



ЭЛКУБ

Электроника контроля,
управления, безопасности

Программное обеспечение

ЭЛКУБ-SCADA

Руководство пользователя

ООО «НПФ «ЭЛКУБ»

Новосибирск, 2025

Содержание

Введение	4
Общие сведения	4
Назначение.....	4
Архитектура.....	5
Функции сервера.....	6
Функции клиента.....	7
Работа через браузер.....	7
Основные сведения	7
Пользователи и группы.....	7
Администратор.....	8
Редактор.....	8
Диспетчер.....	8
Наблюдатель.....	8
Элементы основного окна.....	8
Основное меню.....	8
Базовые понятия.....	10
Мнемосхема.....	11
Примитив.....	11
Элемент.....	11
Пользовательский элемент.....	12
Источник данных.....	12
Регистры.....	12
Виртуальные переменные.....	12
Скрипт.....	13
Журнал событий.....	13
Основные инструменты.....	14
Редактор мнемосхем.....	14
Редактор элементов.....	14
Редактор кода.....	15
Редактор источников данных.....	15
Режимы работы.....	16
Режим редактирования.....	16
Режим отладки.....	16
Режим наблюдения.....	17
Работа с журналом проекта.....	18
Работа с проектами	20
Запуск клиента.....	20
Авторизация.....	21
Управление пользователями.....	21

Описание встроенных групп пользователей	21
Создание пользователя	23
Изменение существующего пользователя	24
Удаление пользователя	25
Управление проектами	25
Открытие или создание нового проекта.....	25
Дерево проекта	28
Панель свойств	31
Панель отладки.....	32
Компиляция и отладка проекта.....	33
Запуск проекта в работу.....	36
Просмотр проекта	38
Переход в режим наблюдения.....	38
Масштабирование и перемещение	39
Создание проекта	41
Создание пользовательских элементов.....	41
Редактор элементов	41
Использование базовых примитивов	46
Прямоугольник.....	56
Круг.....	57
Текст	58
SVG Изображение	59
Предупреждающее сообщение	64
Датчик заштыбовки	64
Конвейер	65
Насос.....	66
Двигатель.....	67
Создание анимации.....	68
Преобразование элементов.....	68
Добавление всплывающего окна для пользовательских элементов	68
Создание мнемосхем.....	72
Редактор мнемосхем	72
Свойства мнемосхемы	72
Базовые примитивы.....	73
Стандартные элементы	73
Пользовательские элементы	73
Изменение свойств элементов	74
Редактирование элементов на мнемосхеме	75
Подключение к источникам данных	76
Создание и настройка источника данных.....	79
Типы источников данных.....	79

Добавление нового источника данных.....	80
Добавление устройства в источник данных типа Modbus	80
Редактор кода	80
Общие принципы	81
Регистры.....	81
Виртуальные переменные	82
Скрипты	83
Синтаксис	83
Список команд скриптов для управления логикой отображения	83
Общие свойства примитивов и элементов	83
Специфичные для конкретных примитивов свойства	84
Пример использования	89
Работа в режиме наблюдения (на примере проекта автоматизации конвейерной линии).....	97
Запуск клиента	97
Цветовая схема индикации состояний	98
Мониторинг состояния конвейера.....	99
Индикация режима работы - местный, ремонтный, дистанционный	101
Индикация состояния конвейера на мнемосхеме	101
Индикация наличия напряжения	102
Индикация состояния блокировок по дискретным входам	103
Индикация скорости ленты	103
Аварийное состояние	104
Индикация пересыпа	104
Индикация состояния двигателей конвейера	105
Индикация состояния контроллера конвейера.....	105
Информационная табличка	106
Окно вывода состояния конвейера	107

Введение

Настоящее Руководство пользователя включает в себя сведения, необходимые для ознакомления с программным обеспечением ЭЛКУБ-SCADA (далее ПО) компании ООО «НПФ «ЭЛКУБ».

Руководство предназначено для сотрудников технических служб предприятий, эксплуатирующих ПО, инженеров-проектировщиков, операторов. В Руководстве приведены описание системы, ее характеристики, рекомендации по использованию.

Эксплуатация ПО должна проводиться лицами, ознакомленными с настоящим Руководством, имеющие образование не ниже среднего и соответствующий опыт работы.

Общие сведения

Назначение

ПО предназначено для построения систем мониторинга и управления различными технологическими процессами в режиме реального времени.

Преимуществом ПО являются простота и удобство для конечного пользователя, отсутствие перегруженности элементов управления, ориентация на горнодобывающую промышленность, а также поддержка специфического оборудования и систем.

ПО имеет современную кроссплатформенную универсальную клиент-серверную архитектуру, что позволяет легко создавать новый функционал, ориентируясь на требования заказчика.

Работа клиента возможна в операционных системах Windows и Linux. Также возможна полноценная работа в веб-браузере.

Характеристики

ПО имеет следующие характеристики:

- поддержка протоколов Modbus TCP, Modbus RTU over UDP, Modbus RTU over UDP;
- наличие интеграции с оборудованием компании ООО НПФ «Элкуб»;
- возможность интеграции с оборудованием сторонних производителей (при наличии описания протоколов связи);
- встроенный редактор экранных форм (мнемосхем);
- возможность импорта векторных графических изображений в формате svg;
- палитра готовых технологических изображений,
- возможность задания динамики поведения графических объектов в зависимости от заданных переменных: цвет, фон, положение, размер, угол вращения, видимость и др.,
- автоматическое масштабирование мнемосхем под текущее разрешение монитора;
- наличие механизма шаблонов для легкой конфигурации и изменения сети АСУ ТП;
- возможность создания скриптов обработки данных (глобальные или привязанные к объектам), выполняемые по событиям, по таймеру или по расписанию;
- система регистрации операторов и уровни доступа, защита от несанкционированного доступа, разделение пользователей по группам с различными полномочиями;
- журналирование в отдельную базу данных (данные оборудования, аварии, события, действия пользователей);

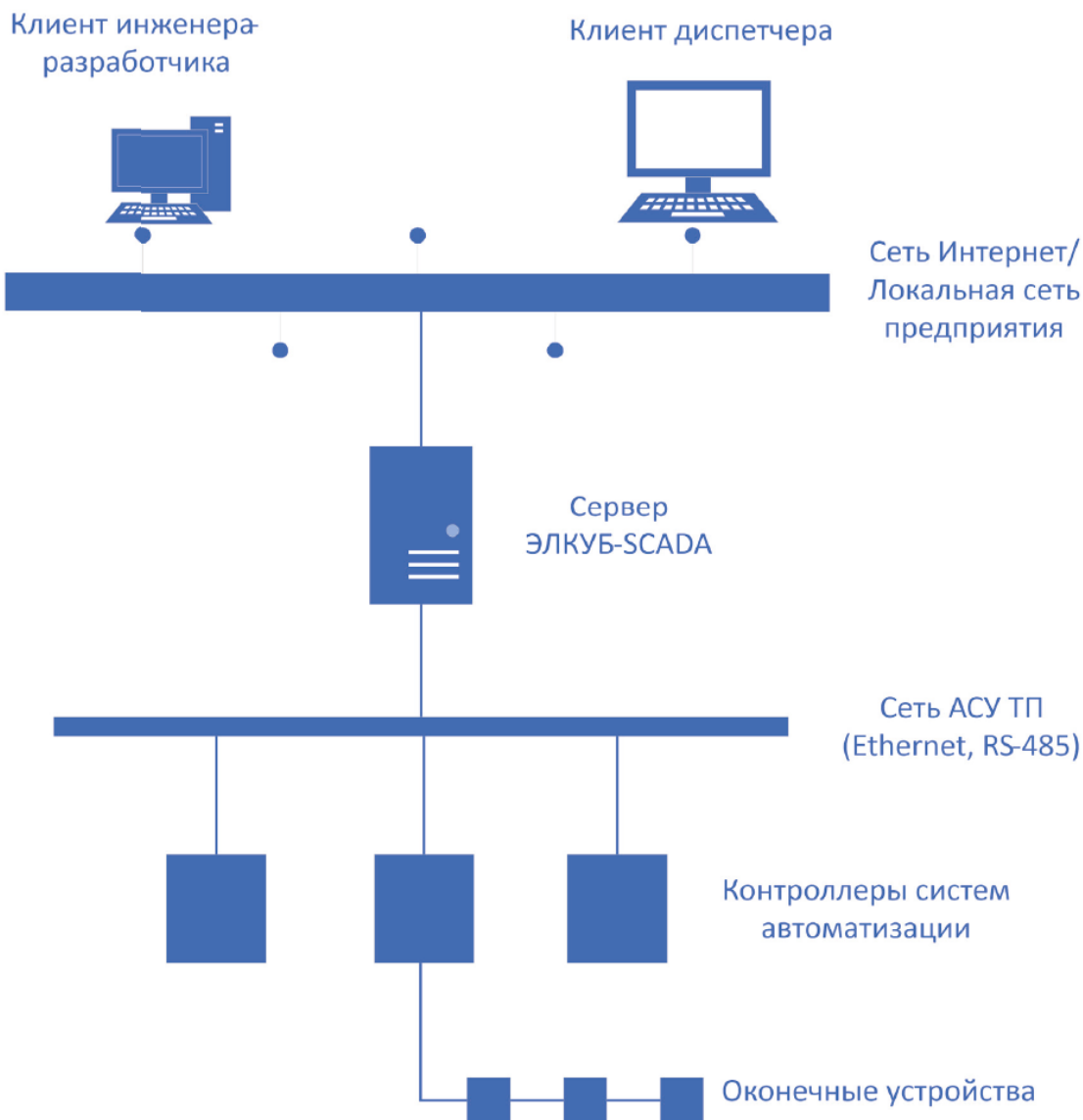
Архитектура

ПО ЭЛКУБ-SCADA разработано на основе клиент-серверной архитектуры, что позволяет удешевить стоимость развертывания и получить возможность гибкого расширения системы для наращивания функциональности. Разделение единого приложения на клиентскую и серверную часть позволяет:

- уменьшить нагрузку на локальную сеть предприятия;

- уменьшить частоту опроса контроллеров и нагрузку на сеть АСУ ТП;
- упростить процедуры поддержки обновления системы;
- строить резервируемые системы с несколькими серверами.

Структура подключения оборудования приведена на рисунке ниже.



Функции сервера

- получение данных из сети АСУ ТП предприятия;
- опрос контроллеров, выполнение обработки сырых данных;
- выполнение скриптов автоматизации;
- обработка и сохранение данных в базе данных;
- работа с клиентами (API).

Функции клиента

Основной функцией клиента является предоставление графического интерфейса пользователя для просмотра и создания проектов автоматизации.

Работа через браузер

Работа через веб-браузер происходит без ограничений, возможны как просмотр мнемосхем, так и создание полноценных проектов автоматизации.

Основные сведения

Пользователи и группы

Список групп пользователей является фиксированным и приведен в таблице ниже.

Группа	Описание
Администратор	Полный доступ, в том числе возможность добавления и удаления пользователей.
Редактор	Полный доступ, в том числе к редактированию и запуску проектов в работу, но без возможности добавления и удаления пользователей.
Диспетчер	Доступ только к просмотру мнемосхем в режиме наблюдения, без возможности редактирования; возможность управления конвейерами и квитирования сообщений в журнале.
Наблюдатель	Только просмотр мнемосхем в режиме наблюдения без возможности управления.

Администратор

Полный доступ, в том числе возможность добавления и удаления пользователей.

Редактор

Полный доступ, в том числе к редактированию и запуску проектов в работу, но без возможности добавления и удаления пользователей.

Диспетчер

Доступ только к просмотру мнемосхем в режиме наблюдения, без возможности редактирования; возможность управления конвейерами и квитирования сообщений в журнале.

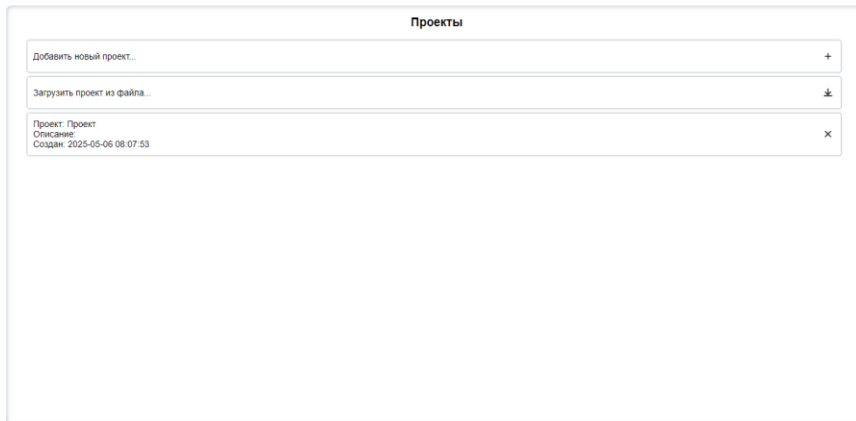
Наблюдатель

Только просмотр мнемосхем в режиме наблюдения без возможности управления.

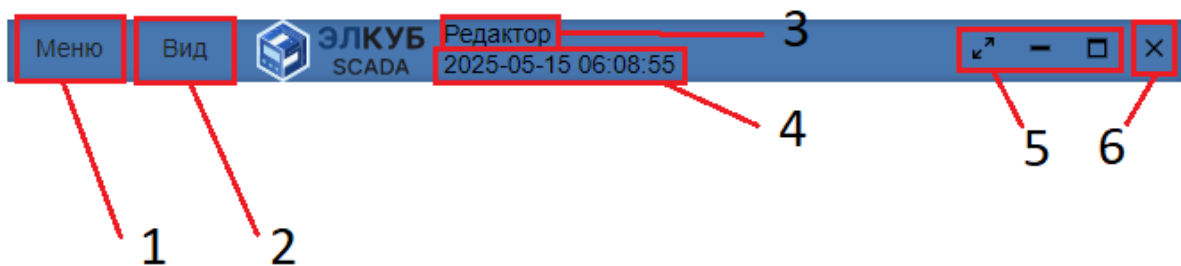
Элементы основного окна

Основное меню

При успешной авторизации пользователь попадает на страницу выбора проектов, доступных ему.



В верхней части приложения расположены общие элементы, такие как выпадающее меню навигации “Меню” (1), меню параметров отображения приложения “Вид” (2), имя авторизовавшегося пользователя (3), текущее время на сервере (4), опции для управления размерами и положением окна (5), кнопка закрытия окна (6).

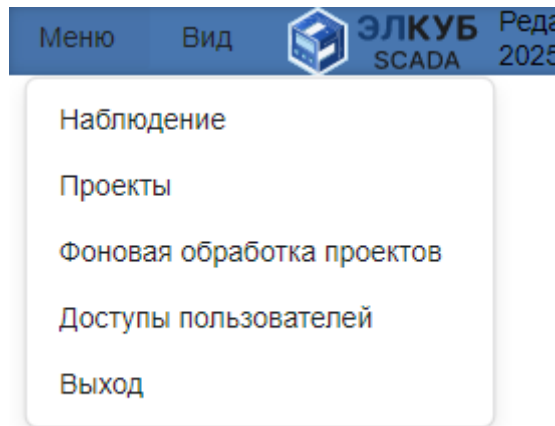


Меню навигации “Меню” содержит следующие пункты:

Наблюдение - перейти в режим наблюдения

Проекты - перейти на страницу выбора проектов

Выход - перейти на вкладку авторизации, выйти из аккаунта

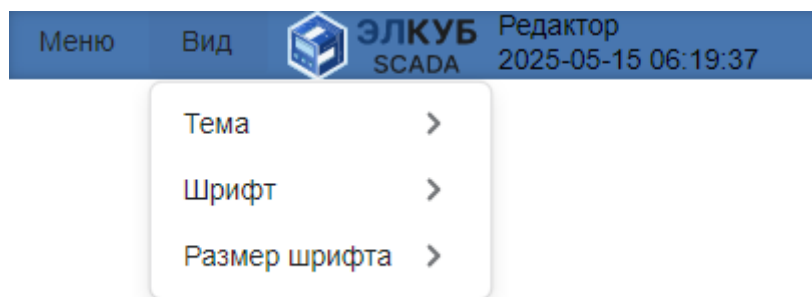


Меню параметров отображения “Вид” содержит следующие пункты:

Тема - выбор текущей темы (тёмная/светлая/системная)

Шрифт - задать значение общего шрифта текста в приложении

Размер шрифта - задать значение общего размера текста в приложении



Базовые понятия

При работе с ПО ЭЛКУБ-SCADA используются следующие базовые понятия:

- Мнемосхема
- Примитив
- Элемент
- Пользовательский элемент
- Источник данных
- Регистры
- Виртуальные переменные
- Скрипты

Мнемосхема

Базовая функциональность склада-системы, предназначена для отображения технологического процесса и управления им.

Примитив

Примитивы предназначены для создания мнемосхем и элементов и представляют собой элементы с базовой функциональностью:

- Круг
- Квадрат
- Текст
- SVG изображение
- PNG изображение
- Таблица
- График
- Кнопка
- Поле ввода
- Предупреждающее сообщение
- Датчик заштыбовки
- Конвейер
- Дробилка
- Лавный комбайн
- Насос

Элемент

Элемент представляет собой сущность, объединяющую графические примитивы, источники данных, регистры, виртуальные переменные и скрипты, которые обеспечивают функциональность устройства.

При добавлении элемента автоматически добавляются все связанные с ним, виртуальные переменные и скрипты, необходимые для его работы.

После добавления элемента на мнемосхему его необходимо привязать к источнику данных

Пользовательский элемент

Пользователи могут создавать свои элементы и использовать их в качестве шаблонов. Шаблон устройства представляет собой набор графических примитивов, источников данных, виртуальных переменных и скриптов, которые обеспечивают функциональность устройства. После создания пользовательского элемента он автоматически добавляется в дерево проекта и в список доступных элементов в редакторе мнемосхем.

Источник данных

Источники данных представляют собой основную сущность, которая поставляет данные из разнородных устройств в скада-систему в едином формате.

Для источника данных типа Modbus поставляемыми данными являются значения регистров. В разделе создания источника данных можно просматривать и задавать адреса регистров и параметры опроса. По результату опроса регистров в системе появляются значения (регистры) можно использовать:

- для создания виртуальных переменных
- в скриптах устройств

Регистры

Регистры напрямую связаны с опрашиваемым физическим устройством. Для конфигурирования списка опрашиваемых регистров для элемента необходимо точно знать карту регистров конкретного устройства, которое будет связано с отображаемым элементом. После конфигурирования регистров у элемента, их значения будут обновляться в режиме наблюдения согласно приходящим от устройства значениям этих регистров.

Виртуальные переменные

Виртуальные переменные представляют собой унифицированные переменные, задаваемые пользователем, которые можно легко использовать при создании мнемосхем. Могут быть заданы для всего проекта, для мнемосхемы, для пользовательского элемента, для примитива.

Виртуальные переменные принадлежат тому объекту, в котором они прописаны. То есть виртуальные переменные элемента принадлежат этому конкретному элементу, соответственно виртуальная переменная с таким же названием у другого элемента будет являться отдельной переменной, никак не связанной с переменной у первого элемента.

С виртуальными переменными связан код, который будет выполняться для обновления значения этой виртуальной переменной. Результат return в коде переменной будет присвоен как значение этой переменной. В коде виртуальной переменной могут быть использованы виртуальные переменные того же объекта, в котором используется виртуальная переменная, или выше по иерархии объектов (проект <- мнемосхема <- пользовательский элемент <- примитив) через обращение вида parent.VAR().Value, либо у любого объекта проекта через уникальный псевдоним.

Виртуальные переменные могут быть как созданными вручную, так и сгенерированными автоматически на этапе редактирования элемента. Это связано с особой логикой работы некоторых примитивов или элементов. Например, примитив Кнопка обладает автогенерируемой переменной, отвечающей за нажатие по кнопке. Автогенерируемые переменные нельзя удалить или отредактировать. Автогенерируемые переменные нужно использовать для обработки соответствующих им событий.

Скрипт

Для каждого устройства можно прописать набор скриптов для управления отображением примитивов и пользовательских элементов. В скрипте могут быть использованы виртуальные переменные того же объекта, в котором используется скрипт, или выше по иерархии объектов (проект <- мнемосхема <- пользовательский элемент <- примитив), либо у любого объекта проекта через уникальный псевдоним. Вызов скриптов происходит только при изменении виртуальных переменных, от которых зависит скрипт.

Журнал событий

Обеспечивает накопление и просмотр информации о произошедших событиях. Является единым для всего проекта.

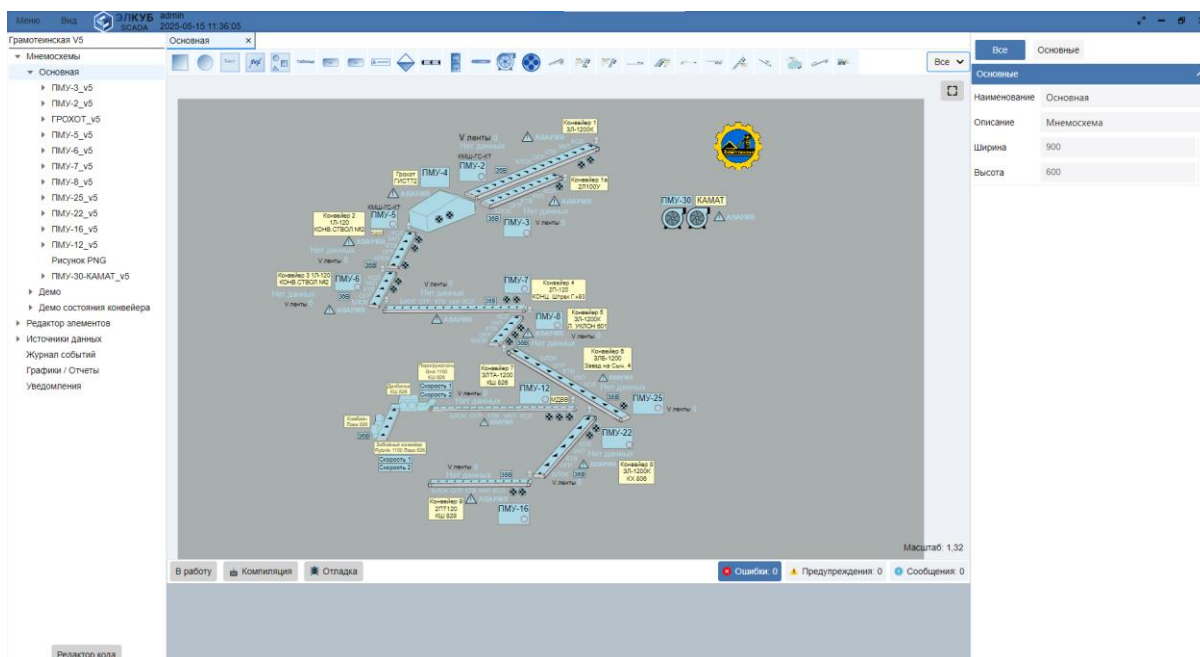
Регистрируемые события:

- аварии
- события,
- действия пользователей

Основные инструменты

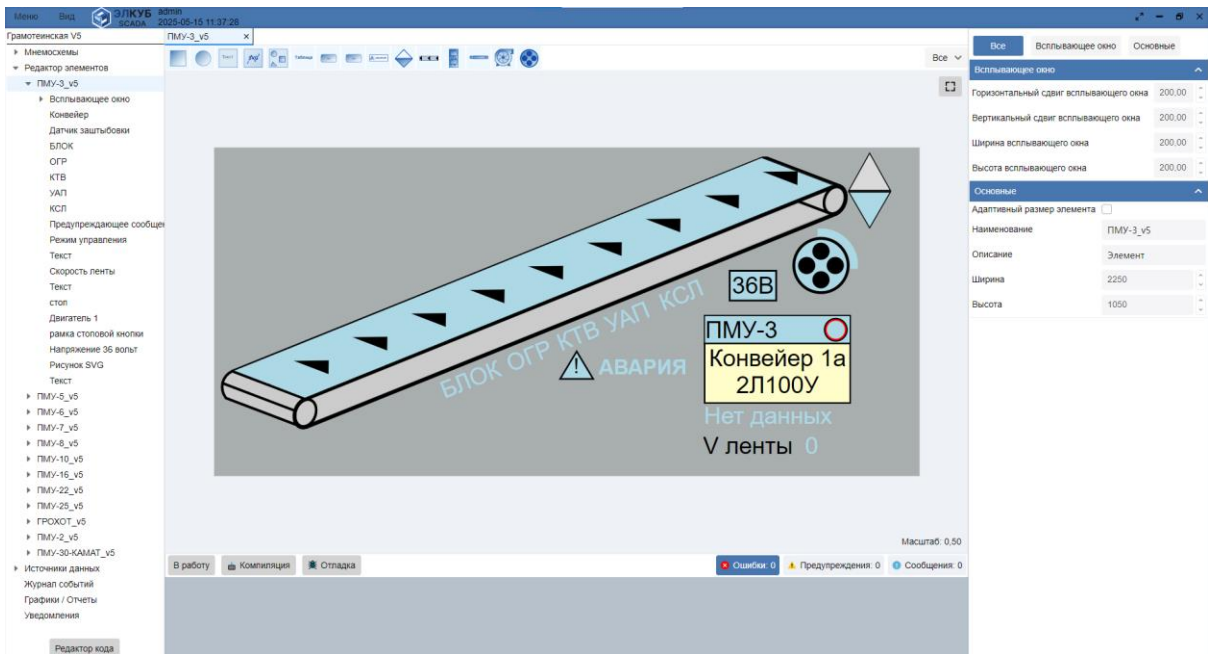
Редактор мнемосхем

Редактор мнемосхемы предназначен для создания схемы отображения всех примитивов и элементов, которые будут наблюдаться.



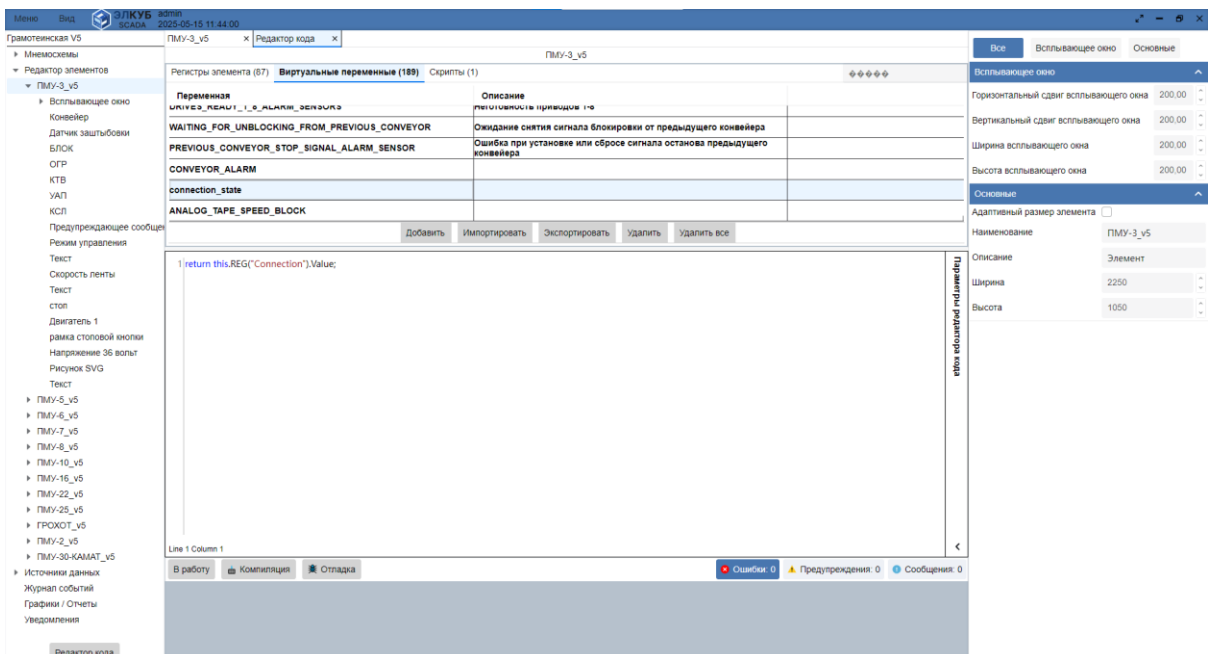
Редактор элементов

Редактор элементов нужен для создания шаблонов пользовательских элементов, которые будут отображать поведение и состояние физических устройств на мнемосхеме.



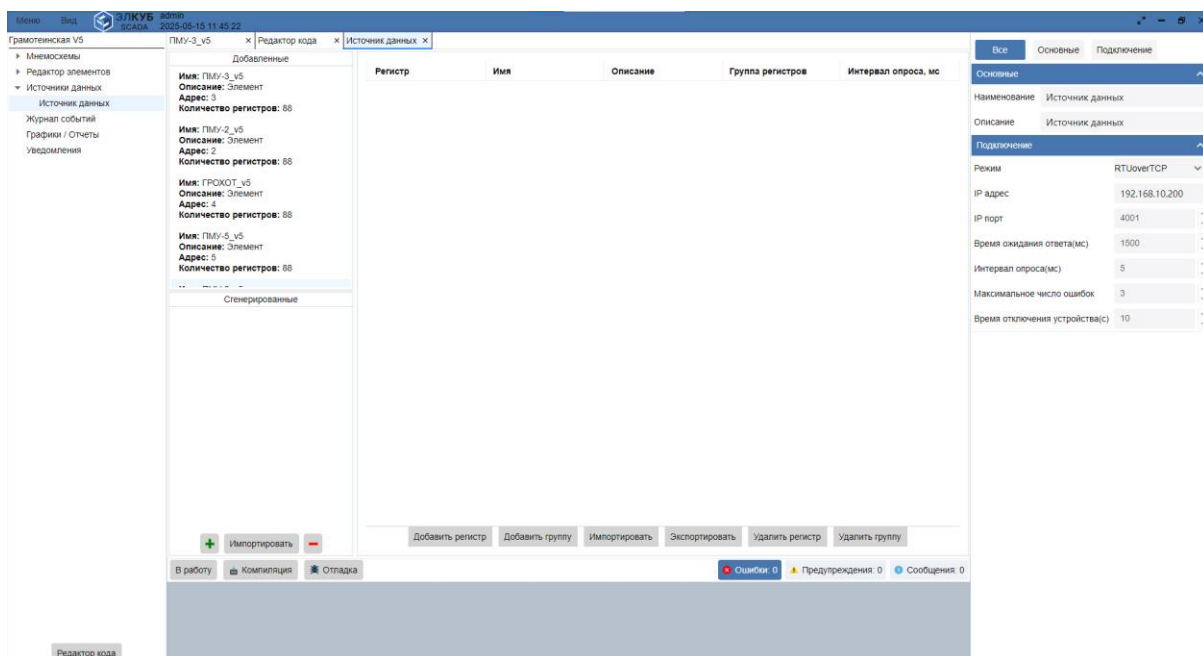
Редактор кода

Редактор кода позволяет создать и описать опрашиваемые регистры, виртуальные переменные и скрипты.



Редактор источников данных

В редакторе источников данных задаются параметры подключения к физической линии, а также добавляются устройства к источнику данных и задаются адреса устройств.



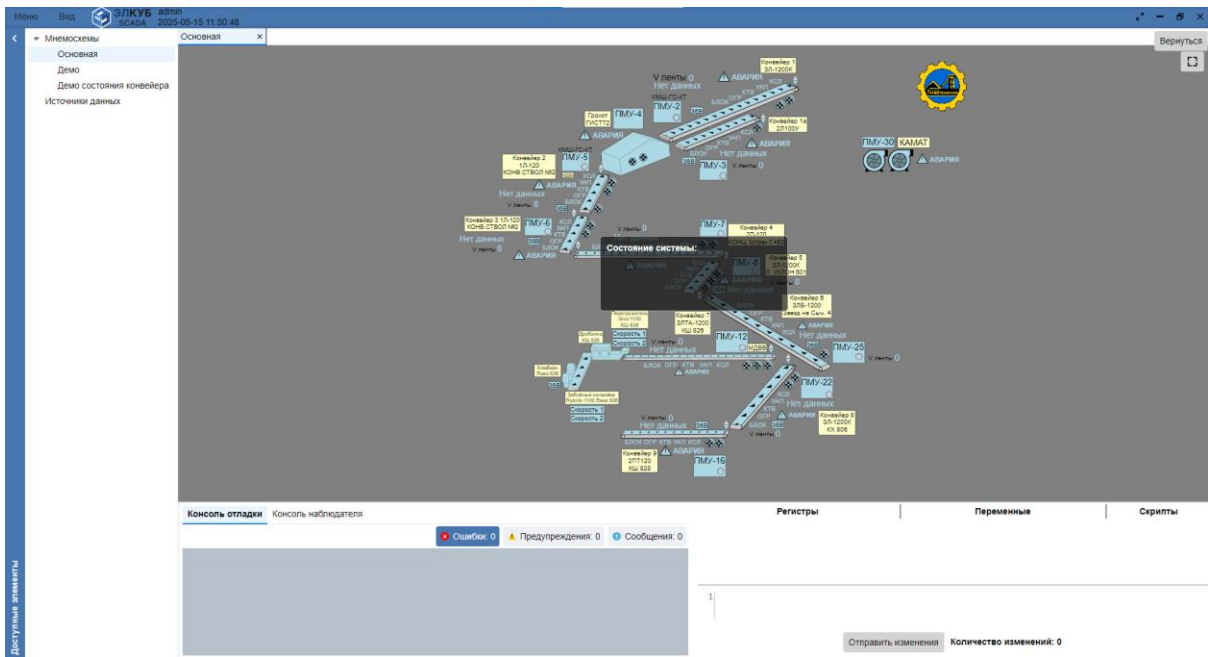
Режимы работы

Режим редактирования

Режим редактирования предназначен для создания мнемосхем, которые будут работать в режиме наблюдения. В этом режиме создаются все пользовательские элементы, описываются виртуальные переменные, скрипты, создаются источники данных, мнемосхемы.

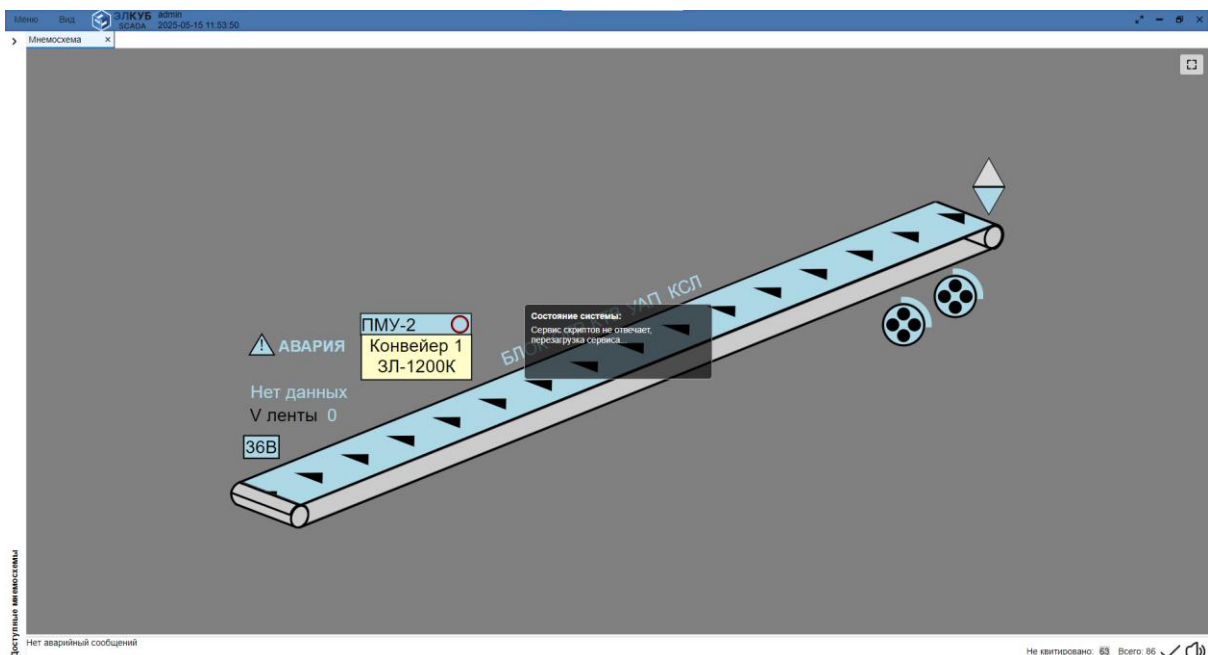
Режим отладки

Режим отладки используется, чтобы проверить разработанные элементы, скрипты, виртуальные переменные без подключения к физическому устройству. В этом режиме происходит полная сборка всего проекта, как если бы он работал в наблюдении, выполняется компиляция кода скриптов и виртуальных переменных, отображаются ошибки при компиляции. В режиме отладки можно вручную задавать значения для регистров и виртуальных переменных, тем самым тестируя разработанные алгоритмы.



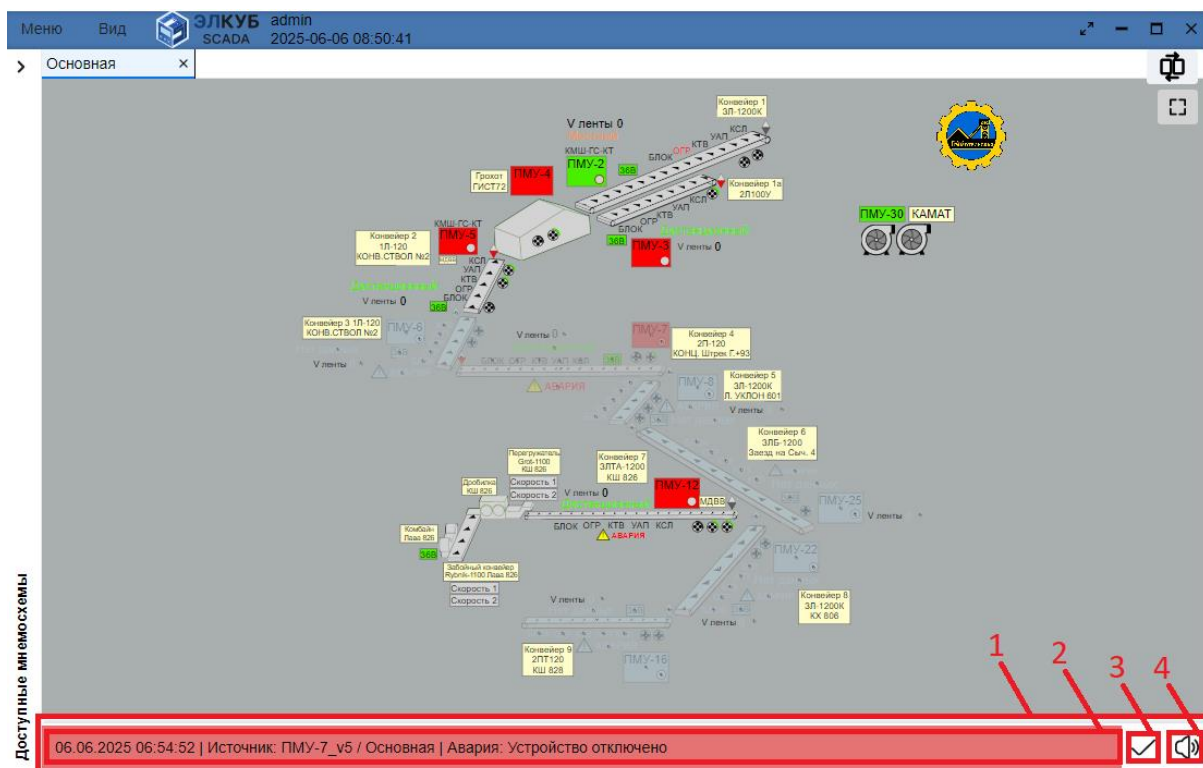
Режим наблюдения

Этот режим предназначен для наблюдения за физическими устройствами после создания в редакторе всех необходимых мнемосхем и алгоритмов поведения элементов отображения на них. В этом режиме нет возможности отредактировать отображаемые элементы.



Работа с журналом проекта

Для логгирования информации о происходящих на мнемосхеме изменениях (изменение состояния конвейера, запуск двигателя, сигнал от КСЛ-1, и т.д.) используется журнал, общий для всех мнемосхем в проекте. Журнал отображается только в режиме наблюдения. Он расположен под наблюдаемыми мнемосхемами(1).



