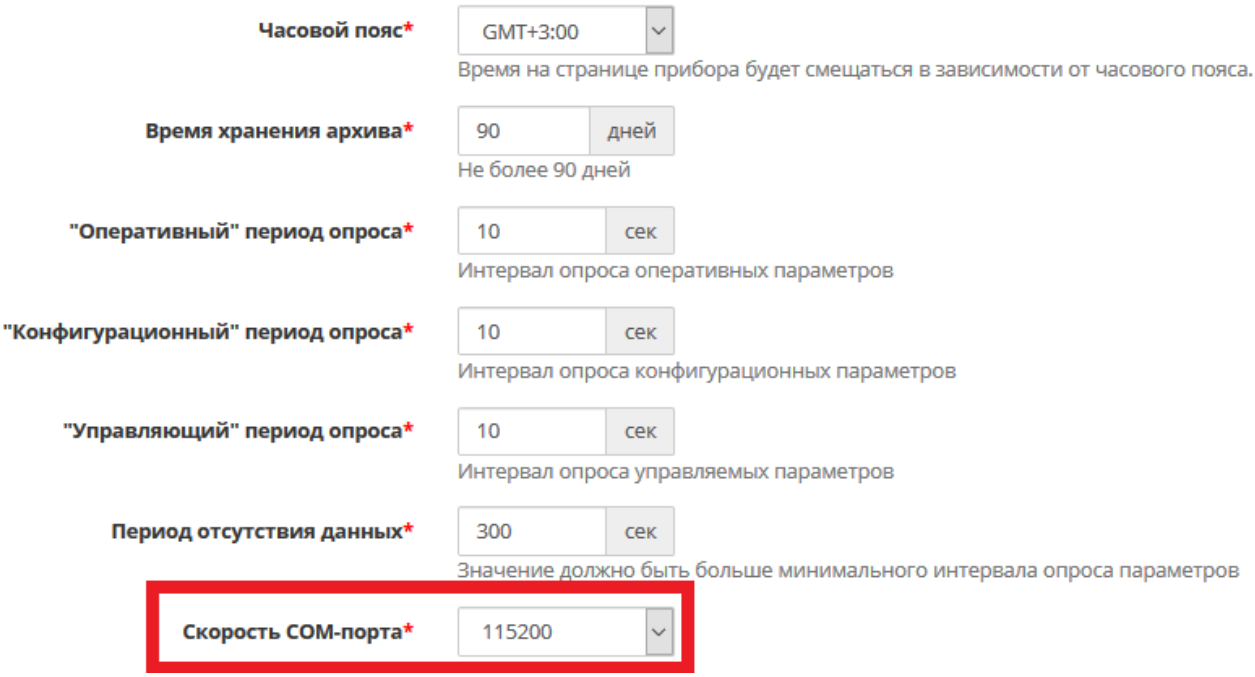


Рисунок 9.24

5. В появившейся странице нажать кнопку **Настройки параметров**. Нажать на выпадающий список **Импортировать** и выбрать вариант **Загрузить из JSON**. В открывшемся меню выбрать ранее созданный файл в формате **.json* и нажать **Загрузить параметры**.

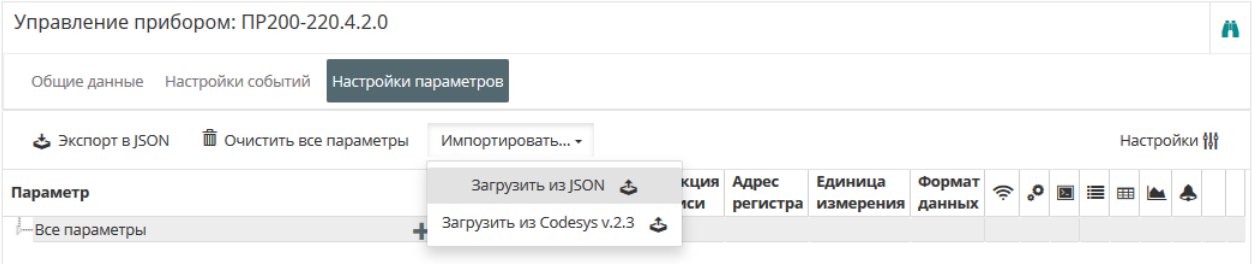


Рисунок 9.25

6. Modbus переменные прибора будут добавлены в OwenCloud.

10 Сочетания клавиш

Сочетание клавиш	Действие
Меню/Файл	
Ctrl + N	Создать новый проект
Ctrl + O	Открыть существующий проект
Ctrl + Alt + S	Сохранить открытый проект под другим именем
Ctrl + S	Сохранить открытый проект
Ctrl + P	Печать
Ctrl + Shift + C	Открыть менеджер компонентов
Меню/Вид	
Ctrl + Z	Отменить последнее изменение
Ctrl + Y	Вернуть (восстановить) отмененное действие
Меню/Прибор	
Ctrl + F7	Записать программу в прибор
Ctrl + Shift + V	Открыть таблицу переменных
Ctrl + Shift + S	Открыть настройки прибора
Меню/Сервис	
Shift + F5	Перейти в режим симуляции
F6	Запустить симуляцию
F7	Остановить симуляцию
F8	Пауза симуляции
F10	Шаг на один цикл
Ctrl + F5	Перейти в режим отладки
Меню/Расширения	
Ctrl + Shift + P	Открыть управление расширениями
Меню/Помощь	
F1	Вызов справки
Панель вставки	
Ctrl + Shift + F	Создать функцию на ST
Ctrl + Shift + M	Создать макрос
Ctrl + M	Создать макрос из выделенного участка
Панель «Места использования»	
F5	Обновить места использования функции на ST
Клавиши работы с элементами на схеме	
Ctrl + C	Копировать выделенный элемент(ы) в буфер обмена
Ctrl + V	Вставить из буфера обмена
Delete	Удаление выбранного элемента
Клавиши изменения размера элемента на схеме	
Ctrl + →	Увеличение ширины выделенного элемента
Ctrl + ←	Уменьшение ширины выделенного элемента
Ctrl + ↓	Уменьшение высоты выделенного элемента
Ctrl + ↑	Увеличение высоты выделенного элемента
Масштабирование холста	
Ctrl + Колесо мыши	При вращении колеса мыши «от себя» масштаб холста увеличивается. Если «на себя», то уменьшается
Ctrl + «+»	Увеличение масштаба
Ctrl + «–»	Уменьшение масштаба

Сочетание клавиш	Действие
Переход между вкладками	
Tab + →	Переход между вкладками
Tab + ←	
Перемещение элементов визуализации на экране	
Alt + →	Сдвиг элемента (группы элементов) на 1 px вправо, при включенной сетке (вне зависимости от шага сетки)
Alt + →	Сдвиг элемента (группы элементов) на 1 px вправо, при включенной сетке (вне зависимости от шага сетки)
Alt + ←	Сдвиг элемента (группы элементов) на 1 px влево, при включенной сетке (вне зависимости от шага сетки)
Alt + ↓	Сдвиг элемента (группы элементов) на 1 px вниз, при включенной сетке (вне зависимости от шага сетки)
Alt + ↑	Сдвиг элемента (группы элементов) на 1 px вверх, при включенной сетке (вне зависимости от шага сетки)
Alt + ЛКМ	Сдвиг элемента (группы элементов) по направлению движения курсора мыши с шагом 1 px, при включенной сетке (вне зависимости от шага сетки)
Shift + →	Сдвиг элемента (группы элементов) на 10 px вправо
Shift + ←	Сдвиг элемента (группы элементов) на 10 px влево
Shift + ↓	Сдвиг элемента (группы элементов) на 10 px вниз
Shift + ↑	Сдвиг элемента (группы элементов) на 10 px вверх

11 Примеры проектов

В данном разделе на примерах объясняется процесс создания программ в Owen Logic.

- Вывод значения переменной на экран;
- Задание уставок таймеров с экрана;
- Подключения ПР200 к OwenCloud через шлюз ПМ210 по протоколу Modbus RTU;
- Подключение приборов второго поколения к OwenCloud;
- Упаковка/распаковка бит/целых чисел;
- Обработка битовых переменных с помощью маски.

11.1 Вывод значения переменной на экран

Задача – на приборе с подключенным к нему по интерфейсу связи модулю ввода необходимо считать значения на входах 1 и 2 модуля и вывести их на дисплей прибора.

Для разработки программы следует:

1. Создать новый проект в **Owen Logic**.
2. Подключить шаблон сетевого устройства в проект. Для этого следует запустить **менеджер компонентов** и отсортировать компоненты по категории **Шаблоны сетевых устройств**. Из сгенерированного списка выбрать **MB110–8A** и нажатием на кнопку **Загрузить в библиотеку проекта** скачать шаблон.

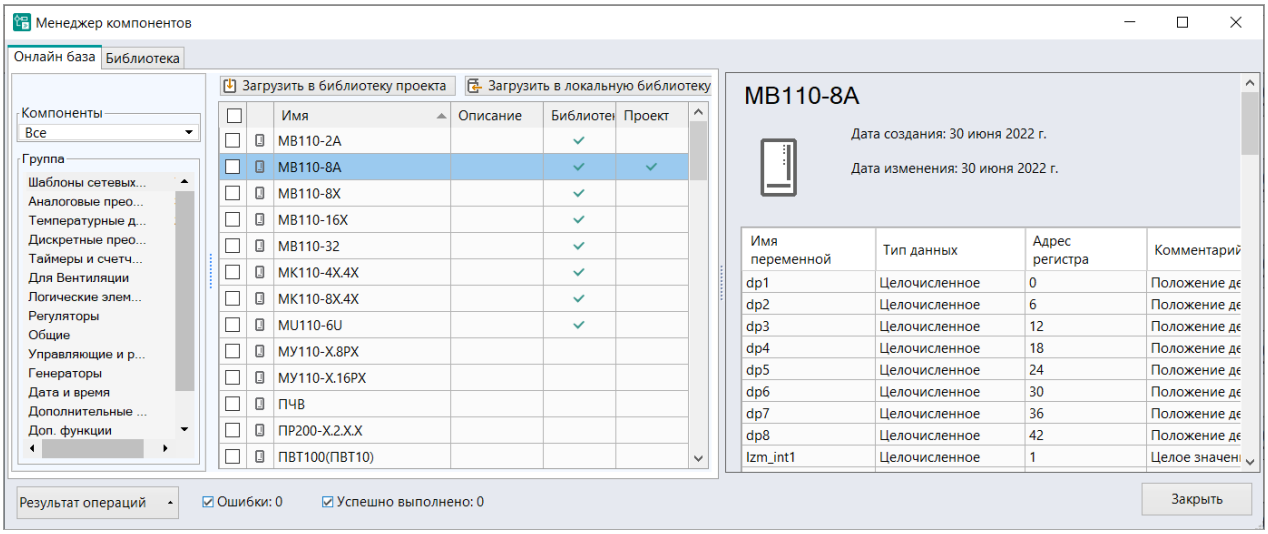


Рисунок 11.1

3. В окне **настроек прибора** подключить этот шаблон .
4. После добавления шаблона сетевого устройства в **таблице переменных** следует уточнить имена и типы переменных, которые планируется выводить на экран (**Izm_fl** и **Izm_fl2**). Это температуры, измеренные модулем, и передаваемые в прибор по протоколу Modbus. Переменные имеют тип с плавающей запятой.

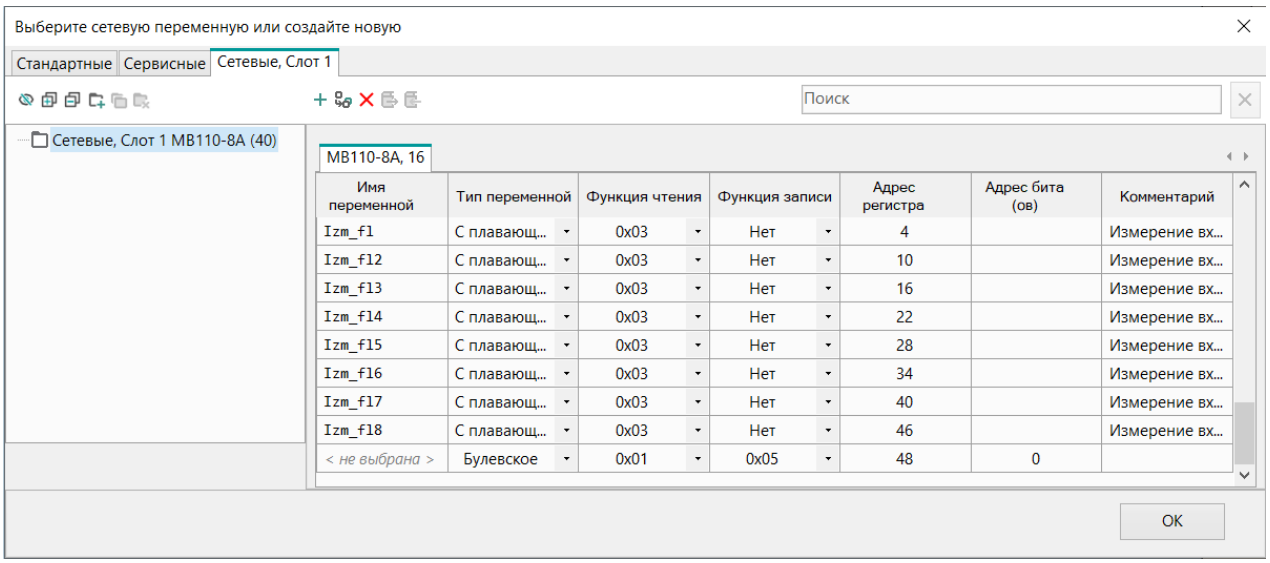


Рисунок 11.2

5. Перейти в **менеджер экранов**.
6. Двойным нажатием по экрану открыть редактор экранов.
7. Из библиотеки компонентов добавить элемент **Ввод/вывод (int/float)**.
8. Открыть свойства элемента и указать тип переменной – **Вещественный**. Затем в строке **Переменная** нажать на кнопку «**…**».

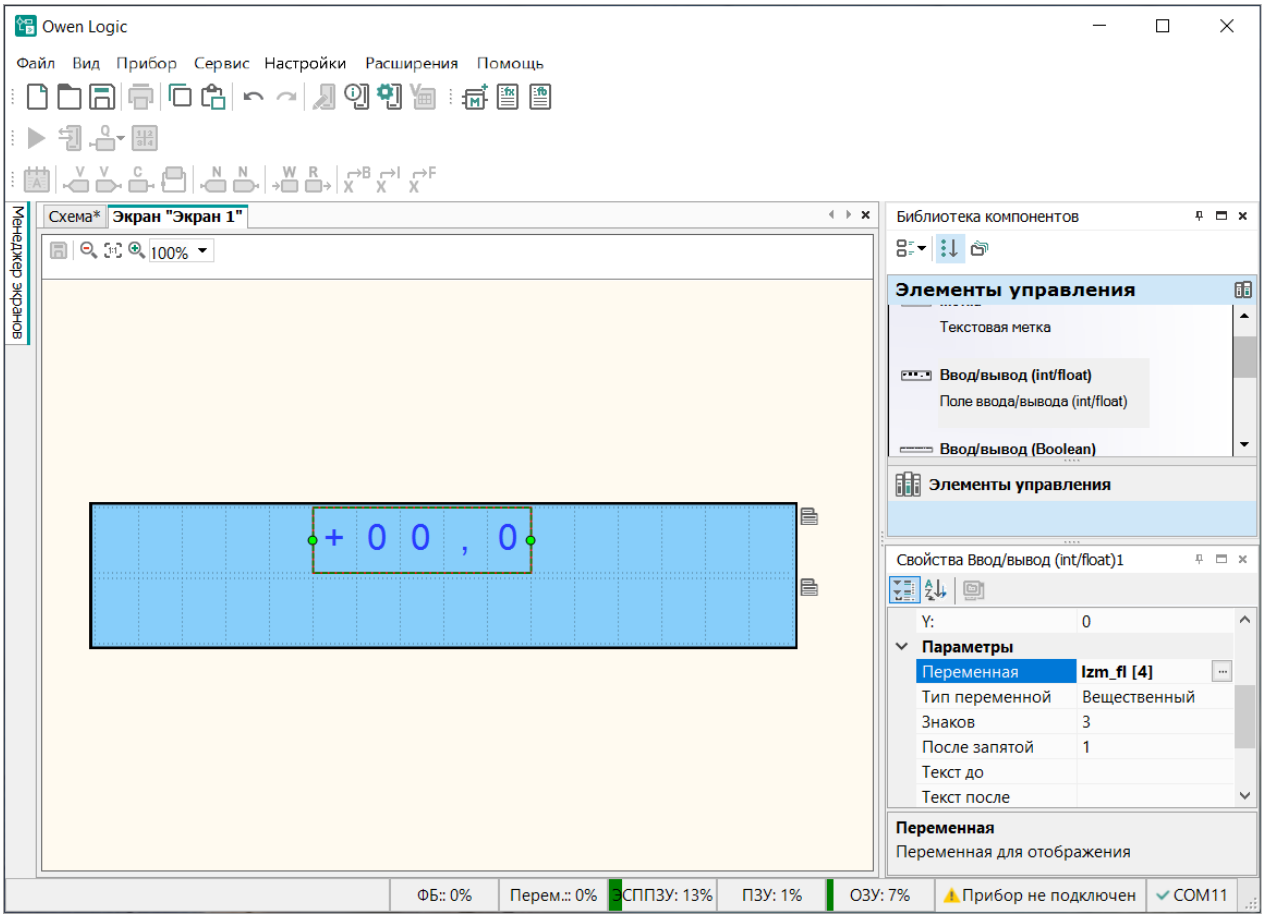


Рисунок 11.3

9. В **таблице переменных** выбрать переменную **Izm_fl** на вкладке **Сетевые, Слот1**.
10. В поле **Знаков** в свойствах элемента указать необходимое количество знаков для вывода значения выбранной переменной. В поле **После запятой** указать количество дробных разрядов. В поле **Длина**

указать общее количество знаков элемента. В поле **Текст до** набрать **T1** и поставить пробел для отделения значения от текста

11. В поле **Редактируемо** поставить **Нет**, так как менять измеренные на модуле значения температур нельзя.

