

АО «Лаборатория технологий автоматизации»

ЛАЦЕРТА



Платформа «Лацерта»

Автоматизированная система управления
технологическими процессами

Описание платформы

(Описание программного обеспечения)

Версия 1.0

Москва, 2026

Настоящий документ является публичным и доступен для распространения без каких либо изменений. Запрещается использование материалов, содержащихся в нём в любой форме без согласования с правообладателем. По любым вопросам вы можете связаться с представителями правообладателя по email: info@apcslab.ru.

Все права защищены ©

АО «Лаборатория Технологий Автоматизации», ОГРН 1257700513113

Свидетельство о регистрации программы для ЭВМ №[2022660501](#)

Настоящее программное обеспечение охраняется патентом на изобретение №[2797756](#) от 08.06.2023, выданным Роспатентом

Настоящее программное обеспечение охраняется патентом на изобретение №[045687](#) от 15.12.2023, выданным Евразийской патентной организацией

Запись в реестре российского программного обеспечения Минцифры РФ №[17502](#)

Оглавление

1. Введение.....	6
1.1. Задачи решаемые платформой.....	7
1.2. Область применения.....	8
2. Архитектура платформы.....	9
2.1. Особенности архитектуры.....	9
2.2. Состав платформы.....	10
2.3. Ядро платформы.....	11
2.3.1. Структура данных ядра.....	12
2.3.2. Оперативные данные.....	12
2.3.3. Метаданные.....	13
2.3.4. Архив.....	13
2.3.5. Типы данных.....	14
2.3.6. Модели.....	14
2.3.7. Объекты.....	15
2.4. Событийная модель.....	16
2.4.1. Тревоги.....	17
2.5. Сервисная модель.....	18
2.5.1. Сервисы.....	18
2.5.2. Драйверы.....	19
2.5.3. Клиентские интерфейсы.....	19
2.6. Вычислительная модель.....	20
2.6.1. Калькулятор.....	20
2.6.2. Имитатор.....	21
2.6.3. Проектно зависимые вычисления.....	21
2.7. Кластерная модель.....	21
2.7.1. Кластерный узел.....	23
2.7.2. Интерконнект.....	23
2.7.3. Клиент кластера.....	23
2.8. Разграничение доступа.....	24
2.9. Диагностика.....	25
2.10. Визуализация данных.....	26
2.10.1. Типовые элементы.....	27
2.10.2. Виджеты.....	27
2.10.3. Мнемосхемы.....	28
2.10.4. Привязка переменных.....	28
2.10.5. Программирование.....	29

2.10.6. Табличное представление данных.....	31
2.10.7. Визуализация временных рядов.....	31
2.10.8. Внешние приложения и виджеты.....	31
2.11. Импорт и экспорт.....	32
3. Интегрированная среда разработки.....	33
3.1. Структура данных проекта.....	34
3.2. Редактор моделей.....	34
3.2.1. Модель Boolean.....	40
3.2.2. Модель Directory.....	41
3.2.3. Модель Float.....	42
3.2.4. Модель Integer.....	42
3.2.5. Модель String.....	43
3.2.6. Модель Link.....	44
3.2.6. Составные модели.....	45
3.2.7. Создание новой модели.....	47
3.2.8. Удаление модели.....	50
3.3. Редактор объектов.....	52
3.3.1. Параметры объектов.....	53
3.3.2. Привязка драйвера к переменным объекта.....	54
3.3.3. Настройка имитации переменных объектов.....	58
3.3.4. Настройка вычисления расчётных переменных.....	61
3.3.5. Параметры аналоговых объектов.....	64
3.3.6. Параметры дискретных объектов.....	65
3.3.7. Графические признаки объектов.....	70
3.3.8. Создание объекта.....	71
3.3.9. Удаление объекта.....	74
3.4. Компоненты.....	76
3.5. Состав мнемосхемы.....	77
3.5.1. Создание компонента.....	77
3.5.2. Создание мнемосхемы.....	81
3.5.3. Импорт.....	84
3.5.4. Синхронизация.....	85
3.5.5. Редактор привязок мнемосхем к данным.....	88
3.5.6. Редактор сценариев поведения компонентов.....	103
3.5.7. Редактор сценариев мнемосхем.....	105
3.5.8. Функции скрипта мнемосхемы/компонитной мнемосхемы.....	107
3.5.9. Функции скрипта компонентов и элементов мнемосхемы.....	114
3.5.10. Специальные возможности.....	118
3.5.11. Фрагменты.....	119
3.6. Драйверы протоколов АСУ ТП.....	120

3.6.1. Список драйверов Платформы.....	121
3.6.2. Конфигуратор драйверов.....	124
3.6.3. Драйвер ICMP.....	135
3.6.4. Драйвер SNMP.....	137
3.6.5. Драйвер OPC UA.....	140
3.6.6. Драйвер Modbus TCP.....	142
3.6.7. Драйвер Profinet (S7Comm).....	145
3.6.8. Драйвер MQTT.....	148
3.6.9. Драйвер BACnet.....	151
3.6.10. Драйвер IEC 61850.....	154
3.6.11. Драйвер IEC 60870-5-104.....	157
3.7. Драйверы сопряжения с внешними провайдерами.....	160
3.7.1. Драйвер Trassir.....	160
3.7.2. Драйвер Perco.....	164
3.7.3. Драйвер FindFace.....	168
3.7.4. Драйвер Zabbix.....	172
3.7.5. Драйвер OrionPro.....	176
3.8. Сервисы.....	179
3.8.1. Список сервисов Платформы.....	179
3.8.2. Сервис расчетных сигналов.....	183
3.8.3. Сервис обработки событий.....	183
3.8.4. Сервис управления архивом.....	183
3.8.5. Сервис имитации данных.....	183
3.8.6. Сервис обновления настроек интеграции.....	184
3.9. Кластер.....	184
3.10. События.....	184
3.10.1. Управление классами событий.....	185
3.10.2. Управление наборами событий.....	192
3.11. Пользователи и роли.....	194
3.11.1. Настройка ролей.....	195
3.11.2. Настройка пользователей.....	198
3.12. Личный кабинет.....	207
3.13. Статистика.....	208
4. Среда исполнения.....	210
4.1. Авторизация.....	212
4.2. Стартовый экран и навигация.....	215
4.3. Мнемосхемы и виджеты.....	218
4.3.1. Композитная мнемосхема.....	220
4.3.2. Состояния компонентов.....	228
4.3.3. Состояние тревоги.....	230

4.4. Таблицы.....	230
4.5. Тренды.....	243
4.6. События.....	257
4.7. Тревоги.....	262
4.7.1. Таблица.....	263
4.7.2. Квитирование.....	264
Приложение 1. Термины и определения.....	264
Приложение 2. Принятые сокращения.....	266

1. Введение

Платформа «Лацерта», далее «Платформа» - является инструментом для построения, разработки и эксплуатации автоматизированных систем управления (АСУ) объектами и комплексами информационно-технологической инфраструктуры на всём протяжении их жизненного цикла. Платформа представляет собой совокупность инструментов (сервисов), необходимых для построения АСУ как в режиме проектирования и разработки, так и в режиме эксплуатации (управления объектами информационно-технологической инфраструктуры).

1.1. Задачи решаемые платформой

Платформа предназначена для обслуживания объектов промышленной автоматизации на всех этапах их жизненного цикла.

Основные функциональные возможности Платформы:

- сбор, предварительная обработка данных, поступающих от различных распределенных источников, измерительных приборов, устройств промышленной автоматизации, индустриального интернета вещей и т.д.;
- приведение поступающих данных к единому, принятому в Платформе, формату с целью их дальнейшей обработки и хранения;
- двусторонний обмен данными с внешними информационными системами;
- обработка поступающих данных по заранее заданным алгоритмам;
- формирование управляющих воздействий (команд), передаваемых на управляющие устройства, с учётом логики технологических процессов объекта автоматизации;
- визуализация данных в реальном времени с помощью динамических мнемосхем, виджетов и прочих графических элементов.

Платформа предоставляет пользователю встроенные инструменты:

- прототипирования, проектирования как централизованных, так и распределённых систем мониторинга, управления технологическими и производственными процессами, а также прочих автоматизированных систем;
- сопряжения с внешними автоматизированными, автоматическими и информационно-технологическими системами, с возможностью их интеграции в единый интерфейс управления;
- обеспечения информационной безопасности с учётом требований конкретной целевой системы;
- создания удобных, современных, эргономических пользовательских интерфейсов.

Вышеперечисленные инструменты Платформы доступны пользователю на протяжении всего жизненного цикла, что позволяет адаптировать Платформу при модернизации или изменении функциональностей управляемой информационно-технологической системы.

Достоинством Платформы также является низкий «порог входа» пользователей и разработчиков, что предполагает высокий уровень квалификации в предметной области, но не требует навыков в области программирования и/или графического дизайна.

1.2. Область применения

Платформа предназначена для создания и эксплуатации автоматизированных систем управления широкого спектра применения, таких как:

- централизованные, распределенные, гибридные системы управления;
- автономные системы управления;
- системы диспетчерского управления;
- автоматизированные системы управления технологическими процессами (АСУ ТП);
- автоматизированные системы управления производственными процессами (АСУ ПП);
- системы диагностики и мониторинга промышленного оборудования;
- системы диагностики и мониторинга телекоммуникационного и серверного оборудования;
- системы промышленного интернета вещей (IIoT);
- системы предиктивной аналитики;
- системы поддержки принятия решения.

Платформа позволяет создавать диспетчерские центры управления предприятиями любого масштаба - от локальных до распределенных, в том числе федерального и глобального уровня.

Платформа позволяет обслуживать технологические системы предприятий самых различных отраслей, таких как:

- генерация электроэнергии (тепловая, гидро, атомная, возобновляемая);
- преобразование и транспорт электроэнергии, электрические сети, подстанции;
- добыча, переработка, транспорт сырья;
- металлургия, химическое производство, машиностроение;
- транспортная инфраструктура;
- инфраструктура умного города и ЖКХ;
- ЦОД и ИТ инфраструктура и др.

2. Архитектура платформы

2.1. Особенности архитектуры

- Возможность изменения конфигурации системы и данных «на лету», что не требует перерывов/остановок рабочих процессов или сводит их к минимуму;
- Кластеризация - автоматическое резервирование серверов (2 и более) и синхронизация данных в кластере;
- Возможность работы со значительными объемами данных в т.ч. в условиях высоких нагрузок;
- деградация - перераспределение ресурсов кластера в зависимости от нагрузки;
- линейное масштабирование - возможность добавлять новые узлы для увеличения общей производительности;
- мультитенантность - возможность создать для каждого пользователя или группы пользователей свой набор мнемосхем и отображаемых данных.

2.2. Состав платформы

В состав Платформы входят специализированные инструменты для разработки и эксплуатации автоматизированных систем управления, логически объединенных в интегрированную среду разработки (ИСР¹) и среду выполнения (СВ/runtime)².

Ключевой частью платформы является ее Ядро, которое содержит ключевые, интегрированные друг с другом компоненты:

- система управления данными - оперативными, неоперативными и архивными;
- система управления кластером;
- менеджеры событий, сервисов, драйверов и клиентов.

Интегрированная среда разработки (ИСР) и среда выполнения (СВ) Платформы включает в себя следующие базовые инструменты:

- сервисы (разд. 2.5.1);
 - анализатор изменений;
 - генератор событий;
 - калькулятор (разд. 2.6.1);
 - имитатор значений (разд. 2.6.2).
- драйверы протоколов (разд. 2.5.2);
- клиенты системы, например, среда визуализации.

Для решения специфических задач, свойственных конкретным отраслям, могут потребоваться дополнительные инструменты. Для этого в Платформе предусмотрен инструментальный прикладной программный интерфейс (ППИ), с помощью которого могут разрабатываться целевые сервисы, драйверы и клиенты.

¹ integrated development environment (IDE, designtime)

² runtime

2.3. Ядро платформы

Ядро обеспечивает ключевую и критическую функциональность Платформы с помощью следующих компонентов (рис. 2.1):

- подсистема управления оперативными данными (разд. 2.3.2);
- подсистема управления метаданными³ (разд. 2.3.3);
- подсистема управления архивными данными (разд. 2.3.4);
- подсистема управления кластером (разд. 2.7);
- менеджер событийной модели (разд. 2.4);
- менеджер сервисов (разд. 2.5.1);
- менеджер драйверов (разд. 2.5.2);
- менеджер клиентов (разд. 2.5.3).

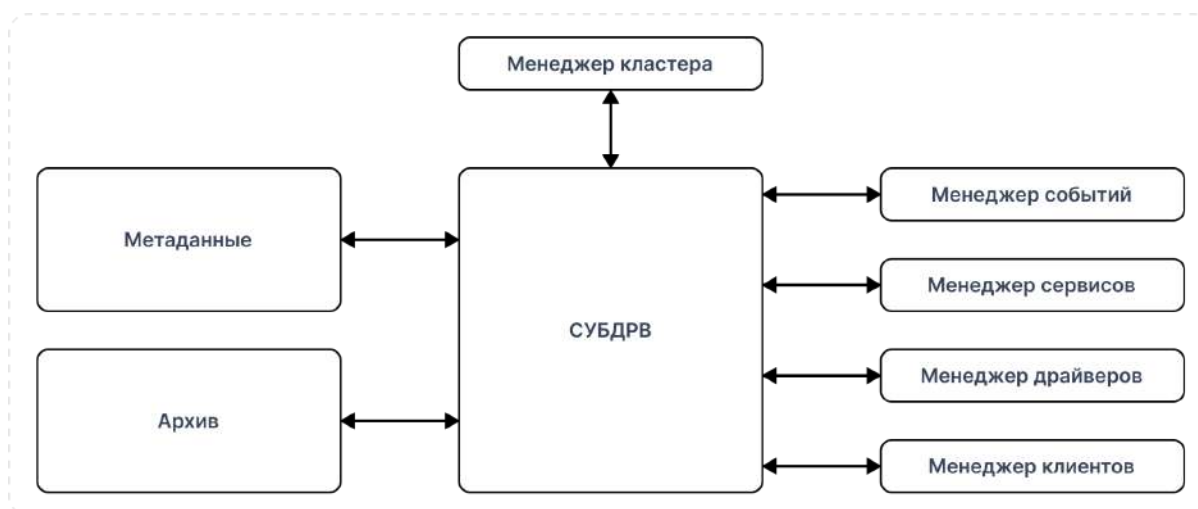


Рис. 2.1. Состав компонентов ядра

³ данные о конфигурации конечных устройств, иерархических связей, сообщений и др.

2.3.1. Структура данных ядра

Ядро Платформы использует внутреннюю специализированную структуру данных. В рамках данной конструкции производится двусторонний обмен данными между компонентами и управляющими элементами ядра. Такая конструкция позволяет решать несколько критически важных задач:

- оперативные данные (ОДА) отделены от метаданных (МДА), следовательно, система не перегружается лишними данными;
- обмен данными в целевой системе может быть сконфигурирован таким образом, чтобы ОДА и МДА передавались по разным физическим каналам, что очень важно с точки зрения обеспечения промышленной кибербезопасности.

Например, данные о конфигурации конечного оборудования (метаданные) могут быть востребованы только в режиме разработки, тогда как в режиме выполнения обращение к метаданным может не осуществляться вообще либо оперативные и метаданные потребуются только на уровне клиента — не полностью, а в части, касающейся конкретного оборудования.

2.3.2. Оперативные данные

Оперативными данными (ОДА) являются наиболее динамично меняющиеся переменные, критичные по времени. Такие переменные размещаются в оперативной памяти под управлением подсистемы реального времени (РВ), встроенной в ядро Платформы. В зависимости от задач целевой системы РВ может обслуживать все или часть таких переменных.

Основная задача подсистемы РВ заключается в обработке изменений переменных ОДА. Изменение переменной может проводиться на основе данных, поступающих из внешних сопрягаемых систем или устройств по заданному протоколу, обслуживаемому драйвером (разд. 2.5.2). Изменение переменных ОДА производится с помощью интегрированных сервисов (разд. 2.5.1), например калькулятора, имитатора и др.

Каждое изменение может сопровождаться совокупностью привязанных к переменной временных меток⁴. Расчетные интервалы между различными временными метками могут быть использованы для вычисления диагностических параметров переменных ОДА, таких как задержка⁵, джиттер⁶, а также оценки предельного времени выполнения операции (дедлайн⁷).

⁴ timestamp

⁵ latency, время отклика (задержки) операции

⁶ jitter, разброс значений времени отклика

⁷ deadline, предельный срок операции

ОДА могут быть использованы в различных оперативных алгоритмических расчетах, в т.ч., необходимых для управления.

ОДА содержат текущие значения переменных, а также прошлые значения за заданный период времени, оперативный (краткосрочный) архив. Оперативный архив хранится в формате временных рядов⁸. Ядро Платформы использует оперативный архив как кэш для дальнейшего вытеснения из него данных в долгосрочный архив.

Возможность обслуживания заданного объема ОДА с оперативным архивом определяется доступной оперативной памятью узла. Возможность обслуживания заданного объема изменений ОДА в единицу времени определяется вычислительной мощностью процессора узла.

При необходимости резервирования ОДА или распределения мощности обработки ОДА по нескольким узлам предусмотрена кластерная обработка ОДА (разд. 2.7).

2.3.3. Метаданные

Метаданные (МДА) предназначены для хранения не критичных относительно времени данных, таких как конфигурации конечных устройств, иерархических связей, сообщений и др.

Платформа предусматривает возможность хранения метаданных в реляционных СУБД.

2.3.4. Архив

Архив (долгосрочный) предназначен для хранения устаревших оперативных данных.

В зависимости от требований целевой системы архивные данные могут быть сжаты, разрежены или модифицированы целевым способом. Например, могут быть сохранены только значимые изменения и информация о местах возникновения тревожных событий. Архив может быть представлен в виде циклических записей, где каждый последующий период имеет больший интервал, чем предыдущий. Такой функционал определяется алгоритмами целевого проектного сервиса архивирования.

Архивные данные (АДА) могут быть использованы в различных неоперативных алгоритмических расчетах.

⁸ TSDB, time series database

АДА хранятся в формате временных рядов. Могут быть использованы СУБД хранения временных рядов (TSDB) и колоночные СУБД. Доступ к АДА существенно медленнее, чем к оперативным данным и определяется эффективностью используемых систем хранения данных (СХД) в целевой системе.

2.3.5. Типы данных

В платформе поддерживаются переменные основанные на простых типах данных. Поддерживаются следующие простые типы данных:

- дискретные, boolean — логическое значение 0/1 - да/нет;
- аналоговые, float — числовое значение с плавающей точкой, в зависимости от конфигурации ОС, может быть с двойной и более точностью;
- строковые, string — строковые значения, состоящие из букв, цифр и служебных символов;
- целочисленные, integer — целочисленные значения 1/2/3.

2.3.6. Модели

Платформа использует объектно-ориентированный подход к организации сложных структурированных данных, который позволяет определять сложно структурированные данные из переменных простых типов. Ядро определяет переменные как простейшие модели данных соответствующего типа.

Переменные могут быть объединены в модель (модель объекта, или объектную модель). Несколько подобных моделей, в свою очередь, могут быть сгруппированы в более сложную модель. Допускается группировка моделей с переменными, то есть с простейшими моделями. Таким образом, может быть создана сколь угодно сложная иерархически вложенная структура данных (рис. 2.2).

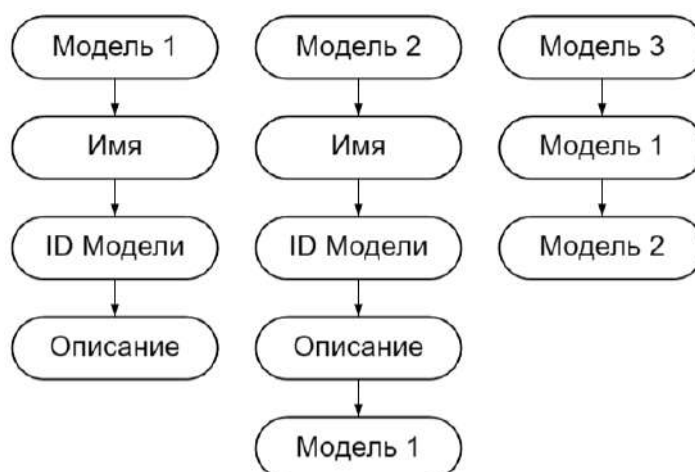


Рис. 2. 2. Структура моделей

Модель может содержать переменные **МДА** и **ОДА**. Важно отметить, что переменная МДА с соответствующим признаком будет воспринята ядром как необходимость создать переменную⁹ в ОДА. Значение такой переменной будет размещаться исключительно в ОДА и впоследствии вытеснена в АДА.

С точки зрения объектно-ориентированного подхода модели являются абстрактными сущностями. На основе моделей могут быть созданы множество объектов (разд. 2.3.6). Такие объекты будут полностью повторять структуру модели. Изменение модели повлечет каскадное изменение структуры всех объектов, созданных на ее основе.

Если в алгоритмической логике целевой системы используются переменные, удаляемые из структуры модели, а следовательно происходит удаление производных (дочерних) объектов, то такие действия могут привести к непредсказуемым последствиям, нарушающим всю логику работы системы. Поэтому такие изменения должны быть детально и тщательно спланированы. Рекомендуется предварительно сохранить структуру объектов путем резервного копирования либо экспорта текущей конфигурации.

2.3.7. Объекты

Объект представляет собой универсальную, удобную для представления и обработки структуру данных. Объект создается на основе модели (разд. 2.3.6). Объект может быть создан исключительно при наличии требуемой модели. Изменение структуры объекта без изменения структуры абстрактной модели невозможно.

Объекты могут образовывать иерархические структуры с любым уровнем вложенности.

Источниками данных для объекта могут являться:

- драйвер устройства (фактические данные, разд. 2.5.2);
- имитатор (искусственно сгенерированные данные, разд. 2.6.2);
- калькулятор (расчётные данные, в т.ч. с использованием данных от разных устройств, разд. 2.6.1)

⁹ в SCADA системах, такую переменную принято обозначать как тег (tag)

2.4. Событийная модель

Событийная модель Платформы основана на концепции событийно ориентированной архитектуры¹⁰.

События являются динамическими объектами платформы и создаются как реакция на изменения значения переменных.

Событие наступает при изменении состояния дискретного объекта, которое, в свою очередь, зависит от значения переменной, числовой или дискретной. Например, дискретный объект отслеживает значение температуры (числовая переменная) и при превышении заданного значения изменяет своё состояние. При этом наступает событие заранее заданного типа и с заданными атрибутами.

Когда в связи с внешним изменением или некоторой внутренней логикой происходит изменение значения переменной отслеживаемого объекта, подписанный на эти изменения специальный объект фиксирует факт изменений. Если выполняются условия реакции, специальный объект создает заранее predetermined событие с заданным текстом сообщения.

Событие имеет следующую совокупность атрибутов:

- объект, в котором произошло событие;
- класс события;
- время создания (фактическое время наступления события);
- текст сообщения (заранее определенный для конкретного события информационный текст, например «Тревога»);
- приоритет (заранее заданное числовое значение от 1 до 10);
- флаг активности (выполнены условия наступления события);
- флаг квитирования (пользователем подтверждено получение информации о событии);
- флаг принадлежности (кем квитировано).

Для каждого класса событий может быть задан приоритет (важность/критичность события), который будет влиять на ранжирование событий.

События могут находиться в активном или неактивном состоянии. Активным событие является до тех пор, пока выполнены условия его создания. Когда условия перестают выполняться, событие переводится в неактивное состояние.

¹⁰ event driven architecture (EDA)

В процессе настройки правил создания события можно задать необходимость его квитирования. Квитированием события пользователь подтверждает получение информации о событии и причине его возникновения.

Платформа фиксирует пользователя, осуществившего квитирование события. При возникновении аналогичного, но более нового события пользователь, квитировавший событие, не отобразится.

2.4.1. Тревоги

В большинстве АСУ принято использовать термин «тревога¹¹». В платформе отдельно не выделяются события типа «тревоги», но можно определить соответствующий класс события, задав релевантное наименование. Таким же образом можно определить и несколько событий класса «тревога» с разным приоритетом. Так как в АСУ тревоги принято считать наиболее критическими событиями, то для них рекомендуется выделять наиболее значимые значения приоритизации.

События типа «тревоги» могут вызывать назначенные пользователем аудиовизуальные эффекты для привлечения внимания операторов с помощью сопряжения с целевой системой специальных технических средств.

¹¹ alarm

2.5. Сервисная модель

Сервисная модель Платформы соответствует концепции сервис ориентированной архитектуры¹².

Ядро платформы обслуживает подключаемые сервисные приложения, которые разделяются на три класса:

- сервисы - сервисные приложения, осуществляют обработку данных близко к ядру системы (разд. 2.5.1);
- драйверы - сервисные приложения, осуществляют сбор и предварительную обработку данных с устройств и внешних сопрягаемых систем, а также передачу данных в обратную сторону (разд. 2.5.2);
- клиенты - сервисные приложения, обслуживают клиентов целевых систем в т.ч. ИСР, СРИ (разд. 2.5.3).

Использование сервисной модели позволяет легко адаптировать Платформу для создания целевых систем самого разного назначения, поскольку набор и функционал сервисных приложений могут быть изменены разработчиками целевой системы в соответствии с ее спецификой.

2.5.1. Сервисы

Сервисы осуществляют управление логическими сервисными приложениями, оперируя данными ОДА, МДА, АДА, непосредственно в оперативной памяти ядра.

Платформа содержит следующие базовые сервисы:

- сервис событий - осуществляет обработку логики событийной модели ядра (разд. 2.4);
- сервис расчетных переменных - осуществляет обработку логики вычислительной модели платформы (разд. 2.7);
- сервис имитации значений - осуществляет имитацию значений переменных по предопределенным функциям (разд. 2.7.2);
- сервис архивации данных - осуществляет управление архивацией данных (разд. 2.3.4).

В состав Платформы входит ППИ для разработки сервисов.

¹² service-oriented architecture (SOA)

2.5.2. Драйверы

Драйверы осуществляют сбор и предварительную обработку данных, поступающих из различных внешних источников (физических устройств, информационных систем), а также осуществляют передачу данных и/или команд управления внешним устройствам (системам).

Обмен данными с внешними системами или физическими устройствами производится по соответствующим протоколам, реализованным в виде драйверов.

Платформа поддерживает широкий спектр протоколов сопряжения, применяемых в АСУ ТП, такие как NTP, SNMP, OPC UA, Modbus, Profinet, Bacnet, MQTT, IEC-60870, IEC-61850 и др.

Платформа поддерживает широкий спектр протоколов сопряжения распространенных ИТ систем, такие как Zabbix, Trassir, Perco, FindFace, Asterisk и др.

Платформа содержит [ППИ](#) для разработки драйверов.

2.5.3. Клиентские интерфейсы

Клиенты могут подключаться к целевой системе платформы по стандартному протоколу [REST](#). Для клиентских подключений преимущественно рекомендуется использовать протокол WebSocket.

Платформа содержит ППИ для обслуживания клиентов¹³.

¹³ REST API

2.6. Вычислительная модель

Сервис вычислений Платформы предоставляет широкие возможности для обработки поступающих данных в соответствии с логикой работы Системы (разд. [2.5.1](#)).

2.6.1. Калькулятор

Платформа содержит базовый сервис, реализующий функции калькулятора. Калькулятор работает в парадигме событийной модели (разд. [2.4](#)) платформы. Это означает, что вычислительный цикл наступает только в случае изменения значения связанного объекта.

Калькулятор осуществляет вычисление переменной согласно заданной формуле или заданного алгоритма.

Базовый калькулятор поддерживает следующие математические операции:

- [+] сложение;
- [-] вычитание;
- [*] умножение;
- [/] деление;
- [^] степень;
- [=+] $a=a+b$;
- [>] больше;
- [<] меньше;
- [=] равно;
- [>=] больше или равно;
- [<=] меньше или равно;
- [!] не;
- [!=] не равно;
- [{.}] интегрирование;
- [Если] оператор ЕСЛИ
(<выражение>;
 <значение если выражение истинно>;
 <значение если выражение ложно>);
- [ЕслиМН] оператор ЕСЛИМН
(<выражение1>;
 <значение если выражение 1 истинно>;
 <значение если выражение 1 ложно>;
 ...
 <выражениеп>;
 <значение если выражениеп истинно>;
 <значение если выражениеп ложно>).

2.6.2. Имитатор

Данный сервис предназначен для генерации значений переменных по заданному закону. Как правило, он используется на стадии разработки для имитации работы реальных устройств целевой системы.

Имитатор работает в синхронном режиме, то есть осуществляет вычисления через заданный пользователем интервал времени.

Имитатор поддерживает следующие функции:

- для аналоговых переменных:
 - постоянное значение;
 - синусоидальный сигнал;
 - треугольный сигнал;
 - пилообразный сигнал;
 - прямоугольный сигнал;
 - случайное значение.
- для дискретных переменных:
 - изменение состояния на каждом шаге;
 - постоянное значение.

Имитатор может быть использован для запуска вычислительного цикла калькулятора. Например, чтобы калькулятор пересчитывал целевую переменную с заданной частотой, нужно создать дискретную переменную и имитировать ее изменение с заданной частотой; далее в целевой переменной добавить условия отслеживания дискретной переменной и задать необходимую расчетную логику.

Таким образом, имитатор может использоваться в качестве функционального элемента, задающего частоту (период) вычислений калькулятора.

2.6.3. Проектно зависимые вычисления

При необходимости, функционал калькулятора может быть расширен с помощью новых функций.

2.7. Кластерная модель

Кластерная модель Платформы поддерживает «мультиактивный»¹⁴ режим работы узлов. Преимущество такого режима в сравнении с «активный-пассивный»¹⁵ заключается в том, что все узлы кластера работают синхронно. Синхронизация узлов производится по высокоскоростному каналу связи, при этом узлы кластера могут физически находиться в одном месте или могут быть разнесены территориально.

Такая архитектура обеспечивает следующие преимущества:

- синхронизация между узлами обеспечивает резервирование данных и их высокую доступность, тем самым повышая надежность всей системы;
- резервирование компонентов целевой системы, расположенных на узлах кластера, что также повышает надежность системы;
- возможность балансировки нагрузки между активными узлами (рис. 2.3) позволяет оптимизировать использование ресурсов Платформы.



Рис. 2.3. Состав компонентов ядра

¹⁴ active-active, master-master

¹⁵ active-passive, master-slave

Архитектура клиентов кластера позволяет им поддерживать активные соединения сразу с несколькими узлами (рис. 2.4).

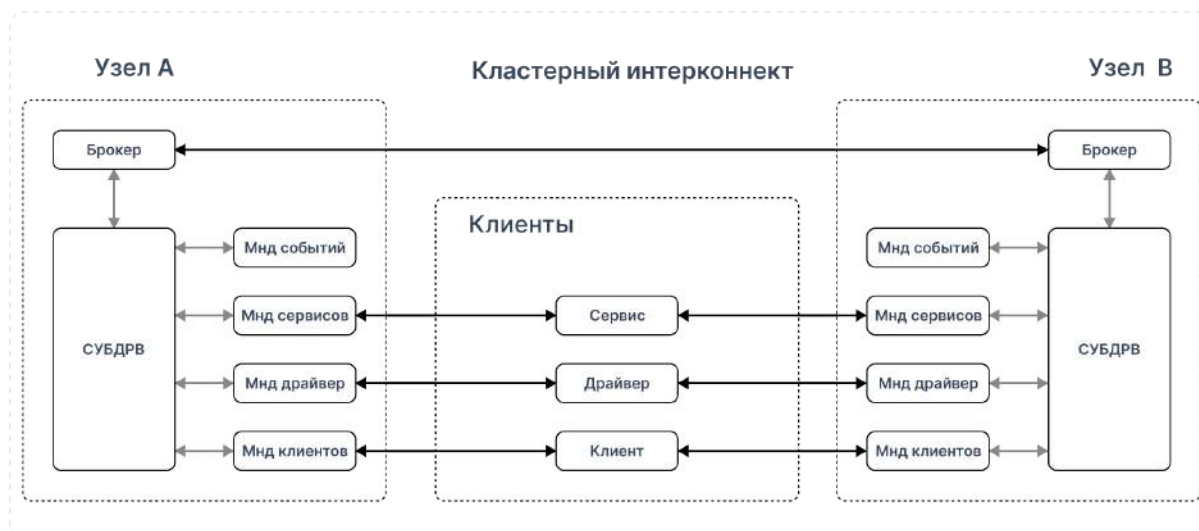


Рис. 2.4. Состав компонентов ядра

2.7.1. Кластерный узел

Кластерный узел обеспечивает такое обслуживание клиентов, при котором каждое изменение его состояния синхронизируется с другими узлами. Кластерный узел обеспечивает контроль нагрузки для того, чтобы при необходимости прекратить обслуживание части клиентов таким образом, чтобы они переключились на другой узел.

2.7.2. Интерконнект

Интерконнект - это механизм двустороннего обмена данными между кластерными узлами. Обмен данными между кластерными узлами производится по защищенному внутриплатформенному протоколу.

Планирование аппаратных ресурсов Платформы следует производить с учётом потенциальной нагрузки на кластер и интерконнект таким образом, чтобы ресурсы кластера были достаточны для обслуживания всех клиентов и канал связи обмена данными между кластерными узлами имел достаточную пропускную способность. Такое планирование рекомендуется производить с учётом особенностей и перспективного развития/расширения целевой системы.

2.7.3. Клиент кластера

Клиент кластера включает базовую платформенную библиотеку, обеспечивающую функции подключения к кластеру, поддержки соединений с кластерными узлами, переключения между кластерными узлами.

Платформа содержит развитый [ППИ](#) для разработки клиентов кластера доступных типов (разд. [2.5](#)).

2.8. Разграничение доступа

Платформа обеспечивает средства разграничения доступа к компонентам целевой системы и ограничения действий пользователей в соответствии с их ролью в Платформе.

Доступ к целевой системе осуществляется путем аутентификации с помощью логина и пароля. Аутентификация также возможна с использованием пользовательских токенов.

При аутентификации присутствует возможность выбрать систему управления пользователями - Встроенная, LDAP, Keycloak.

2.9. Диагностика

Платформа содержит встроенные средства самодиагностики.

2.10. Визуализация данных

Платформа обеспечивает пользователю возможность самостоятельного проектирования подсистем визуализации данных.

Для визуализации данных в Платформе используются различные механизмы, которые могут комбинироваться в соответствии с требованиями целевой системы.

Мнемосхемы являются основным механизмом визуализации в платформе (разд. [2.10.3](#)). Мнемосхема состоит из типовых элементов (разд. [2.10.1](#)) и вспомогательной графики, которая отражает логические связи между ними.

Мнемосхемы и типовые элементы мнемосхем разрабатываются в формате [SVG](#)¹⁶ (масштабируемой векторной графики).

¹⁶ Scalable vector graphics

В состав Платформы не входят средства создания/редактирования векторных изображений. Вместо этого предусмотрена возможность импорта графики требуемого формата (SVG), разработанной в любом доступном векторном редакторе, поддерживающем такой формат. Такой подход позволяет наиболее полно использовать все возможности профессиональных программ работы с графикой и создавать мнемосхемы и типовые элементы с требуемой в целевой системе степенью детализации.

Платформа содержит инструменты интеграции с некоторыми внешними векторными редакторами, позволяющие в автоматическом режиме импортировать векторные изображения как метаданные платформы.

В Платформе предусмотрены инструменты привязки данных к элементам мнемосхем и виджетам (разд. [2.10.5](#)).

Платформа предоставляет средства программирования логики и анимации статических векторных графических элементов в зависимости от состояния (значений переменных) отображаемых этими элементами реальных физических устройств. Таким образом, в составе мнемосхем могут быть реализованы динамические типовые элементы, виджеты, вложенные мнемосхемы¹⁷, интерактивные формы управления и функциональные приложения любой сложности (разд. [2.10.5](#)).

Данные в Платформе могут быть представлены с помощью инструментов табличного представления (разд. [2.10.6](#)), временных рядов и трендов (разд. [2.10.7](#)).

В состав Платформы также входят средства интеграции внешних библиотек визуализации, виджетов и приложений (разд. [2.10.8](#)).

2.10.1. Типовые элементы

Типовые элементы используются для визуализации на мнемосхеме состояния технологического, производственного или иного физического процесса в реальном времени. Например, типовой элемент может отображать состояние задвижки или клапана (открыто/закрыто), показания различных датчиков (температуры, давления, скорости вращения и т.п.).

¹⁷ видеограммы

Изображение типового элемента может изменять свои характеристики (цвет, размер, форму и т.д.) в зависимости от состояния визуализируемого процесса. Например, при открытой задвижке/клапане в отображающем её типовом элементе используется один цвет, а при закрытой - другой. Логика изменений этих характеристик задается с помощью встроенных в Платформу инструментов программирования (скриптов).

Типовые элементы могут быть объединены в библиотеки для удобства дальнейшего их использования.

2.10.2. Виджеты

Более сложными составляющими визуальной среды Платформы являются виджеты - небольшие графические приложения, работающие в среде исполнения в составе мнемосхемы или отдельно. Как правило, виджет состоит из нескольких типовых элементов, но может быть реализован и обособленно. Примерами таких виджетов могут быть изображения стрелочных или индикаторных приборов, таких как термометр, барометр, вольтметр и т.д.

Виджеты могут отображать как текущее (мгновенное) состояние процесса, так и изменение состояния параметров процесса во времени. Такими виджетами являются, например, графики, тренды, гистограммы и т.д.

Для интеграции сложного контента в виджет предусмотрен механизм фрагмента - это кадр, встроенный в мнемосхему, в котором размещается результат работы внешних или встроенных в Платформу приложений. В отличие от остальных типовых элементов и виджетов, содержимое фрагмента может быть отображено только в режиме исполнения.

Платформа содержит набор стандартных виджетов для визуализации данных:

- виджет отображения объектов на карте;
- виджет трендов;
- виджет событий;
- виджет вывода изображения с камеры.

В Платформе для обозначения типовых элементов, виджетов и фрагментов используется термин «компоненты» (визуальные компоненты).

2.10.3. Мнемосхемы

Основной формой визуализации технологического или производственного процесса является мнемосхема, которая включает в себя набор элементов, виджетов и графических изображений, наглядно описывающий входящие в состав управляемой системы физические объекты (приборы, устройства, исполнительные механизмы), а также связи между ними (трубопроводы, цепи питания и т.п.). Состав мнемосхемы можно разделить на несколько групп:

- статические элементы - отображают числовые или логические значения переменных (сигналов);
- динамические элементы - в зависимости от значения сигналов изменяют свои графические характеристики;
- виджеты (разд. [2.10.2](#));
- вспомогательные графические элементы - изображение связей между компонентами системы.

В большинстве АСУ мнемосхемы располагаются на мониторе оператора, экране коллективного пользования, панелях визуализации и т.п. Однако в отдельных отраслях часто используются многоэкранные системы, объединенные в единый информационный стенд. Инструменты Платформы предоставляют возможность масштабирования и адаптации мнемосхем для многоэкранных систем отображения информации.

2.10.4. Привязка переменных

Подготовленная мнемосхема является лишь графическим изображением технологического процесса. Для того чтобы мнемосхема стала частью управляющей Платформы, необходимо сопоставить данные (переменные), поступающие от целевой системы, и элементы мнемосхемы.

На стадии проектирования системы для каждого типового элемента и/или виджета должна быть определена соответствующая абстрактная модель (разд. [2.3.6](#)). В дальнейшем это упростит привязку переменной (объекта) к типовому элементу, входящему в состав мнемосхемы. Кроме того, это предотвратит вероятность создания ошибочной связки «переменная-элемент-модель», т.к. среда разработки просто не позволит этого сделать.

Каждой абстрактной модели может быть поставлено в соответствие несколько разных типовых элементов и/или виджетов. Например, значение температуры может быть отображено на мнемосхеме разными способами: как типовым элементом типа «счётчик», так и виджетом типа «термометр». Разумеется, эти элементы должны быть предварительно определены (см [2.10.1](#)).

Обратная связь между элементами и моделями также имеет место. Каждый типовой элемент или виджет может быть связан с несколькими абстрактными моделями одного типа. Так, элемент типа «счётчик», отображающий некоторое числовое значение «float», может отображать на мнемосхеме значение любой переменной соответствующего типа. Такой подход значительно упрощает создание мнемосхем, так как для отображения различных числовых значений температуры, давления, скорости ветра и т.д. не нужно создавать отдельные элементы, а нужно лишь разместить на мнемосхеме несколько экземпляров типового элемента «счётчик» и связать их с соответствующими переменными.

2.10.5. Программирование

В Платформе существует два вида типовых элементов - статические и динамические.

Статические элементы, как правило, просто отображают значения числовых переменных - их часто называют «показометры».

Динамические элементы, к которым можно отнести и виджеты, могут изменять свои графические характеристики (цвет, размер, положение на экране и т.д.) в зависимости от значения переменных, с которыми они связаны. Для любого типового элемента может быть написан программный код (скрипт), с помощью которого разработчик Платформы может определить такие изменения. Такой скрипт может быть написан как для типового элемента в общем случае, так и для типового элемента в контексте конкретной мнемосхемы.

Интегрированная среда разработки (ИСР) предоставляет инструменты написания сценариев и программного кода на поддерживаемом Платформой языке (TypeScript).

Эти инструменты включают в себя:

- встроенный редактор кода с подсветкой синтаксиса;
- перечень доступных платформенных функций.

Платформа позволяет программировать как поведение мнемосхемы целиком, так и поведение отдельных её элементов, с учётом специфики технологических процессов системы.

Таким образом, программные возможности Платформы позволяют разрабатывать полноценные интерактивные мультиплатформенные приложения для самых разных устройств, от мобильного планшета до рабочей станции и мультиэкранного стенда.

2.10.6. Табличное представление данных

Оперативные и архивные данные Платформы могут быть представлены в виде таблиц следующих видов:

- срез текущих значений, определенных пользователем переменных;
- изменения текущих значений переменных, определенных пользователем;
- относительных изменений текущих значений на временной шкале.

Текущие данные, а также запросы архивных данных (среза или интервала значений), также могут быть представлены в табличной форме.

Данные в табличной форме могут быть экспортированы во внешний файл или выведены на печать.

2.10.7. Визуализация временных рядов

Оперативные и архивные данные Платформы могут быть представлены в виде временных рядов:

- тренды распределенные по вертикали;
- наложенные тренды (разных переменных) с горизонтальным распределением осей значений.

Тренды могут быть представлены как без локальной экстраполяции данных, так и в режиме экстраполяционного сглаживания.

Текущие данные и результаты запроса к архивным данным также могут быть отображены в виде временных рядов с заданным интервалом значений. Предусмотрен интерфейс быстрого обращения к архивным данным за любой период.

Формы представления данных временных рядов могут быть экспортированы во внешний файл или выведены на печать.

2.10.8. Внешние приложения и виджеты

К Платформе могут быть подключены внешние визуальные библиотеки, содержащие виджеты, компоненты, тренды и т.д. Для подключения внешних библиотек используются встроенные в мнемосхему кадры (фрагменты). При этом сами библиотеки могут быть запущены с сервера стороннего разработчика при наличии соответствующего интерфейса и если это не противоречит требованиям информационной безопасности предприятия, на котором функционирует Платформа.

2.11. Импорт и экспорт

В Платформе предусмотрена возможность импорта и экспорта данных как оперативных/архивных, так и данных конфигурации Платформы.

Для удобства разработки или развертывания типовых проектов автоматизации целевых систем Платформа предоставляет инструменты экспорта данных конфигурации проекта в табличный файл и импорта этих данных из файла.

Содержание файла импорта/экспорта:

- список используемых моделей и их настройки;
- список объектов и их настройки;
- конфигурация событийной модели;
- привязки переменных к элементам мнемосхем;
- список ролей и пользователей Платформы.

Типовые элементы, виджеты и мнемосхемы могут быть импортированы в Платформу из файлов формата SVG. Кроме того, импорт таких данных может быть произведён в автоматизированном режиме из векторного редактора.

3. Интегрированная среда разработки

Интегрированная в состав Платформы среда разработки предназначена для проектирования целевых систем и предоставляет для этой цели встроенные инструменты:

- редактор/конфигуратор моделей (разд. [2.3.6](#));
- редактор/конфигуратор объектов (разд. [2.3.7](#));
- конфигуратор событий и тревог (разд. [2.4](#));
- конфигуратор драйверов и сервисов (разд. [2.5](#));
- конфигуратор вычислительной логики (разд. [2.6](#));
- конфигуратор кластера (разд. [2.7](#));
- конфигуратор СРД (разд. [2.8](#));
- конфигуратор мнемосхем и компонентов (разд. [2.10](#)).

3.1. Структура данных проекта

В состав платформы входит базовый типовой набор моделей, объектов, стандартная библиотека типовых компонентов, стандартный набор драйверов. С учётом специфики конкретного проекта технологической системы дополнительно должны быть определены:

- проектные модели (разд. 3.2);
- проектные объекты (разд. 3.3);
- проектные компоненты (разд. 3.4);
- проектные мнемосхемы и виджеты (разд. 3.5);
- проектный набор драйверов (разд. 3.6) и сервисов (разд. 3.8).

В процессе проектирования должен быть определен и сформулирован перечень событий для каждого элемента системы (разд. 3.10).

Кроме того, должны быть определены роли пользователей и для каждой из ролей сконфигурирована политика разграничения доступа к разделам Платформы (разд. 3.11).

Доступ к набору вышеперечисленных инструментов предоставляется на главном экране среды разработки (рис. 3.1).

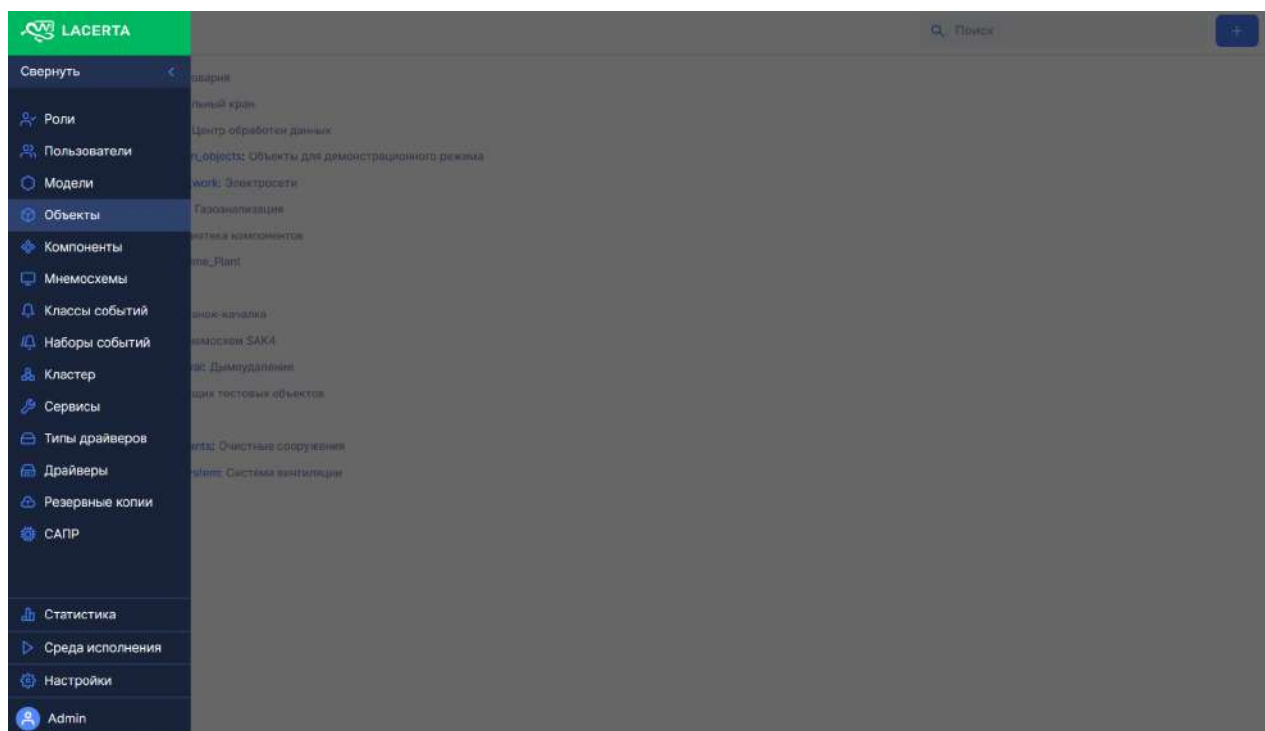


Рис. 3.1. Основное меню среды разработки

3.2. Редактор моделей

Редактор моделей (разд. 2.3.6) представляет собой форму с иерархической навигацией. При нажатии на модель справа отобразится форма описания атрибутов выбранной модели (рис. 3.2). Пользователю также будут доступны изменения атрибутов модели, возможность поиска по моделям/атрибутам, создания и удаления моделей и атрибутов.



Рис. 3.2. Интерфейс обзора конфигурации модели

Цифрами на рис. 3.2 обозначены:

1. Поиск;
2. Создание модели;
3. Удаление модели/атрибута;
4. Создание атрибута модели.

У базовых моделей по типу Boolean, Directory, Float, Integer, Link, String стоит ограничение на их изменение и удаление, о чем сигнализирует соответствующая пиктограмма, при наведение на которую всплывает предупреждающее сообщение, которое также можно увидеть в обзоре конфигурации модели (рис. 3.3).



Рис. 3.3. Ограничение на изменение и удаление базовых моделей

При добавлении новых атрибутов у модели они также появятся и у объекта, созданного на основе этой модели, но при условии, что в этом объекте ничего не менялось самостоятельно. Если в объекте что-то изменить (например, добавить атрибут), то связь с моделью теряется до тех пор, пока объект не «восстановить», тем самым убрав самостоятельные изменения в нём.

Если модель создана на основе существующей (материнской) модели, то в обзоре конфигурации модели будет отражена соответствующая информация (рис. 3.4).



Рис. 3.4. Интерфейс обзора конфигурации дочерней модели

Модель может соответствовать нескольким типовым графическим элементам, расположенным в разделе «компоненты», свернутом по умолчанию (рис. 3.5). Нажатием кнопки «+ Добавить компонент» пользователь может добавить новые графические элементы модели.

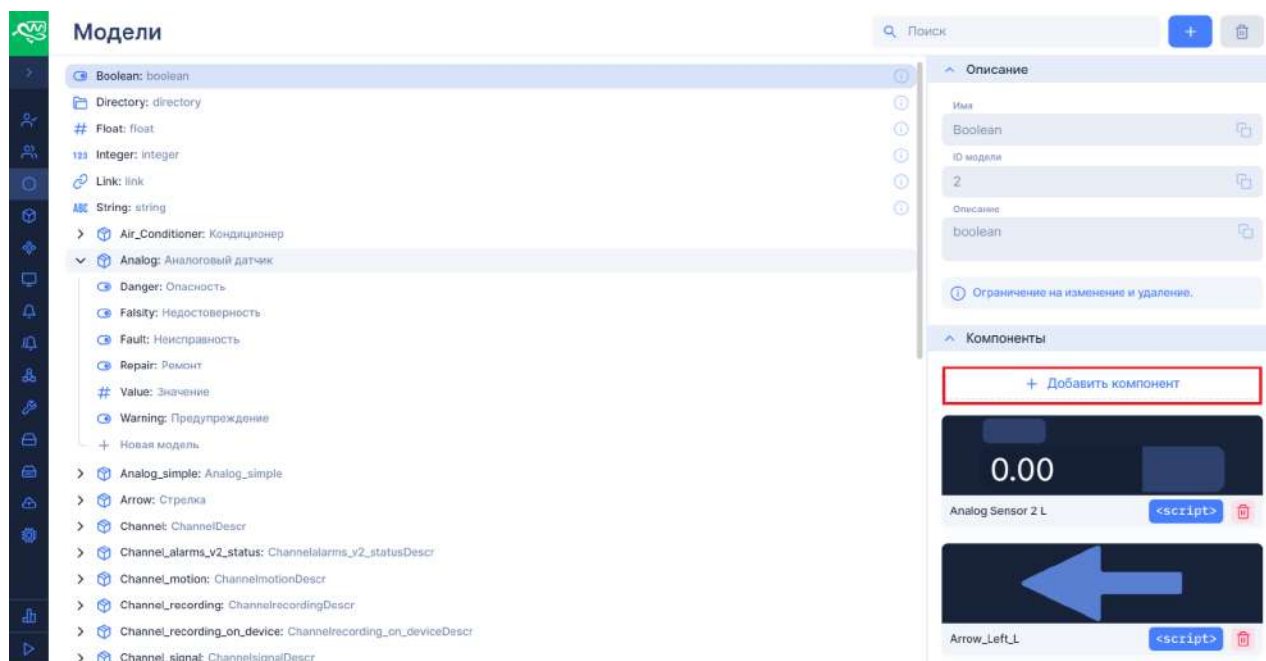


Рис. 3.5. Интерфейс обзора конфигурации дочерней модели

В системе имеется возможность редактирования скрипта компонента. Наличие добавленного скрипта в компонент отображается соответствующим состоянием кнопки «Script» (рис. 3.6).

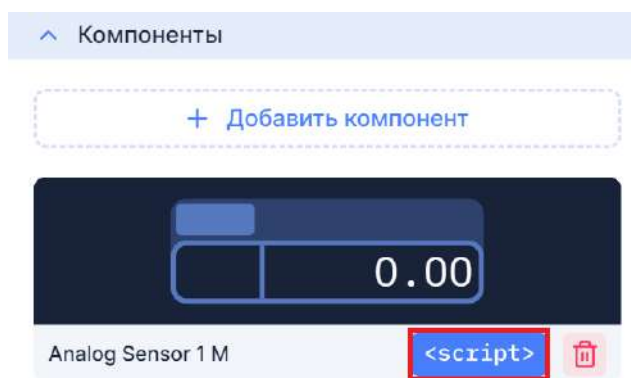


Рис. 3.6. Кнопка редактирования скрипта

Типовые корневые модели, соответствующие базовым типам данных, принятым в платформе (разд. 2.3.5):

- Boolean (разд. 3.2.1) - логического типа (да/нет);
- Directory (разд. 3.2.2) - составного типа;
- Float (разд. 3.2.3) - числовые с плавающей точкой;
- Integer (разд. 3.2.4) - числовые, целочисленные;
- Link (разд. 3.2.6) - ссылочного типа;
- String (разд. 3.2.5) - строкового типа.