В свёрнутом состоянии журнал отображает последнее не квитированное аварийное сообщение (2), кнопка квитирования последнего сообщения (3), отключение звукового оповещения (4).

При нажатии на область свёрнутого журнала, он расширится для отображения полного списка журналированных сообщений и дополнительных операций с журналом.

№	Время	Тип	Источник	Сообщение	Комментарий к вкв
16	2025-06-06 03:54:52.881	Авария	ПМУ-1_v5 / Основная	Устройство отключено	
17	2025-06-06 05:57:46.560	Инфо	ПМУ-2_v5 / Основная	Пользователь admin: нажата кнопка перехода в дистанционный режим управления	
18	2025-06-06 05:57:57.802	Инфо	ПМУ-2_v5 / Основная	Режим работы переключен на Дистанционный	
19	2025-06-06 05:58:12.329	Инфо	ПМУ-2_v5 / Основная	Пользователь admin: нажата кнопка сброса аварий	
20	2025-06-06 05:58:14.327	Инфо	ПМУ-2_v5 / Основная	Сброс аварии: ОГРАЖДЕНИЕ 1	
21	2025-06-06 05:58:14.458	Инфо	ПМУ-30-КАМАТ_v5 / Основ	Сброс аварии: ОГРАЖДЕНИЕ 1	
22	2025-06-06 05:58:21.473	Инфо	ПМУ-2_v5 / Основная	Пользователь admin: нажата кнопка запуска конвейера	
23	2025-06-06 05:58:30.731	Предупреж	ПМУ-2_v5 / Основная	Конвейер запускается	
24	2025-06-06 05:59:03.496	Предупреж	ПМУ-2_v5 / Основная	Конвейер запущен и работает	

Над таблицей со всеми сообщениями журнала находится горизонтальная панель опций фильтрации отображаемых сообщений. К журналу можно применить следующий набор фильтров:

Приоритет (1) - отфильтровать сообщения по приоритету. Приоритет может принимать три значения: Низкий, Средний, Высокий.

Тип сообщения (2) - отфильтровать сообщения по типу. Тип сообщения может принимать произвольные значения в зависимости от того, что передал пользователь при вызове команды логгирования в скриптах. В проекте, представленном на рисунке, используется три типа сообщений: Авария для приоритета 2, Предупреждение для приоритета 1, Инфо для приоритета 0.

Источник сообщения (3) - отфильтровать сообщения по источнику сообщения. В этом выпадающем списке отображаются все доступные в проекте уникальные элементы, от которых приходят сообщения в журнал.

Не квитированные (4) - отображать только не квитированные сообщения.

Временной промежуток (5) - отображать только сообщения из этого промежутка времени.

Под таблицей находятся кнопки перемещения по страницам журнала (6), значения общего количества сообщений в журнале (7), неквитированных сообщений (8), а также кнопка для квитирования всех сообщений (9).

Таблица сообщений журнала содержит множество столбцов: Номер сообщения(1) в текущей открытой странице журнала. Время прихода сообщения (2). Тип сообщения (3). Элемент-источник сообщения (4). Содержимое сообщения (5). Комментарий, оставляемый квитирующим специалистом перед квитированием (6), при нажатии на ячейку комментария квитирования в таблице его можно ввести. Флаг состояния квитирования сообщения (7), при нажатии на него происходит квитирование сообщения. Время квитирования сообщения (8), изначально пустое, устанавливается автоматически при квитировании сообщения. Пользователь, квитировавший это сообщение (9).

№	Время	Тип	Источник	Сообщение
30	2025-05-30 03:06:13.460	Авария	ПМУ-2_v5 / Основная	Устройство отключено

Комментарий квитирования	Квитирование	Время квитирования	Квитировал
	<input checked="" type="checkbox"/>	2025-05-30 03:21:58.009	admin

Работа с проектами

Запуск клиента

Работать со SCADA можно либо при помощи специального клиента для Windows, либо в браузере по заранее известным данным сервера SCADA. После установки приложения Elsub.SCADA на своё устройство, нужно запустить его, нажав дважды по ярлыку на рабочем столе. Для работы через браузер нужно открыть любой доступный браузер и указать в строке URL ip адрес сервера.

Авторизация

Для авторизации необходимо получить авторизационные данные у администратора сервера.

При запуске приложения пользователь видит окно авторизации. Для входа нужно ввести данные своей учётной записи в соответствующие поля окна “Логин” и “Пароль”. После ввода нужно нажать кнопку “Войти”. Если данные верны и связь с сервером установлена, авторизация произойдёт успешно. Пользователи с уровнем доступа Наблюдатель и Диспетчер будут перенаправлены на страницу Наблюдения, Редактор и Администратор после авторизации попадают в меню выбора проекта для редактирования и могут перейти в режим наблюдения вручную через меню.



Авторизация

Логин

Пароль

Войти

Подключено

Управление пользователями

Описание встроенных групп пользователей

Всего существует четыре группы-роли пользователей, имеющих различный уровень ограничений на использование SCADA системы.

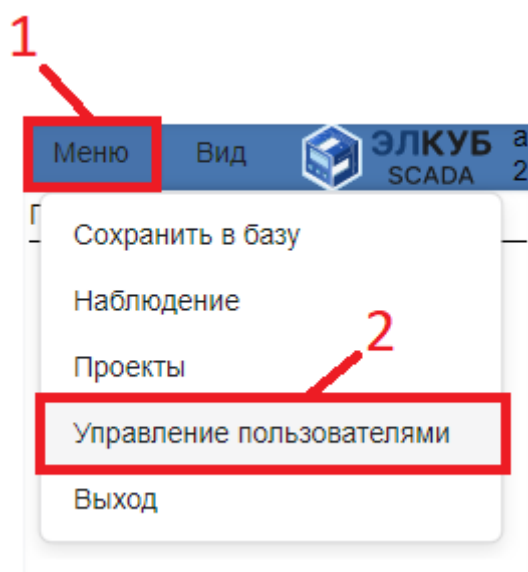
Наблюдатель - базовая роль, имеет доступ к просмотру мнемосхем, находящихся в режиме наблюдения.

Диспетчер - всё, доступное Наблюдателю, с возможностью взаимодействовать с интерактивными элементами на мнемосхеме в наблюдении (например, с кнопками во всплывающем окне).

Редактор - всё, доступное Диспетчеру, с возможностью открывать проекты в режиме редактирования, вносить правки, обновлять проект в наблюдении.

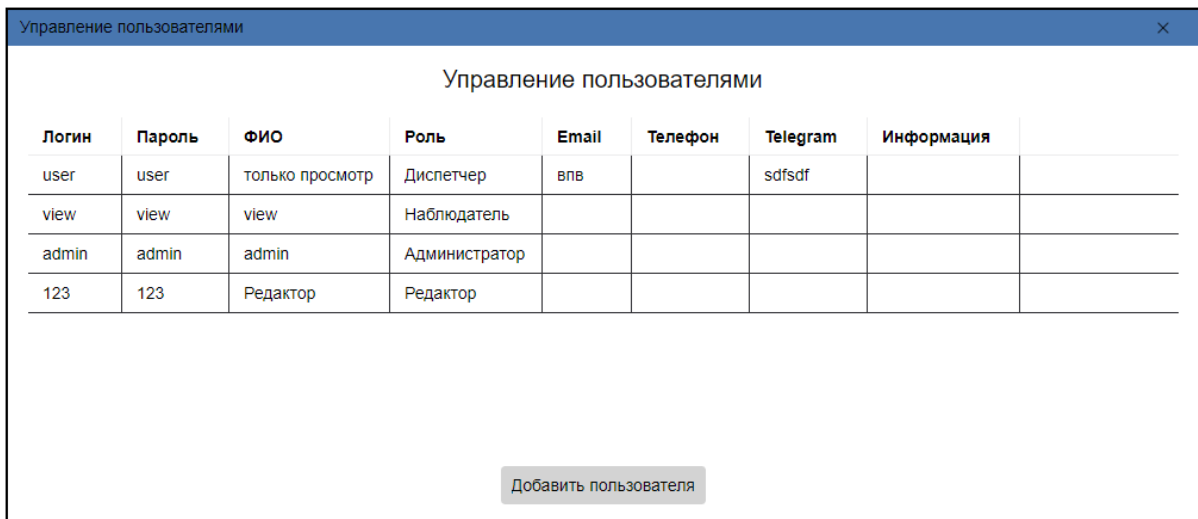
Администратор - полный доступ, возможность редактировать данные пользователей.

Для редактирования данных пользователей у Администратора в доступе есть специальная панель управления, доступ к которой можно получить из Меню:



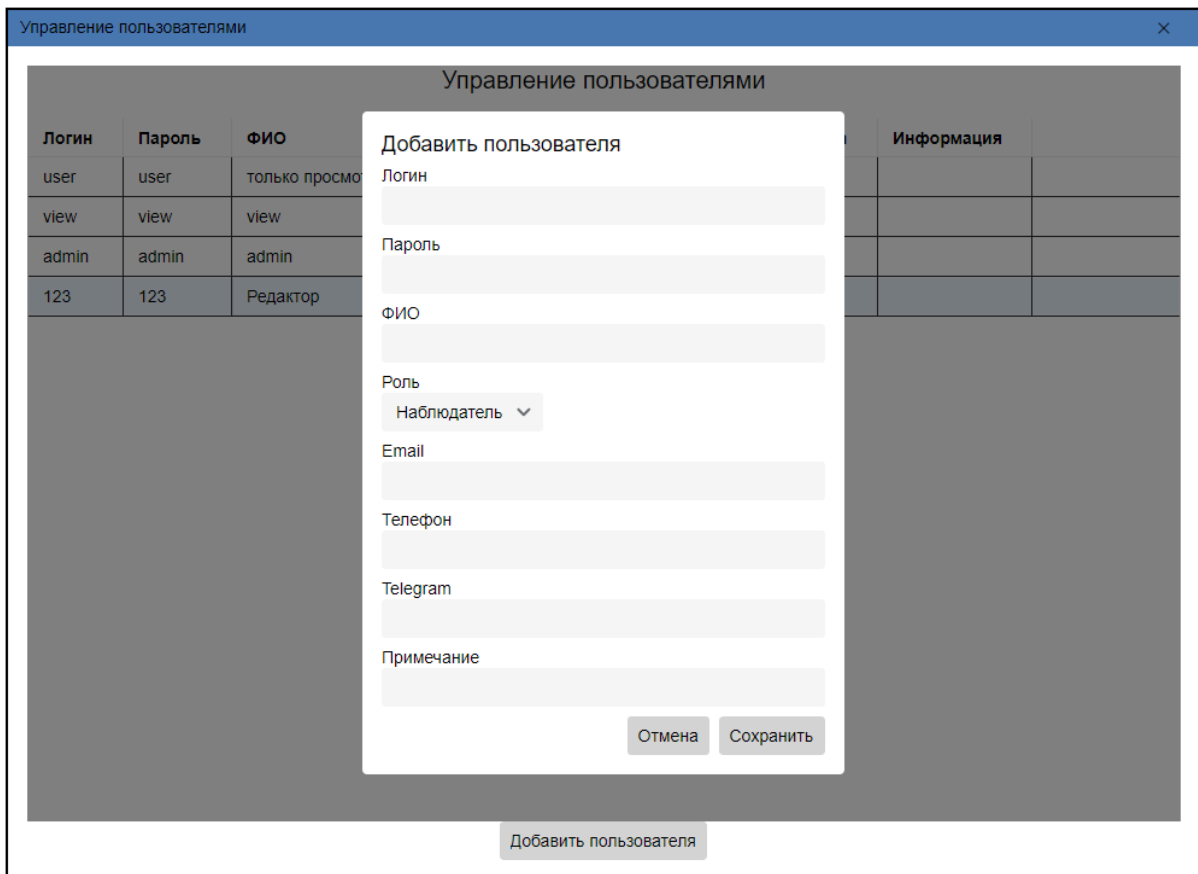
После нажатия на пункт меню “Управление пользователями”, будет открыто меню управления пользователями, в котором отображены все существующие в системе пользователи, их данные авторизации, права, а также дополнительная информация.

Администратор может поменять данные уже существующих пользователей, добавить нового пользователя, удалять пользователей.



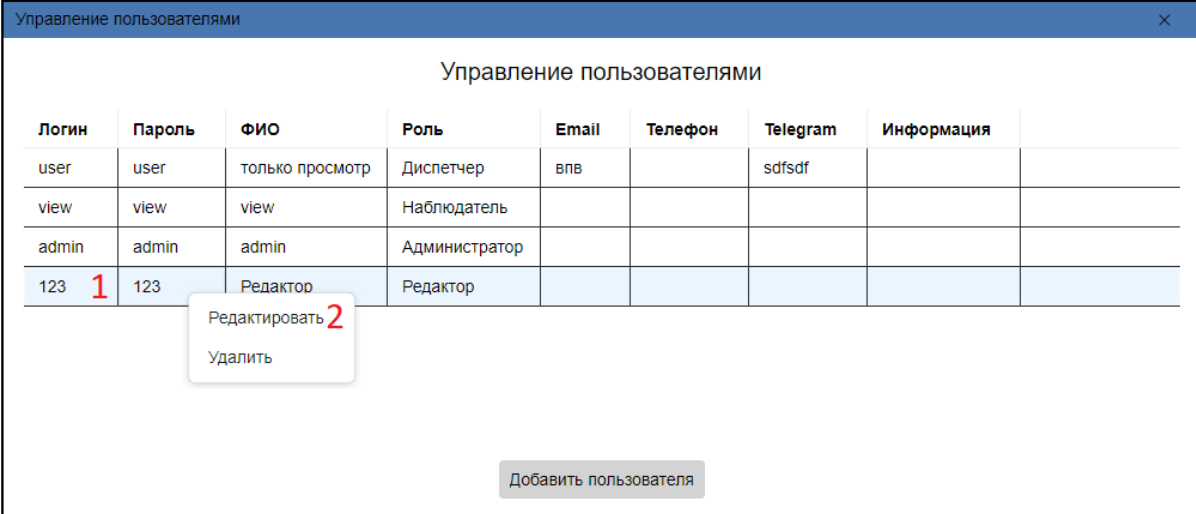
Создание пользователя

Для добавления нового пользователя нужно нажать на кнопку “Добавить пользователя” и в появившейся форме заполнить поля и нажать “Сохранить”. Обязательными для заполнения являются поля “Логин” и “Пароль”.



Изменение существующего пользователя

Для изменения данных нужно нажать ПКМ по строке пользователя (1), данные которого нужно отредактировать, после чего в появившемся меню выбрать “Редактировать” (2).

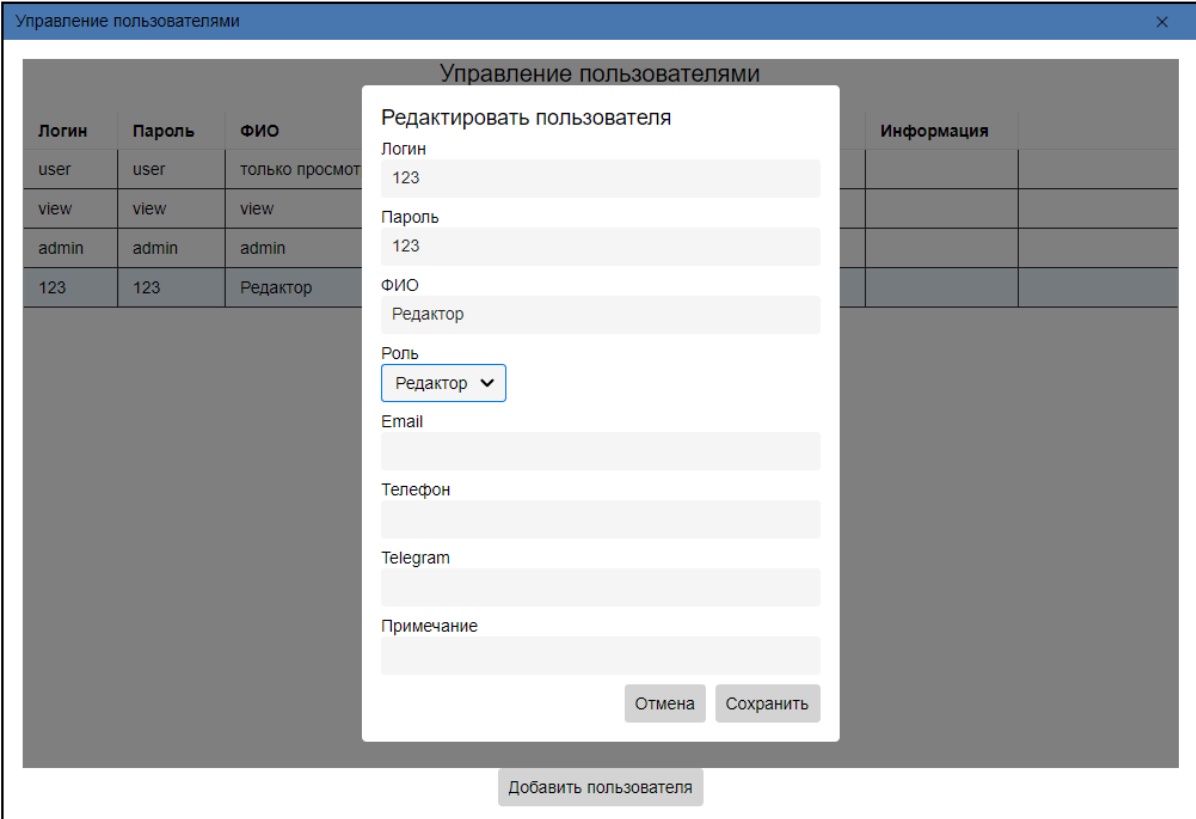


Управление пользователями

Логин	Пароль	ФИО	Роль	Email	Телефон	Telegram	Информация
user	user	только просмотр	Диспетчер	влв		sdfsdf	
view	view	view	Наблюдатель				
admin	admin	admin	Администратор				
123	123	Редактор	Редактор				

Добавить пользователя

Откроется окно для редактирования данных пользователя. Все поля являются текстовыми, Роль пользователя можно выбрать из списка существующих ролей.



Управление пользователями

Логин	Пароль	ФИО	Информация
user	user	только просмотр	
view	view	view	
admin	admin	admin	
123	123	Редактор	

Редактировать пользователя

Логин
123

Пароль
123

ФИО
Редактор

Роль
Редактор

Email

Телефон

Telegram

Примечание

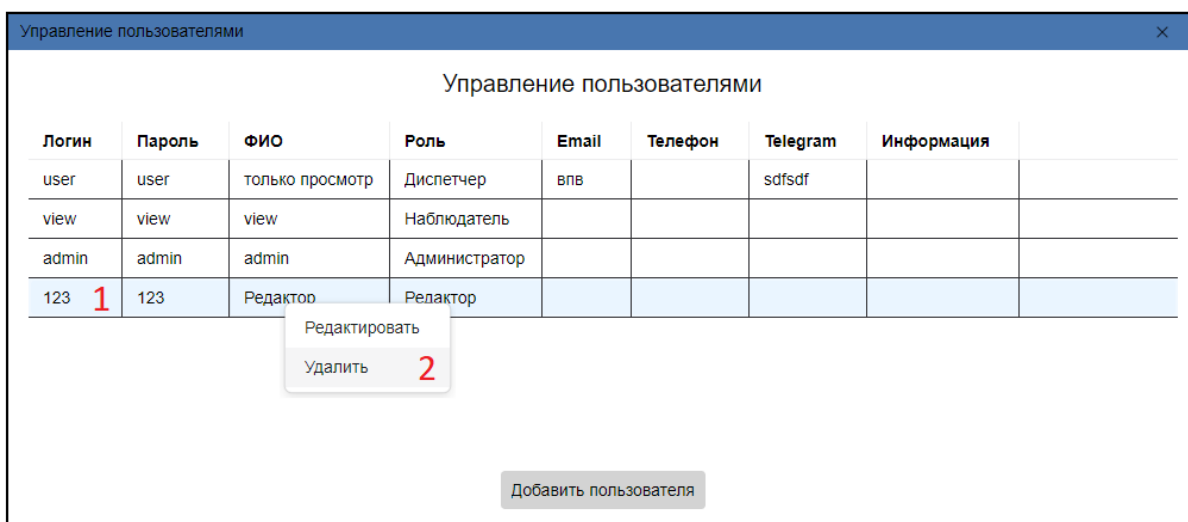
Отмена Сохранить

Добавить пользователя

После редактирования необходимо нажать на кнопку “Сохранить” для применения внесённых изменений. Для отмены редактирования и закрытия окна редактирования данных пользователя необходимо нажать кнопку “Отмена”.

Удаление пользователя

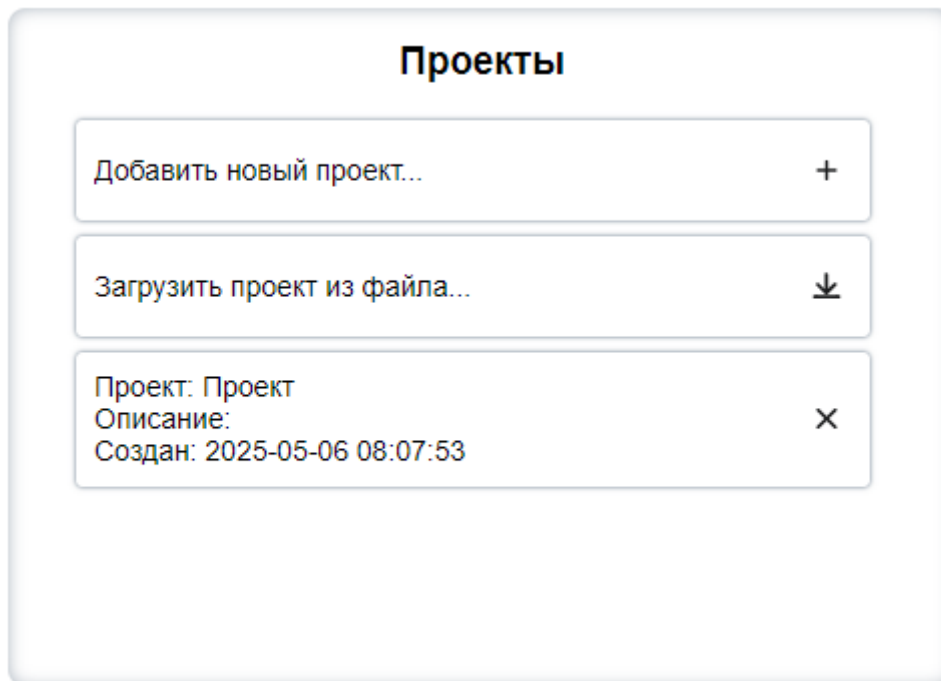
Для удаления пользователя нужно нажать ПКМ по строке с данными пользователя (1), после чего в появившемся меню нажать “Удалить” (2).



Управление проектами

Открытие или создание нового проекта

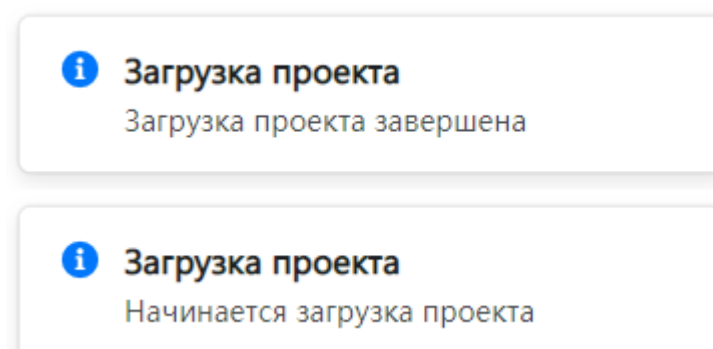
После авторизации в основном меню производится выбор редактируемого проекта.



При нажатии на панель “Добавить новый проект...” будет открыт новый пустой проект, не содержащий мнемосхем, элементов и так далее. При создании нового проекта важно помнить, что он не сохраняется автоматически.

Внимание! Пользователь должен вручную сохранить его, иначе при выходе из приложения или из проекта все данные будут утеряны.

При нажатии на панель “Загрузить проект из файла...” будет открыт браузер файлов для поиска файла проекта на локальном устройстве. При подтверждении будет произведена попытка загрузить проект из файла. При успешной загрузке на экране в правом нижнем углу появятся соответствующие оповещения:



Если загрузить проект из файла не удастся, появятся другие оповещения:



Загрузка проекта

Возникли проблемы при восстановлении файла

Внимание! Важно помнить, что при открытии проекта из файла он не сохраняется автоматически, его необходимо сохранить вручную для последующей работы с ним без повторной загрузки из файла.

Две предыдущие панели, имеют фиксированное положение и присутствуют вне зависимости от пользователя и проектов. Ниже от них идут панели с описанием уже созданных проектов, нажатие на которые приводит к их загрузке.

Проект: Проект

Описание:

Создан: 2025-05-06 08:07:53



При нажатии на крестик в правой части панели проекта произойдёт попытка удалить проект. Появится предупреждающее сообщение для подтверждения удаления проекта:



Удаление проекта



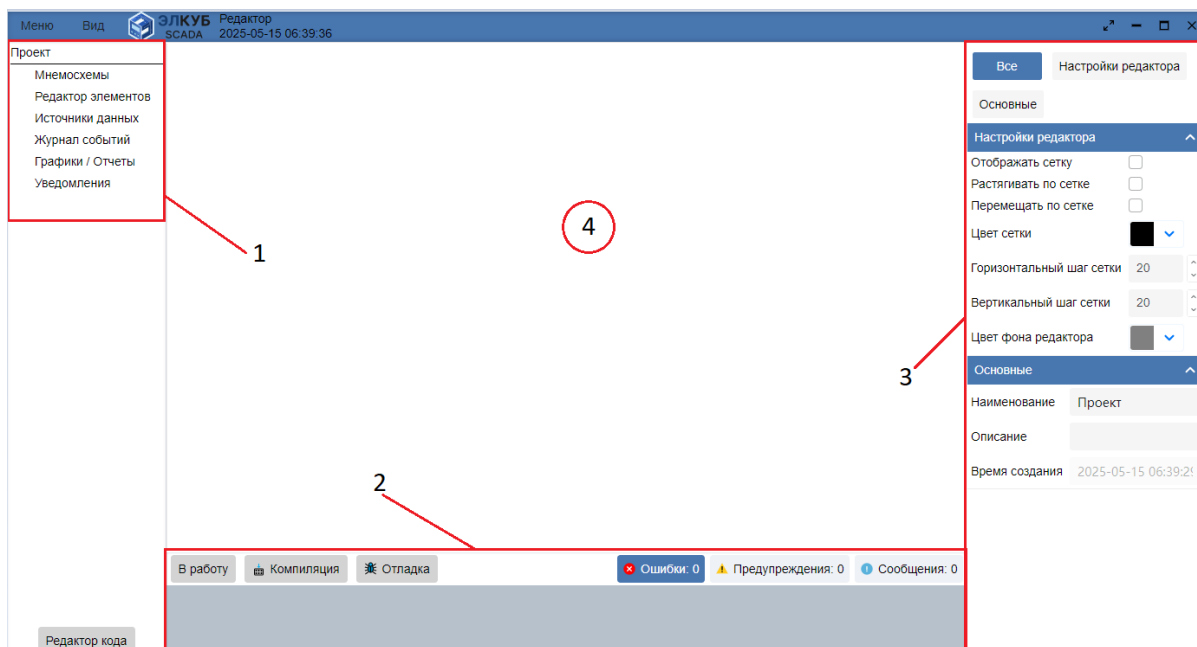
Вы уверены что хотите удалить проект?

No

Yes

При нажатии “Да” проект будет удалён и убран из списка доступных пользователю проектов.

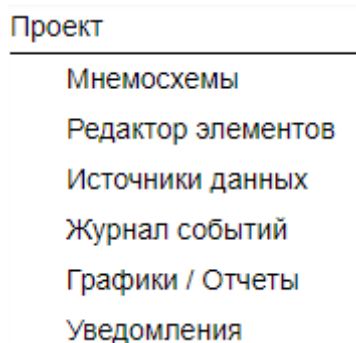
После создания нового проекта пользователь будет видеть следующую страницу:



1 - дерево проекта, 2 - панель отладки, 3 - панель свойств, 4 - рабочая область.

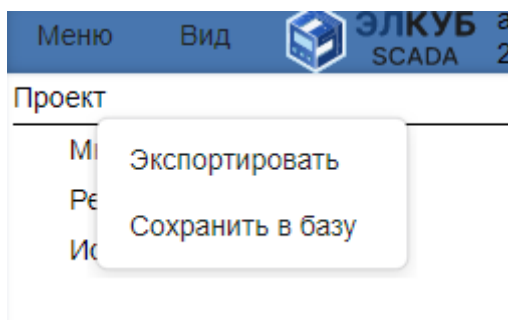
Дерево проекта

Изначально дерево проекта содержит в себе только фиксированный набор корневых элементов.

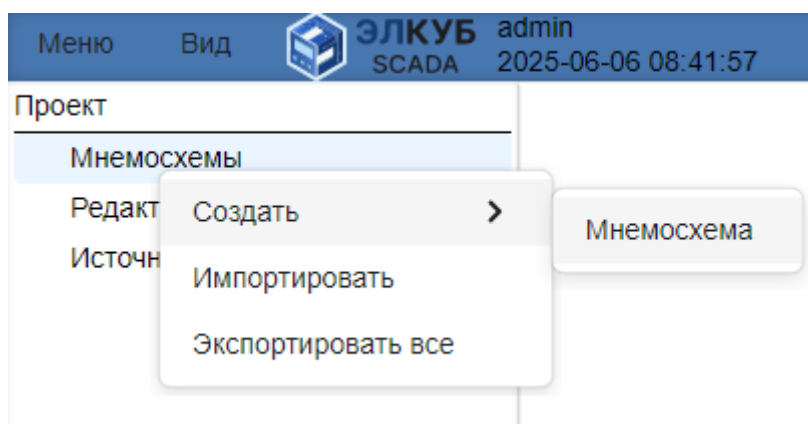


Над чертой находится название открытого проекта.

При нажатии ПКМ по названию проекта открывается меню, содержащее в себе пункты “Экспортировать” - для сохранения в файл на локальном устройстве, “Сохранить в базу” - для сохранения проекта на сервере.



При нажатии правой кнопкой мыши на пункте меню “Мнемосхемы” появляется всплывающее меню:

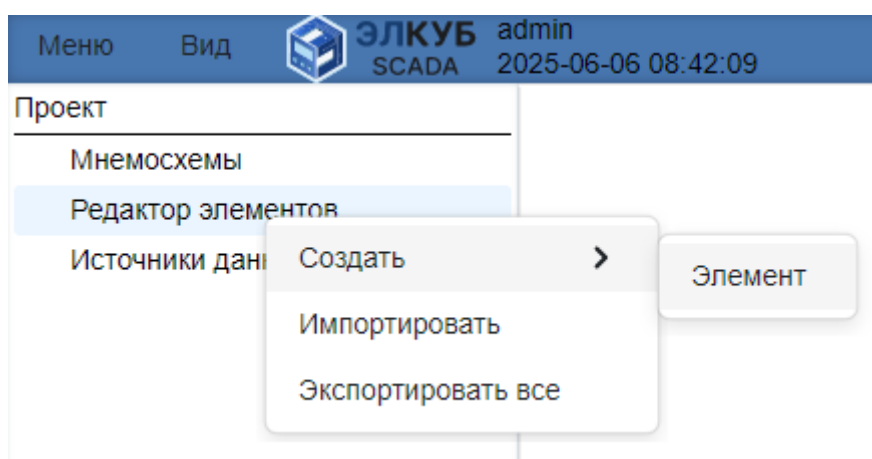


Пункт меню “Создать” позволяет создать новую пустую мнемосхему.

Пункт меню “Импортировать” позволяет выбрать локальный файл сохранённой ранее мнемосхемы и загрузить его в проект.

Пункт меню “Экспортировать все” открывает выбор папки, в которую будут экспортированы все существующие в проекте мнемосхемы в отдельные локальные файлы, после чего в выбранной папке появятся файлы всех сохранённых мнемосхем.

При нажатии правой кнопкой мыши на корневому элементу “Редактор элементов” появляется всплывающее меню:

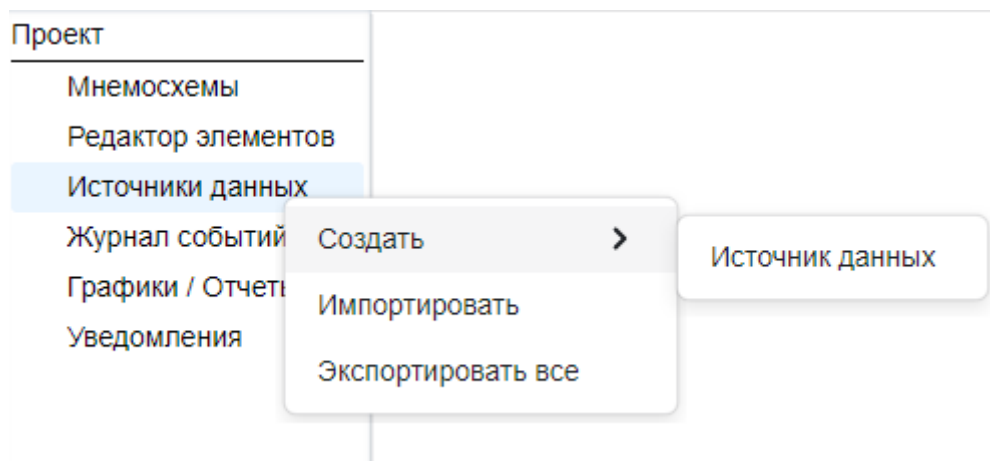


Пункт меню “Создать” позволяет создать новый пустой элемент.

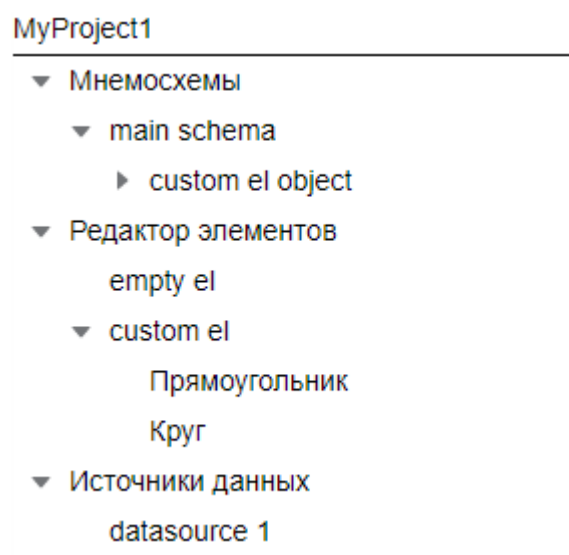
Пункт меню “Импортировать” позволяет выбрать локальный файл сохранённого ранее элемента и загрузить его в проект.

Пункт меню “Экспортировать все” открывает выбор папки, в которую будут экспортированы все существующие в проекте элементы в отдельные локальные

файлы, после чего в выбранной папке появятся файлы всех сохранённых элементов.



Пример развёрнутого дерева рабочего проекта:



В этом примере проект “MyProject1” содержит 1 мнемосхему “main schema”, на которой расположен объект пользовательского элемента “custom el object”. Также в проекте присутствуют два редактора элементов: первый элемент “empty el” не содержит в себе примитивов, второй пользовательский элемент “custom el” содержит в себе два примитива - “Прямоугольник” и “Круг”. Помимо этого в проекте есть источник данных “datasource 1”.

Панель свойств

В правой части приложения находится панель свойств. В этой панели отображаются все доступные свойства выбранного объекта. Свойства могут отображаться для проекта, мнемосхемы, пользовательского элемента, примитива, редактора элемента, источника данных. Свойства объекта, отображаемые в этой панели, могут быть изменены пользователем, изменения будут применены мгновенно.

Все Настройки редактора

Основные

Настройки редактора ^

Отображать сетку

Растягивать по сетке

Перемещать по сетке

Цвет сетки v

Горизонтальный шаг сетки 20 ^ v

Вертикальный шаг сетки 20 ^ v

Цвет фона редактора v

Основные ^

Наименование MyProject1

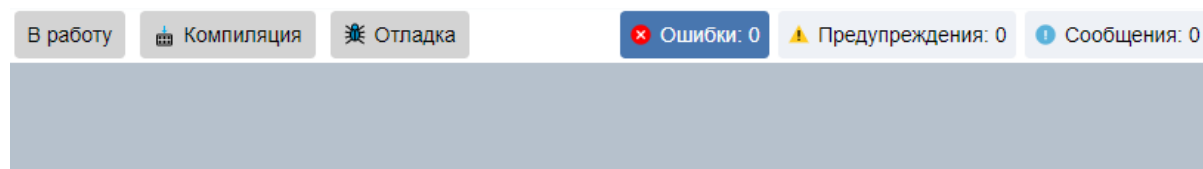
Описание

Время создания 2025-05-15 06:39:29

Панель отладки

Под рабочей областью расположена панель отладки. Кнопка “В работу” переводит проект в состояние наблюдаемого проекта. “Компиляция” выполняет

сборку всех написанных пользователем скриптов и виртуальных переменных, после чего в поле ниже будут выведены ошибки, предупреждения и сообщения, полученные при компиляции. “Отладка” позволяет перейти в режим отладки, который предназначен для тестирования созданных виртуальных переменных и скриптов без реальных физических устройств.



Компиляция и отладка проекта

Для работы проекта все скрипты и виртуальные переменные, написанные пользователем, собираются на сервере в единую библиотеку. Для этого они компилируются. При компиляции происходит проверка кода скриптов и виртуальных переменных на синтаксические ошибки. Функция компиляции создаёт библиотеку и возвращает пользователю список ошибок, предупреждений и сообщений, возникших при компиляции. Важно понимать, что всё, что скомпилировалось корректно, будет работать. То есть если при компиляции проекта произошла ошибка в одном конкретном скрипте, то именно он не будет выполняться, всё остальное продолжит работу согласно логике в других скриптах. Скомпилировать проект можно либо в режиме редактирования по нажатию на кнопку “Компиляция”, либо перейти в режим отладки, при переходе проект будет автоматически скомпилирован и все ошибки отобразятся на панели отладки.

Рассмотрим пример отладки. Есть готовый элемент со скриптами и виртуальными переменными. В режиме отладки при нажатии на двигатель (1) в окне справа внизу (2) отображаются регистры, виртуальные переменные и скрипты выбранного элемента. На рисунке уже заданы значения для виртуальных переменных двигателя (3), благодаря чему на мнемосхеме двигатель стал серым, а тормоз зелёный. Произошло это из-за скриптов, связанных с этими виртуальными переменными. Для отладки необходимо вписать в поле значения тестируемой переменной или регистра новое значение, после чего нажать “Отправить изменения”. Это вызовет обновление значения

виртуальной переменной и зависящие от этой виртуальной переменной скрипты будут выполнены.

The screenshot displays the SCADA interface for 'ЭЛКУБ SCADA'. The main window shows a 3D perspective view of a conveyor belt system. A red box labeled '1' highlights a motor icon on the belt. A red box labeled '2' highlights the 'Конвейер 1' (Conveyor 1) object. A red box labeled '3' highlights the 'Переменные' (Variables) panel in the bottom right corner.

The 'Переменные' panel contains the following data:

Регистр	Переменные	Скрипты
engine_state	0	engine_state_c
stopper_state	0	stopper_state_

Below the table, there is a text input field containing the number '1' and a button labeled 'Отправить изменения' (Send changes). The status 'Количество изменений: 0' (Number of changes: 0) is displayed next to the button.

The top status bar shows the user 'admin' and the date '2025-05-15 12:48:03'. The left sidebar contains a tree view with 'Мнемосхемы' (Mnemonic Schemes) and 'Источники данных' (Data Sources).