12. На этом настройка одного элемента **Ввод/вывод (int/float)** завершена. Для упрощения дальнейшей работы следует скопировать элемент со всеми настройками, нажав на него ПКМ и выбрав в контекстном меню **Копировать** (или следует воспользоваться сочетанием клавиш **Ctrl + C**).

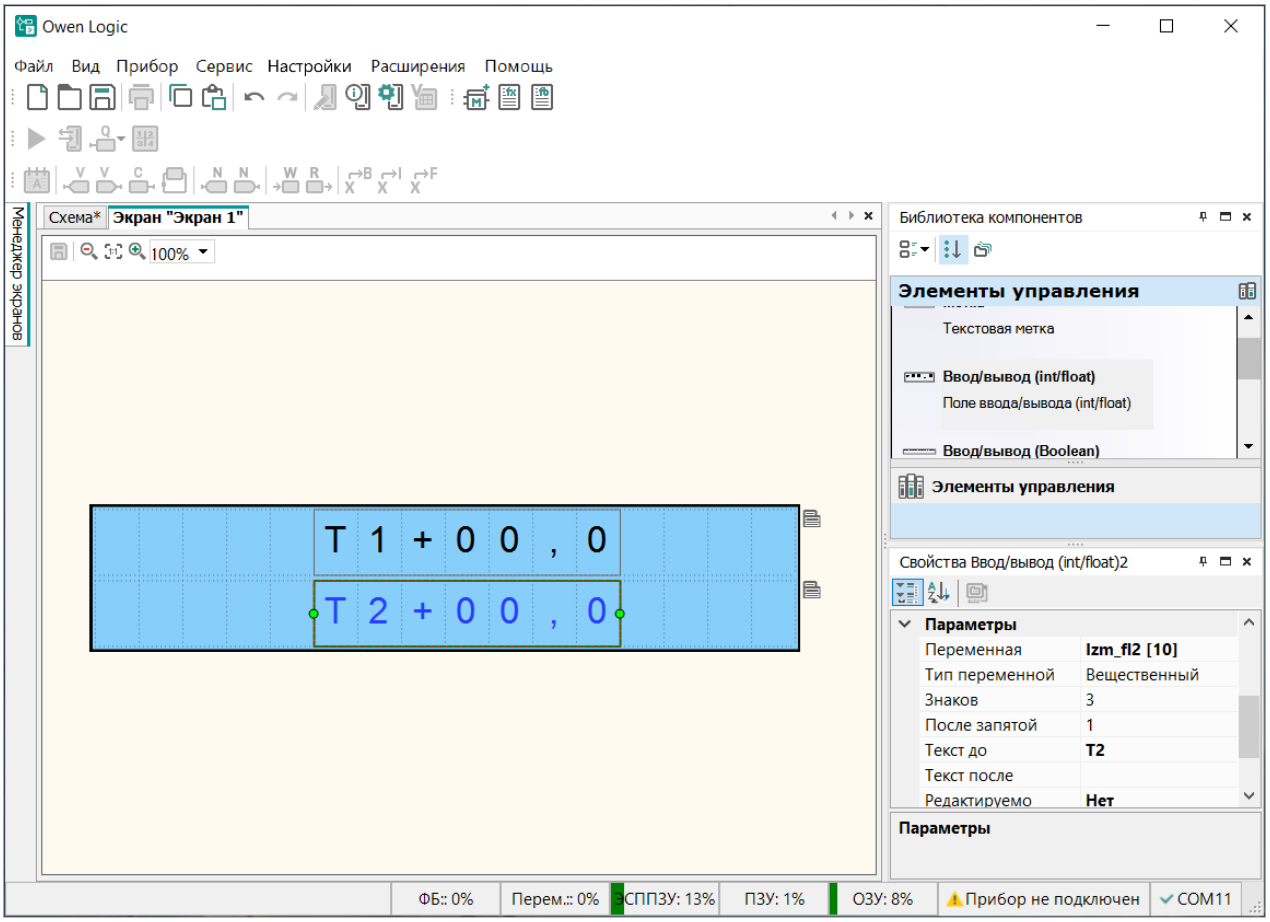


Рисунок 11.4

13. Вставить скопированный элемент поверх существующего, нажав ПКМ на экран и выбрав в контекстном меню **Вставить** (или воспользоваться сочетанием клавиш **Ctrl + V**).

14. На вставленном элементе изменить переменную на **lzm_fl2** и заменить текстовый блок на **T2**. Программа готова к загрузке в прибор.

11.2 Задание уставок таймеров с экрана

Задача – необходимо управлять задержками ФБ Blink (генератор прямоугольных импульсов) и TP (импульс включения заданной длительности) с дисплея прибора. Предусмотреть возможность преобразования задержек в миллисекунды, секунды или часы. Дополнительно необходимо отображать заданную уставку ФБ на дисплее прибора.

Для реализации задачи был выбран ПР200.

Холст проекта с элементами приведен на иллюстрации ниже.

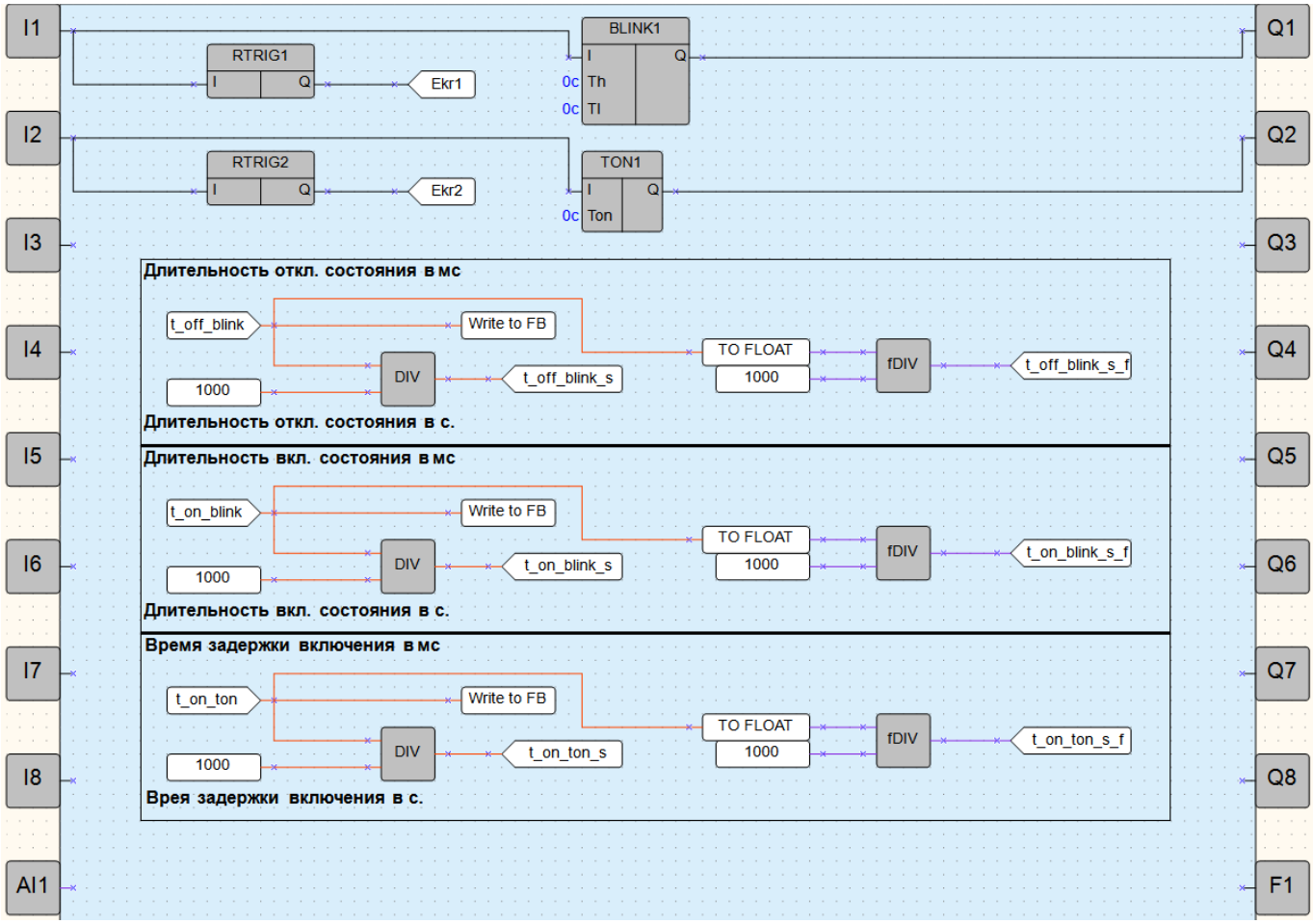


Рисунок 11.5 – Схема алгоритма

ФБ **Blink** (генератор прямоугольных импульсов) и **TP** (импульс включения заданной длительности) принимают в качестве уставок числа в целочисленном формате с дискретностью 1 мс.

Чтобы пересчитать уставки под нужные единицы времени следует:

- для отображения значений в секундах, следует разделить введенное значение на 1000;
- для отображения уставок в минутах следует разделить введенное значение на (60 · 1000);
- для отображения уставок в часах. следует разделить введенное значение на (60 · 60 · 1000).

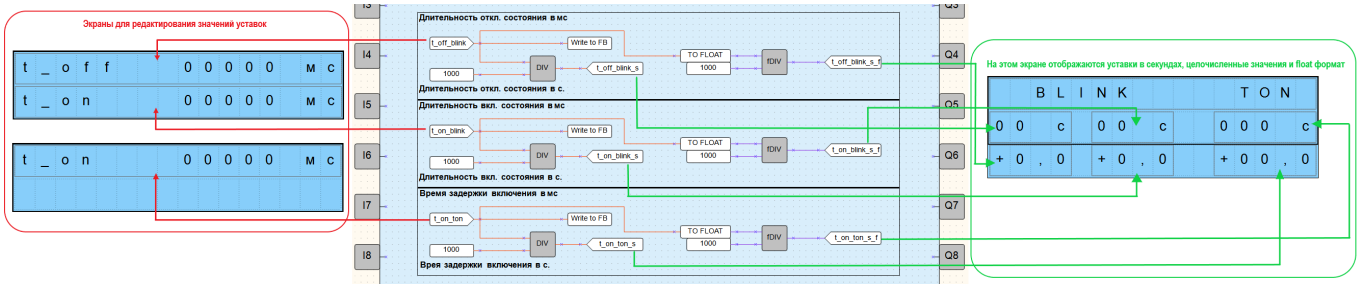


Рисунок 11.6

Если необходимо выводить значение уставки с десятичными знаками, следует перевести число в тип с плавающей запятой и поделить на соответствующий коэффициент.

Пример

Для перевода 2500 миллисекунд в секунды следует: 2500 мс ÷ 1000 = 2,5 с.

11.3 Подключение ПР200 к OwenCloud через шлюз ПМ210 по протоколу Modbus RTU

ПРИМЕЧАНИЕ
Подключение приборов первого поколения к OwenCloud доступно только через шлюзы линейки Пх210 и будет рассматриваться на примере ПР200 и шлюза ПМ210. Для подключения других приборов первого поколения выполните аналогичные действия.

Будем считать, что ПМ210 уже подключен к ПР200 и настроен для работы.

Для подключения следует:

1. Создать проект для ПР200 в Owen Logic.
2. В окне настроек прибора задать параметры, как на иллюстрации ниже.

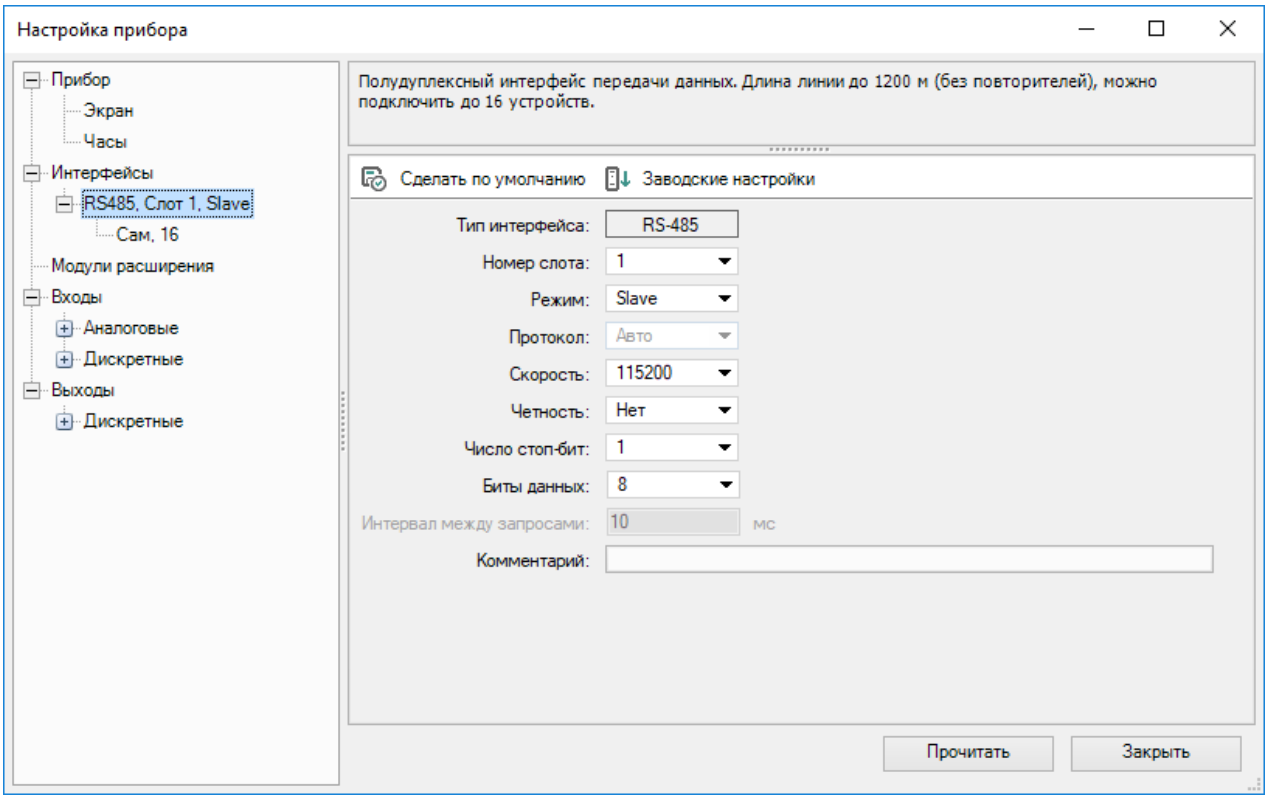


Рисунок 11.7

3. В настройках слота указать адрес **1** и добавить сетевые переменные, как на иллюстрации ниже.

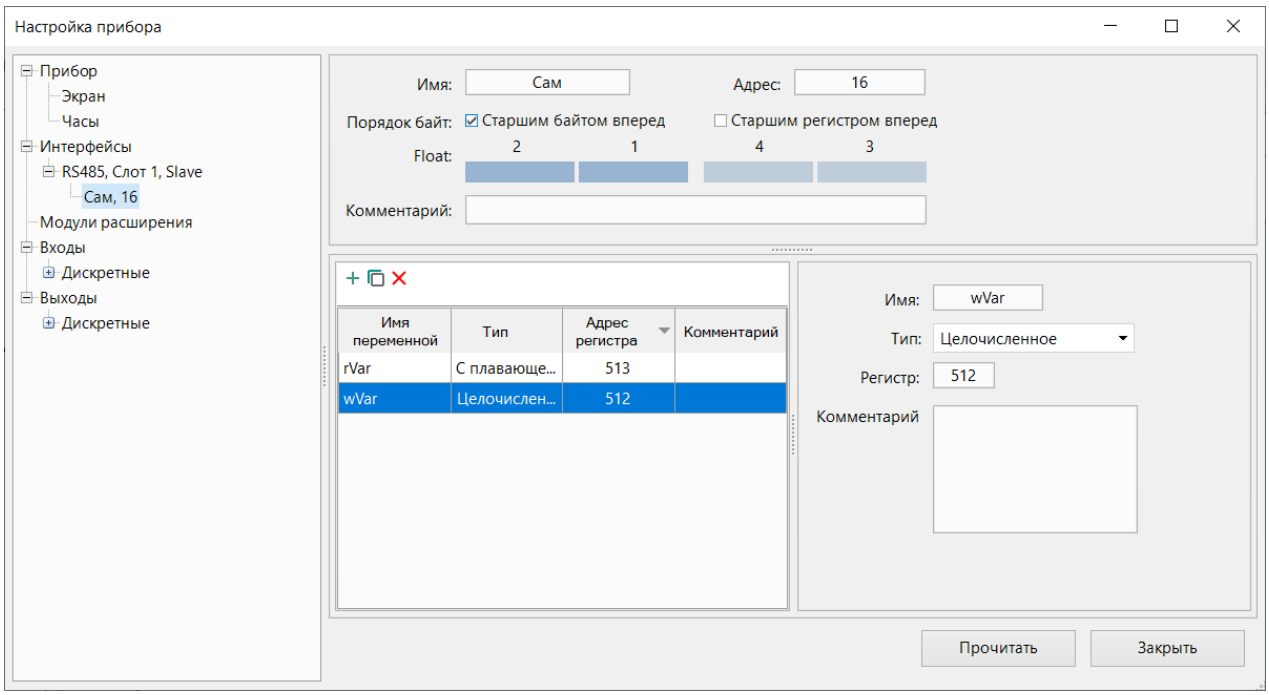


Рисунок 11.8

4. ***Следует обратить внимание***, что переменная с плавающей запятой (**rVar**) занимает два регистра в памяти прибора (в данном случае – 513 и 514).