Каждая типовая модель имеет свое графическое обозначение (рис. 3.7).

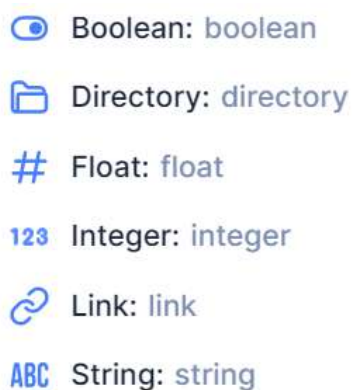


Рис. 3.7. Список типовых моделей

Составные производные модели, состоящие из других подмоделей, имеют следующее графическое изображение (рис. 3.8).

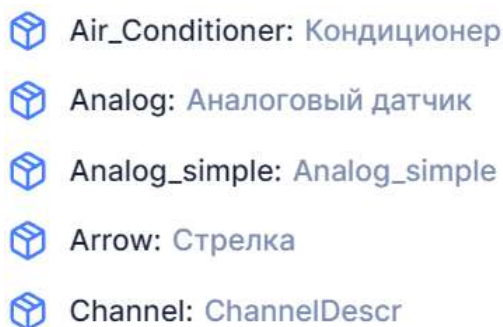


Рис. 3.8. Отображение сложных моделей

3.2.1. Модель Boolean

Модель Boolean является базовой моделью и предназначена для описания дискретных типов данных (рис. 3.9). Данная модель связана с изменением значения логической переменной (типа Boolean).



Рис. 3.9. Модель Boolean

3.2.2. Модель Directory

Базовая модель Directory предназначена для группировки объектов (описания сложных объектов древовидной структуры, рис. 3.10). С помощью модели этого типа может быть описан сложный объект или группа более простых объектов. Например, метеостанция, включающая в себя комплекс приборов (термометры, гигрометры, барометры и т.п.), описывается моделью Directory, которая, в свою очередь, может содержать вложенные модели, описывающие каждый конкретный прибор. Степень вложенности моделей при этом может быть сколь угодно большой в зависимости от особенностей технологической системы. Данная модель не связана с изменениями значений переменных заданного типа.



Рис. 3.10. Модель Directory

3.2.3. Модель Float

Модель Float является базовой и предназначена для описания числовых типов данных (рис. 3.11). Она связана с изменениями значений переменных типа Float.

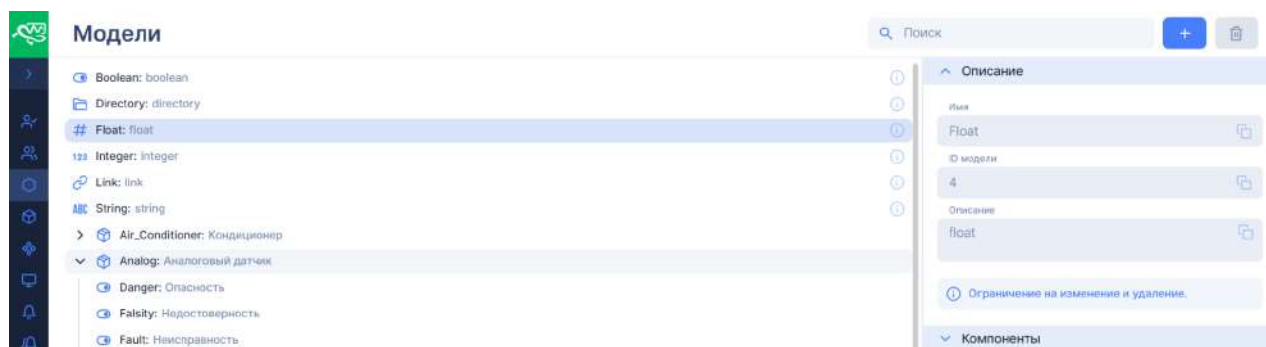


Рис. 3.11. Модель Float

3.2.4. Модель Integer

Базовая модель Integer предназначена для описания целочисленных типов данных (рис. 3.12). Она связана с изменениями значений переменных типа Integer.



Рис. 3.12. Модель Integer

3.2.5. Модель Link

Базовая модель Link является ссылочным типом данных, переменная которого хранит ссылку на объект или структуру данных, а не сам объект. Когда объект изменяется через ссылку, изменения затрагивают исходный объект, так как обе ссылки (переменная и объект) указывают на одно место в памяти. Модель предназначена для хранения не самих данных, а адресов (ссылок) на данные, хранящиеся в памяти. Это противоположность типам данных, которые содержат сами значения (рис. 3.13). По данной модели можно создавать объекты и к этим объектам привязывать другие.



Рис. 3.13. Ссылочная модель Link

3.2.6. Модель String

Модель String является базовой моделью и предназначена для описания строковых переменных (рис. 3.14).



Рис. 3.14. Модель String

3.2.7 Составные модели

Составные модели являются основным инструментом создания сложных структур данных целевой системы.

В качестве примера можно привести модель Analog, отвечающую за описание простейшего аналогового датчика. Модель включает в себя пять дискретных моделей и одну аналоговую.

Аналоговая модель предназначена для хранения значения переменной аналогового датчика. Дискретные модели предназначены для хранения диагностических дискретных параметров признаков, таких как: недоверность, неисправность, опасность, ремонт (рис. 3.15).

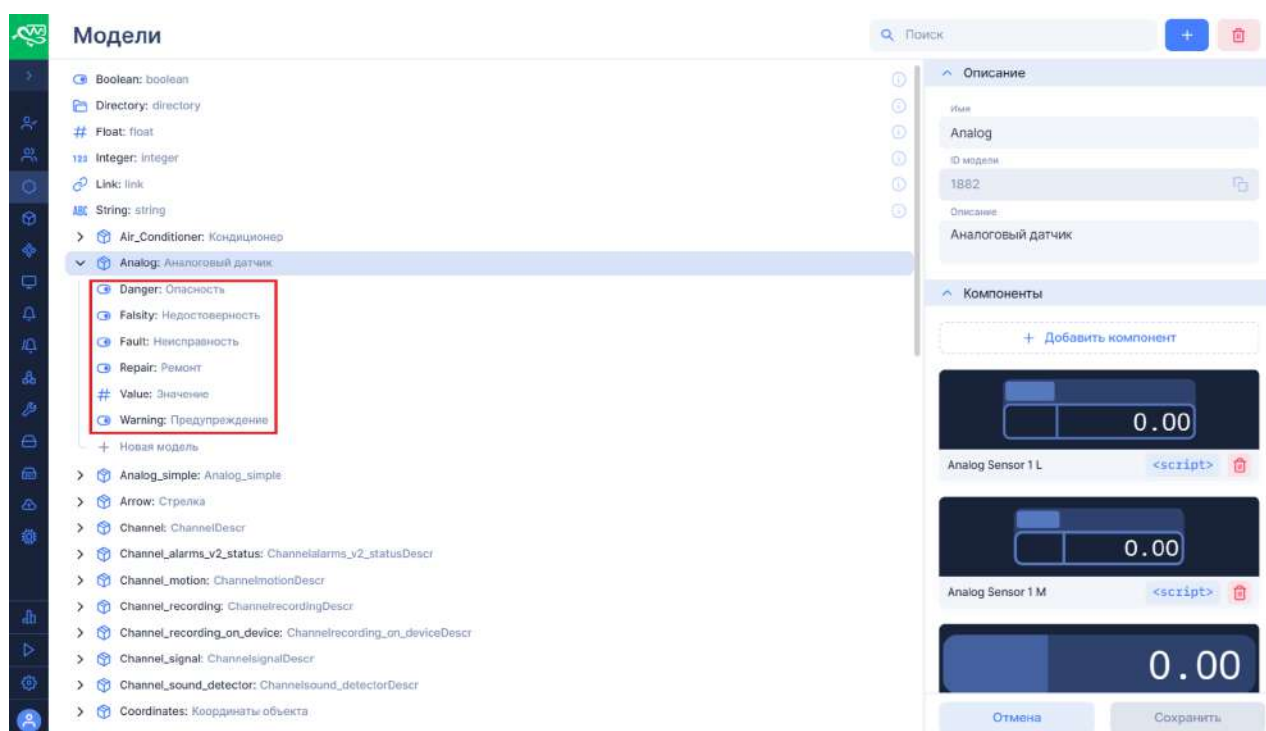


Рис. 3.15. Модель Analog

В качестве еще одного примера можно привести модель Discrete, описывающую простейший дискретный датчик. Модель состоит из четырех вложенных дискретных моделей.

Одна из вложенных дискретных моделей предназначена для хранения значения переменной дискретного датчика. Остальные предназначены для хранения диагностических дискретных параметров-признаков (флагов), таких как: недостоверность, неисправность, ремонт (рис. 3.16).

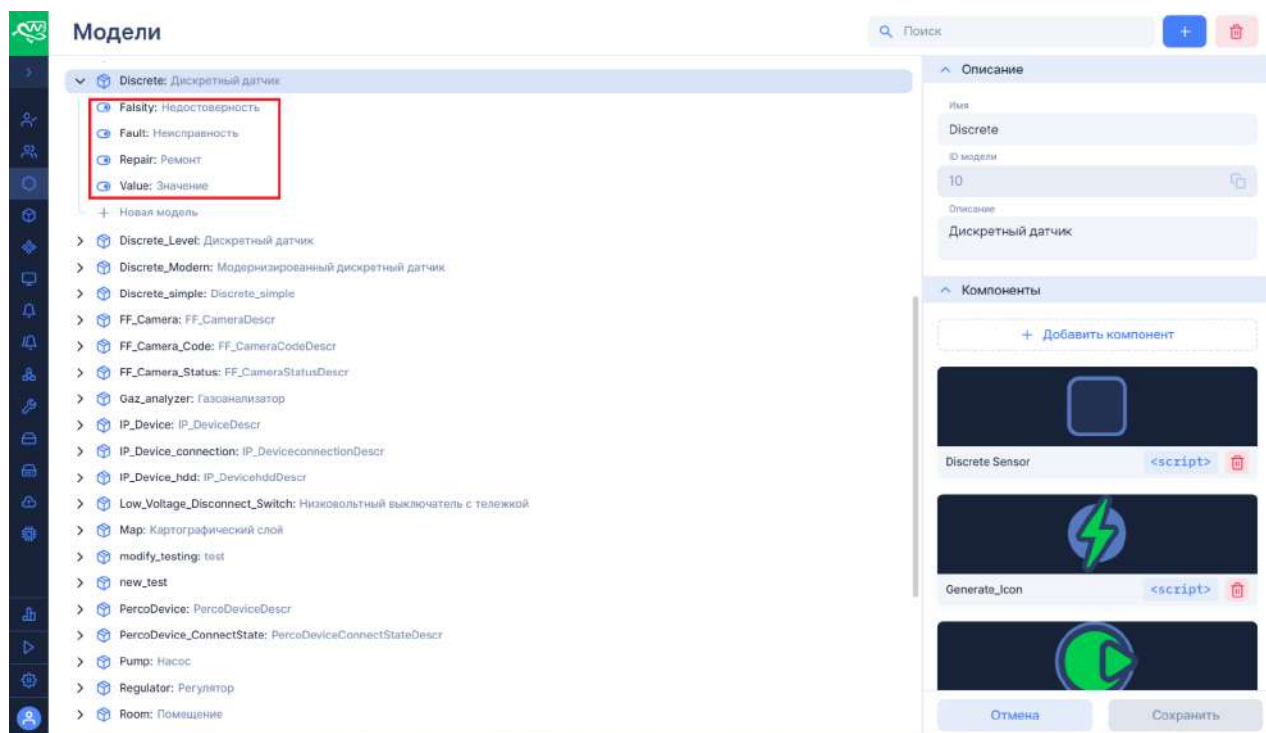


Рис. 3.16. Модель Discrete

Таким образом, пользователь платформы может создать структуру, описывающую модель любой сложности.

3.2.8. Создание новой модели

Для создания корневой модели необходимо нажать на кнопку «+» (рис. 3.17).

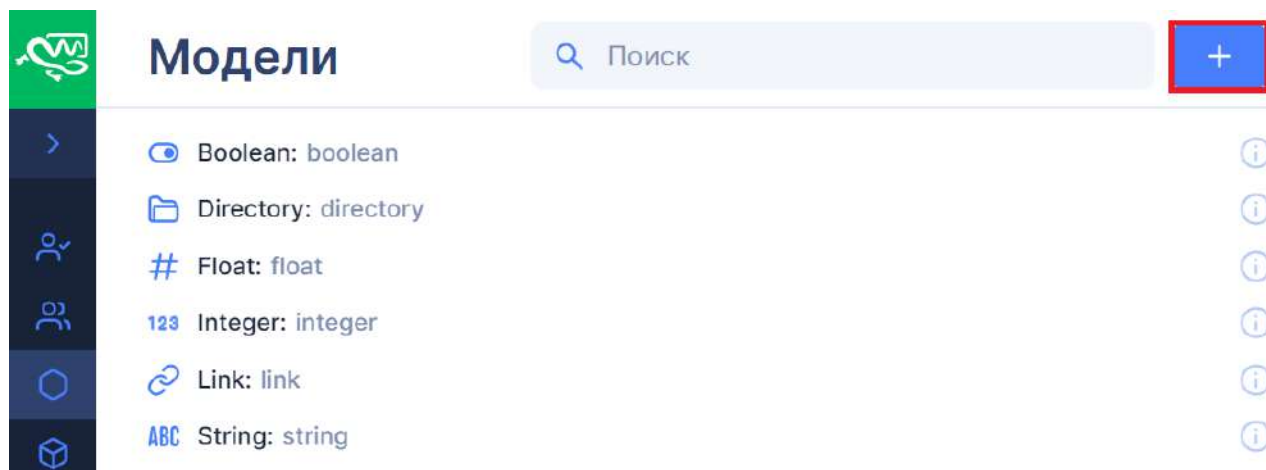


Рис. 3.17. Создание новой корневой модели

Далее задать наименование модели и ее описание, после чего нажать на кнопку «Создать» (рис. 3.18).

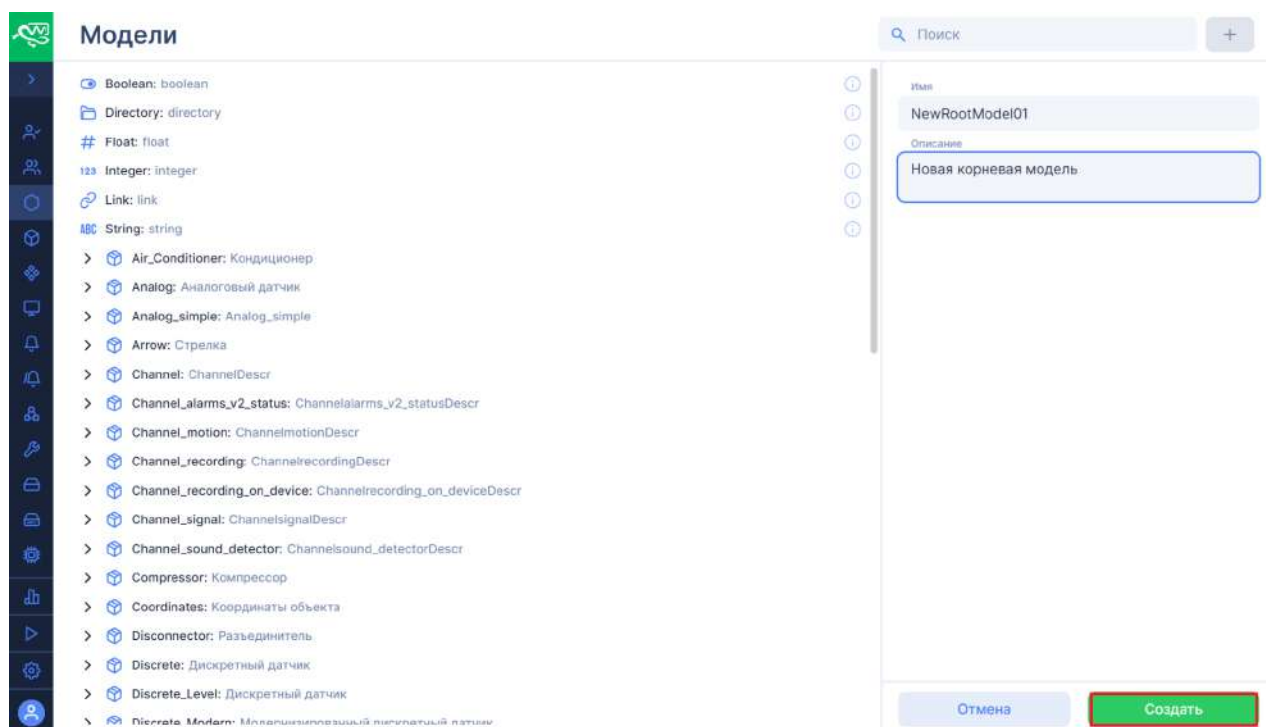
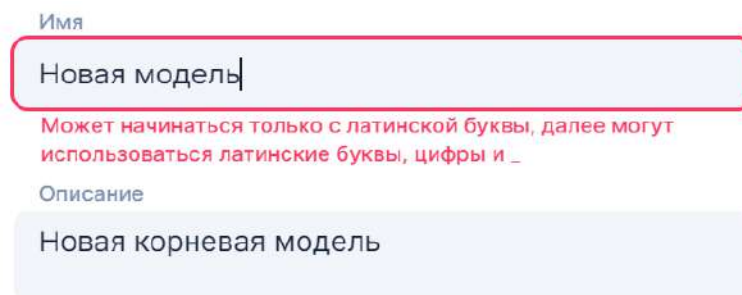


Рис. 3.18. Параметры модели

Наименование модели должно быть задано с использованием символов латинского алфавита, цифр и знака «_», иначе будет выдано предупреждение (рис. 3.19).



Имя

Новая модель

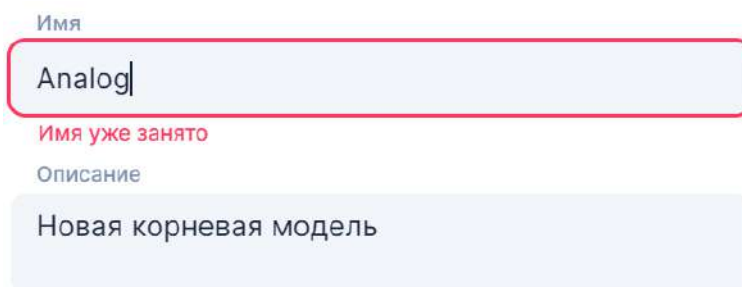
Может начинаться только с латинской буквы, далее могут использоваться латинские буквы, цифры и _

Описание

Новая корневая модель

Рис. 3.19. Предупреждение наименования модели, раскладка

Если наименование модели уже используется в Платформе, пользователю будет выдано соответствующее предупреждение (рис. 3.20).



Имя

Analog

Имя уже занято

Описание

Новая корневая модель

Рис. 3.20. Предупреждение наименования модели, название занято

В случае, если все поля заполнены корректно, будет создана новая корневая модель (рис. 3.21).



Рис. 3.21. Создание корневой модели

Для добавления вложенной модели в корневую необходимо ее раскрыть и нажать «+ Новая модель». После чего задать имя, описание и выбрать наследуемую модель. По завершении нажать «Создать» (рис. 3.22).

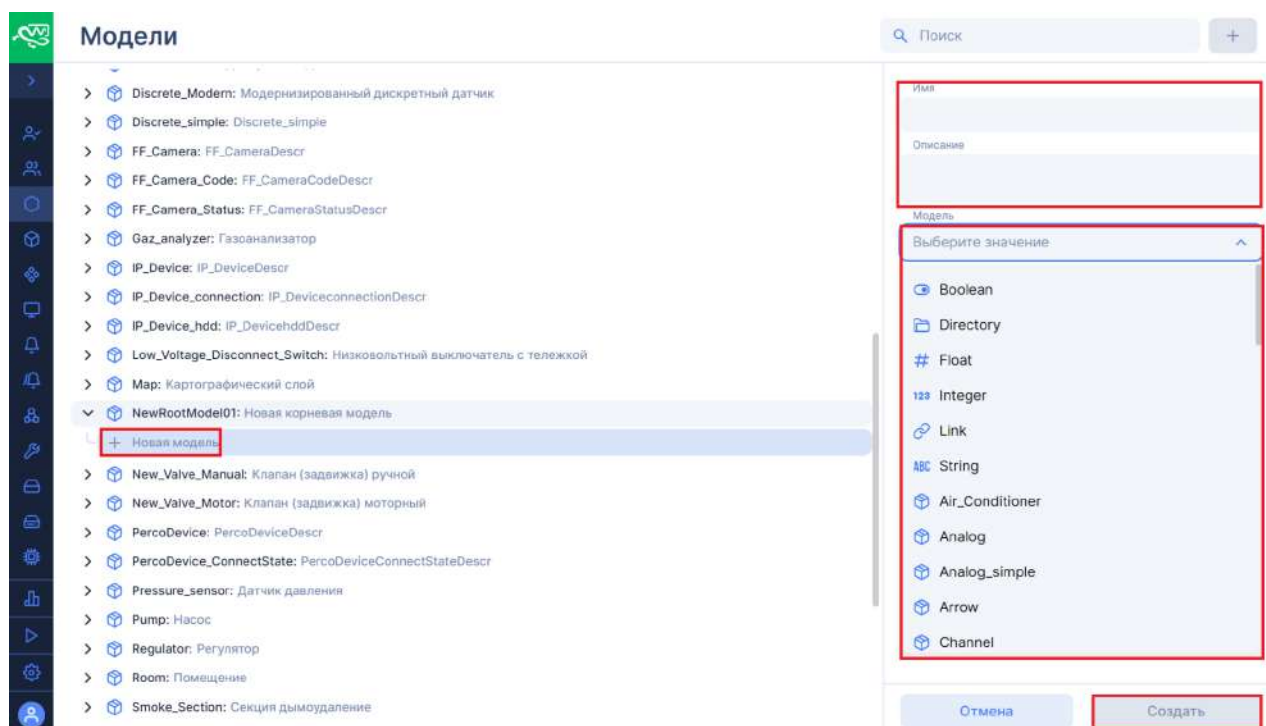


Рис. 3.22. Создание вложенной модели

Таким образом, могут быть созданы модели, описывающие структуры любой сложности.

3.2.9. Удаление модели

Для того чтобы удалить корневую модель со всеми вложенными или отдельно составную модель, необходимо выбрать модель и нажать на кнопку удаления с соответствующим обозначением (рис. 3.23).

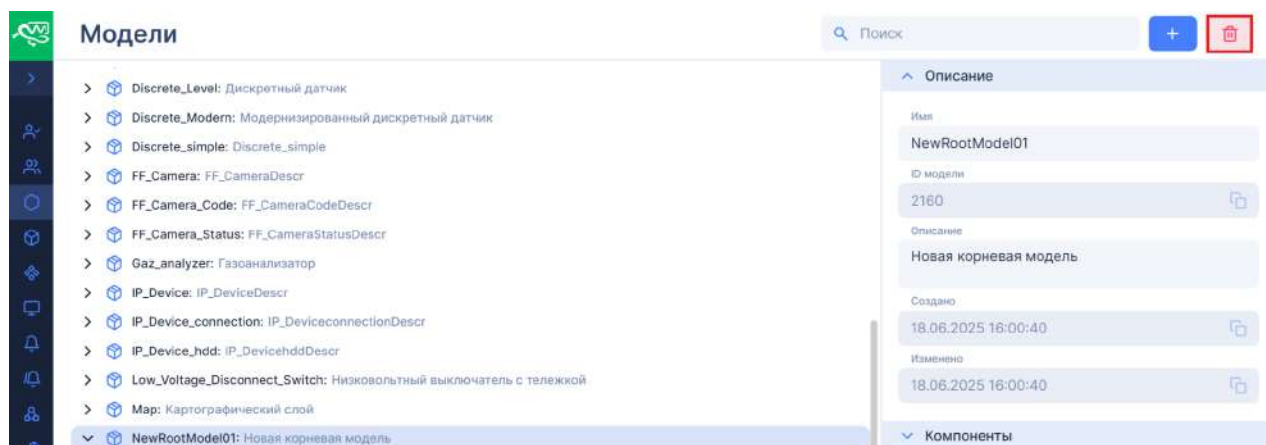


Рис. 3.23. Кнопка удаления

После чего возникнет всплывающее окно для подтверждения удаления модели (рис. 3.24).

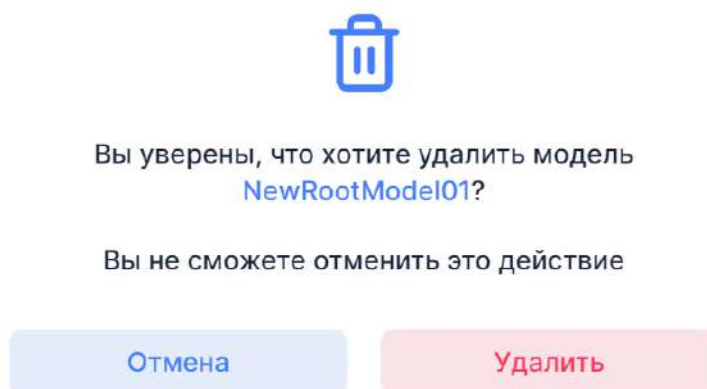


Рис. 3.24. Подтверждение удаления модели

В случае подтверждения удаления, возникнет второе окно с сообщением о возможности сохранения модели, но удаления связи зависимых объектов (рис. 3.25).

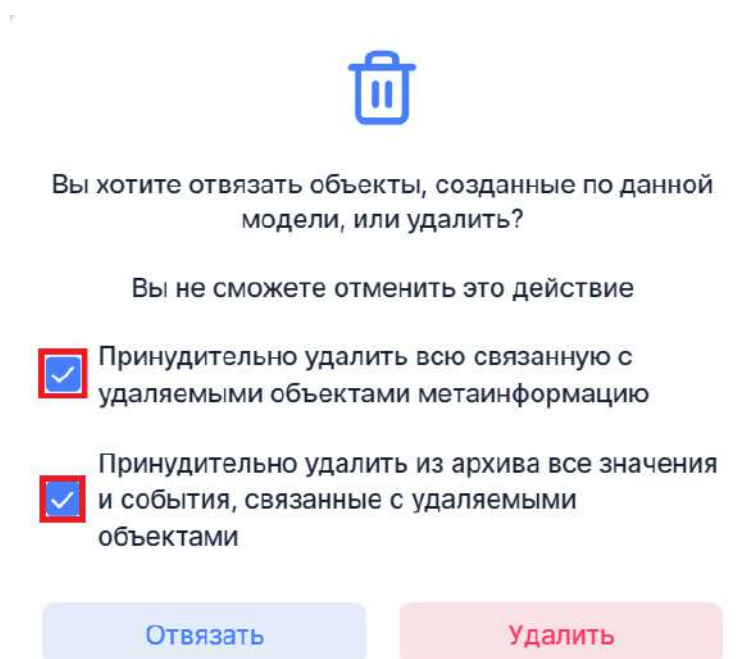


Рис. 3.25. Подтверждение удаления модели или отвязки

При выборе «отвязать» связи с объектами, созданными на базе удаляемой модели, будут удалены и их материнской моделью станет модель Directory. В случае выбора «удалить» все объекты, созданные на основе удаляемой модели, будут также удалены вместе с ней. При удалении объектов можно принудительно удалить всю связанную с ними метainформацию, а также значения и события из архива, отметив нужное флагом (рис.3.25).

3.3. Редактор объектов

Редактор объектов (разд. 2.3.7) представляет собой форму с иерархической навигацией. При выборе объекта - справа откроется конфигуратор его параметров. Для раскрытия дерева объектов (атрибутов) необходимо нажать на кнопку «>» перед объектом, напротив каждого атрибута будет его описание. Также, присутствует возможность поиска по объектам и атрибутам, создания, удаления и восстановления объекта (рис. 3.26).

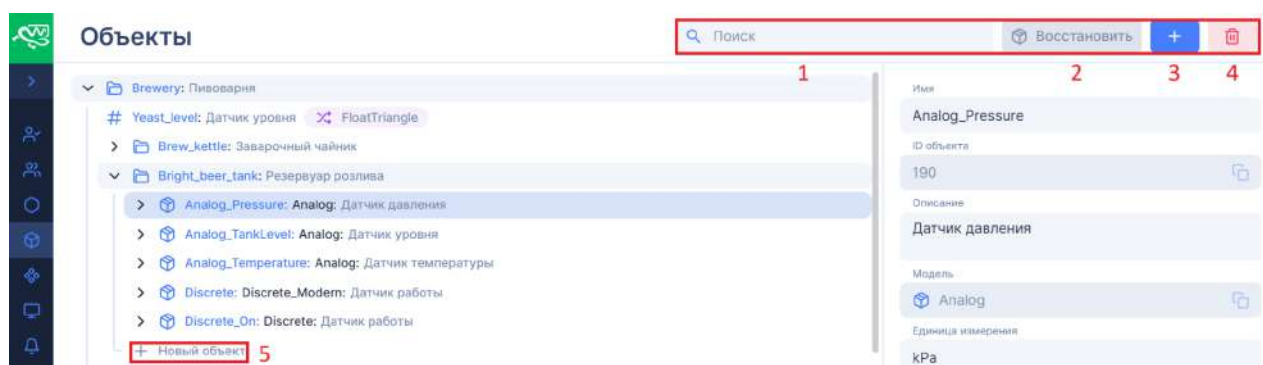


Рис. 3.26. Редактор объектов

Цифрами на рис. 3.26 обозначены:

1. Поиск;
2. Кнопка восстановления;
3. Создание объекта;
4. Удаление объекта/атрибута;
5. Создание атрибута объекта.

Кнопка восстановления приводит выбранный объект к структуре модели, удаляя все «лишние» (пользовательские) атрибуты и возвращая недостающие (рис. 3.27).



Рис. 3.27. Кнопка восстановления

3.3.1. Параметры объектов

В конфигураторе пользователь может задать параметры объектов, например аналоговых и дискретных (рис. 3.28).

Имя	Имя
Value	Danger
ID объекта	ID объекта
9	10060
Описание	Описание
АН: Значение	АН: Опасность
Модель	Модель
# Float	Boolean
Единица измерения	Единица измерения
Значение	Значение
95.44510232132502	false

Рис. 3. 28. Значения переменных объектов

Для аналоговых и дискретных объектов набор параметров будет отличаться. Редактор параметров позволяет подключить только один тип из источников данных (рис. 3.30):

- подключение драйвера сопряжения (разд. 2.5.2, 3.3.2);
- имитация изменений (разд. 2.6.2);
- вычисление расчетных переменных (разд. 2.6.1);
- установление значения вручную.

Драйвер	☒
Имитация	☐
Вычисление	☐
Ручной источник	☐

Рис. 3. 30. Типовые параметры объекта

Источники данных объекта делятся на два блока - «Запись» и «Чтение». «Запись» - это запись из источника в объект, а «Чтение» - это запись значений в источник (драйвер, рис. 3.31).

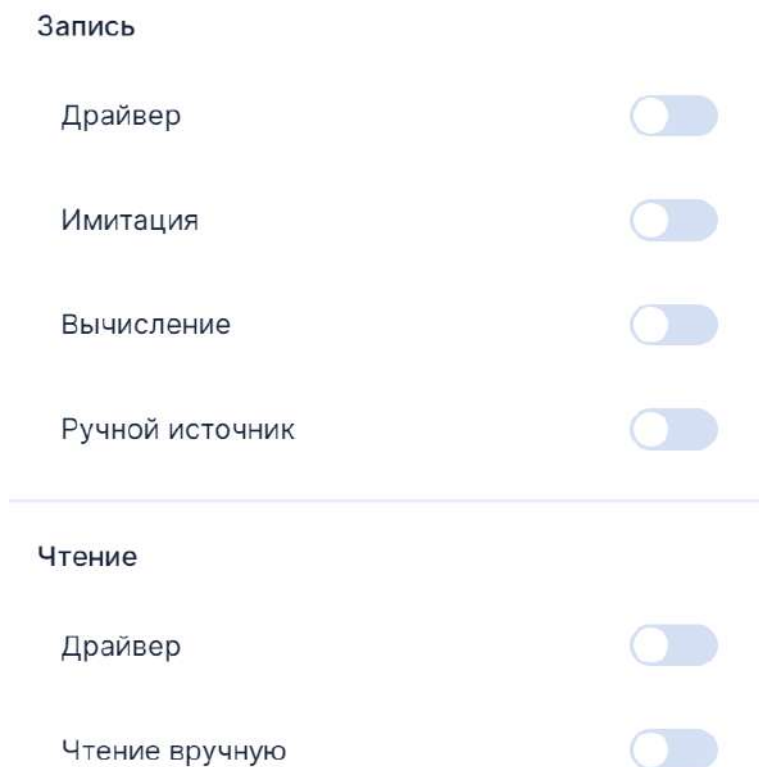


Рис. 3.31. Блоки источников данных

Для удаление заданных настроек любого источника можно воспользоваться соответствующей кнопкой (рис. 3.31).

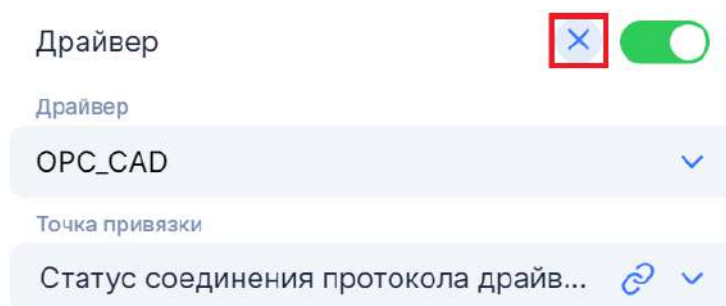


Рис. 3.31. Удаление настроек источника

3.3.2. Привязка драйвера к переменным объекта

В редакторе параметров объектов для привязки источника или цели изменений данных через драйвер необходимо активировать опцию «Драйвер» (рис. 3.32).

Драйвер ✕ ☒

Драйвер

Выберите драйвер ▼

Точка привязки

Выберите объект для привязки ▼

Рис. 3.32. Привязка драйвера к переменной

Далее в выпадающем списке выбрать требуемый драйвер из списка доступных в системе (рис. 3.33).

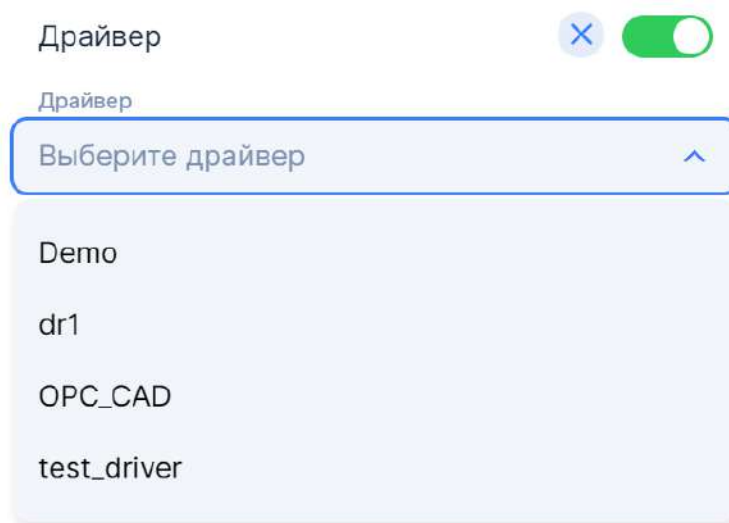


Рис. 3.33. Выбор экземпляра драйвера

После чего задать адрес точки привязки драйвера (рис. 3.34).

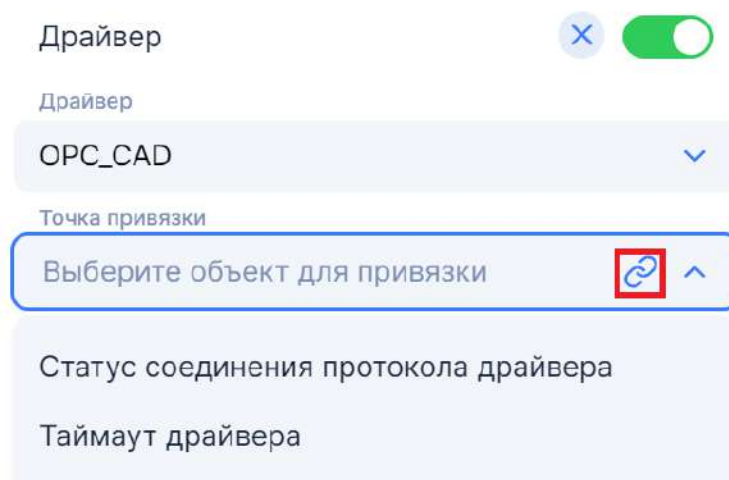


Рис. 3.34. Выбор адреса переменной драйвера.

Задать адрес привязки можно с помощью вызова специального интерфейса обзора доступных точек привязок (тегов) драйвера либо задать его вручную. Для каждого типа драйвера существует свой формат адресации и свой интерфейс обзора точек привязки.

Если у драйвера есть дерево, то в качестве привязки можно выбрать либо объект из драйверного дерева, либо из списка статусов (рис. 3.34). При отсутствии дерева можно задать точку привязки самостоятельно либо выбрать из списка статусов.

По завершении настройки необходимо сохранить изменения. После чего конфигурируемый объект будет отмечен графическим признаком операции с уточнением наименования выбранного драйвера (рис. 3.37).



Рис. 3.37. Графический признак применения драйвера

3.3.3. Настройка имитации переменных объектов

Для имитации значений изменений объекта в редакторе параметров объектов необходимо активировать опцию «Имитация» и в выпадающем списке выбрать требуемый тип имитации из списка доступных в системе (рис. 3.38).

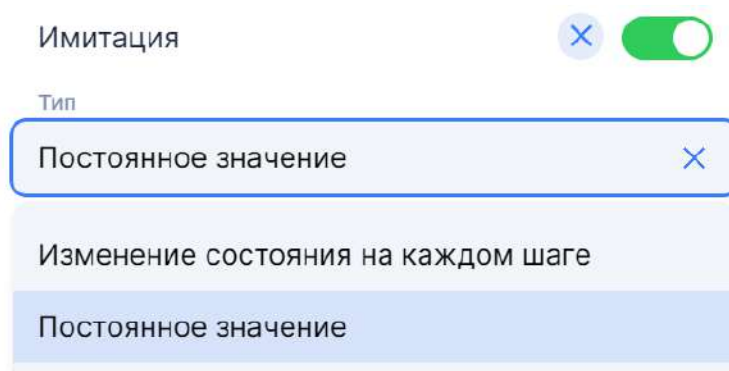


Рис. 3.38. Настройка имитации переменной

Доступные типы имитации для объектов будут отличаться в зависимости от модели на основании которой они созданы. Например, тип имитации Случайное значение будет доступен только для объектов созданных по моделям Integer и Float (таблица 1).

Значение поля	Описание сигнала	Тип значения
1	Переключение состояния на каждом шаге (дискрете)	Bool
2	Постоянное значение	Bool
3	Постоянное значение	Float
4	Синусоидальный сигнал	Float
5	Треугольный сигнал	Float
6	Пилообразный сигнал	Float
7	Прямоугольный сигнал	Float
8	Постоянное значение	Int
9	Синусоидальный сигнал	Int
10	Треугольный сигнал	Int
11	Пилообразный сигнал	Int
12	Прямоугольный сигнал	Int
13	Случайное значение	Float
14	Случайное значение	Int

Таблица 1. Доступные типы имитации

Каждый тип имеет свой набор параметров имитации. Например для типа Постоянное значение - это период измерений и значение (рис. 3.39).

The screenshot shows the 'Имитация' (Simulation) settings. At the top, there is a toggle switch labeled 'Имитация' which is turned on (green). Below it, the 'Тип' (Type) is set to 'Постоянное значение' (Constant value). Underneath, the 'Период измерений (мс)' (Measurement period (ms)) is set to '1000'. Finally, the 'Значение' (Value) is set to 'False'.

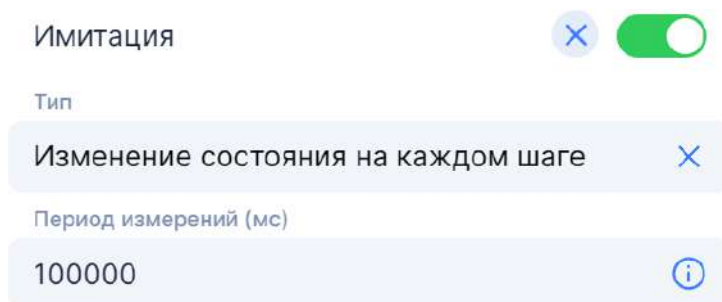
Рис. 3.39. Настройка параметров функции имитации

Для других типов имитации данных (Синусоидальный сигнал, Треугольный сигнал, Пилообразный сигнал, Прямоугольный сигнал) существует иной набор параметров (рис. 3.40).

The screenshot shows the 'Имитация' (Simulation) settings for a 'Синусоидальный сигнал' (Sinusoidal signal). The 'Имитация' toggle is on. The 'Тип' (Type) is 'Синусоидальный сигнал'. The 'Период функции (сек)' (Function period (sec)) is set to '5'. The 'Амплитуда' (Amplitude) is set to '1'. The 'Период измерений (мс)' (Measurement period (ms)) is set to '1000'. The 'Смещение по вертикали' (Vertical offset) is set to '30'. The 'Смещение по горизонтали (мс)' (Horizontal offset (ms)) is set to '300'.

Рис. 3.40. Набор параметров

Для аналоговых объектов (Изменение состояния на каждом шаге, Постоянное значение) только период измерений (рис. 3.40).



Имитация ☒

Тип

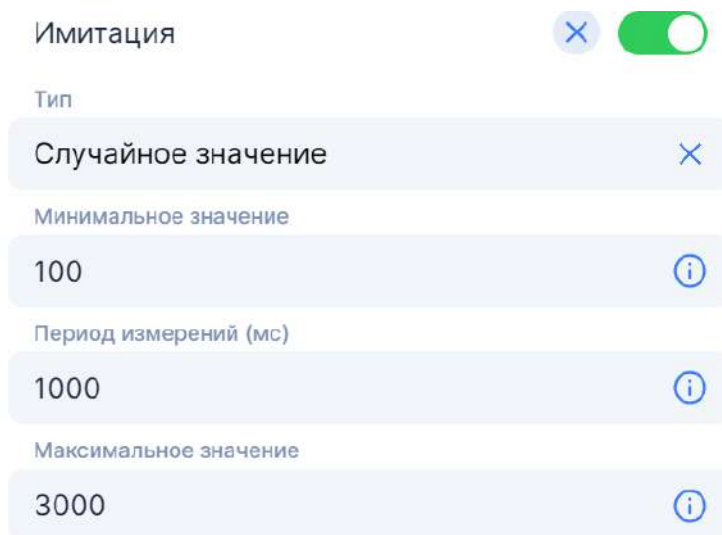
Изменение состояния на каждом шаге

Период измерений (мс)

100000

Рис. 3.40. Набор параметров аналоговых объектов

Для объектов с типом Случайное значение - минимальное и максимальное значение, период измерений (рис. 3.41).



Имитация ☒

Тип

Случайное значение

Минимальное значение

100

Период измерений (мс)

1000

Максимальное значение

3000

Рис. 3. 41. Набор параметров для объекта типа Случайное значение

По завершении настройки необходимо сохранить изменения. После чего конфигурируемый объект будет отмечен графическим признаком имитации (рис. 3.42).



Рис. 3.42. Графический признак имитации

3.3.4. Настройка вычисления расчётных переменных

В Платформе предусмотрена возможность математической обработки переменных, поступающих от различных объектов. Для задания правил (формул) такой обработки необходимо активировать опцию «Вычисление» в редакторе параметров объектов (рис. 3.43).

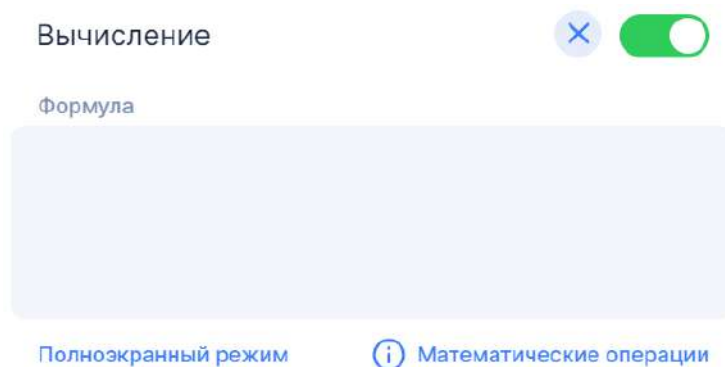


Рис. 3.43. Настройка расчетной переменной

В поле «Формула» необходимо заполнить формулу расчетной переменной. Объекты в формулу можно задать вручную либо воспользоваться кнопкой «Выбрать» напротив объекта в дереве объектов (рис. 3.44).



Рис. 3.44. Задание расчетной формулы

Внутри поля ввода формулы присутствует подсказка в виде допустимых математических операций. Для ее открытия надо привести курсор на «Математические операции» (рис. 3.45).

Математические операции:	
[+]	Сложение
[-]	Вычитание
[*]	Умножение
[/]	Деление
[>]	Больше
[<]	Меньше
[=]	Равно
[^]	Степень
[!]	Не
[>=]	Больше или равно
[<=]	Меньше или равно
[!=]	Не равно
[{ . }]	Интегрирование
[ЕСЛИ]	Оператор ЕСЛИ (<выражение>; <значение если выражение истинно>; <значение если выражение ложно>)
[ЕСЛИМН]	Оператор ЕСЛИМН (<выражение 1>; <значение если выражение 1 истинно>; <значение если выражение 1 ложно>; ... <выражение n>; <значение если выражение n истинно>; <значение если выражение n ложно>)

Рис. 3.45. Доступные математические операции

Для вывода формулы в полноэкранный режим следует нажать на кнопку «Полноэкранный режим» (рис. 3.46).

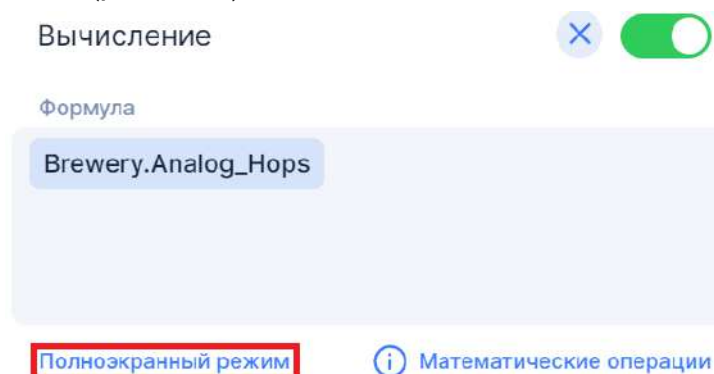


Рис. 3.46. Кнопка полноэкранного режима

После чего поле ввода расширится на всю правую часть экрана (рис. 3.47).

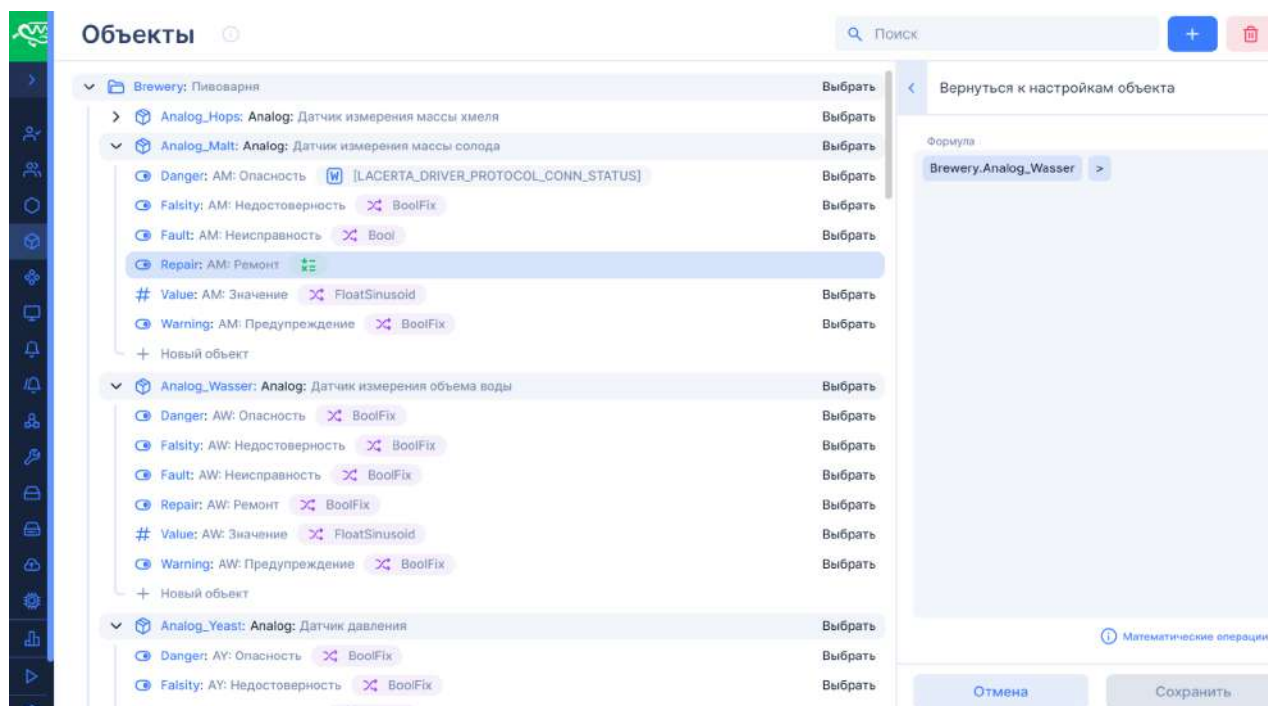


Рис. 3.47. Вывод полноэкранного режима

По завершении настройки сохранить изменения. После чего конфигурируемый объект будет отмечен графическим признаком вычисления (рис. 3.48).

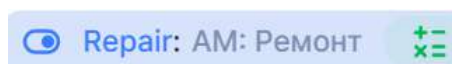


Рис. 3.48. Графический признак вычисляемого объекта

3.3.5. Параметры аналоговых объектов

Параметры аналоговых объектов позволяют настроить сопряжение переменных через доступные драйверы, осуществить имитацию данных, а также настроить расчетные переменные (рис. 3.49).

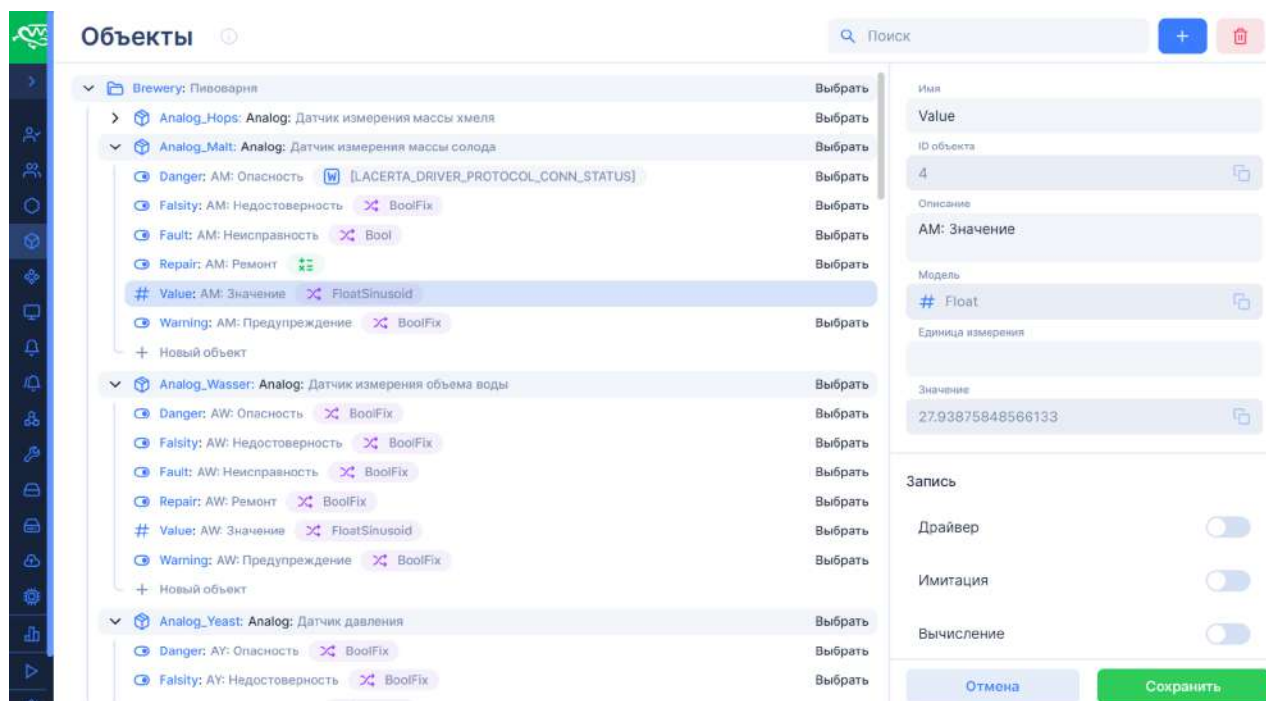


Рис. 3.49. Параметры аналогового объекта

3.3.6. Параметры дискретных объектов

В отличие от параметров аналогового объекта, дискретный объект имеет дополнительные параметры, связанные с возможностью конфигурации событий (рис. 3.50).