В редакторе виртуальные переменные двигателя заданы следующим образом:

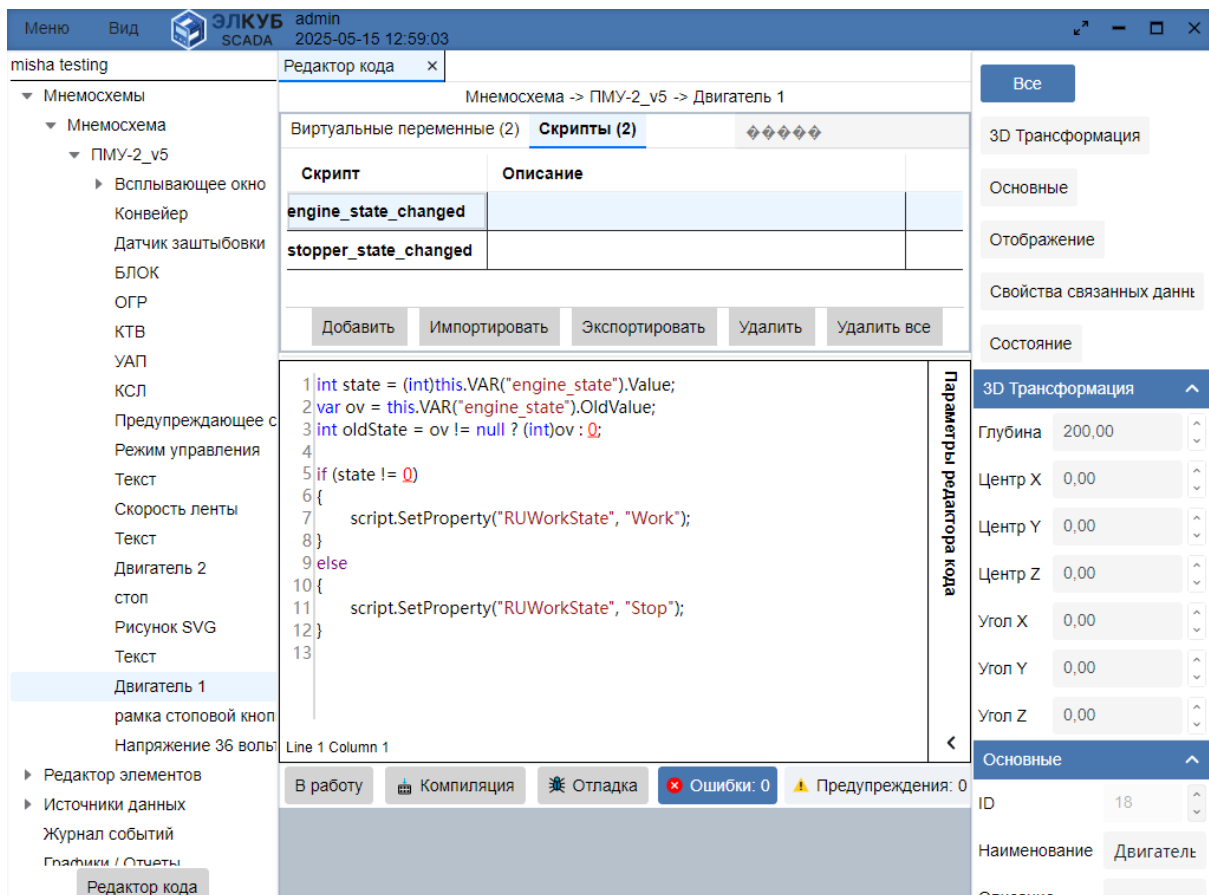
The screenshot shows the 'Редактор кода' (Code Editor) window for 'Мнемосхема -> ПМУ-2_v5 -> Двигатель 1'. The window title is 'misha testing' and the user is 'admin' with the date '2025-05-15 12:54:32'. The interface is divided into several sections:

- Left Panel (Tree View):** Shows a hierarchy of 'Мнемосхемы' (Schematics) under 'Мнемосхема'. The selected element is 'Двигатель 1' (Motor 1).
- Center Panel (Code Editor):** Contains a table of virtual variables and a code editor.

Переменная	Описание
engine_state	
stopper_state	

```
1 int mask = 0b0000_0000_0000_0001;
2 int reg_value = (int)parent.VAR("DRIVES_TRIGGERED_1_8_SENSORS").Value;
3 return reg_value & mask;
```
- Right Panel (Parameter Editor):** Shows '3D Трансформация' (3D Transformation) parameters with values for Depth, Center X, Y, Z, and Angles X, Y, Z, all set to 0.00.
- Bottom Panel:** Contains buttons for 'В работу' (Start), 'Компиляция' (Compile), 'Отладка' (Debug), and status indicators for 'Ошибки: 0' (Errors: 0) and 'Предупреждения: 0' (Warnings: 0).

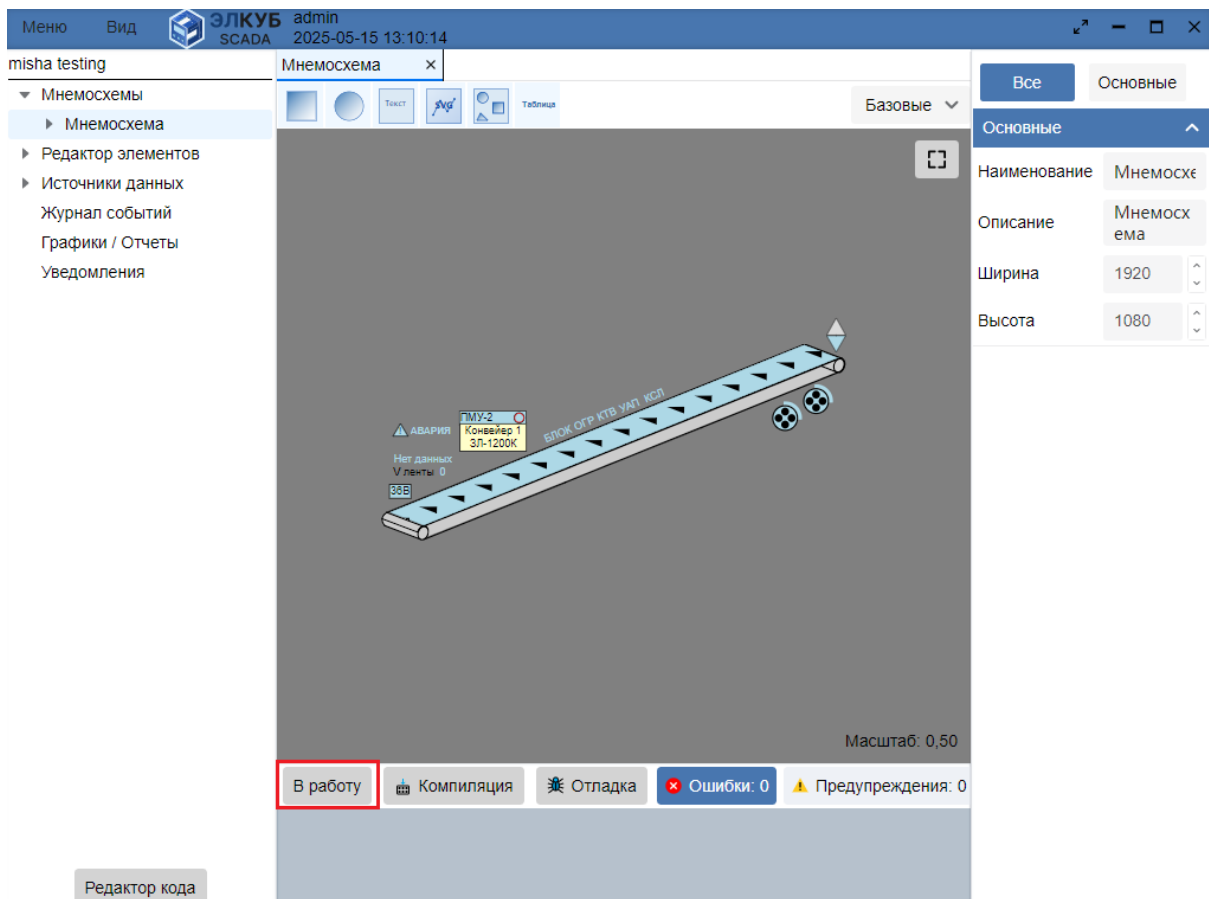
У родительского элемента (это весь пользовательский элемент) для примитива двигателя (в дереве он обозначен как Двигатель 1) есть виртуальная переменная DRIVES_TRIGGERED_1_8_SENSORS, которая получает значения из регистра физического устройства. Двигатель 1, согласно карте регистров физического устройства, должен запускаться, если первый бит регистра, отвечающего за состояния двигателей, выставлен в единицу. Соответственно для проверки этого условия на значение всего регистра накладывается маска, проверяющая значения исключительно первого бита. Скрипт для изменения отображения примитива выглядит следующим образом:



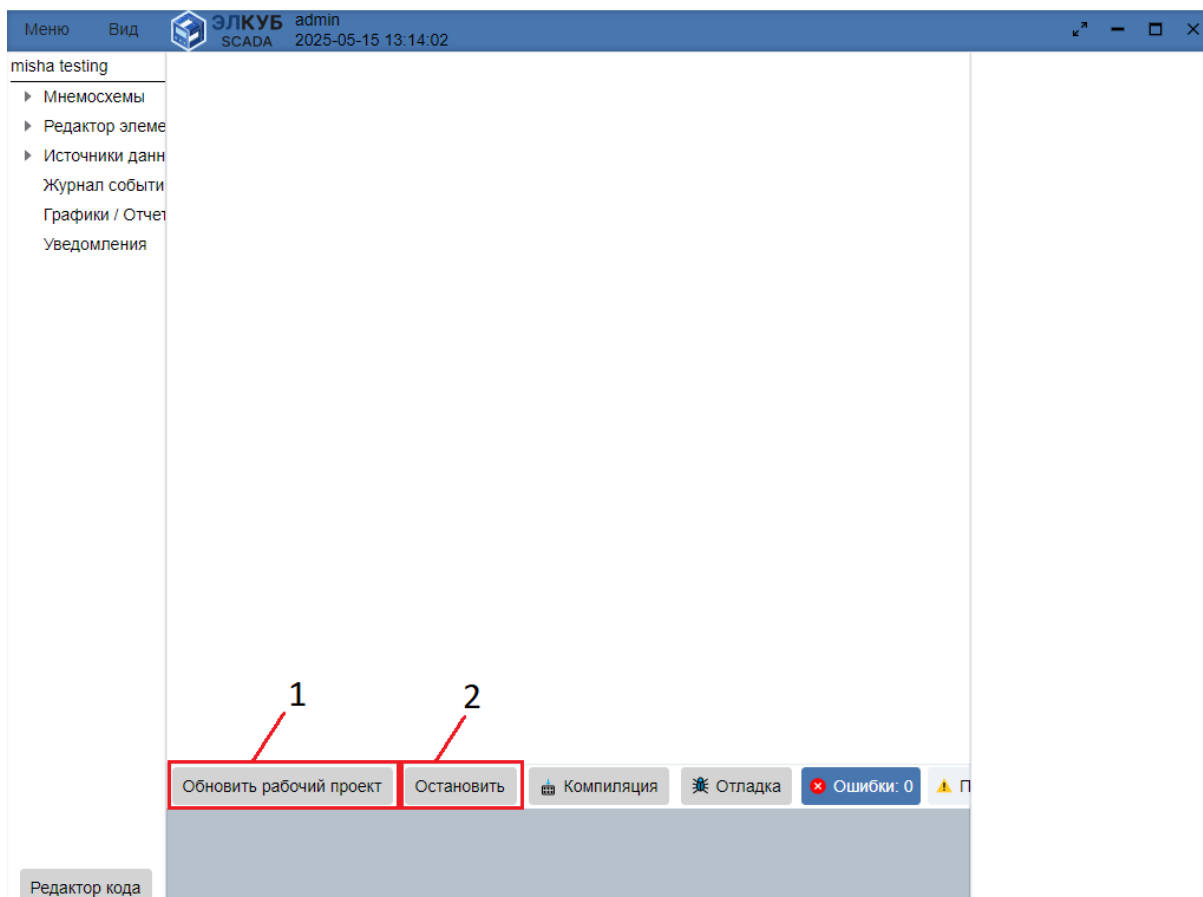
Скрипт срабатывает при изменении переменной `engine_state`, берёт текущее состояние переменной, если оно не равно 0 (то есть бит, отвечающий за Двигатель 1, равен единице), то у примитива меняется отображение - его состояние переходит в значение "Work", что соответствует успешно запущенному двигателю.

Запуск проекта в работу

Для того, чтобы разработанные в режиме редактирования мнемосхемы отобразить в режиме наблюдения, их нужно перевести в рабочий режим. Для этого в режиме редактирования необходимо нажать кнопку "В работу" на панели отладки.



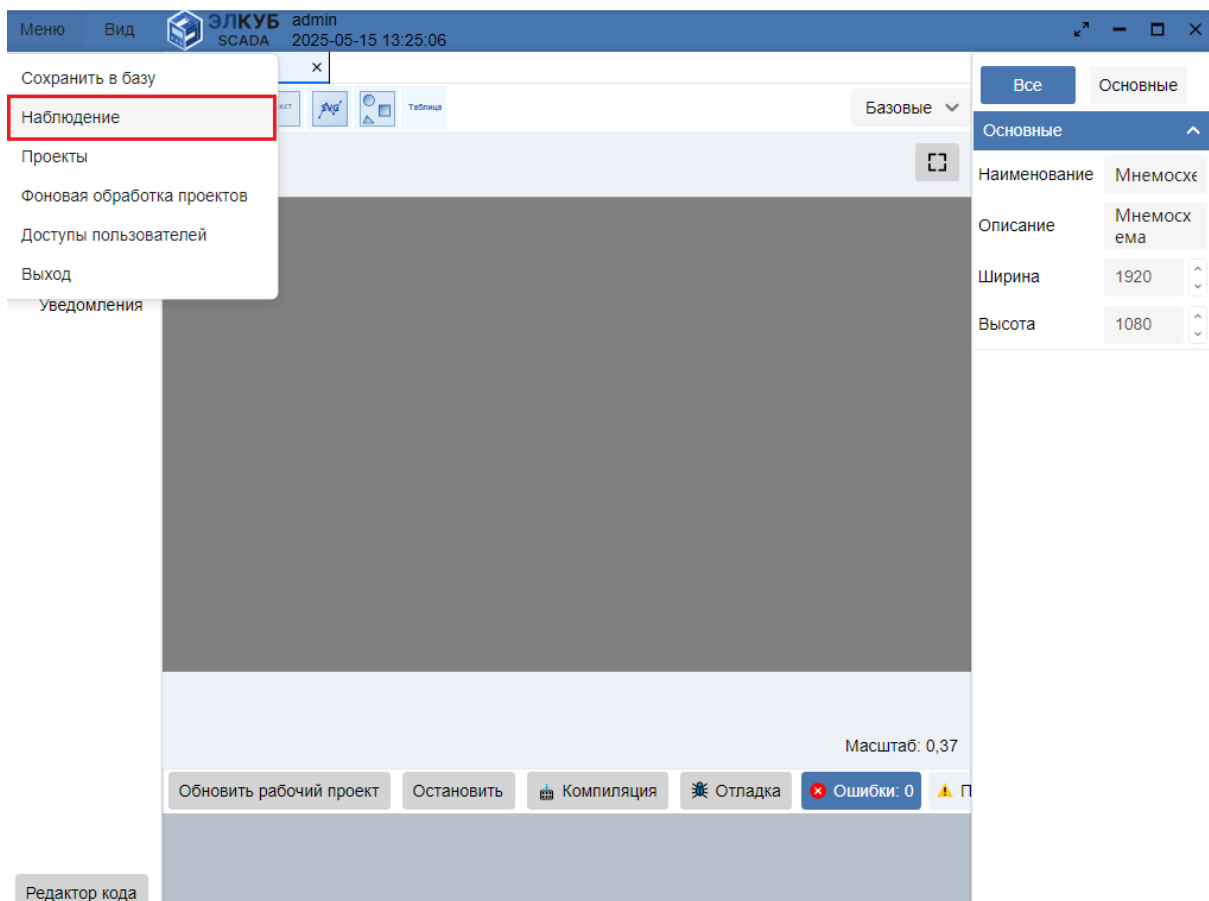
После нажатия на кнопку “В работу” выполнится сохранение проекта на сервере, появится окно подтверждения переноса проекта в наблюдение. После подтверждения проект будет скомпилирован (процесс компиляции идентичен тому, что происходит при нажатии на кнопку “Компиляция” или переходе в режим отладки), а на панели отладки изменятся кнопки. Кнопка “Обновить рабочий проект” (1) вызовет повторную отправку проекта в работу, применив все изменения. Важно помнить, что сохранение проекта на сервере и отправка его в работу не эквивалентные операции. Проект не обновится в режиме наблюдения, если не обновить рабочий проект. Таким образом можно вносить правки в проект, не меняя отображаемый в режиме наблюдения проект. Кнопка “Остановить” (2) уберёт проект из режима наблюдения.



Просмотр проекта

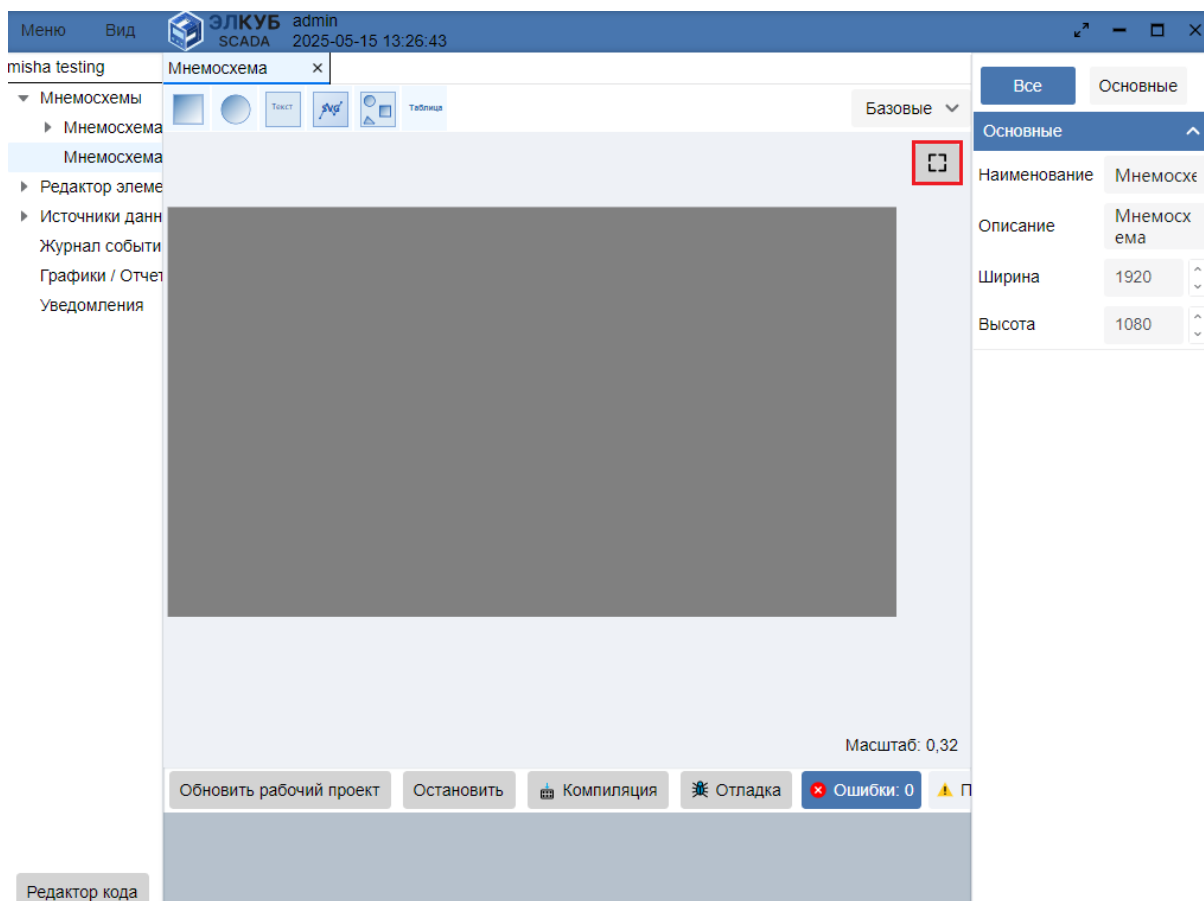
Переход в режим наблюдения

Для перехода в режим наблюдения нужно открыть Меню в левом верхнем углу и выбрать пункт “Наблюдение”:



Масштабирование и перемещение

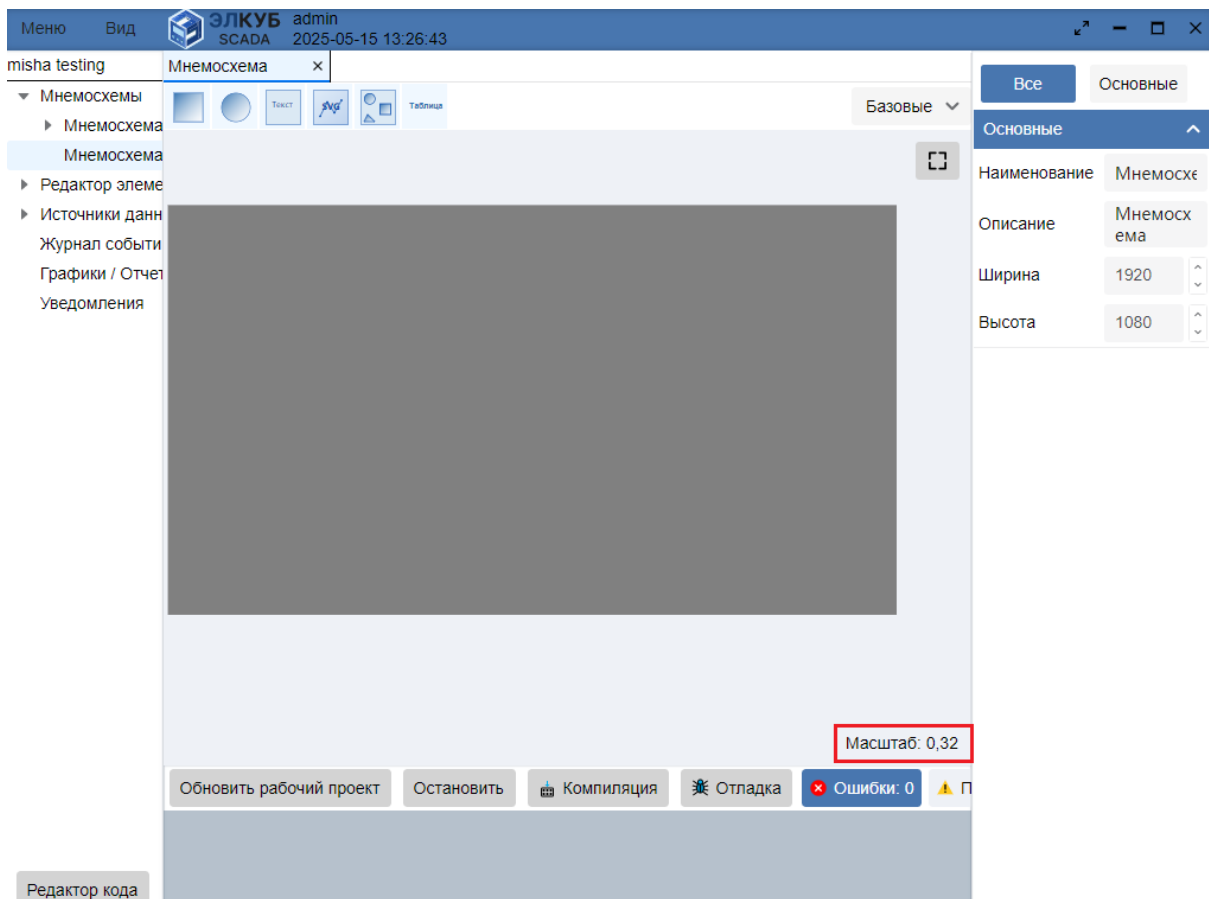
Для масштабирования редактора и перемещения по нему используются ЛКМ, колёсико мыши, CTRL и SHIFT. Дополнительно для масштабирования по общему размеру всех добавленных элементов в правом верхнем углу рабочей области расположена кнопка автоматического масштабирования:



Прокрутка колеса мыши без зажатых кнопок CTRL и SHIFT выполняет вертикальное перемещение редактора.

Прокрутка колеса мыши с зажатым SHIFT выполняет горизонтальное перемещение редактора.

Прокрутка колеса мыши с зажатым CTRL выполняет масштабирование редактора. Текущий масштаб указан в правом нижнем углу рабочей области:

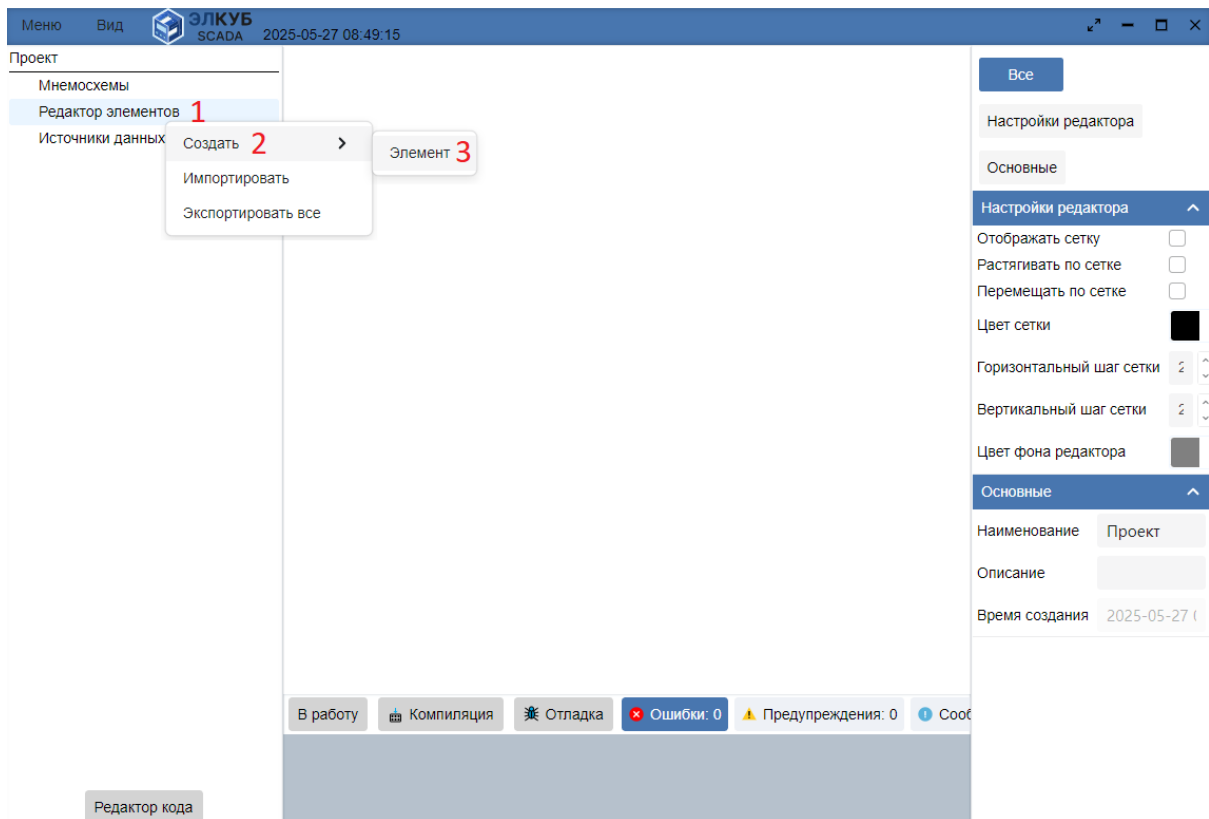


Создание проекта

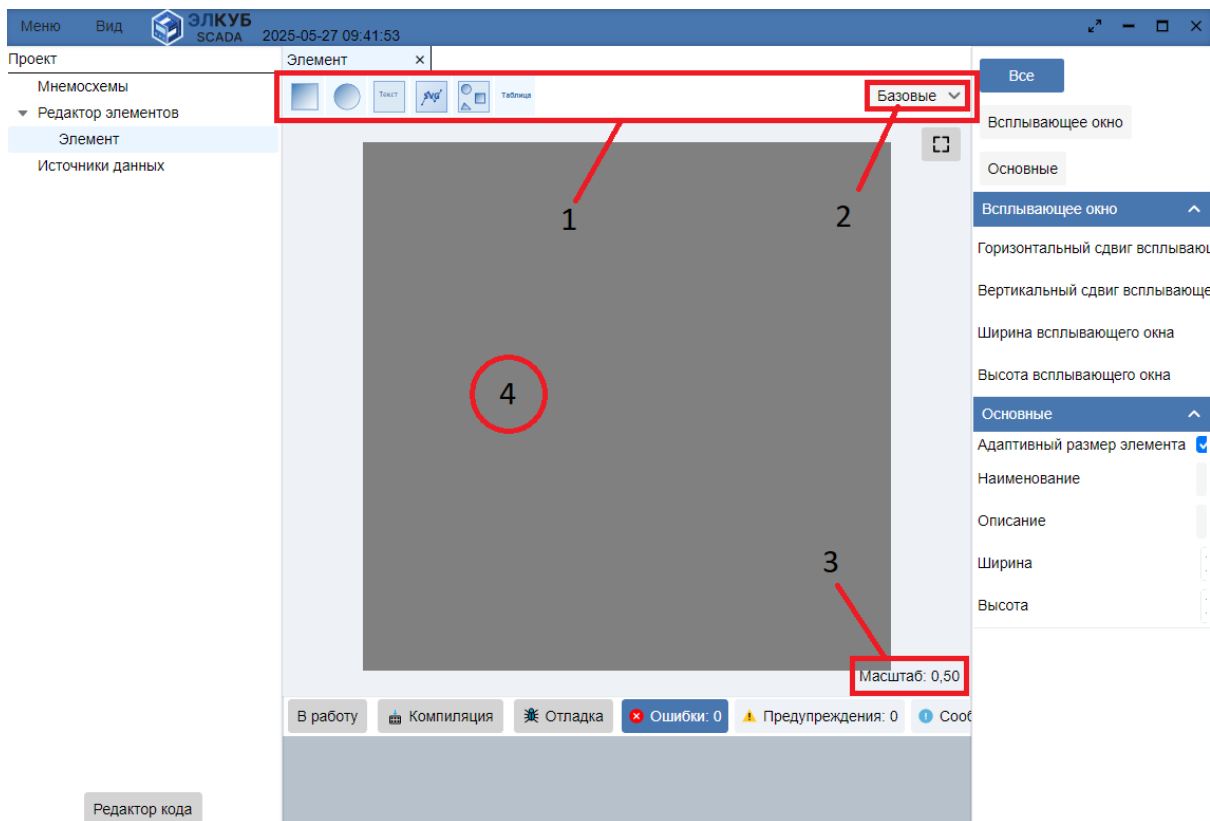
Создание пользовательских элементов

Редактор элементов

Для создания элемента (набора примитивов, связанных общей логикой или контроллером), необходимо создать новый редактор элемента. Для этого в открытом проекте необходимо нажать по пункту “Редактор элементов” и выбрать в появившемся меню “Создать”, “Элемент”.



После этого в рабочей области появится редактор элемента. Область 1 отвечает за перечень примитивов текущей выбранной группы примитивов. Выпадающее меню 2 позволяет выбрать группу элементов, которые будут отображаться в области 1. В правом нижнем углу расположена надпись 3 с текущим масштабом редактора. Вся внутренняя область, помеченная цифрой 4, является рабочим пространством редактора, куда можно размещать примитивы.



Примитивы размещаются в редакторе при помощи перетягивания с панели примитивов в рабочую область редактора. При размещении примитива в правой панели свойства отображаются свойства созданного примитива. Редактирование этих свойств мгновенно применяется к отображаемому примитиву. Перемещаться в пространстве рабочей области редактора можно аналогично перемещению в редакторе мнемосхемы.

Для масштабирования редактора и перемещения по нему используются ЛКМ, колёсико мыши, CTRL и SHIFT. Дополнительно для масштабирования по общему размеру всех добавленных элементов в правом верхнем углу рабочей области расположена кнопка автоматического масштабирования:

Прокрутка колеса мыши без зажатых кнопок CTRL и SHIFT выполняет вертикальное перемещение редактора.

Прокрутка колеса мыши с зажатым SHIFT выполняет горизонтальное перемещение редактора.

Прокрутка колеса мыши с зажатым CTRL выполняет масштабирование редактора.

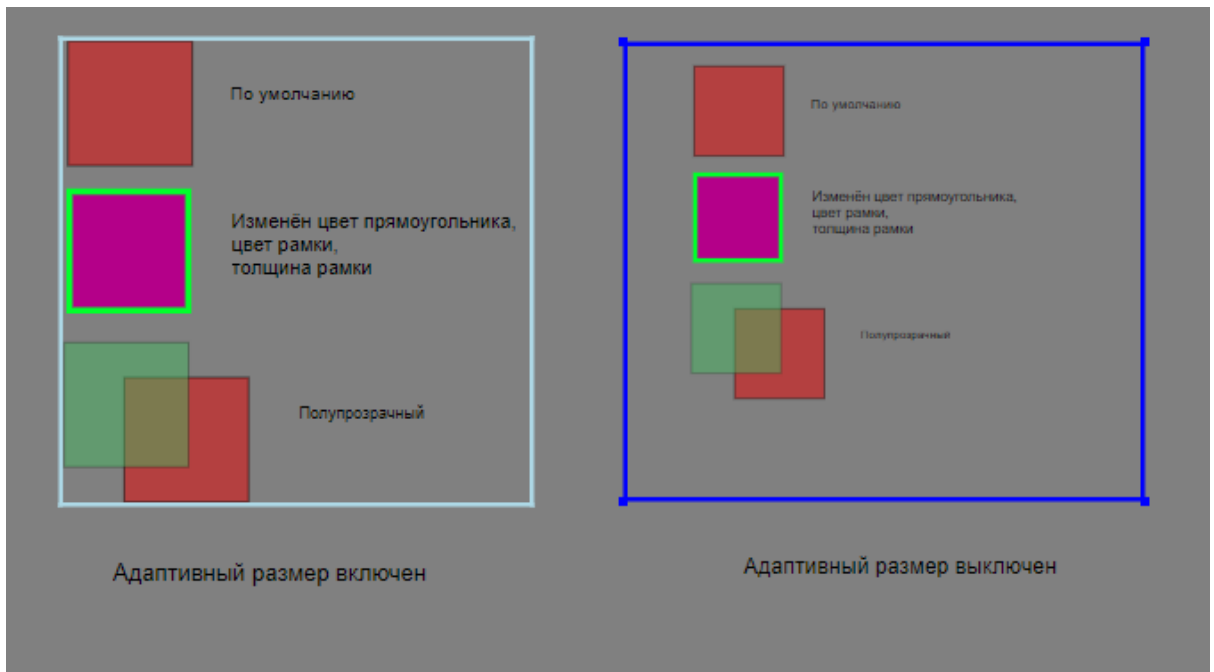
У элемента, создаваемого в редакторе, есть следующие свойства:

Все		Всплывающее окно	Основные
Всплывающее окно ^			
Горизонтальный сдвиг всплывающего окна	200,00	^ v	
Вертикальный сдвиг всплывающего окна	200,00	^ v	
Ширина всплывающего окна	200,00	^ v	
Высота всплывающего окна	200,00	^ v	
Основные ^			
Адаптивный размер элемента	<input checked="" type="checkbox"/>		
Наименование	Элемент		
Описание	Элемент		
Ширина	1000	^ v	
Высота	1000	^ v	

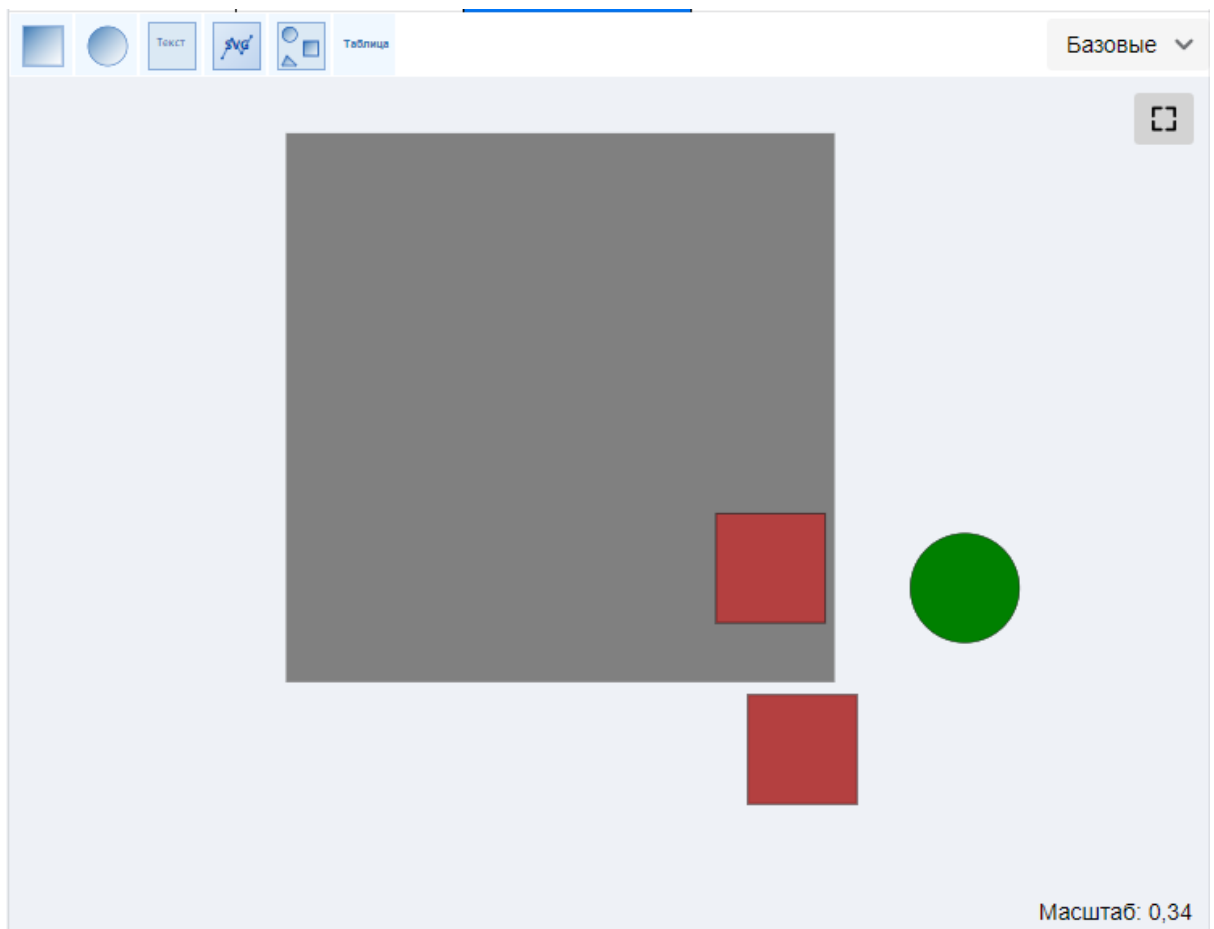
У всех элементов есть две категории свойств: Всплывающее окно и Основные. В категории Всплывающее окно настраиваются параметры размеров всплывающего окна элемента.