5. Создать экран в менеджере экранов 3.8 и добавить на него элементы **Ввод-выход Int** и **Ввод-вывод Float**. Привязать к ним переменные **wVar** и **rVar**. В настройках элементов для параметра **Редактируемо** следует поставить значение **Да**, чтобы иметь возможность изменять их с дисплея ПР200.

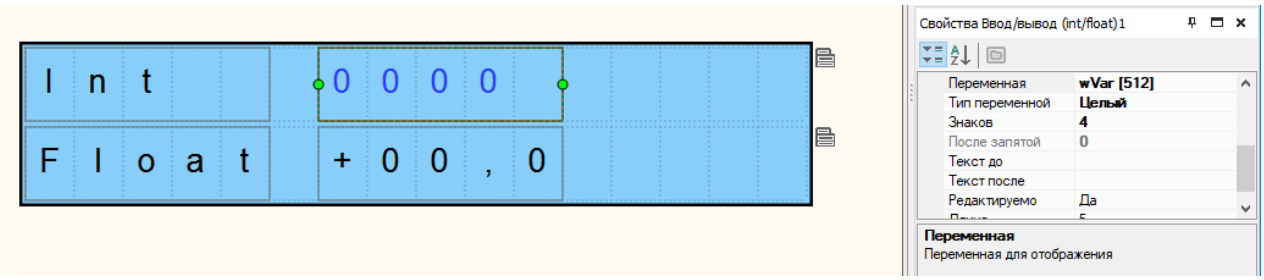
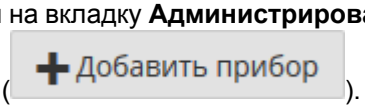


Рисунок 11.9

6. Загрузить проект в ПР200 (**Прибор** → **Загрузить программу в прибор**).

7. Зайти на главную страницу сервиса **OwenCloud**. Если еще не зарегистрированы в сервисе – необходимо пройти процедуру регистрации.

8. Перейти на вкладку **Администрирование**, открыть вкладку **Приборы** и нажать кнопку **Добавить прибор** (

).

9. Указать следующие обязательные настройки:
   - **Тип прибора** – выбрать тип **Произвольное устройство Modbus**;
   - **Идентификатор** – ввести **заводской номер** сетевого шлюза (указан на корпусе шлюза);
   - **Адрес в сети** – указать адрес **1**;
   - **Название прибора** – ввести название прибора;
   - **Часовой пояс** – указать часовой пояс, в котором находится прибор.

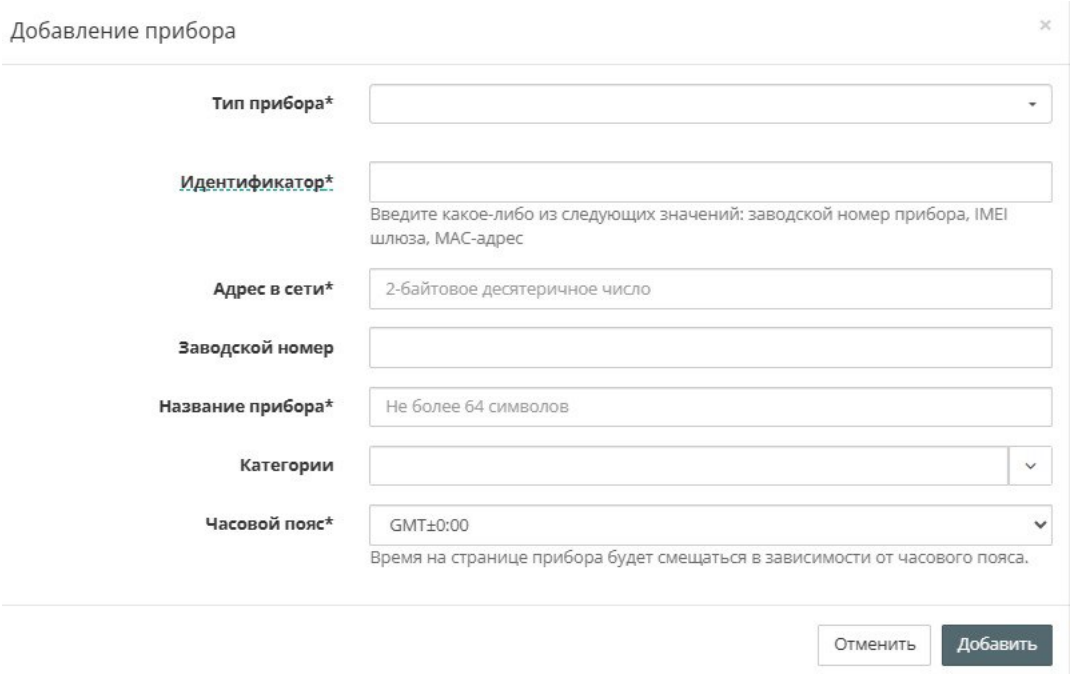


Рисунок 11.10

10. Нажать кнопку **Добавить**.
11. Нажать на название прибора, чтобы перейти к его настройке:

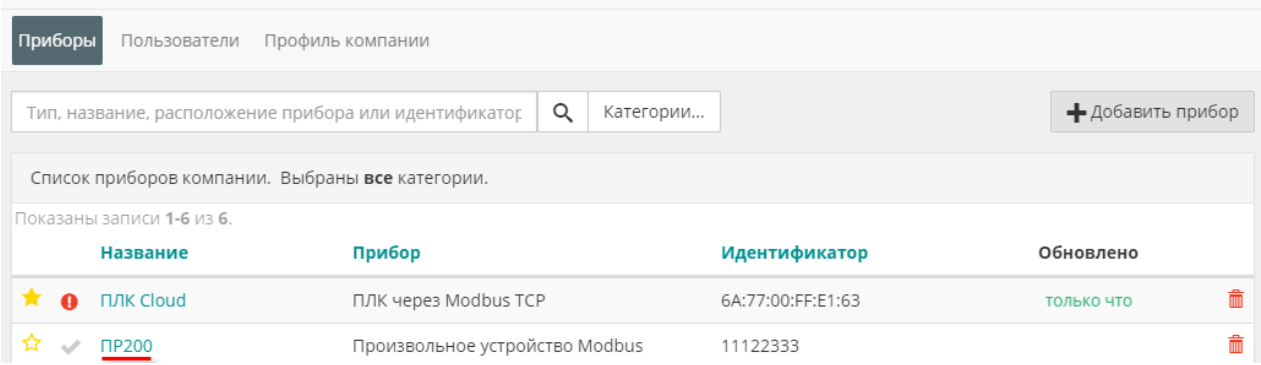


Рисунок 11.11

12. На вкладке **Общее/Общие настройки** указать скорость опроса и настройки COM-порта прибора в соответствии с настройками в Owen Logic. Нажать кнопку **Сохранить** для применения новых настроек. В случае необходимости можно изменить и другие настройки (например, период опроса).

Общие | События | Параметры

Общие настройки | Настройки расположения на карте

Текущий идентификатор	11122333
Тип прибора	Произвольное устройство Modbus
Новый идентификатор	GSM-шлюз => IMEI, ПЛК => MAC-адрес
Заводской номер	Целое, не более 17 знаков
Название прибора*	ПР200
Часовой пояс*	GMT+3:00 Время на странице прибора будет смещаться в зависимости от часового пояса.
Время хранения архива*	90 дней
"Оперативный" период опроса*	15 сек Интервал опроса оперативных параметров
"Конфигурационный" период опроса*	15 сек Интервал опроса конфигурационных параметров
"Управляющий" период опроса*	15 сек Интервал опроса управляемых параметров
Скорость COM-порта*	115200
	Аппаратное RTS/CTS согласование Использовать аппаратное RTS/CTS согласование при обмене через RS-232.
Настройка COM-порта*	8N1
Адрес в сети*	1
Таймаут между символами*	100 мс
Таймаут всего сообщения*	100 мс
Протокол Modbus*	RTU
	Разрешать пакетное чтение Система будет группировать запросы к соседним Modbus-регистрам

Сохранить

Рисунок 11.12

13. На вкладке **Параметры/Настройки параметров Modbus** добавить параметры вручную. **Следует обратить внимание**, что адреса регистров указываются в шестнадцатеричной системе – поэтому используемые значения отличаются от тех, которые приведены в Owen Logic – 512 (DEC) = 200 (HEX).

Управление прибором: ПР200

Общие | События | Параметры

Настройка параметров Modbus | Общая настройка параметров

Добавить | Удалить все параметры | Импортировать...

Всего 2 записи.

Код параметра	Функция чтения	Функция записи	Адрес регистра	Формат хранения	Порядок хранения байт	Отображаемая единица измерения	Название
rVar	03	16	201	float	Младший байт сзади	none: 3 цифр после точки, без единиц	FLOAT
wVar	03	06	200	uint16	Младший байт сзади	none: 3 цифр после точки, без единиц	INT

Экспорт в JSON

Рисунок 11.13

14. Перейти к просмотру значений параметров прибора. Изменить значения переменных с дисплея ПР200 и наблюдать соответствующие изменения в **OwenCloud**.
15. В случае необходимости изменения значений следует перейти на вкладку **Запись параметров**.

ПР200-24.4.2.0

обновлено
только что

Параметры | Таблицы | Графики | Лог активных событий | Запись параметров

Код параметра	Параметр	Значение
rVar	rVar	5.500
wVar	wVar	9.000

Рисунок 11.14

11.4 Прямое подключение приборов второго поколения к OwenCloud

Для приборов второго поколения доступно прямое подключение к OwenCloud.

Прибор должен быть настроен для работы и подключен к сети.

Для подключения к OwenCloud следует:

1. Подключить прибор к ПК и создать проект с использованием сетевых переменных.
2. Установить пароль для доступа к прибору.
3. В окне настроек прибора разрешить доступ к OwenCloud.

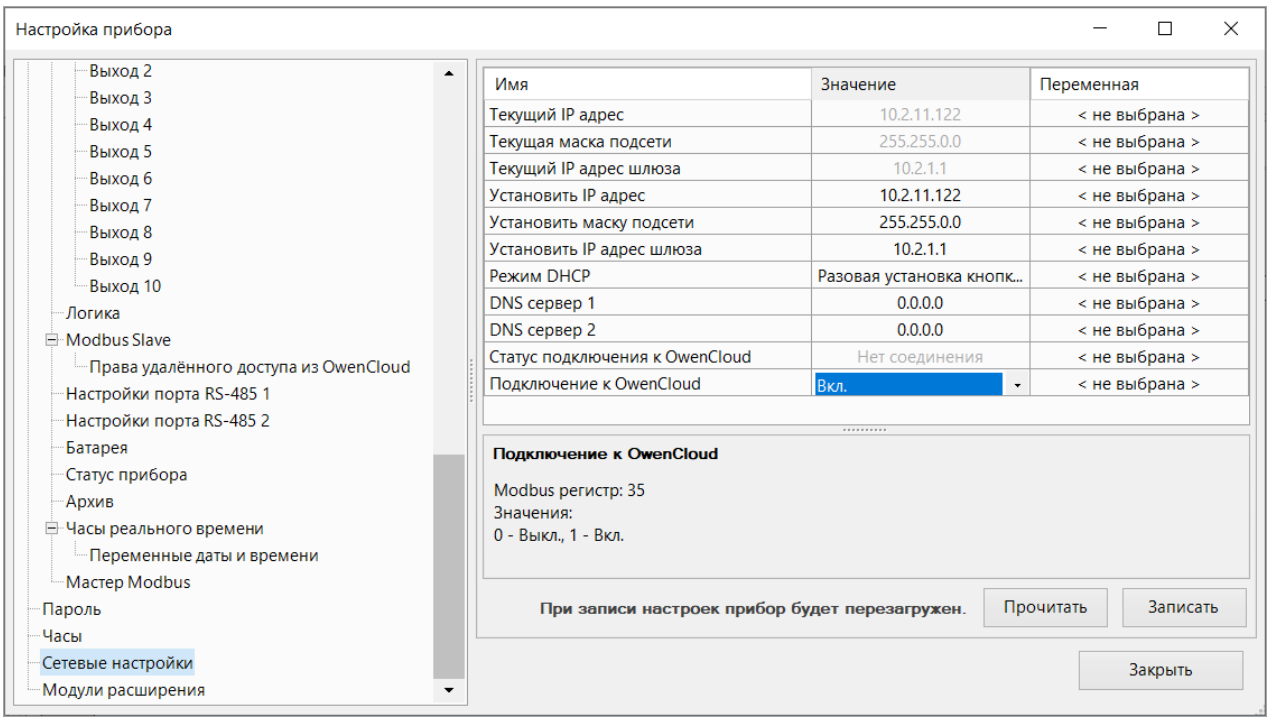


Рисунок 11.15 – Окно Настройка прибора

ПРИМЕЧАНИЕ
Значение параметра **Статус подключения к OwenCloud** можно считать по Modbus или привязать переменную пользовательской программы. Описание параметра приведено в таблице ниже.

Таблица 11.1 – Возможные состояния параметра «Статус подключения к OwenCloud»

Состояние	Значение	Описание
Нет соединения	0	Выключен обмен с OwenCloud
Идентификация	1	Устанавливается подключение к OwenCloud
Работа	2	OwenCloud управляет прибором, ошибок нет
Ошибка сети	3	Прибор не может установить соединение с сервером OwenCloud
Нет пароля	4	Не установлен пароль на прибор

4. Разрешить удаленный доступ к регистрам Modbus.

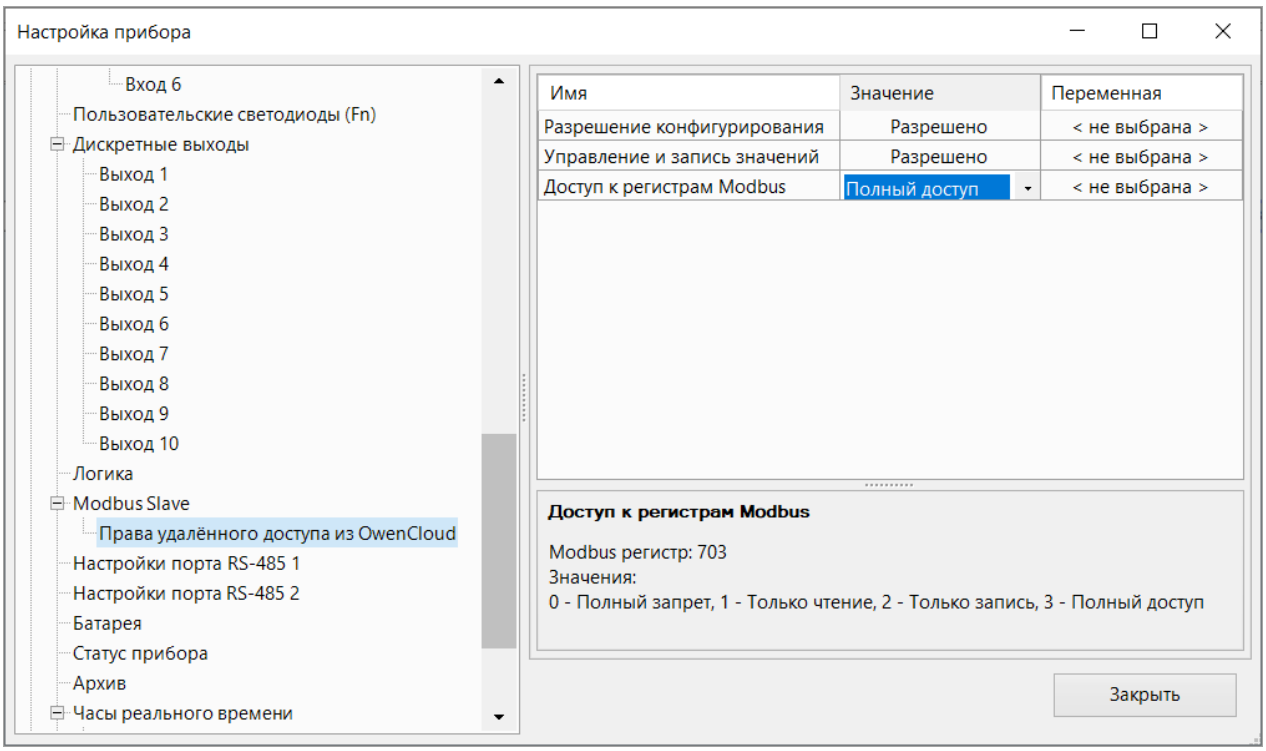
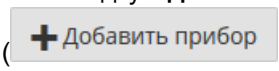


Рисунок 11.16 – Окно Настройка прибора

5. Загрузить программу в прибор(**Прибор** → **Загрузить программу в прибор**).
6. Зайти на главную страницу сервиса OwenCloud. Если регистрация отсутствует, то необходимо пройти процедуру регистрации.
7. Перейти на вкладку **Администрирование**, открыть вкладку **Приборы** и нажать кнопку **Добавить прибор** (

).
8. Появится окно с выбором типа прибора.

ПРИМЕЧАНИЕ
Прямое подключение к OwenCloud будет рассматриваться на примере ПР103. Для подключения других приборов второго поколения выполните аналогичные действия.