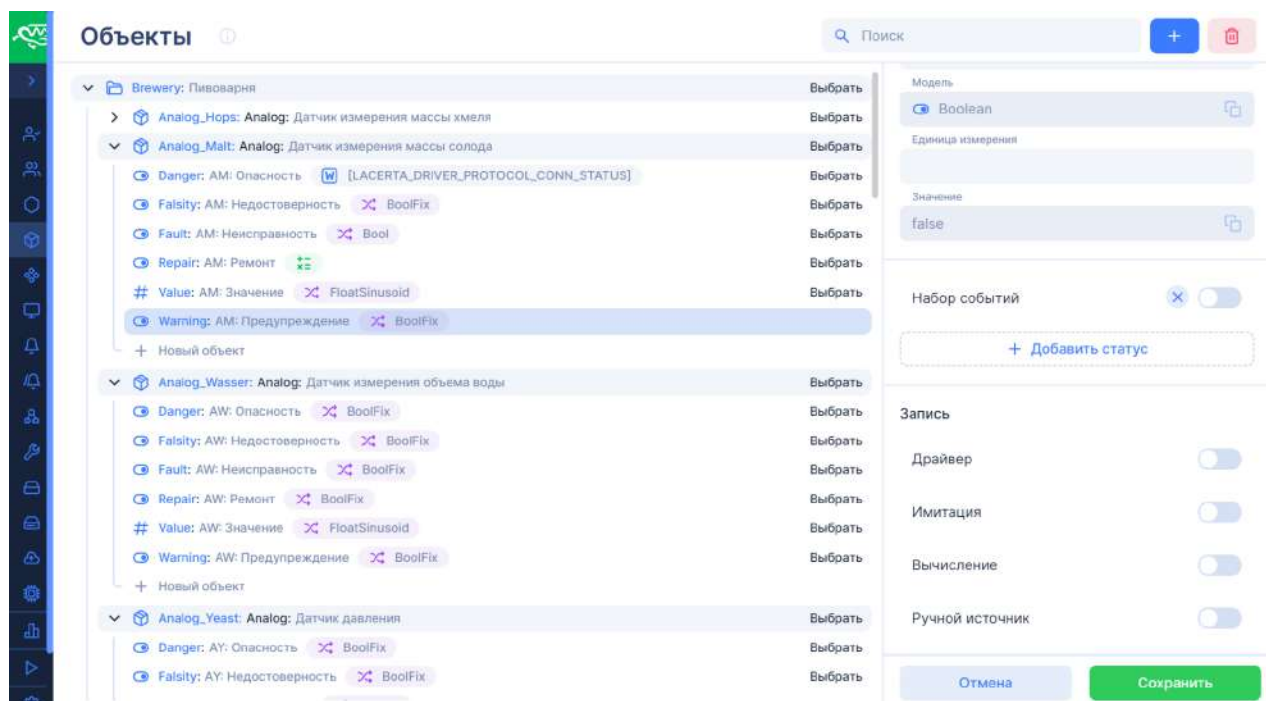


Рис. 3.50. Параметры дискретного объекта

В параметрах дискретных объектов присутствует «Набор событий». Для добавления статуса в «Набор событий» необходимо нажать на кнопку «+ Добавить статус» (рис. 3.51).

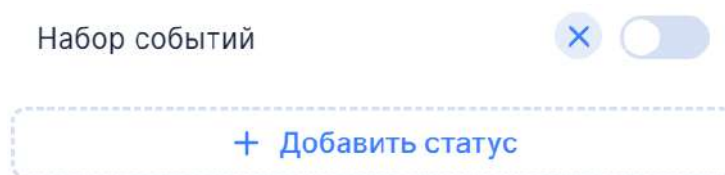


Рис. 3.51. Набор событий

После чего выбрать значение (True/False), класс события и задать его текст (рис. 3.52).

Рис. 3.52. Добавление статуса

Изменение значения дискретной переменной (False/True) может использоваться для вызова события (разд. 2.4). Активировать создание события можно выбрав соответствующее значение в статусе (рис. 3.53).

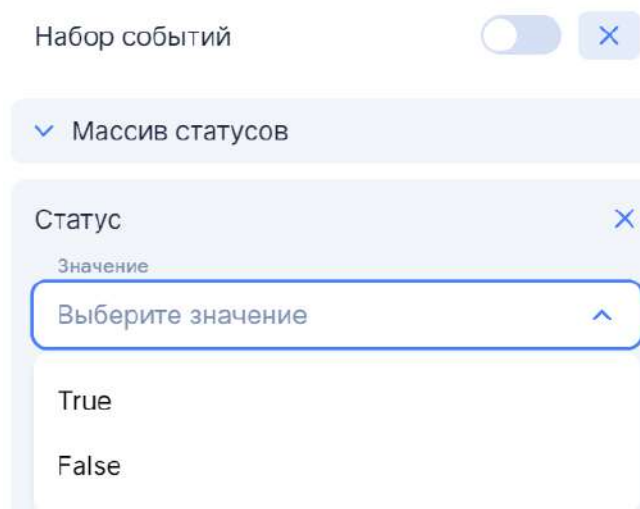


Рис. 3.53. События при достижении False или True

Далее выбрать класс события. Классы событий должны быть определены заранее (рис. 3.54).

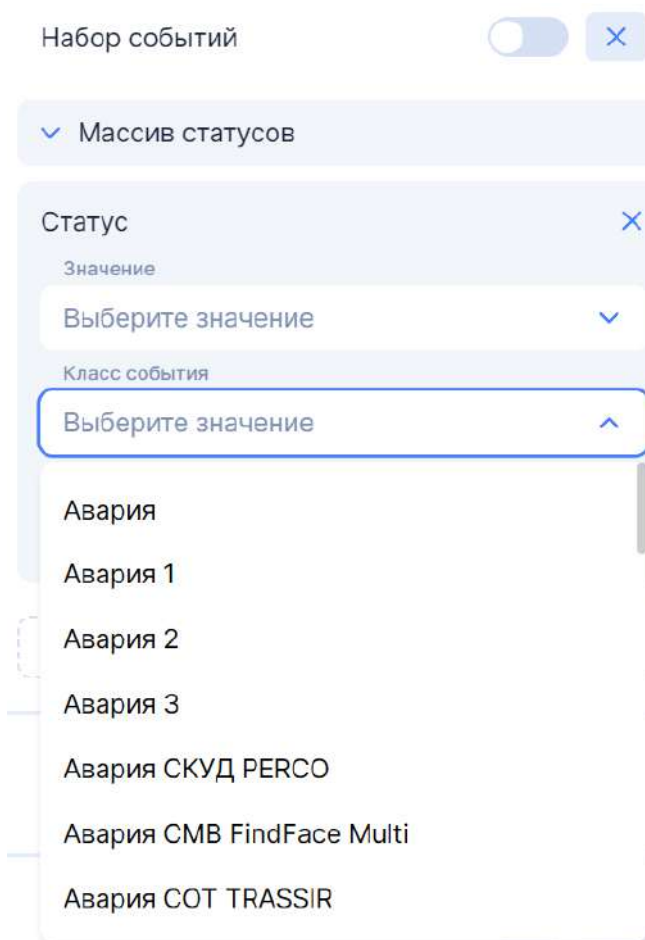


Рис. 3.54. Выбор класса события

По завершении задать текст сообщения, которое будет выводиться при наступлении события (рис. 3.55).

Набор событий

Массив статусов

Статус

Значение

Выберите значение

Класс события

Выберите значение

Текст события

Рис. 3.55. Текст сообщения при наступлении события

После чего сохранить настройки статуса нажатием на соответствующую кнопку (рис. 3.56).

Набор событий

Массив статусов

Статус

Значение

Выберите значение

Класс события

Выберите значение

Текст события

Отмена

Сохранить

Рис. 3.56. Сохранение настроек статуса

Также можно выбрать уже готовый набор событий. Для этого необходимо активировать свитчер в «Наборе событий», после чего нужно выбрать значение (рис. 3.57).

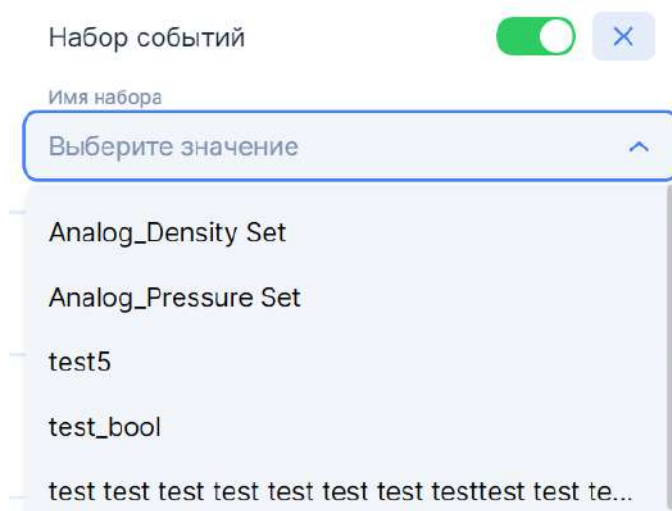


Рис. 3.57. Выбор набора событий

3.3.7. Графические признаки объектов

В зависимости от активированных опций объекты Платформы выделяются с помощью графических изображений (признаков).

В папке объектов «Directory» - объекты распределены по принадлежности к моделям. Напротив модели каждого объекта указан его графический признак (рис. 3.58).

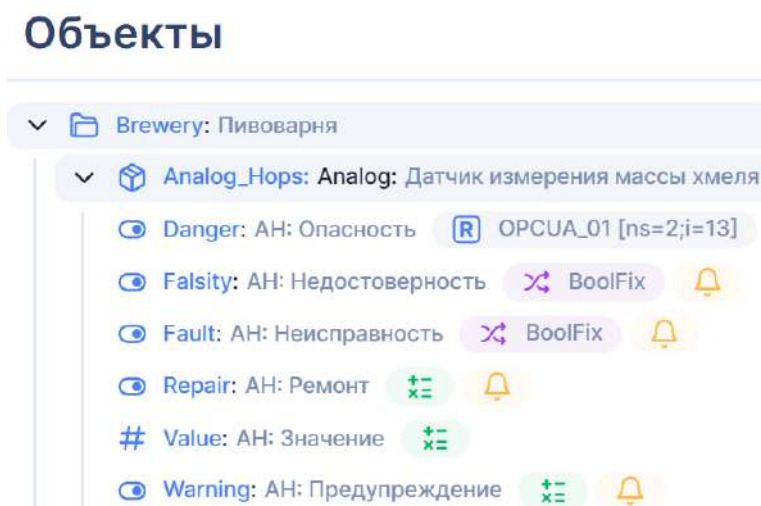


Рис. 3.58. Папка Directory

При активации опции напротив строки объекта появится соответствующий графический признак (рис. 3.59).

Danger: AM: Опасность OPC_CAD	Подключен драйвер (R - чтение, W - запись)
Falsity: AH: Недостоверность BoolFix	Имитация
Value: AH: Значение	Вычисление
Falsity: AH: Недостоверность BoolFix	Несколько опций (имитация+событие)
Falsity: AH: Недостоверность	Набор событий/добавление статуса
Danger: BK → Датчик плотности: Опасность	Установление источника вручную (Ручной источник)

Рис. 3.59. Графические признаки

3.3.8. Создание объекта

Для создания корневого объекта необходимо нажать на кнопку «+» (рис. 3.60).



Рис. 3.60. Создание нового корневого объекта

Далее задать наименование объекта и его описание (рис. 3.61).

Имя

Demo_01

Описание

Demo

Модель

Boolean

Рис. 3.61. Наименование и описание объекта

Наименование объекта должно начинаться с буквы латинского алфавита и может содержать буквы, цифры и знак «_» (рис. 3.62).

Рис. 3.62. Предупреждение наименования объекта, раскладка

Если наименование объекта уже используется в Платформе, то пользователю также будет выдано предупреждение (рис. 3.63).

Рис. 3.63. Предупреждение наименования объекта, название занято

Затем выбрать модель, на основании которой будет создан объект (рис. 3.64).

Рис. 3.64. Выбор модели

По завершении нажать на кнопку «Создать» (рис. 3.65).

Имя
Demo_01

Описание
Demo

Модель
Boolean

Отмена Создать

Рис. 3.65. Создание объекта

В случае, если все поля заполнены корректно, будет создан новый объект (рис. 3.66).

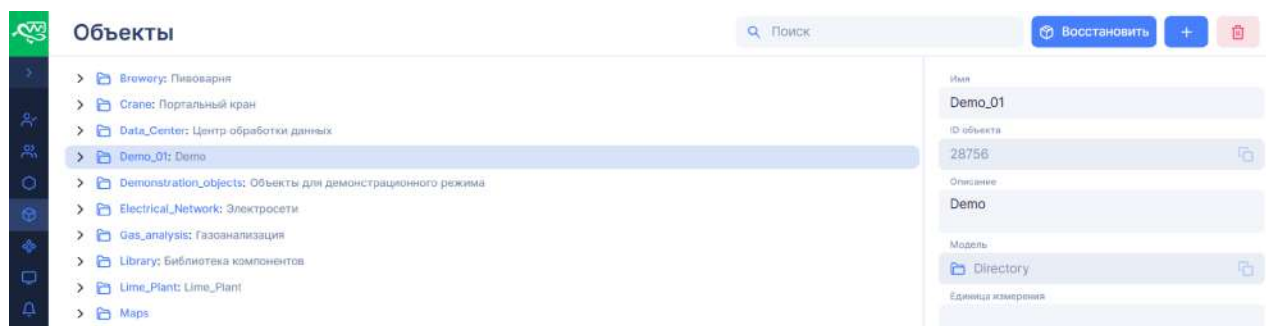


Рис. 3.66. Создание корневого объекта

Для добавления вложенного объекта в корневой объект необходимо развернуть корневой объект и нажать «+ Новый объект». После чего задать параметры объекта и выбрать модель на основании которой будет создан объект. По завершении нажать «Создать» (рис. 3.67).



Рис. 3. 67. Создание вложенного объекта

3.3.9. Удаление объекта

Для удаления объекта со всеми вложенными объектами, необходимо выбрать объект и нажать на соответствующую кнопку удаления в форме корзины. После чего возникнет всплывающее окно с подтверждением удаления объекта и предупреждением о последующем удалении вложенных в него объектов. При подтверждении выбранный объект будет удален со всеми вложенными объектами (рис. 3.68).

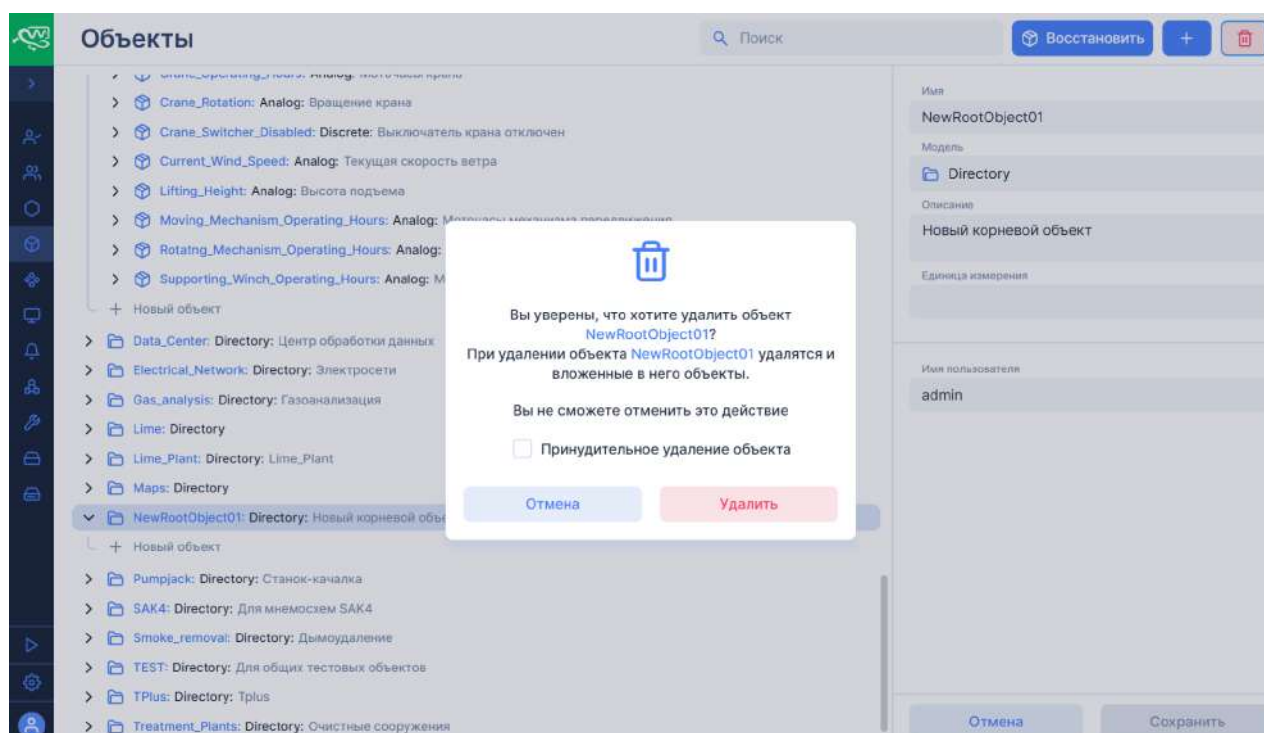


Рис. 3.68. Подтверждение удаление объекта

Если объект используется в качестве сигнала на мнемосхеме, добавлен в выборку, либо задействован в формуле вычисления другого объекта, то удалить его будет невозможно. Для удаления такого объекта используется чекбокс «Принудительное удаление объекта» (рис. 3.68).

3.4. Компоненты

Компонент — это графическое изображение модели или объекта, построенное по иерархическому принципу. Технически это изображение объекта, размещенное на нескольких слоях, один из которых является базовым и, как правило, служит для отображения нормального (штатного) состояния объекта. Количество слоёв и, следовательно, количество экземпляров и видов базового компонента определяется числом возможных состояний объекта (значения переменных). На каждом слое могут задаваться различные характеристики базового изображения (цвет, форма, размер и т. д.). Видимость слоев и, следовательно, отображение компонента определяются при помощи скриптов, входными данными для которых являются значения переменных объекта (разд. 3.5.6).

Один и тот же компонент может быть связан с несколькими разными моделями — так же как и одна и та же модель может быть отображена в Платформе разными компонентами. Платформа поставляется с библиотекой готовых графических компонентов, при этом пользователь может создавать свои компоненты или изменять существующие в соответствии со спецификой проекта (рис. 3.69).

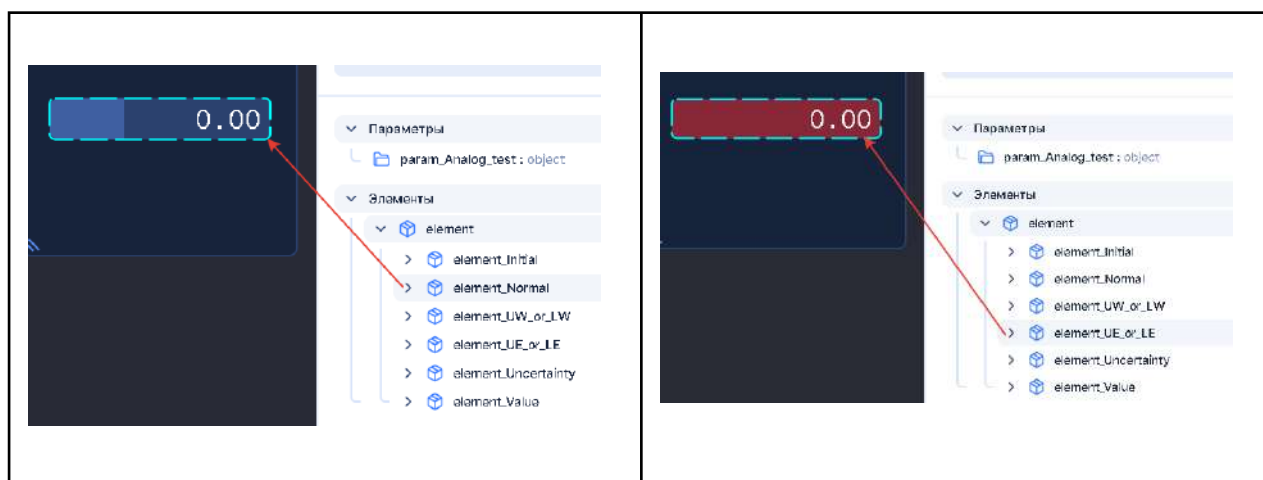


Рис. 3.69. Пример компонента «Аналоговый датчик»

3.5. Состав мнемосхемы

3.5.1. Создание компонента

Для создания компонентов используется внешний редактор, поддерживающий формат SVG или имеющий возможность экспорта файлов в этом формате (например, онлайн-редактор Figma). Система поддерживает импорт графических элементов в формате SVG. Количество и сложность необходимых компонентов зависит от специфики проекта (рис. 3.70).

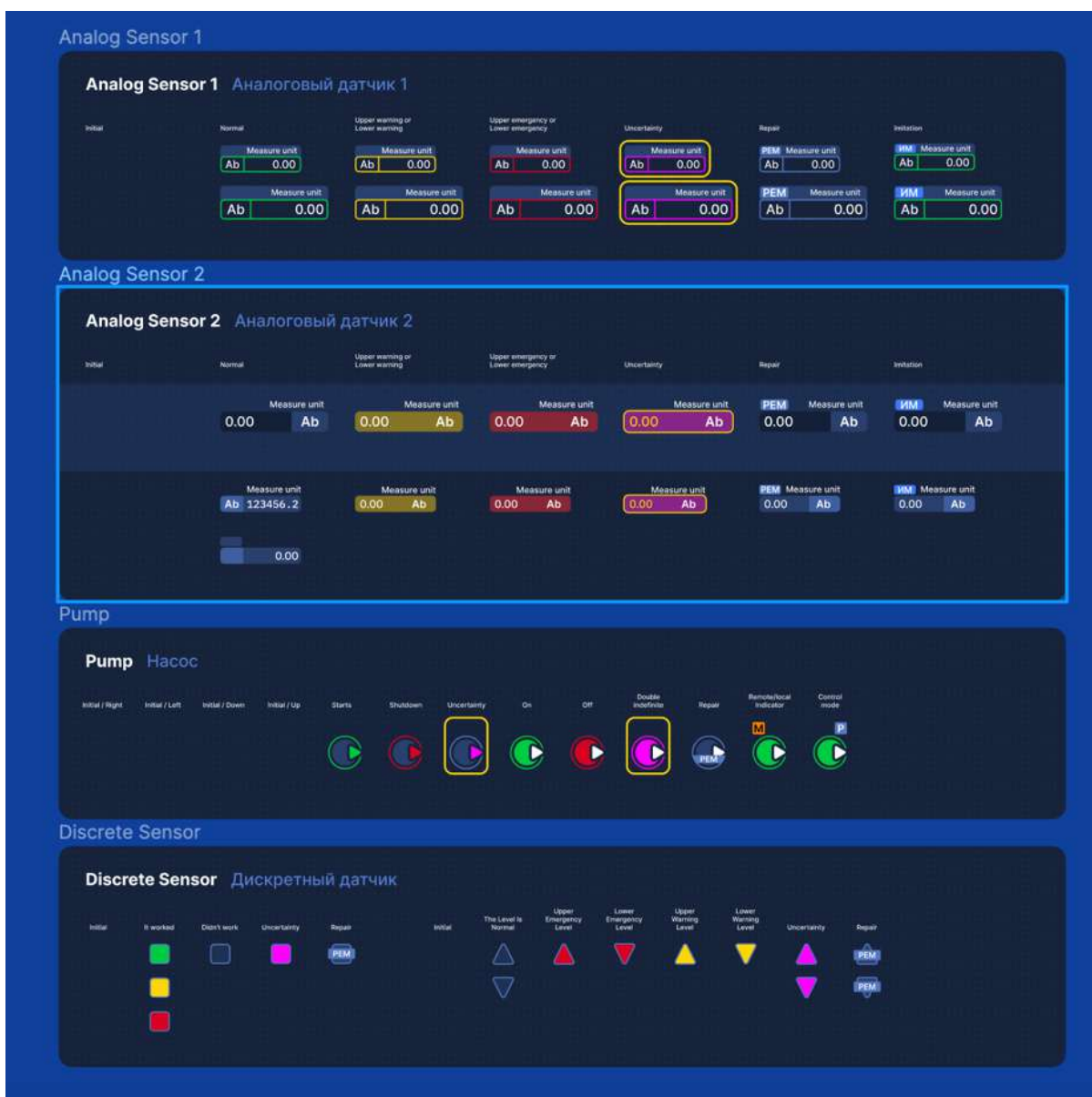


Рис. 3.70. Компоненты Платформы

Для создания компонентов мнемосхемы в онлайн-редакторе Figma пользователь должен:

1. Создать Figma Team (команду), в рамках которой будет вестись вся работа над мнемосхемами;
2. Внутри команды Figma Team создать отдельный файл:
 - а) Библиотека компонентов — файл, содержащий библиотеку компонентов. Назначение: содержит все базовые элементы (насосы, датчики, клапаны и т.д.), из которых будут собираться мнемосхемы.

После этих действий пользователь сможет создавать графические элементы, соответствующие типовым объектам на мнемосхемах (насосы, клапаны, датчики, резервуары и т.п.).

Пример создания компонента Насос:

1) Создайте новый Frame (F) и задайте ему размер, соответствующий виду датчика. Назовите фрейм в слоях, например, “Pump / Right” (рис. 3.71).

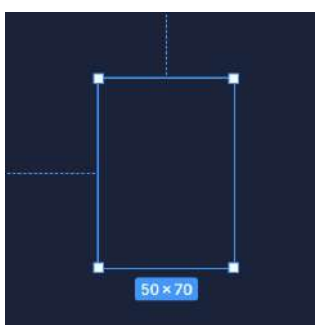


Рис. 3.71. Создание фрейма

2) Внутри фрейма схематично отрисуйте вид Насоса в основном его состоянии (рис. 3.72).



Рис. 3.72. Отрисовка насоса

3) Объедините в общую группу все графические элементы из которых состоит Насос и назовите её «Initial» (переводится, как «Исходная», рис. 3.73).



Рис. 3.73. Группа графических элементов

4) Создайте все остальные необходимые состояния для Насоса. Для каждого состояния создайте отдельную группу внутри главного фрейма компонента.

Примеры состояний: Включено (On), Выключено (Off), Ремонт (Repair) и так далее. Названия для групп внутри Figma необходимо задавать латиницей (ограничения как в javascript, рис. 3.74).

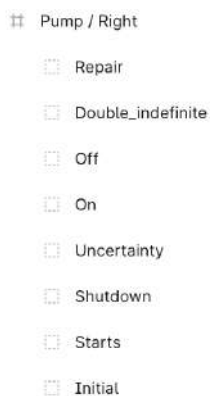


Рис. 3.74. Состояния насоса

5) Настройка видимости. Оставьте видимым только состояние Initial. Все остальные состояния (Repair, Off и т.д.) скройте.

Это необходимо сделать через панель свойств в правой части интерфейса Figma, задав каждой группе настройку Opacity равной 0,01% (рис. 3.75).

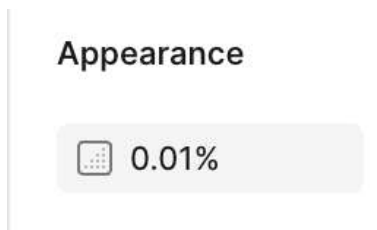


Рис. 3.75. Настройка видимости

6) Объедините созданный Насос в компонент. Выделите главный фрейм (со всеми вложенными состояниями) и нажмите Ctrl/Cmd + Alt + K или ПКМ → Create Component (рис. 3.76).



Рис. 3.76. Объединение насоса в компонент

По аналогии отрисовываются и создаются другие типовые элементы и из них также создаются компоненты (рис. 3.77);

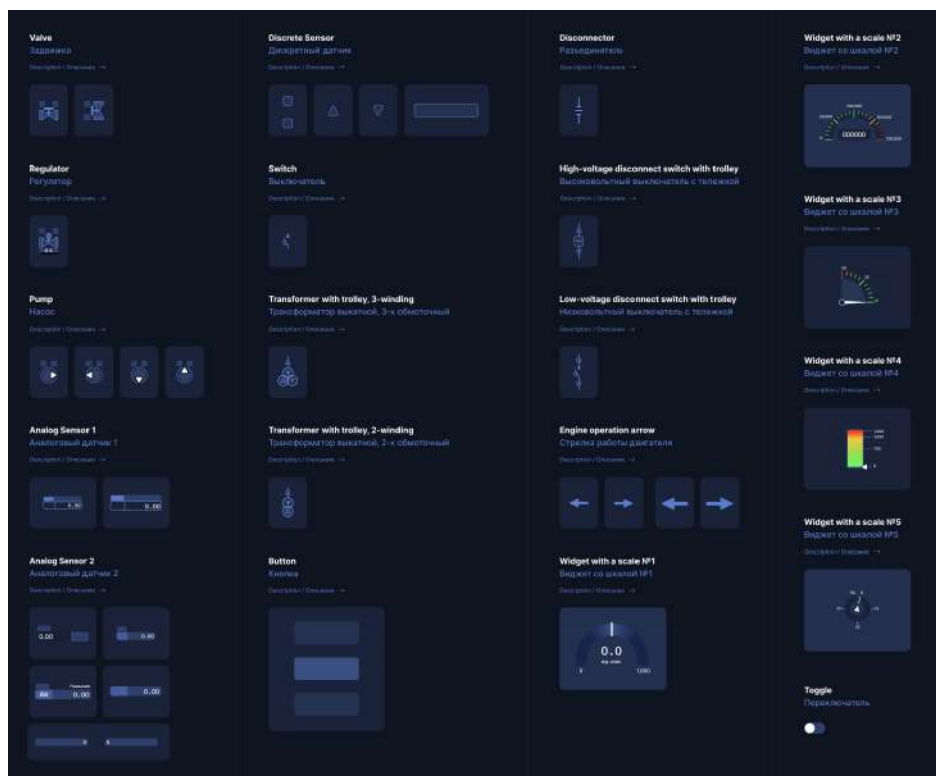


Рис. 3.77. Компоненты мнемосхемы

3. Подготовка библиотеки. После того как были созданы все компоненты, необходимо проверить, чтобы все они имеют корректные имена, структуру и состояния.

Далее опубликовать файл с компонентами как библиотеку: File → Assets → Library → Publish. В будущем пользователь также сможет добавлять новые компоненты и публиковать библиотеку с ними.

3.5.2. Создание мнемосхемы

Процесс создания мнемосхемы заключается в комбинации различных компонентов с элементами графического дизайна и размещении их на экране. В целом, процесс создания мнемосхемы аналогичен созданию компонентов: в нем также задействован внешний редактор (рис. 3.78).

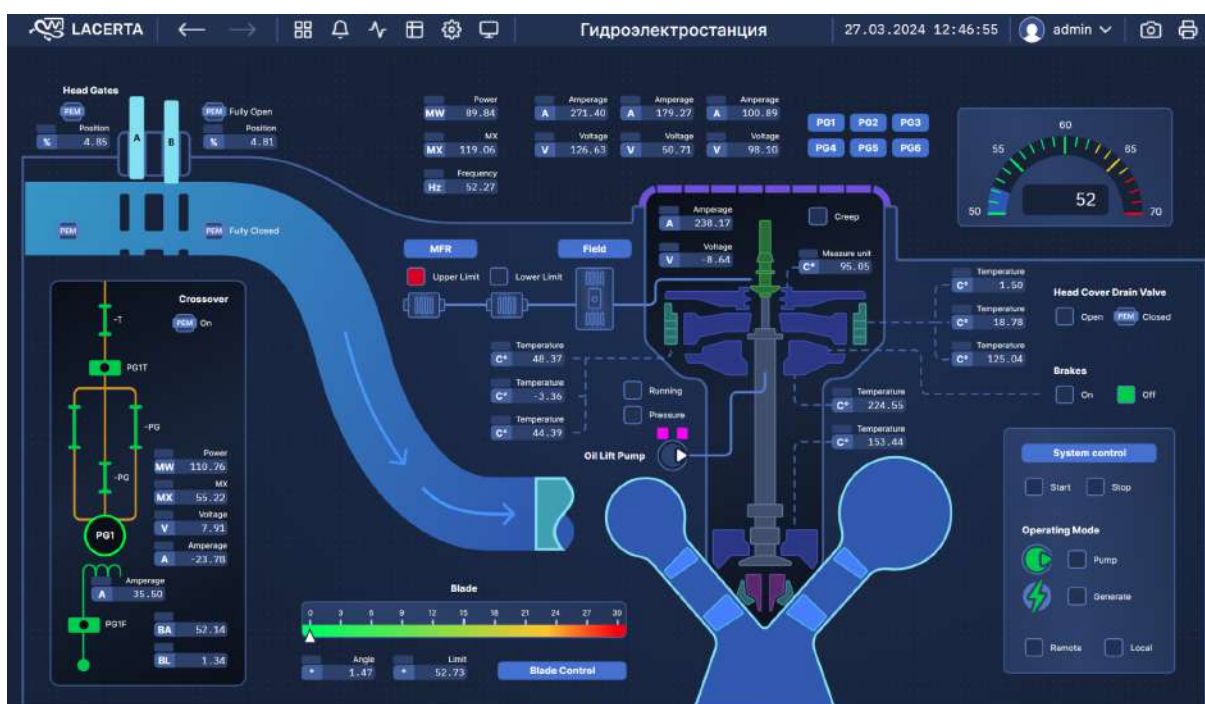


Рис. 3.78. Создание Мнемосхемы

Для создания мнемосхемы в онлайн-редакторе Figma пользователь должен:

1. Создать Figma Team (команду), в рамках которой будет вестись вся работа над мнемосхемами;
2. Внутри команды Figma Team создать отдельный файл:
 - а) Файл мнемосхем — файл, в котором будут собираться конечные экраны мнемосхем с использованием компонентов из библиотеки;

3. Подключение библиотеки. В файле "Мнемосхемы" подключить библиотеку компонентов: Figma → Libraries → Add to file (рис. 3.79);

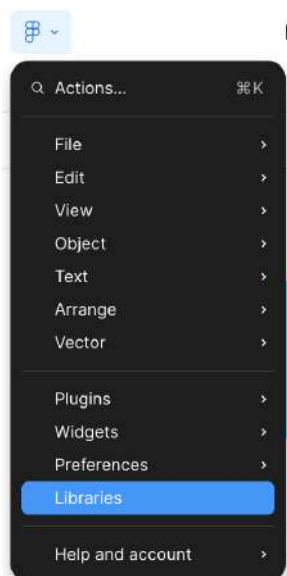


Рис. 3.79. Подключение библиотеки компонентов

4. Сборка экранов. Создать отдельные фреймы (экраны) для каждой мнемосхемы (например: Цех №1, Водоснабжение, Газоснабжение и т.п.);
5. Разместить компоненты из библиотеки на экранах, выстраивая логическую и визуальную структуру схемы.

Пример: мнемосхема «Пивоварня», созданная из таких компонентов как: насос, аналоговые сенсоры, дискретные сенсоры, задвижки, различные резервуары и так далее (рис. 3.80).

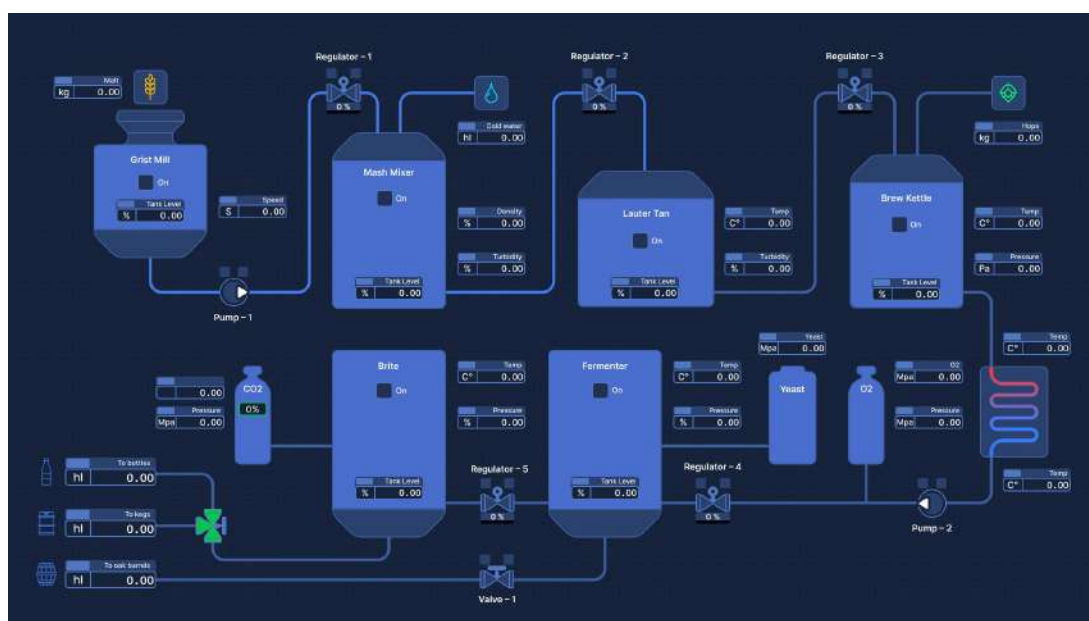


Рис. 3.80. Мнемосхема «Пивоварня»

Для того чтобы найти нужный компонент из библиотеки, можно воспользоваться поиском во вкладке Assets в левой части интерфейса Figma (рис. 3.81).

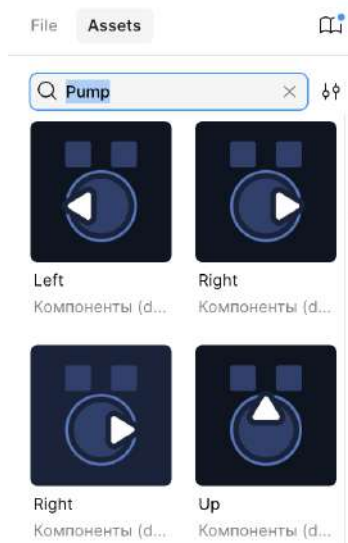


Рис. 3.81. Поиск компонентов

3.5.3. Импорт

Импорт компонентов и мнемосхем в Платформу производится непосредственно из внешнего редактора. Система способна импортировать компоненты и мнемосхемы в формате SVG.

Для того чтобы импортировать данные, необходимо в разделе Настройки перейти в «Figma» и указать ProjectID и Token (рис. 3.82), после чего сохранить изменения.

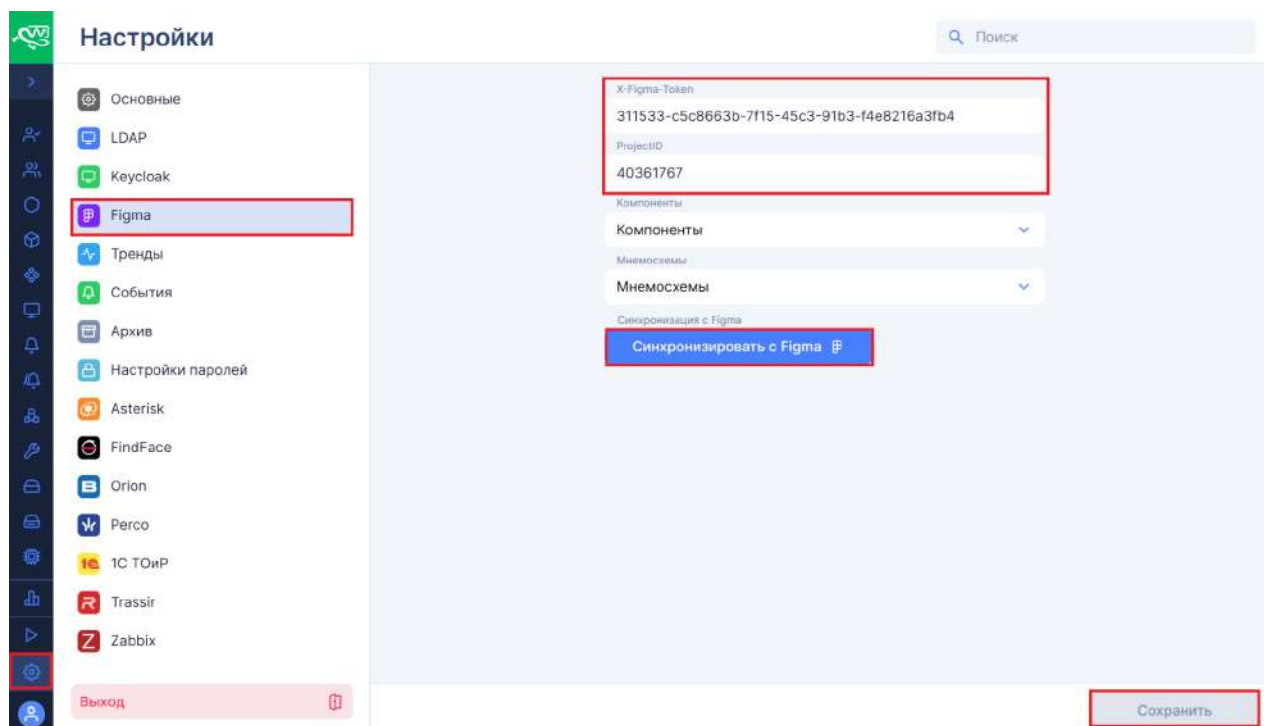


Рис. 3.82. Интеграция с Figma

Для синхронизации данных с редактором необходимо нажать «Синхронизировать с Figma» (рис. 3.82).

3.5.4. Синхронизация

Синхронизация мнемосхем и компонентов между проектами в графическом внешнем редакторе и Системой также осуществляется через специальный подраздел настроек.

Для того чтобы добавить компоненты в Систему, необходимо в подразделе «Figma» развернуть параметр «Компоненты», где выбрать нужное значение в выпадающем списке. По завершении нажать кнопку «Сохранить» (рис. 3.83).

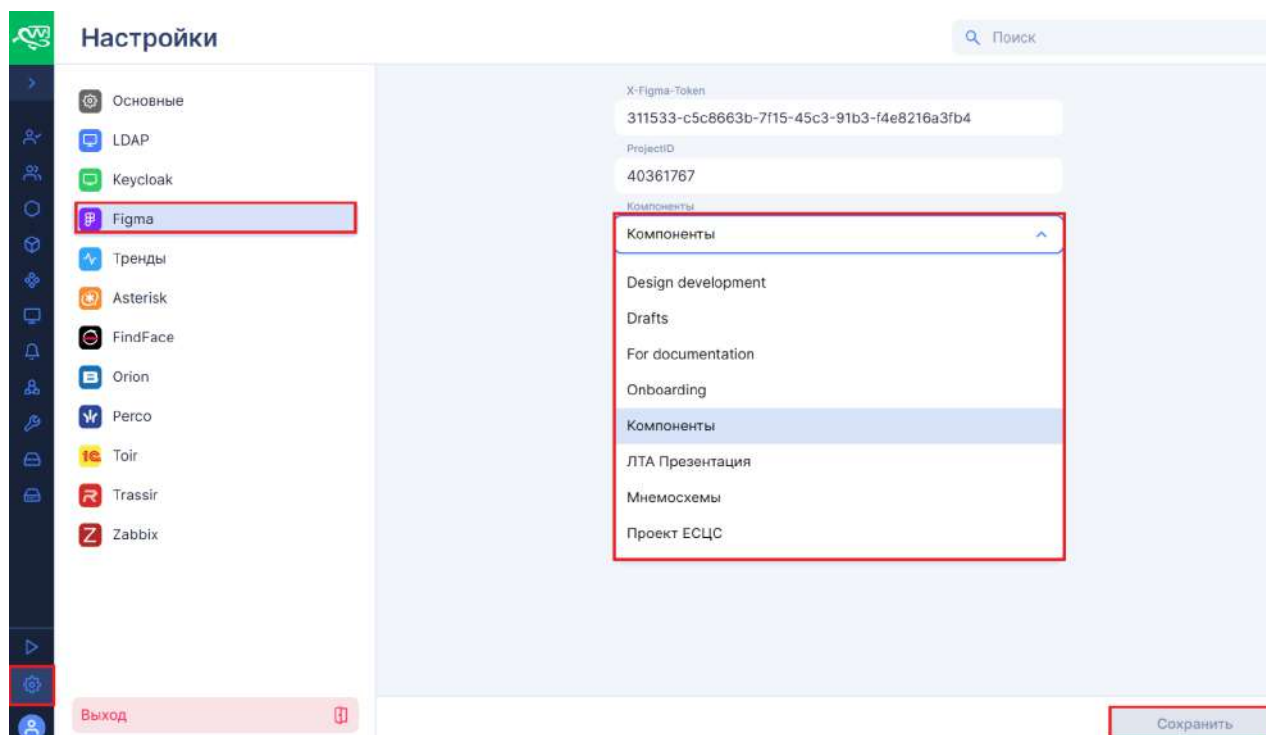


Рис. 3.83. Добавление компонентов

Сохраненные компоненты будут отображаться в соответствующем разделе Среды Администрирования «Компоненты» (рис. 3.84).

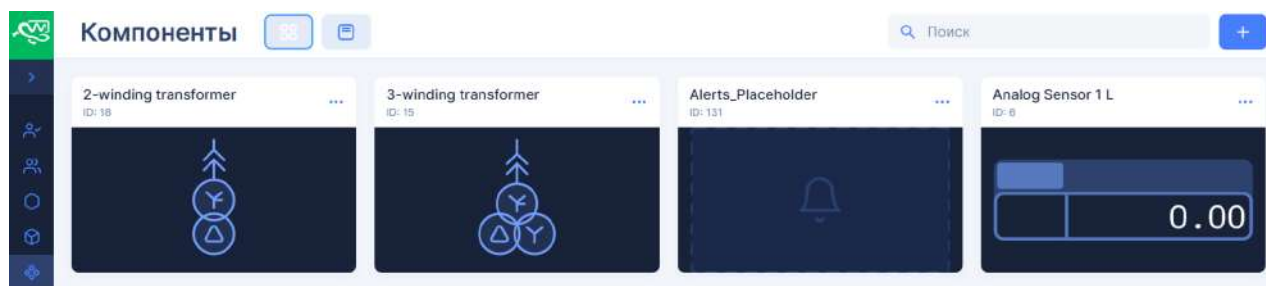


Рис. 3.84. Раздел «Компоненты»

Для того чтобы добавить мнемосхему в Систему, необходимо подразделе «Figma» развернуть параметр «Мнемосхемы», где выбрать нужное значение в выпадающем списке. По завершении нажать «Сохранить» (рис. 3.85).

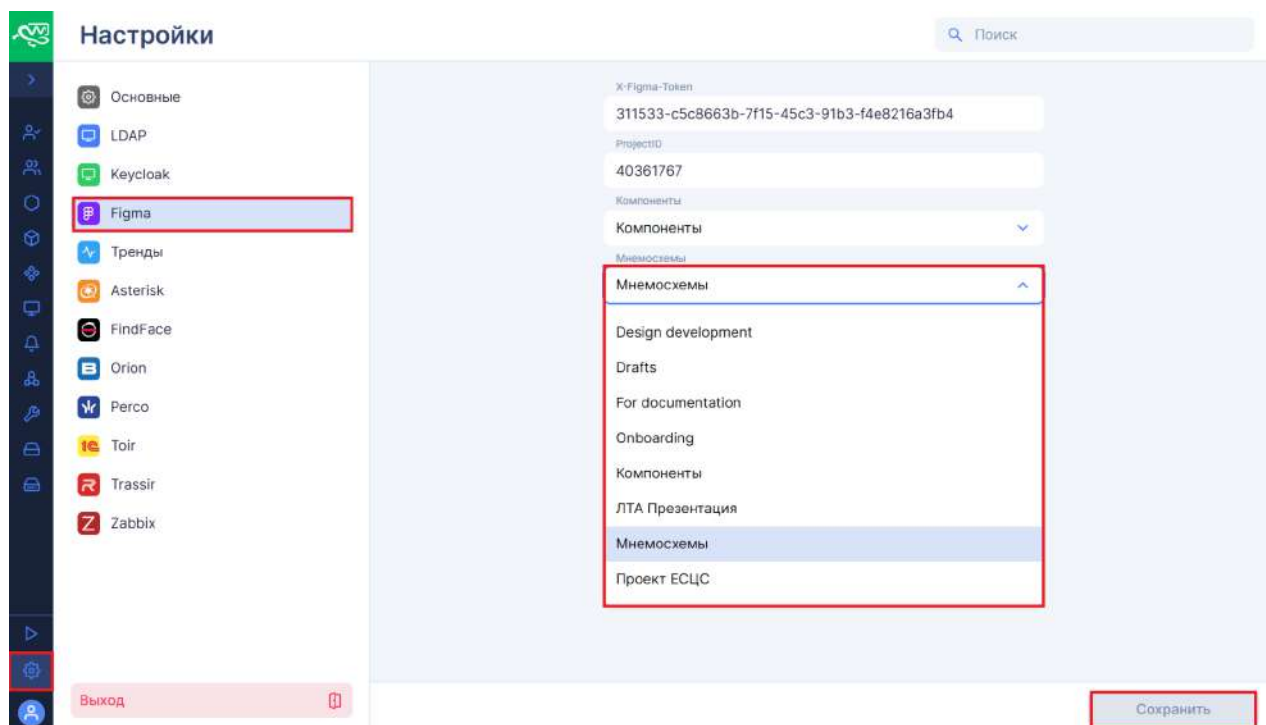


Рис. 3.85. Добавление мнемосхемы

Сохраненные мнемосхемы будут отображаться в соответствующем разделе Среды Администрирования «Мнемосхемы» (рис. 3.86).

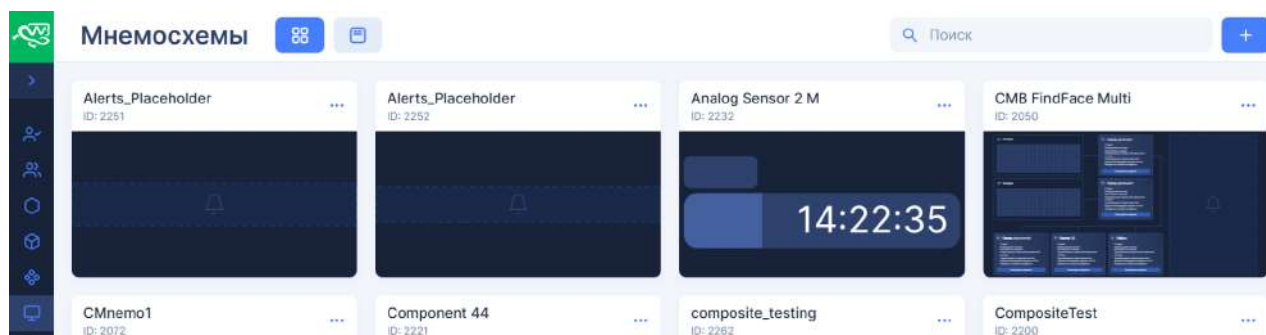


Рис. 3.86. Раздел «Мнемосхемы»

3.5.5. Редактор привязок мнемосхем к данным

Привязка сигналов к компонентам мнемосхемы осуществляется в специальном редакторе в разделе «Мнемосхемы» Среды Администрирования (рис. 3.87).

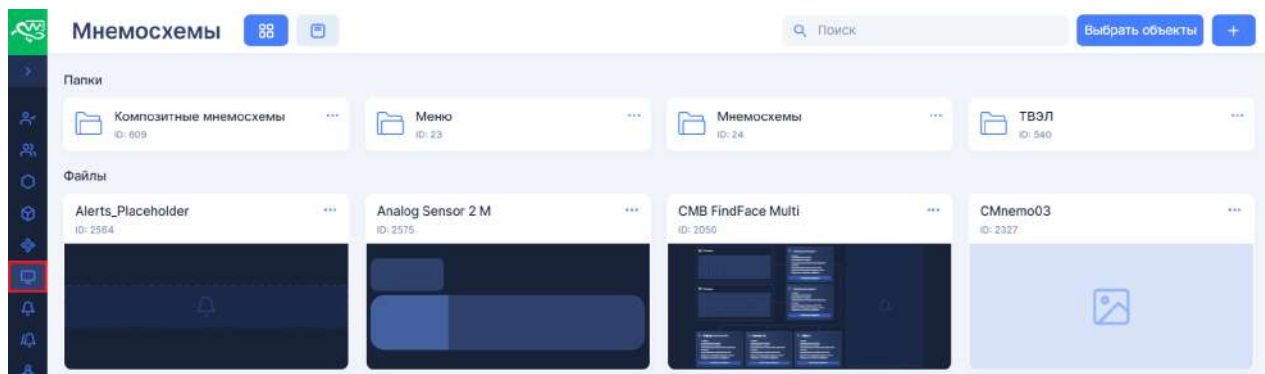


Рис. 3.87. Выбор мнемосхемы

Отображение мнемосхем также доступно и в виде списка. Для этого требуется нажать на соответствующую кнопку (рис. 3.88).