Категория “Основные”:

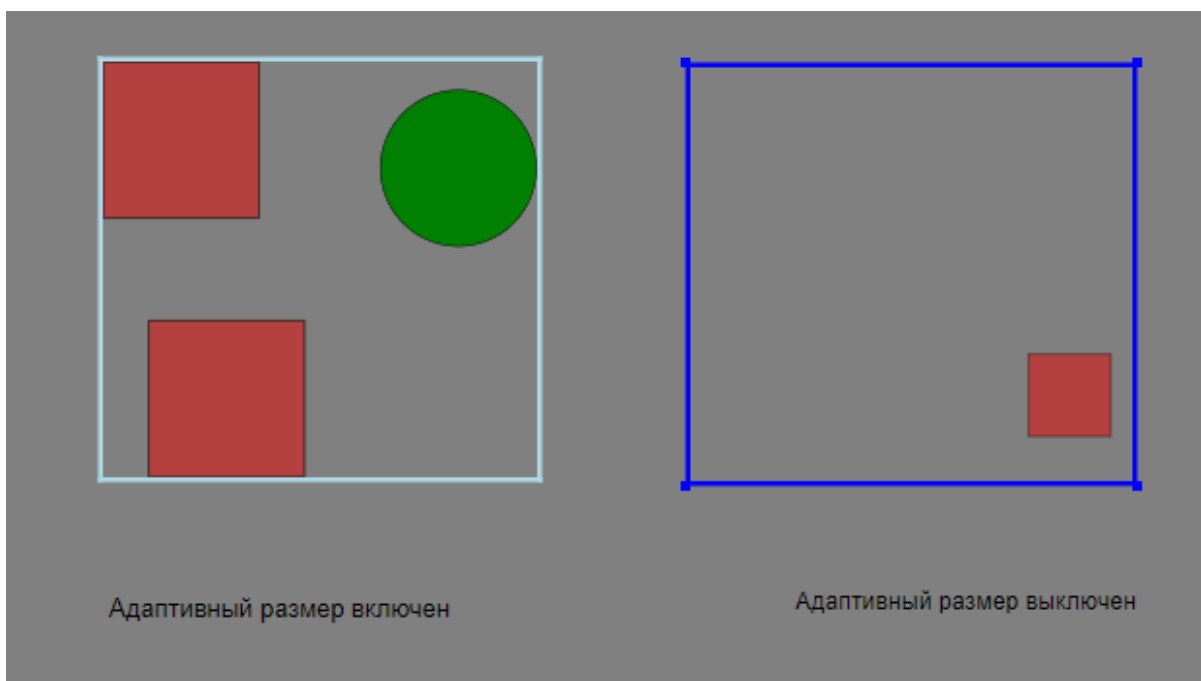
Адаптивный размер элемента - после сохранения элемента и переноса его на мнемосхему размеры канвы элемента будут адаптированы для обтекания примитивов внутри. Если этот параметр отключен, то в элемент будет отображать то, что находится у него в области канвы. Пример со включенным параметром адаптивного размера:



Важно помнить, что при отключенном адаптивном размере элементы, размещённые в редакторе элемента вне области канвы, не будут отображаться корректно после переноса на мнемосхему. Пример элемента:



Пример отображения этого элемента на мнемосхеме:



Наименование - название элемента.

Описание - описание создаваемого элемента.

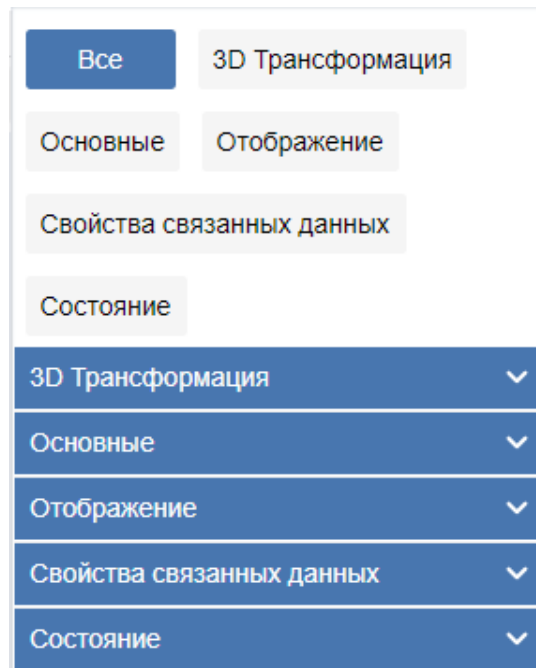
Ширина - ширина канвы элемента.

Высота - высота канвы элемента.

Использование базовых примитивов

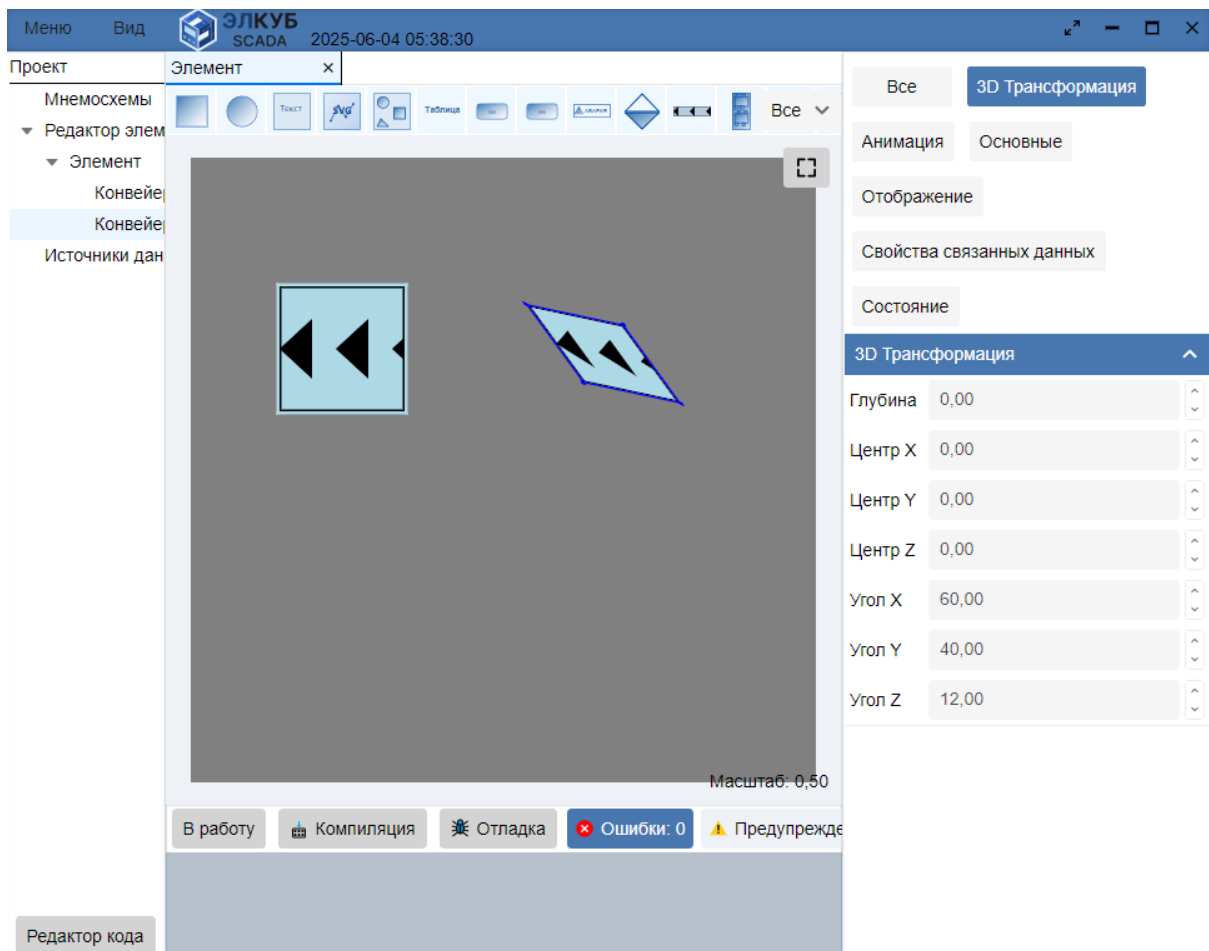
У всех примитивов есть набор общих не уникальных свойств. В основном они используются для изменения отображения примитива в элементе или на мнемосхеме. Общие свойства примитивов: 3D Трансформация, Основные, Отображение, Свойства связанных данных, Состояние. Эти свойства применимы как к примитивам в Редакторе примитивов, так и к уже созданным и размещённым на мнемосхеме элементам.

Внимание! Свойства, уникальные для конкретного примитива, могут располагаться не только в некоторой уникальной для него группе, но и в общих группах согласно смысловой нагрузке уникальных свойств примитива.



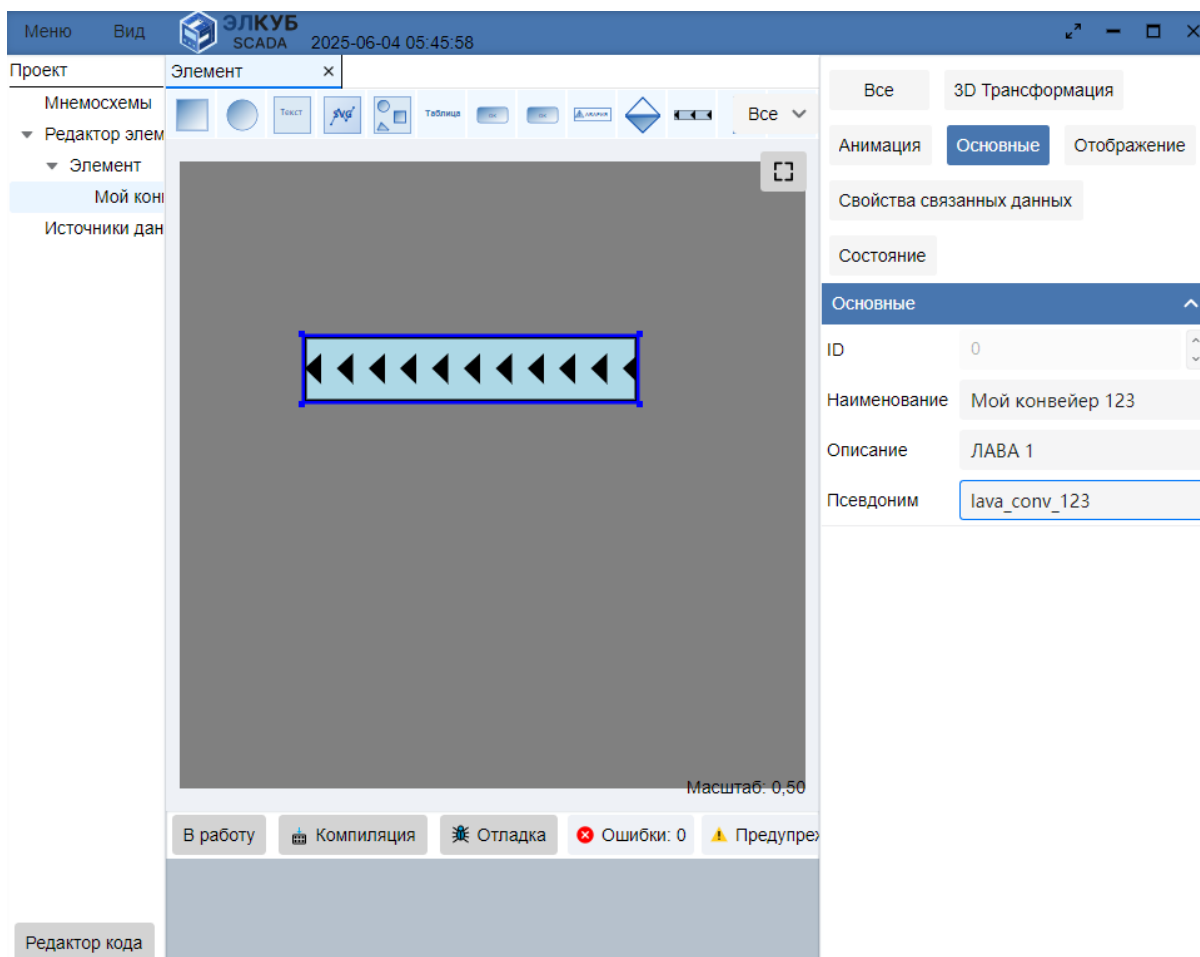
3D Трансформация:

Эта группа свойств используется для применения матрицы поворота примитива, создавая эффект 3D. На рисунке справа изображен конвейер, к которому применено 3D преобразование. Свойства в этой группе: Глубина, Центр X, Центр Y, Центр Z, Угол X, Угол Y, Угол Z. Каждое из этих свойств отвечает за одно значение в матрице поворота, применяемой к примитиву.



Основные:

В группе “Основные” находятся свойства, отвечающие за описание примитива.



ID - уникальный идентификатор примитива в базе данных. Это свойство не изменяется пользователем.

Наименование - имя устройства. Это имя будет использовано в дереве проекта, всплывающих подсказках при наведении на примитив, и так далее.

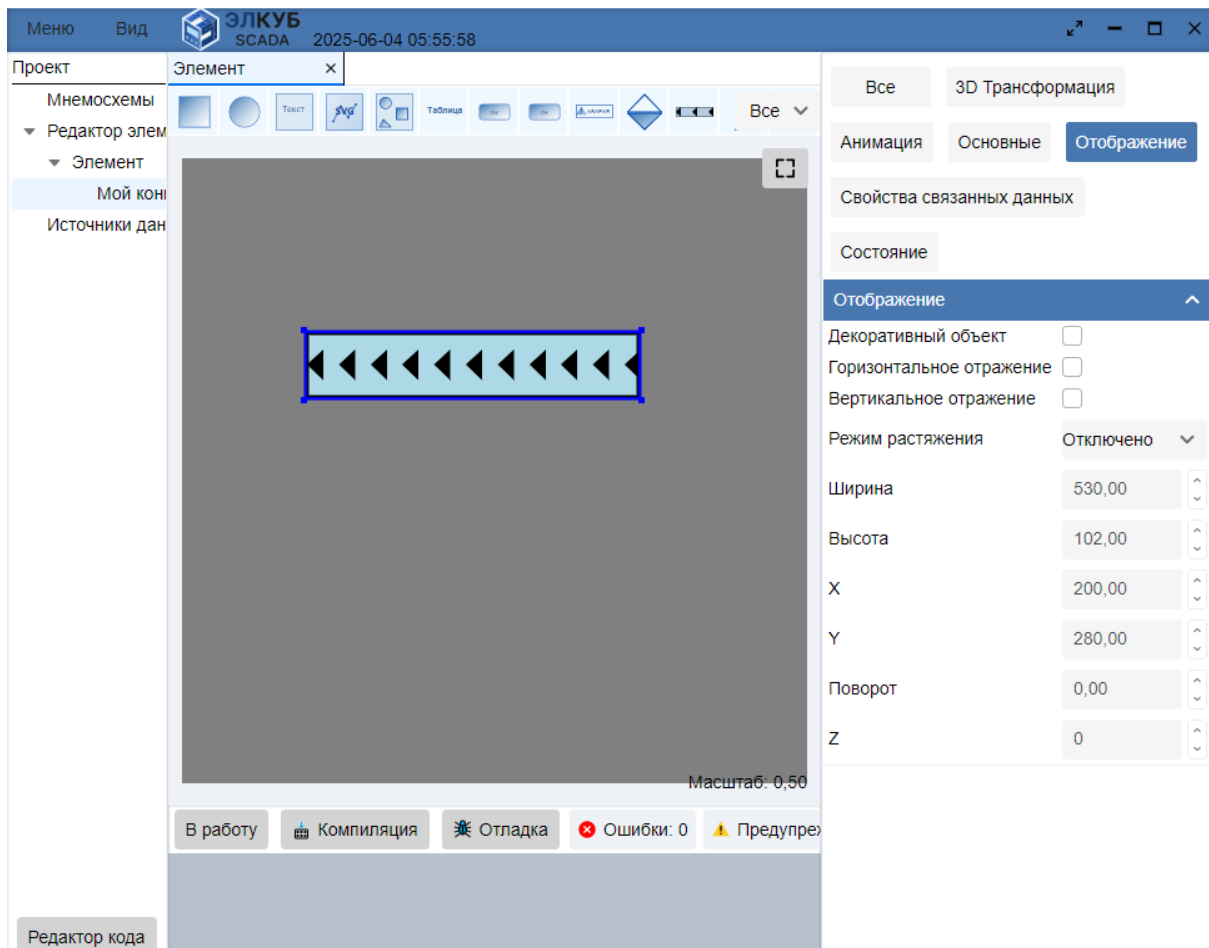
Описание - пояснение для пользователя о назначении и свойствах примитива. Не является необходимым для заполнения.

Псевдоним - название для уникального имени примитива, по которому потом можно будет обращаться к нему из любого места в проекте.

Внимание! Псевдоним должен быть уникальным для всего проекта, не повторяться между элементами и примитивами. В случае наличия нескольких примитивов с одинаковыми псевдонимами произойдёт конфликт имён, ни один из примитивов с одинаковыми псевдонимами не будет доступен для использования.

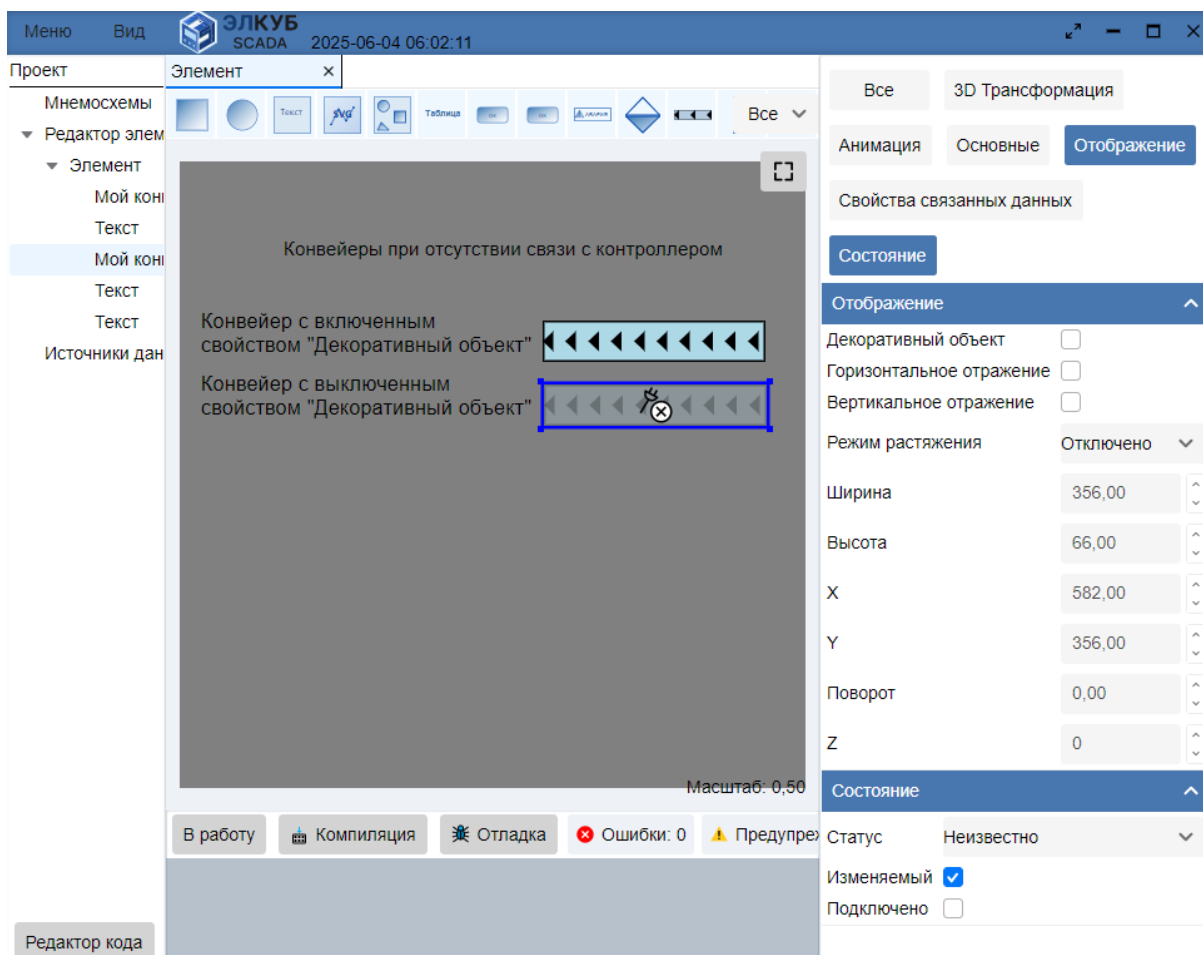
Отображение:

Группа свойств “Отображение” отвечает за параметры отображения примитива в элементе.



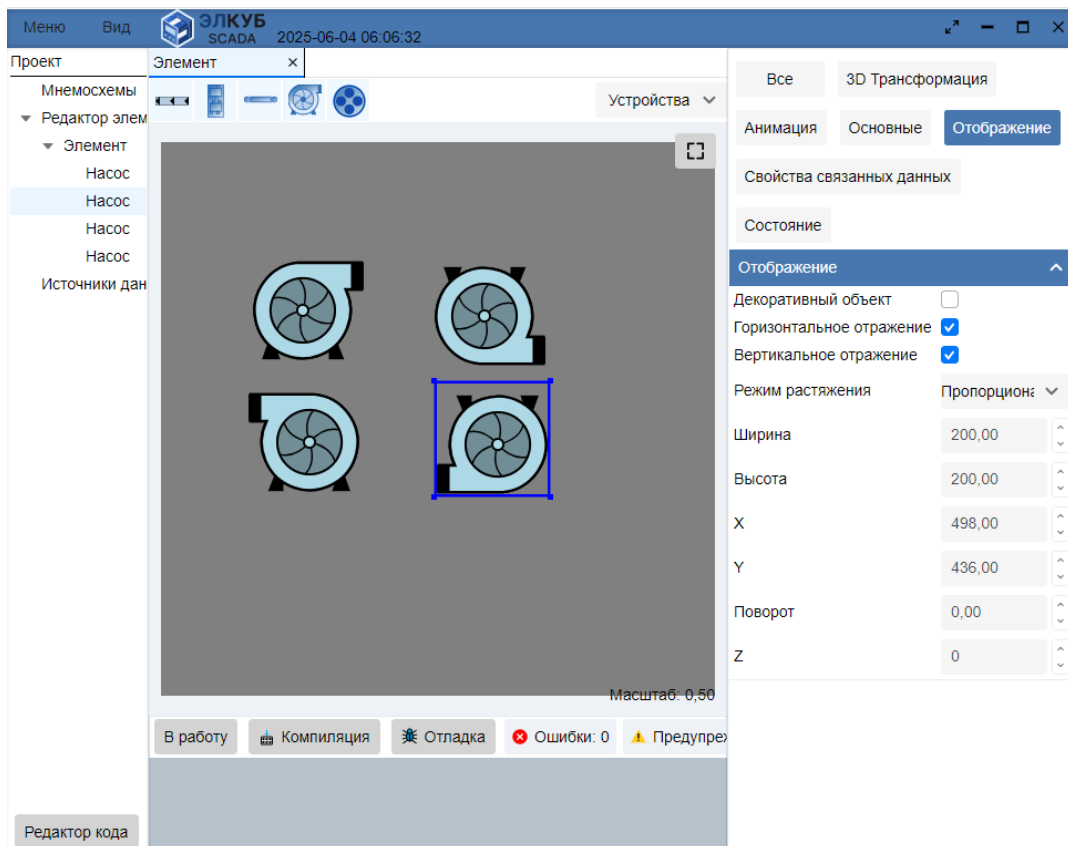
Декоративный объект - свойство, используемое для отключения изменения отображения при отсутствии связи у элемента, в котором находится примитив. Может применяться, например, для сохранения отображения поясняющих надписей на в элементе при отсутствии связи с контроллером, к которому подключен элемент.

Пример:

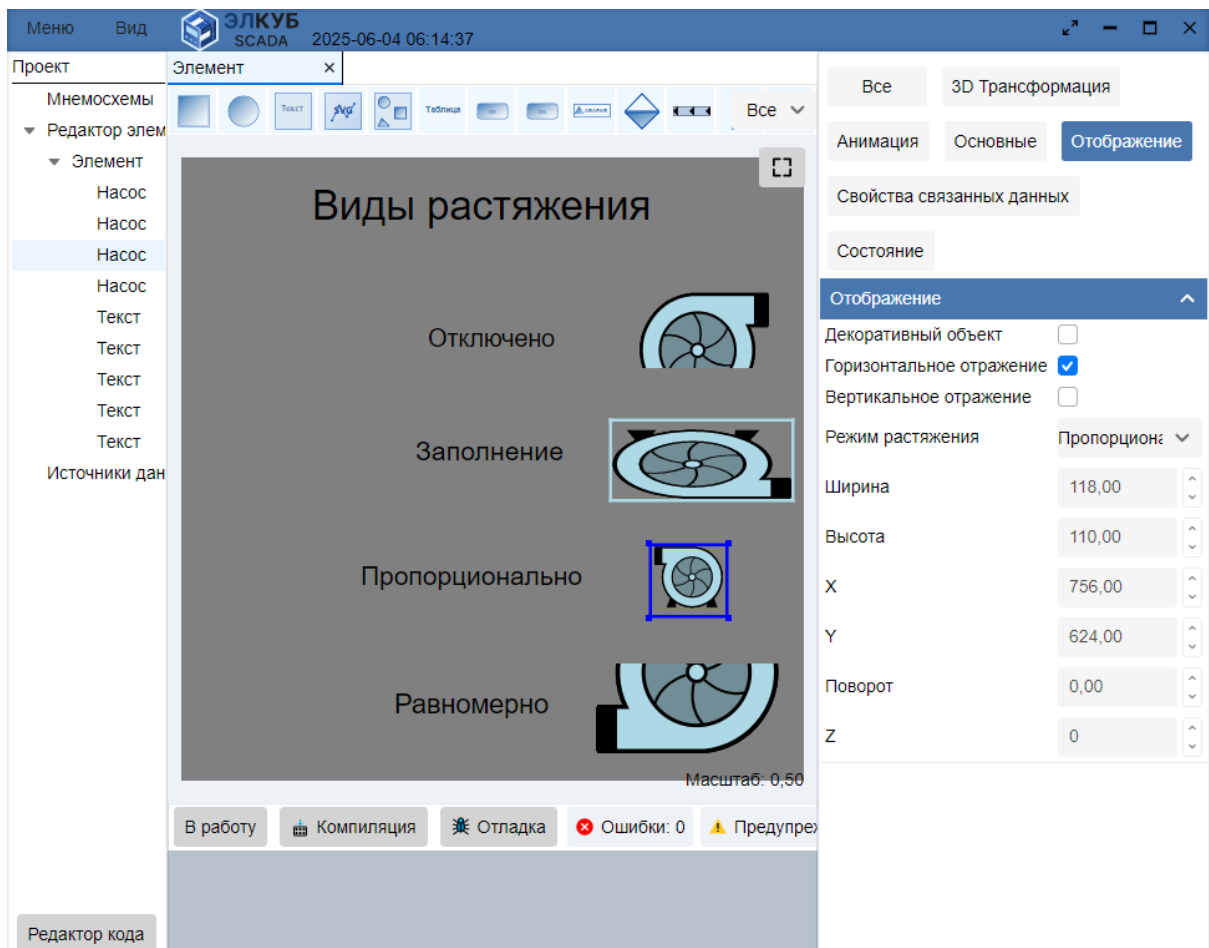


Горизонтальное и вертикальное отражения - применяют трансформацию к примитиву для его горизонтального или вертикального отражения соответственно.

Пример:

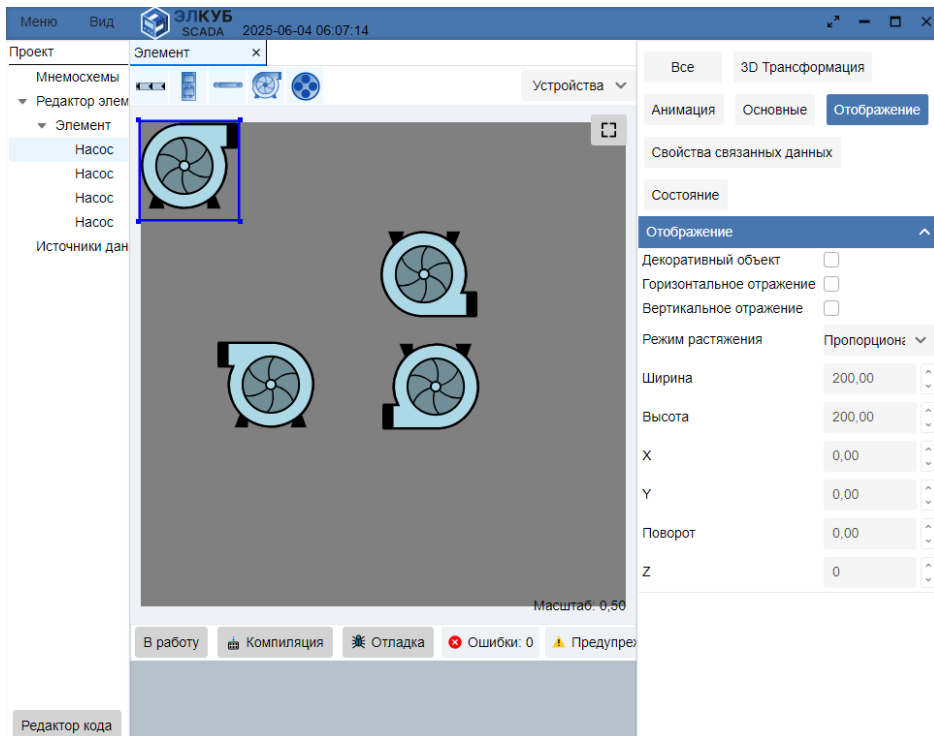


Режим растяжения - свойство, позволяющее изменить поведение отображения примитива при растяжении. Может принимать значения: Отключено, Пропорционально, Заполнение, Равномерно. Пример:

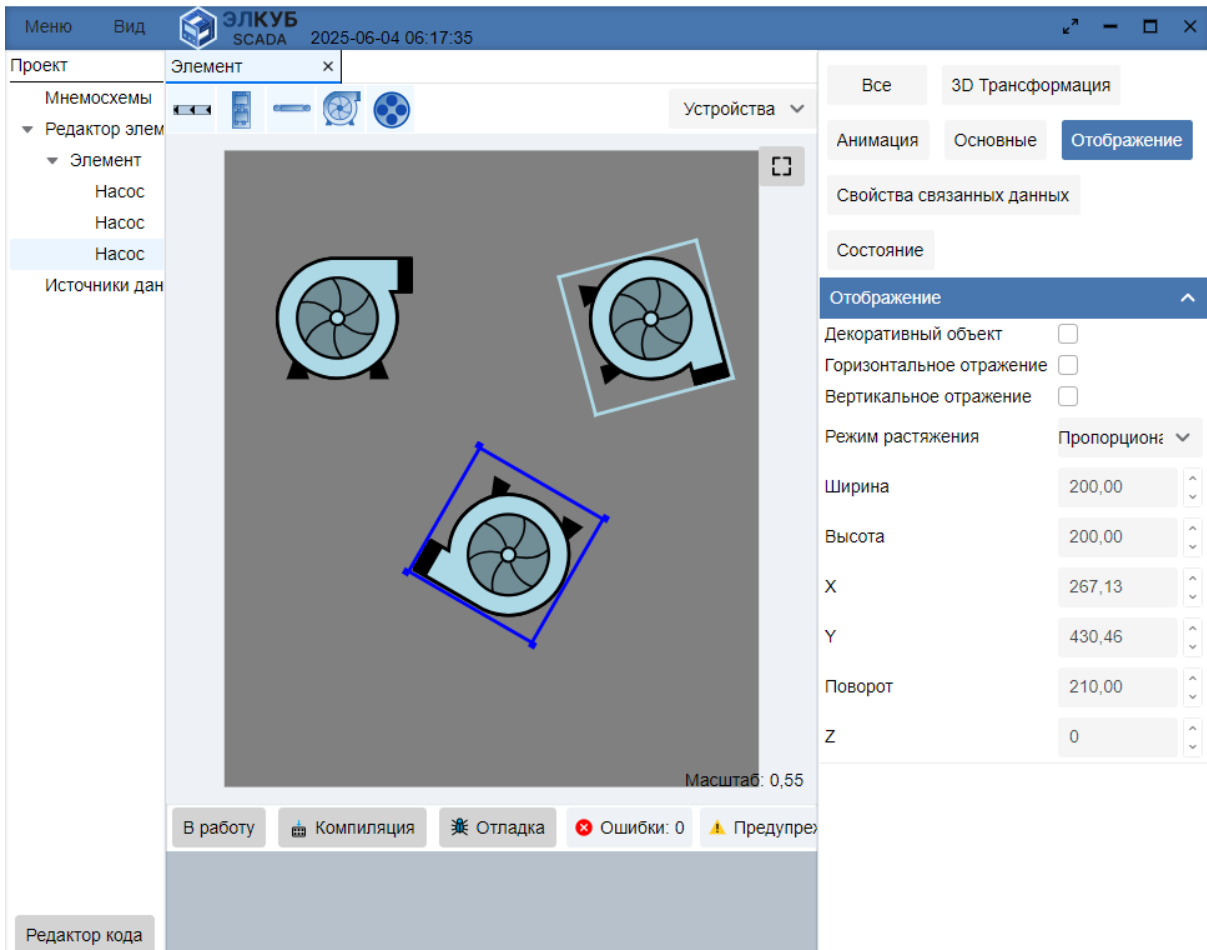


Ширина, высота - ширина и высота примитива в координатной плоскости канвы редактора, не зависят от масштаба отображения в редакторе.

X, Y - координаты примитива относительно верхнего левого угла канвы редактора. Пример:



Поверот - применение трансформации поворота на угол от 0 до 360 по часовой стрелке к примитиву. Пример:



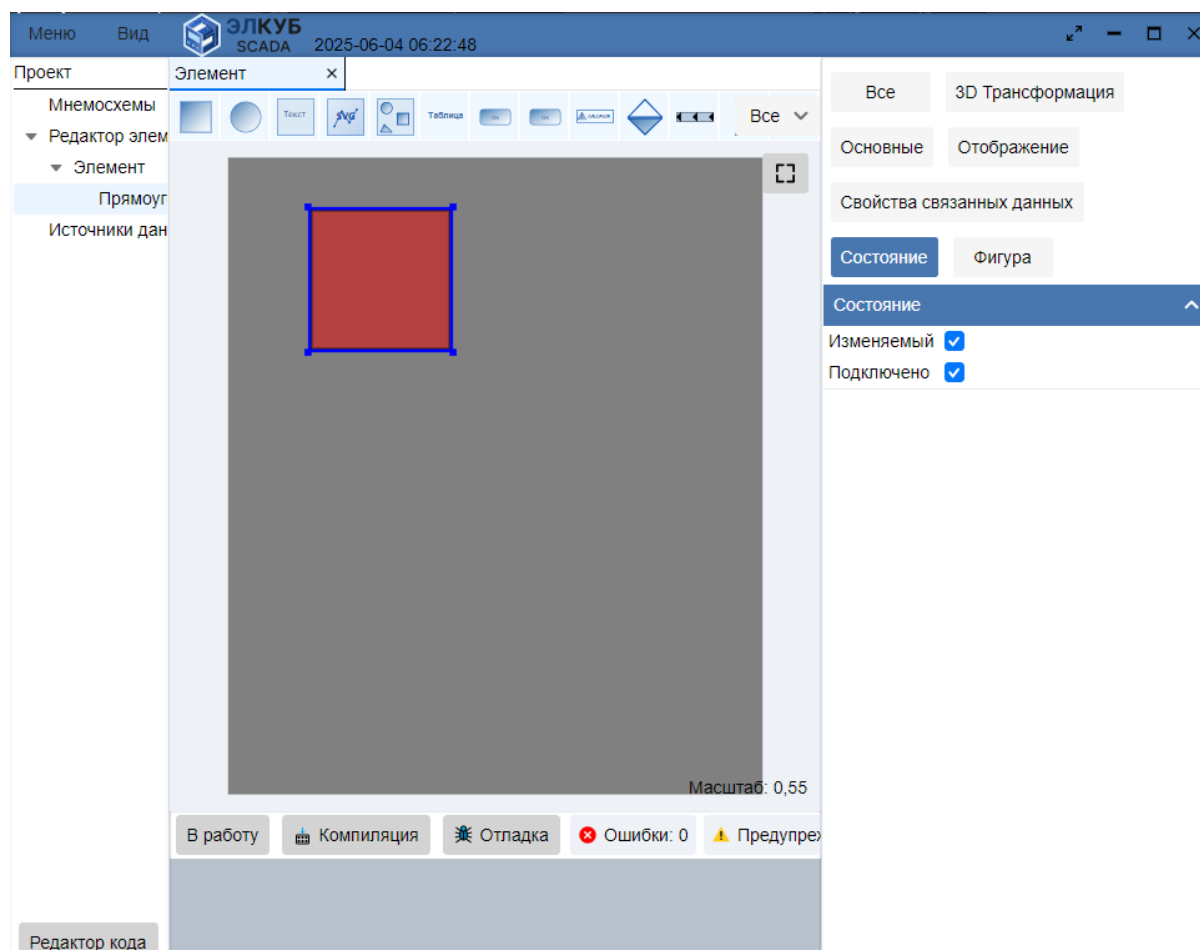
Z - координата примитива по оси Z. Изменение этого свойства позволяет контролировать расположение примитивов относительно друг друга при их наложении. Примитив с наибольшим значением Z координаты будет отображаться поверх остальных при их наложении.

Свойства связанных данных:

Эта группа отображает кол-во добавленных виртуальных переменных и скриптов. Не может быть изменена пользователем.

Состояние:

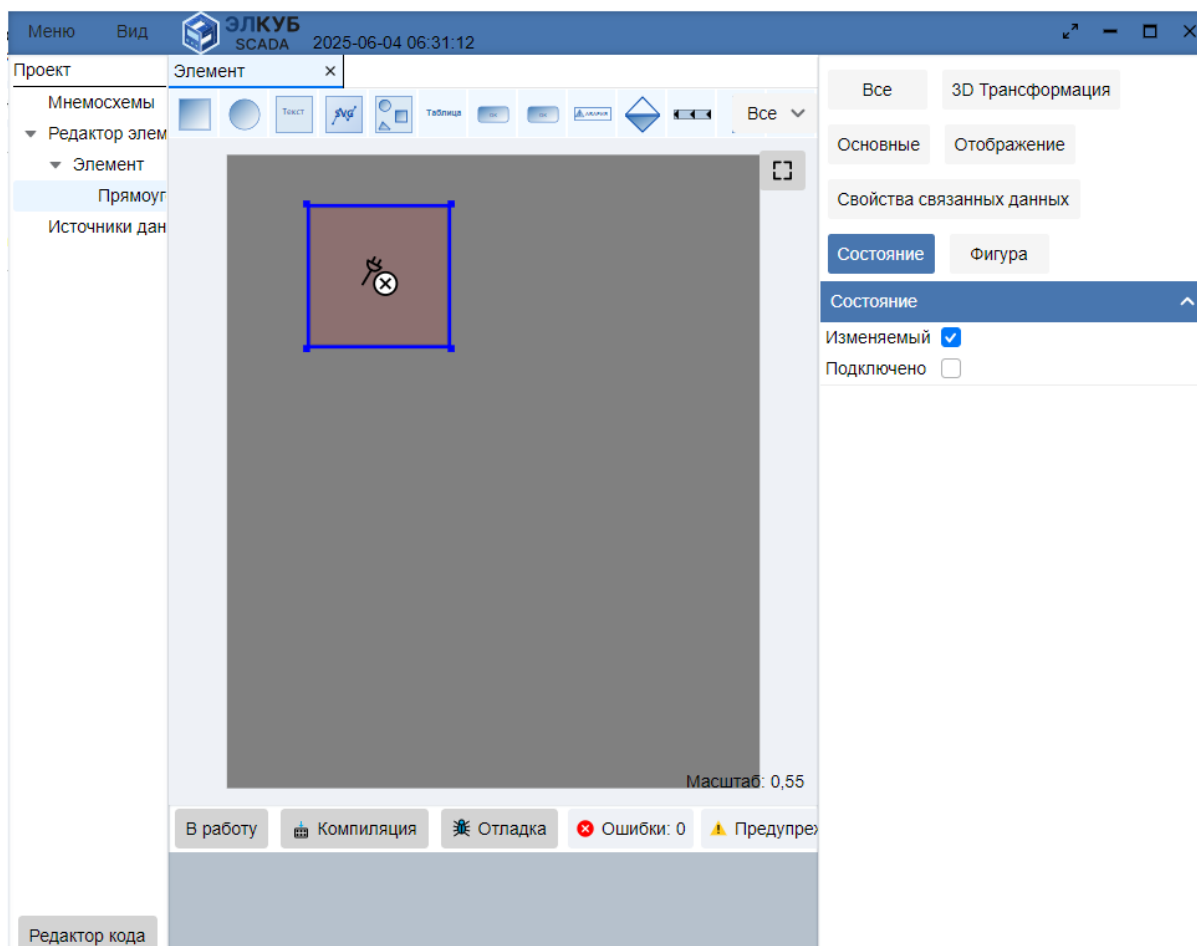
Эта группа предназначена для отображения состояния реального физического устройства в примитиве. Все примитивы имеют два общих свойства в этой группе: **Изменяемый** и **Подключено**.