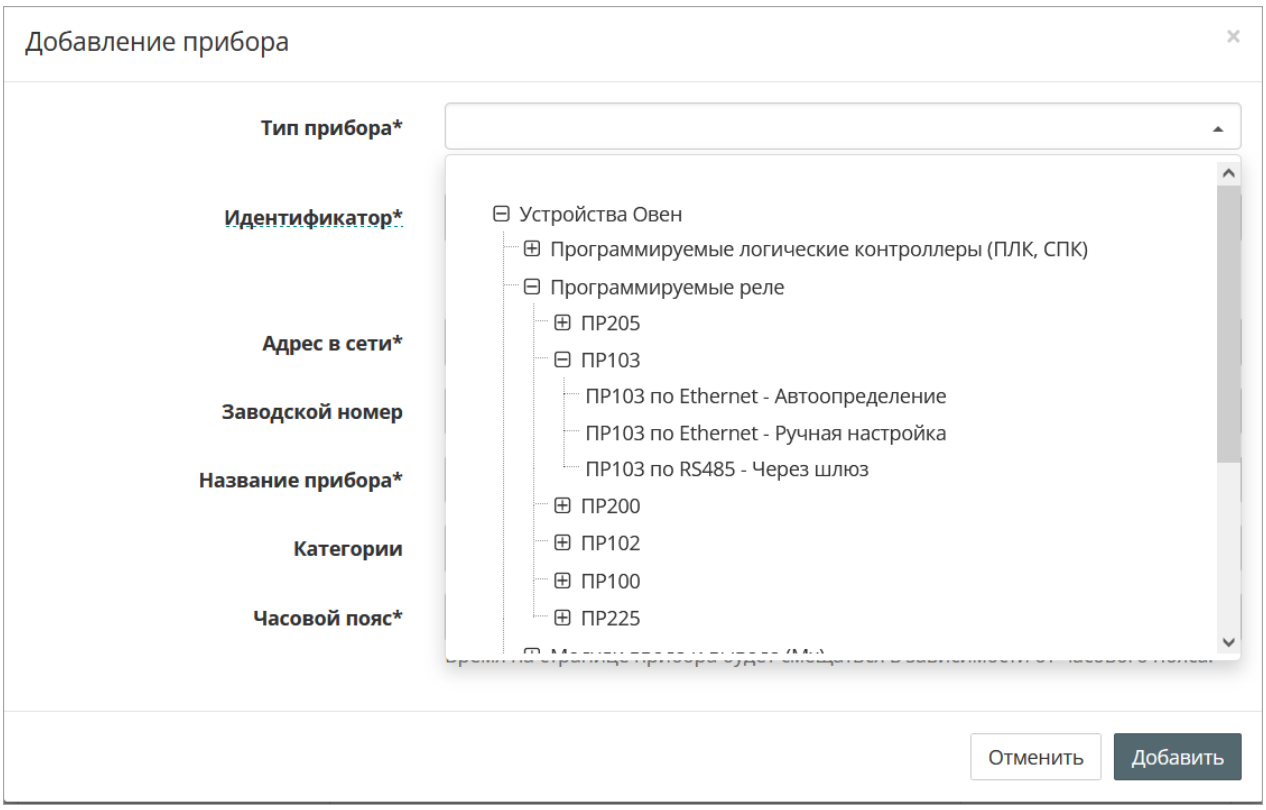


Рисунок 11.17 – Выбор типа прибора

Вариант **ПР103 по Ethernet — Автоопределение**:

ПРИМЕЧАНИЕ
При этом способе добавления прибора выводятся не только сетевые переменные, но и все дерево параметров прибора.

a. В появившемся окне в полях:
   - **Идентификатор** ввести заводской номер прибора;
   - **Адрес в сети** оставить без изменения адрес **1**;
   - **Название прибора** – ввести название прибора;
   - **Часовой пояс** – указать часовой пояс, в котором находится прибор.

b. Нажать кнопку «Добавить». Откроется интерфейс базовых настроек прибора.

c. Ввести пароль от прибора. Если необходимо, можно изменить и другие настройки (например, период опроса).

d. Нажать кнопку «Сохранить» для применения новых настроек.

e. OwenCloud подключится к прибору и считает все параметры из него.

Вариант **ПР103 по Ethernet — Ручная настройка**:

a. В появившемся окне в полях:
   - **Идентификатор** ввести заводской номер прибора;
   - **Адрес в сети** оставить без изменения адрес **1**;
   - **Название прибора** – ввести название прибора;
   - **Часовой пояс** – указать часовой пояс, в котором находится прибор.

b. Нажать кнопку «Добавить». Откроется интерфейс общих настроек прибора.

c. Ввести пароль от прибора. Если необходимо, можно изменить и другие настройки. Нажать кнопку «Сохранить» для применения новых настроек.

d. Далее во вкладке **Параметры**/**Настройки параметров** добавить сетевые параметры прибора одним из доступных способов:
   - вручную, в соответствии с заданными в ПР103 сетевыми переменными. **Следует обратить внимание**, что адреса регистров указываются в шестнадцатеричной системе — поэтому используемые значения отличаются от тех, которые приведены в Owen Logic - 16384 (DEC) = 4000 (HEX).

- в виде файла *.json, если воспользоваться расширением **Экспорт устройства в OwenCloud** Owen Logic. Для добавления параметров следует нажать на выпадающий список **Импортировать** и выбрать вариант **Загрузить из JSON**. В открывшемся меню выбрать ранее созданный файл в формате *.json и нажать кнопку **Загрузить параметры**.

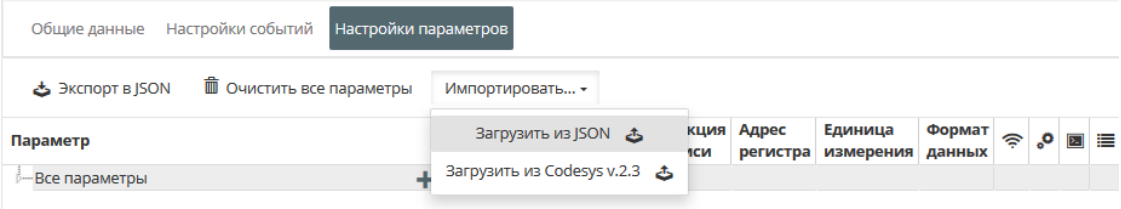


Рисунок 11.18 – Импорт параметров

Вариант **ПР103 по RS485 - Через шлюз**:

a. В появившемся окне в полях:
   - **Идентификатор** ввести заводской номер шлюза;
   - **Адрес в сети** установить в соответствии с адресом порта к которому подключен шлюз;
   - **Название прибора** – ввести название прибора;
   - **Часовой пояс** – указать часовой пояс, в котором находится прибор.

b. Нажать кнопку «Добавить». Появится окно настройки обмена с OwenCloud.

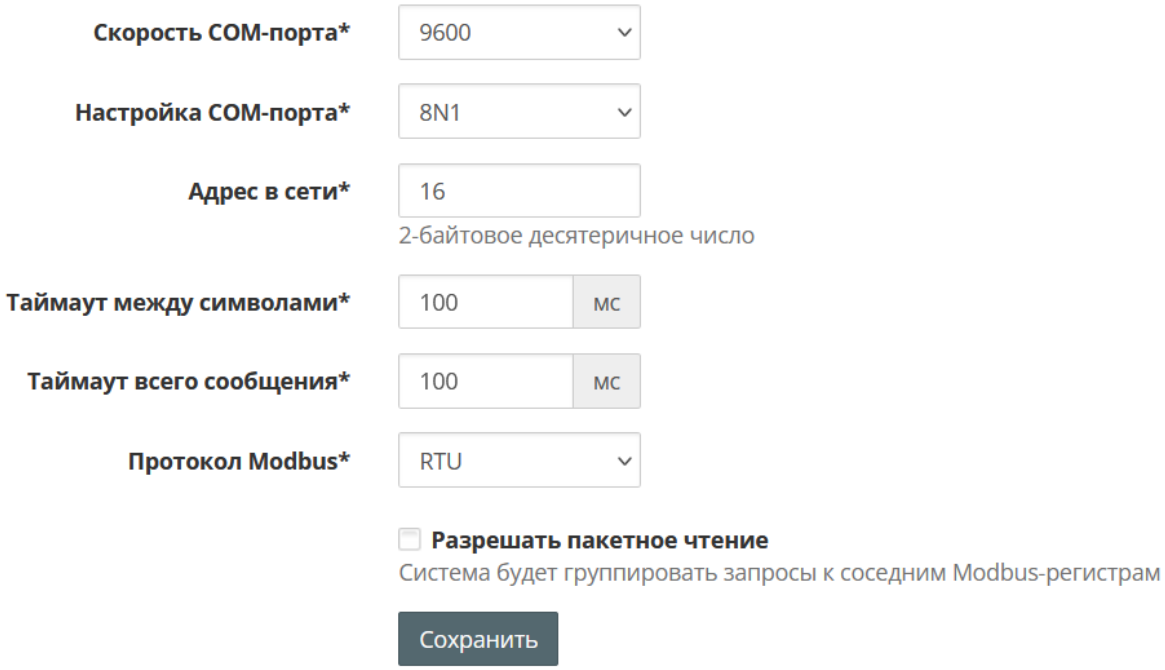


Рисунок 11.19 – Настройки обмена с OwenCloud

Поля должны соответствовать аналогичным настройкам порта к которому подключен шлюз.:

- **Скорость COM-порта**
- **Настройка COM-порта** установить в соответствии с адресом порта к которому подключен шлюз;
- **Протокол Modbus** – ввести название прибора;

c. Нажать кнопку «Сохранить» для применения новых настроек.

d. Далее во вкладке **Параметры**/**Настройки параметров** добавить сетевые параметры прибора одним из доступных способов:
   - вручную, в соответствии с заданными в ПР103 сетевыми переменными. **Следует обратить внимание**, что адреса регистров указываются в шестнадцатеричной системе — поэтому используемые значения отличаются от тех, которые приведены в Owen Logic - 16384 (DEC) = 4000 (HEX).
   - в виде файла *.json, если воспользоваться расширением **Экспорт устройства в OwenCloud** Owen Logic. Для добавления параметров следует нажать на выпадающий

список **Импортировать** и выбрать вариант **Загрузить из JSON**. В открывшемся меню выбрать ранее созданный файл в формате *.json и нажать кнопку **Загрузить параметры**.

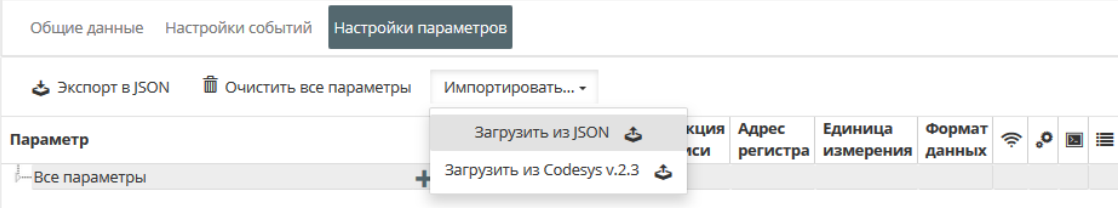


Рисунок 11.20 – Импорт параметров

Если все настройки были корректными, то на вкладке **Общие данные** отобразятся данные от прибора.

Если прибор не опрашивается из OwenCloud, то следует проверить сетевые настройки, статус подключения к OwenCloud:

Рисунок 11.21 – Пример отображения подключения к OwenCloud

Если значения не пишутся в прибор из OwenCloud следует проверить наличие установленной галочки в столбце **Управляемый параметр**.

Рисунок 11.22 – Управляемый параметр

11.5 Упаковка/распаковка бит/целых чисел

Задача – необходимо считывать состояние входов прибора и выводить на экран состояние выбранного входа.

На иллюстрации ниже приведен холст проекта с использованием макросов из менеджера компонентов 4.10 и настройки свойств элементов вывода на экран.

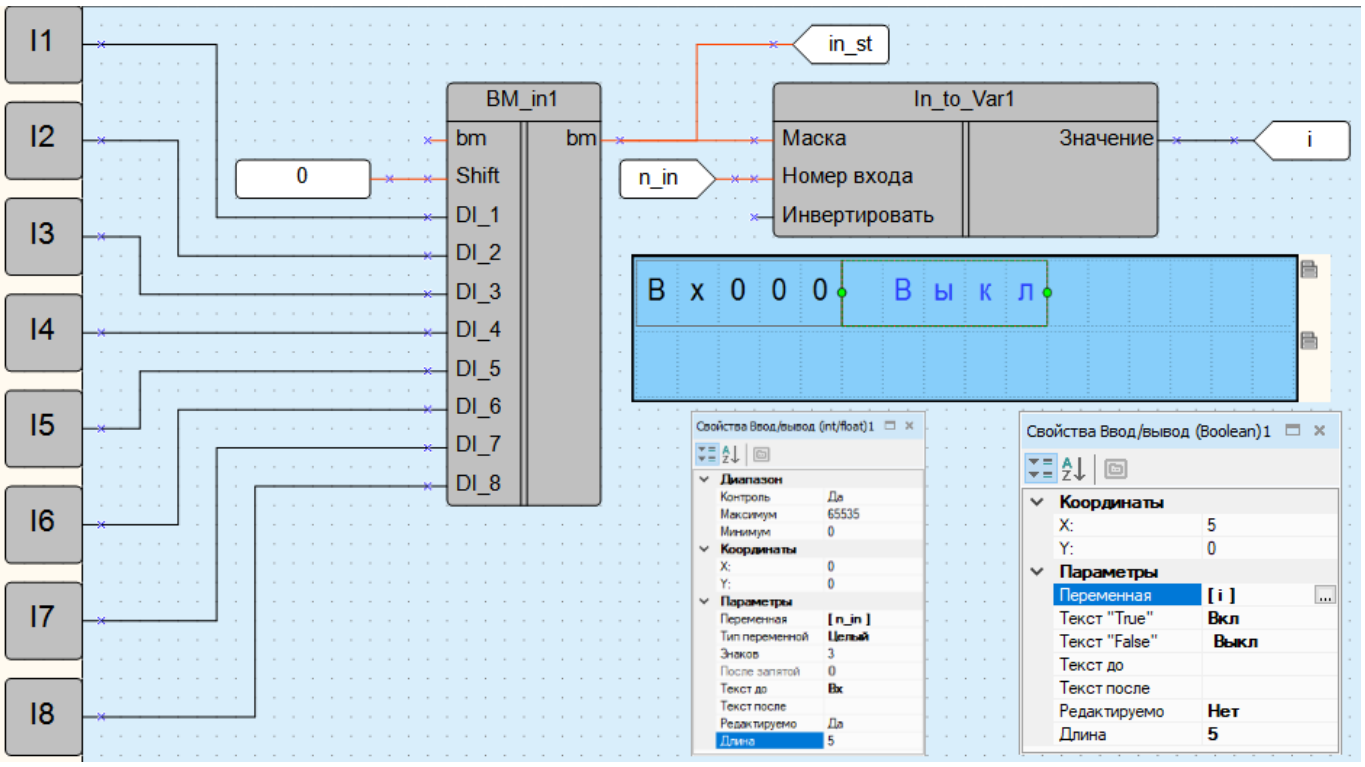


Рисунок 11.23

11.6 Обработка битовых переменных с помощью маски

Задача – по сигналу выходы включаются или выключаются. Если реализуется дополнительное условие — срабатывает вход 1, то выходы 2, 4, 6 и 8 должны принимать инвертированное состояние относительно текущего.

Для задания состояний выходов следует использовать макросы из менеджера компонентов 4.10 — **BOOL_INT** и **INT_BOOL**.

Дополнительное условие данного примера можно реализовать с помощью побитного исключения XOR. В качестве маски можно воспользоваться целочисленной константой — **170** (10101010 в двоичном виде).

Холст проекта в режиме симуляции приведен ниже.

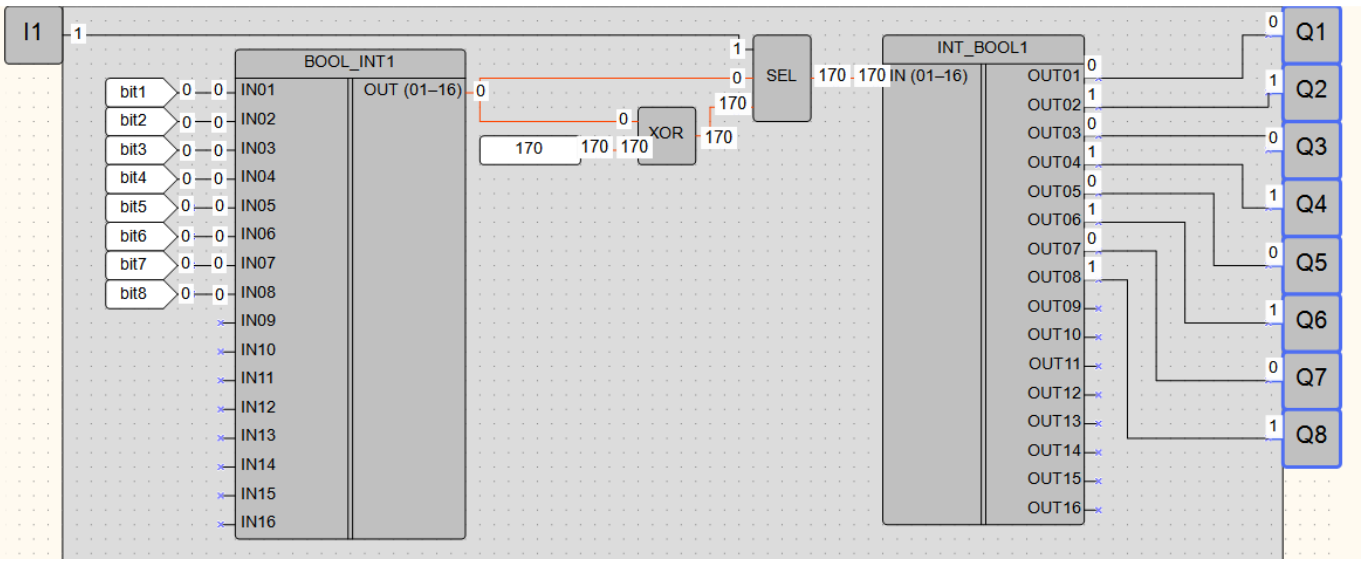


Рисунок 11.24

12 Язык программирования ST

ST (Structured Text) - это текстовый язык высокого уровня. Является одним из пяти языков, поддерживаемых стандартом МЭК 61131-3.