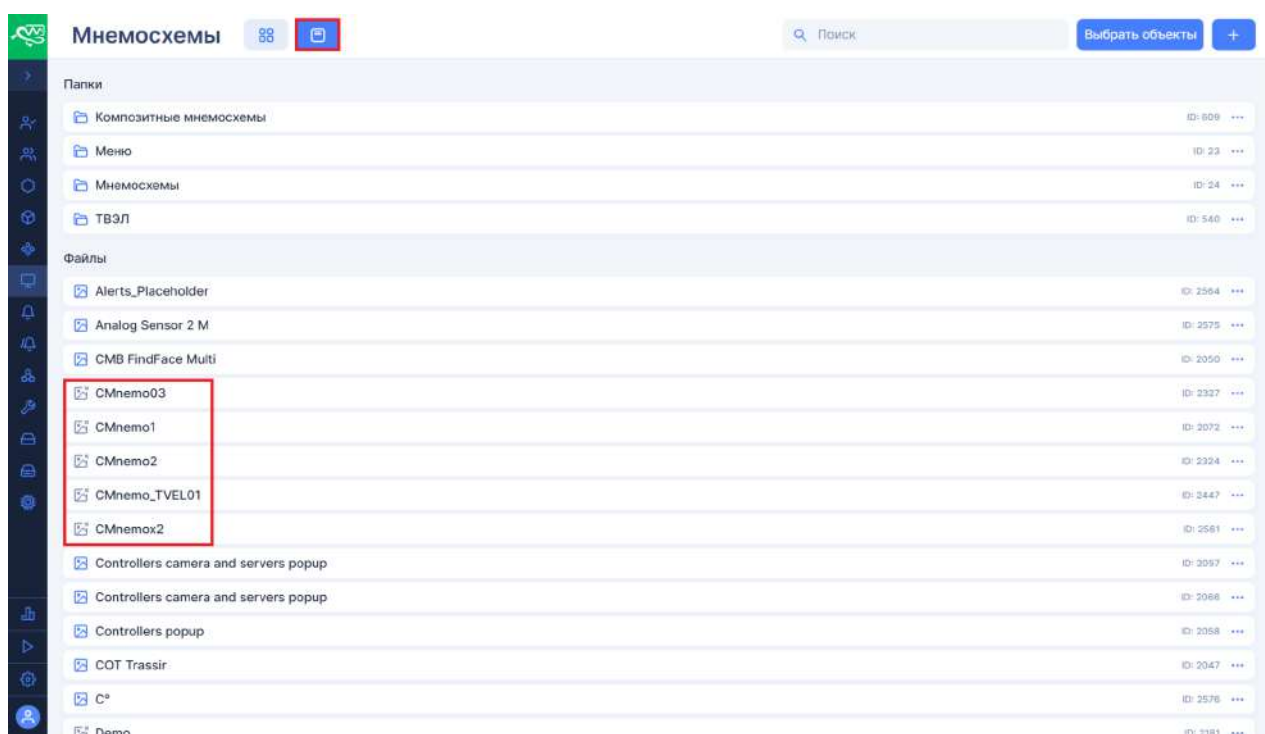


Рис. 3.88. Отображение мнемосхем в виде списка

Композитные мнемосхемы в данном списке будут отображаться пиктограммами серого цвета (рис. 3.88).

Пользователь имеет возможность открыть, выбрать, переименовать, переместить или удалить мнемосхему нажатием на кнопку «...» (рис. 3.89).

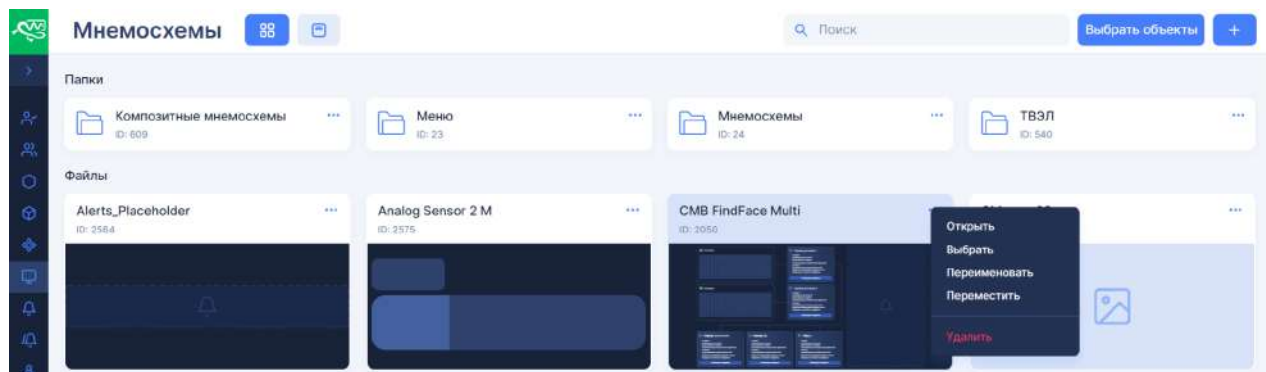


Рис. 3.89. Переименование или удаление мнемосхемы

Также можно создавать папки и группировать в них мнемосхемы или другие папки. Для того чтобы создать папку, необходимо нажать кнопку «+» и выбрать «Создать папку» (рис. 3.90).



Рис. 3.90. Создание папки

Функционал действий с папкой аналогичен мнемосхеме и открывается нажатием «...» (рис. 3.91).

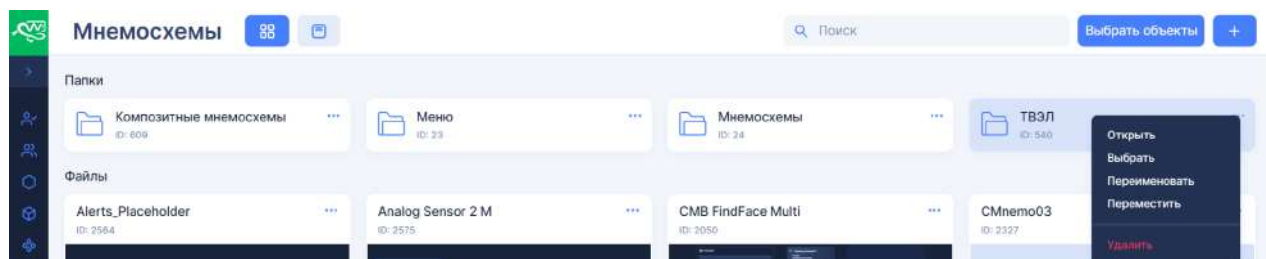


Рис. 3.91. Действия с папкой

Для выбора сразу нескольких папок/мнемосхем необходимо воспользоваться кнопкой «Выбрать объекты», позволяющей переместить или удалить сразу несколько объектов (рис. 3.92).

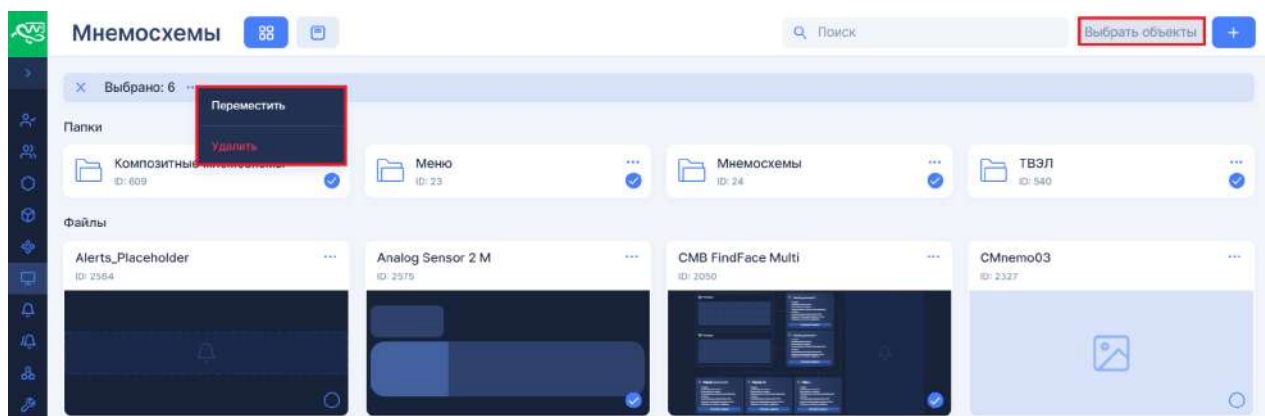


Рис. 3.92. Кнопка выбора объектов

Либо воспользоваться кнопкой «...», что также позволит выбрать несколько объектов с последующим перемещением или удалением (рис. 3.93).

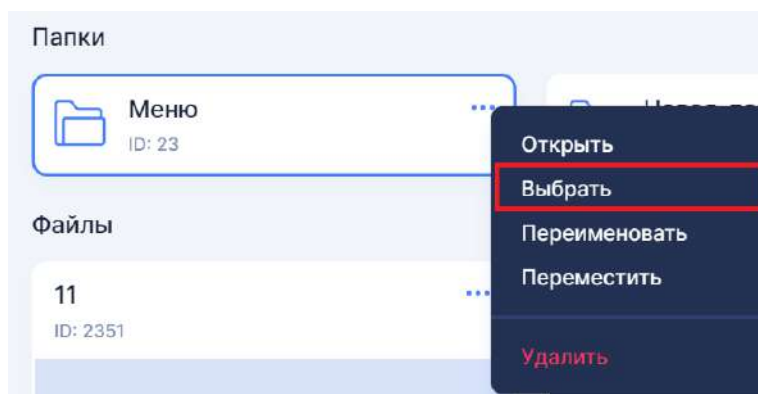


Рис.3.93. Выбор нескольких объектов

После нажатия на мнемосхему откроется ее редактор (рис. 3.94).

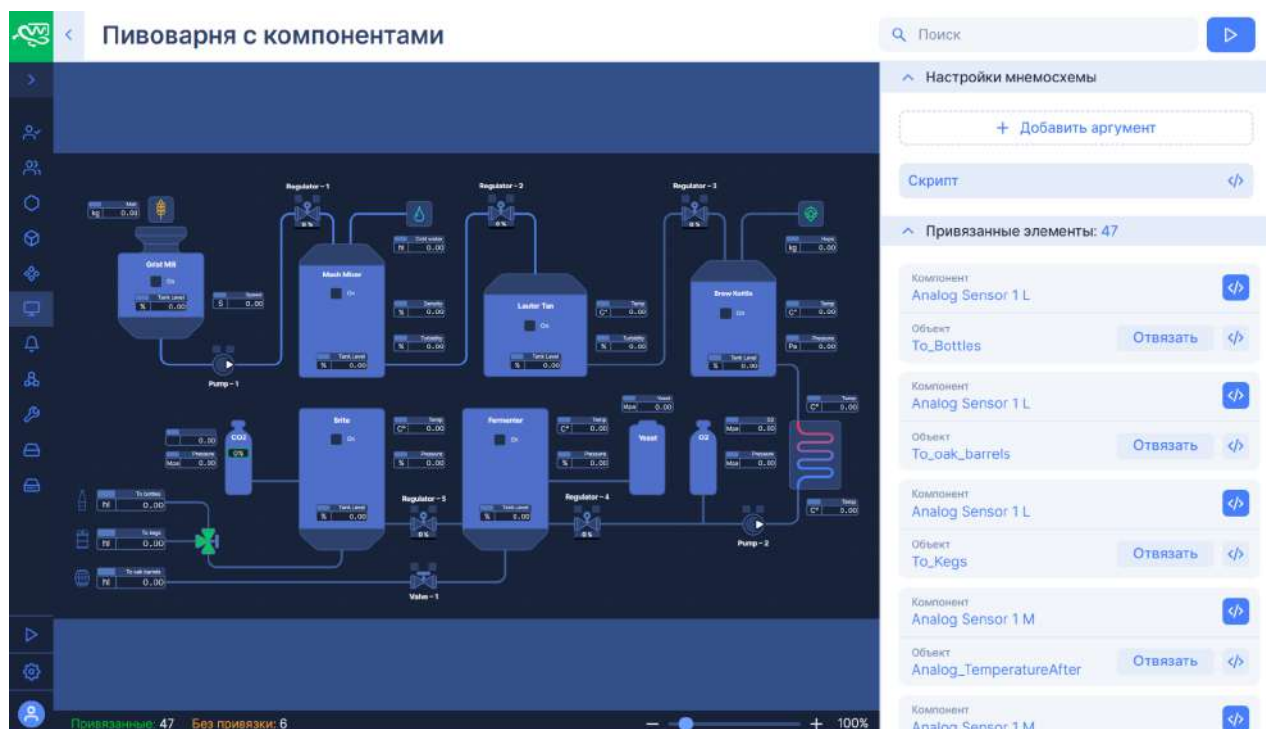


Рис. 3.94. Редактор привязки мнемосхемы

Для отображения элементов, к которым привязаны сигналы (переменные), необходимо воспользоваться кнопкой «Привязанные», где отображается и их количество (рис. 3.95).



Рис. 3.95. Привязанные элементы

Аналогично функционирует и отображение элементов без привязки. Для их отображения необходимо воспользоваться кнопкой «Без привязки», где также будет отображаться и их количество (рис. 3.96).



Рис. 3.96. Элементы без привязки

Можно также установить требуемый масштаб изображения мнемосхемы (от 90% до 200%). Для этого следует выбрать нужное значение масштаба на панели (рис. 3.97).



Рис. 3.97. Масштабирование

Для привязки сигнала к элементу мнемосхемы нужно выбрать элемент без привязки на самой мнемосхеме либо воспользоваться редактором и выбрать необходимый элемент из списка непривязанных (рис. 3.98).



Рис. 3.98. Привязка сигнала к элементу

Для удобства можно воспользоваться поиском по элементам в правом верхнем углу страницы (рис. 3.99).



Рис. 3.99. Поиск

Для доступа к редактору сценариев управления элементами мнемосхемы следует воспользоваться кнопкой «Скрипт» (рис. 3.100).

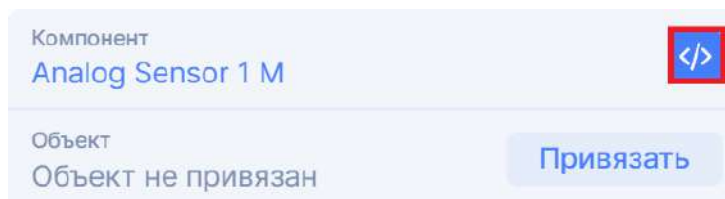


Рис. 3.100. Кнопка «Скрипт»

Для того чтобы привязать элемент к модели на мнемосхеме, необходимо добавить модель к мнемосхеме с помощью нажатия кнопки «+ Добавить аргумент» (рис. 3.101).



Рис. 3.101. Добавление аргумента

После чего выбрать модель и нажать «Добавить модель» (рис. 3.102).

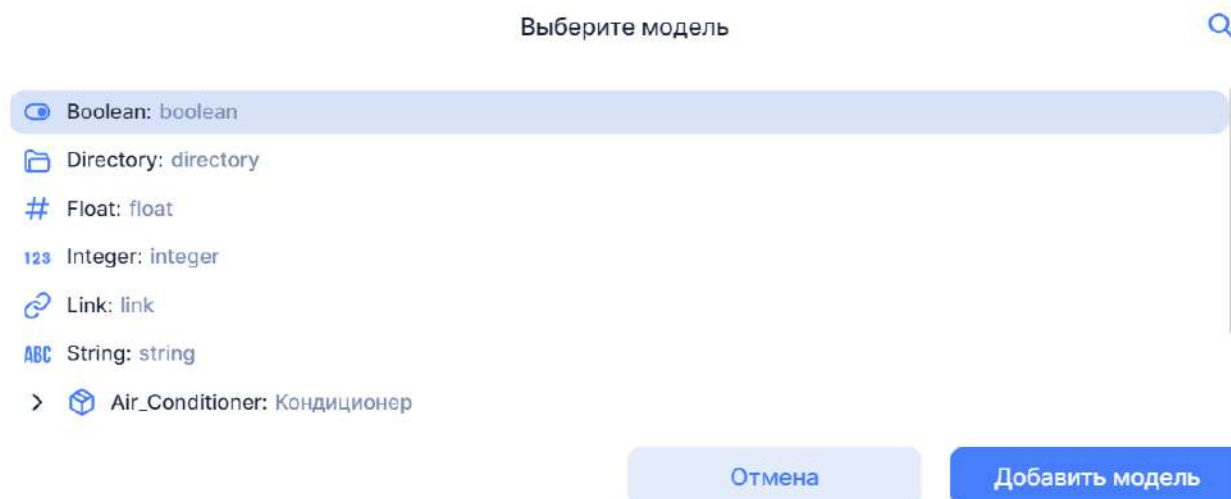


Рис. 3.102. Выбор и добавление модели

Для привязки к элементу выбранная модель предварительно должна быть привязана к компоненту на мнемосхеме. Чтобы привязать модель к компоненту, необходимо нажать на компонент без привязки в боковом редакторе (рис. 3.103).



Рис. 3. 103. Привязка модели к компоненту

После чего откроется редактор компонента в котором нужно добавить модель планируемую к привязке. Для этого следует нажать «+ Добавить модель» и затем выбрать соответствующую модель в всплывающем окне (рис. 3.104).

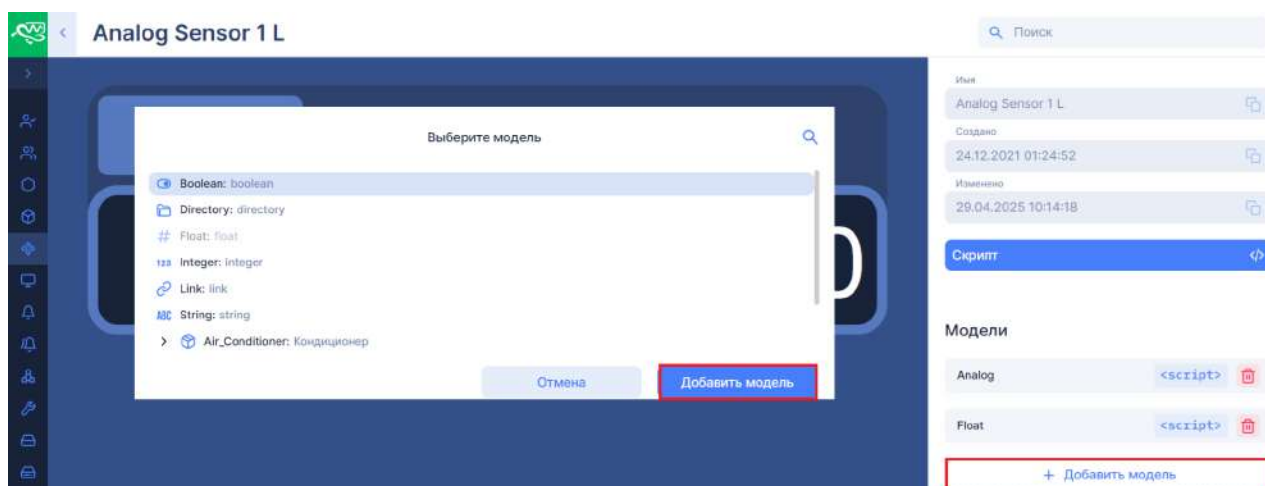


Рис. 3.104. Привязка модели к компоненту

Теперь когда модель привязана к компоненту, можно привязать модель к элементу на мнемосхеме. Для этого в редакторе необходимо нажать кнопку «Привязать», после чего в всплывающем окне выбрать модель (рис. 3.105).

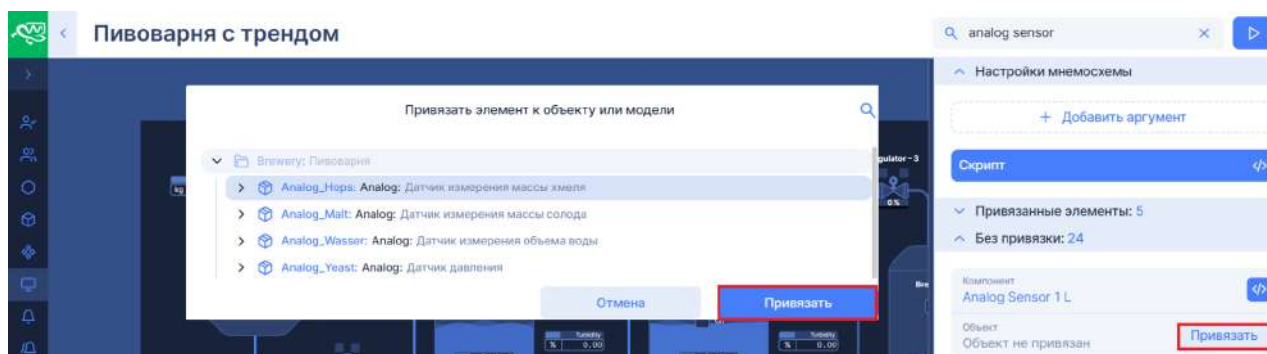


Рис. 3.105. Привязка модели к элементу

Для того чтобы удалить связь («Отвязать» элемент), нужно выбрать его на мнемосхеме и нажать кнопку «Отвязать» (рис. 3.106).

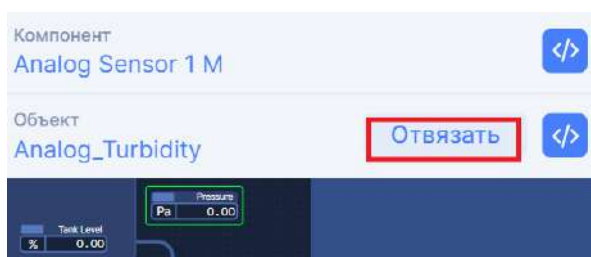


Рис. 3.106. Удаление связи с элементом

Либо воспользоваться боковым редактором, развернув «Привязанные элементы» (рис. 3.107).

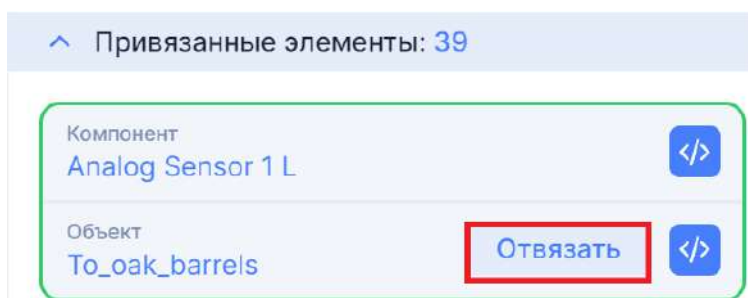


Рис. 3.107. Отвязка элементов

При наведении курсора на имя объекта отобразится его полный путь, а при нажатии произойдет переход в объект в редакторе объектов Среды Администрирования (рис. 3.108).

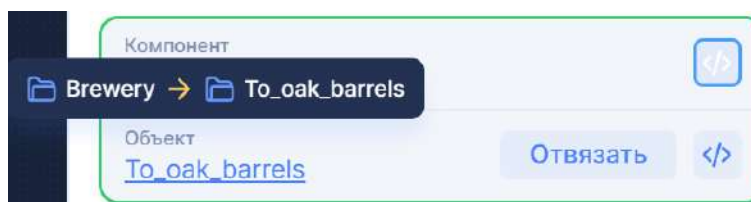


Рис. 3.108. Отображение полного пути объекта

По нажатию на любой из отрезков пути можно перейти непосредственно в соответствующую ветку редактора объектов Среды Администрирования (рис. 3.109).

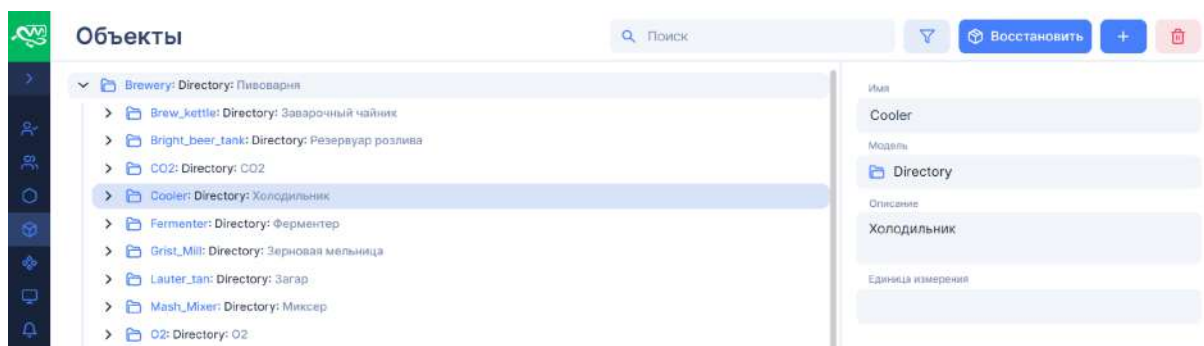


Рис. 3.109. Дерево объектов

Предварительная анимация мнемосхемы позволяет оценить корректность её работы. Для запуска анимации необходимо нажать на соответствующую кнопку (рис. 3.110).



Рис. 3.110. Кнопка запуска анимации

После чего появится страница анимации мнемосхемы (рис. 3.111).

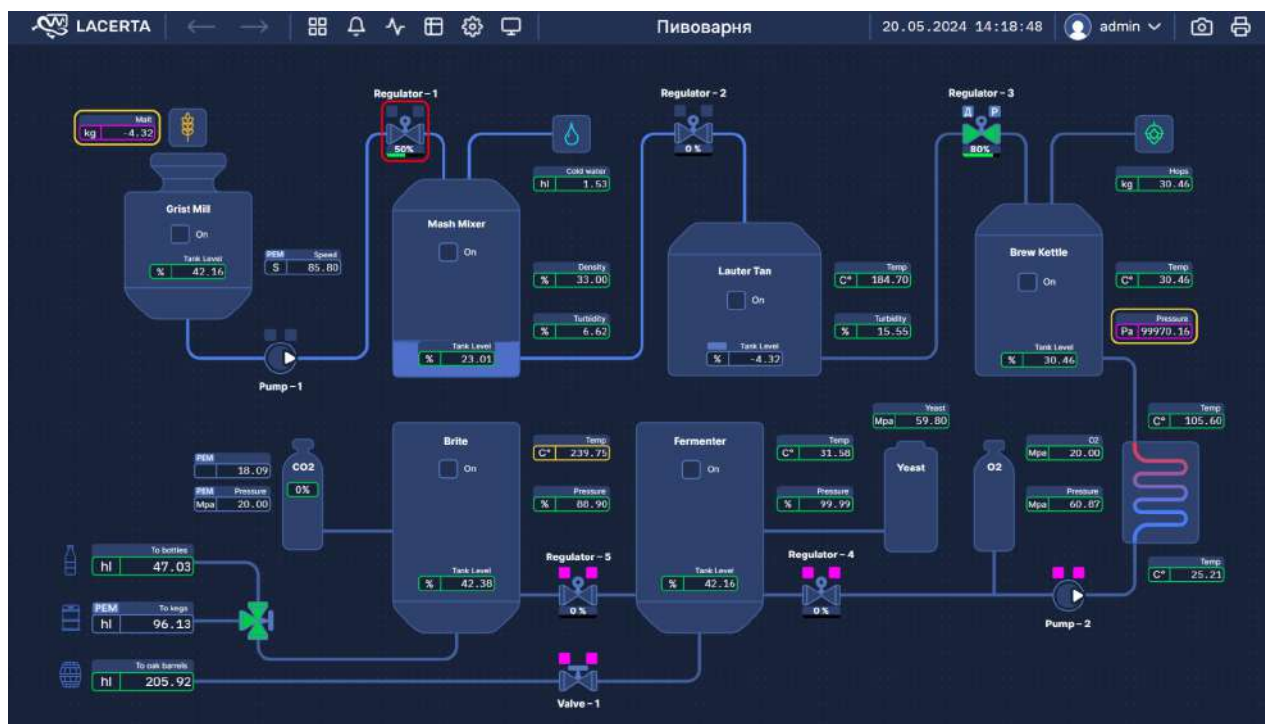


Рис. 3.111. Анимация Мнемосхемы

На мнемосхеме возможно отображение тренда отдельного сигнала с помощью специальной функции в редакторе скрипта (рис. 3.112).

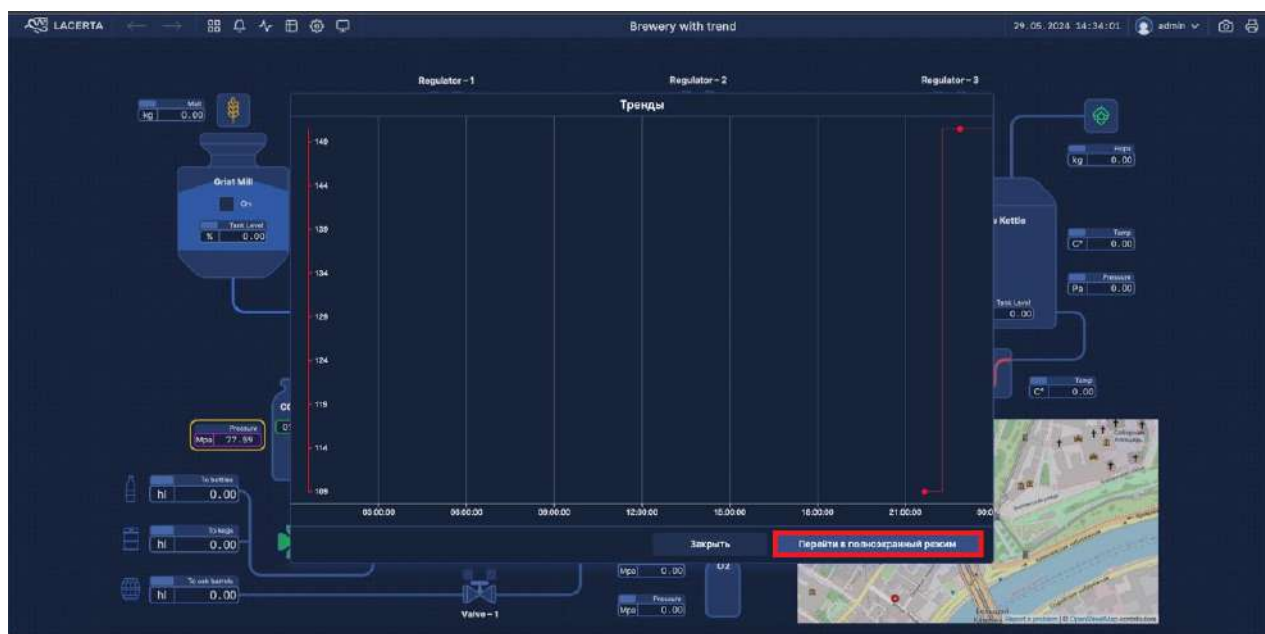


Рис. 3.112. Отображение тренда сигнала

Для перехода на страницу тренда необходимо нажать «Перейти в полноэкранный режим» (рис. 3.112).

Для управления отдельными элементами Системы на мнемосхеме следует выбрать элемент и во всплывающем окне выбрать действие. Для примера приведено окно управления насосом (включение/выключение, рис. 3.113).

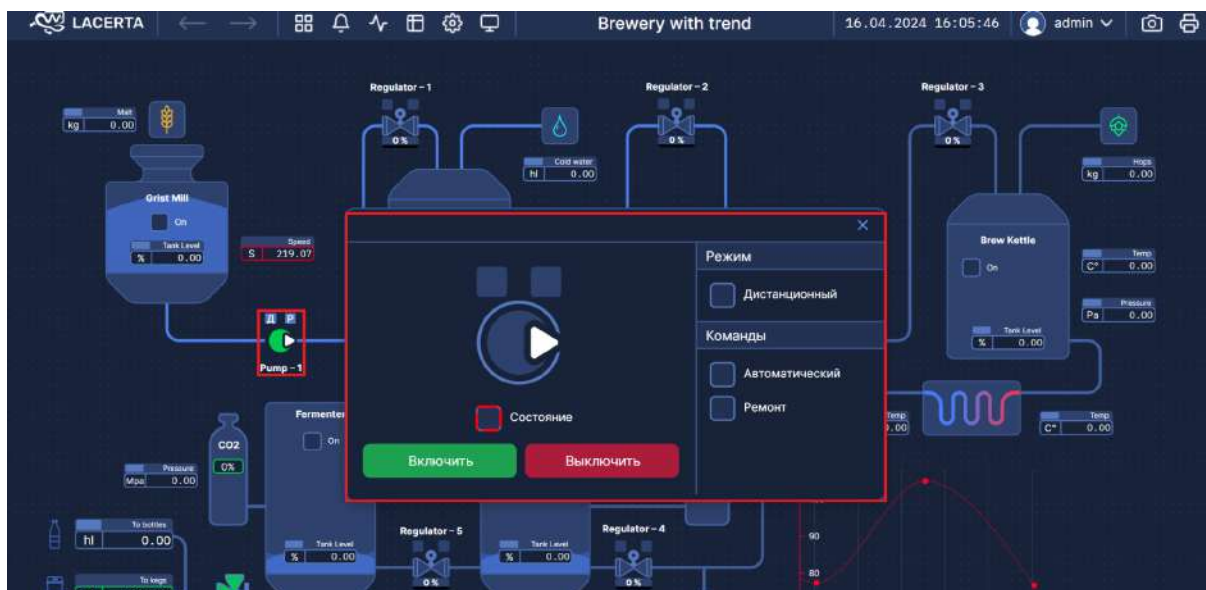


Рис.3 113. Окно управления насосом

При данных командах пользователю дается 5 секунд на подтверждение действия, по истечении которых окно автоматически закрывается.

В режиме запуска у пользователя есть возможность оставить комментарий к элементу. Для этого необходимо выбрать элемент и нажать «Добавить комментарий», после чего откроется окно, в котором нужно ввести комментарий. По завершении нажать «Применить» (рис. 3.114).

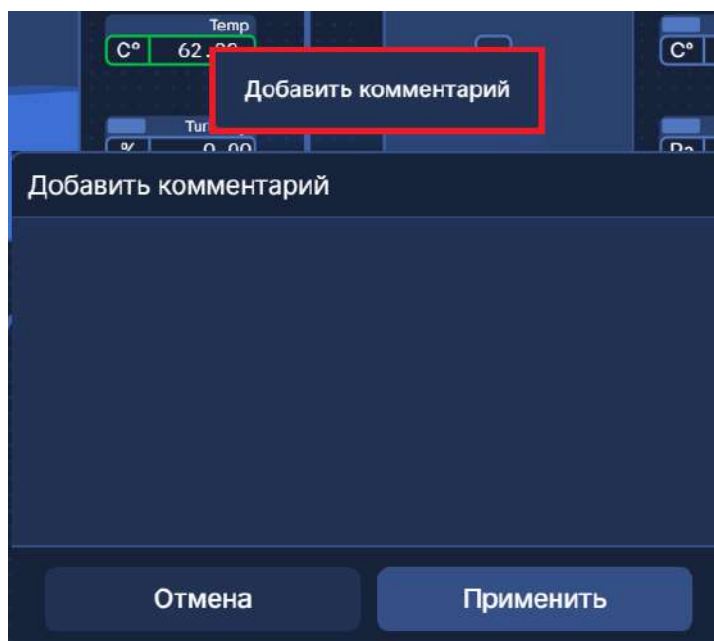


Рис. 3.114. Добавление комментария к элементу

После добавления комментария напротив элемента отобразится соответствующий значок отображающий его наличие (рис. 3.115).

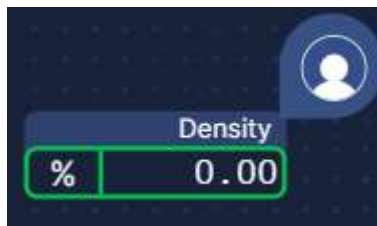


Рис. 3.115. Отображение наличия комментария

Редактирование и удаление комментариев происходит аналогичным образом (рис. 3.114). Для того чтобы удалить/отредактировать комментарий нужно нажать на элемент и во всплывшем окне задать комментарий. По завершении нажать на «Применить».

3.5.6. Редактор сценариев поведения компонентов

При выборе компонента в разделе «Компоненты» пользователь получает возможность создания/редактирования сценария его работы.

Для добавления сценария работы компонента необходимо добавить соответствующую ему модель. Затем перейти на страницу редактирования скрипта компонента (рис. 3.116).

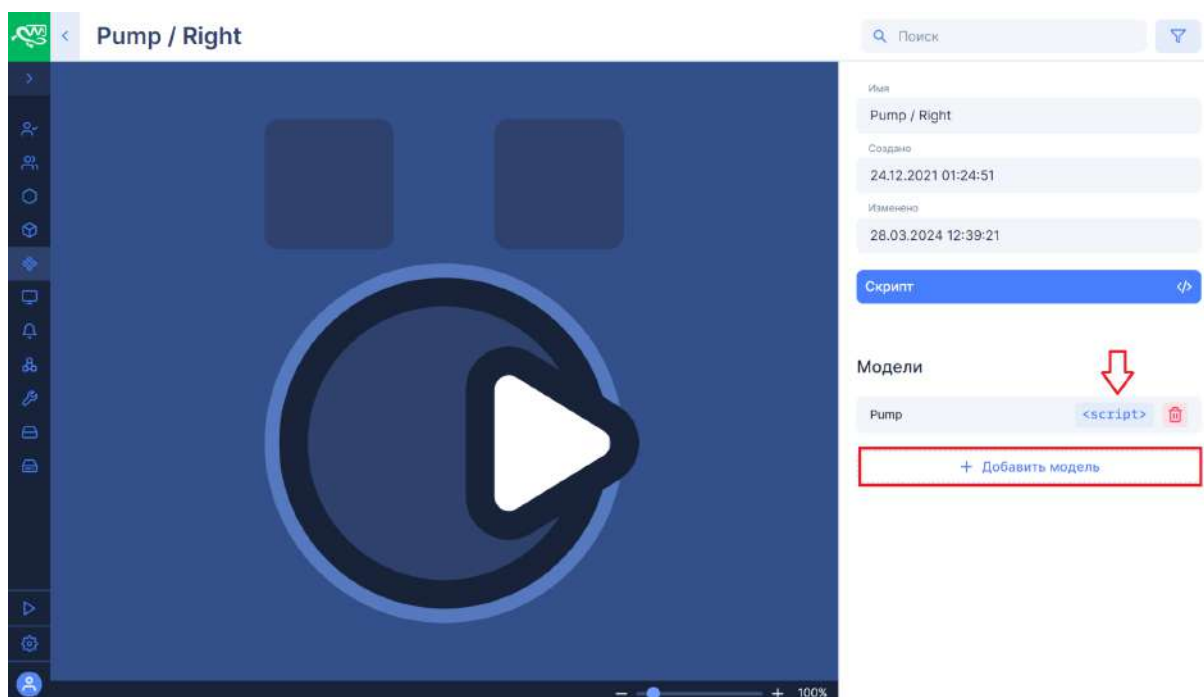


Рис. 3.116. Добавление модели в компонент

Для сценариев поведения компонента используется язык TypeScript (рис. 3.117).

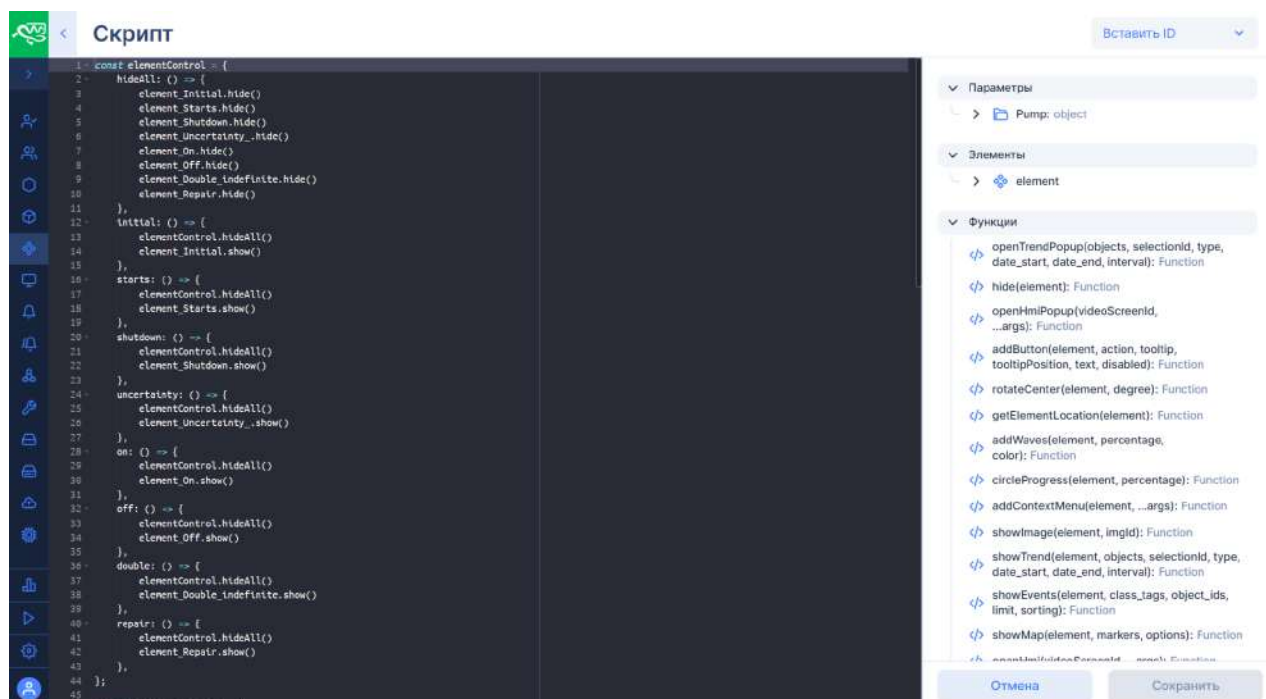


Рис. 3.117. Окно редактора сценариев

На странице редактора отображен список доступных параметров и функций. Таким образом, пользователю не требуется вводить весь текст скрипта вручную, для вставки функции/параметра достаточно выбрать из списка соответствующее значение.

В списке «Параметры» доступна структура привязанной к компоненту модели. При нажатии на атрибуты модели в редактор подставляется весь путь к атрибуту. Тем самым, можно обращаться к id привязанного объекта, его значению, описанию и имени. Аналогично работает и со всеми его вложенными моделями (рис. 3.118).

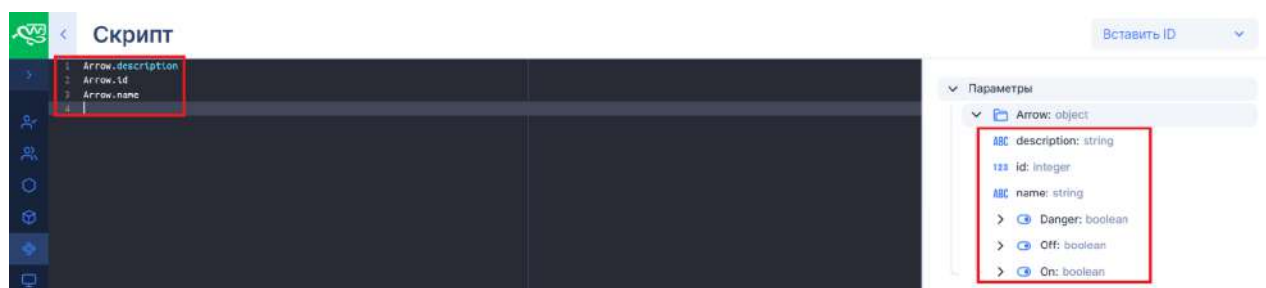


Рис. 3. 118. Работа с атрибутами модели

В списке «Элементы» отображаются всевозможные модификации (состояния) выбранного компонента (рис. 3.119).

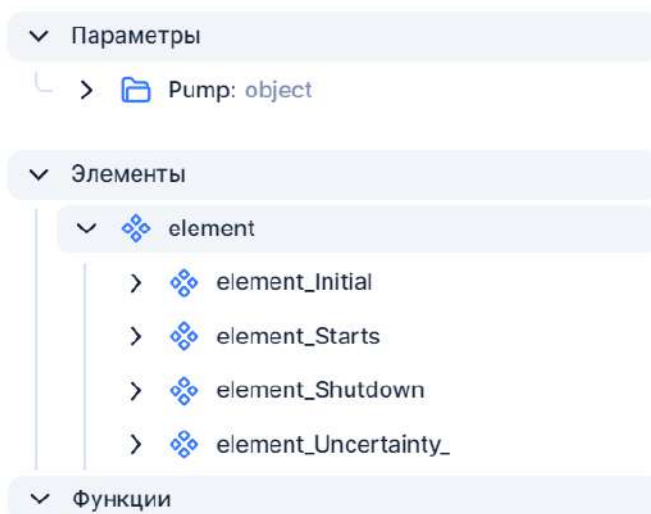


Рис. 3.119. Список состояний компонента

3.5.7. Редактор сценариев мнемосхем

Сценарий поведения мнемосхемы определяется глобальным скриптом мнемосхемы и скриптами элементов, размещенных на этой мнемосхеме. Редактор глобального скрипта и редактор скрипта компонентов имеет одинаковый интерфейс, но список их функций будет отличаться.

Для перехода в глобальный скрипт мнемосхемы необходимо нажать на кнопку «Скрипт» в редакторе мнемосхемы (рис 3.120).



Рис. 3.120. Скрипт мнемосхемы

Для сценариев поведения мнемосхемы используется язык TypeScript (рис. 3.121).

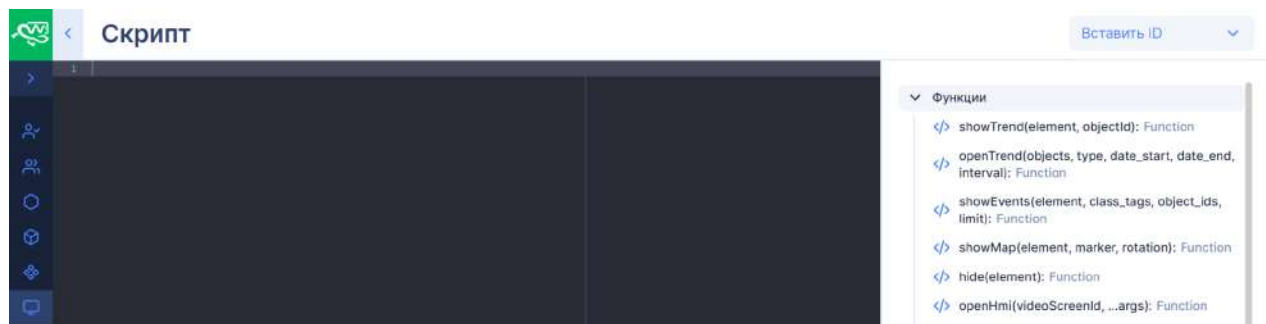


Рис. 3.121. Редактор сценариев

На странице редактора отображается список доступных функций с возможностью их выбора (рис. 3.121).

Для удобства можно воспользоваться выпадающим списком, ID из которого могут выступать в качестве параметров функций. В выпадающем списке пользователь должен выбрать нужное значение (вставка ID объекта, ID выборки, ID мнемосхемы, ID группы событий, рис. 3.122).

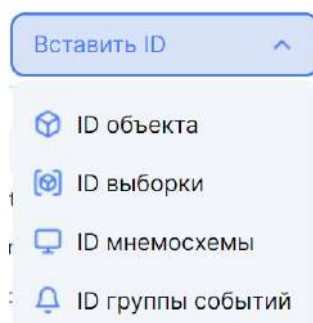


Рис. 3.122. Выпадающий список идентификаторов

После чего откроется модальное окно, в котором также выбрать параметр и нажать «Вставить» (рис. 3.123).

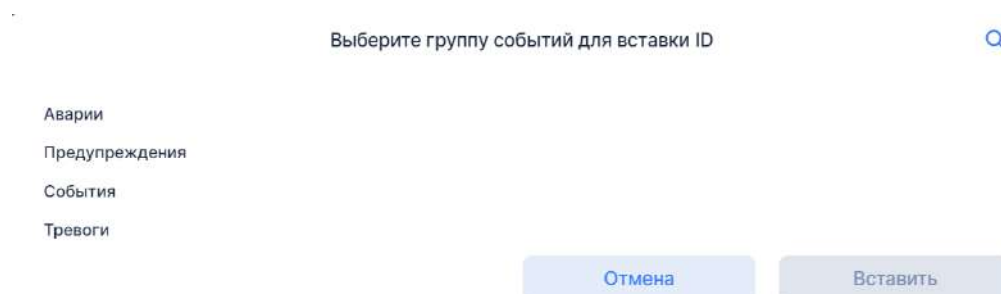


Рис. 3.123. Модальное окно параметров

3.5.8. Функции скрипта мнемосхемы/композиционной мнемосхемы

С помощью языка TypeScript пользователь задает сценарии поведения мнемосхемы в Скрипте (рис. 3.124).

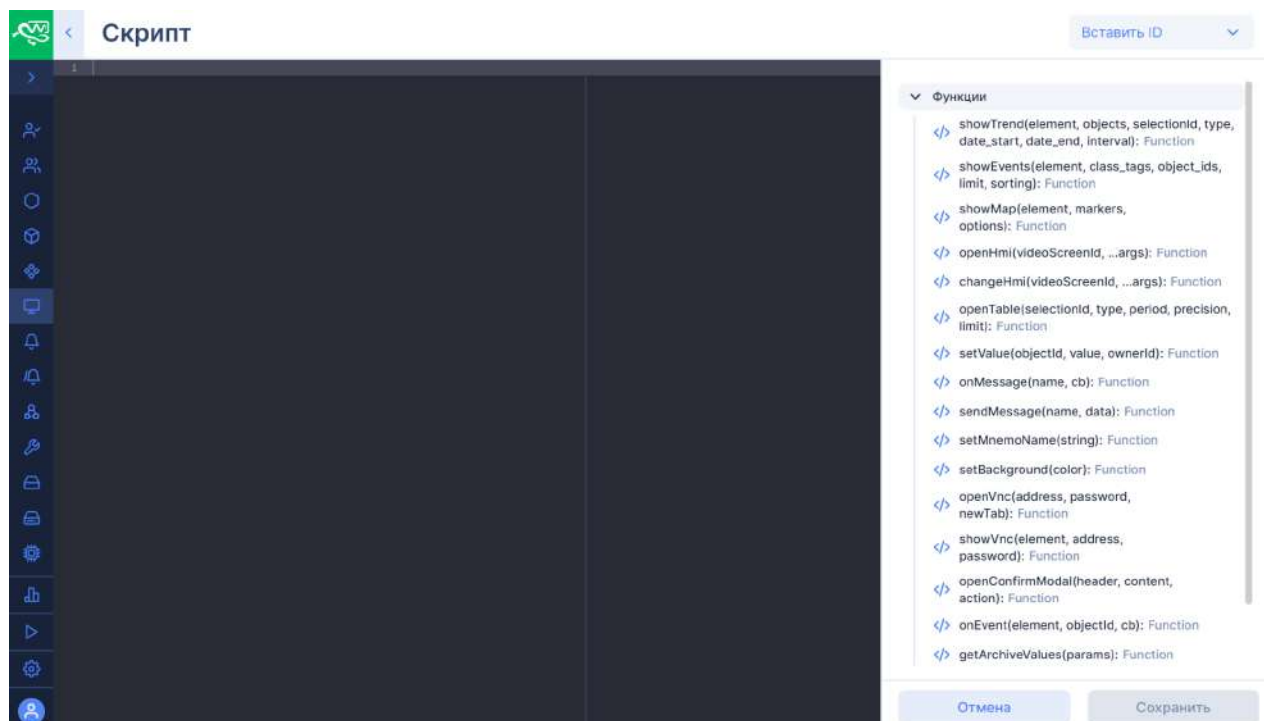


Рис. 3.124. Редактор сценариев

Сценарии поведения задаются через функции. Каждая функция имеет свое назначение и определенные параметры.

В редакторе представлены следующие функции:

1. `showTrend(element: Element, objects?: number | number[], selectionId?: number, type?: number, date_start?: number, date_end?: number | null, interval?: number)` - функция отображения тренда. Используется для отображения тренда на отдельной мнемосхеме (фрагменте мнемосхемы).

Типы данных функции:

- `number` - число.

Параметры функции:

- `element` - элемент мнемосхемы (компонент элемента мнемосхемы);
- `objects` - параметр, в который передается массив `id` объектов для отображения тренда;
- `selectionId` - `id` выборки;

- type - тип графика (0 - «Кривая», 1 - «Ступенчатая», 2 - «Мультикривая», 3 - «Мультиступенчатая», 4 - «Мультикривая (Одна ось)», 5 - «Мультиступенчатая (Одна ось)»);
- date_start - начало (левая граница) периода (в миллисекундах);
- date_end - конец (правая граница) периода (в миллисекундах);
- interval - интервал (в миллисекундах).

2. showEvents(element: Element, class_tags?: number[], object_ids?: string[], limit?: number, sorting?: number[]) - функция отображения событий. Используется для отображения событий на отдельной мнемосхеме (фрагменте мнемосхемы).

Типы данных функции:

- number - число;
- string - строка.

Параметры функции:

- element - элемент мнемосхемы (компонент элемента мнемосхемы);
- class_tags - массив id групп событий;
- object_ids - массив id объектов;
- limit - количество отображаемых событий на страницу (при игнорировании параметра отображается режим «Видимые»);
- sorting - массив чисел, каждое из которых отвечает за сортировку:

priority (приоритет) = 0,

activity (активность) = 1,

check (квитирование) = 2.

3. showMap(element, markers, option) - функция отображения карты. Используется для отображения карты с маркером и возможностью вращения маркера на отдельном фрагменте мнемосхемы .

Параметры функции:

- element - ID или класс HTML-элемента, в который будет вставлена карта;
- markers - объект в формате json, который содержит информацию о маркерах, отображающихся на карте;
- option - объект в формате json с дополнительными параметрами.

4. `openHmi(videoScreenId: number, ...args: number[])` - функция глобального перехода к мнемосхеме.

Типы данных функции:

- `number` - число.

Параметры функции:

- `videoScreenId` - id мнемосхемы;
- `...args` - перечень аргументов мнемосхемы.

5. `changeHmi(videoScreenId: number, ...args: number[])` - функция смены текущей мнемосхемы. Используется в композитной мнемосхеме.

Типы данных функции:

- `number` - число.

Параметры функции:

- `videoScreenId` - id мнемосхемы;
- `...args` - перечень аргументов мнемосхемы.

6. `openTable(selectionId?: number, type?: number, period?: number, precision?: number, limit?: number)` - функция глобального перехода в табличное представление данных.

Типы данных функции:

- `number` - число.

Параметры функции:

- `selectionId` - id выборки объектов;
- `type` - тип таблицы (0 - "Все значения", 1 - "Актуальные значения", 2 - "Сводная таблица");
- `period` - величина периода отображения данных;
- `precision` - отображение количества точек после запятой у числовых значений;
- `limit` - количество строк.

7. setValue(objectId: number, value: string, ownerId?: number) - установить значение для объекта.

Типы данных функции:

- number - число;
- string - строка.

Параметры функции:

- objectId - id объекта;
- value - значение;
- ownerId - идентификатор того, кто выставляет значение (необязательный параметр).