Изменяемый - если включено (по умолчанию включено), свойства примитива могут быть изменены. Если это свойство выключено, ни одно из свойств, кроме свойства "Изменяемый", не может быть изменено.

Подключено - это свойство используется в скриптах для отображения состояния подключения к контроллеру. Общее базовое поведение для

элементов при переключении этого свойства в состояние false - затемнение примитива. Однако некоторые примитивы имеют дополнительную индикацию отсутствия подключения, например, значок.



Прямоугольник

Уникальными свойствами для примитива Прямоугольник находятся в группе “Фигура”.

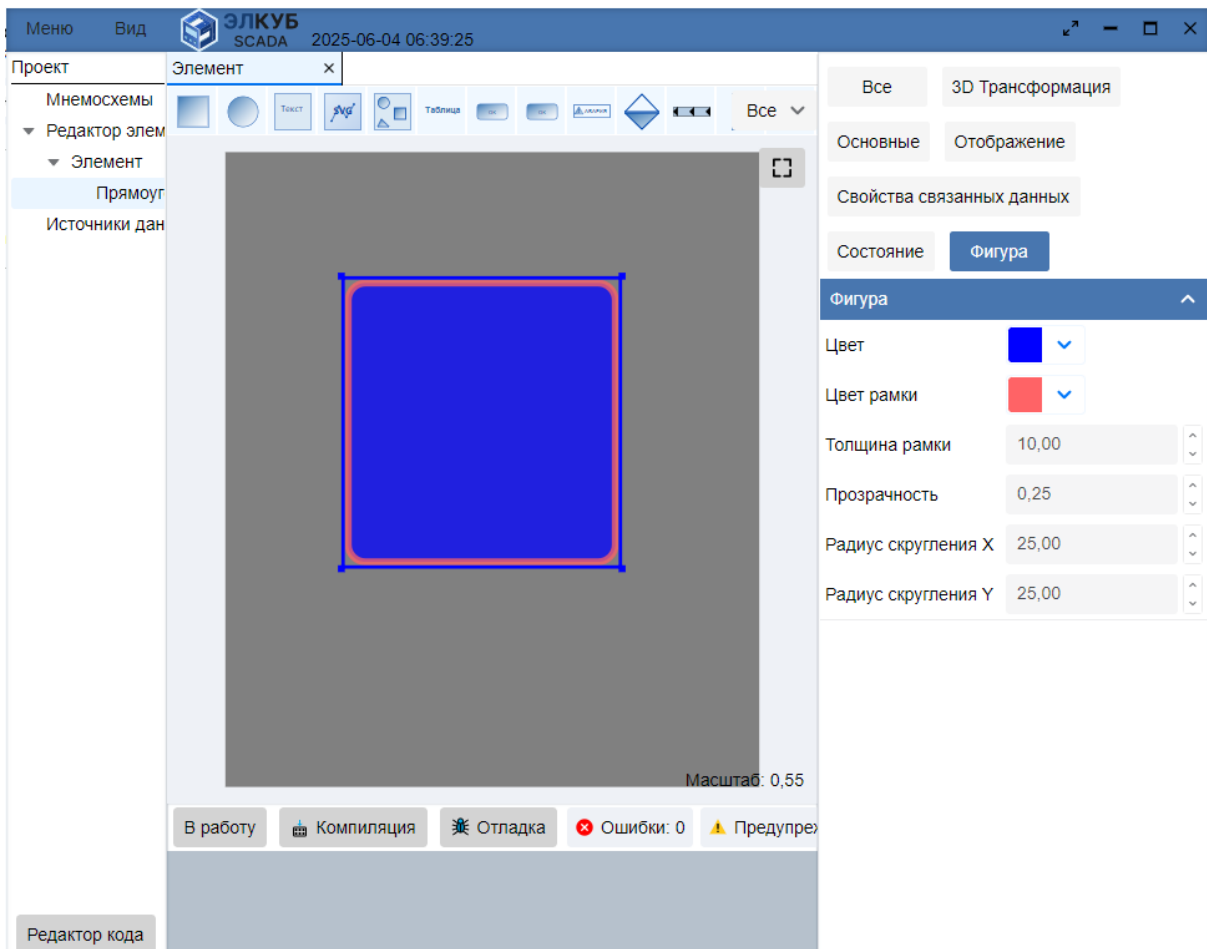
Цвет - цвет внутри прямоугольника.

Цвет рамки - цвет рамки прямоугольника.

Толщина рамки - толщина рамки прямоугольника.

Прозрачность - прозрачность прямоугольника.

Радиус скругления X, Радиус скругления Y - радиусы скругление рамки прямоугольника по оси X и Y соответственно.



Круг

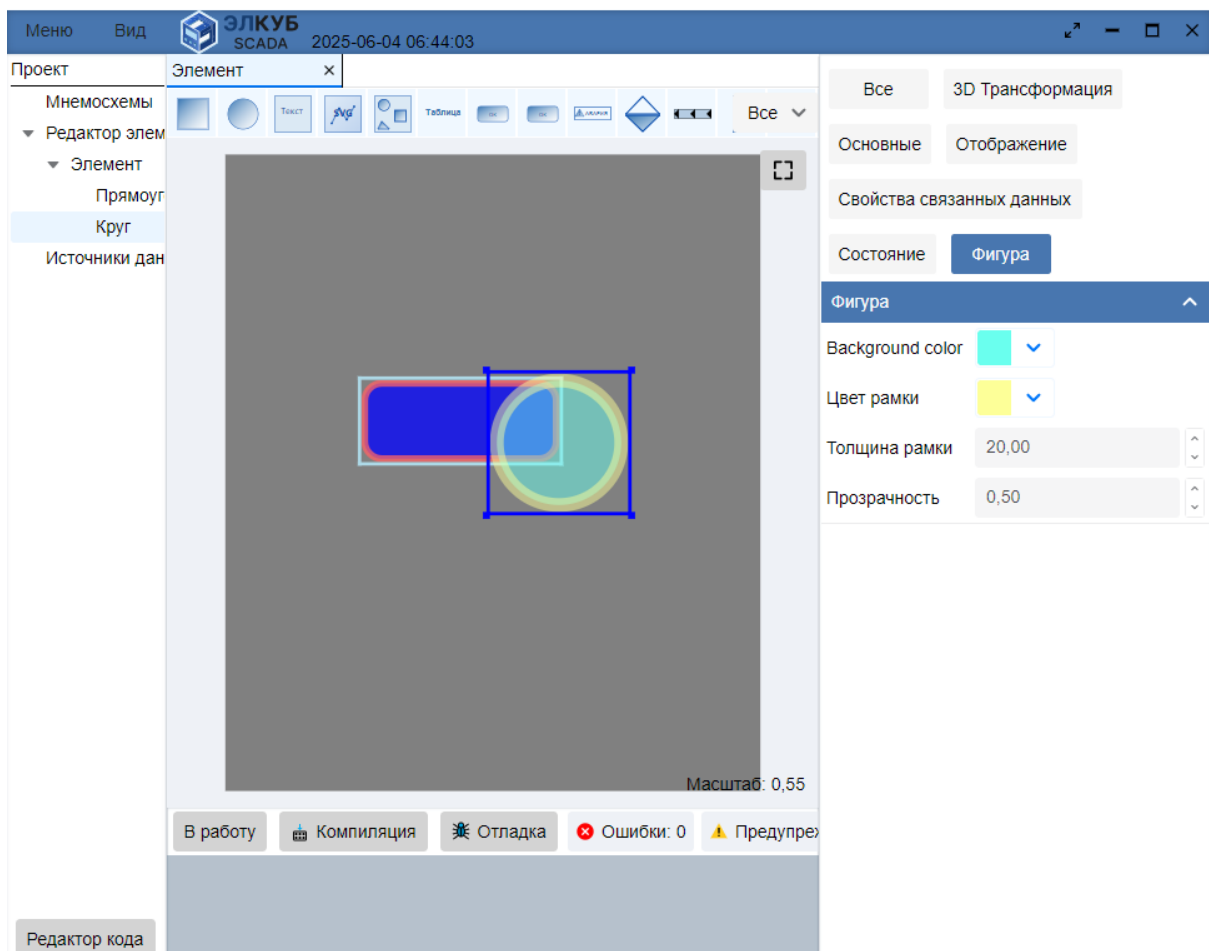
Уникальными свойствами для примитива Круг находятся в группе “Фигура”.

Цвет - цвет внутри круга.

Цвет рамки - цвет рамки круга.

Толщина рамки - толщина рамки круга.

Прозрачность - прозрачность круга.



Текст

У примитива Текст есть одно уникальное свойство в группе “Отображение” и уникальная группа свойств “Текст”.

Толщина рамки (группа “Отображение”) - толщина рамки вокруг текста.

Группа “Текст”:

Текст - содержащийся в примитиве текст.

Автоизменение шрифта - применять автоматическое изменение размера шрифта при изменении размеров всего примитива.

Выравнивание текста - применять к тексту выравнивание внутри примитива.

Размер шрифта - размер шрифта текста в примитиве. При включенном Автоизменении шрифта будет отображаться текущий шрифт текста, будет автоматически меняться при растяжении примитива. При отключенном Автоизменении шрифта размер шрифта текста будет соответствовать значению этого свойства, растяжение примитива не повлияет на размер шрифта.

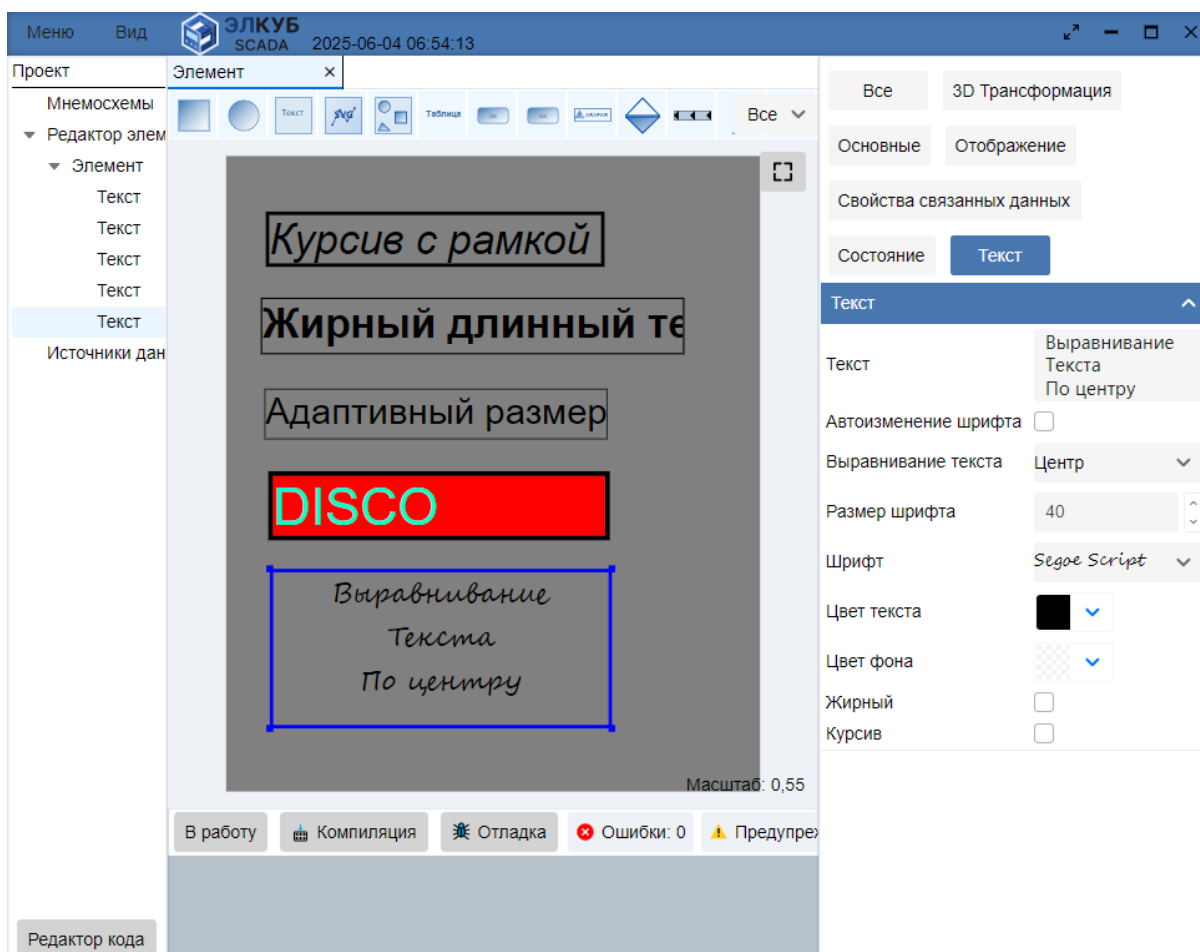
Шрифт - шрифт текста.

Цвет текста - цвет текста в примитиве.

Цвет фона - цвет фона текста в примитиве.

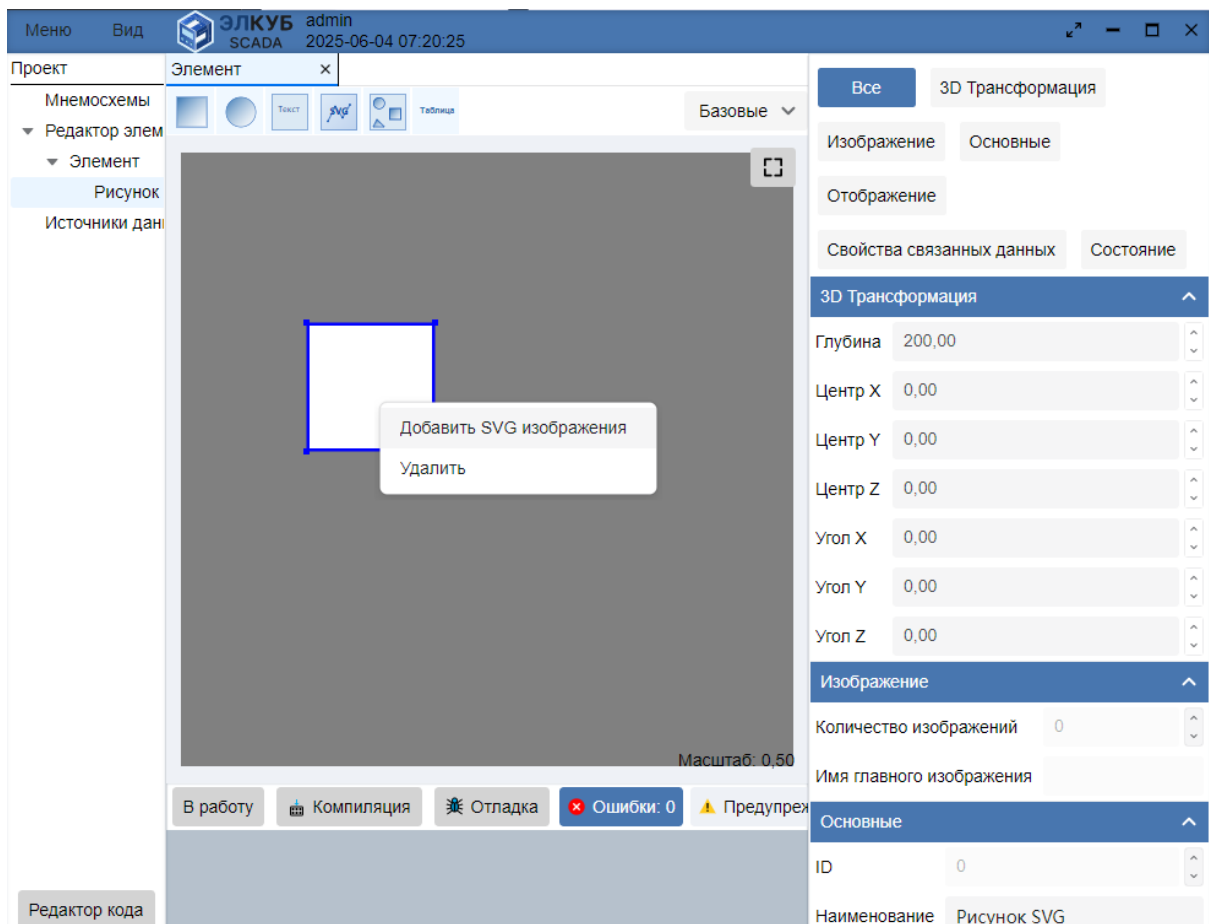
Жирный - применение жирности к тексту.

Курсив - применение курсива к тексту.

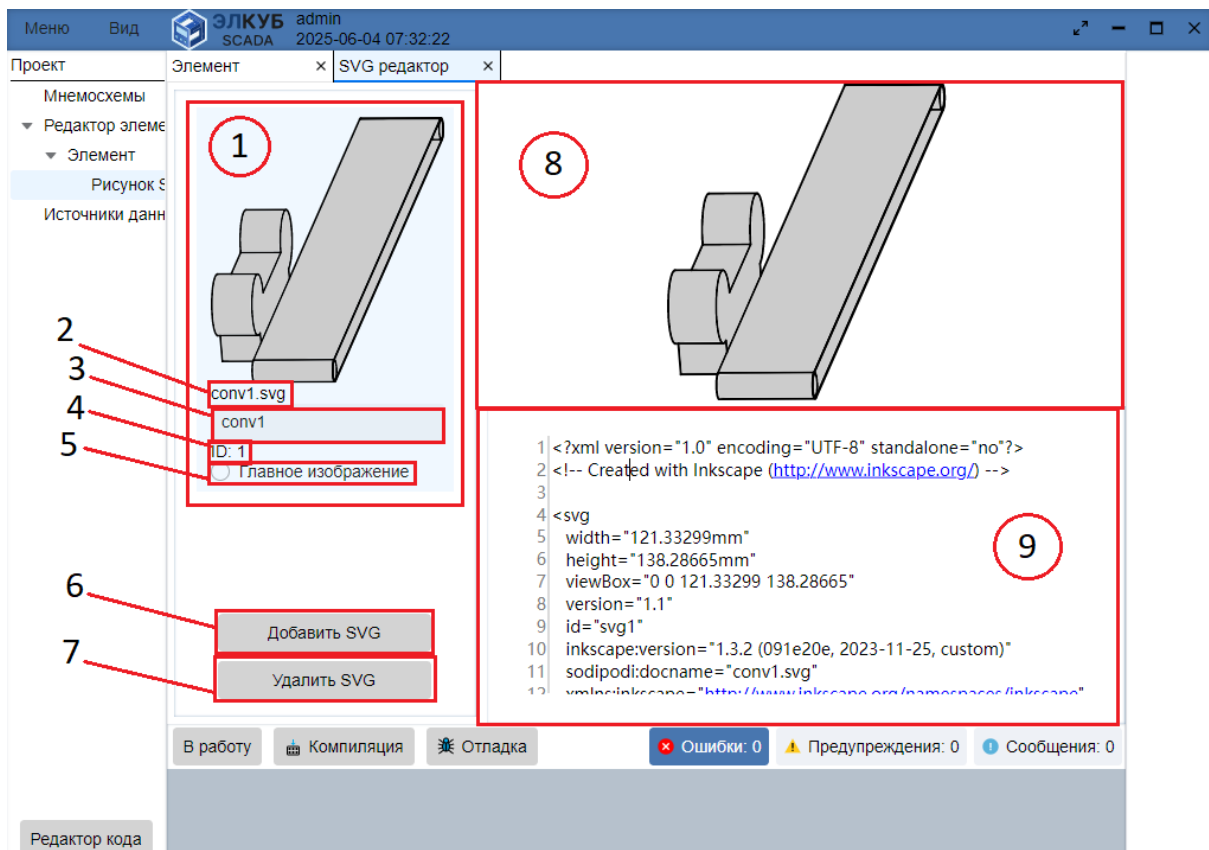


SVG Изображение

Примитив SVG изображение содержит уникальный пункт "Добавить SVG изображения" в меню, вызываемом нажатием ПКМ по примитиву.



При нажатии на пункт “Добавить SVG изображение” открывается редактор SVG изображений примитива.



- 1 - Добавленное через кнопку “Добавить SVG” изображение и его свойства.
 - 2 - conv1.svg - название открытого файла svg изображения.
 - 3 - conv1 - псевдоним изображения, с его помощью можно в скриптах примитива SVG изображения можно заменять картинку.
- Пример в скрипте примитива SVG изображения:
- ```
script.SetProperty(“MainImageByAlias”, “conv1”);
```
- 4 - ID: 1 - уникальный идентификатор картинки в рамках конкретного примитива. Используется по аналогии с псевдонимом из п.3. Пример в скрипте примитива SVG изображения: `script.SetProperty(“MainImageByID”, “1”);`
  - 5 - установить это изображение в качестве текущего отображаемого в редакторе у этого примитива. Одновременно только одно изображение из всех прикрепленных к примитиву может отображаться.
  - 6 - кнопка “Добавить SVG” используется для добавления SVG изображения, при нажатии открывается системный проводник файлов с предложением выбрать файлы SVG изображений.
  - 7 - кнопка “Удалить SVG” используется для удаления текущего выбранного в списке выше изображения.
  - 8 - отображение SVG изображения по коду из блока 9.

9 - код выбранного загруженного SVG изображения. При его изменении мгновенно применяются изменения к отображаемой картинке в блоке 8.

PNG изображение работает в точности так же, как SVG изображение, за исключением того, что вместо SVG изображений необходимо использовать PNG изображения, а также нет возможности посмотреть и изменить код PNG изображения.

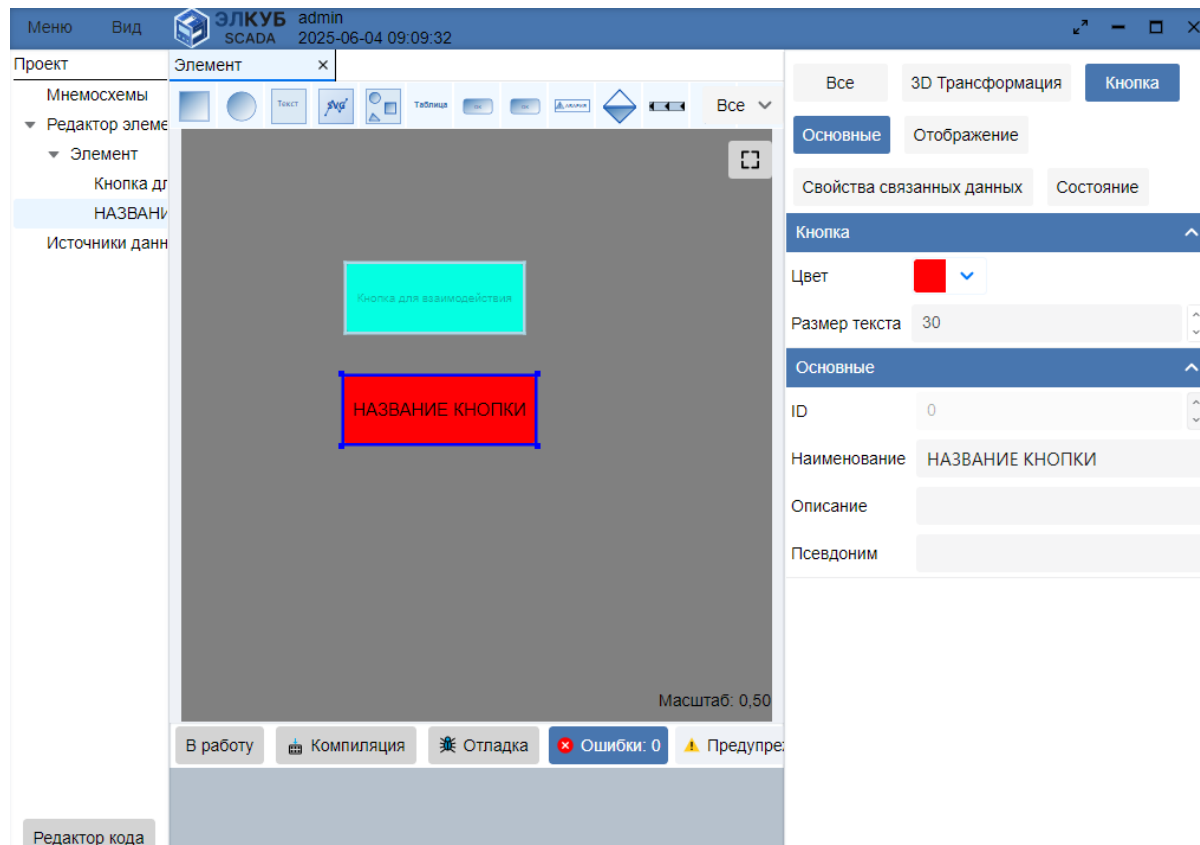
Примитив Кнопка используется для интерактивного взаимодействия пользователя и устройствами или прочими примитивами и элементами. Всего у примитива Кнопка два уникальных свойства в отдельной группе свойств “Кнопка”:

Цвет - цвет кнопки.

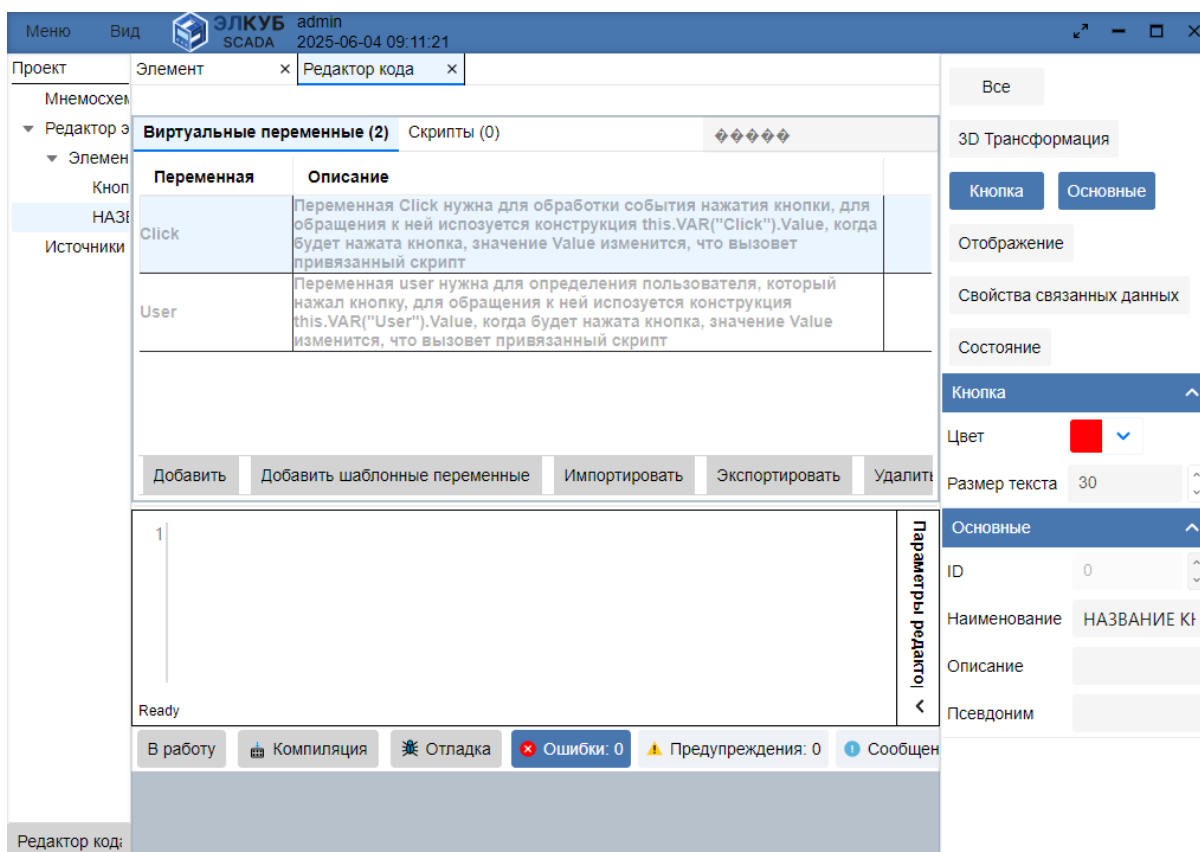
Размер текста - размер надписи на кнопке.

Надпись на кнопке соответствует наименованию кнопки (группа свойств “Основные”).

При наведении на примитив кнопки происходит его затемнение: в редакторе кнопка не является кликабельной и не вызывает исполнение соответствующих ей скриптов.



Примитив кнопки имеет автоматически генерируемые виртуальные переменные Click и User, которые используются для обработки нажатия по кнопке в режиме наблюдения. Переменная User статичная и соответствует ФИО текущего авторизованного пользователя, используется для логгирования нажатия. Переменная Click меняет своё значение при нажатии кнопки на 1, а потом сразу на 0. Таким образом обновление состояния переменной Click происходит дважды, это необходимо учитывать при помощи условного оператора в скрипте примитива Кнопка.



Пример скрипта, использующего переменную Click:

```
int ClickValue = (int)this.VAR("Click").Value;
string user = (string)this.VAR("User").Value;
if (ClickValue == 1 && EnabledValue == 1)
{
 script.SetRegister(3, 1);
 script.Log(0, "Инфо", $"Пользователь: {user}, нажата кнопка");
}
```

## Предупреждающее сообщение

Примитив предупреждающего сообщения обладает двумя уникальными свойствами.

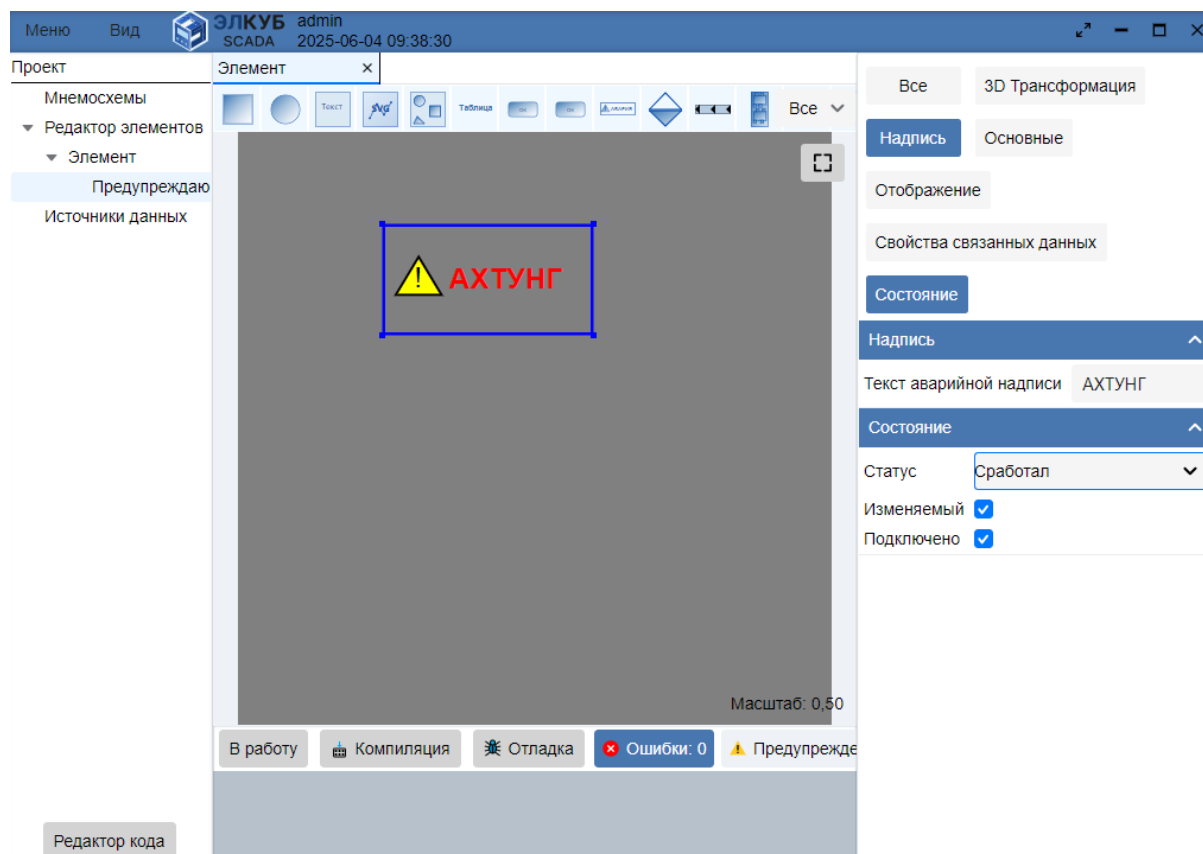
Текст аварийной надписи (группа “Надпись”) - текст внутри надписи.

Статус (группа “Состояние”) - текущее состояние надписи. Имеет всего три возможных состояния:

Неизвестно - базовое состояние, данных не поступало, цвет голубой.

Отключен - всё в норме, надпись полностью прозрачная.

Сработал - сработало событие, надпись красная и моргает раз в полсекунды.



## Датчик заштыбовки

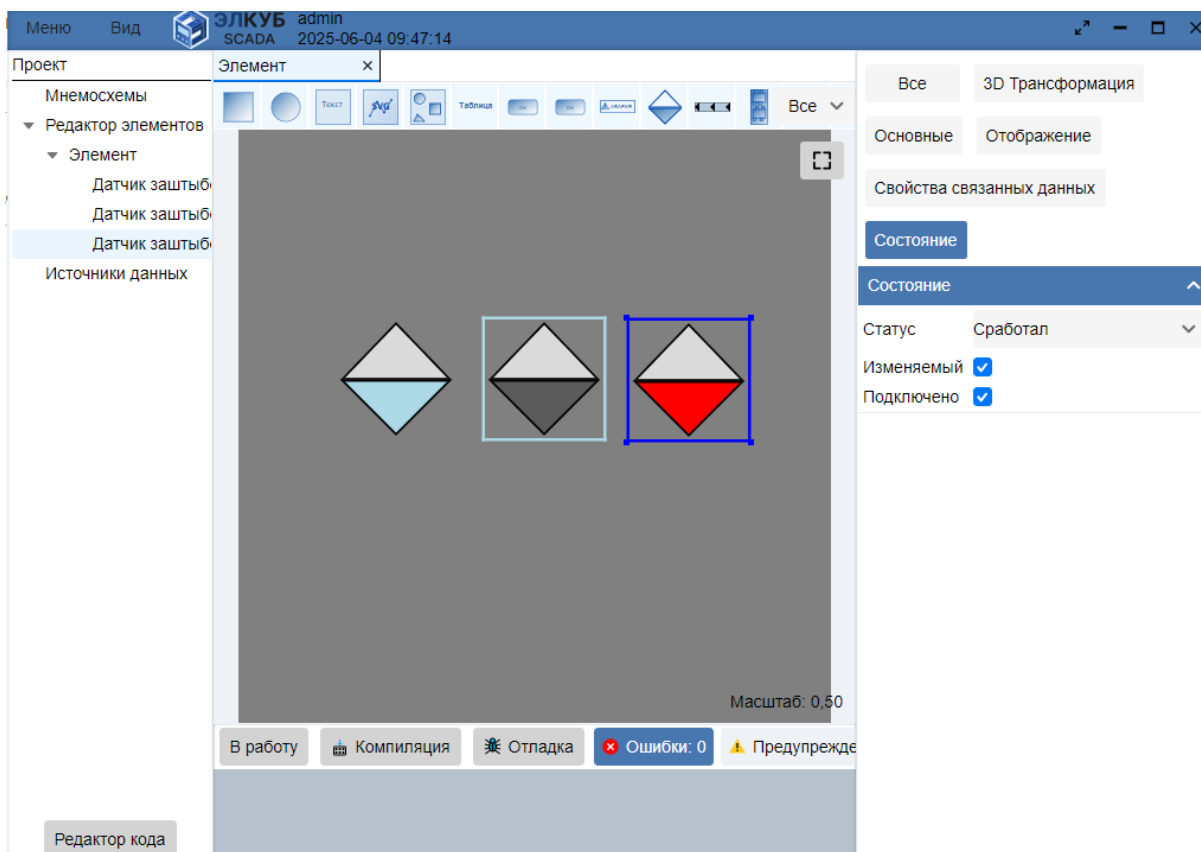
Датчик имеет одно уникальное свойство: Статус (группа “Состояние”).

Статус (группа “Состояние”) - текущее состояние датчика заштыбовки. Имеет всего три возможных состояния:

Неизвестно - базовое состояние, данных не поступало, цвет голубой.

Отключен - всё в норме, цвет чёрный/серый.

Сработал - сработало событие, датчик красный.



## Конвейер

Примитив Конвейер имеет уникальную группу свойств “Анимация” и свойство Статус (группа “Состояние”).

Статус (группа “Состояние”) - отображает текущее состояние конвейера.

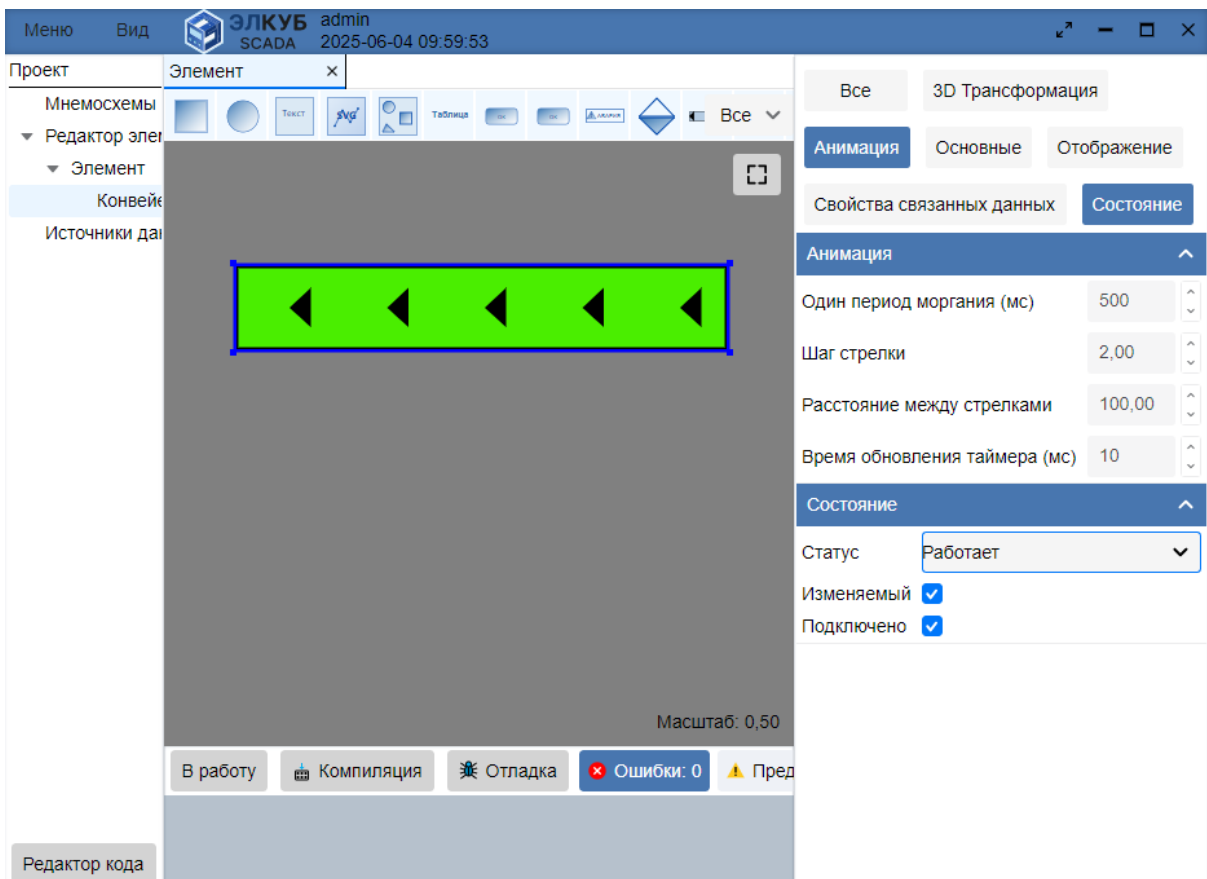
Группа “Анимация”:

Один период мигания (мс) - длительность одного мигания (от полной видимости до нулевой видимости). Используется для тех состояний конвейера, при котором он мигает (например, Запускается).