В Owen Logic доступно создание функций и функциональных блоков на языке ST.

- Синтаксис 12.1
- Типы данных 12.3
- Конструкции языка 12.4
- Системные функции 12.5
- Системные функциональные блоки 12.6

12.1 Синтаксис

Ключевые слова могут быть введены в символах верхнего и нижнего регистра. Пробелы и табуляция не влияют на синтаксис и могут быть использованы везде.

Имена переменных, функций и функциональных блоков подчиняются следующим правилам:

- Имя не должно содержать пробелов и спецсимволов (например, !, @ и т.д.). Исключение – символ нижнего подчеркивания (_);
- Имя должно начинаться с буквы;
- Имя переменной может содержать только буквы латинского алфавита;
- Имя не должно содержать несколько символов нижнего подчеркивания(_), размещенных подряд (т.е. имя **i__Test** недопустимо, а имя **i_Te_st** - допустимо);
- Регистр имен объектов не учитывается (**ITest** и **iTEST** будет интерпретироваться как одно и то же имя);
- На длину имени не накладывается никаких ограничений;
- Имя не должно совпадать ни с одним из зарезервированных ключевых слов (например, VAR, INT, и т. д.);
- Рекомендуется для имен переменных использовать венгерскую нотацию и стиль lowerCamelCase.

12.1.1 Побитовый доступ к переменным

Побитовый доступ доступен для переменных типа UDINT. Для обращения нужно использовать следующую форму:

```
Переменная.Номер требуемого бита
```

Пример

```
Variable.1 := TRUE;
Variable.31 := FALSE;
```

Номер бита при этом должен быть задан целым числом от 0 до 31. Использование переменных в качестве номера бита не допускается.

12.1.2 Использование функций в других элементах программы на ST

Функцию можно вызвать в другой функции или функциональном блоке. Для этого нужно использовать следующий формат (неформальный вызов):

Имя функции (перечисление через запятую входов функции)

Пример

```
FUNCTION rFun1: REAL;

    VAR_INPUT
        rIn1 : REAL;
        rIn2 : REAL;
    END_VAR

    rFun1 := rIn1 + rIn2;

END_FUNCTION
```

Есть функция **rFun1**:

Ее вызов в функции **rFun2** будет выглядеть следующим образом:

```
FUNCTION rFun2: REAL;
```

```
    VAR_INPUT
        rIn1_0 : REAL;
        rInf1_1 : REAL;
        rInf1_2 : REAL;
    END_VAR

    rFun2 := rIn1_0 * rFun1(rInf1_1, rInf1_2);

END_FUNCTION
```

Функция **rFun2** будет возвращать число, равное произведению **rIn1_0** на сумму **rInf1_1** и **rInf1_2**, которые задаются на входе этой функции.

12.1.3 Использование одного функционального блока в другом

Экземпляры одного функционального блока могут быть созданы в другом функциональном блоке, а именно в области объявления локальных переменных в формате:

(краткое обозначение):(имя функционального блока)

После создания экземпляра функционального блока можно начинать работу с его данными. Входы экземпляра блока доступны для записи извне. Выходы – для чтения.

Вызвать экземпляр можно по-разному. В качестве примера рассмотрим вызов счетчика для прямого счета, который создается как шаблон при создании функционального блока.

1. Через формальный вызов:

```
FUNCTION_BLOCK fb2

    VAR_INPUT
        xIn : BOOL;
    END_VAR

    VAR_OUTPUT
        xAlarmMax : BOOL;
        udiQ : UDINT;
    END_VAR

    VAR
        fb1 : functionblock1; //объявление экземпляра функционального блока
    END_VAR

    fb1 (U := xIn, Q => udiQ); //вызов экземпляра функционального блока
    IF fb1.Q > 10 THEN
        xAlarmMax := TRUE;
    END_IF

END_FUNCTION_BLOCK
```

ПРИМЕЧАНИЕ
Обратите внимание на специфический оператор копирования значения из выходных переменных блока («=>»).

2. Через обращение к входам/выходам функционального блока (объявление переменных аналогичное):

```
    fb1.U := xIn; //запись данных на вход ФБ
    fb1 (); //вызов экземпляра функционального блока
    udiQ := fb1.Q; //чтение выхода экземпляра ФБ
    IF fb1.Q > 10 THEN
        xAlarmMax := TRUE;
    END_IF
```

На выходе **fb2** будет индикация превышения 10 импульсов и накопленное значение счетчика.

ПРЕДУПРЕЖДЕНИЕ
Объявлять экземпляр функционального блока в теле функции нельзя.

ПРЕДУПРЕЖДЕНИЕ
В функциональном блоке на ST максимальная вложенность блоков – не более 8.

ПРЕДУПРЕЖДЕНИЕ
Функциональные блоки на ST не поддерживают переменные типа RETAIN.

ПРЕДУПРЕЖДЕНИЕ
Функциональные блоки на ST резервируют место в памяти ПЗУ после их добавления в библиотеку проекта, независимо от того используются они в проекте или нет.

12.1.4 Комментирование в редакторе ST

Редактор ST в Owen Logic поддерживает функцию комментирования. Доступны два типа комментариев:

1. Однострочный. Его маркером является двойной слеш – //
2. Многострочный. Его маркером является:
   - (* – начало комментария;
   - *) – конец комментария.

12.1.5 Копирование элементов на ST между проектами

Для копирования компонентов на ST между проектами следует:

1. Выделить на схеме исходного проекта те элементы на ST, которые необходимо скопировать.
2. Скопировать выбранные элементы с помощью сочетания клавиш **Ctrl + C** или с помощью контекстного меню.

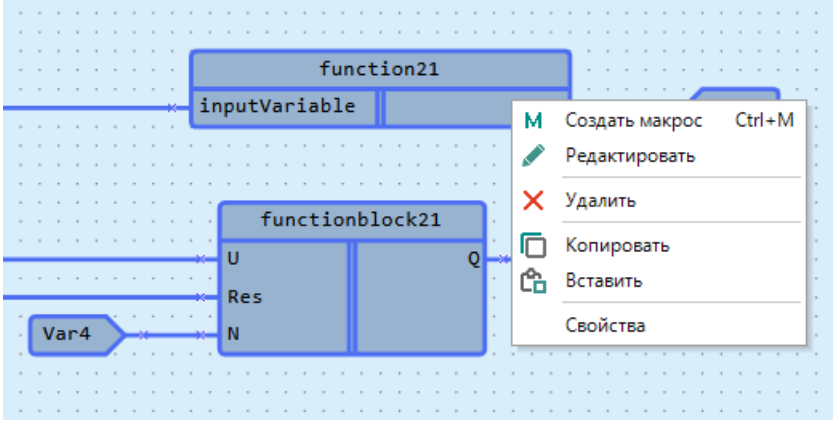


Рисунок 12.1

3. Открыть проект для вставки скопированных элементов.
4. Вставить элементы на схему второго проекта с помощью сочетания клавиш **Ctrl + V** или с помощью контекстного меню.

В результате в библиотеку компонентов проекта в соответствующий раздел добавятся шаблоны всех скопированных элементов.

ПРИМЕЧАНИЕ

1. Если в библиотеке компонентов нет группы, которая была указана для скопированного компонента, она будет создана.
2. Если группа скопированного компонента не указана, шаблон будет добавлен в папку **Другое**.

При копировании сохраняются все связи между всеми компонентами, которые вошли в операцию копирования.

ПРЕДУПРЕЖДЕНИЕ
Если какой-либо из скопированных элементов содержит ошибку, вставка элементов на схему проекта будет отменена.
Отобразится сообщение об ошибке с указанием имени некорректного элемента.

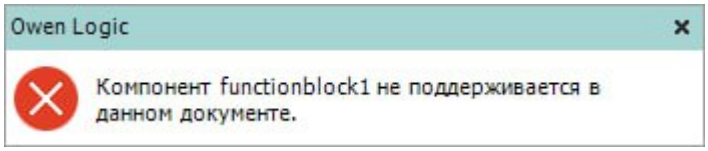


Рисунок 12.2

Прочие компоненты отобразятся в библиотеке компонентов в соответствующих разделах.

12.2 Документирование в редакторе ST

Редактор ST поддерживает функцию документирования.

Маркером документирования является тройной слэш «///».

Документирование добавляется над объектами: объявление функции / функционального блока, объявление входной / выходной переменной.

Теги для документирования перечислены в таблице ниже.

Тег	Описание
<Description>…</Description>	Описание элемента программы (функции, функционального блока, входной и выходной (только для функционального блока) переменной)
<Author>…</Author>	Имя создателя функции или функционального блока
<GroupName>…</GroupName>	Имя группы для группировки функции или функционального блока в библиотеке компонентов
<OutputDescription>… </OutputDescriotion>	Описание выхода функции
<Password>…</Password>	Пароль для доступа в блок

Информация из тегов документирования будет отображаться в сниппетах при вызове функции или функционального блока.

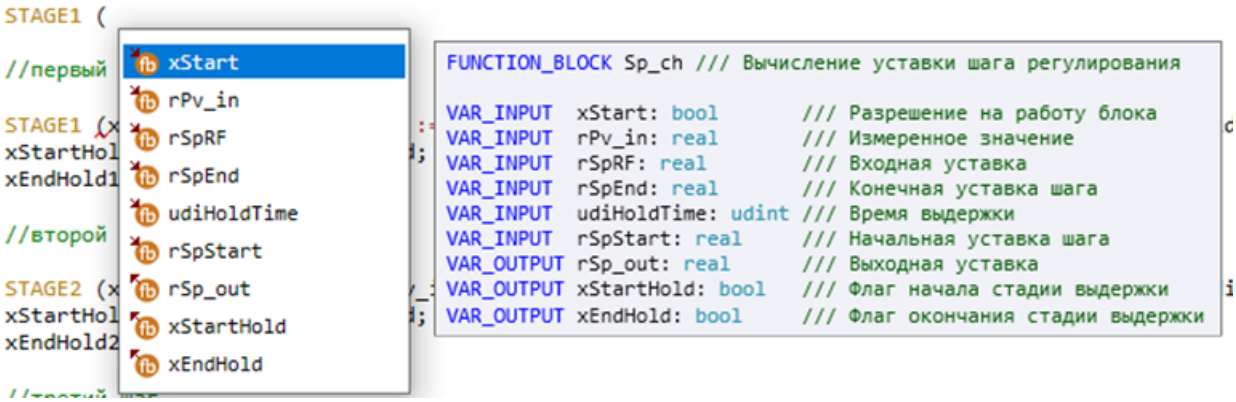


Рисунок 12.3

Документирование функции

```
///<Description>Преобразователь сопротивления в температуру (Pt1000)</Description>
///<OuptutDescription>Температура</OutputDescription>
///<Author>ОВЕН</Author>
///<GroupName>Температурный датчики</GroupName>

FUNCTION f_PT1000: REAL; // функция для датчика термосопротивления PT1000

    VAR_INPUT

        ///<Description>Сопротивление</Description>
        R : REAL;
```

```
    END_VAR
```

После импорта в менеджер компонентов описание функции будет представлено в соответствии с указанными тегами документировани:

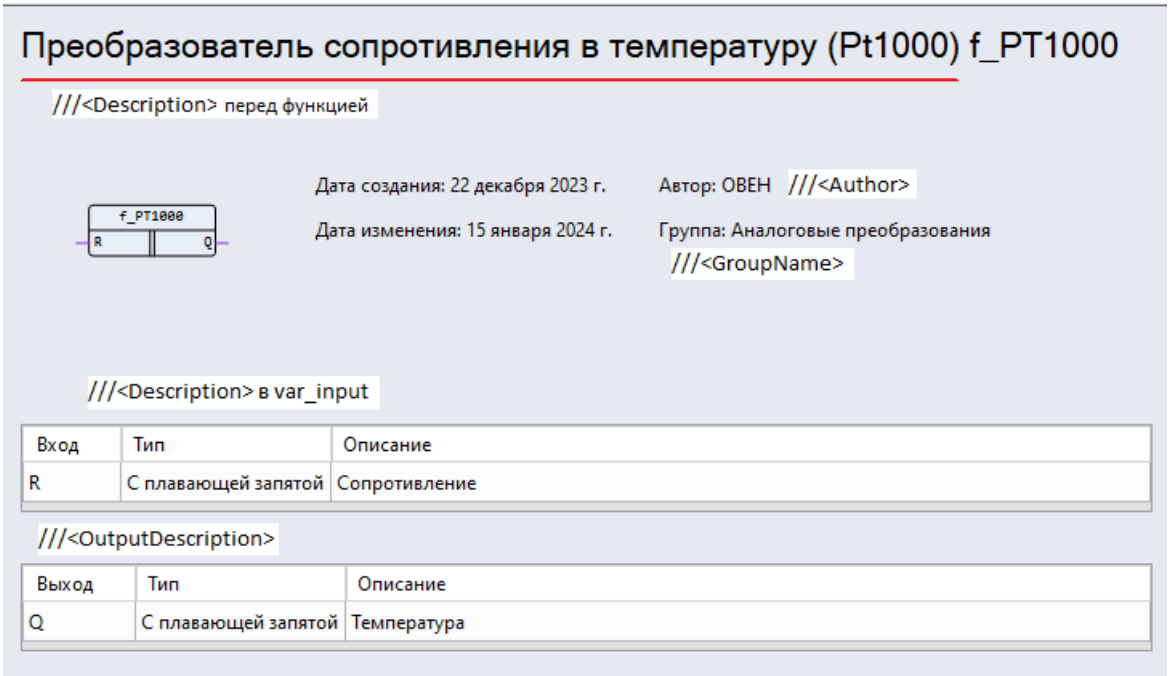


Рисунок 12.4

Пример документирования функционального блока

```
///<Description>Счетчик для прямого счета</Description>
///<Author>ОВЕН</Author>
///<GroupName>Таймеры и счетчики</GroupName>

FUNCTION_BLOCK fb_Counter

    VAR_INPUT

        ///<Description>Детектор импульсов</Description>
        U : BOOL; //входная переменная с типом данных bool

        ///<Description>Флаг сброса счетчика</Description>
        Res : BOOL; //входная переменная с типом данных bool

        ///<Description>Предустановленное значение счетчика</Description>
        N : UDINT; //входная переменная с типом данных bool

    END_VAR

    VAR_OUTPUT

        ///<Description>Значение счетчика</Description>
        Q : UDINT; //выходная переменная с типом данных udint

    END_VAR
```

После импорта в менеджер компонентов описание функционального блока будет представлено в соответствии с указанными тегами документирования:

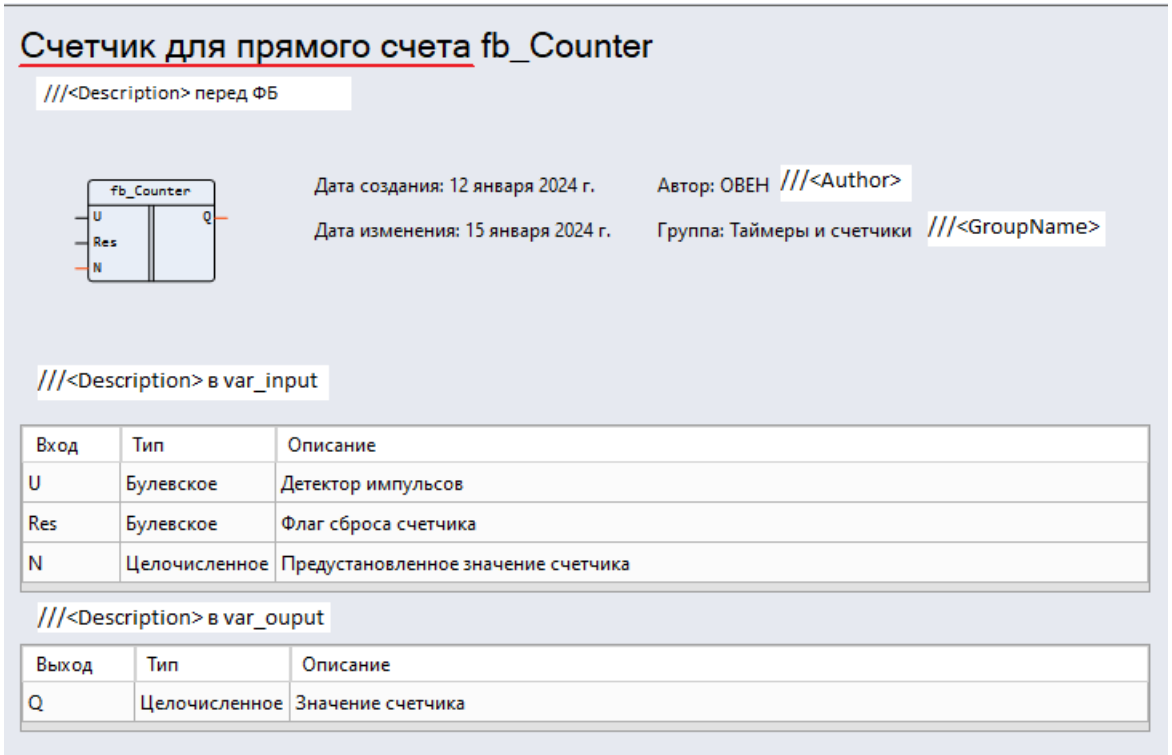


Рисунок 12.5

12.3 Типы данных

Тип данных переменной определяет род информации, диапазон представлений и множество допустимых операций.