8. setMnemonicName(string: string) - установить имя мнемосхемы в режиме исполнения.

Типы данных функции:

- string - строка.

Параметры функции:

- string - имя мнемосхемы.

9. setBackground(color: string) - установка фона.

Типы данных функции:

- string - строка.

Параметры функции:

- color - значение цвета (HEX/RGB)

10. openVnc(address?: string | string[], password?: string, newTab?: boolean) - открытие VNC-сессии в текущем или новом окне.

Типы данных функции:

- string - строка;
- boolean - 1/0.

Параметры функции:

- address - адрес;
- password - пароль;

- newTab - 0 (в текущем окне) / 1 (в новом окне).

11. showVnc(element: Element, address: string | string[], password: string)-
открытие VNC-сессии в элементе.

Типы данных функции:

- number - число;
- string - строка.

Параметры функции:

- element - элемент мнемосхемы (компонент элемента мнемосхемы);
- address - адрес;
- password - пароль.

12. openConfirmModal(header: string, content: string, action: Function) -
отображение модального окна подтверждения действия.

Типы данных функции:

- string - строка.
- Function - функция.

Параметры функции:

- header - заголовок окна;
- content - содержание (текст) внутри модального окна;
- action - действие.

13. onEvent(element: Element, objectId: number, cb: (events: IEvent[]) => void) -
подписка на события объекта.

Типы данных функции:

- number - число;
- function - функция.

Параметры функции:

- element - элемент мнемосхемы (компонент элемента мнемосхемы);
- objectId - id объекта;
- cb - callback-функция в случае получения события.

14. `getArchiveValues`(`archiveOptions`: {`date_start`?: number, `date_end`?: number, `object_ids`: number[], `interval`: number, `count`?: number, `from`?: number}) - получение значений из архива.

Типы данных функции:

- number - число.

Параметры функции:

- `date_start` - начало (левая граница) периода (миллисекунды);
- `date_end` - конец (правая граница) периода (миллисекунды);
- `object_ids` - массив id объектов;
- `interval` - интервал (миллисекунды);
- `count` - количество значений (строк) на странице;
- `from` - смещение относительно начала списка (для постраничного вывода на экран).

15. `formatTime`(`time`: number, `format`?: string) - форматирование времени с поправкой на серверное время.

Типы данных функции:

- number - число;
- string - строка.

Параметры функции:

- `time` - значение времени;
- `format` - формат даты (YYYY-MM-DDTHH:mm:ss.sssZ).

16. `showLoader`(`element`: Element, `loading`: boolean) - отображение элемента загрузки.

Типы данных функции:

- boolean - 1/0.

Параметры функции:

- `loading` - 0 (не отображать элемент загрузки)/ 1 (отображать элемент загрузки).

Следующие функции используются в связке:

1. `sendMessage(name: string, data: any)` - отправка сообщения.

Типы данных функции:

- `string` - строка.

Параметры функции:

- `name` - название сообщения;
- `data` - данные для получения ответа в функции `onMessage()`.

2. `onMessage(name: string, cb: (data: any) => void)` - прослушивание сообщения.

Типы данных функции:

- `string` - строка;
- `function` - функция.

Параметры функции:

- `name` - название сообщения;
- `cb` - callback - функция в случае получения сообщения.

3.5.9. Функции скрипта компонентов и элементов мнемосхемы

С помощью языка TypeScript пользователь задает сценарии поведения компонентов и элементов мнемосхемы в Скрипте (рис. 3.125).

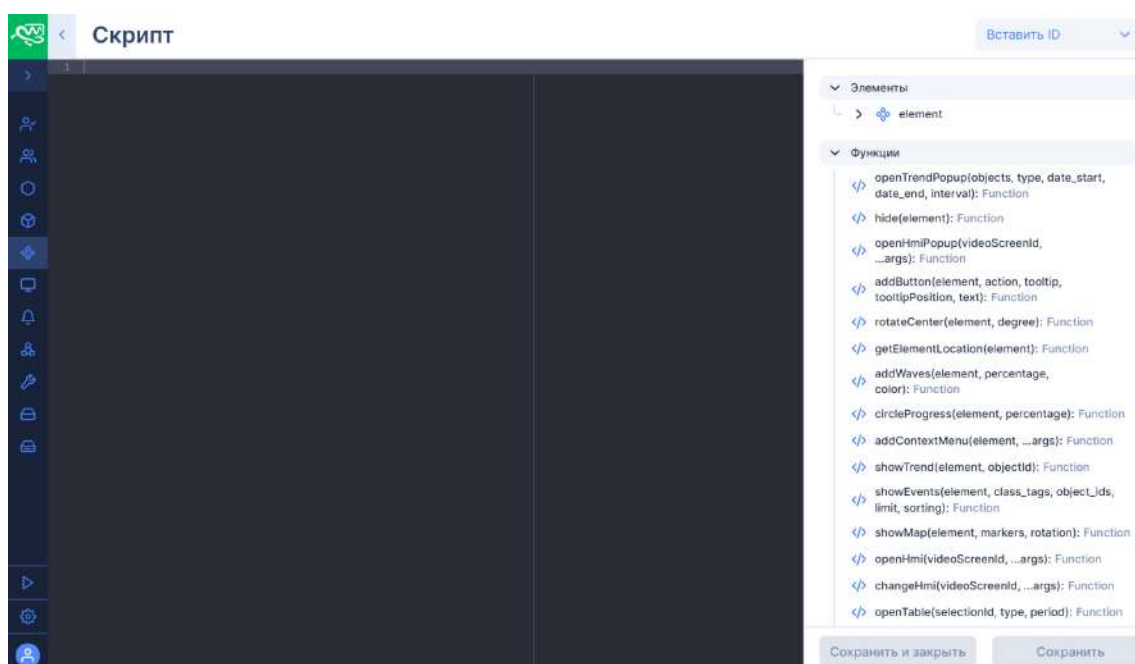


Рис. 3.125. Редактор сценариев

Сценарии поведения задаются через функции. Каждая функция имеет свое назначение и определенные параметры.

В редакторе представлены следующие функции:

1. `openTrendPopup(objects?: number | number[], selectionId?: number, type?: number, date_start?: number, date_end?: number | null, interval?: number)` - открыть тренд в модальном окне.

Типы данных функции:

- `number` - число.

Параметры функции:

- `objects` - массив id объектов;
 - `selectionId` - id выборки;
 - `type` - тип графика (0 - «Кривая», 1 - «Ступенчатая», 2 - «Мультикривая», 3 - «Мультиступенчатая», 4 - «Мультикривая (Одна ось)», 5 - «Мультиступенчатая (Одна ось)»);
 - `date_start` - начало (левая граница) периода (в миллисекундах);
 - `date_end` - конец (правая граница) периода (в миллисекундах);
 - `interval` - интервал (в миллисекундах).
2. `hide(element: SVGSVGElement)` - скрыть элемент.
 3. `openHmiPopup(videoScreenId: number, ...args: number[])` - открыть мнемосхему в модальном окне

Типы данных функции:

- `number` - число.

Параметры функции:

- `videoScreenId` - id мнемосхемы;
 - `args` - перечень аргументов.
4. `addButton(element: SVGSVGElement, action: any, tooltip?: string, tooltipPosition?: 'top' | 'left' | 'right' | 'down', text?: string, disabled?: boolean)` - добавить кнопку на элемент.

Типы данных функции:

- `string` - строка;
- `boolean` - 1/0.

Параметры функции:

- action - действие;
 - tooltip - заголовок тултипа (всплывающее окно при наведении на элемент с кнопкой);
 - tooltipPosition - расположение тултипа ('top' - сверху, 'left' - слева, 'right' - справа, 'down' - снизу);
 - text - отображение текста на элементе;
 - disabled – состояние кнопки (1 – отключена (некликабельна); 0 – включена (кликабельна)) .
5. rotateCenter(element: SVGSVGElement, degree: number[]) - повернуть элемент на заданное количество градусов относительно центра.

Типы данных функции:

- number - число;

Параметры функции:

- degree - значение величины градуса.

6. getElementLocation(element: SVGSVGElement) - получить позицию элемента.
7. addWaves(element: SVGGraphicsElement, percentage: number, color: string) - добавить волны на элемент.

Типы данных функции:

- number - число;
- string - строка.

Параметры функции:

- percentage - процент заполнения элемента;
- color - значение цвета.

8. circleProgress(element: SVGGraphicsElement, percentage: number) - показать прогресс.

Типы данных функции:

- number - число.

Параметры функции:

- percentage - процент заполнения элемента.

9. `addContextMenu(element: SVGGraphicsElement, ...args: (string | Function)[])` - добавление контекстного меню к элементу (правая кнопка мыши).

Типы данных функции:

- `string` - заголовок;
- `function` - функция (действие).

Параметры функции:

- `...args` - перечень кнопок в контекстном меню.

10. `showImage(element: SVGGraphicsElement, imgId: number)` - открытие изображения.

Типы данных функции:

- `number` - число.

Параметры функции:

- `imgId` - id объекта с изображением.

Следующие функции совпадают с функциями из раздела [3.5.8](#) и относятся к глобальному редактору скрипта (раздел [3.5.7](#)):

- `showTrend();`
- `showEvents();`
- `showMap();`
- `openHmi();`
- `changeHmi();`
- `openTable();`
- `setValue();`
- `onMessage();`
- `sendMessage();`
- `setMnemonicName();`
- `setBackgroundColor();`
- `openVnc();`
- `showVnc();`
- `openConfirmModal();`
- `onEvent();`
- `getArchiveValues();`
- `formatTime();`
- `showLoader()` .

3.5.10 Специальные возможности

Пользователь имеет возможность отобразить объект на карте (рис. 3.126).

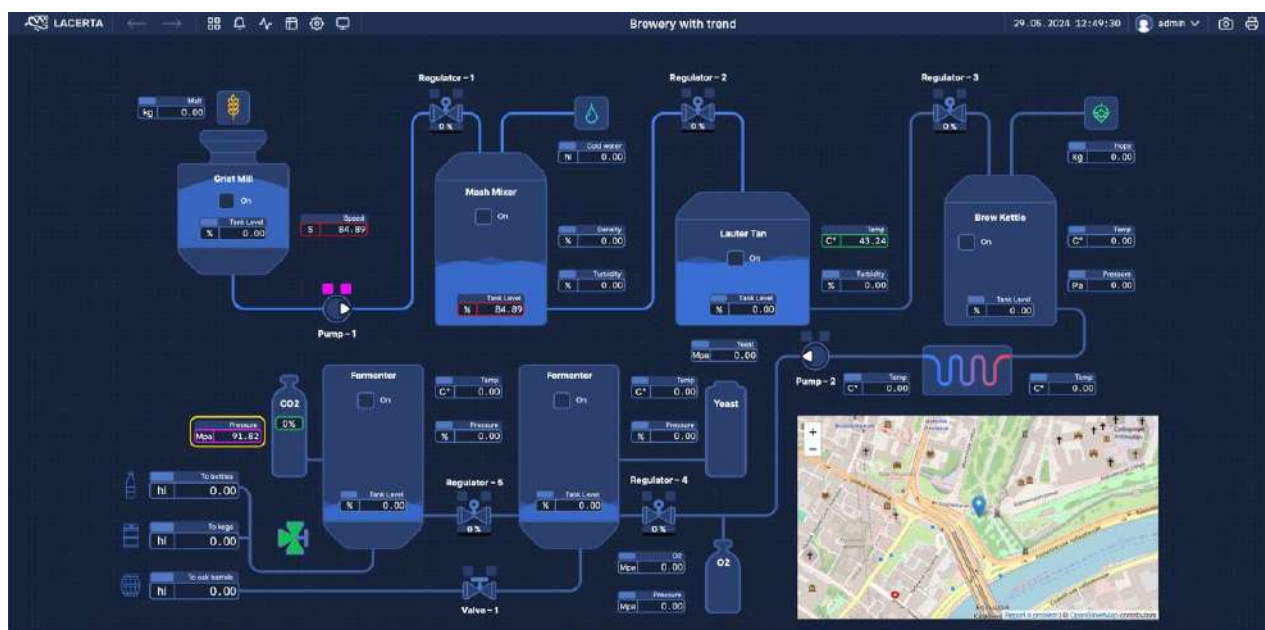


Рис. 3.126. Отображение объекта на карте

Для этого необходимо задать функцию, в которой будут указаны координаты и коэффициент масштабирования карты (рис. 3.127).



Рис. 3.127. Функция отображения объекта на карте

3.5.11. Фрагменты

Для размещения на мнемосхеме сложных элементов или виджетов в Платформе реализован механизм фрагментов (рис. 3.128).

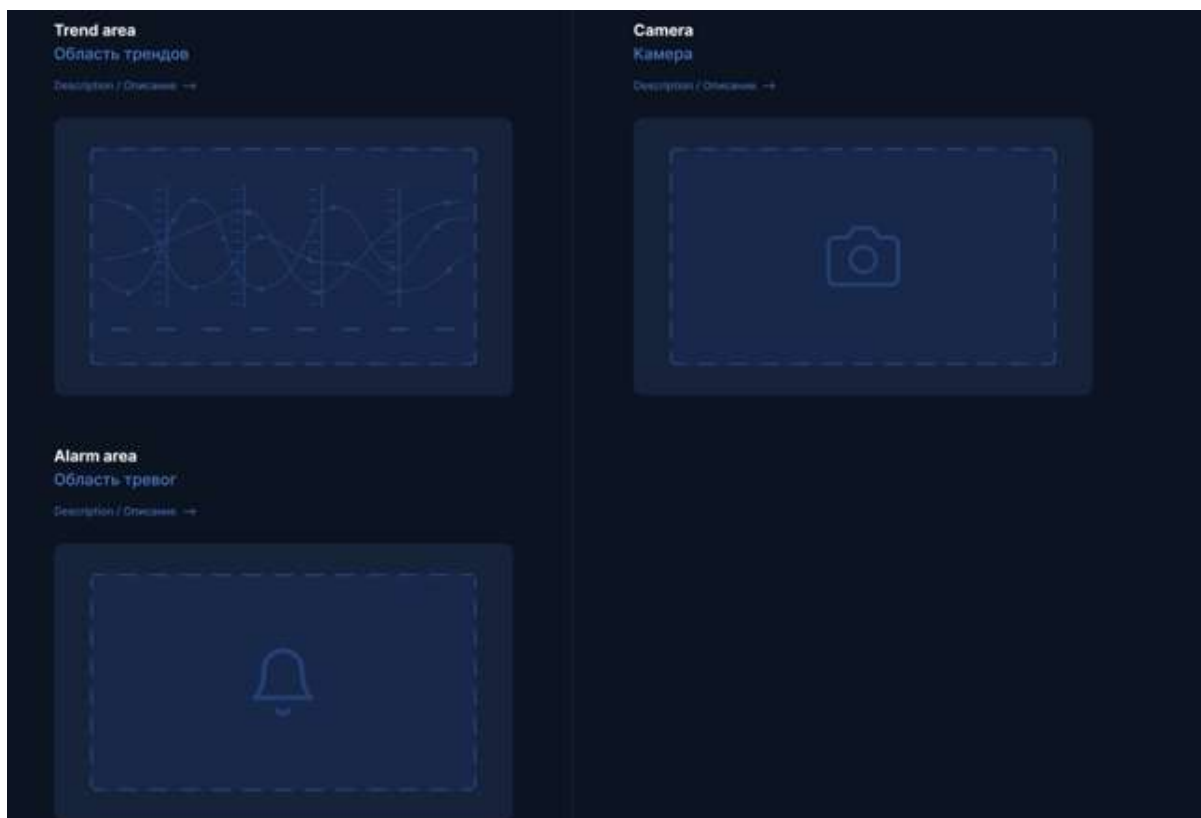


Рис. 3.128. Примеры фрагментов

Фрагмент - это встроенный в мнемосхему кадр, который может размещаться как стационарно, так и вызываться модально (всплывающий фрагмент) и внутри которого отображаются результаты работы отдельных приложений (например, выводятся графики, тренды, таблицы, списки событий, видеотрансляция и т.п.). Фрагмент также может содержать мнемосхему отдельных узлов системы, например, мнемосхема насосной станции может содержать набор фрагментов, в которых размещены мнемосхемы отдельных насосов.

На фрагменте мнемосхемы может быть выведен сигнал в виде графика (тренда, рис. 3.129).

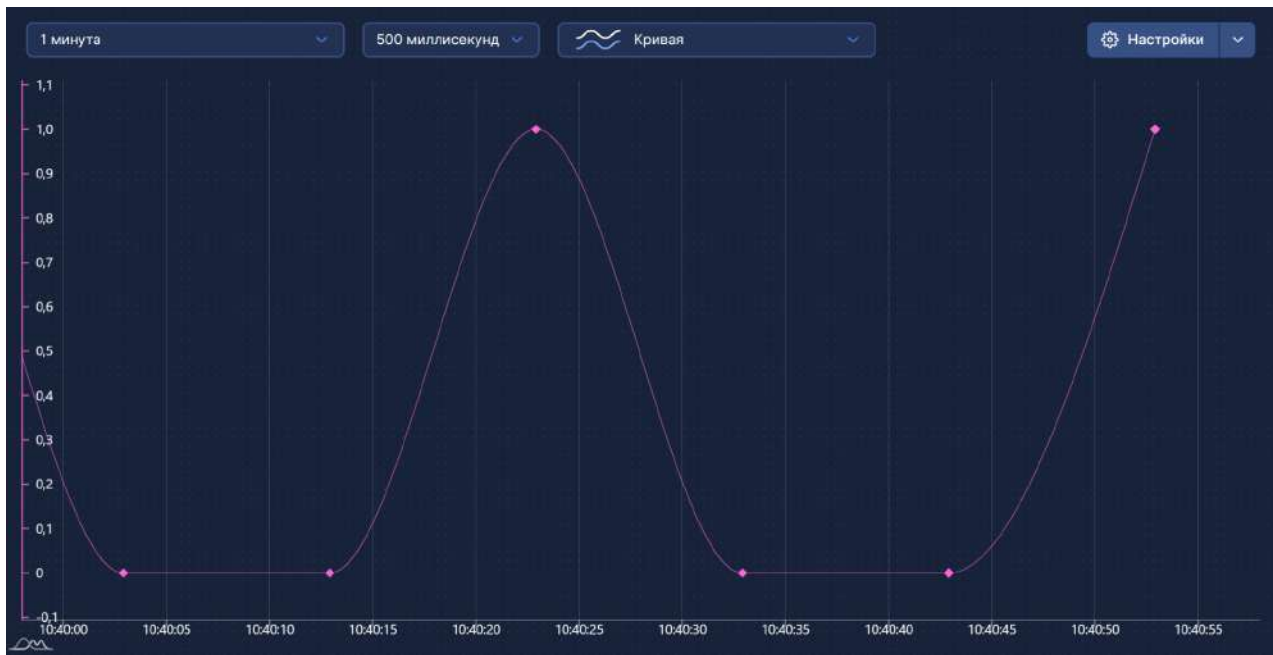


Рис. 3.129. График

Пример вывода списка событий с активными элементами: выбор только актуальных/архивных событий, выбор групп событий, отображение только активных событий, отображение по дате и времени, отображение событий по приоритету, выбор страниц, отображение количества событий на страницу, прокручивание списка, возможность квитировать событие (рис. 3.130).

Объект	Дата и время	Сообщение	Пользователь	Состояние	Класс	Приоритет	Действие
Value	07.08.2024 16:27:33	Резервуар розлива выключен		Активное	Событие 2	2	Квитировать
High_gas	07.08.2024 15:21:03	35 кв79 Кон. газ прев.		Активное	Тревога	10	Квитировать
High_gas	07.08.2024 15:21:03	33 Ка14 Кон. газ прев.		Активное	Тревога	10	Квитировать
Warning	07.08.2024 15:21:03	Заварочный резервуар: Объем жидкости в норме		Активное	Событие 3	3	Квитировать
Warning	07.08.2024 15:21:02	С7 Клапан открыт		Активное	Событие	3	Квитировать
High_gas	07.08.2024 15:21:02	31 Ка4 Кон. газ прев.		Активное	Тревога	10	Квитировать
Valve_close	07.08.2024 15:21:02	36 Ка33 Клапан закрыт		Активное	Событие	5	Квитировать
No_connection_gas	07.08.2024 15:21:02	32 Ка83 Нет связи		Активное	Событие	5	Квитировать
Valve_no_connection	07.08.2024 15:21:02	34 Ка20 Нет связи с клап.		Активное	Предупр.	8	Квитировать
Fault	07.08.2024 15:21:00	Заварочный резервуар: Датчик мутности неисправен		Активное	Тревога 2	6	Квитировать
Value	07.08.2024 15:20:55	Пиво не вырабатывается!		Активное	Авария 1	1	Квитировать
Danger	07.08.2024 15:20:55	С3 МОП Влажность превышена		Активное	Событ.	9	Квитировать
Danger	07.08.2024 15:20:55	С2 Ка4 Давление снижено		Активное	Тревог.	1	Квитировать
Fault	07.08.2024 15:20:55	Заварочный резервуар: датчик уровня исправен		Активное	Событие 1	1	Квитировать
Fault	07.08.2024 15:20:55	Заварочный резервуар: Датчик плотности исправен		Активное	Событие 2	2	Квитировать

Актуальные

Группы событий

123>

Показать

Видимые

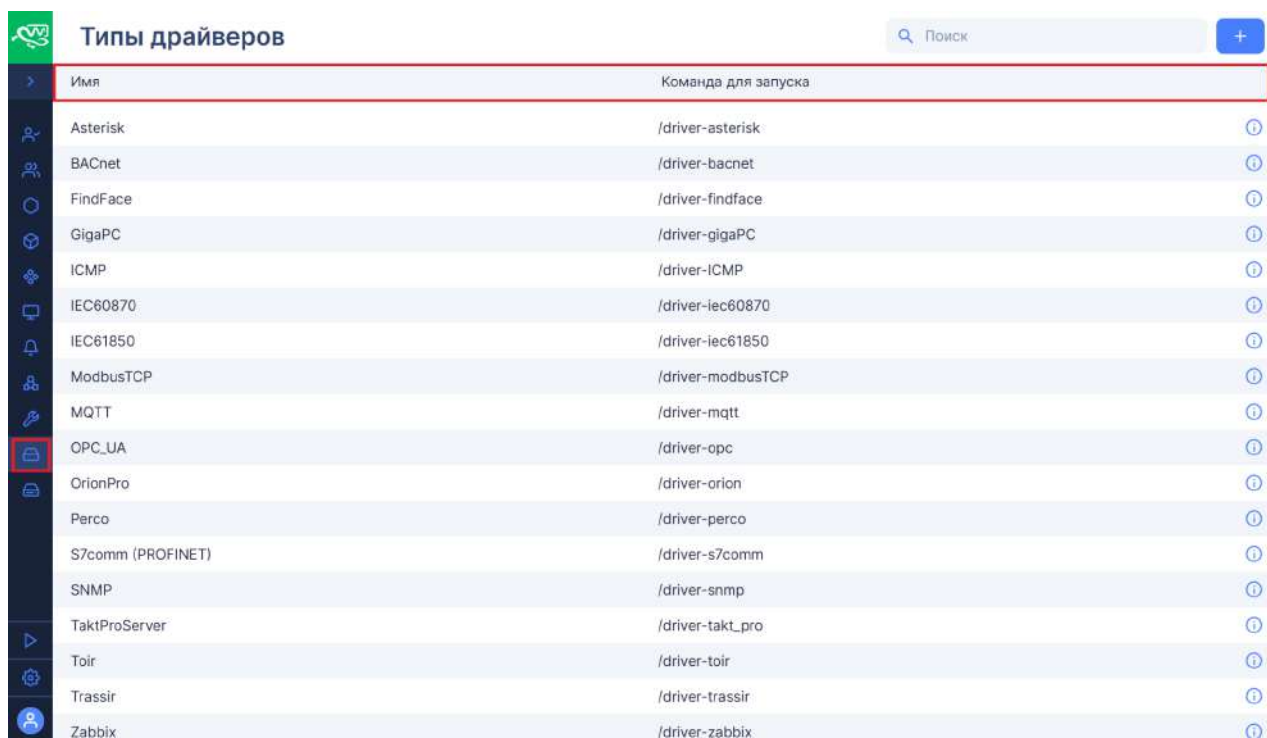
Рис. 3.130. Список событий с активными элементами

3.6. Драйверы протоколов АСУ ТП

Драйверы осуществляют сбор и предварительную обработку данных, поступающих из различных внешних источников (физических устройств, информационных систем), а также осуществляют передачу данных и/или команд управления на внешние устройства или информационные системы (разд. 2.5.2).

3.6.1. Список драйверов Платформы

Для перехода к списку Типов драйверов необходимо перейти в соответствующий раздел через кнопку бокового меню (рис. 3.131).

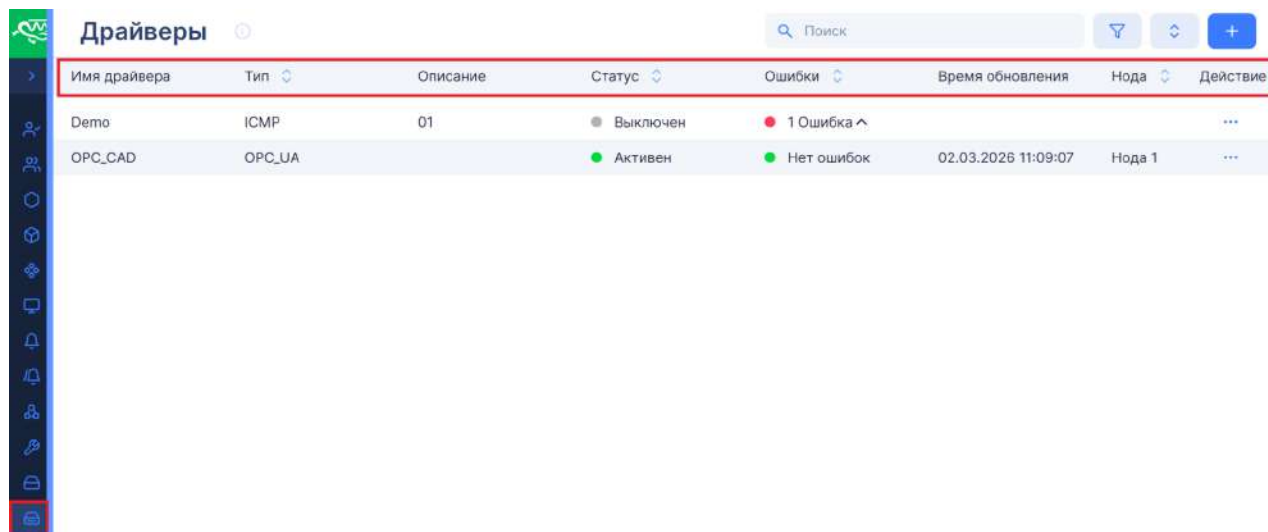


Типы драйверов		Поиск	+
Имя	Команда для запуска		
Asterisk	/driver-asterisk		
BACnet	/driver-bacnet		
FindFace	/driver-findface		
GigaPC	/driver-gigaPC		
ICMP	/driver-icmp		
IEC60870	/driver-iec60870		
IEC61850	/driver-iec61850		
ModbusTCP	/driver-modbusTCP		
MQTT	/driver-mqtt		
OPC-UA	/driver-opc		
OrionPro	/driver-orion		
Perco	/driver-perco		
S7comm (PROFINET)	/driver-s7comm		
SNMP	/driver-snmp		
TaktProServer	/driver-takt_pro		
Toir	/driver-toir		
Trassir	/driver-trassir		
Zabbix	/driver-zabbix		

Рис. 3.131. Таблица типов драйверов

В таблице содержатся имя типа драйвера и команда для запуска драйверов данного типа.

Для перехода к списку драйверов, используемых в Системе, необходимо перейти в раздел «Драйверы» через боковую панель меню (рис. 3.132).

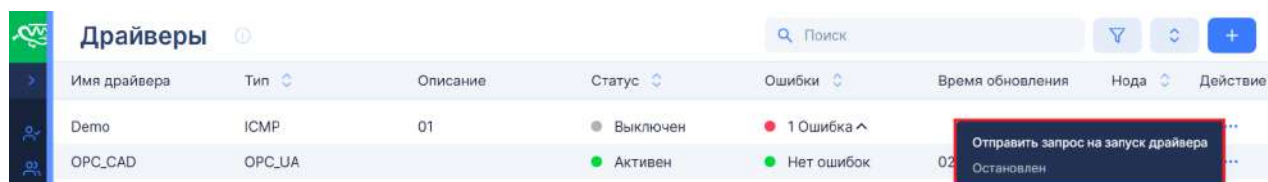


Имя драйвера	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
Demo	ICMP	01	Выключен	1 Ошибка			...
OPC_CAD	OPC-UA		Активен	Нет ошибок	02.03.2026 11:09:07	Нода 1	...

Рис. 3.132. Таблица драйверов Платформы

В таблице драйверов содержатся: имя драйвера, его тип, описание, статус, ошибки в работе, время обновления, ID узла (Нода), кнопка действия.

В колонке «Действие» выбранного драйвера при нажатии на «...» открывается список действий, доступных для пользователя (рис. 3.133).



Имя драйвера	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
Demo	ICMP	01	Выключен	1 Ошибка			...
OPC_CAD	OPC-UA		Активен	Нет ошибок	02		Отправить запрос на запуск драйвера Остановлен

Рис. 3.133. Выбор действия

Для того чтобы открыть дерево элементов драйвера или отправить запрос на запуск драйвера, необходимо воспользоваться одноименными кнопками в боковом окне параметров драйвера (рис. 3.134).

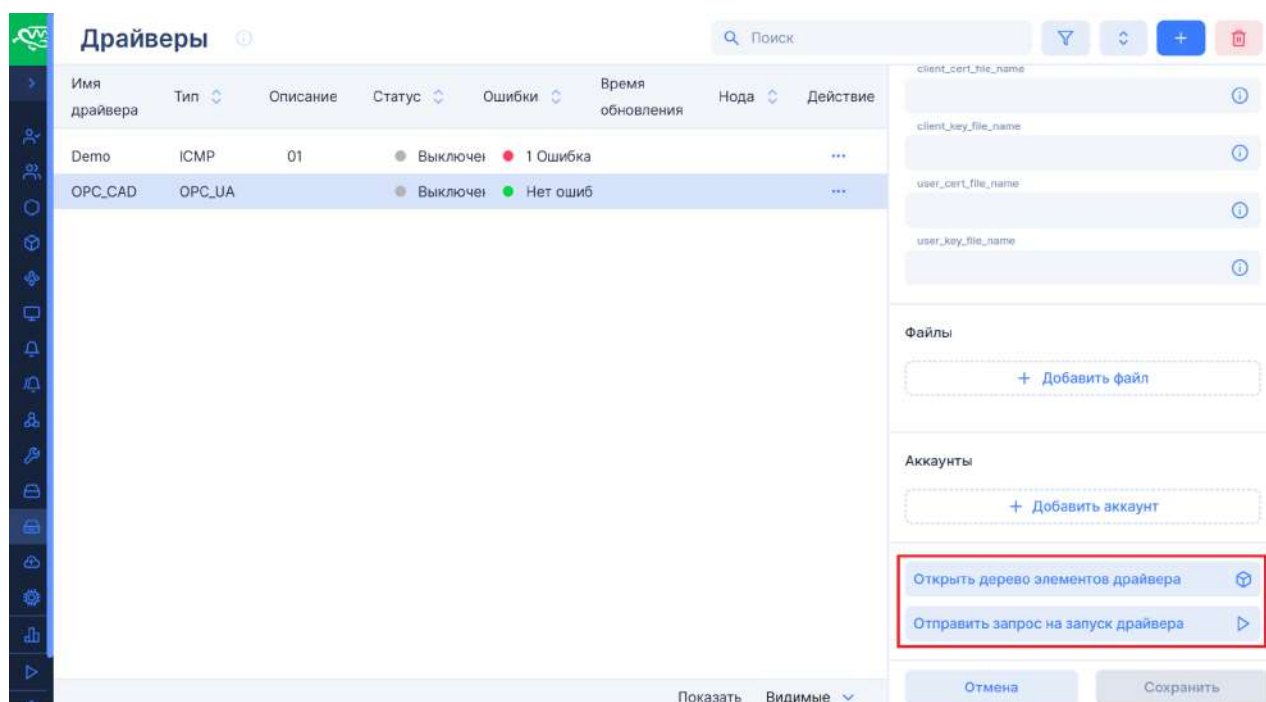


Рис.3.134. Кнопки действий

После запуска драйвера у пользователя появляется возможность обновить дерево элементов драйвера (при условии активного статуса и наличия дерева) нажав на соответствующую кнопку (рис. 3.135).

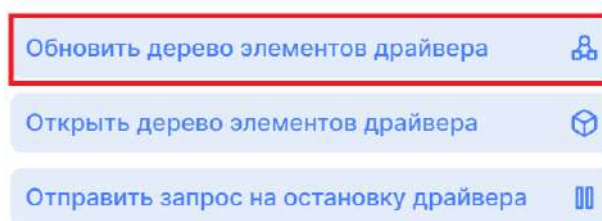


Рис. 3.135. Обновление дерева элементов драйвера

Конфигурация драйвера может быть описана во внешнем файле. Для загрузки конфигурации драйвера из внешнего файла следует нажать на кнопку «+ Добавить файл» и выбрать файл конфигурации в открывшемся меню (рис. 3.136).

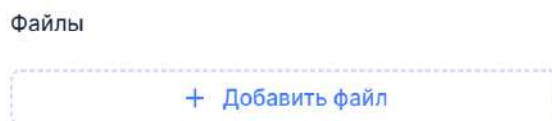


Рис. 3.136. Кнопка добавления файла

Некоторые драйверы могут выступать в роли отдельных серверов. Для подключения к таким драйверам-серверам пользователю необходимо добавить аккаунт в конфигурации драйвера (рис. 3.137).

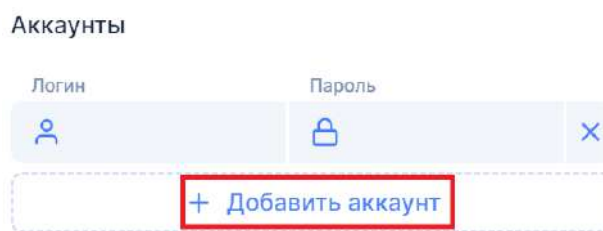


Рис. 3.137. Добавление аккаунта

Список драйверов можно отфильтровать по статусу, ошибкам и ноде при запуске. Для этого следует нажать на кнопку фильтра в верхней панели параметров (рис. 3.138).

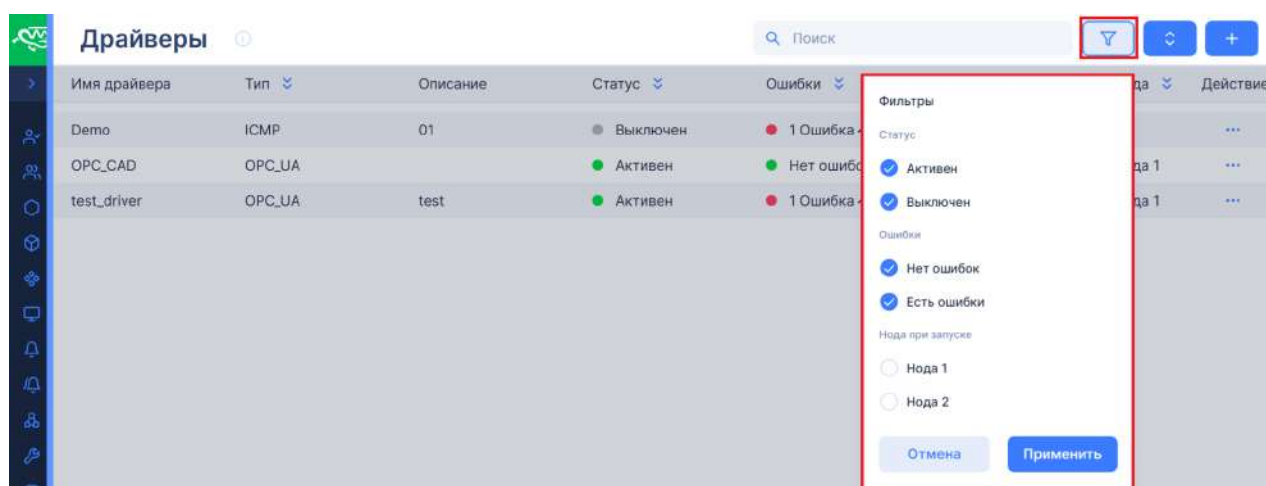


Рис. 3.138. Фильтрация списка драйверов

Также список драйверов можно отсортировать по их типу, статусу, ошибкам и ноде при запуске. Для этого нужно воспользоваться соответствующей кнопкой в верхней панели параметров (рис. 3.139).

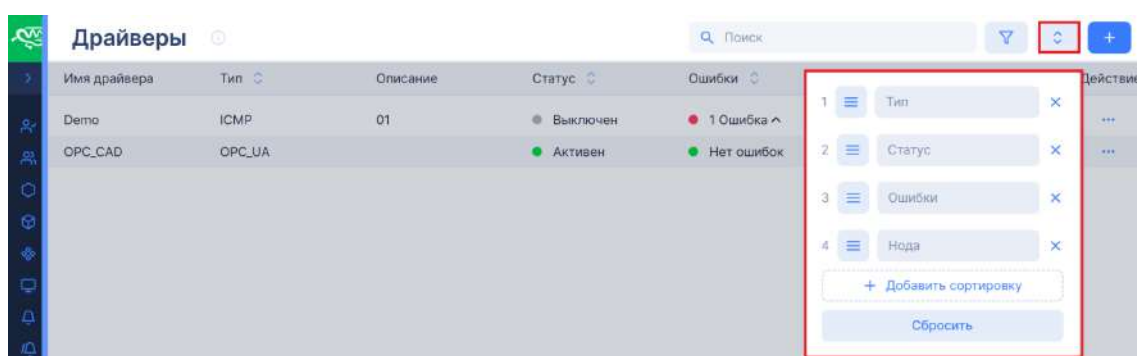


Рис. 3.139. Настройка сортировки

Добавить сортировку можно либо нажатием кнопки «+ Добавить сортировку» (рис. 3.139), либо выбрать в окне параметра сортировки (рис. 3.140).

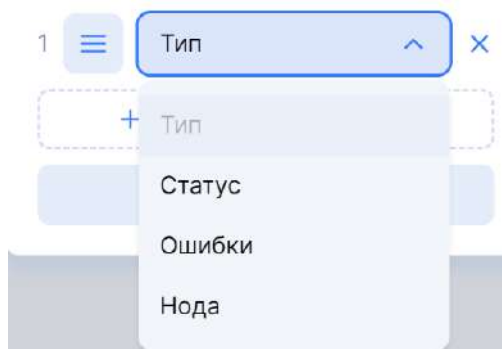


Рис. 3.140. Выбор параметра сортировки

Приоритет сортировки можно менять между собой местами с помощью перетаскивания (Drag-and-drop). Для этого следует нажать на соответствующую иконку напротив выбранного параметра и оттащить ее вниз или вверх, изменив расположение (рис. 3.141).

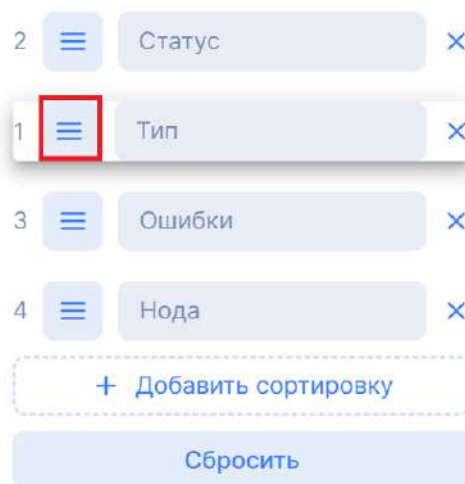


Рис. 3.141. Изменение приоритета сортировки

Аналогично можно воспользоваться сортировкой вручную напротив каждого из параметров (рис. 3.142).



Рис. 3.142. Сортировка вручную

3.6.2. Конфигуратор драйверов

Пользователь Платформы может самостоятельно конфигурировать, добавлять и удалять драйверы и их типы.

Для добавления нового типа драйвера необходимо перейти в раздел «Типы драйверов» через боковую панель меню (рис. 3.139).

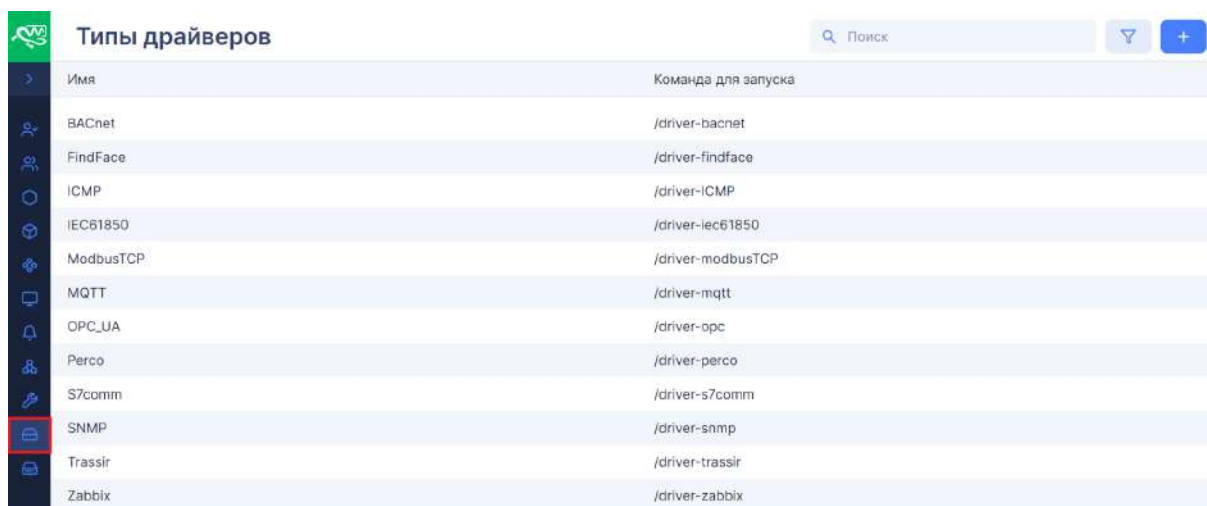


Рис. 3.139. Раздел Типы драйверов

Затем нажать на кнопку «Создать» в правом верхнем углу страницы (рис. 3.140).

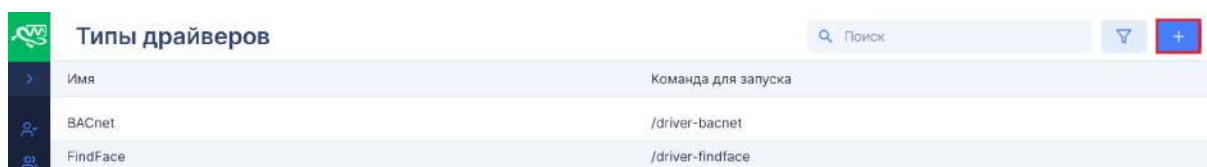


Рис. 3.140. Создание нового типа драйвера

После чего ввести необходимые параметры и нажать «Создать» по завершении (рис. 3.141).

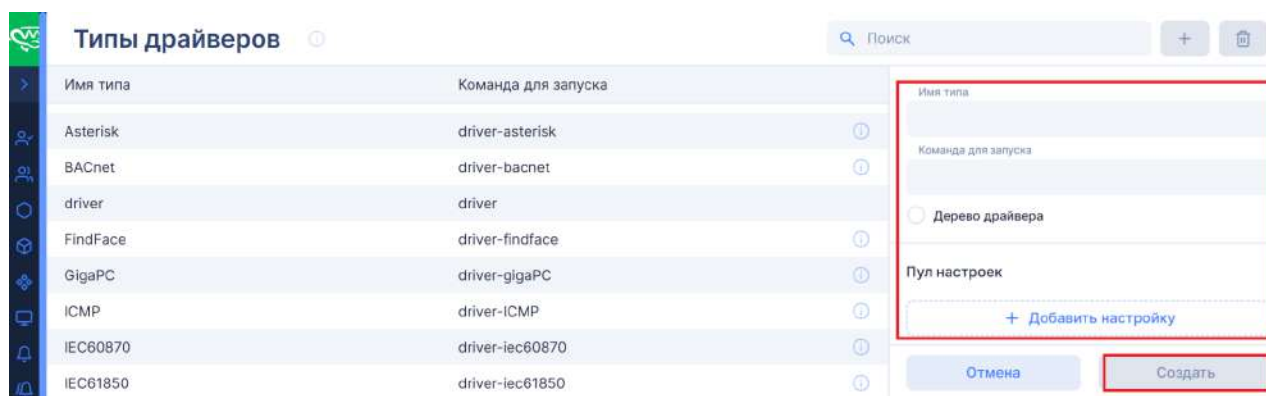


Рис. 3.141. Параметры созданного типа драйвера

После создания новый тип драйвера отобразится в списке (рис. 3.142).

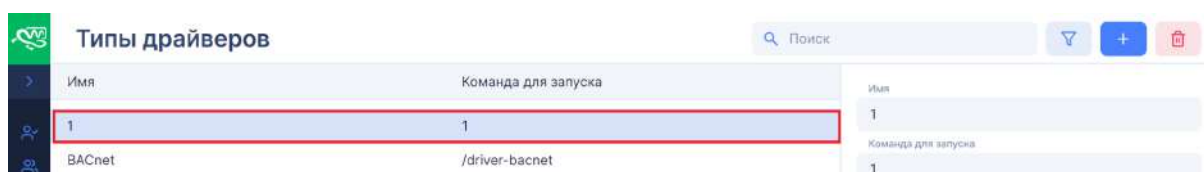


Рис. 3.142.Список Типы драйверов

Для того чтобы открыть пул настроек необходимо нажать на «+ Добавить настройку» в боковой вкладке с параметрами рис. 3.143).



Рис. 3.143. Открытие пула настроек

Далее откроется окно, в котором следует ввести параметры (имя и описание) и выбрать тип данных (рис. 3.144).

Пул настроек

The screenshot shows a dialog box titled 'Настройка' (Setting) with a close button. It contains three fields: 'Имя' (Name) with a text input, 'Тип' (Type) with a dropdown menu showing 'Выберите значение' (Select value), and 'Описание' (Description) with a text input.

Рис.3.144. Пул настроек

Для выбора доступны следующие типы данных (рис. 3.145).

Рис. 3.145. Доступные типы данных

При типе Enum можно создать массив значений с именами. Для этого необходимо нажать на «+Добавить объект», после чего задать имя и значение (рис. 3.146).

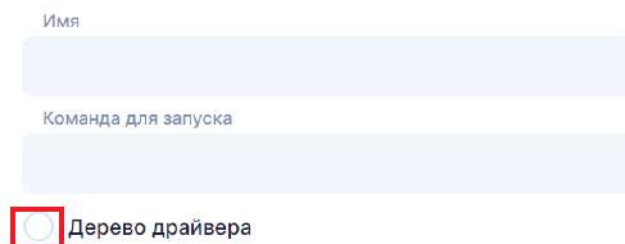
Рис. 3.146. Создание массива

Имя каждого значения должно быть уникальным, иначе появится предупреждение. Предупреждение также появится, если пропустить ввод одного из параметров (рис. 3.147).

Рис. 3.147. Предупреждения

Для удаления значения следует воспользоваться соответствующей кнопкой с пиктограммой корзины (рис. 3. 147).

При создании типа драйвера также можно задать возможность драйвера обладать деревом, поставив галочку напротив соответствующего параметра (рис. 3.145).



Имя

Команда для запуска

☒ Дерево драйвера

Рис. 3.145. Дерево драйвера

Для того чтобы удалить тип драйвера, необходимо его выбрать и затем нажать на соответствующую кнопку в правом верхнем углу (рис. 3.146).

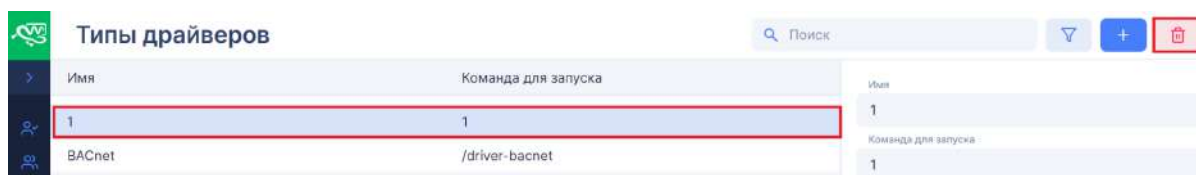


Рис. 3.146. Удаление типа драйвера

После чего в всплывающем окне подтвердить удаление (рис. 3.147).

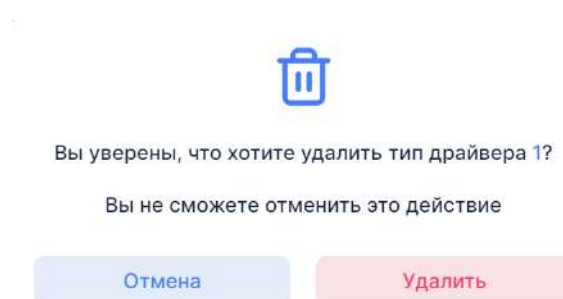


Рис. 3.147. Подтверждение удаления

Если в Платформе существует драйвер данного типа, то он также будет удалён (рис. 3.148).

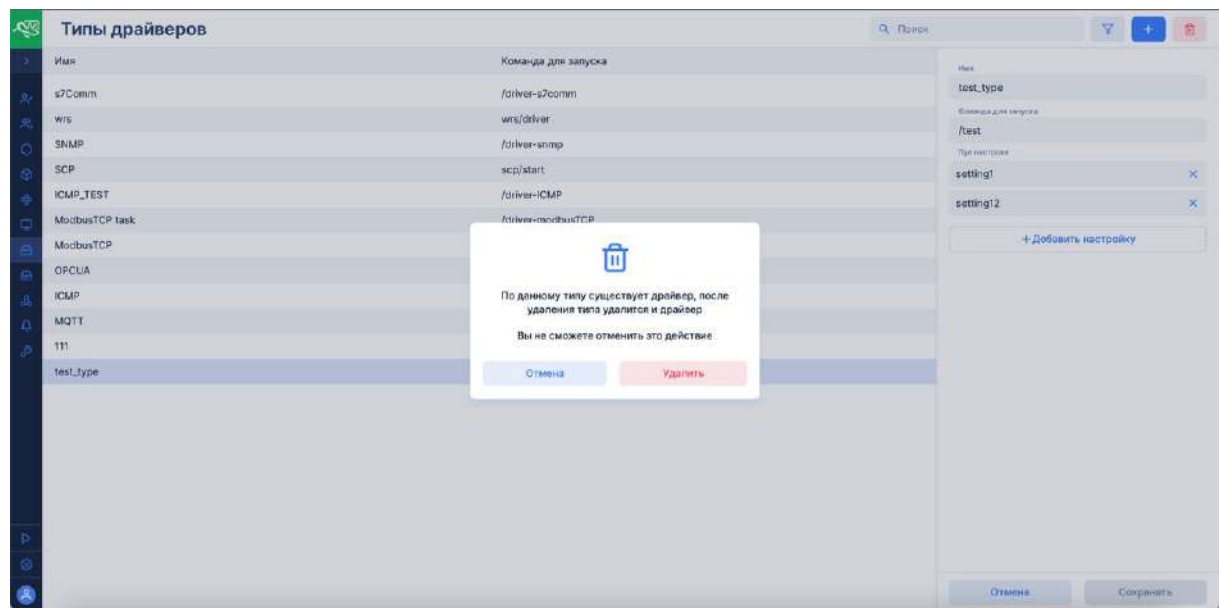


Рис. 3.148. Проверка наличия созданных драйверов данного типа

Некоторые типы драйвера могут иметь ограничение на изменение и удаление. Об этом символизирует соответствующий значок при наведении на который выводится предупреждение. Подобное предупреждение также отображается и на вкладке с параметрами при нажатии на тип драйвера (рис. 3.149).

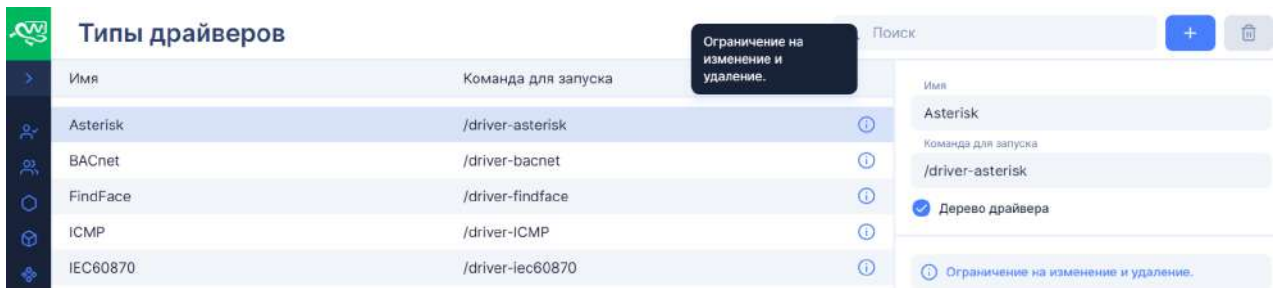


Рис. 3.149. Предупреждение об ограничениях

Для того чтобы добавить новый драйвер, необходимо нажать на кнопку «Добавить» (рис. 3.150).

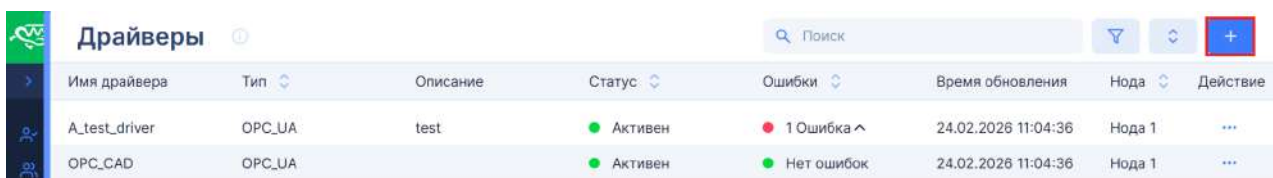


Рис. 3.150. Кнопка добавления нового драйвера

После чего откроется вкладка с параметрами, в которой необходимо ввести имя драйвера, его описание, выбрать тип драйвера и узел (ноду), на котором драйвер запускается по умолчанию. По завершении нажать кнопку «Создать» (рис. 3.151).