Шаг стрелки - значение смещения, применяющегося к каждой стрелке при обновлении таймера.

Расстояние между стрелками - расстояние между стрелками на конвейере.

Время обновления таймера (мс) - время, через которое будет происходить обновление отрисовки конвейера.



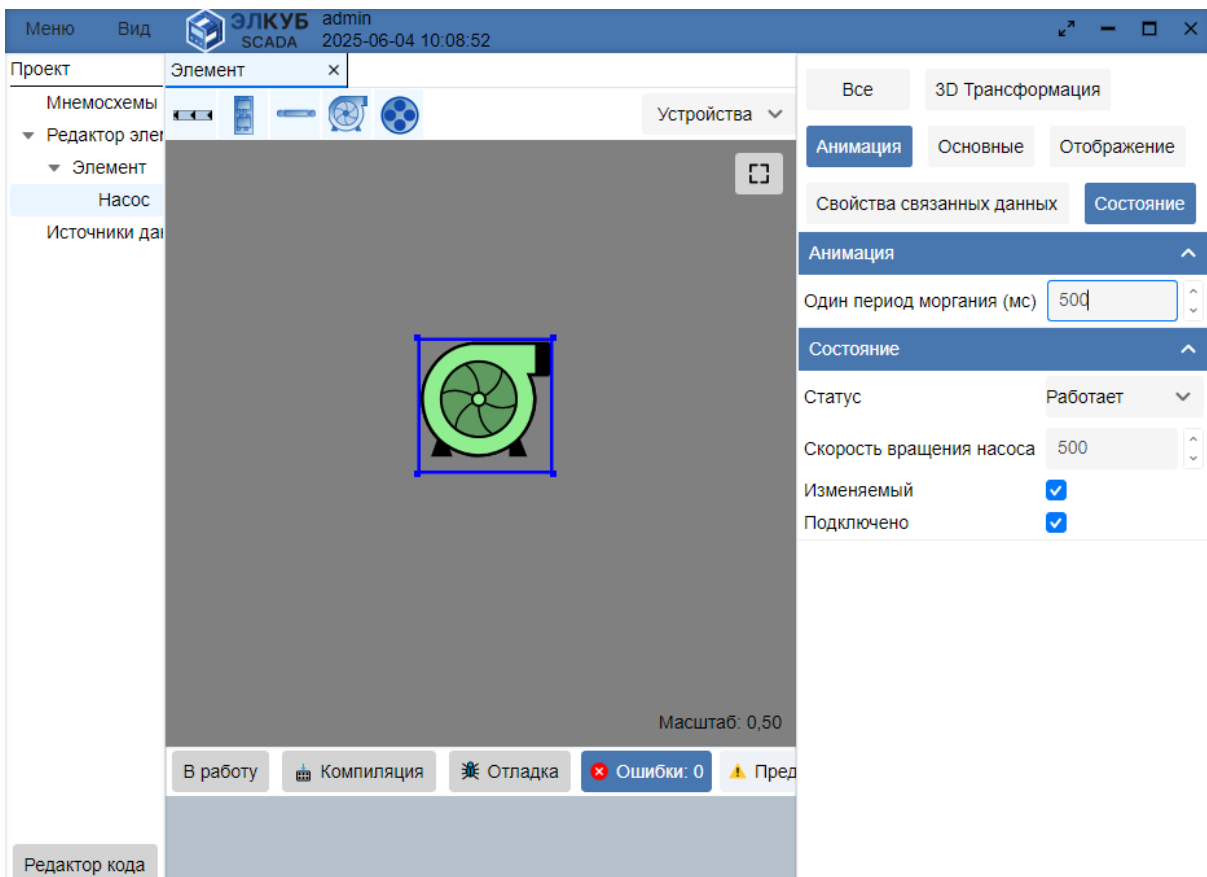
## Насос

Примитив Насос имеет уникальную группу свойств “Анимация” и свойство Статус (группа “Состояние”).

Статус (группа “Состояние”) - отображает текущее состояние насоса.

Скорость вращения насоса (группа “Состояние”) - скорость, с которой вращается ротор насоса в работе.

Один период моргания (мс) (группа “Анимация”) - длительность одного моргания (от полной видимости до нулевой видимости). Используется для тех состояний насоса, при котором он моргает (например, Запускается).



## Двигатель

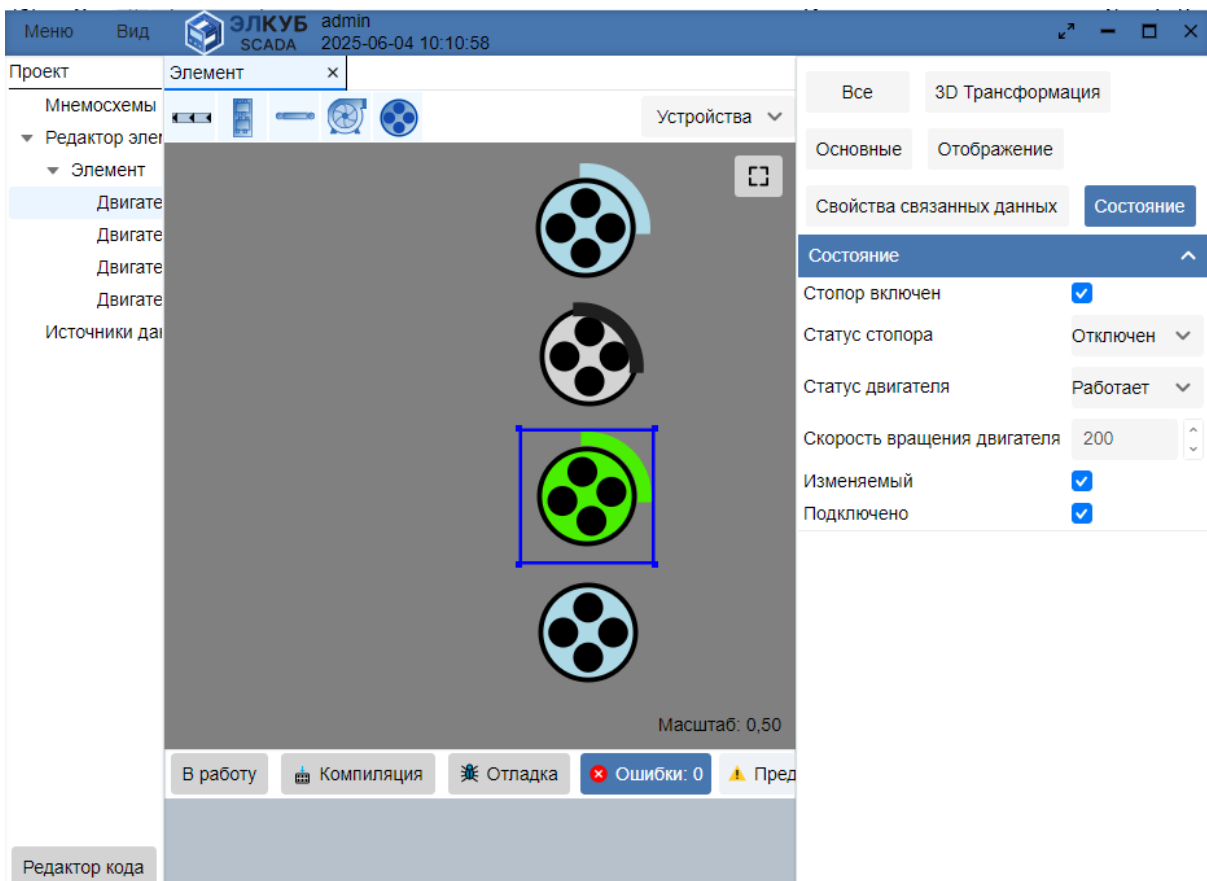
Двигатель имеет четыре уникальных свойства в группе “Состояние”.

Стопор включен - включить отображение стопора у двигателя.

Статус стопора - текущее состояние стопора.

Статус двигателя - текущее состояние двигателя.

Скорость вращения двигателя - скорость вращения ротора двигателя в работе.

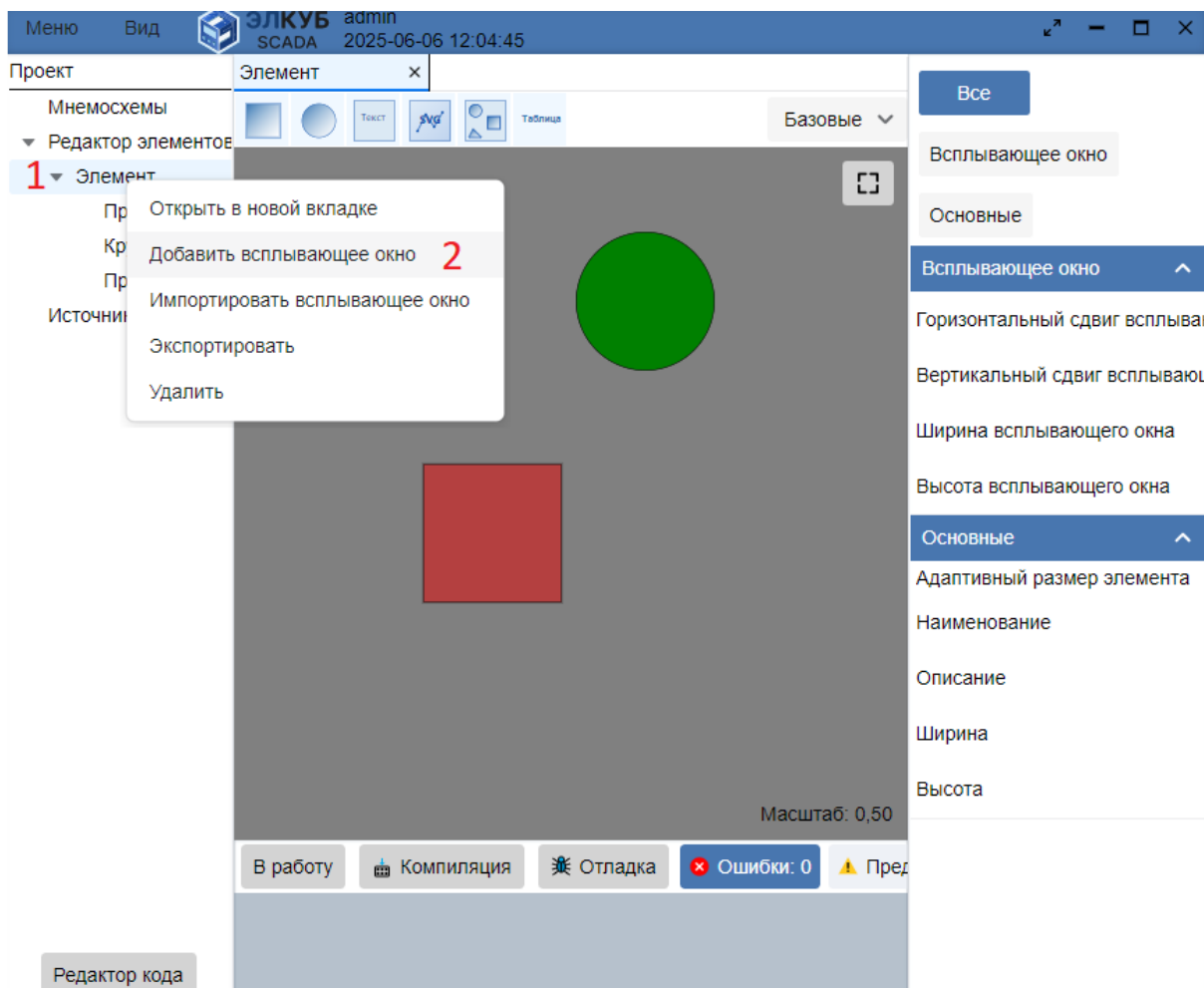


## ***Создание анимации***

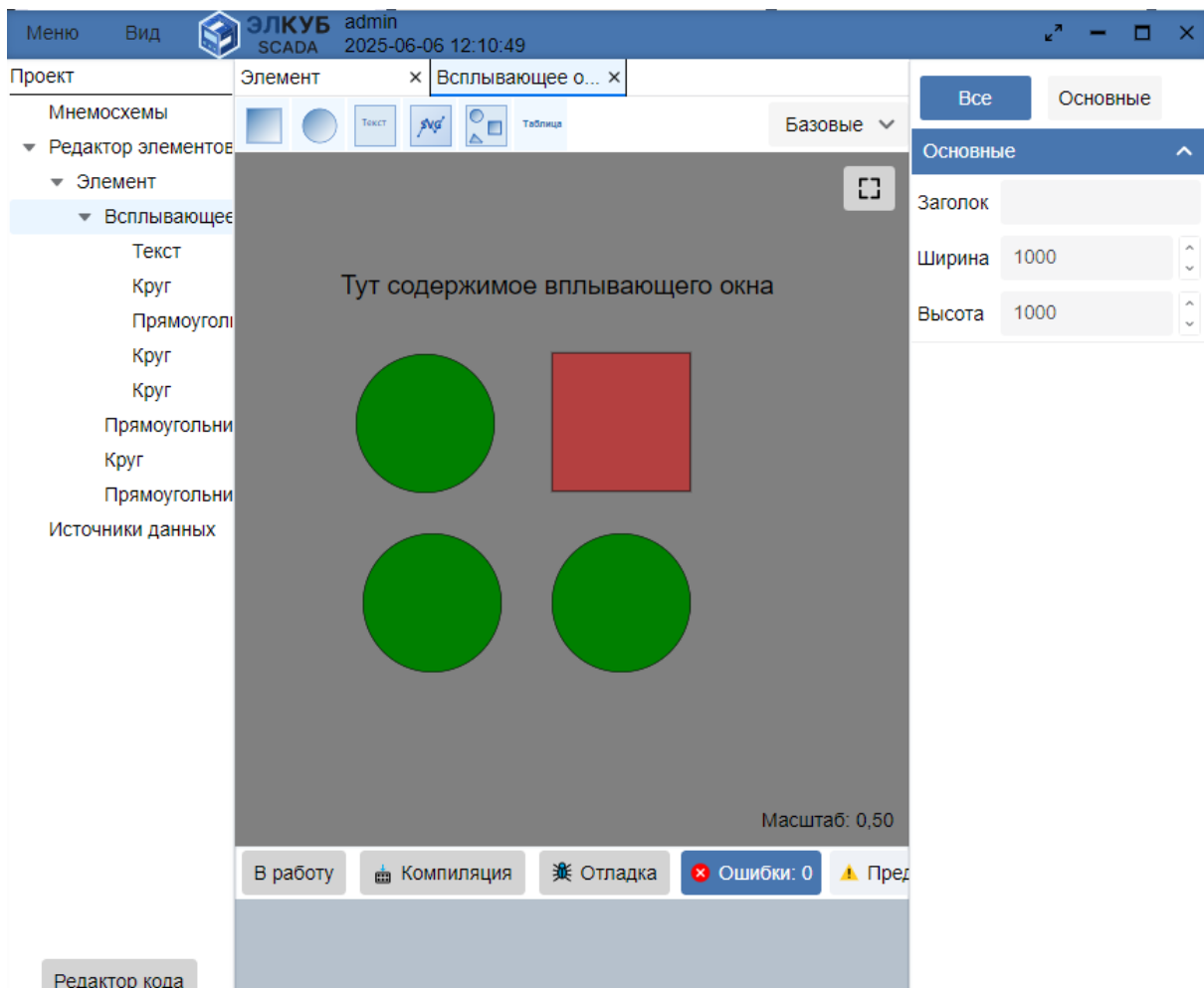
## ***Преобразование элементов***

## ***Добавление всплывающего окна для пользовательских элементов***

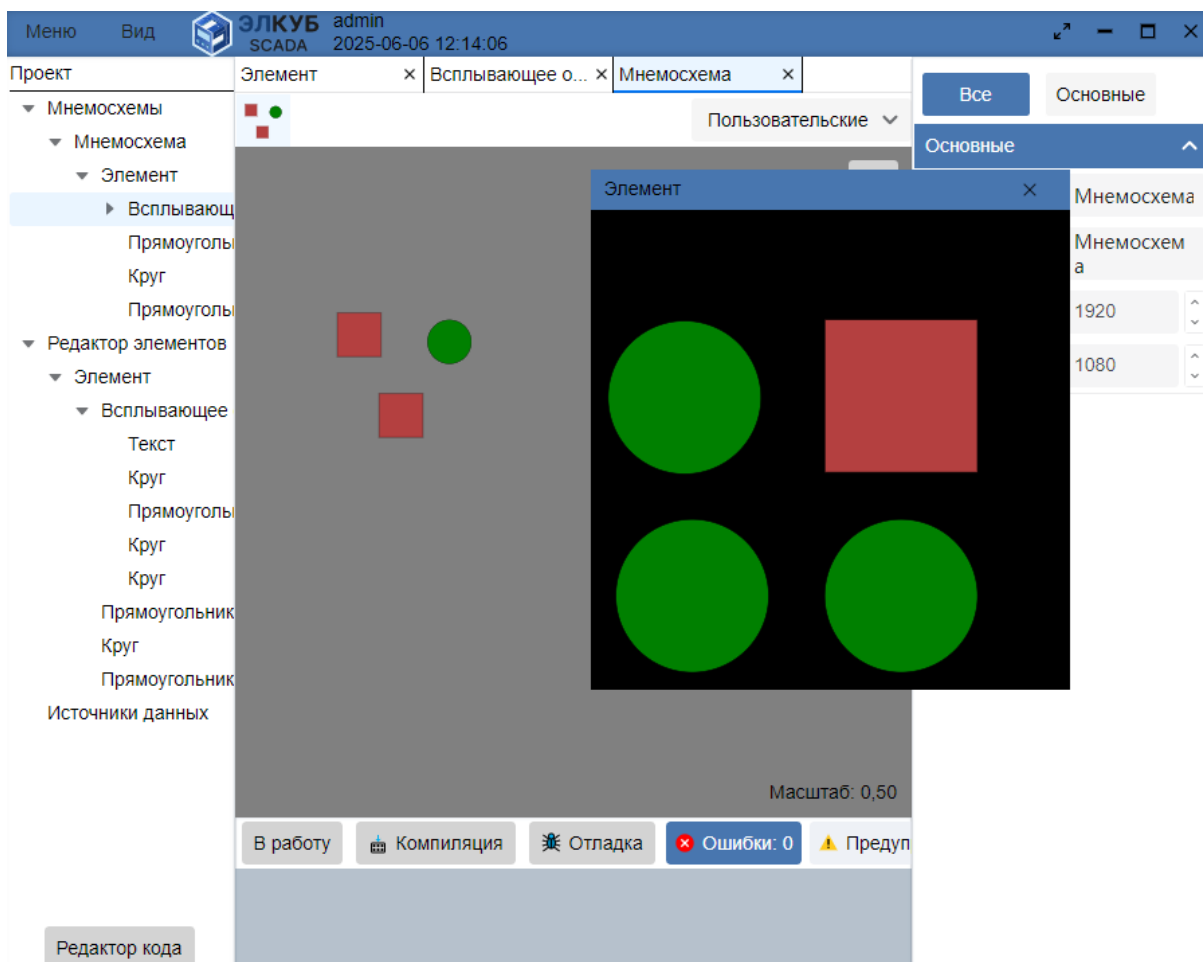
Для добавления всплывающего окна необходимо нажать ПКМ по элементу в дереве (1) проекта и выбрать пункт меню "Добавить всплывающее окно" (2).



После этого откроется вкладка редактирования всплывающего окна. Всплывающее окно изменяется точно таким же редактором, какой используется для создания элементов, отличие в свойствах.



Элемент, создаваемый в редакторе всплывающего окна, будет отображаться при двойном нажатии на элемент с этим всплывающим окном на мнемосхеме. Название окна будет совпадать с названием элемента, от которого оно вызвано.



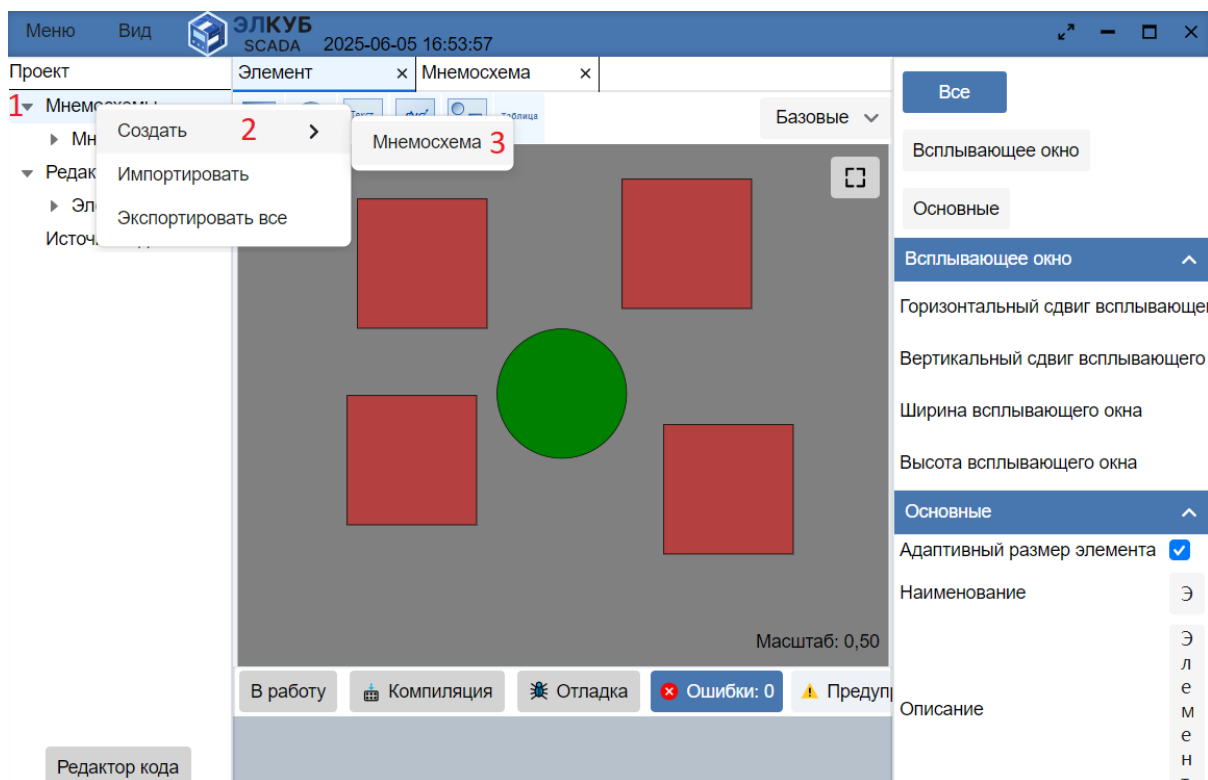
Всплывающее окно находится поверх всего содержимого приложения, поэтому его можно произвольно перемещать в приложении. Всплывающее окно не блокирует дальнейшее взаимодействие с элементами на мнемосхеме. Одновременно в приложении может быть открыто несколько всплывающих окон от разных элементов. Один и тот же элемент на мнемосхеме не может породить более одного всплывающего окна. Изменение размеров всплывающего окна происходит с помощью нажатия курсора по одному из углов окна и перемещении с зажатой ЛКМ. Растяжение происходит с сохранением пропорций содержимого всплывающего окна.

## Создание мнемосхем

### *Редактор мнемосхем*

Для создания мнемосхемы в дереве проекта необходимо нажать правой кнопкой мыши по корневому элементу мнемосхем “Мнемосхемы”(1), после чего

выбрать “Создать”(2), “Мнемосхема”(3). После этого откроется вкладка с новой созданной мнемосхемой.



### ***Свойства мнемосхемы***

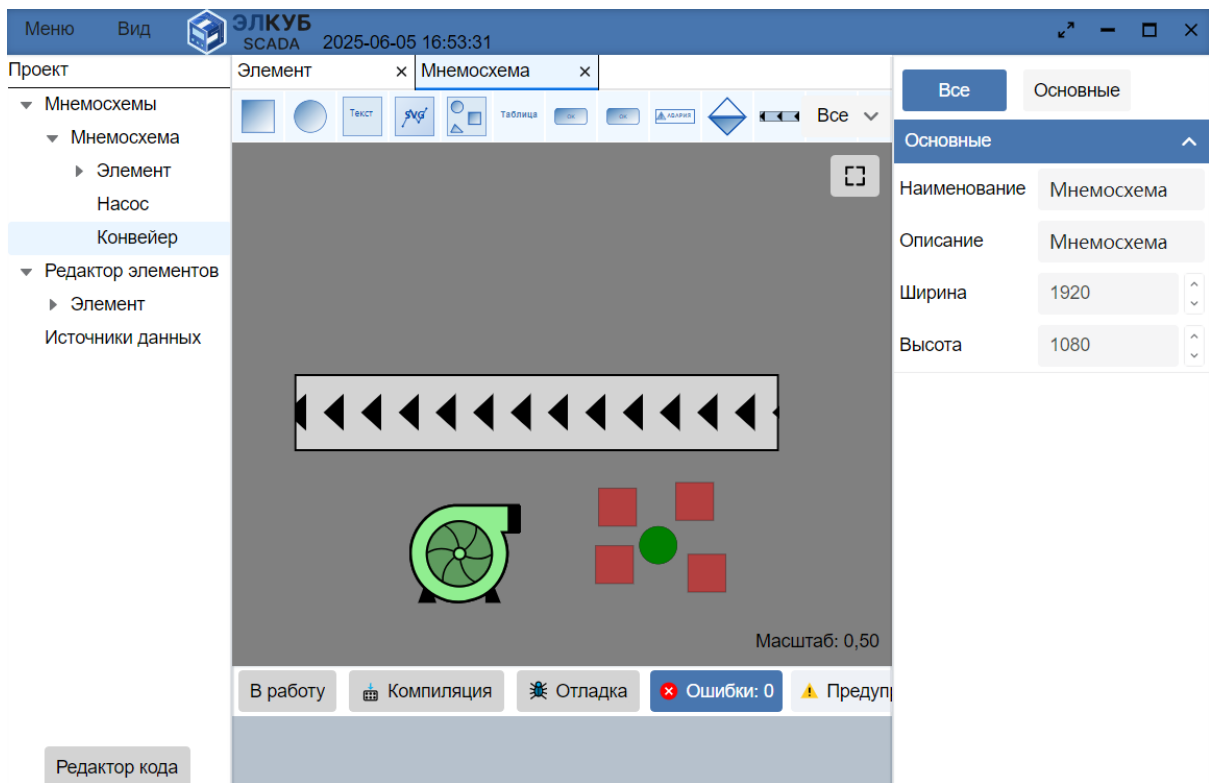
Мнемосхема обладает следующим набором свойств:

Наименование - название мнемосхемы, отображается в дереве проекта.

Описание - описание мнемосхемы.

Ширина - ширина канвы редактора мнемосхемы.

Высота - высота канвы редактора мнемосхемы.



По аналогии с редактором элемента, в редакторе мнемосхемы сверху от рабочей области расположена панель, в которой находятся и примитивы, и добавленные элементы. Размещение элементов на мнемосхеме происходит по тому же принципу, что и расположение примитивов, путём перетаскивания выбранного элемента из панели в область редактора.

### ***Базовые примитивы***

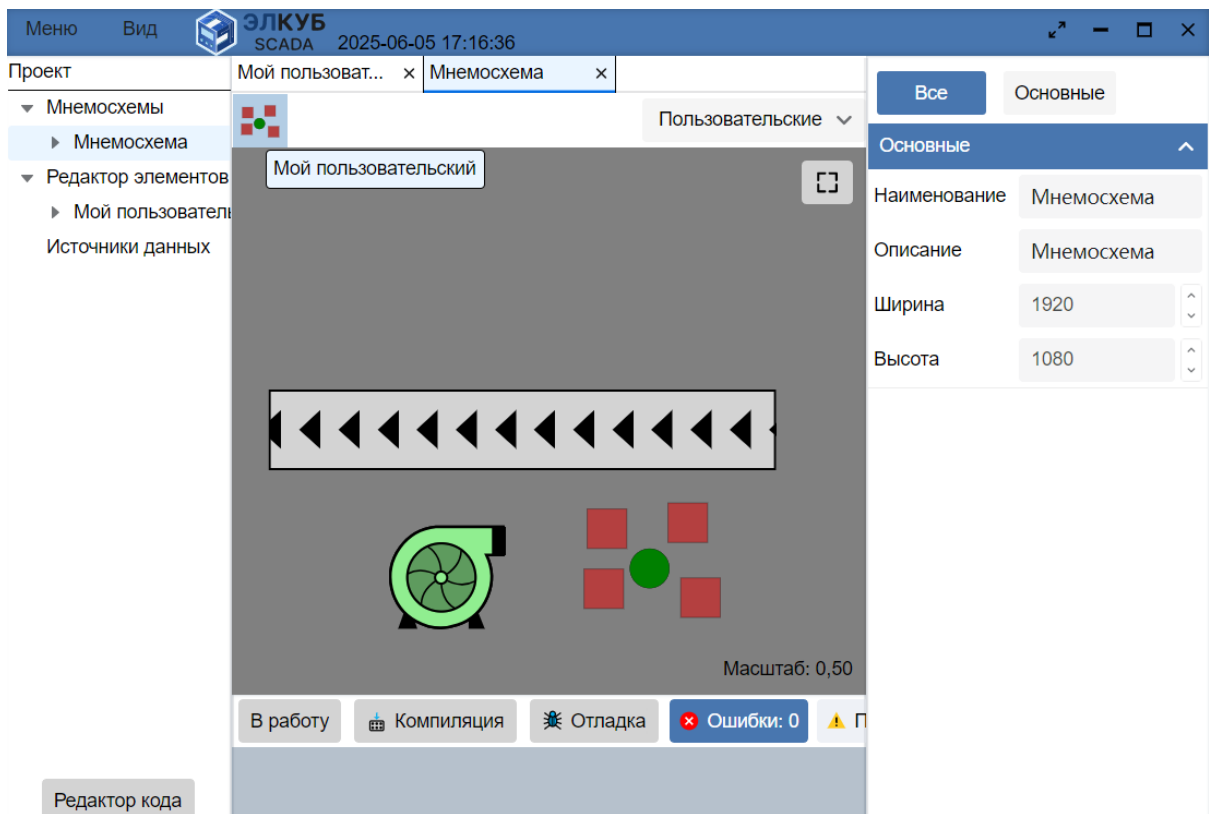
Список базовых примитивов, которые можно использовать на мнемосхеме, полностью совпадает со списком примитивов редакторе элемента.

### ***Стандартные элементы***

Стандартные элементы находятся на панели над рабочей областью.

### ***Пользовательские элементы***

Добавленные пользователем элементы можно увидеть на панели над рабочей областью в группах “Все” и “Пользовательские”.

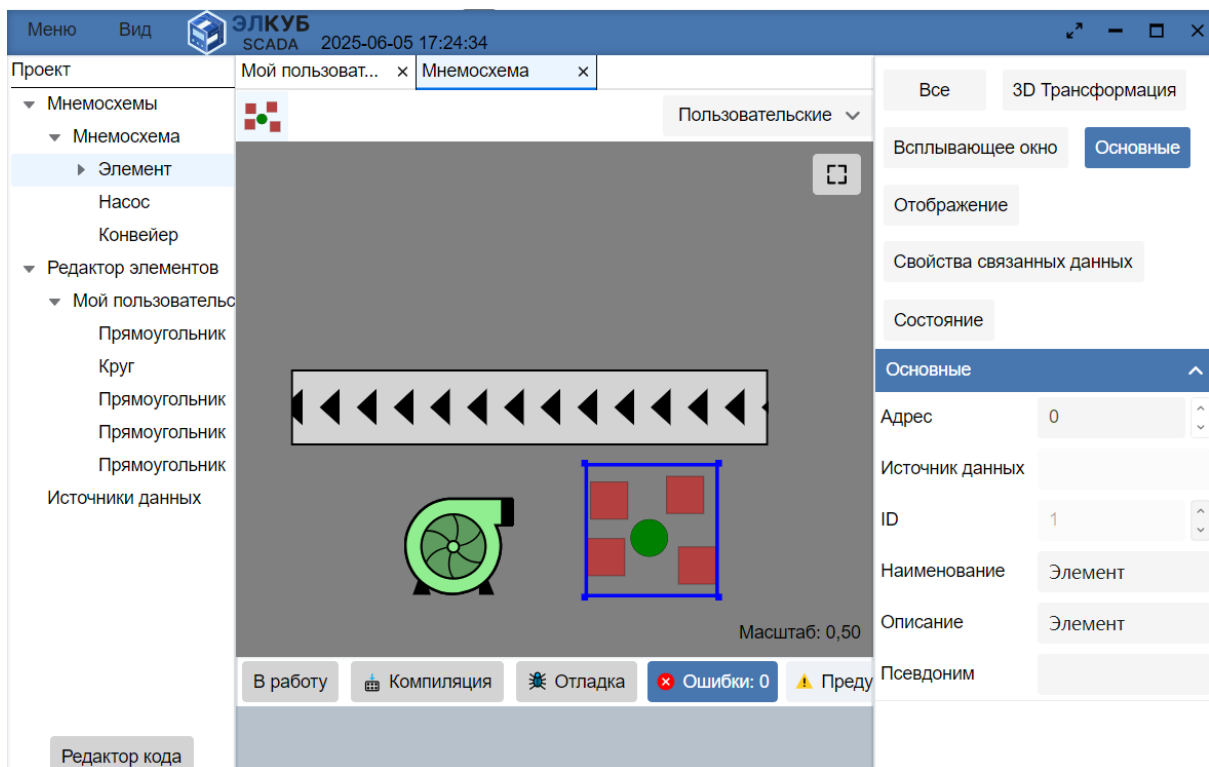


## ***Изменение свойств элементов***

Элемент, размещенный на мнемосхеме, имеет тот же набор свойств, что и примитив в редакторе элемента, с добавлением некоторых дополнительных свойств, связанных с подключением элемента к физическому устройству на линии. Эти свойства находятся в группе “Основные”.

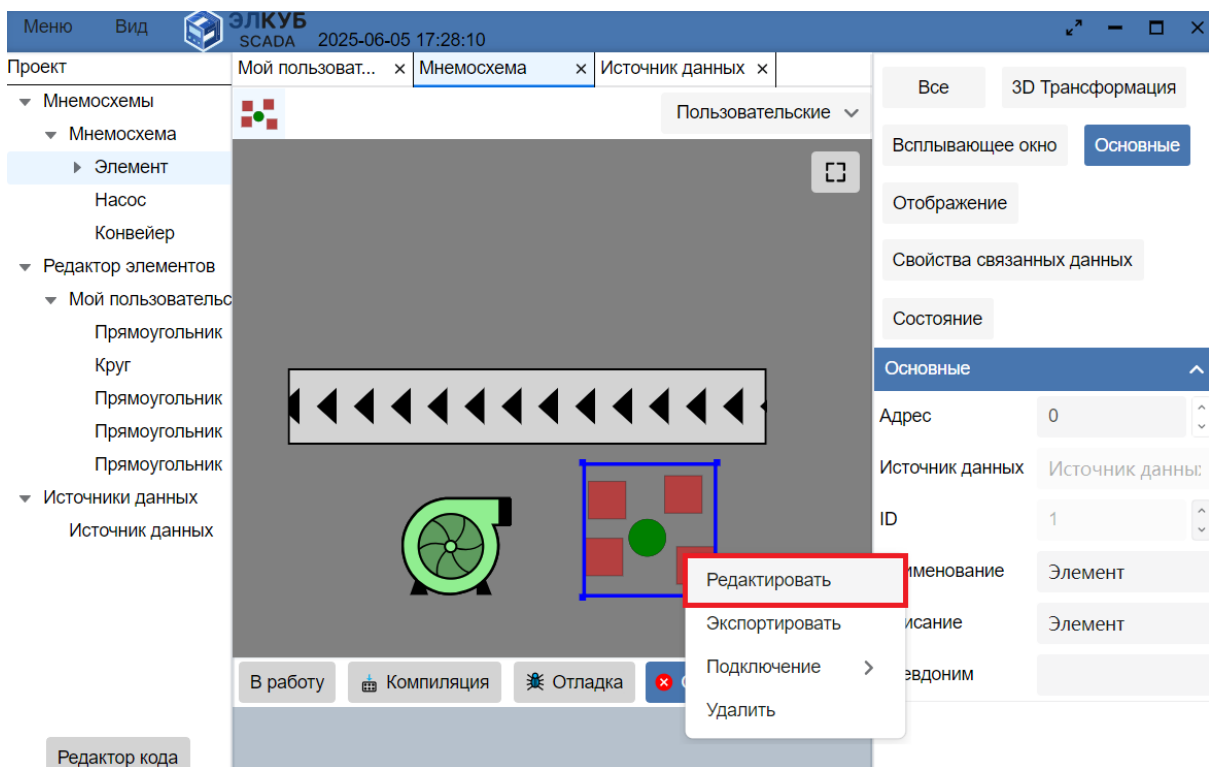
Адрес - адрес устройства на линии.

Источник данных - неизменяемое пользователем свойство, описывающее источник данных, к которому подключен элемент.



## Редактирование элементов на мнемосхеме

Для редактирования элемента, уже добавленного на мнемосхему, необходимо нажать по выбранному элементу ПКМ и выбрать в появившемся меню опцию “Редактировать”.

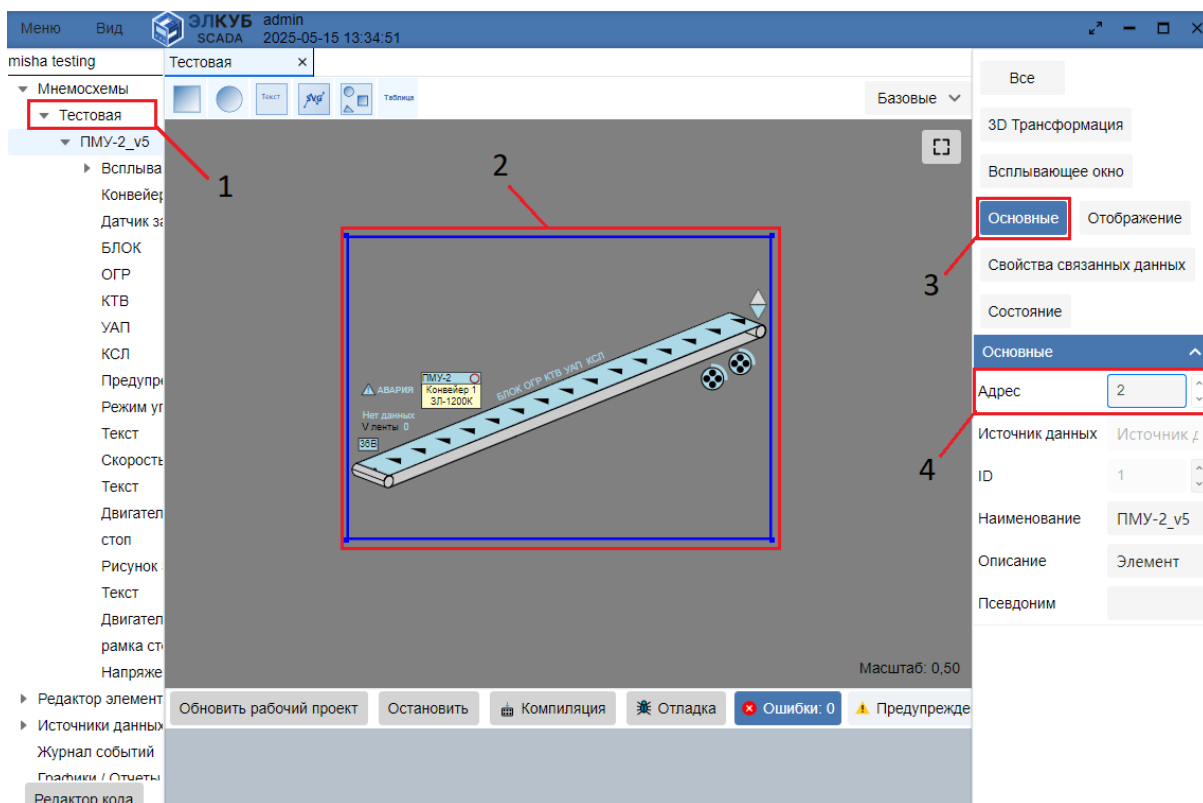


После этого в отдельной вкладке откроется редактор для изменения выбранного элемента. Важно помнить, что изменения, вносимые таким способом, будут применены только к выбранному элементу на мнемосхеме, все остальные элементы, которые были или будут размещены на мнемосхеме с использованием того же шаблона, что и у отредактированного элемента на мнемосхеме, останутся без изменений.