ПРИМЕЧАНИЕ
Любую переменную можно использовать только после её объявления. Присваивать значения одной переменной другой можно, только если они одного типа. В противном случае используются преобразователи типов 12.4.1.3.

ПРЕДУПРЕЖДЕНИЕ
Преобразование больших типов в более мелкие может привести к потере информации.

Типы данных, поддержанные в Owen Logic, перечислены в таблице ниже:

Название	Тип	Возможные значения	Размер в байтах
BOOL	Логический	FALSE, TRUE	4 байта
UDINT	Целочисленный	0…4294967295	4 байта
REAL	Вещественный	$-1{,}2\times10^{-38}\ldots3{,}4\times10^{38}$	4 байта
TIME*	Интервал времени	T#0…4294967295ms T#0..4294967s T#0..71582m T#0..1193h T#0..49d T#0..49d17h02m47s295ms	4 байта
DT*	Время суток и дата	DT#2000-01-01-00:00:00..2136-02-07-6:28:15	4 байта
ARRAY	Массив	—	—

*- Переменные типов TIME и DT предназначены исключительно для использования внутри функций и функциональных блоков, и могут быть объявлены только как локальные переменные в блоке VAR ... END_VAR. Объявление этих переменных как входных/выходных приведет к отображению ошибки в списке ошибок, расположенном в нижней части редактора функции/функционального блока на ST.

Для передачи временных значений между блоками используйте тип данных UDINT, преобразованный с помощью функций TIME_TO_UDINT/DT_TO_UDINT.

12.3.1 Зарезервированные ключевые слова

ПРИМЕЧАНИЕ
В таблице полужирным шрифтом выделены доступные для использования слова.

ABS	END_TRANSITION	**RETURN**	**SYS.TOF**
ACTION	END_TYPE	SEL	**SYS.TON**
AND	**END_VAR**	**SHL**	**SYS.TP**
ARRAY	**END_WHILE**	**SHR**	TASK
AT	ENO	SINT	**THEN**
BEGIN	**EXIT**	STEP	**TIME**
BOOL	F_EDGE	STRING	TIME_OF_DAY
BOOL_TO_REAL	**FALSE**	STRUCT	**TIME_TO_UDINT**
BOOL_TO_UDINT	**FOR**	**SYS.ACOS**	**TO**
BY	**FROM**	**SYS.ASIN**	TOD
CASE	**FUNCTION**	**SYS.ATAN**	TRANSITION
CD32	**FUNCTION_BLOCK**	**SYS.BLINK**	**TRUE**
CONFIGURATION	**GET_DATE_TIME**	**SYS.CLOCK**	TYPE
CONTINUE	**GET_TIME**	**SYS.CLOCKWEEK**	**UDINT**
DATE	**IF**	**SYS.COMPARE_DATE_TIME**	**UDINT_TO_BOOL**
DATE_AND_TIME	INITIAL_STEP	**SYS.COS**	**UDINT_TO_DT**
DC32	INT	**SYS.CT**	**UDINT_TO_REAL**
DINT	LINT	**SYS.CTN**	**UDINT_TO_TIME**
DO	LREAL	**SYS.CTU**	UINT
DT	LWORD	**SYS.DTRIG**	ULINT
DT_TO_UDINT	**MOD**	**SYS.EXP**	**UNTIL**
DWORD	NON_RETAIN	**SYS.FACT**	USINT
ELSE	**NOT**	**SYS.FMAX**	**VAR**
ELSIF	**OF**	**SYS.FMIN**	VAR_ACCESS
EN	ON	**SYS.FTRIG**	VAR_CONFIG
END	**OR**	**SYS.IS_LEAP_YEAR**	VAR_EXTERNAL
END_ACTION	**POW**	**SYS.LG**	VAR_GLOBAL
END_CASE	PROGRAM	**SYS.LN**	VAR_IN_OUT
END_CONFIGURATION	R_EDGE	**SYS.LOG**	**VAR_INPUT**
END_FOR	READ_ONLY	**SYS.MAX**	**VAR_OUTPUT**
END_FUNCTION	READ_WRITE	**SYS.MIN**	VAR_TEMP
END_FUNCTION_BLOCK	**REAL**	**SYS.RS**	**WHILE**
END_IF	**REAL_TO_BOOL**	**SYS.RTRIG**	WITH
END_REPEAT	**REAL_TO_UDINT**	**SYS.SIN**	WORD
END_RESOURCE	**REPEAT**	**SYS.SQRT**	WSTRING
END_STEP	RESOURCE	**SYS.SR**	**XOR**
END_STRUCT	RETAIN	**SYS.TAN**	

12.3.2 Массивы

Синтаксис объявления массива

Для объявления массива применяется следующий синтаксис:

```
<Имя массива>:ARRAY[Индекс1..ИндексN]OF <ТИП ЭЛЕМЕНТА>:=[Значение по умолчанию элемента1, Значение по умолчанию элемента2, Значение по умолчанию элементаN];
```

Где:

Индекс1 - номер первого элемента массива;

ИндексN - номер последнего элемента массива.

Правило для индексов:

- Значение ИндексN не может быть меньше или равно значению Индекс1;
- Индексы должны быть только целого типа и не могут принимать отрицательные значения.

Примеры объявления массива:

```
        Mass1: ARRAY[0..2] OF UDINT; //объявление массива состоящего из трех элементов типа UDINT
        Mass1: ARRAY[0..2] OF UDINT :=[1, 3, 5]; //инициализация элементов массива при объявлении массива
```

Ошибки при объявлении массива:

```
        Mass1: ARRAY[0] OF UDINT :=[1, 5]; //ошибка: размерность массива должна быть указана в виде <целое число>...<целое число>
        Mass1: ARRAY[] OF UDINT; //ошибка: размерность массива должна быть указана в виде <целое число>...<целое число>
```

Синтаксис для доступа к элементам массива

Для доступа к элементам массива применяется следующий синтаксис:

```
<Имя массива>[Индекс1];
```

Примеры работы с элементами массива:

```
VAR:=Mass1[5]; //присвоение переменной Var значения из пятого элемента массива
VAR:=Mass1[5]+5*2;
VAR:=UDINT_TO_BOOL(Mass1[5]); //присвоение переменной Var преобразованного значения из пятого элемента массива
VAR:=(Mass1[i]); //присвоение переменной Var значения из i элемента массива,
 где i - переменная (локальная/входная)
```

При работе с массивами действуют следующие правила:

1. Массив может быть только одномерный.
2. Количество элементов массива ограничено до 32768 элементов.
3. Элементы массива могут быть следующих типов: BOOL, UDINT, REAL, TIME и DT.
4. Поддержка массива для функции:
   - функция может возвращать тип данных типа массив;

     Пример работы функции, которая возвращает массив:

     ```
     //функция MyFunction вернет переменную типа "Массив"
     function MyFunction: ARRAY [0..1] OF  UDINT ;
        var_input
           //входные переменные типа "Массив"
            Mass1 : ARRAY [0..1] OF  UDINT ;
        end_var
           //формируем возвращаемый массив в теле функции
     MyFunction [0]:= Mass1 [0];
     MyFunction [1]:= Mass1 [1];
     end_function
     ```
   - объявление переменной типа массив в функции происходит в области VAR...END_VAR и/или VAR_INPUT...END_VAR;

     Пример объявление переменных типа массив в разных областях функции:

     ```
     //функция MyFunction вернет переменную типа "Массив"
     function MyFunction: ARRAY [0..1] OF  UDINT ;
        var_input
           //входные переменные типа "Массив"
            Mass1 : ARRAY [0..1] OF  UDINT ;
        end_var
        var
     ```

```
        //локальные переменные типа "Массив"
         Mass2 : ARRAY [0..1] OF  UDINT ;
      end_var
   end_function
```

для функционального блока:

- объявление переменной типа массив в ФБ происходит в области VAR...END_VAR, VAR_INPUT...END_VAR и/или VAR_OUTPUT...END_VAR.

```
function_block MyFunctionblock
   var_input
      //объявление входной переменной типа "Массив"
       Mass1 : ARRAY [0..1] OF  UDINT ;
   end_var
   var_output
      //объявление выходной переменной типа "Массив"
       Mass2 : ARRAY [0..1] OF  UDINT ;
   end_var
      //объявление локальной переменной типа "Массив"
       Mass3 : ARRAY [0..1] OF  UDINT ;
   end_var
end_function_block
```

5. Последовательность значений по умолчанию для элементов массива заключается в квадратные скобки.
6. При инициализации элементов массива, количество значений по умолчанию должно соответствовать количеству элементов массива.

   Пример:

```
Mass1: ARRAY[0..3] OF UDINT:=[1,3,5,7];
```

   Ошибка:

```
Mass1: ARRAY[0..1] OF UDINT:=[1,5,7];
Mass1: ARRAY[0..1] OF UDINT:=[1];
```

Ограничения при работе с массивами:

1. Не поддержан побитовый доступ к элементам массива.

   Пример:

```
Mass1: ARRAY[0..1] OF UDINT:=[1,5];
Mass1[1].1:= TRUE; //ошибка
```

2. Не поддержано повторение последовательности элементов массива.

   Пример:

```
Mass1: ARRAY[0..5] OF UDINT:=[1,5,7(2)]; //где (2) означает повторение последовательности [1, 5, 7] 2 раза //ошибка
```

3. Не поддержано присвоение значения сразу нескольким элементам массива.

   Пример:

```
Mass1[1, 5]:=1; //ошибка: присвоение значения 1 первому и пятому элементам массива
```

Пример использования массивов.

12.4 Конструкции языка

К конструкциям языка ST относятся:

- арифметические операции 12.4.1.1;
- битовые операции 12.4.1.2;
- операции преобразования типов данных 12.4.1.3;
- логические операции 12.4.1.4;
- операции сравнения 12.4.1.5;
- операция присвоения 12.4.2;

- конструкция **IF – ELSIF – ELSE** 12.4.3;
- конструкция **CASE** 12.4.4;
- конструкция **RETURN** 12.4.5
- цикл **FOR** 12.4.6;
- цикл **WHILE** 12.4.7;
- цикл **REPEAT – UNTIL** 12.4.8.

ПРИМЕЧАНИЕ
При записи выражений допустимо использовать переменные (входные, выходные и локальные) и константы.

12.4.1 Операции

12.4.1.1 Арифметические операции

Операция	Ключевое слово	Типы данных	Пример использования
Сложение	+	IN, OUT: UDINT/REAL	OUT := IN1 + IN2 + …
Умножение	*		OUT := IN1 * IN2 * …
Вычитание	-		OUT := IN1 - IN2
Деление	/		OUT := IN1 / IN2
Остаток от целочисленного деления	MOD	IN, OUT: UDINT	OUT := IN1 MOD IN2
Абсолютное значение	ABS (IN)	IN, OUT: REAL	OUT := ABS (IN1)
Возведение в степень	POW (IN, N) IN – основание N – степень	IN, N, OUT: REAL	OUT := POW (IN1, N)

Доступные математические константы.

ПРИМЕЧАНИЕ
При записи арифметических выражений допустимо использование скобок для указания порядка вычислений. Действие, заключенное в скобки, имеет наивысший приоритет при вычислениях.

12.4.1.2 Битовые операции

Операция	Ключевое слово	Типы данных	Пример использования
Побитовый сдвиг влево	SHL (IN, N)	IN, OUT: UDINT N: 1..32	OUT := SHL (IN1, N)
Побитовый сдвиг вправо	SHR (IN, N)		OUT := SHR (IN1, N)
Дешифратор. Преобразует двоичный код в позиционный код.	DC32 (IN)	IN, OUT: UDINT	OUT := DC32 (IN1)
Шифратор. Преобразует позиционный код в двоичный код.	CD32 (IN)	IN, OUT: UDINT	OUT := CD32 (IN1)

12.4.1.3 Операции преобразования типов данных

Операция	Ключевое слово	Типы данных	Пример использования
UDINT в REAL	UDINT_TO_REAL (IN)	IN: UDINT OUT: REAL	OUT := UDINT_TO_REAL (IN)
UDINT в BOOL	UDINT_TO_BOOL (IN)	IN: UDINT OUT: BOOL	OUT := UDINT_TO_BOOL (IN)
UDINT в TIME	UDINT_TO_TIME (IN)	IN: UDINT OUT: TIME	OUT := UDINT_TO_TIME (IN)
UDIMT в DT	UDINT_TO_DT (IN)	IN: UDINT OUT: DT	OUT := UDINT_TO_DT (IN)
REAL в UDINT	REAL_TO_UDINT (IN)	IN: REAL OUT: UDINT	OUT := REAL_TO_UDINT (IN)

Операция	Ключевое слово	Типы данных	Пример использования
REAL в BOOL	REAL_TO_BOOL (IN)	IN: REAL OUT: BOOL	OUT := REAL_TO_BOOL (IN)
BOOL в REAL	BOOL_TO_REAL (IN)	IN: BOOL OUT: REAL	OUT := BOOL_TO_REAL (IN)
BOOL в UDINT	BOOL_TO_UDINT (IN)	IN: BOOL OUT: UDINT	OUT := BOOL_TO_UDINT (IN)
TIME в UDINT	TIME_TO_UDINT (IN)	IN: TIME OUT: UDINT	OUT := TIME_TO_UDINT (IN)
DT в UDINT	DT_TO_UDINT (IN)	IN: DT OUT: UDINT	OUT := DT_TO_UDINT (IN)

12.4.1.4 Логические операции

Операция	Ключевое слово	Типы данных	Пример использования
Логическое отрицание «НЕ»	NOT	IN, OUT: BOOL	OUT := NOT IN1
Конъюнкция Логическое умножение «И»	AND &		OUT := IN1 AND IN2 OUT := IN1 & IN2
Дизъюнкция Логическое сложение «ИЛИ»	OR		OUT := IN1 OR IN2
Исключающее ИЛИ	XOR		OUT := IN1 XOR IN2

12.4.1.5 Операции сравнения

Операция	Ключевое слово	Типы данных	Пример использования
Больше	>	IN: UDINT/REAL OUT: BOOL	OUT := IN1 > IN2
Больше или равно	>=		OUT := IN1 >= IN2
Равно	=		OUT := IN1 = IN2
Меньше или равно	<=		OUT := IN1 <= IN2
Меньше	<		OUT := IN1 < IN2
Неравно	<>		OUT := IN1 <> IN2

12.4.1.6 Приоритеты операций

ПРИМЕЧАНИЕ
Операции в таблице расположены **от наивысшего к низшему приоритету**. Чем выше приоритет операции, тем раньше она выполняется.

Операция	Ключевое слово
Скобки	(выражение)
Вызов функции и функционального блока Битовые операции	Пример: fb1(); function1 := … ;
Унарный минус Логическое отрицание	– NOT
Возведение в степень	POW
Умножение Деление Остаток от целочисленного деления	* / MOD
Сложение	+
Вычитание	–

Операция	Ключевое слово
Операции сравнения	>, <, >=, <=
Равно Неравно	= <>
Конъюнкция Логическое умножение «И»	& AND
Исключающее ИЛИ	XOR
Дизъюнкция Логическое сложение «ИЛИ»	OR

12.4.2 Присвоение

Для обозначения присвоения используется парный знак «**:=**». В правой и левой части выражения должны быть операнды одного типа (автоматического приведения типов не предусмотрено). В левой части выражения (принимающая сторона) может быть использована только переменная. Правая часть может содержать выражение или константу.

12.4.3 Оператор IF

Оператор **IF** позволяет проверить одно или несколько условий, и, если хотя бы одно из условий является истинным, выполнить заданные для этого условия выражения. После выполнения выражений производится выход из оператора – то есть оставшиеся условия уже не проверяются.

Рассмотрим работу оператора на примере сигнализации выхода значения температуры за допустимые границы:

```
FUNCTION_BLOCK fb //имя функционального блока

    VAR_INPUT //объявление входных переменных
        rTemp : REAL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        xHigh : BOOL;
        xLow : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        rHighTemp : REAL := 20;
        rLowTemp : REAL := 10;
    END_VAR

    //область кода

    IF rTemp > rHighTemp THEN
        xHigh := TRUE;
    ELSIF rTemp < rLowTemp THEN
        xLow := TRUE;
    ELSE
        xHigh := FALSE;
        xLow := FALSE;
    END_IF

END_FUNCTION_BLOCK
```

Если условие в операторе **IF** выполняется (значение переменной **rTemp** больше **rHighTemp**), то переменной **xHigh** будет присвоено значение **TRUE** и произойдет выход из оператора (следующее условие проверяться не будет). Если же условие не выполняется, то произойдет проверка следующего условия, размещенного во вложенном операторе **ELSIF**. Если условие в **ELSIF** выполняется (значение переменной **rTemp** меньше **rLowTemp**), то переменной **xLow** будет присвоено значение **TRUE** и произойдет выход из оператора (следующее условие проверяться не будет). Если ни одно из условий в **IF** и **ELSIF** не выполняется (то есть значение температуры находится в допустимых пределах), то произойдет выполнение выражений, размещенных во вложенном операторе **ELSE** – присвоение переменным **xHigh** и **xLow** значения **FALSE**.

Использование вложенных операторов **ELSIF** и **ELSE** является необязательным. Внутри оператора **IF** может быть размещено произвольное число операторов **ELSIF**.