Драйверы

Поиск

Имя драйвера	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
A_test_driver	OPC-UA	test	Активен	1 Ошибка	24.02.2026 11:04:36	Нода 1	...
OPC_CAD	OPC-UA		Активен	Нет ошибок	24.02.2026 11:04:36	Нода 1	...

Имя драйвера

Описание

Нода по умолчанию

Выберите значение

☐ Использовать ноду по умолчанию

Тип

Выберите значение

Отмена Создать

Рис. 3.151. Добавление драйвера

После создания новый драйвер отобразится в списке (рис. 3.152).

Драйверы

Поиск

Имя драйвера	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
A_test_driver	OPC-UA	test	Активен	1 Ошибка ^	24.02.2026 11:04:36	Нода 1	...
OPC_CAD	OPC-UA		Активен	Нет ошибок	24.02.2026 11:04:36	Нода 1	...
Demo	ICMP	01	Выключен	Нет ошибок			...

Рис. 3.152. Список драйверов

Во вкладке с параметрами драйвера имеется чекбокс «Использовать ноду по умолчанию» (рис. 3.153), тем самым, если указана нода 1, но драйвер по какой-либо причине был запущен на ноде 2, то при активации данного чекбокса в момент, когда появится возможность запустить драйвер на 1 ноде, он будет перезапущен на ней.

Имя драйвера

Описание

Нода по умолчанию

Выберите значение

☐ Использовать ноду по умолчанию

Тип

Выберите значение

Рис. 3.153. Чекбокс «Использовать ноду по умолчанию»

При выборе соответствующего типа (протокола) драйвера необходимо задать настройки (пул настроек) связанные с этим типом драйвера (протокола, рис. 3.154).

Драйверы

Имя драйвера	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
A_test_driver	OPC_UA	test	Активен	1 Ошибка	24.02.2026 1	Нода 1	...
OPC_CAD	OPC_UA		Активен	Нет ошибок	24.02.2026 1	Нода 1	...
Demo	ICMP	01	Выключен	Нет ошибок			...

Имя драйвера: OPC_CAD

Описание:

Нода по умолчанию: 1

☒ Использовать ноду по умолчанию

Тип: OPC_UA

Пул настроек

server_url: opc.tcp://185.221.152.176:4840

security_policy: Выберите значение

security_mode: Выберите значение

auth_mode: Выберите значение

auto_sec_policy:

Отмена Сохранить

Рис. 3.154. Пул настроек драйвера

Каждый параметр соответствует заданному в пуле настроек Типа драйвера. Доступные значения зависят от выбранного Типа данных. Например, для Boolean - true/false, а для Enum заданный вручную массив значений с именами.

В пуле настроек доступен вывод описания каждой из настроек. Для вывода описания необходимо навести курсор на соответствующую пиктограмму (рис. 3.155).

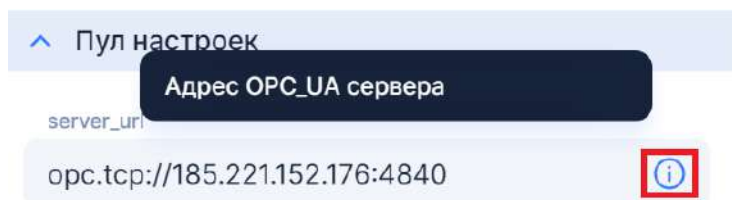


Рис. 3.155. Вывод описания настроек

При наличии ошибок в работе драйвера они отобразятся в виде выпадающего списка в таблице. Для того чтобы раскрыть список, необходимо воспользоваться соответствующей кнопкой (рис. 3.156).

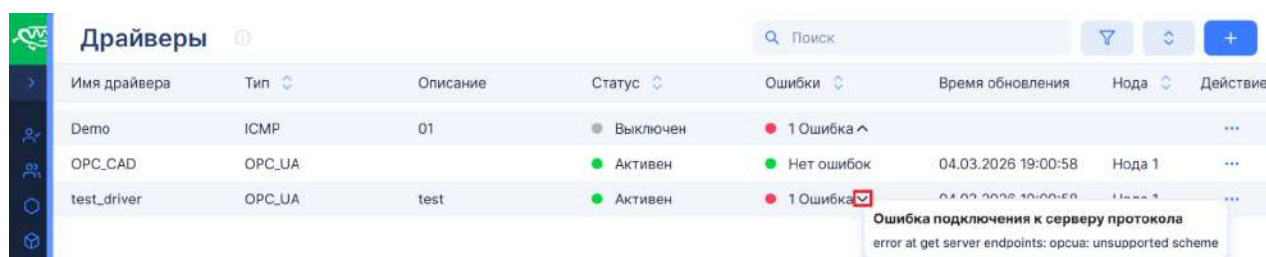


Рис. 3.156. Список ошибок

Для того чтобы удалить драйвер, необходимо выбрать драйвера и нажать на соответствующую кнопку в правом верхнем углу страницы (рис. 3.157).

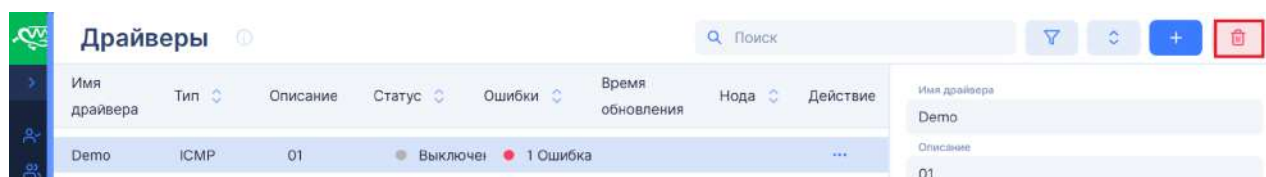


Рис. 3.157. Кнопка удаления

После чего во всплывающем окне подтвердить удаление (рис. 3.158).

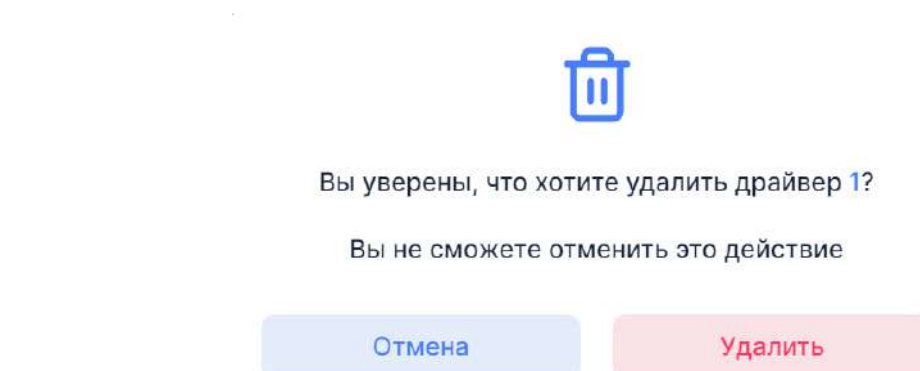


Рис. 3.158. Подтверждение удаления

Для конфигурации драйвера необходимо выбрать его из списка, после чего откроется вкладка с параметрами, в которой можно их изменить. По завершении нажать «Сохранить» (рис. 3.159).

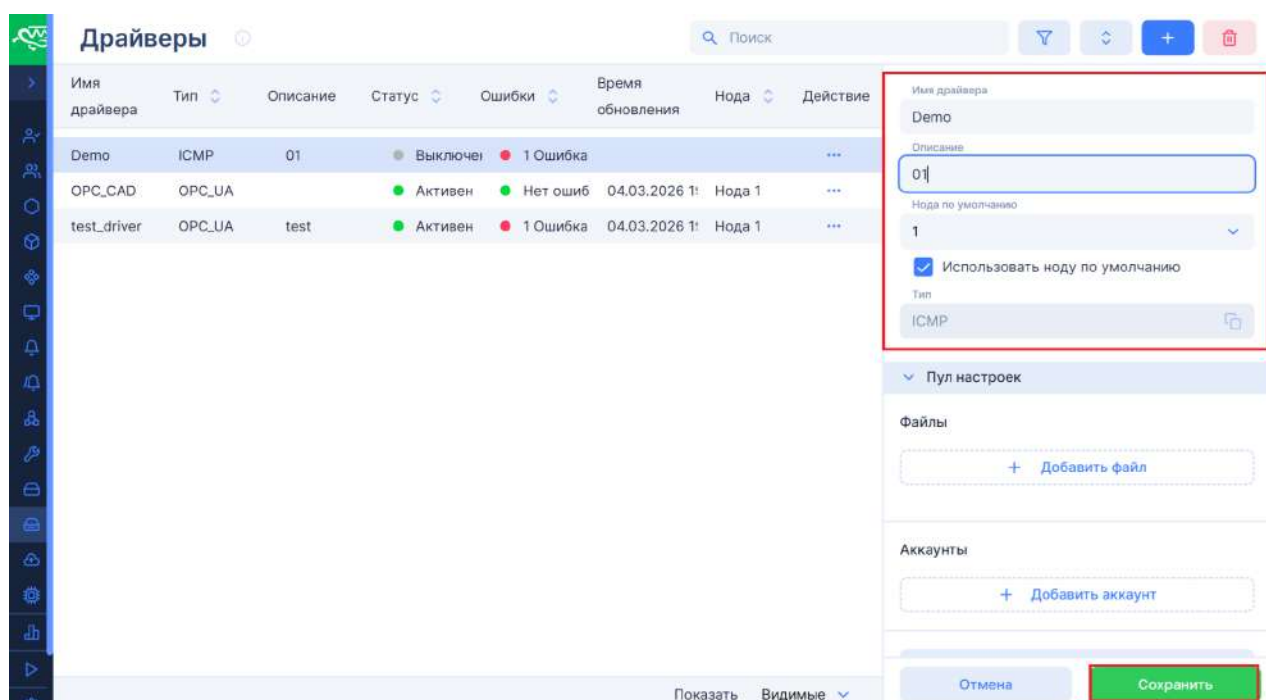


Рис. 3.159. Конфигурация драйвера

3.6.3. Драйвер ICMP

ICMP (Internet Control Message Protocol, протокол межсетевых управляющих сообщений) – это сетевой протокол, входящий в стек протоколов TCP/IP, который используется для передачи сообщений об ошибках и диагностической информации в IP-сетях.

ICMP предназначен для обмена информацией о состоянии сети, диагностики сетевых проблем и управления сетевыми операциями. Он не используется для передачи пользовательских данных, а служит для поддержки работы IP-протокола, обеспечивая обратную связь о проблемах в сети.

Основные функции протокола ICMP:

- Сообщения об ошибках;
- Диагностика сети;
- Управление сетевыми ресурсами;
- Поддержка других протоколов.

Для настройки драйвера ICMP нужно выбрать его из списка и затем задать его параметры (рис. 3.139).

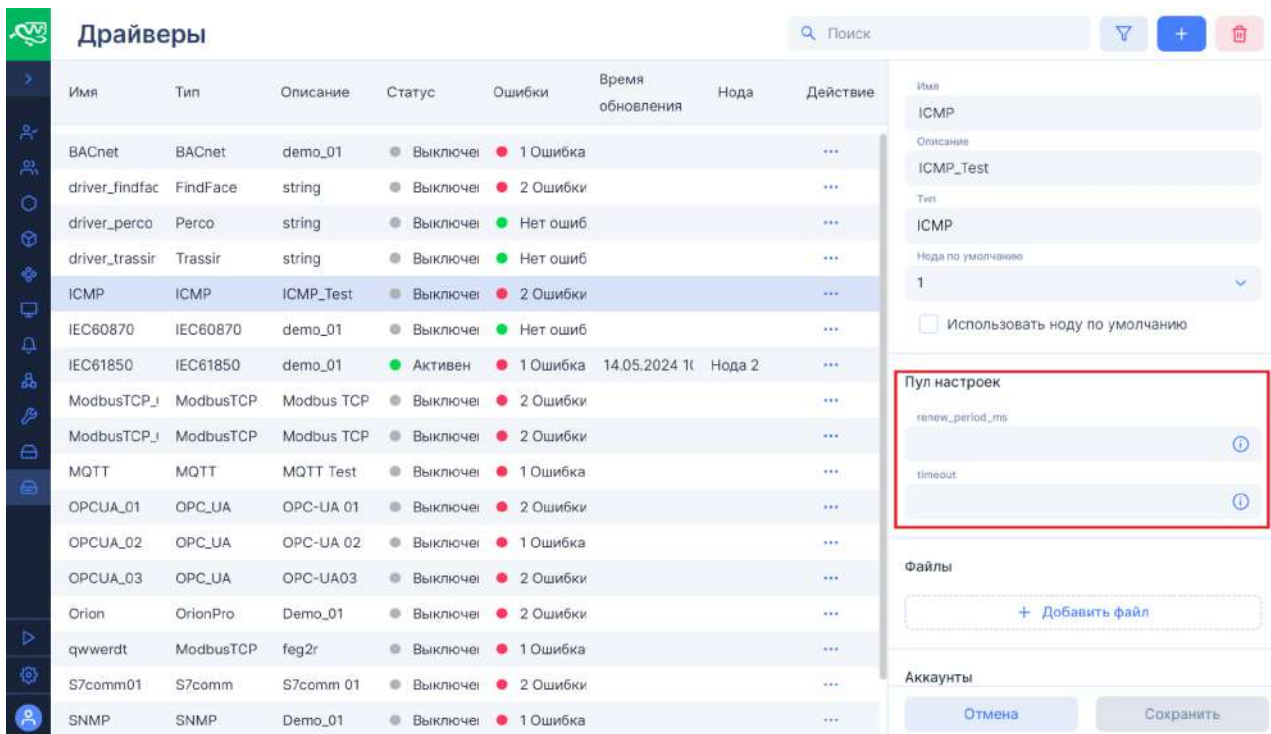


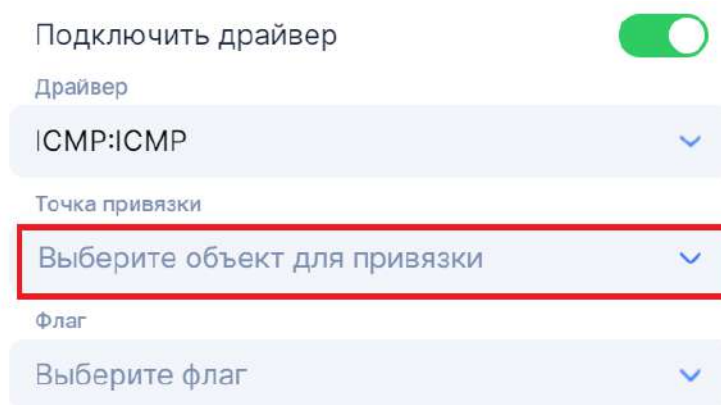
Рис. 3.139. Настройка драйвера ICMP

Настройки драйвера ICMP:

- `renew_period_ms` - период обновления (опроса) системы;
- `timeout` - время в миллисекундах в течение которого драйвер должен получить ответ от диагностируемого устройства.

По завершении нажать «Сохранить».

Для привязки объекта к соответствующему драйверу, необходимо перейти во вкладку объекты, выбрать драйвер для целевого объекта в соответствии с п.3.3.2 и задать точку привязки (Либо выбрать из списка статусов, рис. 3.140).



Подключить драйвер ☒

Драйвер

ICMP:ICMP

Точка привязки

Выберите объект для привязки

Флаг

Выберите флаг

Рис. 3.140. Точка привязки драйвера ICMP

Точкой привязки для ICMP является IP адрес диагностируемого устройства.

3.6.4. Драйвер SNMP

SNMP (Simple Network Management Protocol, простой протокол сетевого управления) — это сетевой протокол прикладного уровня из стека TCP/IP, предназначенный для мониторинга, управления и диагностики устройств в IP-сетях.

SNMP используется для сбора информации о состоянии сетевых устройств (маршрутизаторов, коммутаторов, серверов, принтеров и т.д.), управления их конфигурацией и уведомления о сбоях или событиях. Он обеспечивает централизованное управление сетевой инфраструктурой, упрощая администрирование и мониторинг.

Основные функции протокола SNMP:

- Мониторинг состояния устройств;
- Управление конфигурацией;
- Оповещение о событиях;
- Сбор статистики;
- Диагностика сети.

Для настройки драйвера SNMP нужно выбрать его из списка и затем задать параметры драйвера (рис. 3.141).

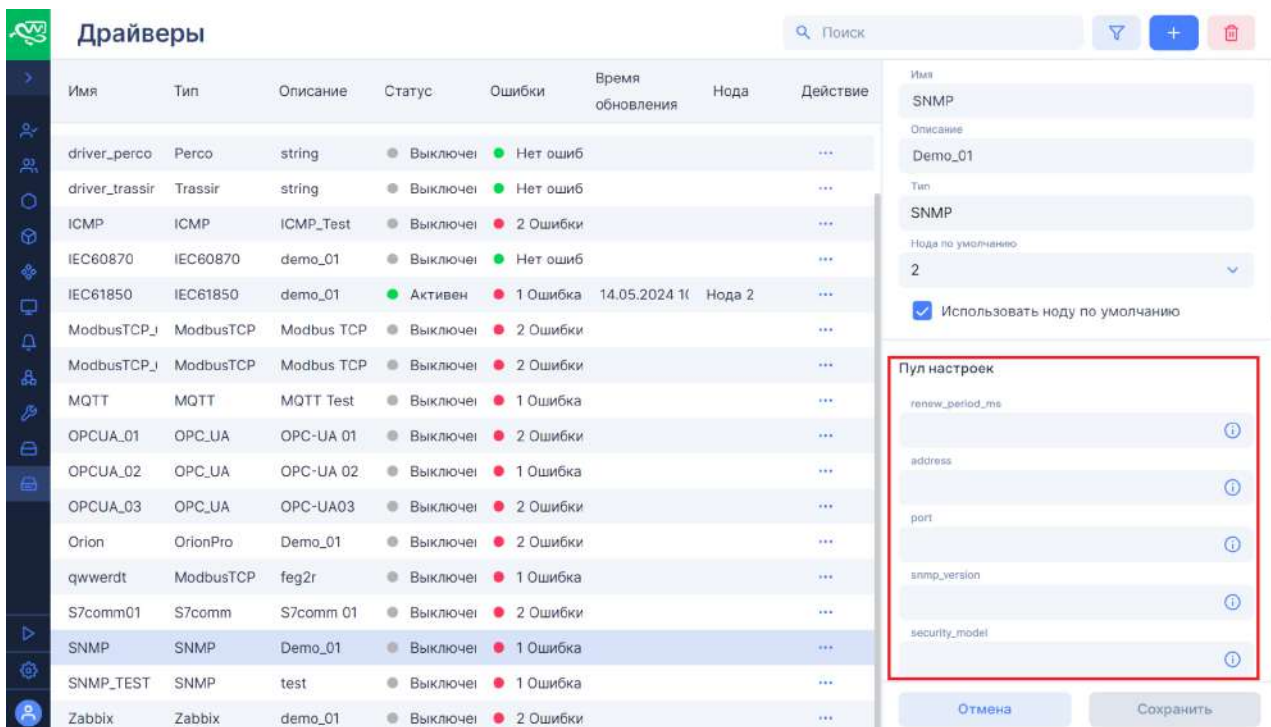


Рис. 3.141. Настройка драйвера SNMP

Настройки драйвера SNMP:

- renew_period_ms - период обновления (опроса) системы;
- address - сетевой адрес подключаемого устройства;
- port - порт подключения;
- snmp_version - версия SNMP;
 - v1;
 - v2c;
 - v3
- security_model - модель безопасности;
 - SNMP_V3_SECURITY_MODEL
- msg_flags - используются только для SNMP v3;
 - NO_AUTH_NO_PRIV (No authentication, and no privacy);
 - AUTH_NO_PRIV (Authentication and no privacy);
 - AUTH_PRIV (Authentication and privacy);
 - REPORTABLE
- user_name - имя пользователя;
- auth_protocol - протокол аутентификации;
 - NO_AUTH;
 - MD5;
 - SHA;
 - SHA224;
 - SHA256;
 - SHA384;
 - SHA512
- auth_passwd - пароль пользователя;
- privacy_protocol - протокол конфиденциальности;
 - NoPriv;
 - DES;
 - AES;
 - AES192;
 - AES256;
 - AES192C;
 - AES256C
- privacy_passwd - пароль конфиденциальности;
- mibs_path - путь к директории, содержащей MIBS-файлы. в случае нескольких директорий, их следует указывать через разделитель ':'. Пример: /dir1/dir2:/dir3/dir4:/dir5/dir6;
- trap_listener_port - порт мониторинга ловушек;
- root_oid - корневой OID для драйвера.

По завершении нажать «Сохранить»..

Для привязки объекта к драйверу необходимо перейти во вкладку объекты, выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.142).

Подключить драйвер ☒

Драйвер
SNMP:SNMP

Точка привязки
Выберите объект для привязки

Флаг
Выберите флаг

Рис. 3.142. Кнопка точки привязки

Откроется окно с деревом тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег (рис. 3.143).

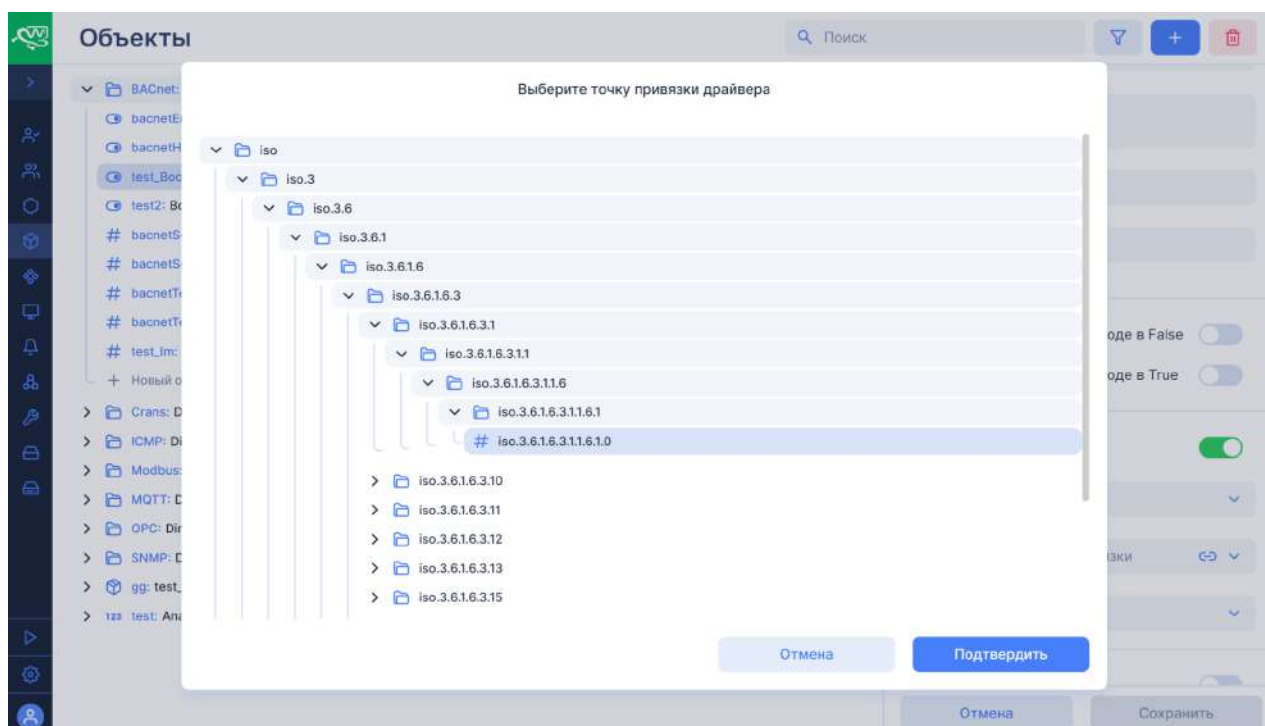


Рис. 3.143. Дерево тегов

3.6.5. Драйвер OPC UA

OPC UA (Open Platform Communications Unified Architecture, унифицированная архитектура открытых платформ связи) — это промышленный протокол обмена данными, разработанный для обеспечения надежного, безопасного и платформонезависимого взаимодействия между устройствами и системами в автоматизированных промышленных системах.

OPC UA предназначен для стандартизации и упрощения обмена данными между различными устройствами, датчиками, контроллерами и программным обеспечением в промышленной автоматизации. Он обеспечивает интероперабельность в гетерогенных системах, включая производство, энергетику, транспорт и другие отрасли, поддерживая как локальные, так и облачные решения.

Основные функции протокола OPC UA:

- Обмен данными;
- Интероперабельность;
- Безопасность;
- Моделирование данных;
- Подписка и уведомления;
- Масштабируемость и гибкость;
- Исторический доступ и аналитика.

Для настройки драйвера OPC UA, нужно выбрать его из списка и затем задать его параметры (рис. 3.144).

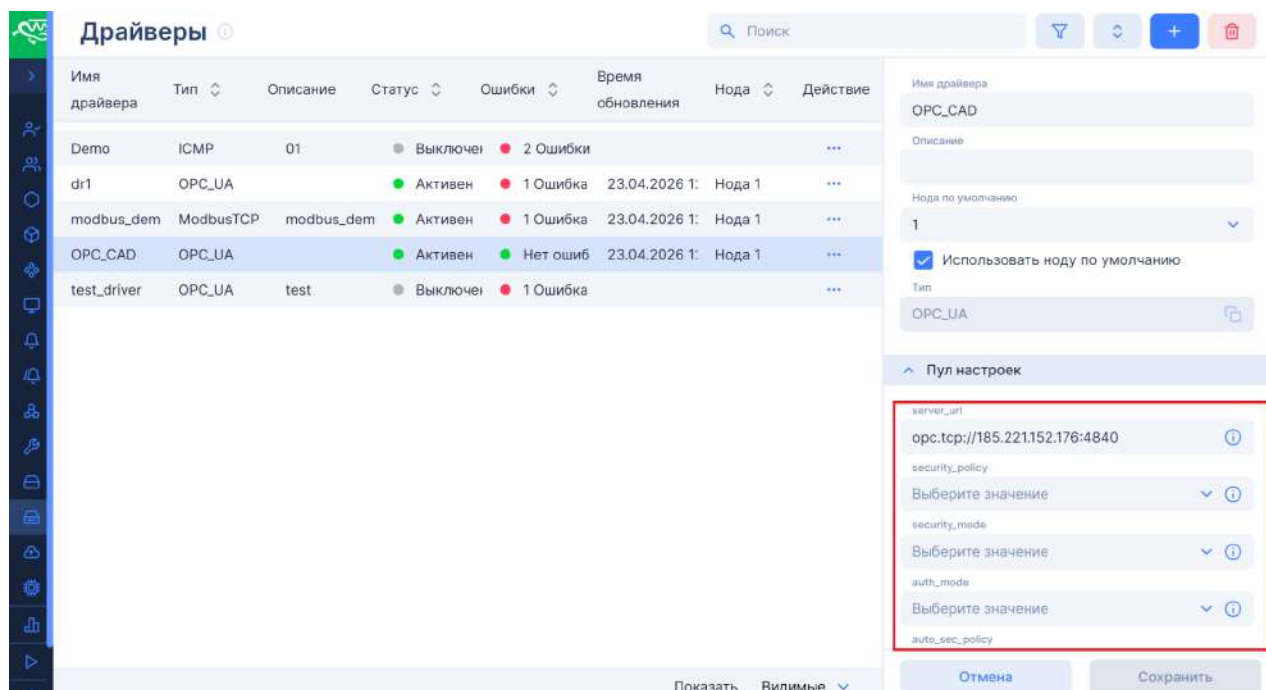


Рис. 3.144. Настройки драйвера OPC UA

Настройки драйвера OPC UA:

- server_url - адрес OPC-UA сервера;
- security_policy - политика безопасности;

NONE;

Basic128Rsa15;

Basic256;

Basic256Sha256;

Aes128_Sha256_RsaOaep;

Aes256_Sha256_RsaPss;

Все кроме NONE требуют клиентского сертификата

- security_mode - способ защиты сообщений;

NONE;

SIGN;

SIGN_AND_ENCRYPT

- auth_mode - метод аутентификации;
ANONYMOUS;
USERNAME;
CERTIFICATE
- auto_sec_policy - автовыбор политики безопасности (будет выбран не ниже указанного в security_policy);
- auto_security_mode - автовыбор способа защиты сообщений (будет выбран не ниже указанного в security_mode);
- username- имя пользователя;
Требуется, если auth_mode=USERNAME
- password - пароль пользователя;
Требуется, если auth_mode=USERNAME
- client_cert_file_name - имя файла клиентского сертификата;
Сам файл должен быть загружен к драйверу с соответствующим именем;
Требуется для любого security_policy сильнее чем NONE
- client_key_file_name - имя файла клиентского ключа;
Сам файл должен быть загружен к драйверу с соответствующим именем;
Требуется для любого security_policy сильнее чем NONE
- user_cert_file_name;
Имя файла пользовательского сертификата;
Сам файл должен быть загружен к драйверу с соответствующим именем;
Требуется если auth_mode=CERTIFICATE
- user_key_file_name;
Имя файла пользовательского ключа;
Сам файл должен быть загружен к драйверу с соответствующим именем;
Требуется если auth_mode=CERTIFICATE.
По завершении нажать «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.145).

Подключить драйвер ☒

Драйвер
OPCUA_01:OPC-UA

Точка привязки
Выберите объект для привязки  

Флаг
Выберите флаг

Рис. 3.145. Кнопка точки привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег (рис. 3.146) и нажать «Подтвердить».

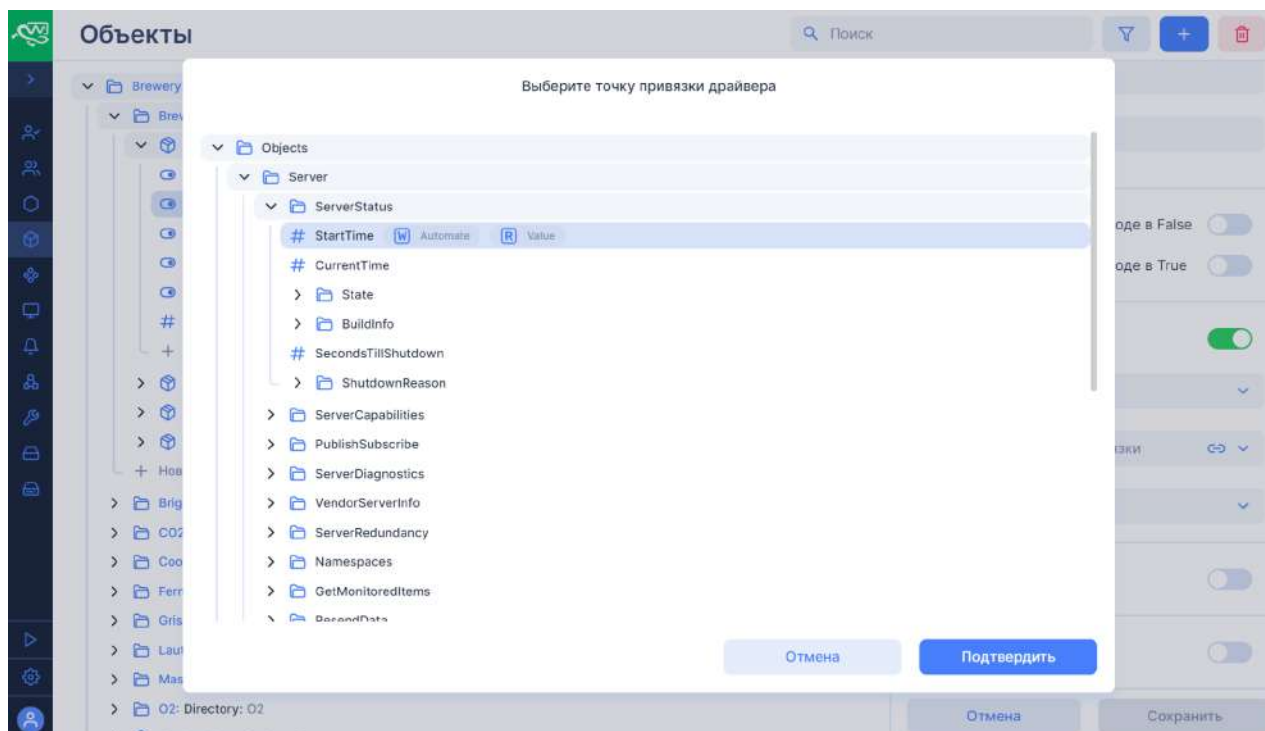


Рис. 3.146. Дерево тегов

3.6.6. Драйвер Modbus TCP

Modbus TCP – это открытый протокол, основанный на клиент-серверной модели, который позволяет устройствам (например, ПЛК, датчикам, HMI-панелям) обмениваться данными через сетевой интерфейс с использованием протокола TCP/IP на порту 502. Он сочетает простоту протокола Modbus с возможностями сетей Ethernet.

Протокол Modbus TCP предназначен для обмена данными между устройствами в промышленных системах автоматизации по сетям TCP/IP, таких как Ethernet. Он используется для передачи данных между управляющими устройствами (например, ПЛК) и датчиками, исполнительными механизмами или другими устройствами, обеспечивая надежную и стандартизированную коммуникацию в реальном времени.

Основные функции протокола Modbus TCP:

- Передача данных;
- Клиент-серверная архитектура;
- Простота и универсальность;
- Надежность передачи;
- Функциональные коды;
- Интеграция с Ethernet.

Для настройки драйвера **Modbus TCP** нужно выбрать его из списка и затем задать его параметры (рис. 3.147).

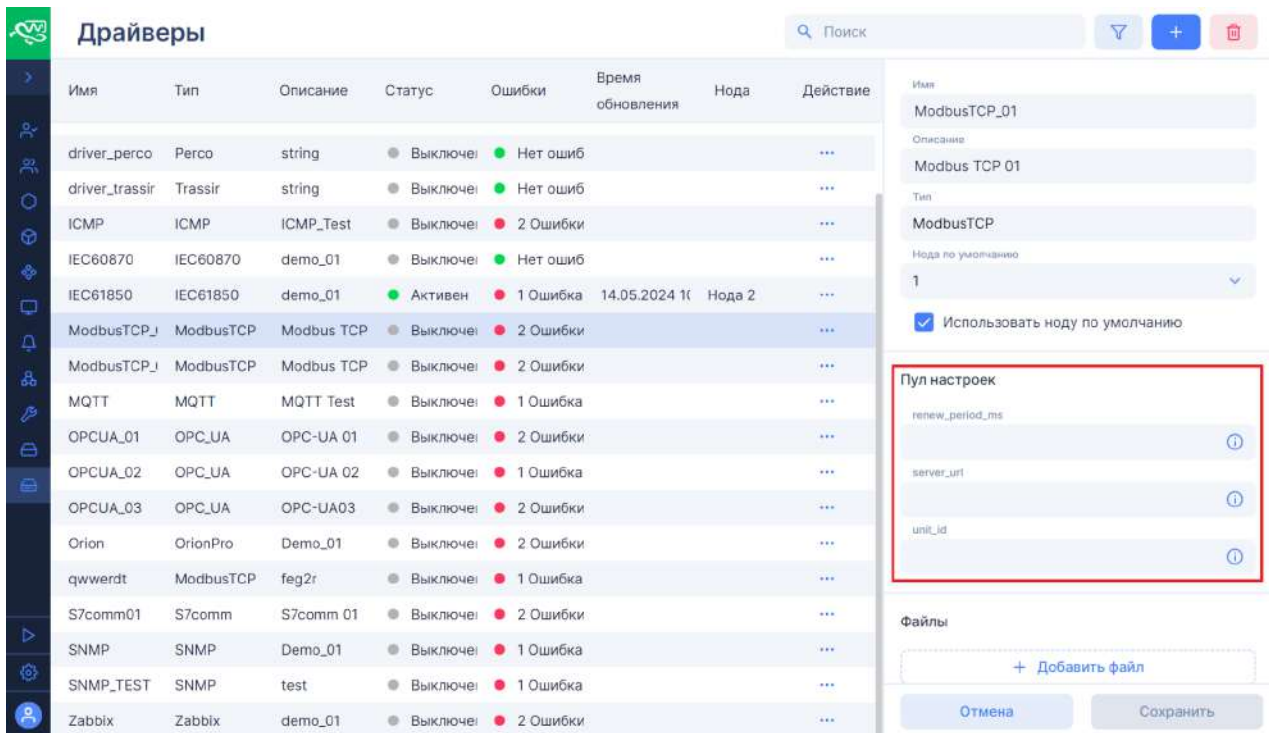


Рис. 3.147. Настройки драйвера

Настройки драйвера Modbus TCP:

- `renew_period_ms` - период обновления (опроса) системы;
- `server_url` - сетевой адрес подключаемого устройства (`tcp://<addr>:<port>`);
- `unit_id` - адрес устройства.

По завершении нажать на кнопку «Сохранить» .

Для того чтобы привязать объект к драйверу, необходимо перейти во вкладку объекты, выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки (Либо выбрать из списка статусов, рис. 3.148).

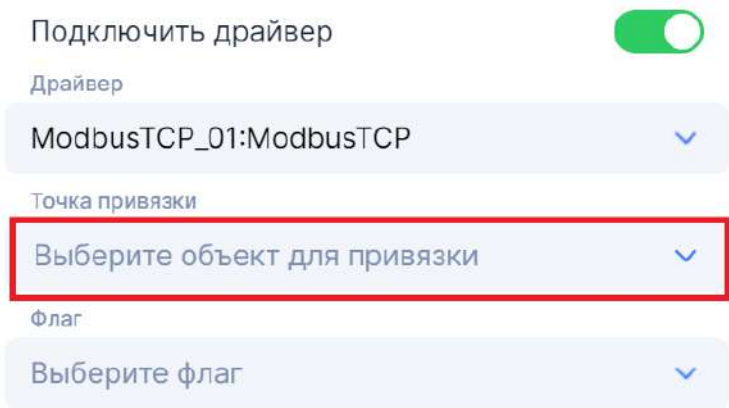


Рис. 3.148. Точка привязки

Формат определения точки привязки:

address; val_type; register_type; encoding[:len], где

- address - адрес регистра;
- val_type - тип значения
 - BYTES = 0
 - COIL = 1
 - COILS = 2
 - DISCRETE_INPUT = 3
 - DISCRETE_INPUTS = 4
 - FLOAT32 = 5
 - FLOAT64 = 6
 - RAW_BYTES = 7
 - REGISTER = 8
 - REGISTERS = 9
 - UINT32 = 10
 - UINT64 = 11
 - UINT16 = 12
 - INT16 = 13
 - INT32 = 14
 - INT64 = 15
- register_type - тип регистра
 - INPUT_REGISTER = 1
 - HOLDING_REGISTER = 2
- encoding - кодирование (порядок байт):
 - BIG_ENDIAN = 1
 - LITTLE_ENDIAN = 2
 - BIG_ENDIAN_BYTE_SWAP = 3
 - LITTLE_ENDIAN_BYTE_SWAP = 4
- len - длина, используется только для множественных типов (массив байтов, и т.д.).

3.6.7. Драйвер Profinet (S7Comm)

Profinet — это открытый промышленный Ethernet-протокол, разработанный организацией PROFIBUS & PROFINET International (PI) для обмена данными в системах автоматизации. Он основан на стандарте Ethernet и предназначен для высокоскоростного, детерминированного обмена данными между устройствами в промышленных сетях.

Profinet поддерживает два основных режима:

1. **Profinet CBA** (Component Based Automation): для обмена данными между системами на уровне компонентов;

2. **Profinet IO:** для связи между контроллерами (ПЛК) и периферийными устройствами (датчиками, приводами).

Протокол Profinet (Process Field Net) предназначен для промышленной автоматизации и обмена данными в реальном времени между устройствами в производственных системах по сетям Ethernet. Он используется для интеграции полевых устройств (датчиков, исполнительных механизмов, приводов) с системами управления (ПЛК, SCADA).

Основные функции протокола Profinet:

- Передача данных в реальном времени;
- Интеграция с IT-системами;
- Гибкая топология сети;
- Диагностика и мониторинг;
- Модульность и масштабируемость;
- Безопасность;
- Поддержка различных приложений.

Для настройки драйвера **S7Comm** нужно выбрать его из списка и затем задать его параметры (рис. 3.149).

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
driver_perco	Perco	string	Выключен	Нет ошибок			...
driver_trassir	Trassir	string	Выключен	Нет ошибок			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошибок			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA 03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...
SNMP_TEST	SNMP	test	Выключен	1 Ошибка			...
Zabbix	Zabbix	demo_01	Выключен	2 Ошибки			...

Пул настроек

- renew_period_ms
- server_ip
- rack
- slot

Файлы

Отмена Сохранить

Рис. 3.149. Настройки драйвера

Настройки драйвера S7Comm:

- renew_period_ms - период обновления значений в формате 1d1h1m1s1ms;
- server_ip - адрес S7comm контроллера;
- rack - номер модуля в корзине;
- slot - номер модуля в слоте.

По завершении нажать на кнопку «Сохранить»

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта (см п.3.3.2) и задать точку привязки (Либо выбрать из списка статусов, рис. 3.150).

Подключить драйвер ☒

Драйвер
S7comm01:S7comm

Точка привязки
Выберите объект для привязки

Флаг
Выберите флаг

Рис. 3.150. Точка привязки драйвера S7comm

Точка привязки задаётся в следующем формате (без пробелов):

addr; start; typeValue; typeRegistor; bitmask(for bool), где:

- addr - адрес региона данных (для DB - 1, для остальных - 0);
- start - адрес регистра;
- typeValue - тип значения (0-Bool, 1-Byte, 2-Real, 3-LReal, 4-Int, 5-DInt, 6-Word);
- typeRegistor - тип регистра (type 0-ltags, 1 - Qtags, 2 - DataBlock);
- bitmask (for bool) - номер бита для данных типа BoolD.

3.6.8. Драйвер MQTT

MQTT (Message Queuing Telemetry Transport) – это легковесный протокол обмена сообщениями, основанный на модели «публикатор-подписчик» (publish/subscribe), предназначенный для передачи данных в условиях ограниченной пропускной способности, высокой задержки или ненадежных сетей. Он работает поверх TCP/IP и широко используется в системах Интернета вещей (IoT) и промышленной автоматизации.

Протокол MQTT (Message Queuing Telemetry Transport) предназначен для эффективного и надежного обмена данными между устройствами в системах с ограниченной пропускной способностью, высокой задержкой или ненадежными сетями, таких как Интернет вещей (IoT) и промышленная автоматизация.

Основные функции протокола MQTT:

- Модель "публикатор-подписчик": Устройства публикуют сообщения в определенные темы (topics), а другие устройства подписываются на интересующие их темы для получения данных. Брокер (сервер) управляет маршрутизацией сообщений;
- Легковесность: Минимизирует объем передаваемых данных, что подходит для устройств с ограниченными ресурсами (например, микроконтроллеры);
- Асинхронная связь: Позволяет устройствам работать независимо, не требуя постоянного соединения;
- Удержание сообщений (Retained Messages): Брокер может сохранять последнее сообщение по теме для новых подписчиков;
- Последняя воля (Last Will and Testament): Устройство может настроить сообщение, которое брокер отправит в случае его отключения;
- Безопасность: Поддерживает шифрование (TLS/SSL) и аутентификацию для защиты данных.

Для настройки драйвера MQTT, нужно выбрать его из списка и затем задать его параметры (рис. 3.151).

The screenshot shows the 'Драйверы' (Drivers) management interface. On the left is a sidebar with navigation icons. The main area displays a table of drivers. The 'MQTT' driver is selected, and its configuration panel is open on the right. The configuration panel includes fields for 'broker_addr', 'broker_port', 'username', and 'password', each with a help icon. A checkbox 'Использовать ноду по умолчанию' (Use default node) is checked. At the bottom of the panel are 'Отмена' (Cancel) and 'Сохранить' (Save) buttons.

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
BACnet	BACnet	demo_01	Выключен	1 Ошибка			...
driver_findfac	FindFace	string	Выключен	2 Ошибки			...
driver_perco	Perco	string	Выключен	Нет ошиб			...
driver_trassir	Trassir	string	Выключен	Нет ошиб			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошиб			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...

Панель настроек MQTT:

- Имя: MQTT
- Описание: MQTT Test
- Тип: MQTT
- Нода по умолчанию: 1
- ☒ Использовать ноду по умолчанию
- broker_addr
- broker_port
- username
- password

Кнопки: Отмена, Сохранить

Рис. 3.151. Настройка драйвера MQTT

Настройки драйвера MQTT:

- broker_addr - сетевой адрес подключаемого устройства;
- broker_port - порт подключения;
- username - имя пользователя;
- password - пароль пользователя.

По завершении нажать на кнопку «Сохранить».

Для того чтобы привязать объект к драйверу, необходимо перейти во вкладку объекты, выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки (Либо выбрать из списка статусов, рис. 3.152).

Подключить драйвер ☒

Драйвер

MQTT:MQTT

Точка привязки

Выберите объект для привязки

Флаг

Выберите флаг

Рис. 3.152. Точка привязки

Точка привязки задается в следующем формате (без пробелов):

topic; valueName; valueType, где:

- *topic* - (топик, тема) канал отправки и получения сообщений между устройствами, реестрами и клиентами брокера;
- *valueName* - имя переменной в топике;
- *valueType* - тип значения (0-Bool, 1-Float, 2-String).

Пример:

device1/temperature;Temperature;0

3.6.9. Драйвер BACnet

BACnet (Building Automation and Control Network) — это стандартизированный коммуникационный протокол, разработанный специально для систем автоматизации зданий. Он обеспечивает взаимодействие между устройствами различных производителей, используемых для управления инженерными системами зданий, такими как отопление, вентиляция, кондиционирование (HVAC), освещение, системы безопасности и другие.

BACnet предназначен для интеграции и управления оборудованием в системах автоматизации зданий (BMS — Building Management Systems), обеспечивая их совместимость, централизованный мониторинг и контроль.

Основные функции протокола BACnet:

- Интероперабельность: Позволяет устройствам разных производителей (например, термостаты, контроллеры, датчики) обмениваться данными по единому стандарту;
- Объектно-ориентированная модель;
- Гибкость транспортных протоколов: Поддерживает различные среды передачи данных;
- Сервисы протокола;
- Масштабируемость;
- Безопасность;
- Диагностика и мониторинг.

Для настройки драйвера BACnet нужно выбрать его из списка и затем задать его параметры (рис. 3.153).

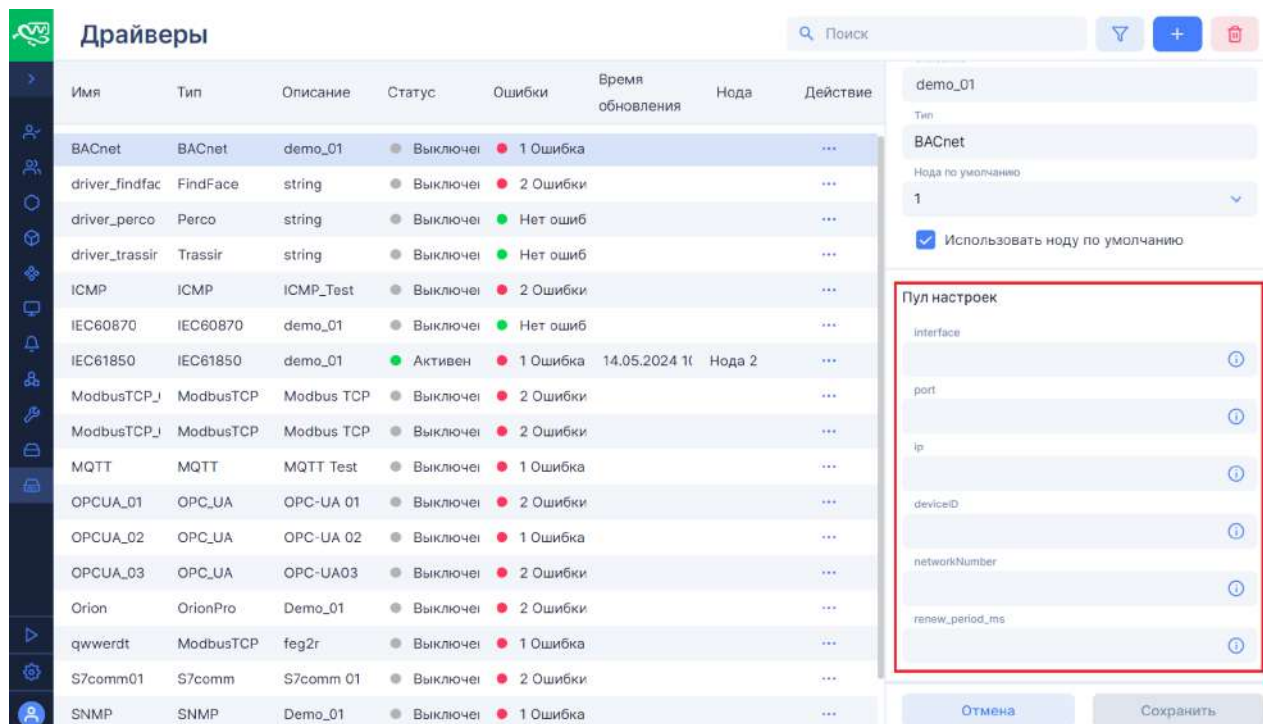


Рис. 3.153. Настройки драйвера

Настройки драйвера BACnet:

- interface - название физического интерфейса;
- port - порт подключения к устройству для подключения данных;
- ip - адрес контроллера;
- deviceId - идентификатор устройства BACnet;
- networkNumber - номер сети уникально идентифицирует сеть в системе BACnet;
- renew_period_ms - период обновления значений в формате 1d1h1m1s1ms.

По завершении нажать на кнопку «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.154).

Подключить драйвер ☒

Драйвер

BACnet:BACnet

Точка привязки

Выберите объект для привязки

Флаг

Выберите флаг

Рис. 3. 154. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.155).

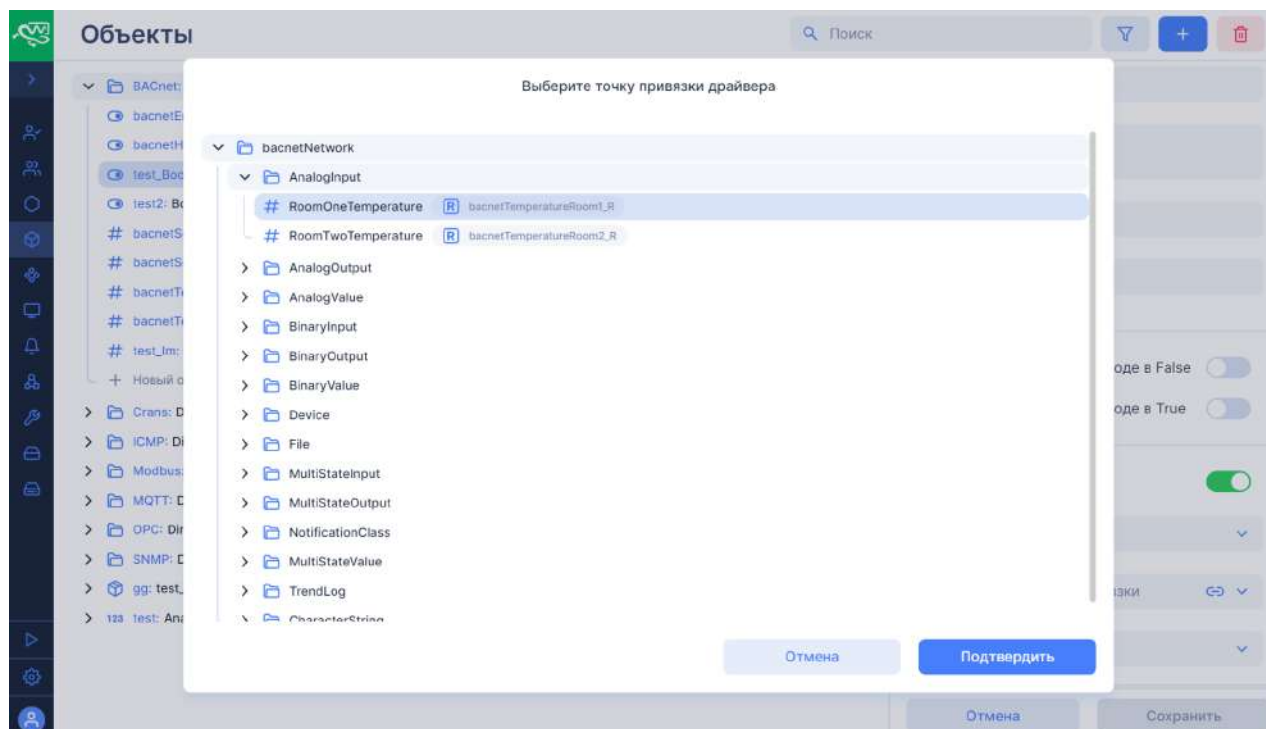


Рис. 3.155. Дерево тегов

3.6.10. Драйвер IEC 61850

IEC 61850 — это международный стандарт, разработанный Международной электротехнической комиссией (IEC), для проектирования и эксплуатации систем автоматизации подстанций (Substation Automation Systems, SAS). Он определяет стандартизированные модели данных, коммуникационные протоколы и методы конфигурации для обмена данными между интеллектуальными электронными устройствами (IED — Intelligent Electronic Devices) в электроэнергетических системах.