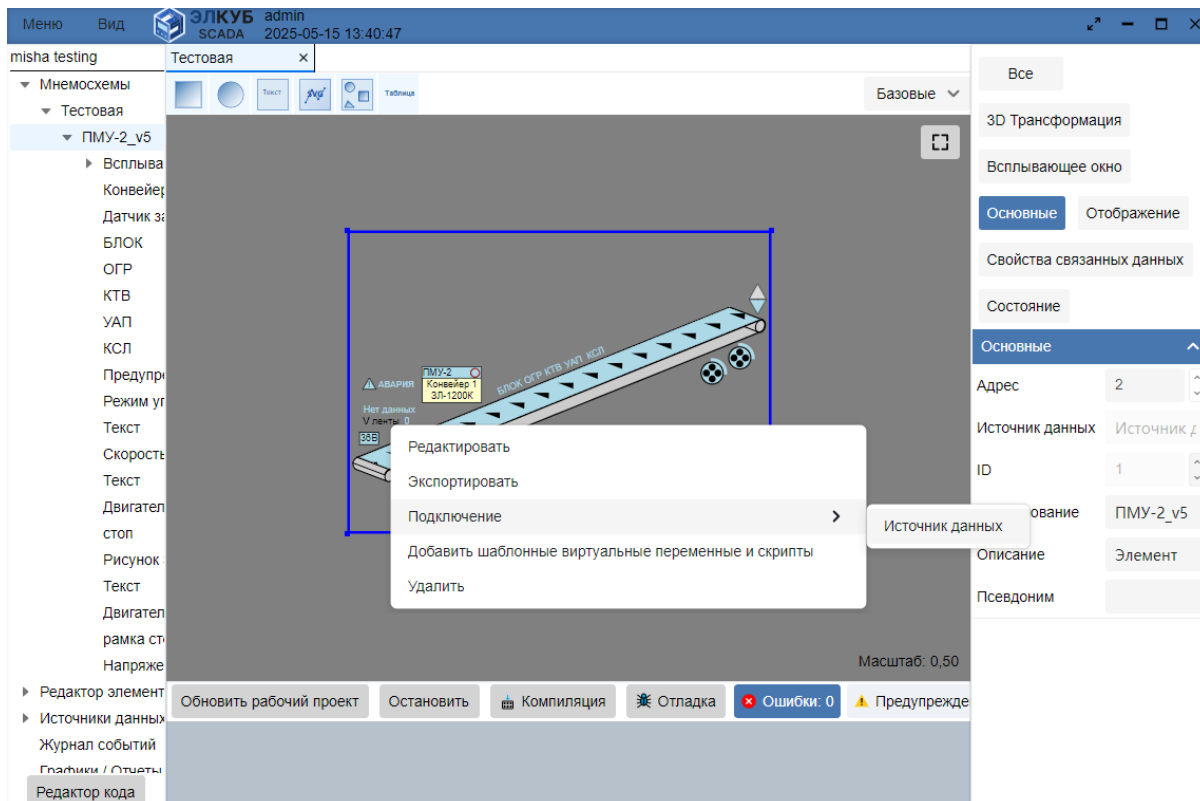
## Подключение к источникам данных

Для подключения к источнику данных элемента, необходимо следующее начальное условие: существует источник данных, к которому мы хотим подключить элемент, мнемосхема и элемент на мнемосхеме, который мы хотим подключить к источнику данных. Далее необходимо выполнить следующую последовательность:

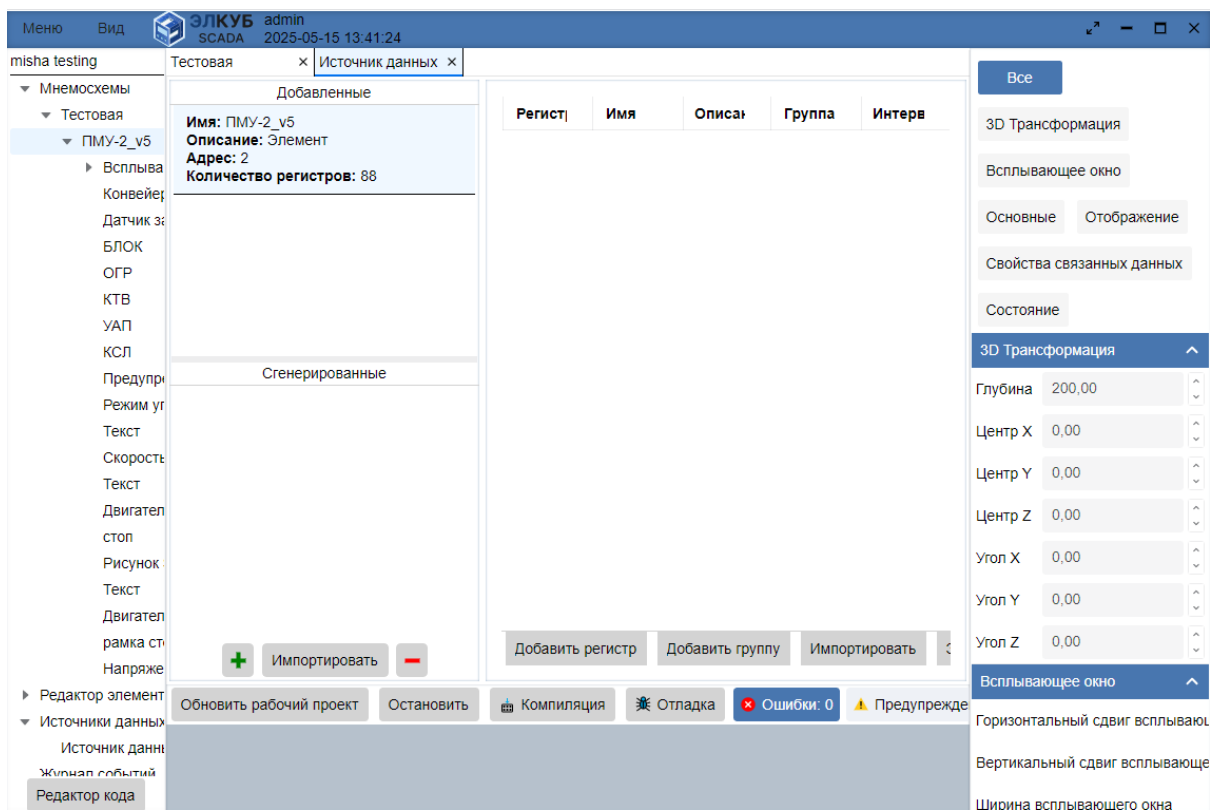
- 1) Задаём адрес устройства на линии. Для этого: в дереве проекта открываем мнемосхему (1), на которой находится элемент. Нажимаем ЛКМ по элементу (2), которому мы хотим задать адрес. На панели свойств появится свойства выбранного элемента. Выбираем Основные (3) свойства. Устанавливаем адрес (4) устройства на линии согласно адресу физического устройства.



- 2) Подключаем элемент к источнику данных. Для этого: нажимаем по подключаемому элементу ПКМ, в появившемся меню выбираем Подключить. Далее выбираем необходимый источник данных и нажимаем на него:



После этого элемент будет подключен к выбранному источнику данных. Убедиться в этом можно, посмотрев список подключенных к источнику данных устройств, выбрав источник данных в дереве проекта:



## Создание и настройка источника данных

### *Типы источников данных*

Источники данных типизируются по способу общения с физическими устройствами (преобразователем интерфейсов, МОХА, и т.д.).

Все типы имеют общие свойства:

Время ожидания ответа (мс) - сколько мс будет ожидаться ответ от устройства на запрос регистров.

Интервал опроса (мс) - время паузы после опроса всех устройств перед повторным опросом всех устройств.

Максимальное число ошибок - макс число ошибок устройства до фиксирования его состояния как отключенного (Нет связи).

Время отключения устройства (с) - сколько секунд устройство не будет опрашиваться при отключении.

Доступно 5 типов источников данных:

RTU - подключение источника данных при помощи COM порта к удалённому терминальному блоку, связывающему множество устройств.

Имеет уникальные свойства:

COM Port - номер COM порта у компьютера, на котором запущена SCADA.

Скорость передачи данных - BAUD Rate COM порта компьютера, на котором запущена SCADA (например, 115200).

UDP, TCP - эти два типа источников данных используются при получении данных с http сервера по протоколу UDP или TCP соответственно. Для работы с источниками данных таких типов необходимо указать:

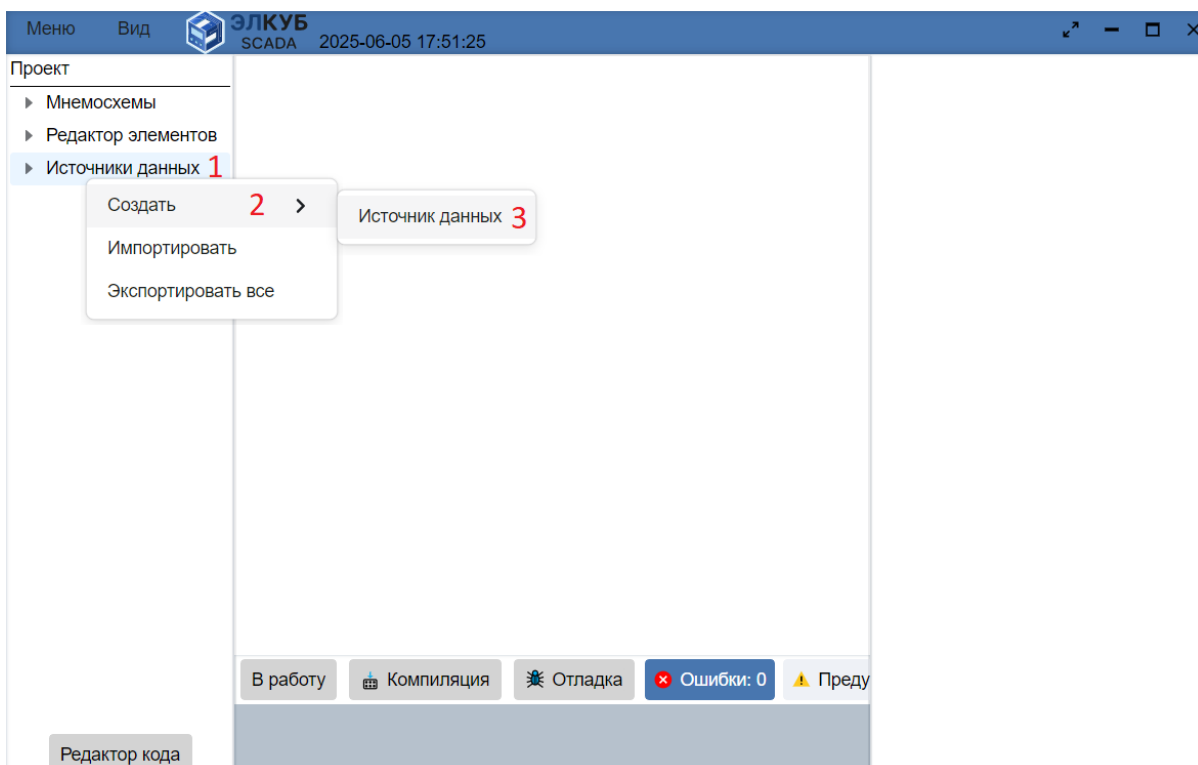
IP адрес - адрес сервера.

IP порт - порт сервера.

RTUoverUDP, RTUoverTCP - эти типы используются при взаимодействии с физ. устройствами через промежуточное устройство (терминальный блок), объединяющее все физ. устройства в линию по протоколам UDP и TCP соответственно.

### ***Добавление нового источника данных***

Для добавления нового источника данных, необходимо в дереве проекта нажать ПКМ по корневому элементу источников данных “Источники данных” (1), после чего в появившемся меню выбрать опцию “Создать” (2), “Источник данных” (3). После этого откроется вкладка с новой созданным источником данных.



## ***Добавление устройства в источник данных типа Modbus***

## **Добавление скриптов автоматизации**

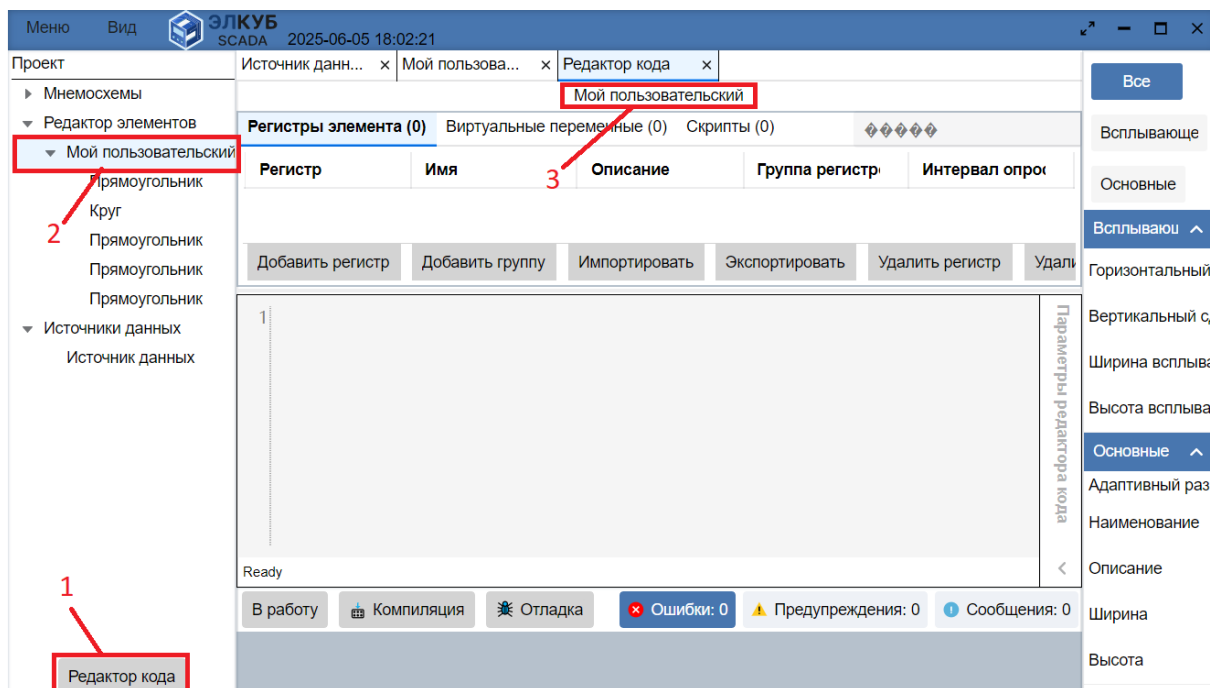
### ***Редактор кода***

Редактор кода предназначен для создания и описания регистров элементов, виртуальных переменных и скриптов. В верхней части редактора кода указывается путь в проекте до элемента, чьи регистры, виртуальные переменные и скрипты сейчас рассматриваются. Регистрами могут обладать только элементы, подключаемые к источникам данных. Прimitives, мнемосхемы, проекты не обладают регистрами.

### ***Общие принципы***

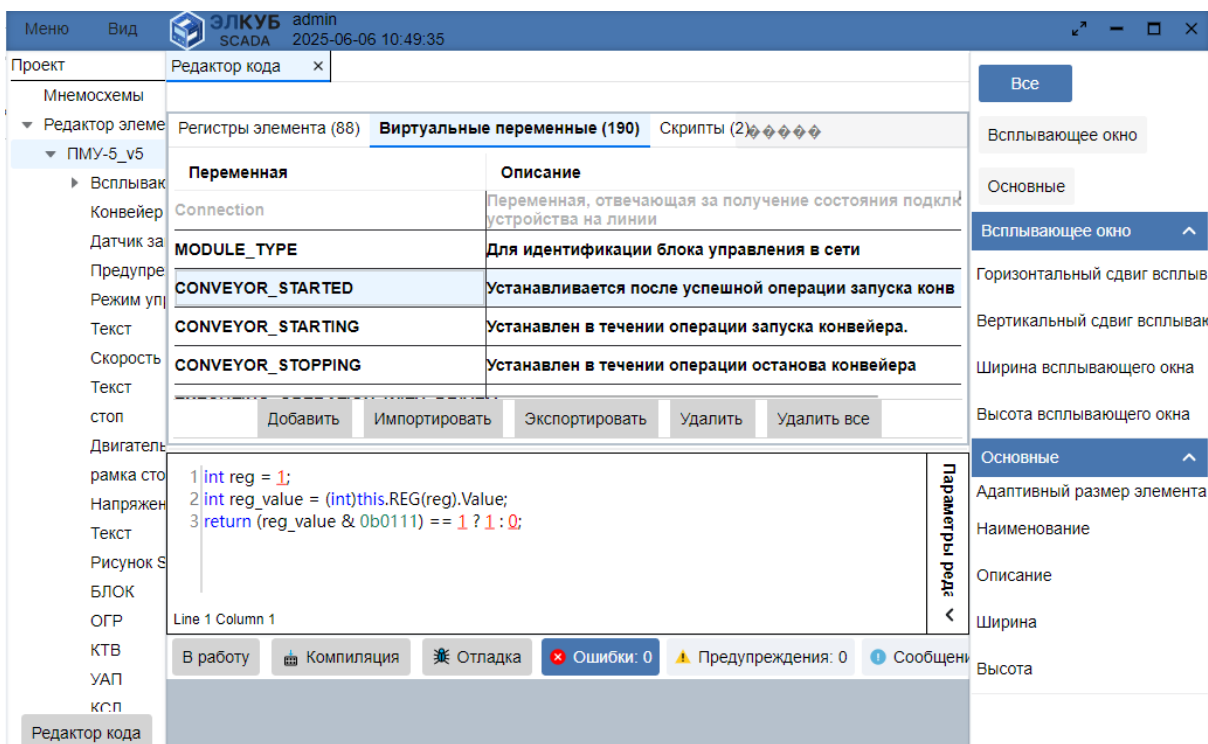
Для добавления регистров, виртуальных переменных и скриптов нужно открыть редактор кода (1) и выбрать в дереве проекта необходимый элемент (2). После этого в заголовке открывшегося редактора кода отобразится название

выбранного элемента. Дальнейшие изменения в подвкладках регистров, виртуальных переменных и скриптов будут применяться к выбранному элементу.



## Регистры

Регистры можно указать только у элементов, размещённых на мнемосхеме, или к шаблону элемента, создаваемого в редакторе элемента. Регистры должны фактически совпадать с теми регистрами, которые есть у физического устройства, связанного с элементом. Добавленные регистры будут опрашиваться на линии, а их значение можно получить в скриптах или виртуальных переменных элемента при помощи следующей инструкции: `int reg_value = (int)this.REG(reg).Value;`



## Виртуальные переменные

Виртуальные переменные предназначены для хранения данных и последующего использования в скриптах. В коде виртуальных переменных можно выполнять логику преобразования данных, получаемых из различных источников. Виртуальная переменная не обязана иметь код, если используется без дополнительных зависимостей от регистров или других виртуальных переменных. Если в виртуальной переменной нужно делать преобразования на другими данными (например, при чтении регистра в коде виртуальной переменной и сохранении значения в эту виртуальную переменную), нужно обязательно выполнять return операцию, иначе значение, вычисленное в коде виртуальной переменной, не будет сохранено в ней. Обращение к виртуальным переменным в других скриптах или виртуальных переменных происходит по их имени, поэтому следует давать осмысленные названия.

Помимо основного значения, виртуальная переменная имеет “старое” значение (OldValue), которое может быть использовано в ситуациях, когда необходимо отслеживать именно событие изменения переменной с одного значения на другое, например, при переключении состояния примитива. Получить старое значение можно аналогично основному, использовав OldValue

вместо Value. Важно помнить, что при первом вызове обновления виртуальной переменной, для которой важно отслеживать OldValue, значение OldValue будет равно null.

## **Скрипты**

Для каждого элемента или примитива можно прописать набор скриптов для управления отображением. В скрипте могут быть использованы виртуальные переменные того же объекта, в котором используется скрипт, или выше по иерархии объектов (проект <- мнемосхема <- пользовательский элемент <- примитив), либо от любого объекта проекта через уникальный псевдоним. Вызов скриптов происходит только при изменении виртуальных переменных, от которых зависит скрипт.

## **Синтаксис**

Для выполнения таких операций, как смена цвета примитива, изменение текста в текстовом блоке и т.д. существуют специальные команды, которые можно использовать в скриптах объекта.

### **Список команд скриптов для управления логикой отображения**

1. SetProperty(string propertyName, object value)

Вызов команды SetProperty позволяет изменить свойства отображаемого объекта.

Для этого нужно в скрипте вызвать команду:

```
script.SetProperty("NameOfProperty", "NewValue");
```

Доступные свойства напрямую зависят от объекта, в котором используется эта команда.

Общие свойства примитивов и элементов

"Height" - высота объекта, double (-inf, +inf)

"Width" - ширина объекта, double (-inf, +inf)

"CoordinateX" - координата X объекта, double (-inf, +inf)

"CoordinateY" - координата Y объекта, double (-inf, +inf)

"Rotation" - угол поворота объекта, int [0, 360)

"IsHorizontalMirrored" - отразить элемент по горизонтали, boolean [true, false]

"IsVerticalMirrored" - отразить элемент по вертикали, boolean [true, false]

"ZIndex" - индекс объекта по оси Z, int (-inf, +inf)

"CurrentConnectionState" - отобразить состояние Нет связи, boolean [true, false]

Специфичные для конкретных примитивов свойства

- круг

"StrokeColor" - цвет границы круга, примеры значения: "Red", "#FFFFFF".

"StrokeThickness" - толщина границы круга, int [0, +inf)

"Opacity" - прозрачность объекта, double [0, 1], 0 = полностью прозрачный.

"CurrentBackgroundColor" - цвет элемента, примеры значения: "Red", "#FFFFFF".

- квадрат

"StrokeColor" - цвет границы квадрата, примеры значения: "Red", "#FFFFFF".

"StrokeThickness" - толщина границы квадрата, int [0, +inf)

"Opacity" - прозрачность объекта, double [0, 1], 0 = полностью прозрачный.

"CurrentBackgroundColor" - цвет элемента, примеры значения: "Red", "#FFFFFF".

"RadiusX" - радиус скругления границы по оси X, int [0, +inf)

"RadiusY" - радиус скругления границы по оси Y, int [0, +inf)

- текст

"Text" - отображаемый текст, string

"AutoFontSize" - автоматическая подстройка размера шрифта, boolean [true, false]

"FontSize" - размер шрифта, int [1, +inf)

"FontFamilyName" - название используемого шрифта, string, пример: "Arial"

"ColorBlock" - цвет заднего фона текста, примеры значения: "Red", "#FFFFFF".

"ColorText" - цвет текста, примеры значения: "Red", "#FFFFFF".

- SVG изображение

"MainImageById" - установить SVG изображение по ID изображения, добавленного в редакторе SVG изображения, int [1, +inf). Если значение

отсутствует (не было корректно установлено при редактировании SVG примитива), то присвоение не работает.

“MainImageByAlias” - установить SVG изображение по псевдониму изображения, добавленного в редакторе SVG изображения, string. Если значение отсутствует (не было корректно установлено при редактировании SVG примитива), то присвоение не работает.

- PNG изображение

“MainImageById” - установить PNG изображение по ID изображения, добавленного в редакторе PNG изображения, int [1, +inf). Если значение отсутствует (не было корректно установлено при редактировании PNG примитива), то присвоение не работает.

“MainImageByAlias” - установить PNG изображение по псевдониму изображения, добавленного в редакторе PNG изображения, string. Если значение отсутствует (не было корректно установлено при редактировании PNG примитива), то присвоение не работает.

- Таблица

Строки в таблице имеют уникальные численные идентификаторы. Строки в таблицу добавляются по конкретному уникальному ID строки. Установить в конкретную строку значение можно следующим образом:

`script.SetProperty($"row:1", "{ 'operation': 'delete', 'id':1}");` - удалить из таблицы строку с ID = 1.

`script.SetProperty($"row:{startTableId+bit}", "{ 'operation': 'add', 'id':1, 'row': 'new:row:value'}");` - добавить значение “new:row:value” в строку с ID 1. Важно заметить, что значения ячеек в строке разделяются специальным разделителем, который можно задавать в свойствах примитива таблицы. Таким образом операция add добавит в таблицу из трёх столбцов, для которой задан разделитель “:”, строку new row value. Обращение по уникальным ID нужно для избежания конфликтов записи одних и тех же значений в таблицу.

- График

“NewValue” - новое значение для добавления на график, double, (-inf, +inf). На графике горизонтальная ось соответствует времени добавления нового значения на график.

- Кнопка

“Enabled” - разблокировать кнопку для возможности нажатия по ней, boolean [true, false]

- Поле ввода

- / -

-

- Предупреждающее сообщение

“CurrentAlarmMessageState” - текущее состояние предупреждающего сообщения, TriggerState [“Triggered” = 1 = “Сработал”,

“Disabled” = 2 = “Отключен”,

“Unknown” = 3 = “Неизвестно”].

Принимает одно из указанных трёх состояний.

- Датчик заштыбовки

“CurrentBlockageSensorState” - текущее состояние датчика заштыбовки, TriggerState [“Triggered” = 1 = “Сработал”,

“Disabled” = 2 = “Отключен”,

“Unknown” = 3 = “Неизвестно”].

Принимает одно из указанных трёх состояний.

- Конвейер

“ArrowStep” - шаг стрелок конвейера, double (0, +inf). От этого параметра зависит скорость движения стрелок конвейера.

“DistanceBetweenArrows” - расстояние между стрелками конвейера, double (0, +inf). Этим параметром можно регулировать количество отображаемых стрелок конвейера.

“CurrentConveyorState” - текущее состояние конвейера, MotionState [

“Unknown” = 1 = “Неизвестно”,

“Stop” = 2 = “Стоп”,

“Starting” = 3 = “Запускается”,  
“Work” = 4 = “Работает”,  
“Stopping” = 5 = “Останавливается”,  
“Error” = 6 = “Ошибка”]

Принимает одно из указанных шести состояний.

- Дробилка

“CurrentGrinderState” - текущее состояние дробилки, WorkState [

“Stop” = 1 = “Стоп”,  
“Work” = 2 = “Работает”,  
“Unknown” = 3 = “Неизвестно”]

Принимает одно из трёх указанных состояний.

- Лавный комбайн

“CurrentLavaHarvesterState” - текущее состояние лавного комбайна, WorkState [

“Stop” = 1 = “Стоп”,  
“Work” = 2 = “Работает”,  
“Unknown” = 3 = “Неизвестно”]

Принимает одно из трёх указанных состояний.

- Насос

“RotationSpeed” - скорость вращения насоса в режиме работы, double (0, +inf)

“CurrentPumpState” - текущее состояние насоса, MotionState [

“Unknown” = 1 = “Неизвестно”,  
“Stop” = 2 = “Стоп”,  
“Starting” = 3 = “Запускается”,  
“Work” = 4 = “Работает”,  
“Stopping” = 5 = “Останавливается”,  
“Error” = 6 = “Ошибка”]

Принимает одно из указанных шести состояний.

## 2. SetRegister(int registerId, int value)

Эта команда выполняет изменение заданного регистра на указанное значение.

Используется для передачи управляющих команд на физические устройства.

Пример: по нажатию на кнопку выполнять команду:

`script.SetRegister(3, 10);` - установить значение 10 в регистр 3. Если 3 - управляющий регистр устройства, предназначенный для управляющих команд, то его изменение вызовет выполнение инструкций, соответствующих команде 10.

### 3. `SetVarValue(VirtualVar var, object value)`

Установить значение `value` в виртуальную переменную `var`. Напрямую установить в виртуальную переменную значение в скрипте нельзя. Для изменения значения виртуальной переменной и скрипта необходимо использовать `SetVarValue`:

`script.SetVarValue(this.VAR("varToSet"), "123");` - установить в виртуальную переменную `varToSet` того же объекта, в котором находится скрипт, вызывающий `SetVarValue`, значение "123".

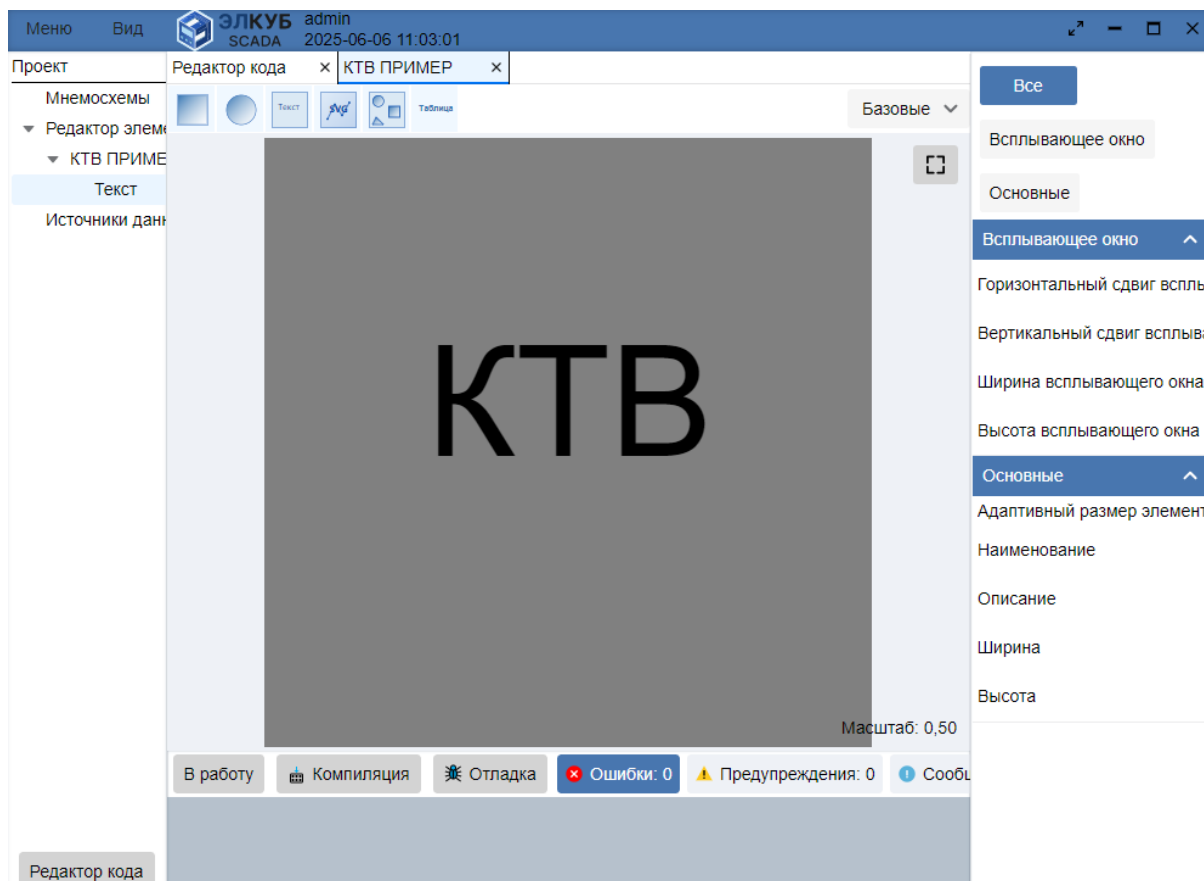
### 4. `Log(int priority, string type, string text, bool autokvit = false)`

Эта команда позволяет записать информацию в журнал событий. `priority` соответствует приоритету сообщения: 0 - инфо, 1 - предупреждение, 2 - авария. `type` - тип сообщения: инфо/предупреждение/авария. Такое разделение позволяет задавать произвольное количество приоритетов и соответствующих им типов. `text` - описание сообщения. `autokvit` - автоматически квитировать это сообщение.

`script.Log("1", "Предупреждение", "Ваше сообщение в журнал", false)` - добавить в журнал запись "Ваше сообщение в журнал" с уровнем приоритета 1 и типом Предупреждение.

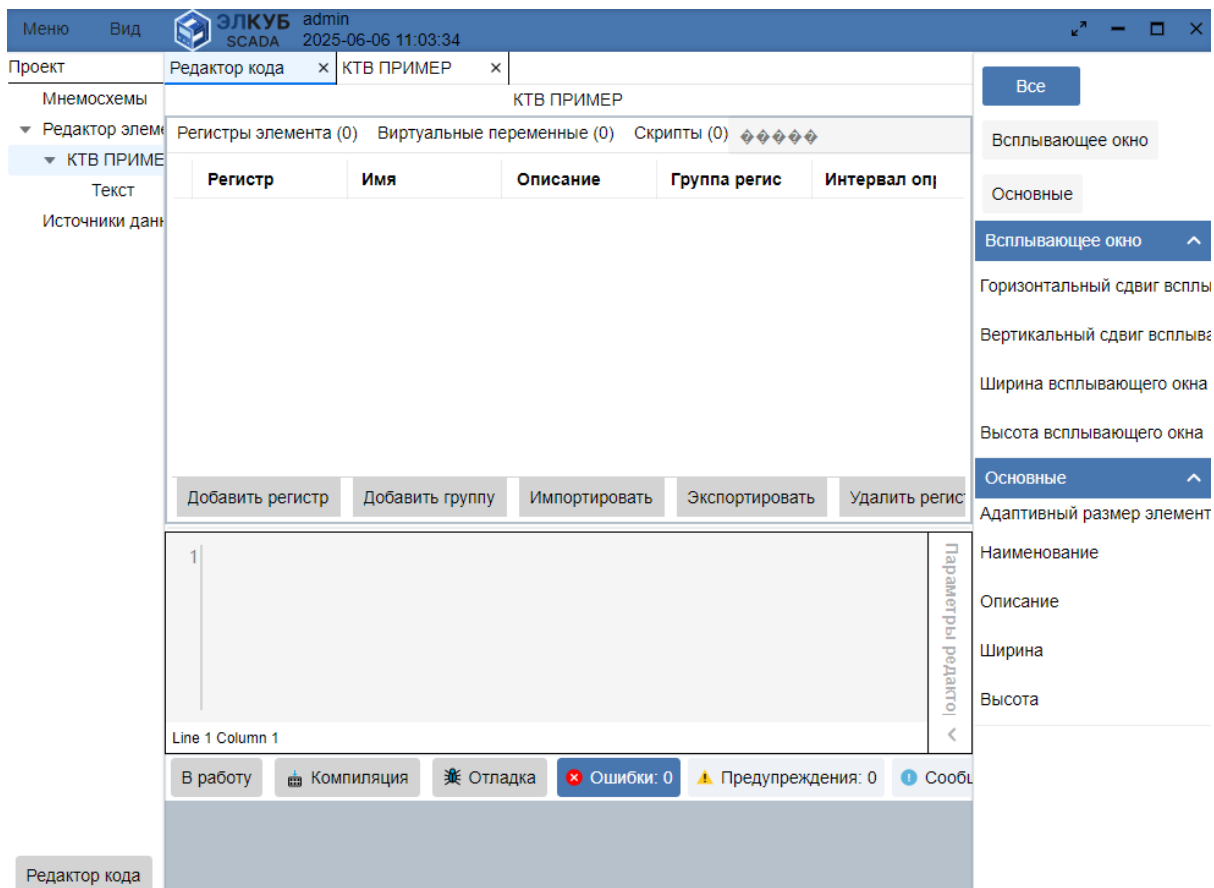
## Пример использования

Предположим, у нас есть пустой проект и с элементом “КТВ ПРИМЕР”, в котором содержится только примитив “Текст” с надписью “КТВ”.

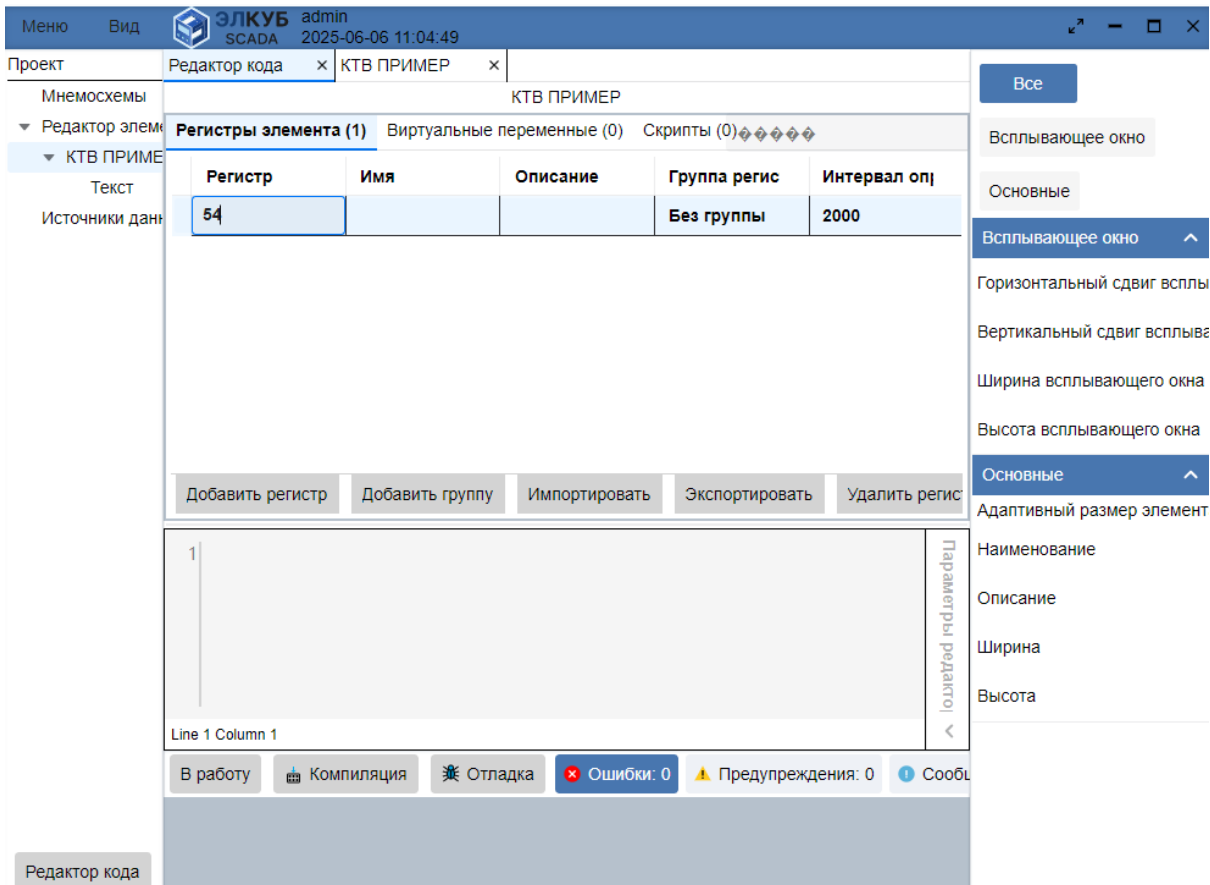


Нужно изменять цвет надписи “КТВ” в элементе при срабатывании одного из датчиков КТВ. Для этого нужно сделать следующее:

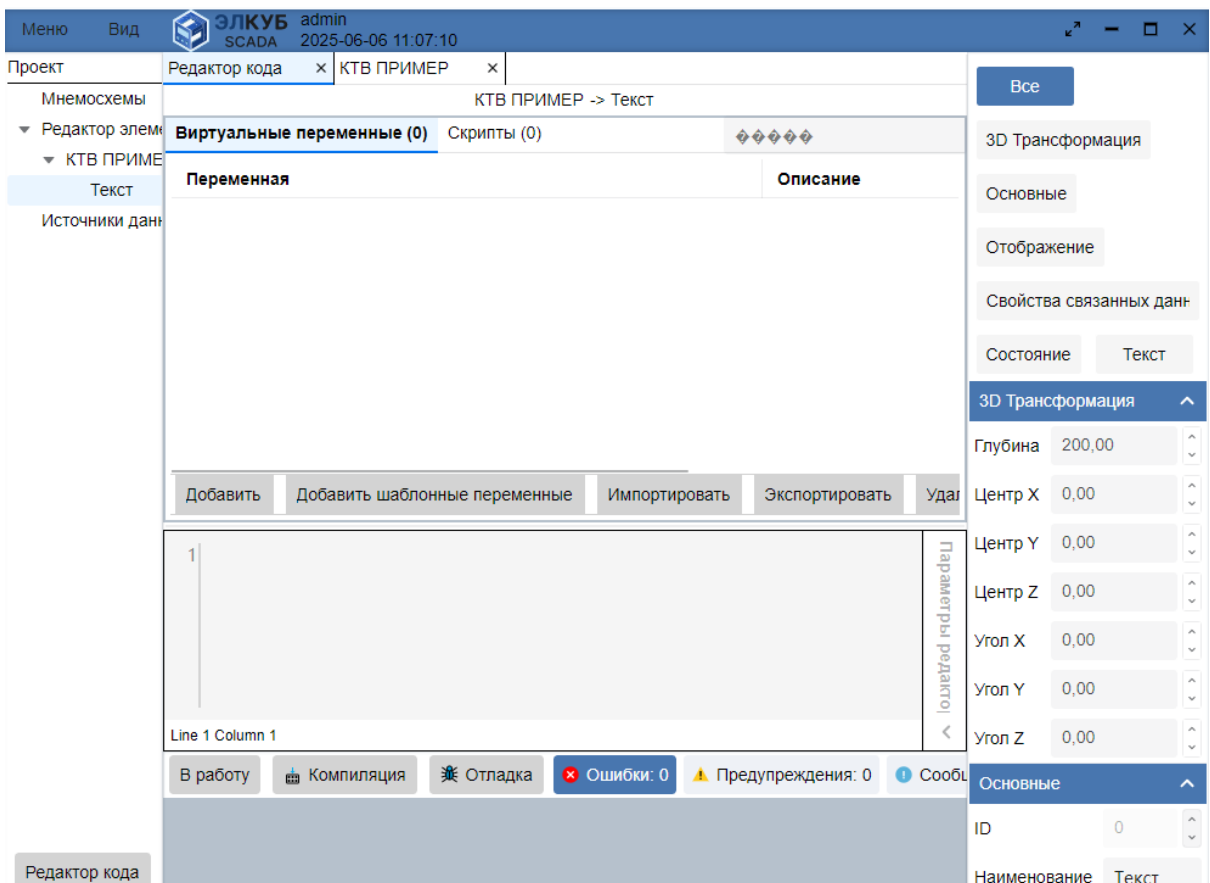
- 0) Открыть редактор кода и выбрать в дереве проекта нужный элемент.