Конструкция допускает вложенность, то есть внутри одного **IF** может быть еще один и т. д. Также внутри оператора **IF** могут быть использованы циклы и оператор **CASE 12.4.4**.

12.4.4 Оператор CASE

Оператор **CASE** позволяет сравнить значение заданной целочисленной переменной (селектора) с набором констант или целочисленных значений (метками), и в случае совпадения выполнить заданные для этой метки выражения. После выполнения выражений производится выход из оператора.

Рассмотрим работу оператора на примере:

```
FUNCTION_BLOCK fb1 //имя функционального блока

    VAR_INPUT //объявление входных переменных
        udiSel : UDINT;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        xOut1 : BOOL;
        xOut2 : BOOL;
        xOut3 : BOOL;
        xOut4 : BOOL;
    END_VAR

    //область кода

    xOut1 := FALSE;
    xOut2 := FALSE;
    xOut3 := FALSE;
    xOut4 := FALSE;

    CASE udiSel OF
        0:
            xOut1 := TRUE;
        1..3:
            xOut2 := TRUE;
        4, 6:
            xOut3 := TRUE;
    ELSE
        xOut4 := TRUE;
    END_CASE

END_FUNCTION_BLOCK
```

Если значение **udiSel**:

- Равно 0, то **xOut1** примет значение **TRUE**;
- Попадет в диапазон 1..3, то **xOut2** примет значение **TRUE**;
- Равно 4 или 6, то **xOut3** примет значение **TRUE**;
- Не попадет ни в одно из указанных значений, то **xOut4** примет значение **TRUE**;

Как видно из примера – метка может включать в себя несколько значений, перечисленных через запятую «4, 6», или диапазон «1..3». При этом значения одной из меток не должны совпадать со значениями других. Также, при указании диапазона значений начало диапазона должно быть меньше его конца.

Вложенный оператор **ELSE** является необязательным; размещенные в нём выражения выполняются в том случае, если значение селектора не совпало ни с одной из меток.

Действия, предусмотренные для обработки каждого из случаев **CASE**, могут использовать циклы, операторы **IF** и **CASE**.

12.4.5 Оператор RETURN

Оператор **RETURN** позволяет выйти из программного объекта.

Пример использования:

```
IF xDone THEN
    RETURN;
END_IF;

udiCounter := udiCounter + 1;
```

Если переменная **xDone** примет значение **TRUE**, то выражение «**udiCounter** := **udiCounter** + 1» выполнятся не будет (как и все, что будет находится ниже в теле программы).

12.4.6 Цикл FOR

Оператор **FOR** применяется для организации цикла с заранее известным числом итераций. Обычно он используется для операций над массивами данных.

Внутри цикла могут использоваться операторы **IF** и **CASE**, а также другие операторы цикла.

В качестве примера рассмотрим реализацию пузырьковой сортировки от меньшего к большему:

```
FUNCTION_BLOCK MaxI_MinI //Максимальное и минимальное число при помощи пузырьковой
 // сортировки от меньшего к большему

    VAR_INPUT //объявление входных переменных
        udiX1, udiX2, udiX3, udiX4, udiX5 : UDINT;
        //Добавление необходимого кол-ва входных переменных и определение типа данных
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        udiMaxI, udiMinI : UDINT;
    END_VAR

    VAR //объявление локальных переменных
        udiI, udiJ, udiN, udiK : UDINT;
        audiX : ARRAY [1..5] OF UDINT; //Задается диапазон массива и тип данных массива
    END_VAR

    //область кода
    //объявление массива
    audiX[1] := udiX1;
    audiX[2] := udiX2;
    audiX[3] := udiX3;
    audiX[4] := udiX4;
    audiX[5] := udiX5;
    udiN := 5; // задается кол-во чисел для сортировки (размер массива)

    FOR udiI := 1 TO udiN-1 DO
        FOR udiJ := 1 TO udiN-udiI DO
            IF audiX[udiJ] > audiX[udiJ+1] THEN
                udiK := audiX[udiJ];
                audiX[udiJ] := audiX[udiJ+1];
                audiX[udiJ+1] := udiK;
            END_IF;
        END_FOR;
    END_FOR;

    udiMaxI := audiX[udiN]; //выводится последнее (максимальное) число массива
    udiMinI := audiX[1]; //выводится первое (минимальное) число массива

END_FUNCTION_BLOCK
```

Переменная **udiI** называется итератором (счетчиком) цикла. Эта переменная должна принадлежать целочисленному знаковому типу (**UDINT**). После каждого выполнения тела цикла происходит изменение значения итератора – по умолчанию на **+1**. Пользователь может сам задать «шаг» итератора с помощью вложенного оператора **BY**. После этого сразу происходит переход к следующей итерации – то есть весь цикл выполняется «от начала и до конца». Структура будет выглядеть следующим образом:

```
// udiI будет принимать значения 1, 4, 7, 10 и т.д.
FOR udiI := 1 TO udiN-1 BY 3 DO
    .. // код цикла
END_FOR
```

Итератор (счетчик) цикла, его начальное и конечное значение, а также приращение являются целочисленными. Они вычисляются до входа в цикл и изменение значений переменных, входящих в любое из этих выражений, не приведет к изменению числа итераций.

Досрочно выйти из цикла можно при помощи оператора **EXIT**. Пример структуры:

```
FOR udiI := 1 TO udiN-1 DO
    IF audiX[2] > 100 THEN
        EXIT;
    ELSE
        .. // код цикла
    END_IF
END_FOR
```

Пропустить шаг цикла можно при помощи оператора **CONTINUE**.

```
FOR udiI := 1 TO udiN-1 DO
    IF udiI = 3 THEN
        CONTINUE;
    ELSE
        .. // код цикла
        END_IF
END_FOR
```

12.4.7 Цикл WHILE

Оператор **WHILE** используется для создания цикла с заранее неизвестным числом итераций. Завершение цикла произойдет, если проверяемое условие вернет **FALSE**. При этом проверка условия выполняется ДО выполнения выражения (цикл с предусловием). Таким образом, если условие сразу возвращает **FALSE**, то цикл не выполнится ни разу.

Внутри цикла могут использоваться операторы **IF** и **CASE**, а также другие операторы цикла.

Пример конструкции:

```
WHILE rVar < 100 DO
    rVar := rVar + 1;
END_WHILE
```

Результатом выполнения данного цикла (с начальным значением переменной **rVar** := 10) будет число 100.

Досрочно выйти из цикла можно при помощи оператора **EXIT**.

Пропустить шаг цикла можно при помощи оператора **CONTINUE**.

12.4.8 Цикл REPEAT UNTIL

Оператор **REPEAT** используется для создания цикла с заранее неизвестным числом итераций. Завершение цикла произойдет, если проверяемое условие вернет **TRUE**. При этом проверка условия выполняется ПОСЛЕ выполнения выражения (цикл с постусловием). Таким образом, если условие сразу возвращает **TRUE**, то цикл выполнится один раз.

Внутри цикла могут использоваться операторы **IF** и **CASE**, а также другие операторы цикла.

Пример конструкции:

```
REPEAT
    IF rVar > 100 THEN
        EXIT;
    END_IF;
    rVar := rVar + 1;
UNTIL rVar > 180
END_REPEAT;
```

Результатом выполнения данного цикла (с начальным значением переменной **rVar** := 10) будет число 101.

Для досрочного выхода из цикла можно использовать оператор **EXIT**.

Пропустить шаг цикла можно при помощи оператора **CONTINUE**.

12.5 Системные функции

В редакторе ST поддержаны системные функции:

- функции работы с датой и временем;
- математические функции;
- логарифмические функции;
- тригонометрические функции.

12.5.1 Функции работы с датой и временем

- функция GET_TIME;
- функция GET_DATE_TIME;
- функция SYS.COMPARE_DATE_TIME;
- функция SYS.IS_LEAP_YEAR.

12.5.1.1 Функция GET_TIME

Функция **GET_TIME** возвращает значение типа TIME (4 байта), содержащее время, прошедшее с момента последнего включения прибора, в миллисекундах.

Пример

```
VAR
    Time_1 : TIME := T#0ms;
    Time_2 : TIME := T#0ms;
    Q :      BOOL := FALSE;
END_VAR

IF Time_1 = T#0ms THEN
   Time_1 := GET_TIME();
END_IF

Time_2 := GET_TIME();

IF (Time_2 - Time_1) >= T#1000ms THEN
    Q      := NOT Q;
    Time_1 := T#0ms;
    Time_2 := T#0ms;
END_IF
```

12.5.1.2 Функция GET_DATE_TIME

Функция GET_DATE_TIME возвращает значение тип DT (4 байта), содержащее данные часов реального времени, в секундах с 00:00:00 1.01.2000 г., с учетом часового пояса, установленного в приборе.

Пример

```
VAR
    Ton_UDINT : UDINT;
    Ton_DT : DT;
END_VAR

Ton_DT := GET_DATE_TIME();

Ton_UDINT := DT_TO_UDINT(Ton_DT);
```

12.5.1.3 Функция SYS.COMPARE_DATE_TIME

Функция SYS.COMPARE_DATE_TIME сравнивает два значения UDINT, поданных на вход по заданной маске даты/времени и возвращает значение типа UDINT, которое приравнивается к:

- 1 – значение 1 больше значения 2;
- 2 – значение 1 меньше значения 2;
- 0 – значение 1 равно значению 2.

Сравнение происходит по количеству секунд, начиная с 00:00:00 01.01.2000

Пример

```
FUNCTION udiCompare: UDINT;

    VAR_INPUT
        udiValue1:  UDINT; //значение 1
    END_VAR

    VAR
        udiValue2 : UDINT; //значение 2
        udiMask :   UDINT := 63; //маска даты/времени
    END_VAR

    udiValue2  := DT_TO_UDINT (GET_DATE_TIME ());
    udiCompare := SYS.COMPARE_DATE_TIME (udiValue1, udiValue2, udiMask);

END_FUNCTION
```

Маска = 63 (0b111111) – используются все разряды:

0 бит – если 1, то используются секунды;

1 бит – если 1, то используются минуты;

2 бит – если 1, то используются часы;

3 бит – если 1, то используются дни;

4 бит – если 1, то используются месяцы;

5 бит – если 1, то используются годы.

12.5.1.4 Функция SYS.IS_LEAP_YEAR

Функция **SYS.IS_LEAP_YEAR** возвращает значение типа BOOL, содержащее данные о том, соответствует ли високосному году число типа UDINT поданное на вход функции (1 – високосный, 0 – нет).

Пример

```
FUNCTION xLeapYear: BOOL; //имя функции и тип данных выхода

    VAR_INPUT //объявление входных переменных
        udiYear : UDINT; //проверяемый год
    END_VAR

    xLeapYear := SYS.IS_LEAP_YEAR (udiYear);

END_FUNCTION
```

12.5.2 Математические функции

- математические константы;
- MIN/FMIN;
- MAX/FMAX;
- SQRT;
- EXP;
- FACT.

12.5.2.1 Математические константы (CONST.*)

Константа	Значение	Назначение
CONST.PI	3.1415927	Число Пи (π)
CONST.E	2.7182817	Число Эйлера (e)
CONST.POS_INF	+ ∞	Положительная бесконечность
CONST.NEG_INF	– ∞	Отрицательная бесконечность

12.5.2.2 MIN/FMIN

Функции MIN/FMIN — возвращают наименьшие значения из двух входных переменных.

Входные переменные и возвращаемое значение для функции:

- **MIN** — для типа UDINT (оба аргумента и результат);
- **FMIN** — для типа REAL (оба аргумента и результат).

Пример 1

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 7;
    Var_3: UDINT := 5;
END_VAR
Var_1 := SYS.MIN(Var_2, Var_3); // возвращаемое значение 5
```

Пример 2

```
VAR
    Var_1: REAL;
    Var_2: REAL := 7.1;
    Var_3: REAL := 8.2;
END_VAR
Var_1 := SYS.FMIN(Var_2, Var_3); // возвращаемое значение 7.1
```

12.5.2.3 MAX/FMAX

Функции MAX/FMAX — возвращают наибольшие значения из двух входных переменных.

Входные переменные и возвращаемое значение для функции:

- **MAX** — для типа UDINT (оба аргумента и результат);
- **FMAX** — для типа REAL (оба аргумента и результат).

Пример 1

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 7;
    Var_3: UDINT := 8;
END_VAR
Var_1 := SYS.MAX(Var_2, Var_3); // возвращаемое значение 8
```

Пример 2

```
VAR
    Var_1: REAL;
    Var_2: REAL := 7.5;
    Var_3: REAL := 5.2;
END_VAR
Var_1 := SYS.FMAX(Var_2, Var_3); // возвращаемое значение 7.5
```

12.5.2.4 SQRT

Функция SQRT - возвращает квадратный корень входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 4.0;
END_VAR
Var_1 := SYS.SQRT(Var_2); // возвращаемое значение 2.0
```

12.5.2.5 EXP

Функция EXP — возвращает значение экспоненты, возведенной в степень входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.0;
END_VAR
Var_1 := SYS.EXP(Var_2); // возвращаемое значение 2.7182817
```

12.5.2.6 FACT

Функция FACT — возвращает факториал входного числа. Функция возвращает корректные значения при значении входной переменной, принадлежащем интервалу от 0 до 12. Во всех остальных случаях функция возвращает 0.

Входная переменная и возвращаемое значение типа UDINT.

Пример

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 5;
END_VAR
Var_1 := SYS.FACT(Var_2); // возвращаемое значение 120
```

12.5.3 Логарифмические функции

- LOG;
- LG;
- LN.

12.5.3.1 LOG

Функция LOG — возвращает логарифм входного числа с произвольным основанием.

Входная переменная и возвращаемое значение типа REAL.

Основание логарифма	Число логарифма	Результат
< 0	Любое	NaN*
NaN	1	0
+ inf**	0	NaN
+ inf	>0	0
+ inf	+ inf	NaN
0	> 0, ≠ 1	0
0	0	NaN
0	+ inf	NaN
0	1	0
1	Любое	NaN
NaN	NaN	NaN
> 0, ≠ 1	> 0	число
> 0, ≠ 1	0	- inf***
> 0, ≠ 1	+ inf	+ inf

*NaN (Not-a-Number) — не число;

**+ inf (infinity) — положительная бесконечность;

***- inf (infinity) — отрицательная бесконечность.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 2; // основание логарифма
    Var_3: REAL := 8; // число логарифма
END_VAR
Var_1 := SYS.LOG(Var_2, Var_3); // возвращаемое значение 3
```

12.5.3.2 LG

Функция LG — возвращает десятичный логарифм входного числа.

Входная переменная и возвращаемое значение типа REAL.

Число логарифма	Результат
0	- inf*
< 0	NaN**
NaN	NaN
+ inf***	+ inf

*- inf (infinity) — отрицательная бесконечность.

**NaN (Not-a-Number) — не число;

**+ inf (infinity) — положительная бесконечность;

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 10;
END_VAR
Var_1 := SYS.LG(Var_2); // возвращаемое значение 1
```

12.5.3.3 LN

Функция LN — возвращает натуральный логарифм входного числа.

Входная переменная и возвращаемое значение типа REAL.

Число логарифма	Результат
0	- inf*
< 0	NaN**
NaN	NaN
+ inf***	+ inf

*- inf (infinity) — отрицательная бесконечность.

**NaN (Not-a-Number) — не число;

**+ inf (infinity) — положительная бесконечность;

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 2.718281;
END_VAR
Var_1 := SYS.LN(Var_2); // возвращаемое значение ~1
```

12.5.4 Тригонометрические функции

- SIN;
- COS;
- TAN;

- ASIN;
- ACOS;
- ATAN.

12.5.4.1 SIN

Функция SIN - возвращает значение sin (синуса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.5708; // в радианах
END_VAR
Var_1 := SYS.SIN(Var_2); // возвращаемое значение 1
```

12.5.4.2 COS

Функция COS - возвращает значение cos (косинуса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.0472; // в радианах
END_VAR
Var_1 := SYS.COS(Var_2); // возвращаемое значение 0.49999794
```

12.5.4.3 TAN

Функция TAN - возвращает значение tan (тангенса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.3734; // в радианах
END_VAR
Var_1 := SYS.TAN(Var_2); // возвращаемое значение 4.999982
```

12.5.4.4 ASIN

Функция ASIN - возвращает значение arcsin (арксинуса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 0.5;
END_VAR
Var_1 := SYS.ASIN(Var_2); // возвращаемое значение 0.52359825
```

ПРИМЕЧАНИЕ
Относительная погрешность результата функции составляет 1e-5.

12.5.4.5 ACOS

Функция ACOS - возвращает значение arccos (арккосинуса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 0.5;
END_VAR
Var_1 := SYS.ACOS(Var_2); // возвращаемое значение 1.047198
```

12.5.4.6 ATAN

Функция ATAN - возвращает значение arctg (арктангенса) входного числа.

Входная переменная и возвращаемое значение типа REAL.

Пример

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.73205; // 3^(1/2)
END_VAR
Var_1 := SYS.ATAN(Var_2); // возвращаемое значение 1.0471973
```

12.6 Системные функциональные блоки

В редакторе ST поддержаны системные функциональные блоки:

- Триггеры;
- Таймеры;
- Генераторы;
- Счетчики.

Использование системных функциональных блоков в других элементах программы аналогично использованию пользовательских функциональных блоков.

Для каждого системного функционального блока редактор предлагает подсказку по его использованию.