IEC 61850 предназначен для обеспечения интероперабельности, надежности и гибкости в системах автоматизации подстанций и других объектов электроэнергетики (например, распределительных сетей, возобновляемых источников энергии). Он используется для унификации обмена данными между устройствами разных производителей, упрощения проектирования, эксплуатации и обслуживания систем, а также повышения их эффективности и безопасности.

Основные функции протокола:

- Стандартизированные модели данных;
- Коммуникационные сервисы;
- Интероперабельность;
- Высокая скорость и надежность;
- Гибкость и масштабируемость;
- Синхронизация времени;
- Поддержка тестирования и диагностики;
- Безопасность.

Для настройки драйвера IEC61850 нужно выбрать его из списка и затем задать его параметры (рис. 3.156).

Драйверы

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
driver_findfac	FindFace	string	Выключен	2 Ошибки			...
driver_perco	Perco	string	Выключен	Нет ошиб			...
driver_trassir	Trassir	string	Выключен	Нет ошиб			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошиб			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...
SNMP TEST	SNMP	test	Выключен	1 Ошибка			...

Настройка драйвера IEC61850:

- Имя: IEC61850
- Описание: demo_01
- Тип: IEC61850
- Нода по умолчанию: 2
- ☒ Использовать ноду по умолчанию

Пул настроек:

- ip: 10
- port: 10
- renew_period_ms: 10

Файлы:

+ Добавить файл

Отмена Сохранить

Рис. 3.156. Настройки драйвера

Настройки драйвера IEC61850:

- ip - адрес устройства;
- port - порт устройства;
- renew_period_ms - период обновления значений в формате 1d1h1m1s1ms.

По завершении нажать на кнопку «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.157).

Подключить драйвер ☒

Драйвер
IEC61850:IEC61850

Точка привязки
Выберите объект для привязки

Флаг
Выберите флаг

Рис. 3. 157. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.158).

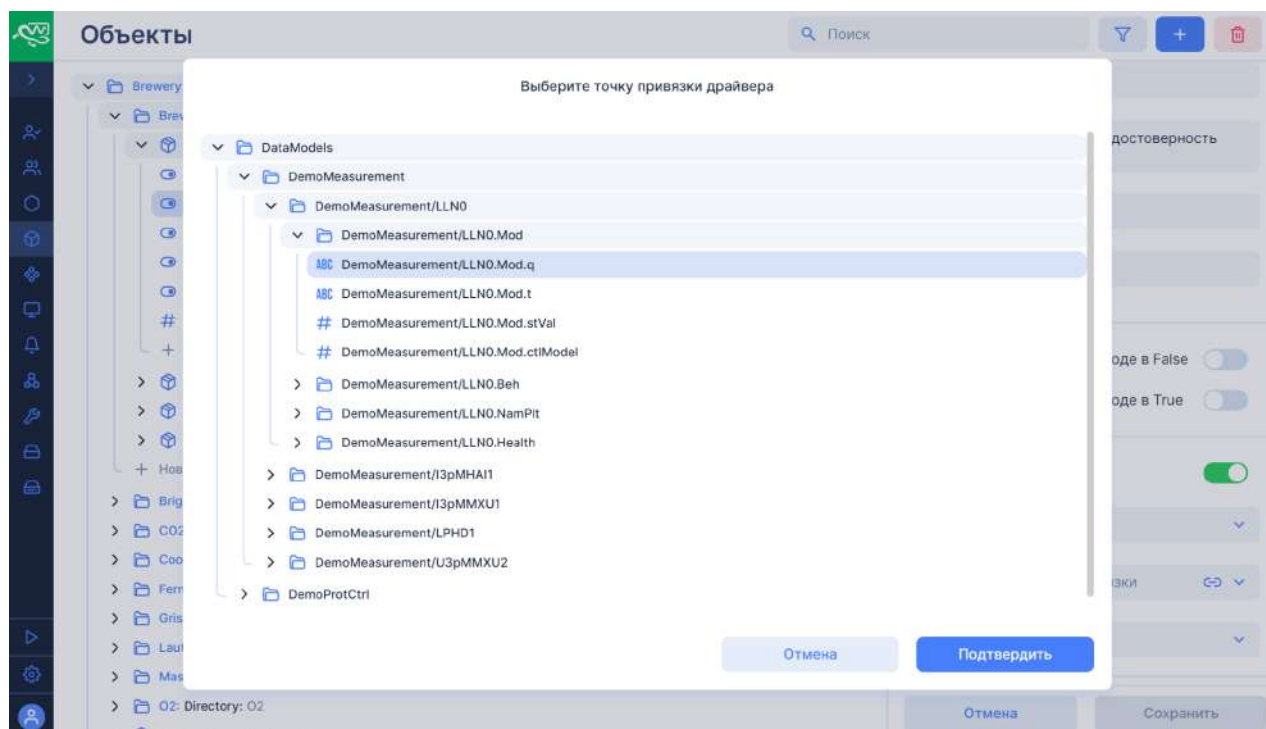


Рис. 3.158. Дерево тегов

3.6.11. Драйвер IEC 60870-5-104

IEC 60870-5-104 (IEC 104) — это международный стандарт протокола передачи данных, разработанный Международной электротехнической комиссией (IEC). Он является частью стандарта IEC 60870-5 и представляет собой расширение протокола IEC 60870-5-101, адаптированное для работы в сетях TCP/IP. Протокол используется для телекоммуникационного обмена данными в системах управления и мониторинга, таких как SCADA (Supervisory Control and Data Acquisition).

IEC 60870-5-104 предназначен для обеспечения надежной и эффективной передачи данных между управляющими центрами (диспетчерскими пунктами) и удаленными устройствами (например, RTU — Remote Terminal Units, IED — Intelligent Electronic Devices) в системах автоматизации, преимущественно в электроэнергетике, но также в других отраслях, таких как водоснабжение, газоснабжение и транспорт.

Основные функции протокола:

- Передача данных;
- Стандартизированный обмен данными;
- Надежность и контроль;
- Работа в сетях TCP/IP;
- Гибкость и масштабируемость;
- Синхронизация времени;
- Безопасность.

Для настройки драйвера IEC 60870-5-104 необходимо выбрать его из списка и затем задать его параметры (рис. 3.159).

Драйверы

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
driver_findfac	FindFace	string	Выключен	2 Ошибки			...
driver_perco	Perco	string	Выключен	Нет ошиб			...
driver_trassir	Trassir	string	Выключен	Нет ошиб			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошиб			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwverdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...
SNMP TEST	SNMP	test	Выключен	1 Ошибка			...

Параметры драйвера IEC60870:

- Имя: IEC60870
- Описание: demo_01
- Тип: IEC60870
- Нода по умолчанию: 2
- ☒ Использовать ноду по умолчанию

Пул настроек:

- ip: 10
- port: 10

Файлы: + Добавить файл

Аккаунты:

Отмена Сохранить

Рис. 3.159. Настройки драйвера

Настройки драйвера IEC 60870-5-104:

- ip - адрес устройства;
- port - порт устройства.

По завершении нажать на кнопку «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки (либо выбрать из списка статусов, рис. 3.160).

Рис. 3. 160. Точка привязки

Точка привязки задается в следующем формате (без пробелов):

addrASDU;addrDevice;Type(operation);доп, где:

- AddrASDU - адрес ASDU;
- AddrDevice - адрес Информационного объекта;
- Type (operation) - тип кадра;
- Доп.аргументы - 1 или более аргументов, также разделенных через символ точки с запятой. Их количество и смысл зависит от выбранного типа данных.

Мониторинг

№	Название	Первый доп. аргумент (значение объекта)	Особенности	Пример использования
1	M_SP_NA_1	0 – value (bool); 1 – qds (int64).		1;2;1;0
2	M_SP_TA_1	0 – value (bool); 1 – qds (int64).		1;2;2;0
3	M_DP_NA_1	0 – value (int64); 1 – qds (int64).		1;2;3;0
4	M_DP_TA_1	0 – value (int64); 1 – qds (int64).		1;2;4;0

5	M_ST_NA_1	0 – value (int64); 1 – has_trancient (bool); 2 – qds (int64).		1;2;5;0
6	M_ST_TA_1	0 – value (int64); 1 – has_trancient (bool); 2 – qds (int64).		1;2;6;0
7	M_BO_NA_1	0 – value (bool); 1 – qds (int64).	При 0 нужен второй доп. аргумент, указывающий на конкретный бит (0–31)	1;2;7;0;3 (для третьего бита), 1;2;7;1 (для qds)
8	M_BO_TA_1	0 – value (bool); 1 – qds (int64).	При 0 нужен второй доп. аргумент, указывающий на конкретный бит (0–31)	1;2;8;0;3 (для третьего бита), 1;2;8;1 (для qds)
9	M_ME_NA_1	0 – value (int64); 1 – qds (int64).		1;2;9;0
10	M_ME_TA_1	0 – value (int64); 1 – qds (int64).		1;2;10;0
11	M_ME_NB_1	0 – value (int64); 1 – qds (int64).		1;2;11;0
12	M_ME_TB_1	0 – value (int64); 1 – qds (int64).		1;2;12;0
13	M_ME_NC_1	0 – value (float64); 1 – qds (int64).		1;2;13;0
14	M_ME_TC_1	0 – value (float64); 1 – qds (int64).		1;2;14;0
15	M_IT_NA_1	0 – value (int64).		1;2;15;0
16	M_IT_TA_1	0 – value (int64).		1;2;16;0
17	M_EP_TA_1	0 – event (int64); 1 – qdp (int64);		1;2;17;0

		2 – msec (int64).		
18	M_EP_TB_1	0 – event (bool); 1 – qdp (int64); 2 – msec (int64).	При 0 нужен второй доп. аргумент, указывающий номер бита (0–7)	1;2;18;0;3 (для третьего бита), 1;2;18;2 (для msec)
19	M_EP_TC_1	0 – event (bool); 1 – qdp (int64); 2 – msec (int64).	При 0 нужен второй доп. аргумент, указывающий номер бита (0–7)	1;2;19;0;3 (для третьего бита), 1;2;19;2 (для msec)
20	M_PS_NA_1	0 – scd (bool); 1 – qds (int64); 2 – msec (int64).	При 0 нужен второй доп. аргумент, указывающий номер бита (0–31)	1;2;20;0;3 (для третьего бита), 1;2;20;2 (для msec)
21	M_ME_ND_1	0 – value (int64).		1;2;21;0
30	M_SP_TB_1	0 – value (bool); 1 – qds (int64).		1;2;30;0
31	M_DP_TB_1	0 – value (int64); 1 – qds (int64).		1;2;31;0
32	M_ST_TB_1	0 – value (int64); 1 – has_trancient (bool); 2 – qds (int64).		1;2;32;0
33	M_BO_TI_1	0 – value (bool); 1 – qds (int64).	При 0 нужен второй доп. аргумент, указывающий на конкретный бит (0–31)	1;2;33;0;3 (для третьего бита), 1;2;33;1 (для qds)
34	M_ME_TD_1	0 – value (int64); 1 – qds (int64).		1;2;34;0
35	M_ME_TE_1	0 – value (int64); 1 – qds (int64).		1;2;35;0

36	M_ME_TF_1	0 – value (float64); 1 – qds (int64).		1;2;36;0
37	M_IT_TB_1	0 – value (int64).		1;2;37;0
38	M_EP_TD_1	0 – event (int64); 1 – qdp (int64); 2 – msec (int64).		1;2;38;0
39	M_EP_TE_1	0 – event (bool); 1 – qdp (int64); 2 – msec (int64).	При 0 нужен второй доп. аргумент, указывающи й номер бита (0–7	1;2;39;0;3 (для третьего бита), 1;2;39;2 (для msec)
40	M_EP_TF_1	0 – event (bool); 1 – qdp (int64); 2 – msec (int64).	При 0 нужен второй доп. аргумент, указывающи й номер бита (0–7)	1;2;40;0;3 (для третьего бита), 1;2;40;2 (для msec)
70	M_EI_NA_1	0 – cause of initialization (int64); 1 – is local change (bool).		1;2;70;0

Команды

№	Название команды	Тип объекта	Параметры команды	Пример использования
45	C_SC_NA_1	bool	addrASDU; addrDevice; Type	1;2;45
46	C_DC_NA_1	int64	addrASDU; addrDevice; Type	1;2;46
47	C_RC_NA_1	int64	addrASDU; addrDevice; Type	1;2;47
48	C_SE_NA_1	int64	addrASDU; addrDevice; Type	1;2;48
49	C_SE_NB_1	int64	addrASDU; addrDevice; Type	1;2;49
50	C_SE_NC_1	float64	addrASDU; addrDevice; Type	1;2;50
51	C_BO_NA_1	int64	addrASDU; addrDevice; Type	1;2;51
58	C_SC_TA_1	bool	addrASDU; addrDevice; Type	1;2;58
59	C_DC_TA_1	int64	addrASDU; addrDevice; Type	1;2;59
60	C_RC_TA_1	int64	addrASDU; addrDevice; Type	1;2;60
61	C_SE_TA_1	int64	addrASDU; addrDevice; Type	1;2;61
62	C_SE_TB_1	int64	addrASDU;	1;2;62

			addrDevice; Type	
63	C_SE_TC_1	float64	addrASDU; addrDevice; Type	1;2;63
64	C_BO_TA_1	int64	addrASDU; addrDevice; Type	1;2;64

У команд также присутствуют дополнительные параметры, которые задаются драйвером (пользователь их задать не может):

Cause of transmission всегда 6 (Activation);

SE всегда 0 (Execute);

QU всегда 0 (No Additional Definition).

3.7. Драйверы сопряжения с внешними провайдерами

3.7.1. Драйвер Trassir

Протокол Trassir — это проприетарный коммуникационный протокол (или набор интерфейсов), разработанный компанией DSSL (Digital Security Systems Lab) для использования в экосистеме видеонаблюдения Trassir. Он предназначен для обеспечения взаимодействия между IP-камерами, видеорегистраторами (NVR/DVR), серверами и клиентским программным обеспечением Trassir, включая передачу видеопотоков, метаданных, событий и команд управления.

Протокол Trassir используется для интеграции и управления компонентами системы видеонаблюдения, обеспечивая передачу видео, аудио, аналитических данных и команд в реальном времени. Он применяется в системах безопасности для мониторинга, записи, анализа и удаленного управления видеонаблюдением на объектах различного масштаба — от частных домов до крупных предприятий.

Основные функции протокола:

- Передача видеопотоков;
- Управление устройствами;
- Обработка событий и аналитики;
- Интеграция с другими системами;
- Удаленный доступ и управление;

- Безопасность данных;
- Мониторинг и диагностика.

Для настройки драйвера Trassir нужно выбрать его из списка и затем задать его параметры (рис. 3.161).

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
BACnet	BACnet	demo_01	Выключен	1 Ошибка			...
driver_findfac	FindFace	string	Выключен	2 Ошибки			...
driver_perco	Perco	string	Выключен	Нет ошибок			...
driver_trassir	Trassir	string	Выключен	Нет ошибок			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошибок			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...

Настройка драйвера driver_trassir:

- Имя: driver_trassir
- Описание: string
- Тип: Trassir
- Нода по умолчанию: 1
- ☒ Использовать ноду по умолчанию

Пул настроек:

- renew_period_ms
- host

Файлы:

+ Добавить файл

Аккаунты:

Отмена Сохранить

Рис. 3.161. Настройки драйвера

Настройки драйвера Trassir:

- renew_period_ms - период обновления (опроса) системы;
- host - IP-адрес системы.

По завершении нажать на кнопку «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.162).

Подключить драйвер ☒

Драйвер

driver_trassir:Trassir

Точка привязки

Выберите объект для привязки

Флаг

Выберите флаг

Рис. 3.162. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.163).

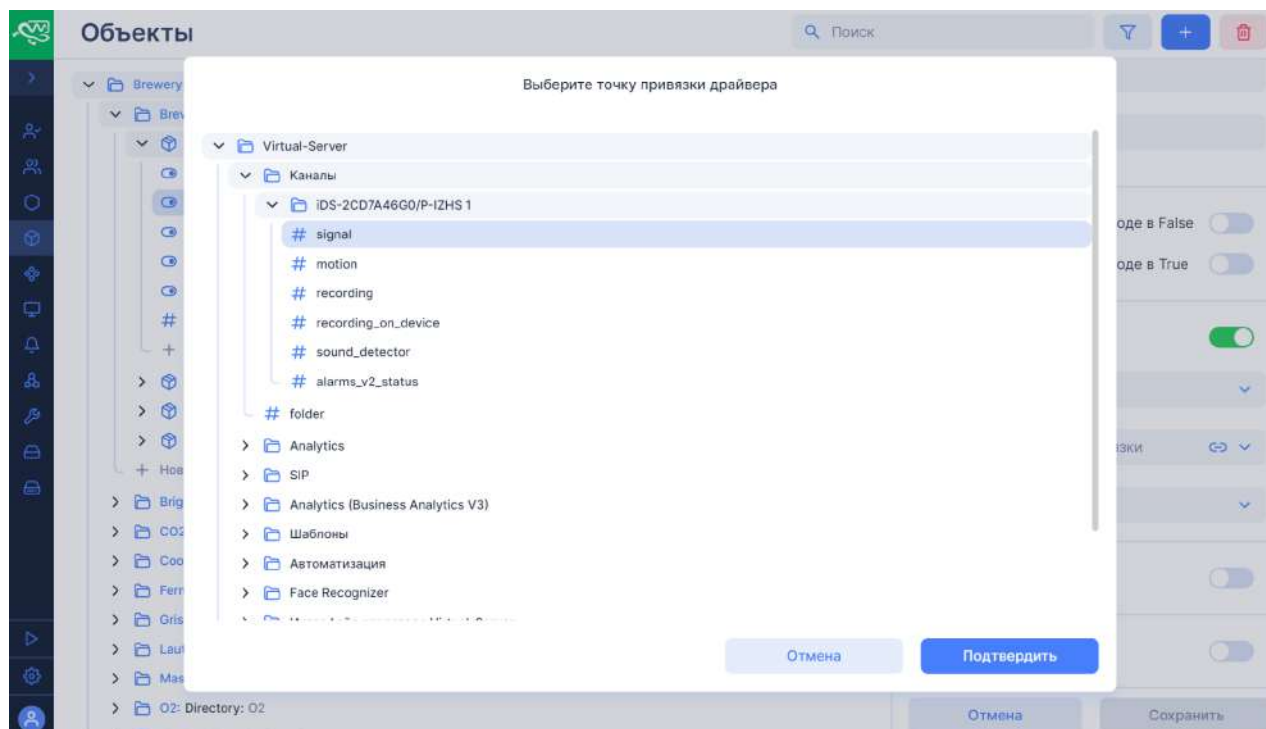


Рис. 3.163. Дерево тегов

Пользователь имеет возможность настраивать подключения по данному драйверу. Для этого необходимо перейти в раздел «Настройки», после чего выбрать «Trassir» и нажать «+ Добавить Новое Подключение», заполнив все необходимые данные для подключения. По завершении нажать «Сохранить».

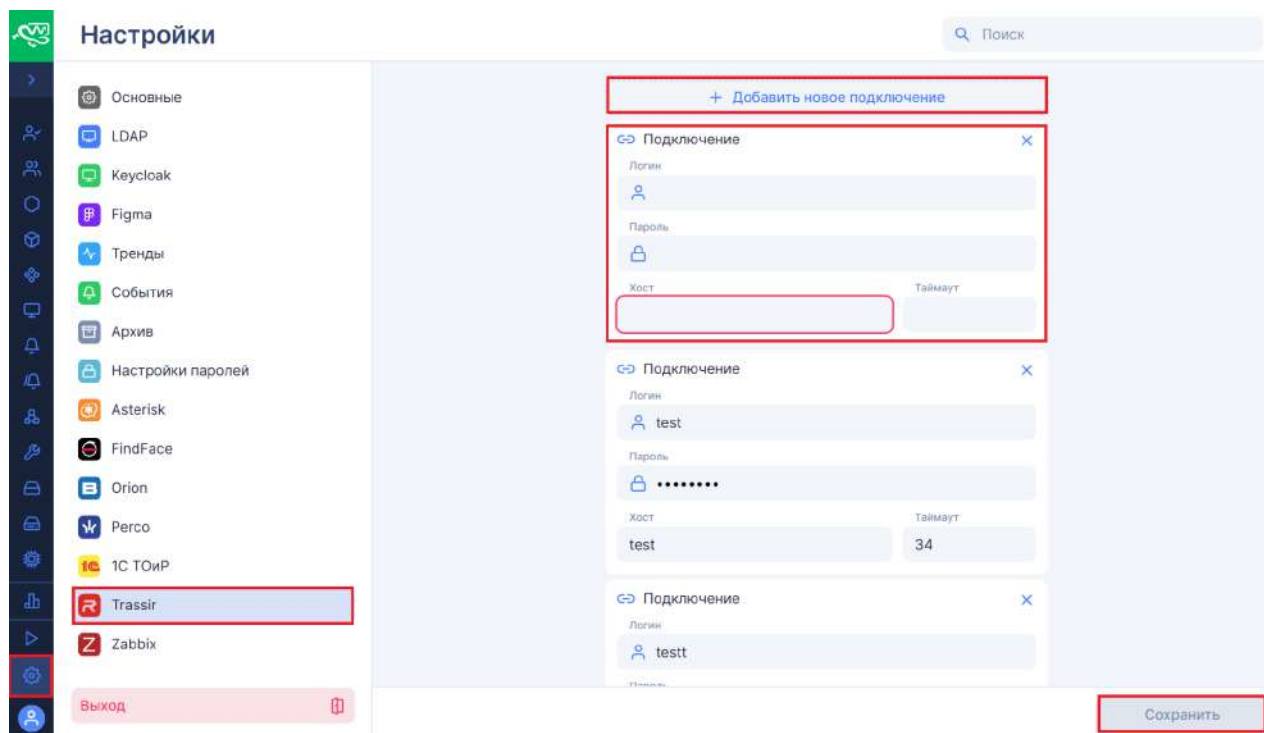


Рис. 3.164. Настройка подключения

3.7.2. Драйвер Perco

Протокол Perco — это проприетарный коммуникационный протокол или набор интерфейсов, используемый в системах контроля и управления доступом (СКУД) компании PERCo. Он предназначен для обмена данными между устройствами PERCo (турникеты, считыватели карт, контроллеры доступа) и управляющим программным обеспечением, обеспечивая управление доступом, регистрацию событий и мониторинг состояния оборудования.

Протокол Perco используется для интеграции и управления компонентами систем контроля доступа PERCo, такими как турникеты, замки, считыватели RFID-карт и биометрические устройства.

Основные Функции протокола:

- Передача данных о событиях;
- Управление устройствами;
- Интеграция с внешними системами;
- Мониторинг и диагностика;
- Безопасность передачи данных;
- Удаленное управление.

Для настройки драйвера Perco нужно выбрать его из списка и затем задать его параметры (рис. 3.165).

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
BACnet	BACnet	demo_01	Выключен	1 Ошибка			...
driver_findfac	FindFace	string	Выключен	2 Ошибки			...
driver_perco	Perco	string	Выключен	Нет ошибок			...
driver_trassir	Trassir	string	Выключен	Нет ошибок			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошибок			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA 03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...

Настройка драйвера Perco:

- Имя: driver_perco
- Описание: string
- Тип: Perco
- Нода по умолчанию: 1
- ☒ Использовать ноду по умолчанию
- Пул настроек:**
 - renew_period_ms: [поле]
 - host: [поле]
- Файлы:** + Добавить файл
- Аккаунты:**
- Кнопки: Отмена, Сохранить

Рис. 3. 165. Настройки драйвера

Настройки драйвера Perco:

- renew_period_ms - период обновления (опроса) системы;
- host - IP-адрес системы.

По завершении нажать на кнопку «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.166).

Подключить драйвер ☒

Драйвер
driver_perco:Perco

Точка привязки
Выберите объект для привязки

Флаг
Выберите флаг

Рис. 3. 166. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.167).

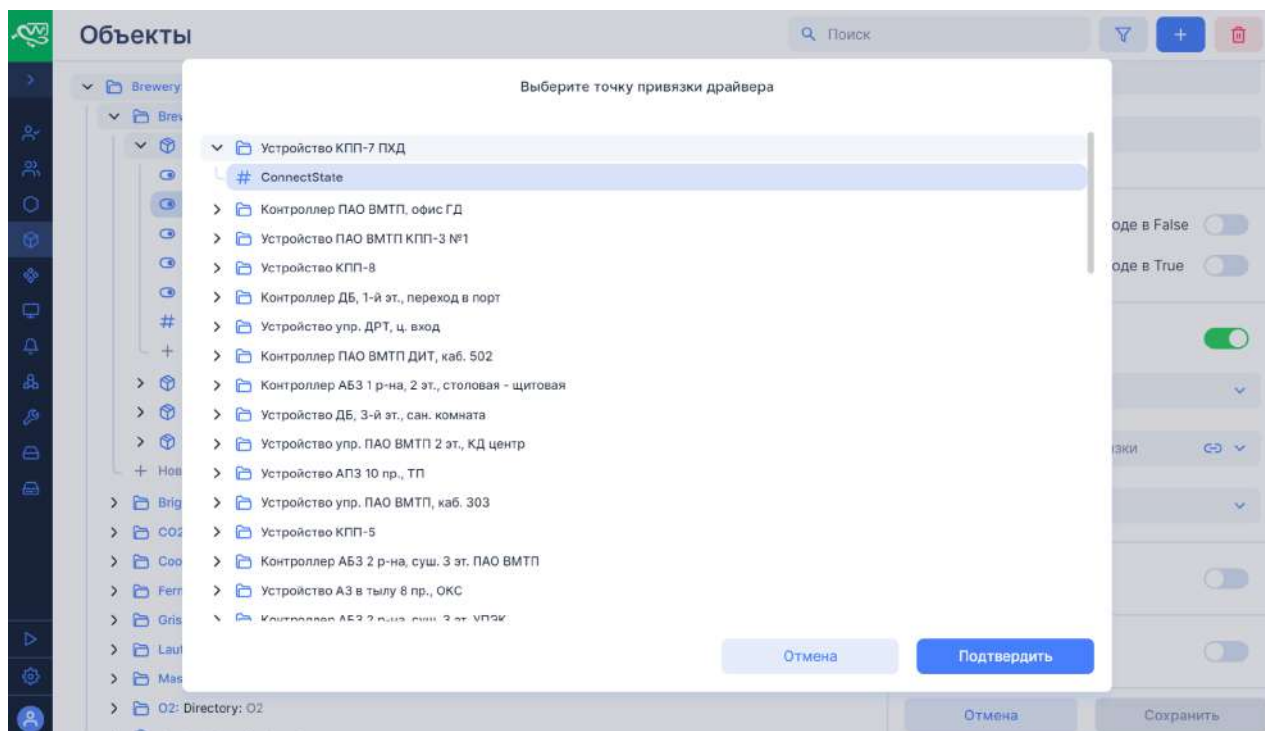


Рис. 3.167. Дерево тегов

Пользователь имеет возможность настраивать подключения по данному драйверу. Для этого необходимо перейти в раздел «Настройки», после чего выбрать «Perco» и нажать «+ Добавить Новое Подключение», заполнив все необходимые данные для подключения. По завершении нажать «Сохранить» (рис. 3.168).

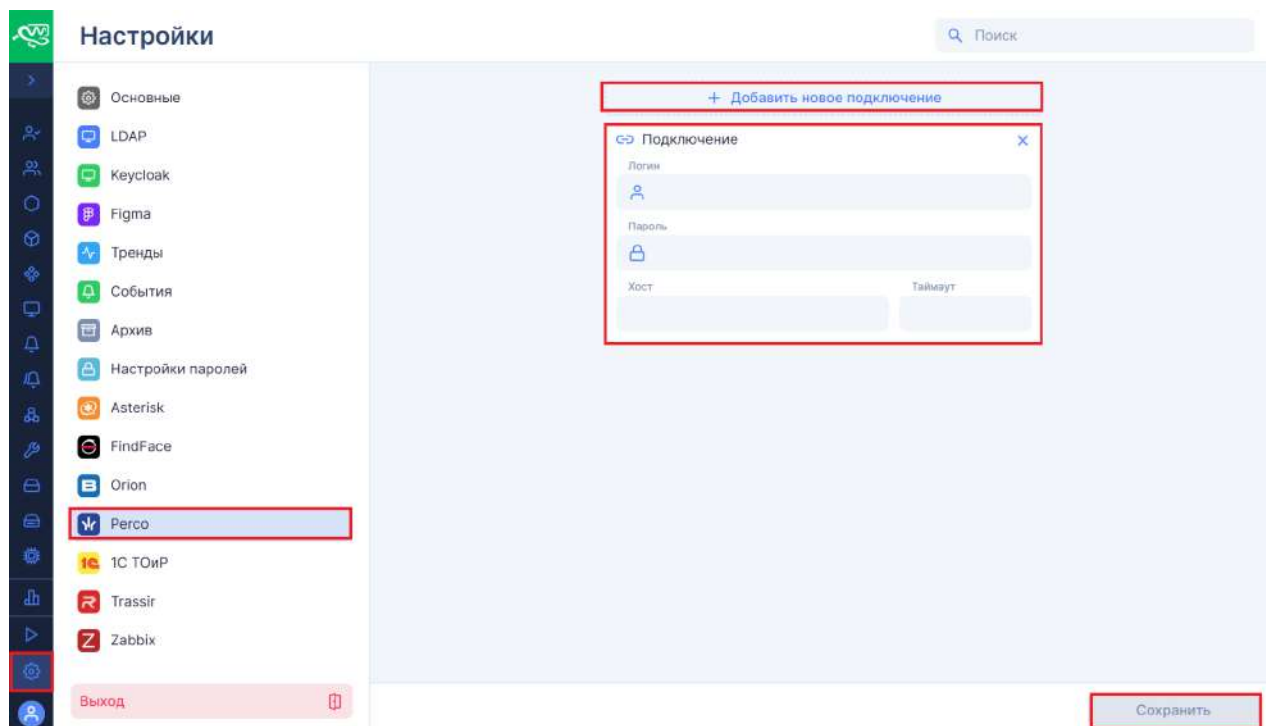


Рис. 3.168. Настройка подключения

3.7.3. Драйвер FindFace

Протокол FindFace — это проприетарный интерфейс или набор API (Application Programming Interface), используемый в продуктах компании NTechLab для реализации функций распознавания лиц и анализа видео в системах видеонаблюдения и безопасности. FindFace, разработанный российской компанией NTechLab, представляет собой технологию на основе нейронных сетей, предназначенную для высокоточного распознавания лиц, тел, действий, автомобилей и номерных знаков в реальном времени или в видеоархивах. Протокол обеспечивает взаимодействие между программным обеспечением FindFace (например, FindFace Multi, FindFace Enterprise Server SDK) и аппаратными устройствами (камерами, серверами), а также интеграцию с внешними системами.

Протокол FindFace предназначен для обеспечения функциональности систем распознавания лиц и видеоаналитики, включая:

- Идентификацию и верификацию лиц в реальном времени или архивных записях;
- Анализ характеристик объектов (возраст, пол, эмоции, одежда, тип транспортного средства);
- Интеграцию с системами видеонаблюдения, базами данных и другими платформами безопасности;
- Обеспечение быстрого поиска лиц или объектов в больших базах данных (до миллиардов записей).

Основные функции протокола:

- Распознавание и идентификация лиц;
- Обработка видеопотоков;
- Анализ характеристик;
- Интеграция и API;
- Безопасность и конфиденциальность;
- Мониторинг и аналитика;
- Масштабируемость.

Для настройки драйвера FindFace нужно выбрать его из списка и затем задать его параметры (рис. 3.169).

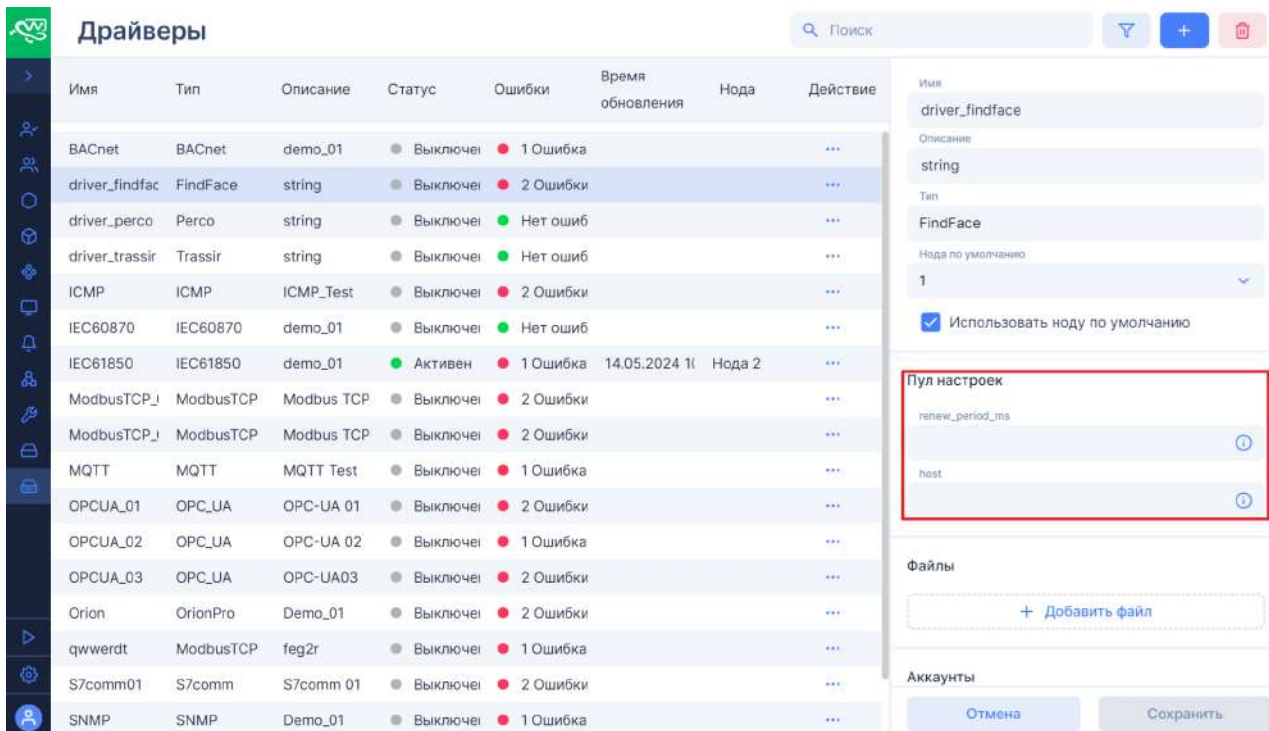


Рис. 3.169. Настройки драйвера

Настройки драйвера FindFace:

- renew_period_ms - период обновления (опроса) системы;
- host - IP-адрес системы.

По завершении нажать «Сохранить».

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.170).

Рис. 3.170. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.171).

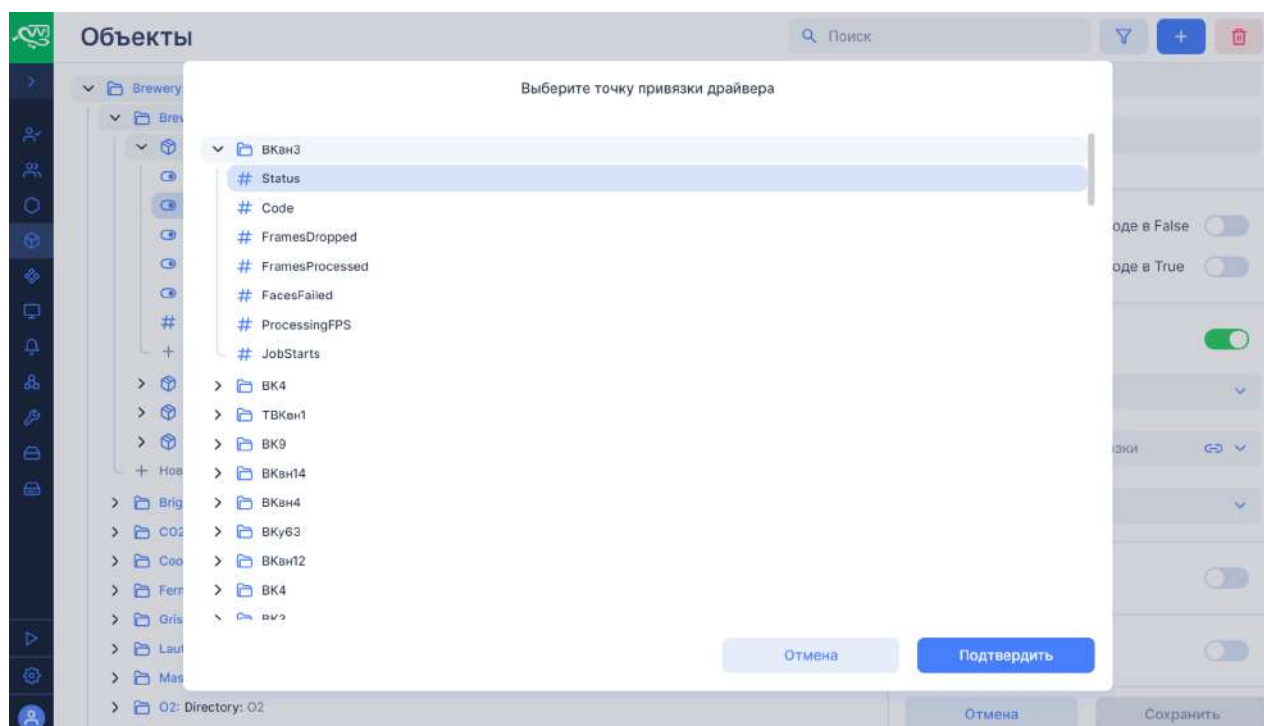


Рис. 3.171. Дерево тегов

Пользователь имеет возможность настраивать подключения по данному драйверу. Для этого необходимо перейти в раздел «Настройки», после чего выбрать «FindFace» и нажать «+ Добавить Новое Подключение», заполнив все необходимые данные для подключения. По завершении нажать «Сохранить» (рис.3.172).

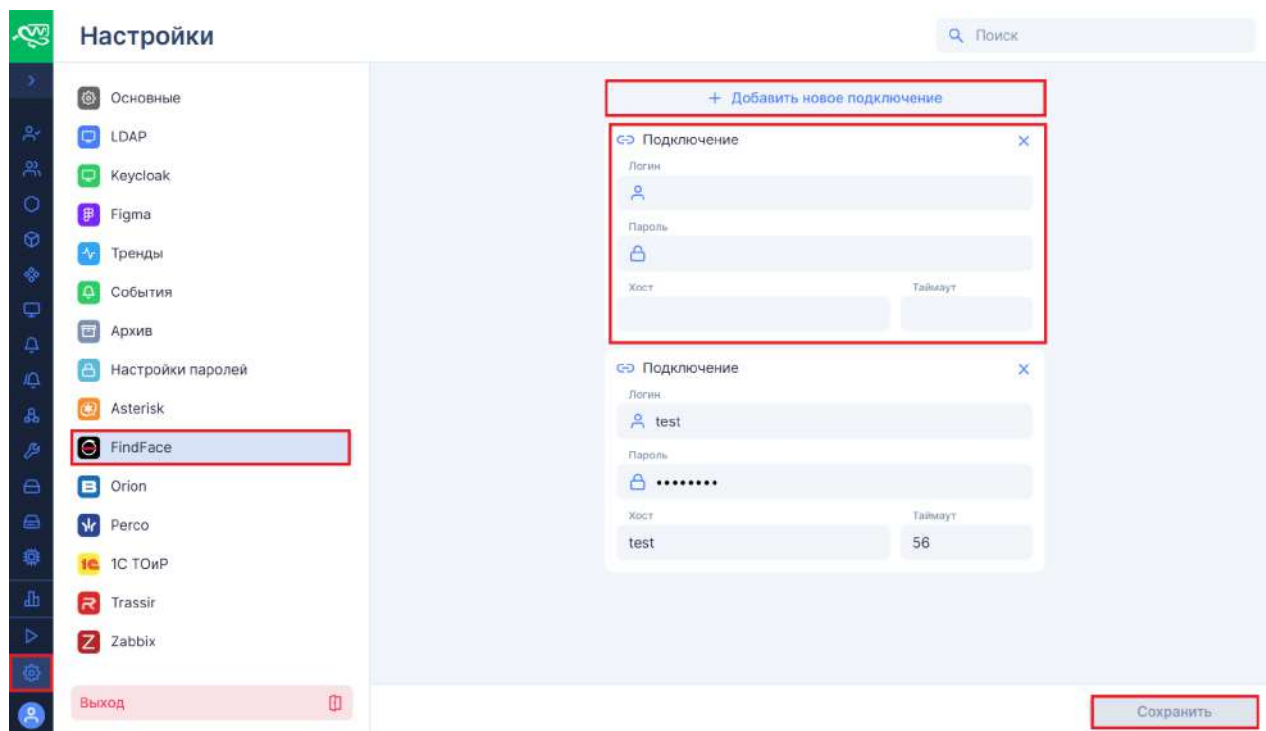


Рис. 3.172. Настройка подключения

3.7.4. Драйвер Zabbix

Протокол Zabbix — это проприетарный коммуникационный протокол, используемый в системе мониторинга Zabbix для обмена данными между компонентами системы, такими как Zabbix Server, Zabbix Proxy, Zabbix Agent и утилитами командной строки (например, Zabbix Sender). Он основан на JSON-формате и работает преимущественно по протоколу TCP (порты 10050 для агентов и 10051 для сервера/прокси), обеспечивая передачу метрик, событий, конфигурационных данных и команд в системах мониторинга IT-инфраструктуры.

Протокол Zabbix предназначен для обеспечения взаимодействия между компонентами системы Zabbix, чтобы собирать, передавать и обрабатывать данные мониторинга (метрики, состояние устройств, события) в реальном времени. Он используется для:

- Мониторинга серверов, сетевых устройств, виртуальных машин, облачных сервисов и приложений;
- Передачи конфигурационных данных между сервером, прокси и агентами;
- Обработки событий, таких как уведомления о сбоях или превышении пороговых значений.

Основные функции протокола:

- Передача метрик;
- Активные и пассивные проверки;
- Обмен конфигурацией;
- Обработка событий и уведомлений;
- Безопасность;
- Интеграция с другими протоколами;
- Масштабируемость.

Для настройки драйвера Zabbix нужно выбрать его из списка и затем задать его параметры (рис. 3.173).

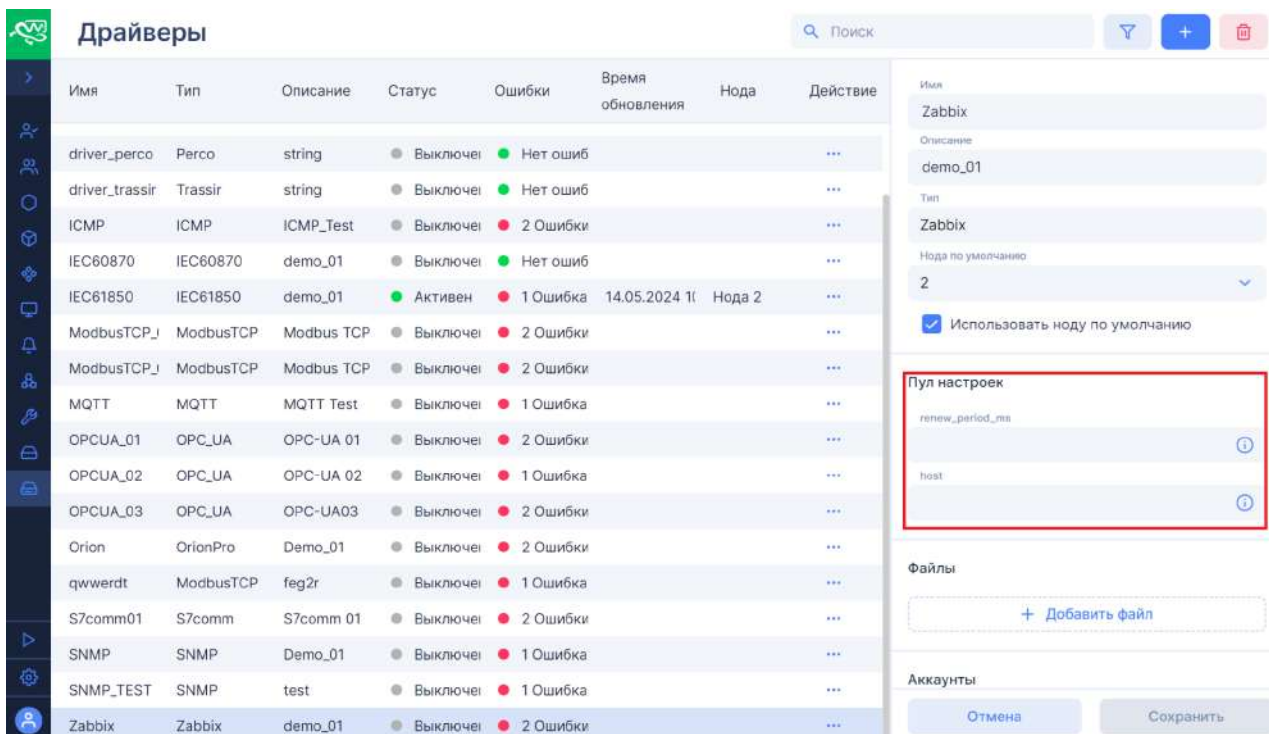


Рис. 3.173. Настройки драйвера

Настройки драйвера Zabbix:

- renew_period_ms - период обновления (опроса) системы;
- host - IP-адрес системы.

По завершении нажать на кнопку «Сохранить»

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.174).

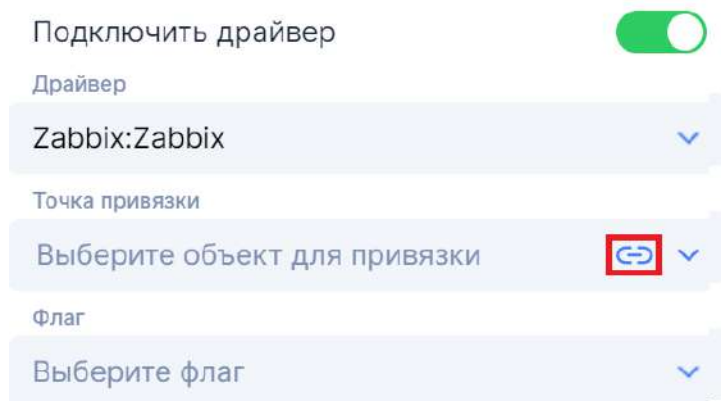


Рис. 3.174. Точка привязки

Откроется дерево тегов (оно считывается непосредственно из устройства с которым настраивается связь), в котором необходимо выбрать нужный тег и нажать «Подтвердить» (рис. 3.175).

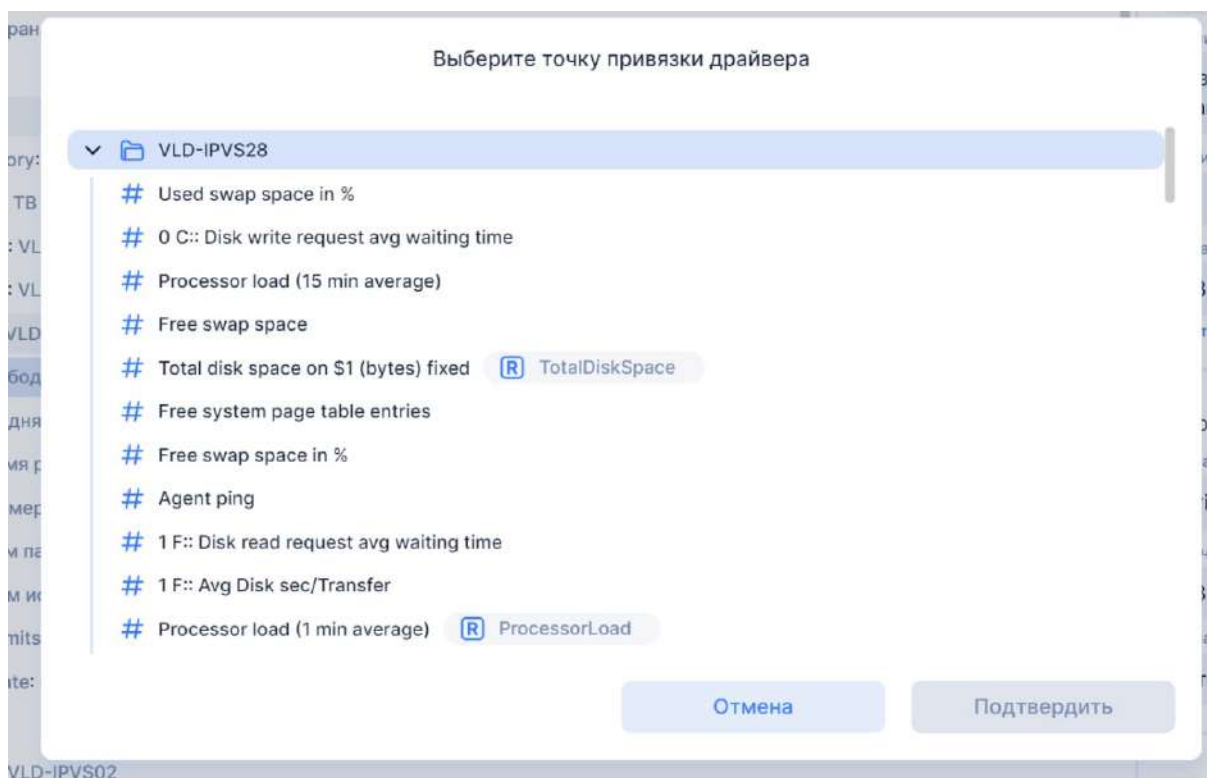


Рис. 3.175. Дерево тегов

Пользователь имеет возможность настраивать подключения по данному драйверу. Для этого необходимо перейти в раздел «Настройки», после чего выбрать «Zabbix» и нажать «+ Добавить Новое Подключение», заполнив все необходимые данные для подключения. По завершении нажать «Сохранить» (рис.3.176).

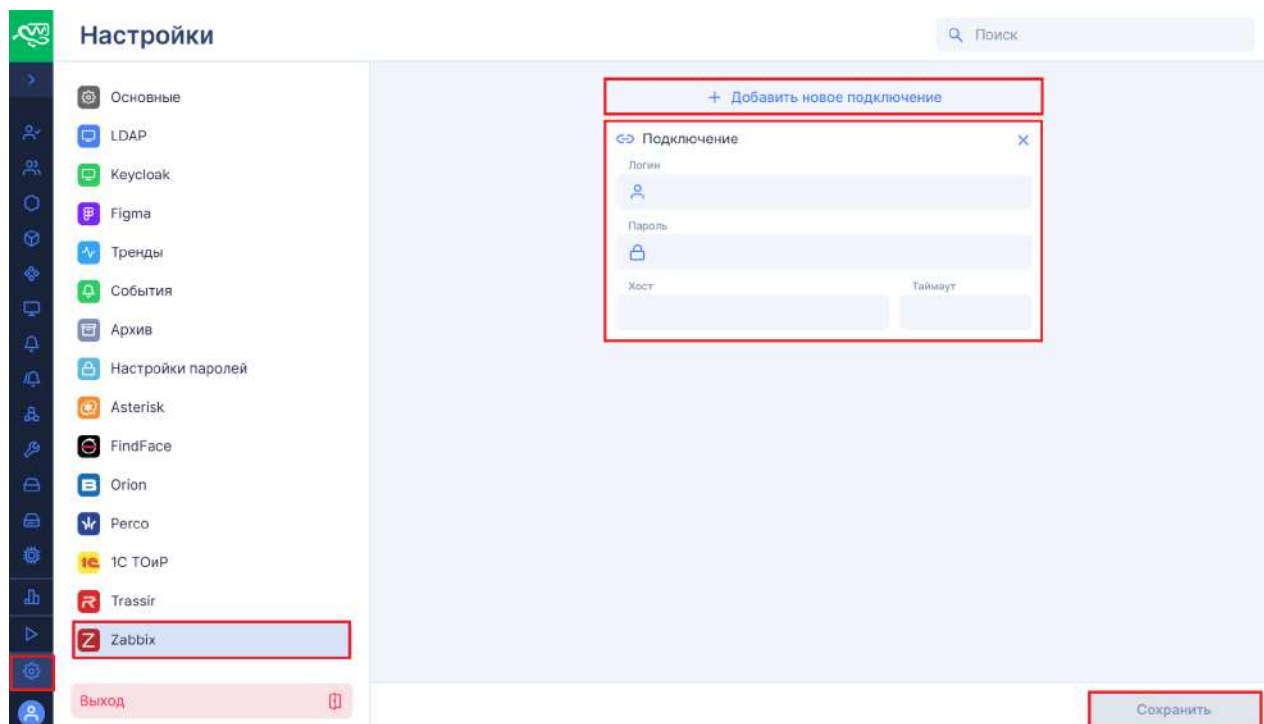


Рис. 3.176. Настройка подключения

3.7.5. Драйвер OrionPro

Протокол OrionPro — это проприетарный коммуникационный протокол, используемый в системах видеонаблюдения и безопасности, разработанных компанией Orion. Протокол OrionPro обеспечивает взаимодействие между устройствами (камеры, датчики, контроллеры) и программным комплексом «Орион Про».

Протокол OrionPro предназначен для обеспечения обмена данными между компонентами системы безопасности, входящими в экосистему «Орион Про».

Основные функции протокола:

- Передача данных и событий;
- Управление устройствами;
- Интеграция подсистем;
- Передача видеопотоков;
- Мониторинг и диагностика;
- Безопасность передачи данных;
- Удаленное управление.

Для настройки драйвера Orion нужно выбрать его из списка и затем задать его параметры (рис. 3.177).