- 1) Добавить опрос регистра, отвечающего за датчики КТВ. Определить, какой именно регистр нужно опрашивать, можно по таблице регистров у производителя опрашиваемого устройства. Предположим, что это регистр 54. В таком случае в мы переходим в подвкладку “Регистры элемента” и добавляем регистр с номером 54: нажать кнопку “Добавить регистр”, в появившемся регистре вписать в ячейку “Регистр” значение “54”.

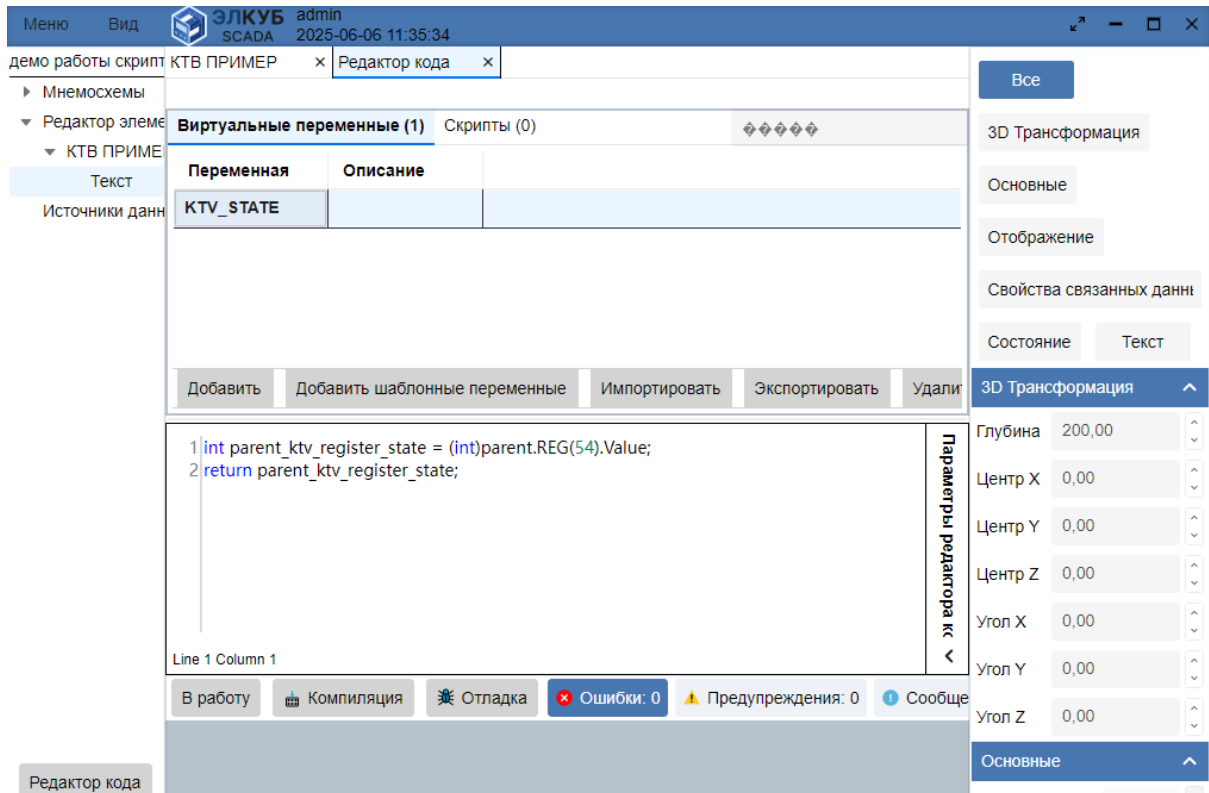


2) Перейти в редактирование кода примитива “Текст” внутри элемента.



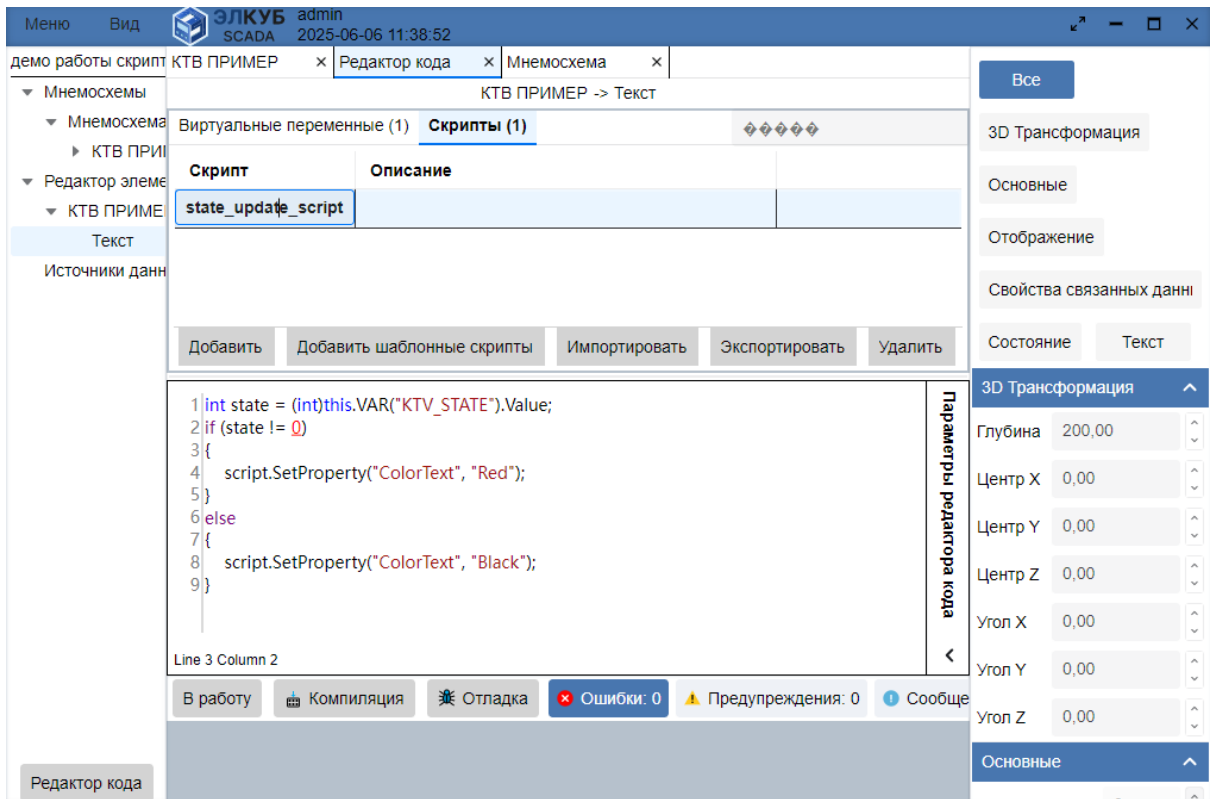
- 3) Добавить виртуальную переменную KTV\_STATE: добавить пустую переменную, нажав кнопку “Добавить”, и вписать в ячейку “Переменная” название “KTV\_STATE”. После этого добавить виртуальной переменной код, который будет устанавливать ей значение регистра датчиков КТВ (что будет говорить о том, что сработал один из датчиков КТВ):

```
int parent_ktv_register_state = (int)parent.REG(54).Value;
return parent_ktv_register_state;
```

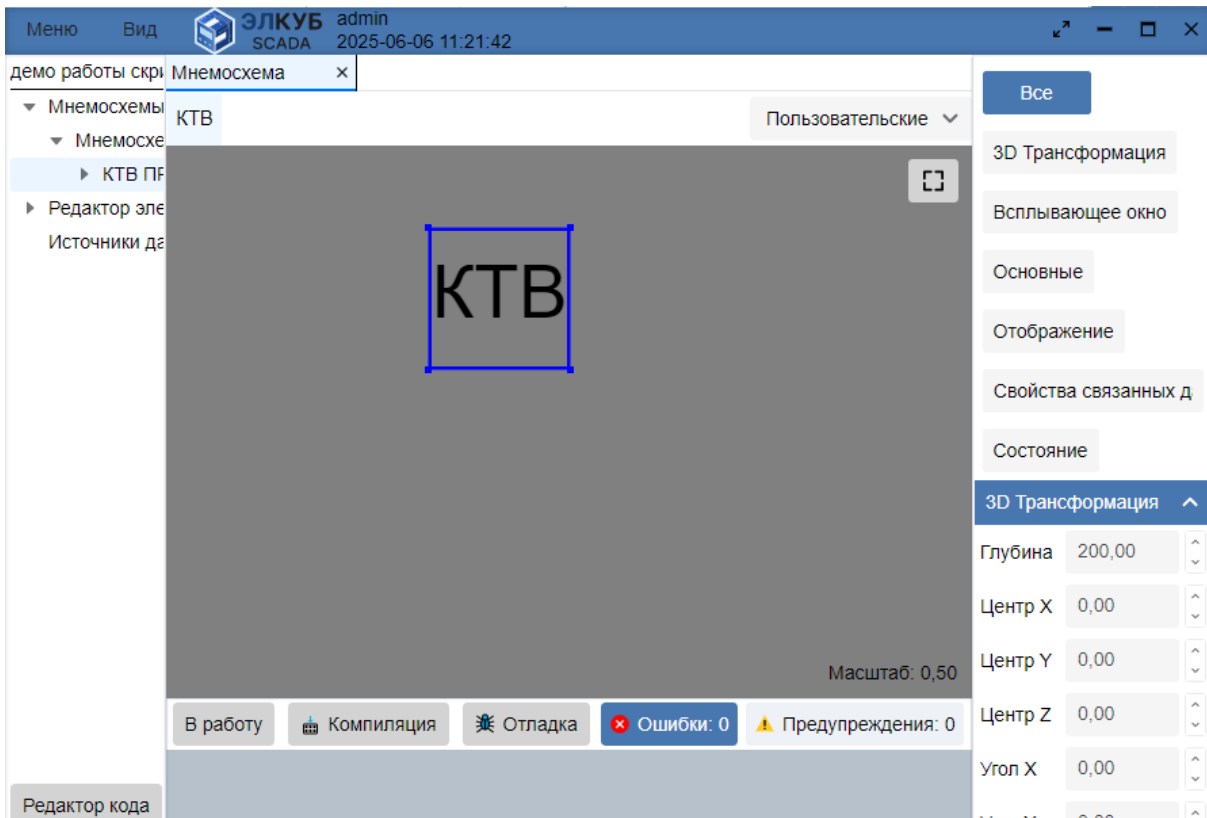


- 4) Далее добавляем скрипт state\_update\_script для этого примитива, в котором указываем, что текст нужно делать чёрным при отсутствии сигналов от датчиков КТВ (KTV\_STATE == 0) и красным при наличии:

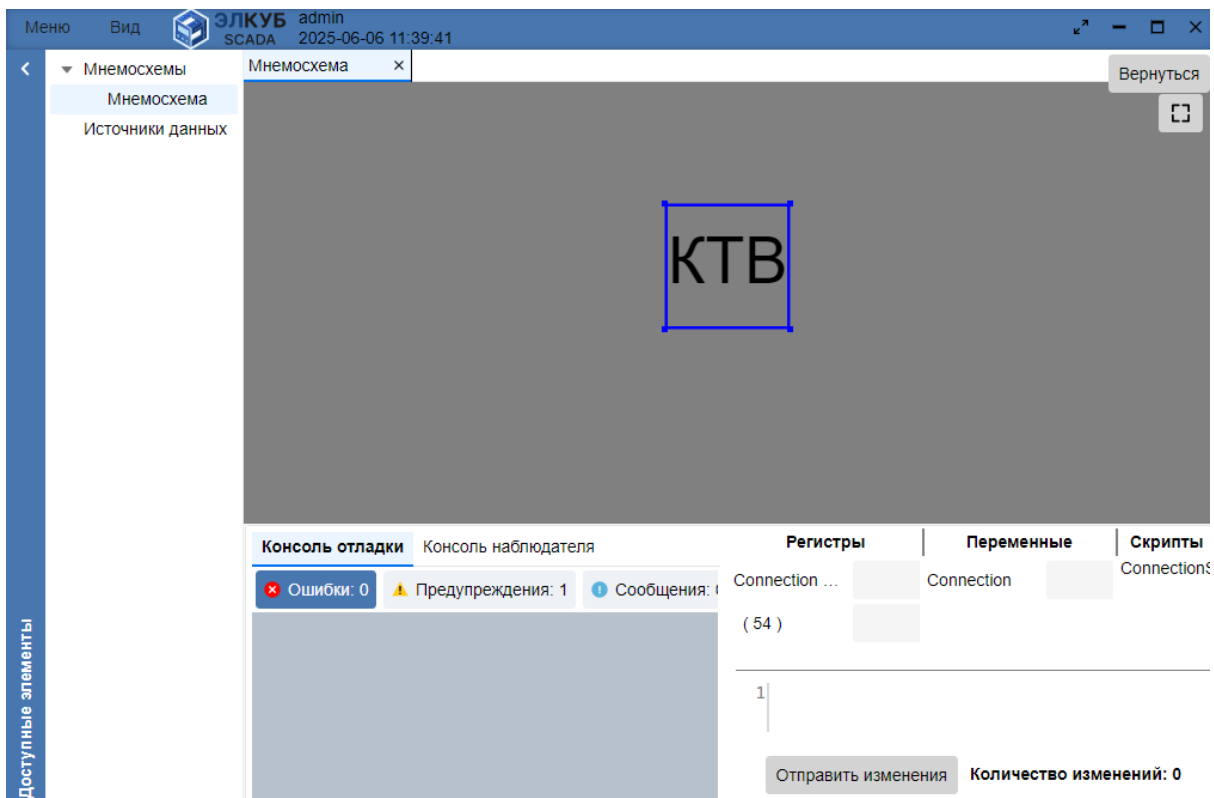
```
int state = (int)this.VAR("KTV_STATE").Value;
if (state != 0)
{
 script.SetProperty("ColorText", "Red");
}
else
{
 script.SetProperty("ColorText", "Black");
}
```



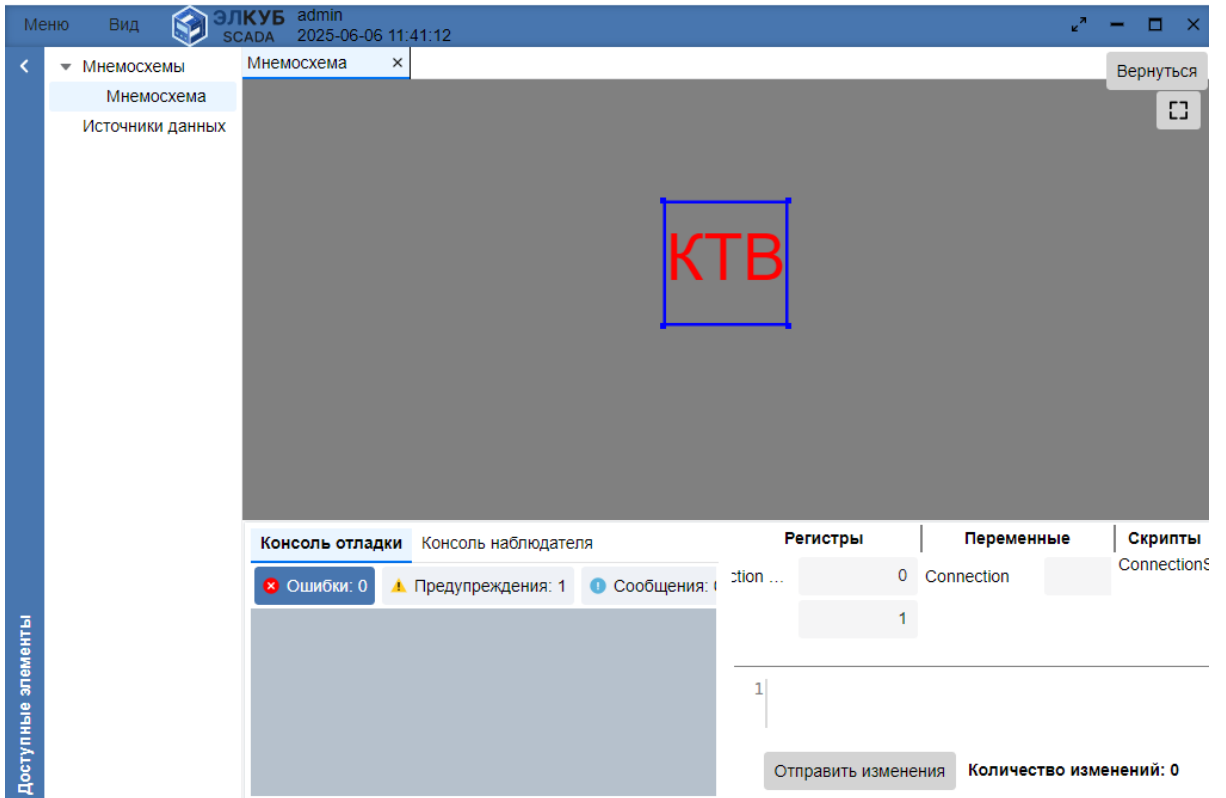
5) Создаём мнемосхему и размещаем на ней экземпляр созданного элемента.



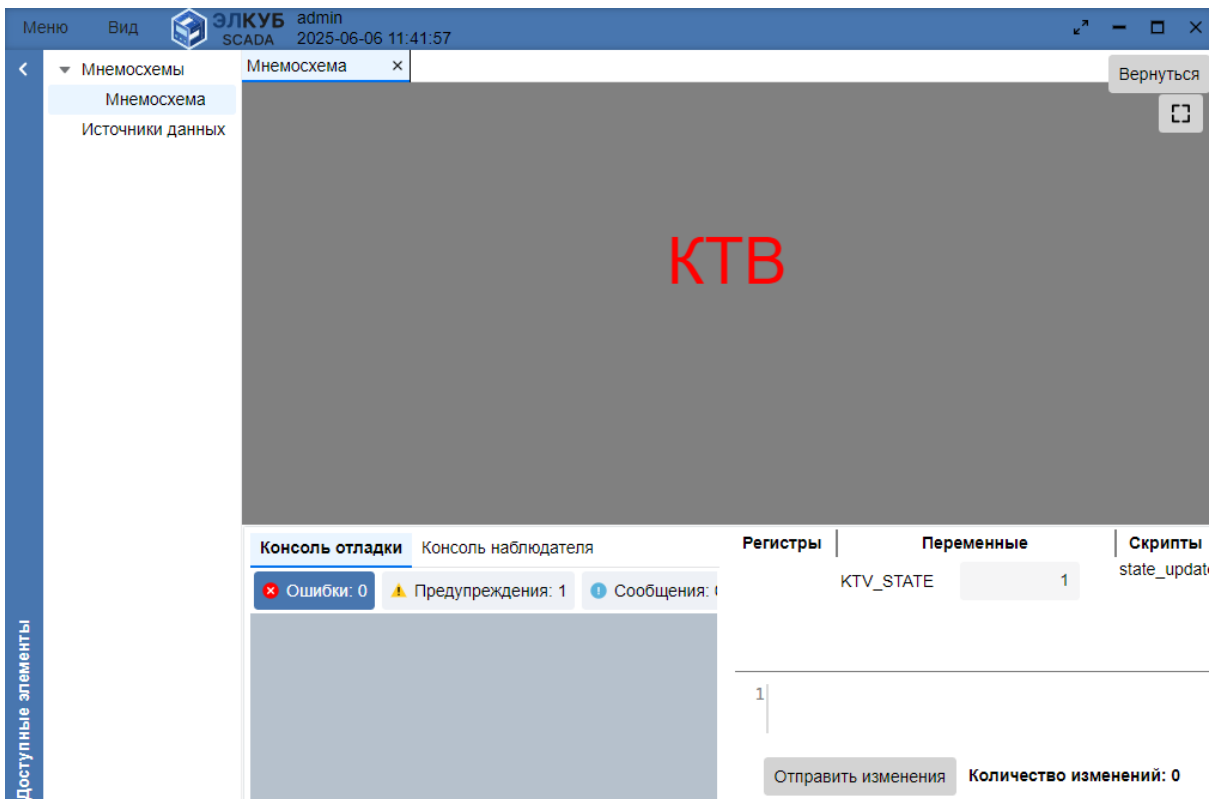
- 6) Переименовываем проект в “демо работы скриптов”, сохраняем в базу.
- 7) Переходим в режим отладки (нажать “Отладка”), открываем мнемосхему с элементом “КТВ ПРИМЕР”, нажимаем на элемент для отображения его регистров.



- 8) Чтобы протестировать корректность написанных переменных и скриптов, укажем в регистр 54 значение 1 и жмём “Отправить изменения”. Надпись КТВ меняет цвет на красный.



Нажав на надпись, можем посмотреть обновлённое значение виртуальной переменной, связанной с регистром 54.



# Работа в режиме наблюдения (на примере проекта автоматизации конвейерной линии)

## Запуск клиента

Нажмите правой кнопкой мыши дважды на иконку клиента ЭЛКУБ-SCADA на рабочем столе. Откроется окно авторизации.



**Авторизация**

Логин

Пароль


Подключено






Введите свои учетные данные и нажмите кнопку “Войти”.

Если вход был выполнен под учетной записью оператора или наблюдателя, то по умолчанию откроется последний активный проект в режиме наблюдения.

## Цветовая схема индикации состояний

В проекте для индикации состояния элементов контроля используется следующая цветовая схема индикации состояний.





| Цвет, анимация                                                                      | Описание                              |
|-------------------------------------------------------------------------------------|---------------------------------------|
|  | Выключено (0), не аварийное состояние |

|                                                                                             |                                                   |
|---------------------------------------------------------------------------------------------|---------------------------------------------------|
|            | Включено (1), не аварийное состояние              |
|            | Аварийное состояние                               |
|            | Состояние неизвестно, т.к. данные еще не получены |
|  (мигание) | Переходный процесс, включение                     |
|  (мигание) | Переходный процесс, выключение                    |

В случае, если отсутствует связь с контроллером, то все связанные с этим контроллером элементы затемняются и на них отображается иконка “нет связи”



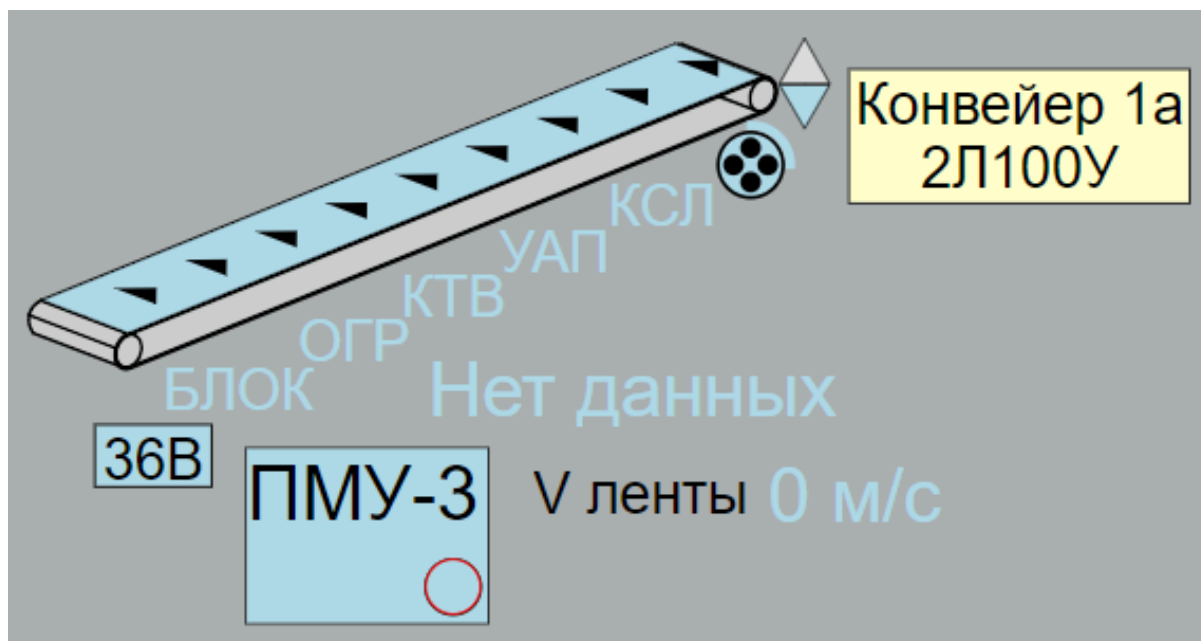
или надпись “НЕТ СВЯЗИ” (для крупных компонентов). При этом цвет элемента (и анимация) в затемненном состоянии обозначает последнее состояние перед потерей связи.

| Цвет, анимация                                                                                | Описание                                                        |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
|            | Нет связи, последнее состояние выключено (0), не аварийное.     |
|            | Нет связи, последнее состояние включено (1), не аварийное       |
|            | Нет связи, последнее состояние аварийное                        |
|            | Нет связи, последнее состояние неизвестно                       |
|  (мигание) | Нет связи, последнее состояние - переходный процесс, включение  |
|  (мигание) | Нет связи, последнее состояние - переходный процесс, выключение |

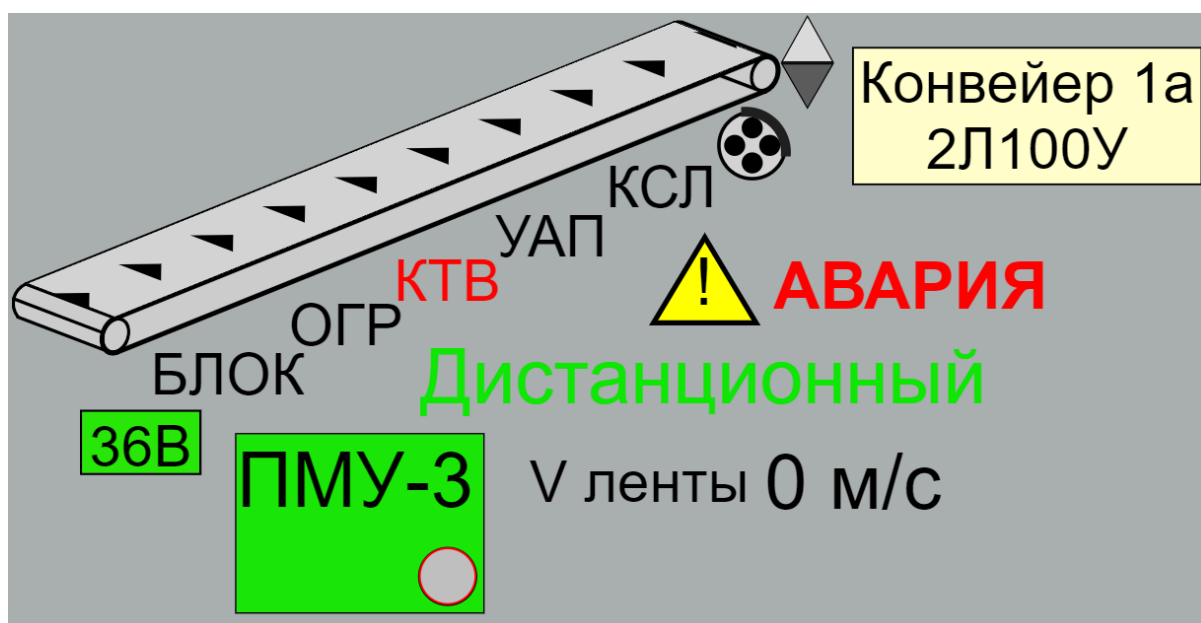
## Мониторинг состояния конвейера

При запуске проекта ЭЛКУБ-SCADA происходит опрос контроллеров и в клиенте ЭЛКУБ-SCADA будет отображаться состояние конвейера.

На рисунке ниже показан базовый блок конвейера в состоянии “отсутствуют данные”. Это означает, что еще не прошел цикл опроса контроллеров в сети и данные по конвейеру не получены.



На рисунке ниже показан базовый блок конвейера в состоянии “остановлен по срабатыванию КТВ”.

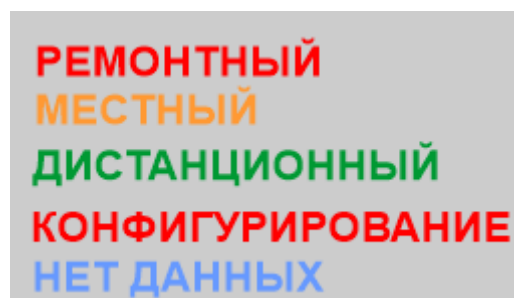


В случае отсутствия связи с контроллером конвейера блок выглядит следующим образом:

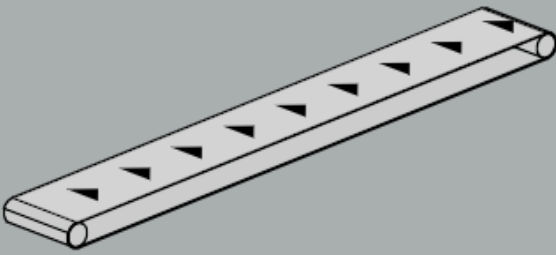


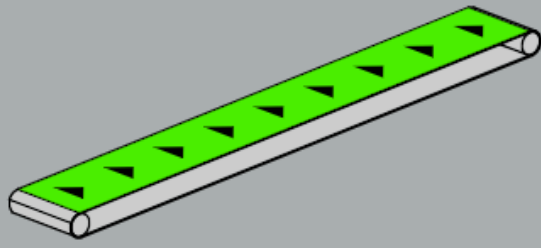
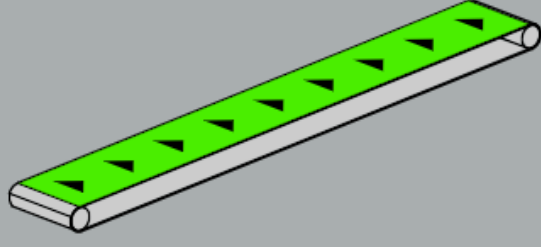
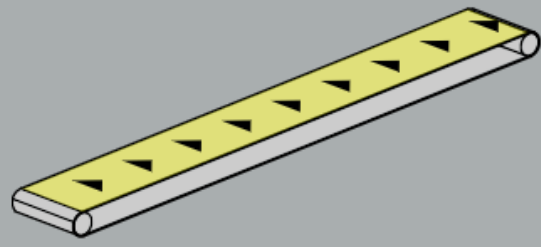
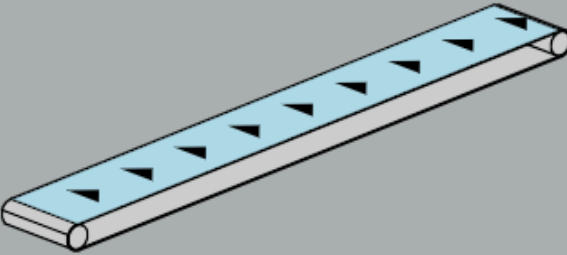
Затенение и значки отсутствия связи говорят о том, что нет связи с контроллером. Цвет показывает последнее известное состояние.

*Индикация режима работы - местный, ремонтный, дистанционный*





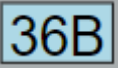
*Индикация состояния конвейера на мнемосхеме*

| Изображение, анимация                                                               | Описание                   |
|-------------------------------------------------------------------------------------|----------------------------|
|  | <p>Конвейер остановлен</p> |

|                                                                                                        |                                      |
|--------------------------------------------------------------------------------------------------------|--------------------------------------|
| <br>(мигание)         | Конвейер запускается                 |
| <br>( движение ленты) | Конвейер запущен                     |
| <br>(мигание)       | Конвейер останавливается             |
|                     | Нет информации о состоянии конвейера |

***Индикация наличия напряжения***

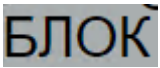
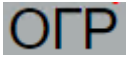



| Изображение                                                                         | Описание                                    |
|-------------------------------------------------------------------------------------|---------------------------------------------|
|  | Напряжение отсутствует, аварийное состояние |

|                                                                                   |                                                            |
|-----------------------------------------------------------------------------------|------------------------------------------------------------|
|  | Напряжение присутствует, нормальное состояние              |
|  | Наличие напряжения неизвестно, ожидается получение данных. |

### ***Индикация состояния блокировок по дискретным входам***

Для индикации состояния блокировок по дискретным входам используются три цвета: Черный - блокировка не сработала, нормальное состояние; Красный - блокировка сработала (аварийное состояние), синий - нет данных о состоянии блокировки.

Блокировка остается сработавшей до тех пор, пока не будет устранена причина блокировки и не выполнен сброс аварий оператором (во всплывающем окне контроллера) или механиком (с контроллера управления конвейером).

| <b>Изображение</b>                                                                  | <b>Описание</b>                                                                         |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|   | Блокировка запуска от следующего конвейера. (Нормальное состояние, не сработавшее.)     |
|  | Блокировка по датчикам ограждения. (Нормальное состояние, не сработавшее.)              |
|  | Блокировка по кабельно-тросовым выключателям. (Блокировка сработала)                    |
|  | Блокировка от системы пожаротушения. (Нормальное состояние, не сработавшее.)            |
|  | Блокировка от датчиков схода ленты. (Состояние неизвестно, ожидается получение данных.) |

### ***Индикация скорости ленты***

Черный - значение скорости находится в допустимых пределах, Красный - значение скорости выходит за допустимые пределы, синий - нет данных о состоянии.

Скорость ленты определяется контроллером и передается на верхний уровень.

| <b>Изображение</b> | <b>Описание</b> |
|--------------------|-----------------|
|--------------------|-----------------|

|                       |                                                           |
|-----------------------|-----------------------------------------------------------|
| V ленты <b>10 м/с</b> | Значение скорости выходит за допустимые пределы.          |
| V ленты <b>0 м/с</b>  | Значение скорости нормальное, конвейер выключен.          |
| V ленты <b>2 м/с</b>  | Значение скорости нормальное, конвейер работает.          |
| V ленты <b>0 м/с</b>  | Значение скорости неизвестно, ожидается получение данных. |

### **Аварийное состояние**


В случае возникновения аварийной ситуации (возникновение аварийной ситуации определяется контроллером конвейера (ПМУ, ПГУ или КМШ-ГС-КТ), в результате чего выставляется соответствующий бит в регистрах состояния) на экране возле соответствующего конвейера возникает **мигающее** сообщение:



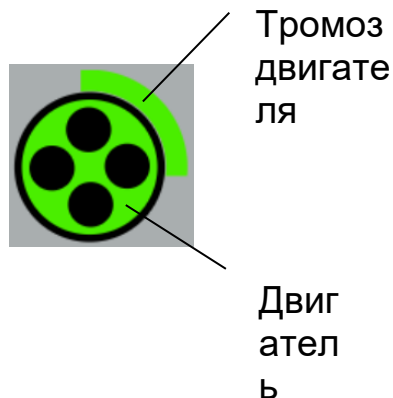
Список активных аварий можно посмотреть во всплывающем окне конвейера, дважды нажав на него мышью.

### **Индикация пересыпа**

| <b>Изображение</b> | <b>Описание</b>                                              |
|--------------------|--------------------------------------------------------------|
|                    | Датчик заштыбовки (пересыпа) сработал. Аварийное состояние   |
|                    | Датчик заштыбовки (пересыпа) неактивен. Нормальное состояние |

|                                                                                   |                                                                                         |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  | <p>Состояние датчика заштыбовки (пересыпа) неизвестно, т.к. данные еще не получены.</p> |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|

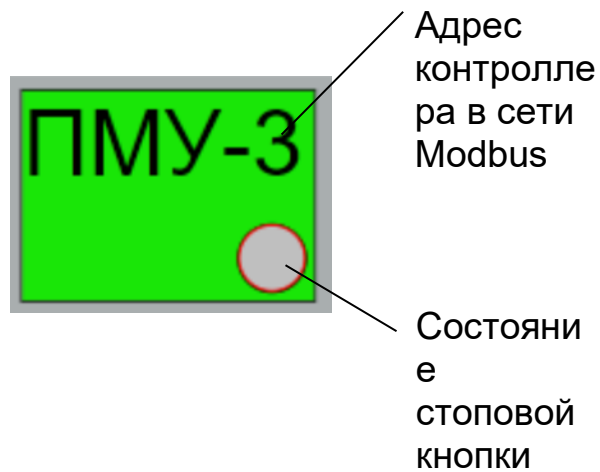
### ***Индикация состояния двигателей конвейера***



| Изображение                                                                                           | Описание                                    |
|-------------------------------------------------------------------------------------------------------|---------------------------------------------|
|  <p>(вращение)</p> | <p>Двигатель работает, тормоз снят</p>      |
|                    | <p>Двигатель остановлен, тормоз наложен</p> |
|                    | <p>Состояние неизвестно</p>                 |

### ***Индикация состояния контроллера конвейера***

Используется следующее обозначение контроллера



| Изображение                                                                         | Описание                                                            |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------|
|   | <p>Нормальное состояние контроллера, стоповая кнопка не нажата.</p> |
|  | <p>Нормальное состояние контроллера, нажата стоповая кнопка</p>     |
|  | <p>Ошибка контроллера</p>                                           |
|  | <p>Состояние контроллера неизвестно</p>                             |

## Информационная табличка

Информационная табличка не меняет своего состояния и служит для вывода вспомогательной статической информации.

|                                           |                                           |
|-------------------------------------------|-------------------------------------------|
| Конвейер 4<br>2П-120<br>КОНЦ. Штрек Г.+93 | Перегрузатель<br>Grot-1100<br>КШ 826      |
| Конвейер 1а<br>2Л100У                     | Забойный конвейер<br>Rybnik-1100 Лава 826 |

## Окно вывода состояния конвейера

Наименование контроллера

Кнопка закрытия всплывающего окна

Кнопка запуска конвейера (активна только в дистанционном режиме)

Кнопка останова конвейера (активна только в дистанционном режиме)

Кнопка сброса аварий (активна только в дистанционном режиме)

Кнопка переключения режима местный/дистанционный

Кнопка переключения в ремонтный режим

Кнопка перезагрузки контроллера

Окно вывода списка аварий контроллера