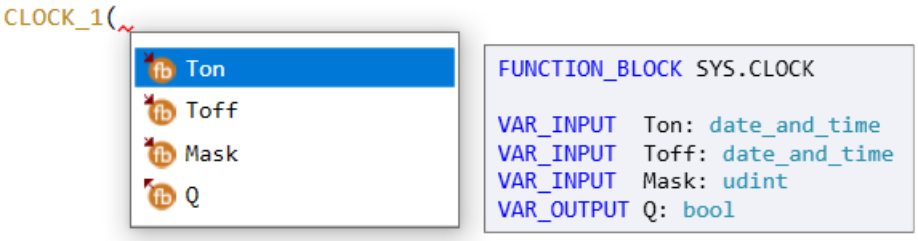


Рисунок 12.6

12.6.1 Триггеры

- RS-триггер с приоритетом выключения (SYS.RS) 12.6.1.1;
- SR-триггер с приоритетом включения (SYS.SR) 12.6.1.2;
- Детектор переднего фронта импульса (SYS.RTRIG) 12.6.1.3;
- Детектор заднего фронта импульса (SYS.FTRIG) 12.6.1.4;
- D-триггер (SYS.DTRIG) 12.6.1.5.

12.6.1.1 RS-триггер с приоритетом выключения (SYS.RS)

RS-триггер с приоритетом выключения (SYS.RS) используется для переключения с фиксацией состояния во время поступления коротких импульсов на соответствующий вход. На выходе Q появится логическая «1» по фронту сигнала на входе S.

Обозначение	Тип данных	Описание
Входы		
S	BOOL	Вход SET
R	BOOL	Вход RESET
Выходы		
Q	BOOL	Выход триггера

```
FUNCTION_BLOCK RS_trigger //имя функционального блока

    VAR_INPUT //объявление входных переменных
        R_in : BOOL;
        S_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        RS_1: SYS.RS;
    END_VAR

    //область кода

    RS_1(R := R_in, S := S_in, Q => Q_out);

END_FUNCTION_BLOCK
```

В случае одновременного поступления сигналов на оба входа приоритетным является сигнал входа R.

12.6.1.2 SR-триггер с приоритетом включения (SYS.SR)

SR-триггер с приоритетом включения (SYS.SR) используется для переключения с фиксацией состояния во время поступления коротких импульсов на соответствующий вход. На выходе Q появится логическая «1» по фронту сигнала на входе S.

Обозначение	Тип данных	Описание
Входы		
S	BOOL	Вход SET
R	BOOL	Вход RESET
Выходы		
Q	BOOL	Выход триггера

```
FUNCTION_BLOCK SR_trigger //имя функционального блока

    VAR_INPUT //объявление входных переменных
        S_in : BOOL;
        R_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        SR_1: SYS.SR;
    END_VAR

    //область кода

    SR_1(S := S_in, R := R_in, Q => Q_out);

END_FUNCTION_BLOCK
```

В случае одновременного поступления сигналов на оба входа приоритетным является сигнал входа S.

12.6.1.3 Детектор переднего фронта импульса (SYS.RTRIG)

Детектор переднего фронта импульса (SYS.RTRIG) используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе Q генерируется единичный импульс по переднему фронту входа I.

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Вход триггера
Выходы		
Q	BOOL	Выход триггера

```
FUNCTION_BLOCK R_trigger //имя функционального блока

    VAR_INPUT //объявление входных переменных
        RT_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        RT_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        RTrig_1: SYS.RTRIG;
    END_VAR

    //область кода

    RTrig_1(I := RT_in, Q => RT_out);

END_FUNCTION_BLOCK
```

12.6.1.4 Детектор заднего фронта импульса (SYS.FTRIG)

Детектор заднего фронта импульса (SYS. FTRIG) используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе Q генерируется единичный импульс по заднему фронту входа I.

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Вход триггера
Выходы		
Q	BOOL	Выход триггера

```
FUNCTION_BLOCK F_trigger //имя функционального блока

    VAR_INPUT //объявление входных переменных
        FT_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        FT_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        FTrig_1: SYS.FTRIG;
    END_VAR

    //область кода

    FTrig_1(I := FT_in, Q => FT_out);

END_FUNCTION_BLOCK
```

12.6.1.5 D-триггер (SYS.DTRIG)

D-триггер (SYS.DTRIG) используется для формирования импульса включения выхода на интервал времени импульса на входе D, выходной интервал будет синхронизирован с тактовой частотой на входе C.

На выходе Q триггера появится сигнал логической «1» по фронту тактовых импульсов на входе C при наличии сигнала логической «1» на входе D. Возврат выхода Q в сигнал логического «0» произойдет по фронту тактовых импульсов на входе C при наличии сигнала логического «0» на входе D.

Вход S принудительно устанавливает выход Q в состояние логической «1».

Вход R является приоритетным и устанавливает выход Q в состояние логического «0».

Обозначение	Тип данных	Описание
Входы		
S	BOOL	Вход SET
D	BOOL	Вход триггера
C	BOOI	Тактовая частота
R	BOOL	Вход RESET
Выходы		
Q	BOOL	Выход триггера

```
FUNCTION_BLOCK D_trigger //имя функционального блока

    VAR_INPUT //объявление входных переменных
        S_in : BOOL;
        D_in : BOOL;
        C_in : BOOL;
        R_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        DTrig_1: SYS.DTRIG;
    END_VAR

    //область кода

    DTrig_1(S := S_in, D := D_in, C := C_in, R := R_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2 Таймеры

- Импульс включения заданной длительности (SYS.TP) 12.6.2.1;
- Таймер с задержкой включения (SYS.TON) 12.6.2.2;
- Таймер с задержкой отключения (SYS:TOF) 12.6.2.3;
- Интервальный таймер (SYS.CLOCK) 12.6.2.4;
- Интервальный таймер с недельным циклом (SYS:CLOCKWEEK) 12.6.2.5.

12.6.2.1 Импульс включения заданной длительности (SYS.TP)

Импульс включения заданной длительности (SYS.TP) используется для формирования импульса включения выхода на заданный интервал времени. На выходе Q блока появляется сигнал логической «1» по фронту входного сигнала I. После запуска выход Q не реагирует на изменение значения входного сигнала в течение интервала $T_{имп}$. По истечение интервала $T_{имп}$ выходной сигнал сбрасывается в логический «0».

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Включение таймера
T	TIME	Длительность импульса
Выходы		
Q	BOOL	Выход таймера

```
FUNCTION_BLOCK TP_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TP_1: SYS.TP;
    END_VAR

    //область кода
    TP_1(I := I_in, T := T#1000ms);
    //где ms - миллисекунды, s - секунды, m - минуты, h - часы, d - дни
    Q_out := TP_1.Q;
END_FUNCTION_BLOCK
```

```
FUNCTION_BLOCK TP_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
        T_in : UDINT := 5000;//миллисекунд
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TP_1: SYS.TP;
        T_time: TIME;
    END_VAR

    //область кода

    T_time := UDINT_TO_TIME(T_in);
    TP_1(I := I_in, T := T_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.2 Таймер с задержкой включения (SYS.TON)

Таймер с задержкой включения (SYS. TON) используется для операции задержки передачи сигнала. На выходе Q таймера появится сигнал логической «1» с задержкой относительно фронта входного сигнала I продолжительностью не менее длительности Ton и выключится по спаду входного сигнала.

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Запуск таймера (по переднему фронту)
T	TIME	Задержка на включение
Выходы		
Q	BOOL	Выход таймера

```
FUNCTION_BLOCK TON_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TON_1: SYS.TON;
    END_VAR

    //область кода

    TON_1(I := I_in, T := T#1000ms);
    //где ms - миллисекунды, s - секунды, m - минуты, h - часы, d - дни
    Q_out := TON_1.Q;

END_FUNCTION_BLOCK
```

```
FUNCTION_BLOCK TON_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
        Ton_in : UDINT := 5000;//миллисекунд
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TON_1: SYS.TON;
        Ton_time: TIME;
    END_VAR

    //область кода

    Ton_time := UDINT_TO_TIME(Ton_in);
    TON_1(I := I_in, T := Ton_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.3 Таймер с задержкой выключения (SYS.TOF)

Таймер с задержкой отключения (SYS. TOF) используется для задержки отключения выхода. На выходе Q таймера появится сигнал логической «1» по фронту сигнала на входе I, отсчет времени задержки отключения Toff начнется по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится сигнал логического «0» с задержкой Toff.

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Запуск таймера (по заднему фронту)
T	TIME	Задержка на включение
Выходы		
Q	BOOL	Выход таймера

```
FUNCTION_BLOCK TOF_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TOF_1: SYS.TOF;
    END_VAR

    //область кода

    TOF_1(I := I_in, T := T#1000ms);
    //где ms - миллисекунды, s - секунды, m - минуты, h - часы, d - дни
    Q_out := TOF_1.Q;

END_FUNCTION_BLOCK
```

```
FUNCTION_BLOCK TOF_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
        Tof_in : UDINT := 5000;//миллисекунд
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        TOF_1: SYS.TOF;
        Tof_time: TIME;
    END_VAR

    //область кода

    Tof_time := UDINT_TO_TIME(Tof_in);
    TOF_1(I := I_in, T := Tof_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.4 Интервальный таймер (SYS.CLOCK)

Интервальный таймер (SYS. CLOCK) используется для формирования импульса включения выхода Q по часам реального времени. Время включения Ton и отключения Toff выхода устанавливают в качестве параметров таймера.

Обозначение	Тип данных	Описание
Входы		
Ton	DT	Время включения
Toff	DT	Время выключения
Mask	UDINT	Выбор используемых величин
Выходы		
Q	BOOL	Выход таймера

ПРИМЕЧАНИЕ
Указание переменной Mask необязательно.
Если значение переменной Mask не указано, то по умолчанию блок работает с маской Mask = 63 (0b111111),
Где:
Mask = 63 (0b111111)
0 бит – если 1, то используются секунды
1 бит – если 1, то используются минуты
2 бит – если 1, то используются часы
3 бит – если 1, то используются дни
4 бит – если 1, то используются месяцы
5 бит – если 1, то используются годы

```
FUNCTION_BLOCK CLOCK_timer //имя функционального блока

    VAR_INPUT //объявление входных переменных
       I_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        CLOCK_1: SYS.CLOCK;
    END_VAR

    //область кода

    CLOCK_1(Ton := DT#2023-09-28-7:20:55, Toff := DT#2023-09-28-12:30:59);

    Q_out := CLOCK_1.Q;

END_FUNCTION_BLOCK
```

12.6.2.5 Интервальный таймер с недельным циклом (SYS.CLOCKWEEK)

Интервальный таймер с недельным циклом (SYS.CLOCKWEEK) используется для формирования импульса включения выхода Q по часам реального времени с учетом дней недели. Время включения Ton и отключения Toff выхода Q и дни недели работы устанавливают в качестве параметров таймера.

Обозначение	Тип данных	Описание
Входы		
Ton	DT	Время включения
Toff	DT	Время выключения
DayOfWeekMask	UDINT	Выбор используемых дней
DateTimeMask	UDINT	Выбор используемых величин
Выходы		
Q	BOOL	Выход таймера

ПРИМЕЧАНИЕ
Указание переменных DayOfWeekMask и DateTimeMask необязательно.
Если значение переменной DayOfWeekMask не указано, то по умолчанию блок работает с маской DayOfWeekMask = 127 (0b1111111),
Если значение переменной DateTimeMask не указано, то по умолчанию блок работает с маской DateTimeMask = 63 (0b111111),
Где:
DayOfWeekMask = 127 (0b1111111)
0 бит – если 1, то учитываются понедельники
1 бит – если 1, то учитываются вторники
2 бит – если 1, то учитываются среды
3 бит – если 1, то учитываются четверги
4 бит – если 1, то учитываются пятницы
5 бит – если 1, то учитываются субботы
6 бит – если 1, то учитываются воскресенья

DateTimeMask = 63 (0b111111)
0 бит – если 1, то используются секунды
1 бит – если 1, то используются минуты
2 бит – если 1, то используются часы
3 бит – если 1, то используются дни
4 бит – если 1, то используются месяцы
5 бит – если 1, то используются годы

```
FUNCTION_BLOCK CLOCKWEEK_timer //имя функционального блока
```

```
    VAR_INPUT //объявление входных переменных
       I_in : BOOL;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        CLOCKWEEK_1: SYS.CLOCKWEEK;
    END_VAR

    //область кода

    CLOCKWEEK_1(Ton := DT#2023-09-28-7:20:55, Toff := DT#2023-09-28-12:30:59);
    Q_out := CLOCKWEEK_1.Q;
END_FUNCTION_BLOCK
```

12.6.3 Генераторы

- Генератор прямоугольных импульсов (SYS.BLINK) 12.6.3.1.

12.6.3.1 Генератор прямоугольных импульсов (SYS.BLINK)

Генератор прямоугольных импульсов (SYS. BLINK) используется для формирования прямоугольных импульсов пульсации. На выходе Q генератора формируются импульсы с заданными параметрами длительности включенного ($T_{вкл}$ – сигнал логической «1») и отключенного ($T_{откл}$ – сигнал логического «0») состояния на время действия управляющего сигнала на входе I (сигнал логической «1»).

Обозначение	Тип данных	Описание
Входы		
I	BOOL	Разрешение на работу
Th	TIME	Длительность логической единицы
Tl	TIME	Длительность логического нуля
Выходы		
Q	BOOL	Выход генератора

```
FUNCTION_BLOCK BLINK_generator //имя функционального блока

    VAR_INPUT //объявление входных переменных
       I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        BLINK_1: SYS.BLINK;
    END_VAR

    //область кода

    BLINK_1(I := I_in, Th := T#1000ms, Tl := T#1000ms);
    //где ms - миллисекунды, s - секунды, m - минуты, h - часы, d - дни

    Q_out := BLINK_1.Q;

END_FUNCTION_BLOCK
```

```
FUNCTION_BLOCK BLINK_generator //имя функционального блока
```

```
    VAR_INPUT //объявление входных переменных
        I_in : BOOL := FALSE;
        Th_in : UDINT := 5000;//миллисекунд
        Tl_in : UDINT := 5000;//миллисекунд
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR
        BLINK_1: SYS.BLINK;
        Th_time: TIME;
        Tl_time: TIME;
    END_VAR

    //область кода

    Th_time := UDINT_TO_TIME(Th_in);
    Tl_time := UDINT_TO_TIME(Tl_in);
    BLINK_1(I := I_in, Th := Th_time, Tl := Tl_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4 Счетчики

- Инкрементный счетчик с автосбросом (SYS.CT) 12.6.4.1;
- Универсальный счетчик (SYS.CTN) 12.6.4.2;
- Инкрементный счетчик (SYS.CTU) 12.6.4.3.

12.6.4.1 Инкрементный счетчик с автосбросом (SYS.CT)

Инкрементный счетчик с автосбросом (SYS. CT) используется для подсчета заданного числа импульсов **N** (вход **N** – уставка числа импульсов). На выходе **Q** счетчика появится импульс сигнала логической «1» с длительностью рабочего цикла прибора (Тцикл), если число приходящих на вход **C** импульсов достигнет установленного значения **N**.

Обозначение	Тип данных	Описание
	Входы	
C	BOOL	Вход счетчика
N	UDINT	Уставка счетчика
	Выходы	
Q	BOOL	Сигнализация достигнутой уставки (длительность один цикл)

```
FUNCTION_BLOCK CT_counter //имя функционального блока

    VAR_INPUT //объявление входных переменных
        C_in : BOOL;
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        CT_1: SYS.CT;
    END_VAR

    //область кода

    CT_1(C := C_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4.2 Универсальный счетчик (SYS.CTN)

Универсальный счетчик (SYS.CTN) используется для прямого и обратного счета. Операция «прямой счет» выполняется по переднему фронту импульса на входе прямого счета **U**, что увеличивает значение выходного сигнала **Q**. Импульсы, приходящие на вход **D** («обратный счет»), уменьшают значение выхода **Q**. В случае поступления на вход **R** сигнала логической «1», выход счетчика **Q** устанавливается в значение входа N.

Обозначение	Тип данных	Описание
Входы		
U	BOOL	Вход для прямого счета
D	BOOL	Вход для обратного счета
R	BOOL	Сброс состояния выхода на значение N
N	UDINT	Уставка
Выходы		
Q	UDINT	Накопительное значение импульсов

```
FUNCTION_BLOCK CTN_counter //имя функционального блока

    VAR_INPUT //объявление входных переменных
        U_in : BOOL;
        D_in : BOOL;
        R_in : BOOL;
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : UDINT;
    END_VAR

    VAR //объявление локальных переменных
        CTN_1: SYS.CTN;
    END_VAR

    //область кода

    CTN_1(U := U_in, D := D_in, R := R_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4.3 Инкрементный счетчик (SYS.CTU)

Инкрементный счетчик (SYS.CTU) используется для подсчета числа импульсов, приходящих на вход **C**. На выходе **Q** счетчика появится импульс сигнала логической «1», если число приходящих на вход импульсов достигнет установленного значения на входе **N** (**N** – уставка).

Обозначение	Тип данных	Описание
Входы		
U	BOOL	Вход для прямого счета
R	BOOL	Сброс состояния счетчика в 0
N	UDINT	Уставка
Выходы		
Q	BOOL	Сигнализация достигнутой уставки

```
FUNCTION_BLOCK CTU_counter //имя функционального блока

    VAR_INPUT //объявление входных переменных
        C_in : BOOL;
        R_in : BOOL;
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //объявление выходных переменных
        Q_out : BOOL;
    END_VAR

    VAR //объявление локальных переменных
        CTU_1: SYS.CTU;
    END_VAR

    //область кода

    CTU_1(C := C_in, R := R_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```

ЦИФРОВЫЕ
РЕШЕНИЯ

ООО "Овен Цифровые решения"
Россия, г. Москва, пл. Семёновская, д. 1А, помещ. 3/1
support@owendigital.ru
www.owendigital.ru

рег.:1-RU-37197-2.10