Имя	Тип	Описание	Статус	Ошибки	Время обновления	Нода	Действие
driver_perco	Perco	string	Выключен	Нет ошибок			...
driver_trassir	Trassir	string	Выключен	Нет ошибок			...
ICMP	ICMP	ICMP_Test	Выключен	2 Ошибки			...
IEC60870	IEC60870	demo_01	Выключен	Нет ошибок			...
IEC61850	IEC61850	demo_01	Активен	1 Ошибка	14.05.2024 10:00	Нода 2	...
ModbusTCP_1	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
ModbusTCP_2	ModbusTCP	Modbus TCP	Выключен	2 Ошибки			...
MQTT	MQTT	MQTT Test	Выключен	1 Ошибка			...
OPCUA_01	OPC-UA	OPC-UA 01	Выключен	2 Ошибки			...
OPCUA_02	OPC-UA	OPC-UA 02	Выключен	1 Ошибка			...
OPCUA_03	OPC-UA	OPC-UA03	Выключен	2 Ошибки			...
Orion	OrionPro	Demo_01	Выключен	2 Ошибки			...
qwwerdt	ModbusTCP	feg2r	Выключен	1 Ошибка			...
S7comm01	S7comm	S7comm 01	Выключен	2 Ошибки			...
SNMP	SNMP	Demo_01	Выключен	1 Ошибка			...
SNMP_TEST	SNMP	test	Выключен	1 Ошибка			...
Zabbix	Zabbix	demo_01	Выключен	2 Ошибки			...

Имя: Orion

Описание: Demo_01

Тип: OrionPro

Нода по умолчанию: 1

☐ Использовать ноду по умолчанию

Пул настроек

url:

renew_period_ms:

login:

password:

Файлы:

Отмена Сохранить

Рис. 3.177. Настройки драйвера

Настройки драйвера OrionPro:

- url - адрес soap-сервиса ОрионПро
- renew_period_ms - Период обновления значений в формате 1d1h1m1s1ms;
- login - Логин пользователя;
- password - Пароль пользователя.

По завершении нажать на кнопку «Сохранить»

Для привязки объекта к драйверу необходимо перейти во вкладку «Объекты», выбрать драйвер для целевого объекта согласно п.3.3.2 и задать точку привязки, нажав на соответствующую кнопку (Либо выбрать из списка статусов, рис. 3.178).

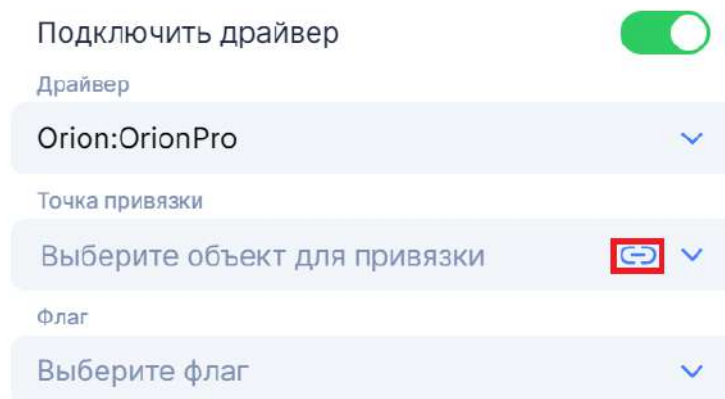


Рис. 3. 178. Точка привязки

Пользователь имеет возможность настраивать подключения по данному драйверу. Для этого необходимо перейти в раздел «Настройки», после чего выбрать «Orion» и нажать «+ Добавить Новое Подключение», заполнив все необходимые данные для подключения. По завершении нажать «Сохранить» (рис.3.179).

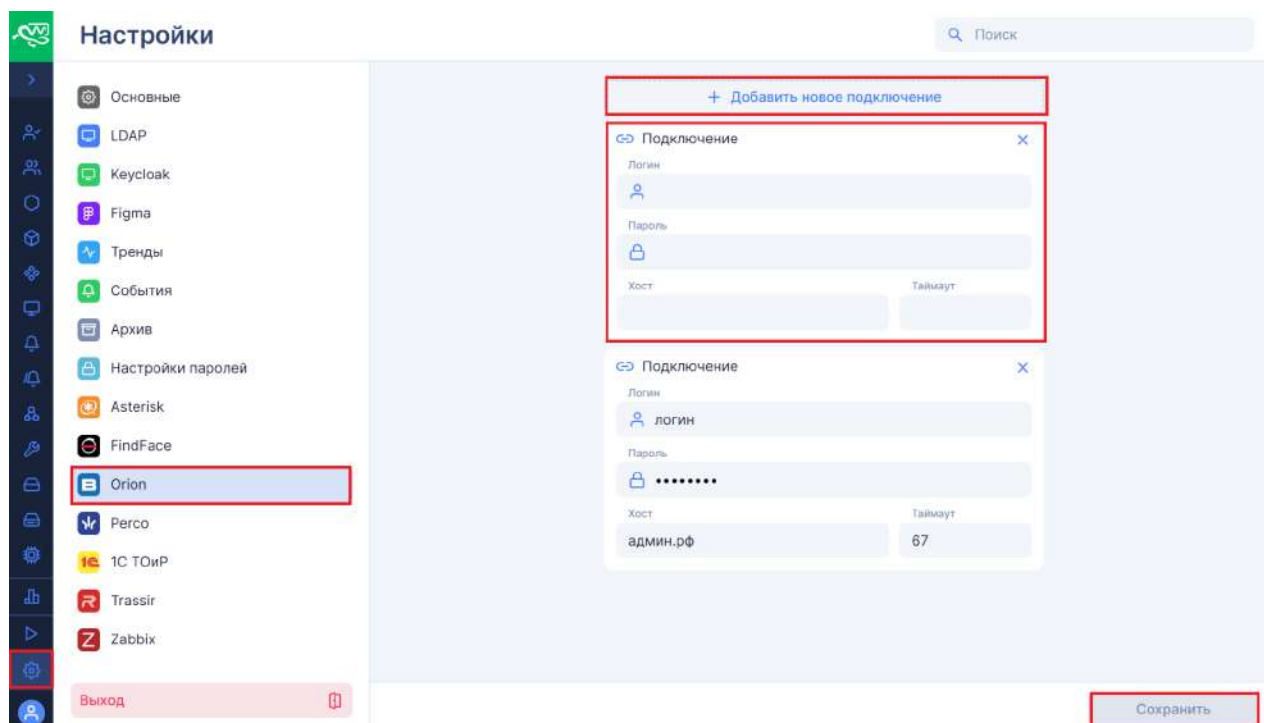


Рис.3.179. Настройка подключения

3.7.6. Драйвер Syslog

Протокол Syslog (System Log) — это стандартизированный сетевой протокол для сбора, передачи и централизованного хранения сообщений о событиях (логов) от различных сетевых устройств и программных компонентов.

Syslog предназначен для автоматизированного сбора, передачи и хранения логов с сетевых устройств и приложений, обеспечивая администраторам инструмент для мониторинга, диагностики и аудита ИТ-инфраструктуры.

Основные функции протокола:

- Сбор логов с разнородных устройств и приложений — серверов, маршрутизаторов, брандмауэров, принтеров и т. д.;
- Централизация данных: объединение логов из разных источников в едином хранилище (на сервере Syslog) для удобства мониторинга и анализа;
- Унификация формата сообщений — обеспечение единого вида логов от разных устройств, что упрощает их обработку и корреляцию;
- Мониторинг состояния инфраструктуры в реальном времени: отслеживание работоспособности сервисов и оборудования;
- Выявление инцидентов и ошибок — фиксация сбоев, предупреждений и аномалий для оперативного реагирования;
- Аудит безопасности: регистрация событий аутентификации, попыток несанкционированного доступа и других действий, важных для оценки защищённости системы;
- Диагностика и отладка — предоставление данных для поиска и устранения неисправностей в ПО и оборудовании;
- Долгосрочное хранение логов — создание архива событий для последующего анализа, расследований и соблюдения нормативных требований (например, по хранению данных).

Для настройки драйвера Syslog необходимо:

1. Создать драйвер по типу syslog, указав в «Пуле настроек» путь к файлу на диске и формат файла (рис. 3.180);

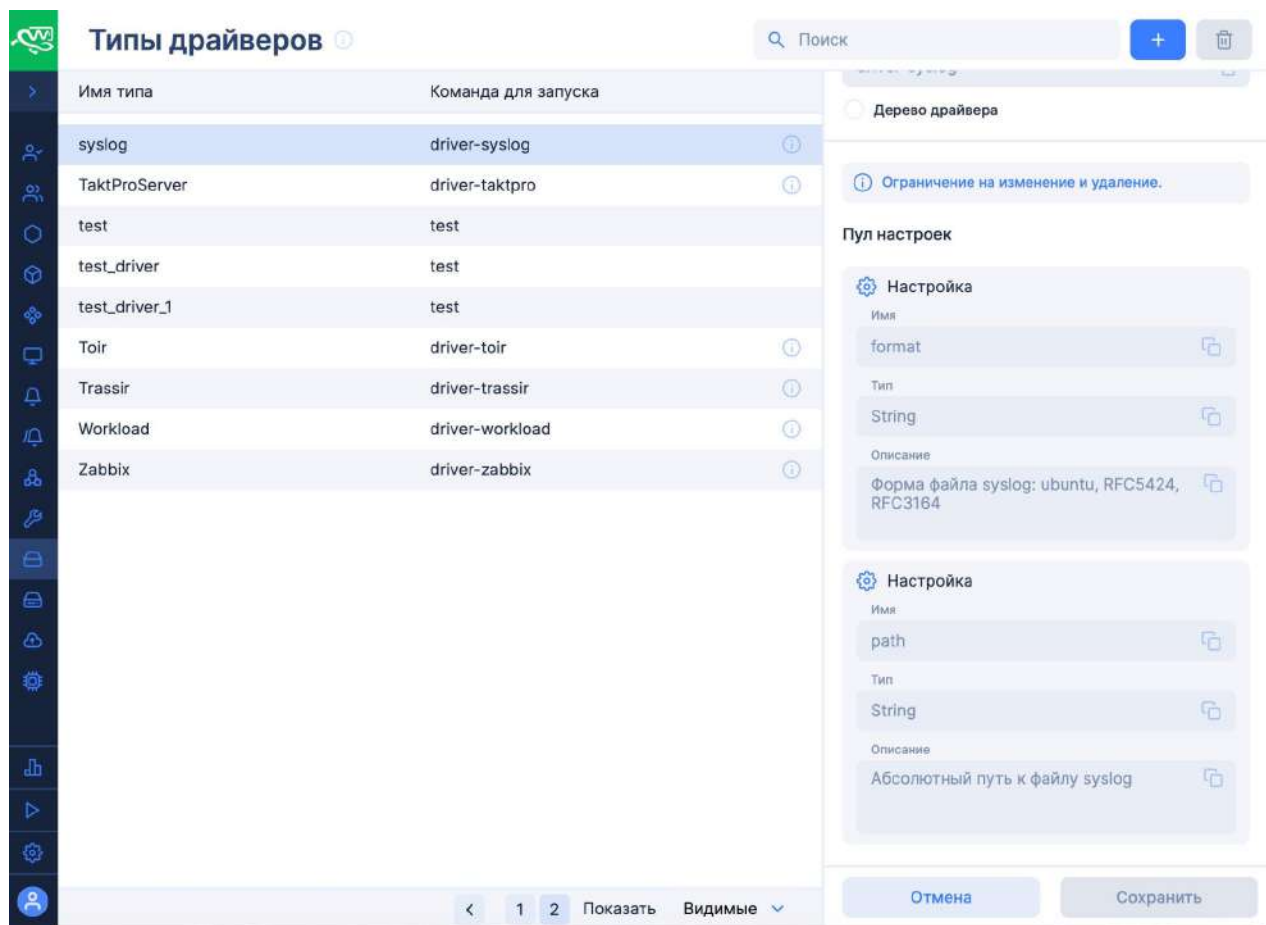


Рис. 3.180. Драйвер Syslog

2. Создать модель, например, Syslog_msg;

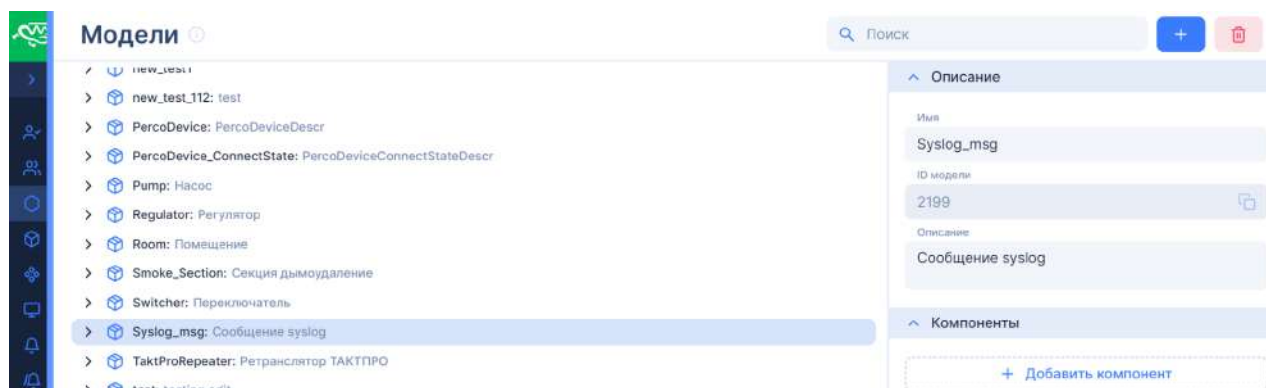
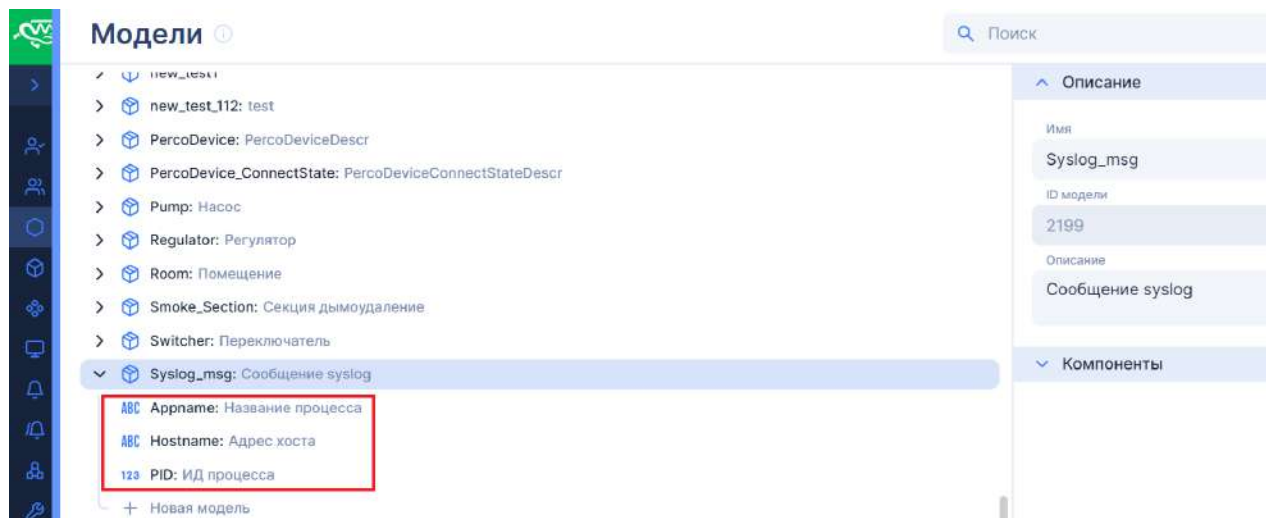


Рис. 3.181. Модель Syslog_msg

3. Добавить дочерние модели в модель (важно сохранить регистр названий, рис. 3.181);



Для формата файла ubuntu:

Appname - название процесса (String);

Hostname - адрес хоста (String);

PID - ИД процесса (Integer);

Message - сообщение от процесса (String).

Для формата файла RFC5424, RFC3164:

Appname - название процесса (String);

Hostname - адрес хоста (String);

PID - ИД процесса (Integer);

Message - сообщение от процесса (String);

Priority - приоритет (Integer);

Facility - код категории/источника (Integer);

Severity - важность (Integer);

MessageCounter - счетчик сообщений (Integer);

Sequence - идентификатор последовательности (Integer);

MsgID - идентификатор сообщения (Integer).

4. Создать объекты на основе модели из п. 2«



Рис. 3.182. Объекты созданной на основе модели Syslog_msg

5. В блоке «Запись» у объектов активировать источник «Драйвер», выбрать драйвер «syslog» и указать в качестве точки привязки названия объектов.

Например, объект Appname - точка привязки Appname.

3.8. Сервисы

Подсистема управления данными включает в себя следующие сервисы:

- Калькулятор (сервис расчетных сигналов);
- Генератор событий;
- Имитатор (сервис имитации значений объектов);
- Архиватор значений (сервис архивации);
- Обновление настроек интеграции.

3.8.1. Список сервисов Платформы

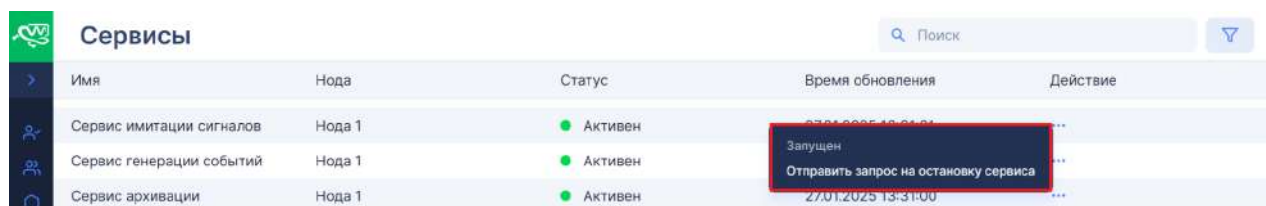
Для перехода в раздел «Сервисы» необходимо воспользоваться боковой панелью меню, нажав на соответствующую кнопку (рис. 3.180).

Имя сервиса	Нода	Статус	Время обновления	Действие
Сервис имитации сигналов	Нода 1	Активен	04.03.2026 19:00:58	...
Сервис генерации событий	Нода 1	Активен	04.03.2026 19:00:57	...
Сервис архивации	Нода 1	Активен	04.03.2026 19:00:57	...
Сервис вычислений	Нода 2	Активен	04.03.2026 19:00:57	...
Сервис обновления настроек ин	Нода 1	Активен	04.03.2026 19:00:58	...
Сервис создания резервных коп	Нода 1	Активен	04.03.2026 19:00:58	...

Рис. 3.180. Список сервисов Платформы

На странице раздела отображается список сервисов Платформы, в котором содержатся: имя сервиса, узел на котором он запущен в данный момент (нода), статус, время обновления (время запуска), и действия, которые можно произвести с сервисом.

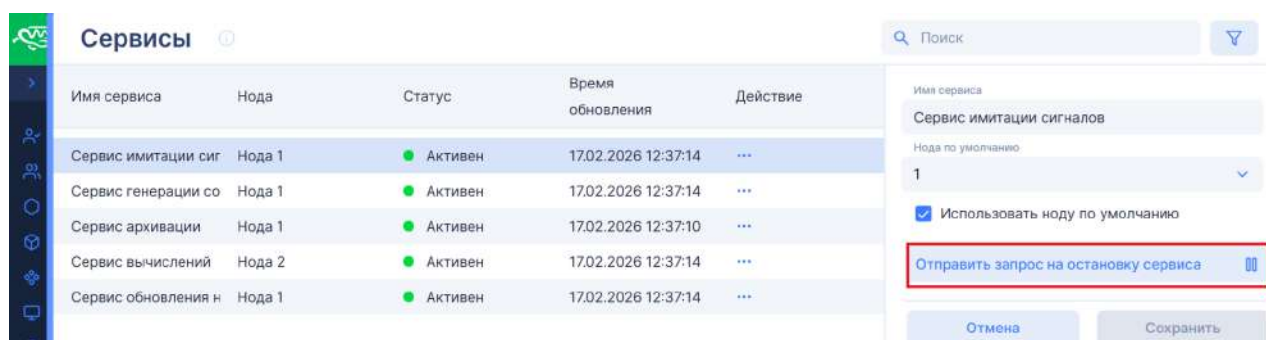
Для того чтобы выполнить действие в соответствующей строке сервиса необходимо нажать на кнопку «....», затем «Отправить запрос на остановку сервиса», после чего сервис будет отключен (рис. 3.181).



Имя	Нода	Статус	Время обновления	Действие
Сервис имитации сигналов	Нода 1	Активен	17.02.2026 12:37:14	...
Сервис генерации событий	Нода 1	Активен	17.02.2026 12:37:14	...
Сервис архивации	Нода 1	Активен	17.02.2026 12:37:10	...

Рис. 3.181. Кнопка «Действие»

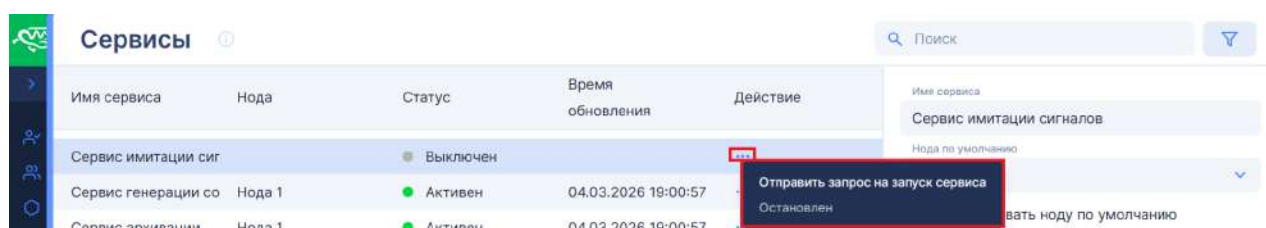
Также сервис можно остановить и через боковую вкладку параметров, открыв ее нажатием на строку сервиса (рис. 3.183).



Имя сервиса	Нода	Статус	Время обновления	Действие
Сервис имитации сиг	Нода 1	Активен	17.02.2026 12:37:14	...
Сервис генерации со	Нода 1	Активен	17.02.2026 12:37:14	...
Сервис архивации	Нода 1	Активен	17.02.2026 12:37:10	...
Сервис вычислений	Нода 2	Активен	17.02.2026 12:37:14	...
Сервис обновления н	Нода 1	Активен	17.02.2026 12:37:14	...

Рис. 3.183. Отключение (остановка) сервиса

Для включения сервиса аналогично следует нажать на кнопку «....» и затем «Отправить запрос на запуск сервиса» (рис. 3.184).



Имя сервиса	Нода	Статус	Время обновления	Действие
Сервис имитации сиг		Выключен		...
Сервис генерации со	Нода 1	Активен	04.03.2026 19:00:57	...
Сервис архивации	Нода 1	Активен	04.03.2026 19:00:57	...

Рис. 3.184. Кнопка действия

Либо воспользоваться кнопкой запуска через боковое окно параметров (рис.3.185).

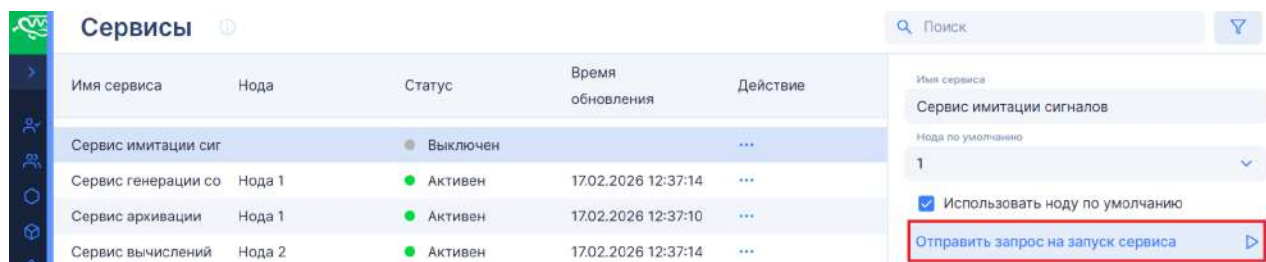


Рис. 3.185. Отправка запроса на запуск сервиса

После чего сервис запустится на выбранной ноде по умолчанию, а значок статуса станет «Активный» (рис. 3.186).

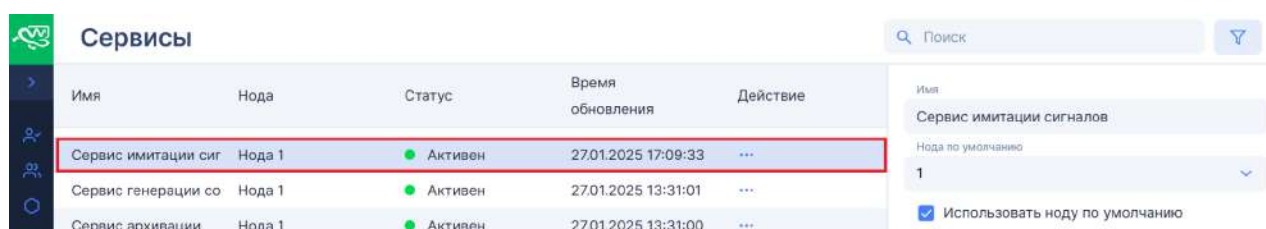


Рис. 3.186. Запуск сервиса

Список сервисов можно отфильтровать по их статусу и ноде при запуске. Для этого следует воспользоваться соответствующей кнопкой в верхней панели параметров (рис. 3.187).

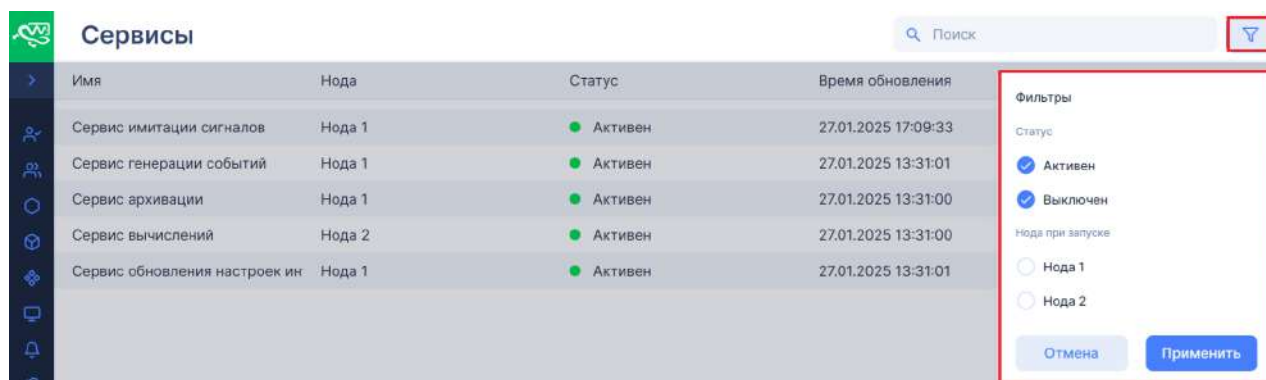


Рис. 3.187. Фильтрация списка сервисов

3.8.2. Сервис расчетных сигналов

Калькулятор (сервис расчетных сигналов) — производит вычисления на основе данных, получаемых из модуля **PB** по расчётным формулам, которые определены пользователем заранее и передаёт полученные результаты в модуль реального времени.



Рис. 3.187. Сервис расчетных сигналов

3.8.3. Сервис обработки событий

Генератор событий — на основе полученных из модуля реального времени значений и при выполнении заранее определенных условий генерирует управляющие команды, в рамках Платформы именуемые «событиями», которые используются для отображения происходящих в реальной системе процессов.



Рис. 3.188. Сервис обработки событий

3.8.4. Сервис управления архивом

Архиватор значений (сервис архивации) — служит для передачи данных из базы данных в оперативный и долгосрочный архивы.



Рис. 3.189. Сервис архивации

3.8.5. Сервис имитации данных

Имитатор (сервис имитации значений объектов) — используется как на стадии проектирования/отладки Системы для имитации реальных технологических процессов, так и на стадии эксплуатации, например, при проведении технического обслуживания отдельных элементов/объектов Системы.



Рис. 3.190. Сервис имитации значений объектов

3.8.6. Сервис обновления настроек интеграции

Сервис обновления настроек интеграции — это компонент Системы, предназначенный для изменения и актуализации параметров подключения между Системой и сервисами интеграций. Он отвечает за обновление настроек интеграций без перезагрузки Системы.

Сервис обновления наст	Нода 1	Активен	13.10.2025 20:57:45	...
------------------------	--------	---------	---------------------	-----

Рис. 3.191. Сервис обновления настроек интеграции

3.8.7. Сервис создания резервных копий

Сервис создания резервных копий - используется для планового создания резервных копий мета бд.

Сервис создания рез	Нода 1	Активен	17.03.2026 08:57:06	...
---------------------	--------	---------	---------------------	-----

Рис. 3.192. Сервис обновления резервных копий

3.9. Кластер

Кластерная модель подробно описана в разделе 2.7.

Для перехода в раздел «Кластер» необходимо воспользоваться боковой панелью меню, нажав на соответствующую кнопку. На странице раздела отображается имя ноды (сервера), IP-адрес узла (в том числе и текущий, с которого происходит обращение) и его статус (активен/выключен, рис. 3.191).

Кластер				
Имя ноды	IP-адрес	Текущий	Статус	
Нода 1	91.201.41.134	Да	Активен	
Нода 2	91.201.41.135	Нет	Активен	

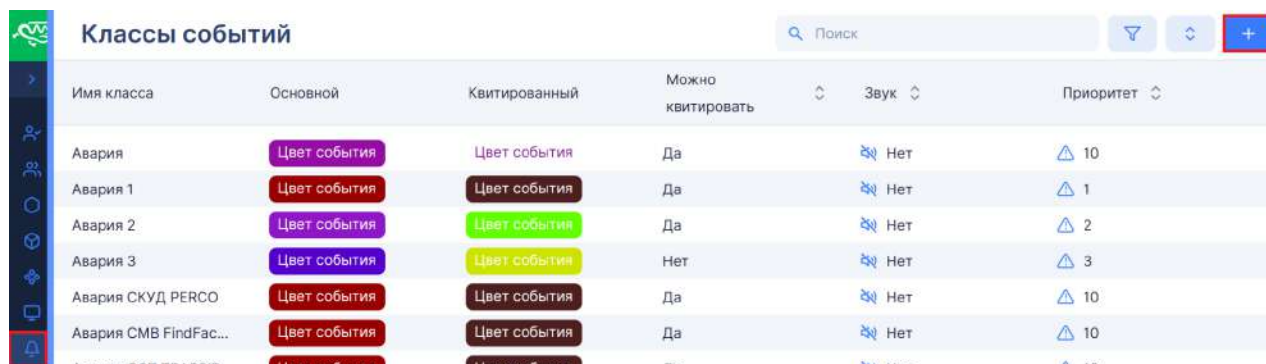
Рис. 3.191. Вкладка «Кластер»

3.10. События

Платформа позволяет создавать классы событий (удалять, редактировать и распределять их по группам) и наборы событий (для моделей Boolean и Integer).

3.10.1. Управление классами событий

Для перехода в раздел «Классы событий» необходимо воспользоваться боковой панелью меню, нажав на соответствующую кнопку (рис. 3.192).

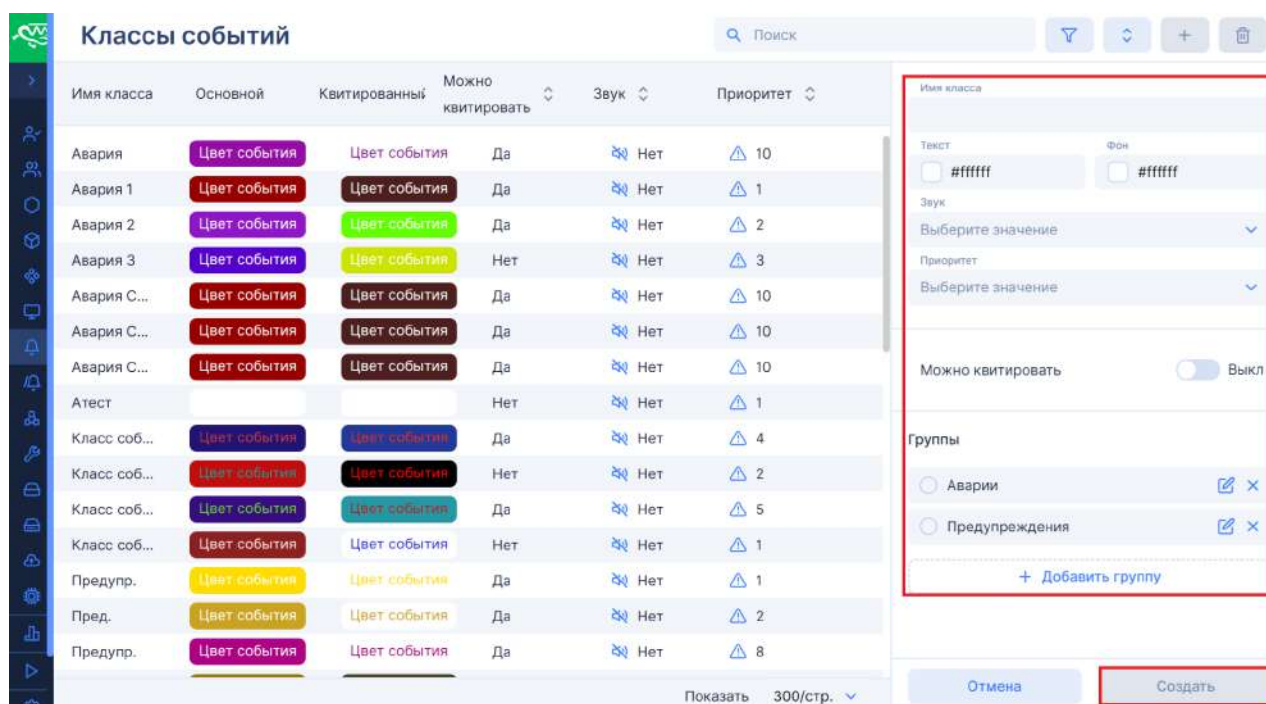


Имя класса	Основной	Квитированный	Можно квитировать	Звук	Приоритет
Авария	Цвет события	Цвет события	Да	Нет	10
Авария 1	Цвет события	Цвет события	Да	Нет	1
Авария 2	Цвет события	Цвет события	Да	Нет	2
Авария 3	Цвет события	Цвет события	Нет	Нет	3
Авария СКУД PERCO	Цвет события	Цвет события	Да	Нет	10
Авария CMB FindFac...	Цвет события	Цвет события	Да	Нет	10
Авария COT TRASSIR	Цвет события	Цвет события	Да	Нет	10

Рис. 3.192. Классы событий

Для того чтобы создать класс события, следует нажать на кнопку «Создать» в правом верхнем углу страницы (рис. 3.192).

После чего справа откроется вкладка в которой потребуется ввести имя класса события, выбрать цвет текста и фона, выбрать звук события, а также значение приоритета и возможность подтверждения класса события с выбором цвета и фона подтверждения. Завершающим этапом создания будет добавление (для того чтобы добавить новую группу необходимо нажать на кнопку «+ Добавить группу») или выбор группы класса события. По завершении нажать кнопку «Создать» (рис. 3.193).



Имя класса	Основной	Квитированный	Можно квитировать	Звук	Приоритет
Авария	Цвет события	Цвет события	Да	Нет	10
Авария 1	Цвет события	Цвет события	Да	Нет	1
Авария 2	Цвет события	Цвет события	Да	Нет	2
Авария 3	Цвет события	Цвет события	Нет	Нет	3
Авария С...	Цвет события	Цвет события	Да	Нет	10
Авария С...	Цвет события	Цвет события	Да	Нет	10
Авария С...	Цвет события	Цвет события	Да	Нет	10
Атест			Нет	Нет	1
Класс соб...	Цвет события	Цвет события	Да	Нет	4
Класс соб...	Цвет события	Цвет события	Нет	Нет	2
Класс соб...	Цвет события	Цвет события	Да	Нет	5
Класс соб...	Цвет события	Цвет события	Нет	Нет	1
Предупр.	Цвет события	Цвет события	Да	Нет	1
Пред.	Цвет события	Цвет события	Да	Нет	2
Предупр.	Цвет события	Цвет события	Да	Нет	8

Рис. 3.193. Вкладка с вводом

Для того чтобы отредактировать группу событий необходимо нажать на соответствующую кнопку напротив группы и по завершении нажать на галочку для сохранения изменений (рис. 3.194).

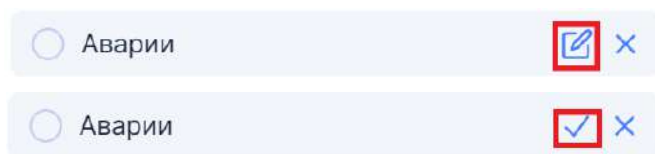


Рис. 3.194. Редактирование группы событий

Для того чтобы добавить новую группу событий, необходимо нажать на кнопку «+ Добавить группу», после чего в пустую строку задать название и нажать на галочку по завершении, тем самым сохранив новую группу событий. (рис. 3.195).

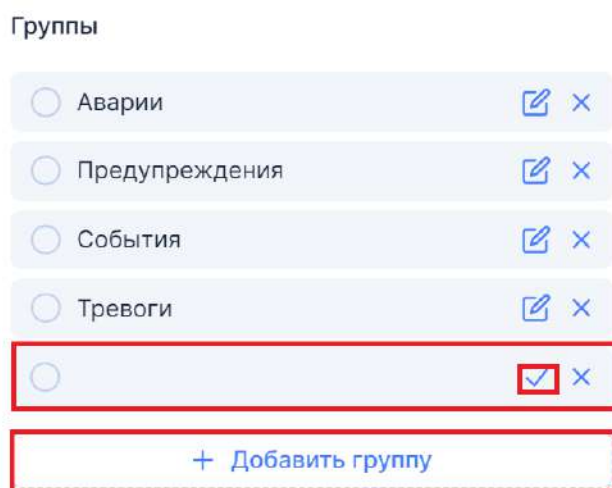


Рис. 3.195. Добавление новой группы событий

Для того чтобы отредактировать класс события, необходимо нажать на на соответствующий класс события. После чего справа появится вкладка с параметрами, где можно изменить имя и выбрать нужный к нему цвет текста и фона. Также выбрать возможность удаления/возможность подтверждения (с выбором цвета и фона подтверждения) и выбрать (или добавить новые) группы классу события и приоритет. Для сохранения изменений необходимо нажать «Сохранить» (рис. 3.196).

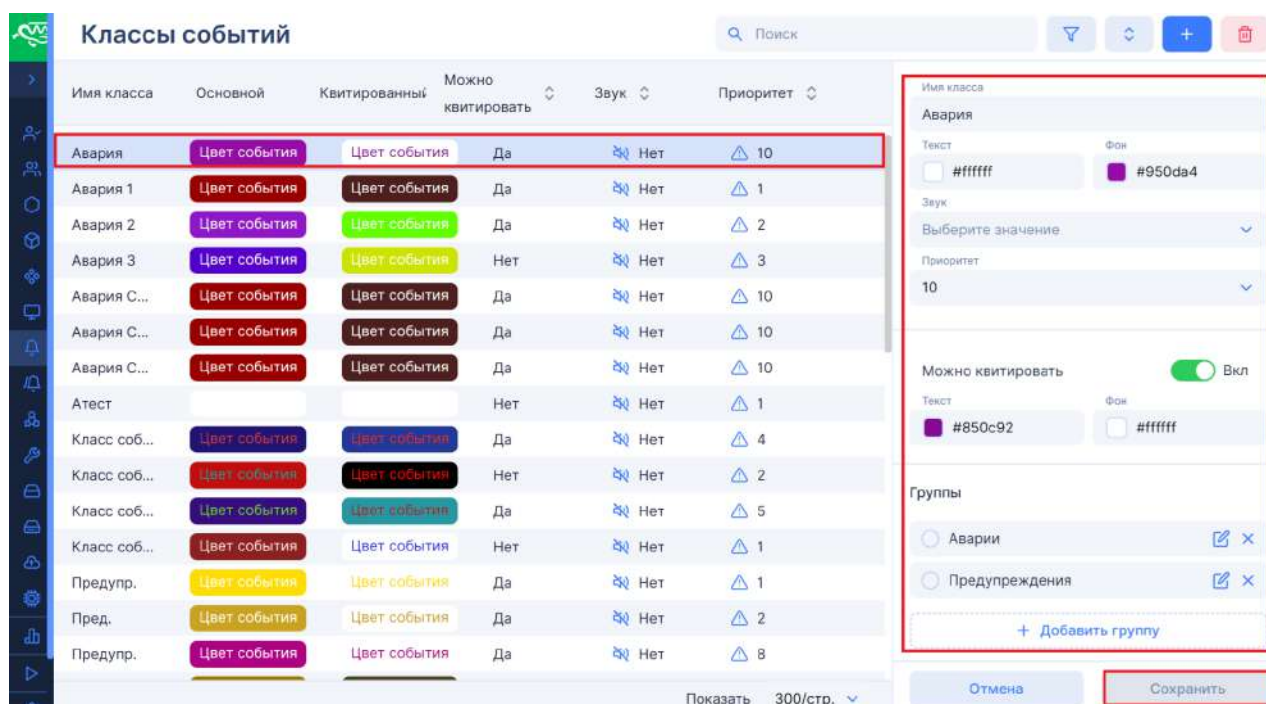


Рис. 3.196. Редактирование класса события

Для того чтобы удалить класс события, следует нажать на соответствующий класс события, после чего в правом верхнем углу появится кнопка удаления на которую необходимо будет нажать (рис. 3.197).

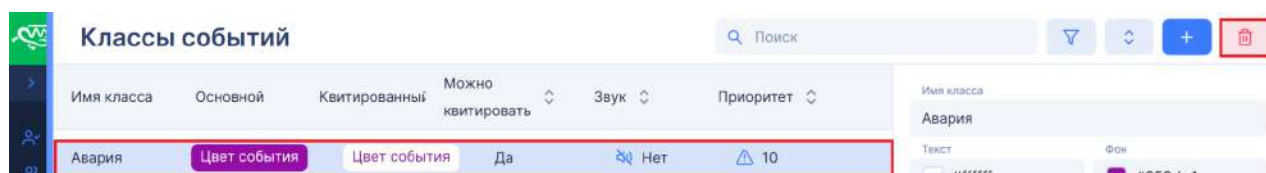


Рис. 3.197. Удаление класса события

После нажатия на кнопку возникнет окно с подтверждением необходимости удалить класс события (рис. 3.198).

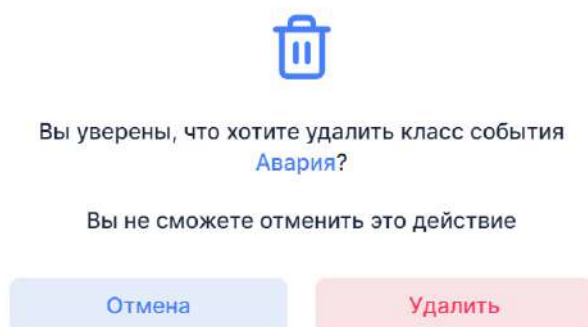


Рис. 3.198. Окно с подтверждением удаления

После повторного нажатия «Удалить» класс события будет удален.

У некоторых классов событий присутствует ограничение на изменение и удаление, напротив таких классов имеется соответствующий значок при наведении на который выходит предупреждение (рис. 3.186). Подобное предупреждение будет также отображаться и в боковой вкладке параметров, при нажатии на класс события (рис. 3.199).

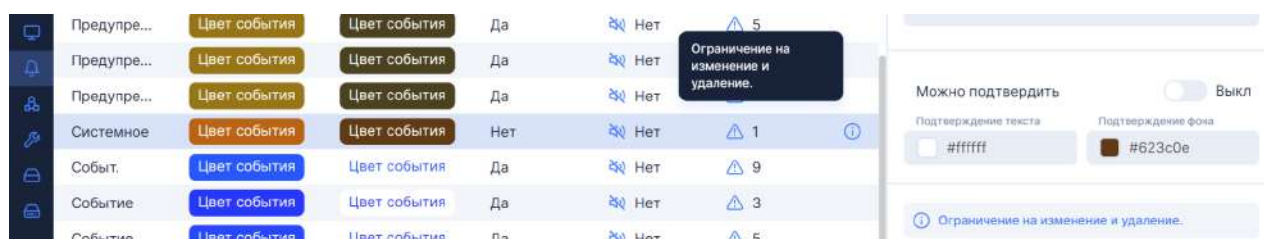


Рис. 3.199. Предупреждение об ограничениях

Пользователь имеет возможность фильтровать классы событий по наличию звука и возможности квитиования. Для этого необходимо нажать на соответствующую кнопку в правом верхнем углу панели (рис. 3.200).

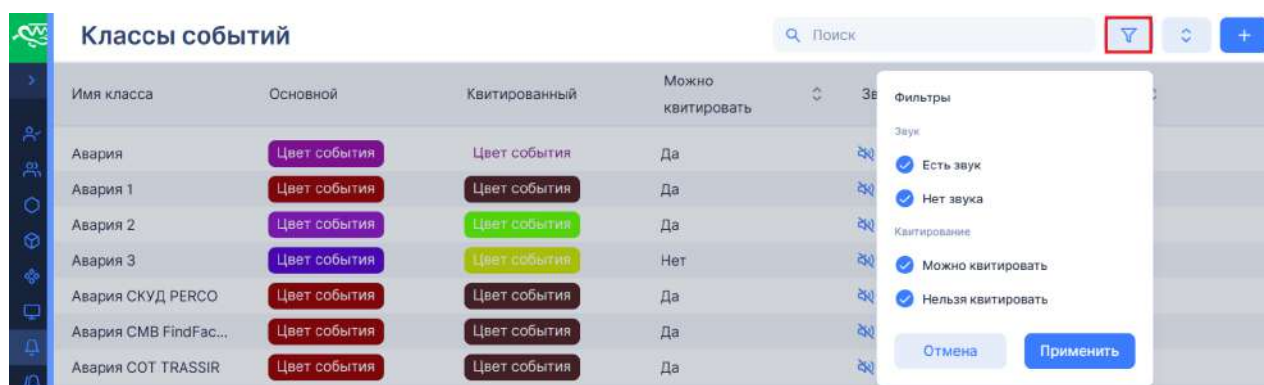


Рис. 3.200. Фильтрация классов событий

Классы событий можно отсортировать по возможности квитирования, звуку, приоритету. Для этого нужно воспользоваться кнопкой сортировки на верхней панели (рис. 3.201).

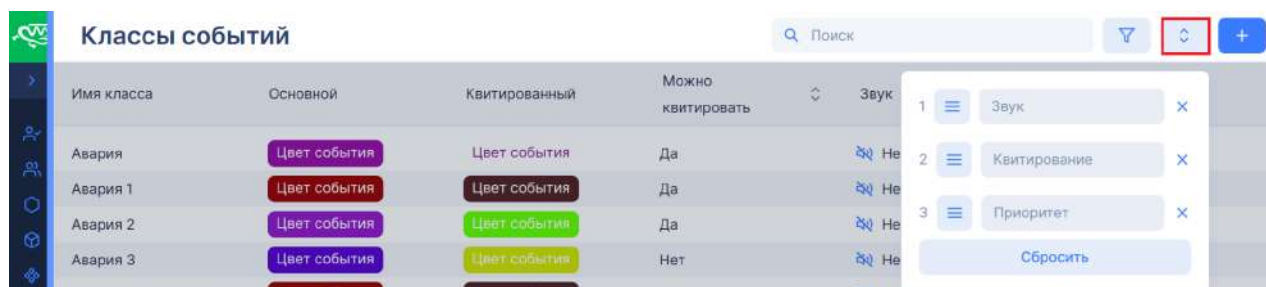


Рис. 3.201. Сортировка классов событий

Добавить сортировку можно либо нажатием кнопки «+ Добавить сортировку» (рис. 3.201), либо выбрать в окне параметра сортировки (рис. 3.202).

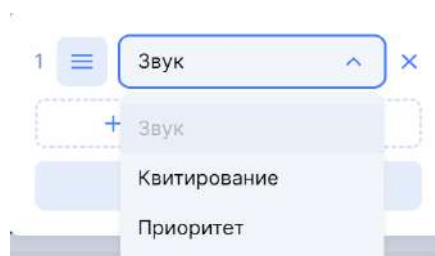


Рис. 3.202. Выбор параметра сортировки

Приоритет сортировки можно менять между собой местами с помощью перетаскивания (Drag-and-drop). Для этого необходимо нажать на соответствующую иконку напротив выбранного параметра и оттащить ее вниз или вверх, изменив расположение (рис. 3.203).

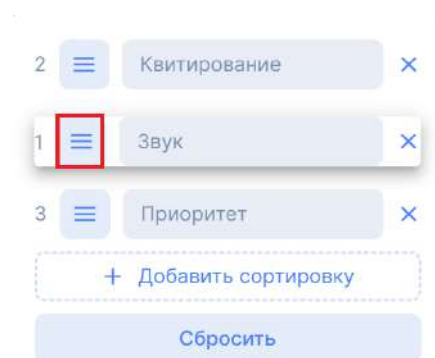


Рис. 3.203. Изменение приоритета сортировки

Аналогично можно воспользоваться сортировкой вручную напротив каждого из параметров (рис. 3.202).



Рис. 3.202. Сортировка вручную

Звуковые оповещения событий можно загрузить через подраздел «События» раздела «Настройки» бокового меню (рис. 3.200).

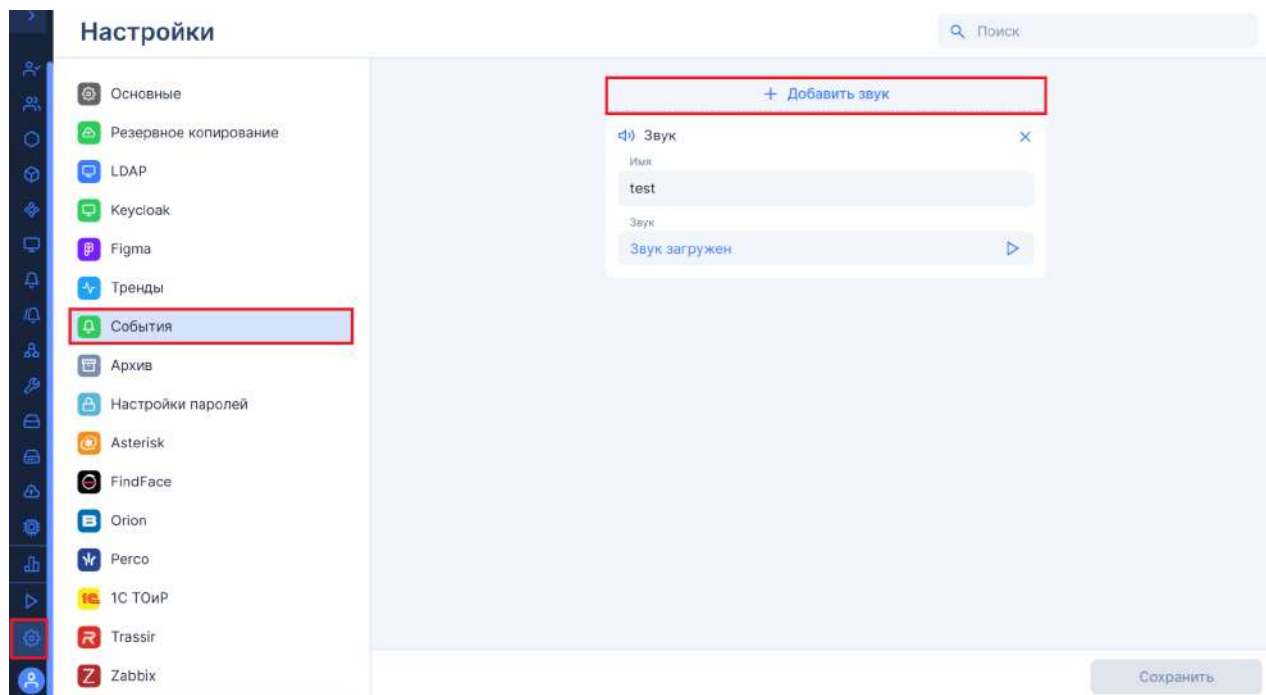
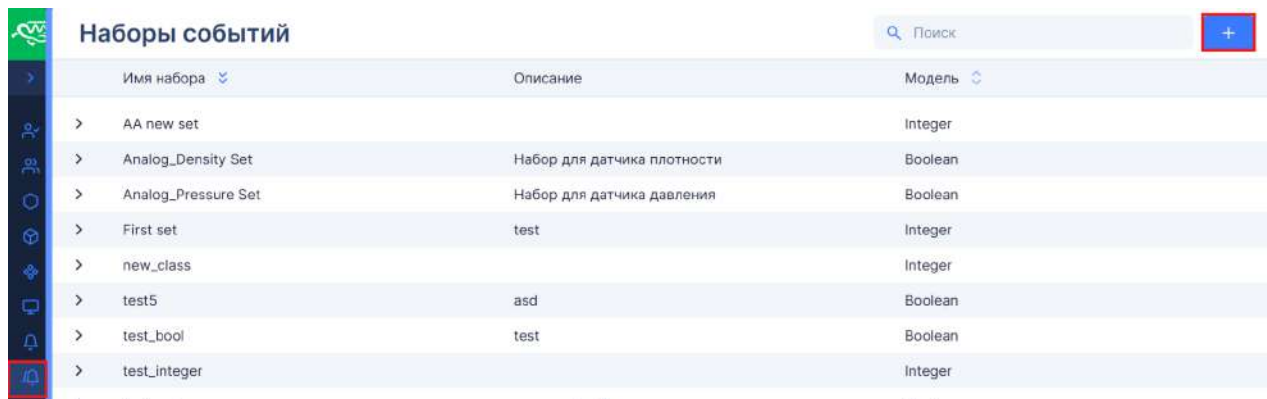


Рис. 3.200. Загрузка звуковых оповещений

Звук воспроизводится у события, класс которого обладает наивысшим приоритетом. Звук воспроизводится последовательно (с интервалом 1 секунда) до получения события с более высоким приоритетом или квитирования текущего.

3.10.2. Управление наборами событий

Для перехода в раздел «Наборы событий» необходимо воспользоваться боковой панелью меню, нажав на соответствующую кнопку (рис. 3.201).

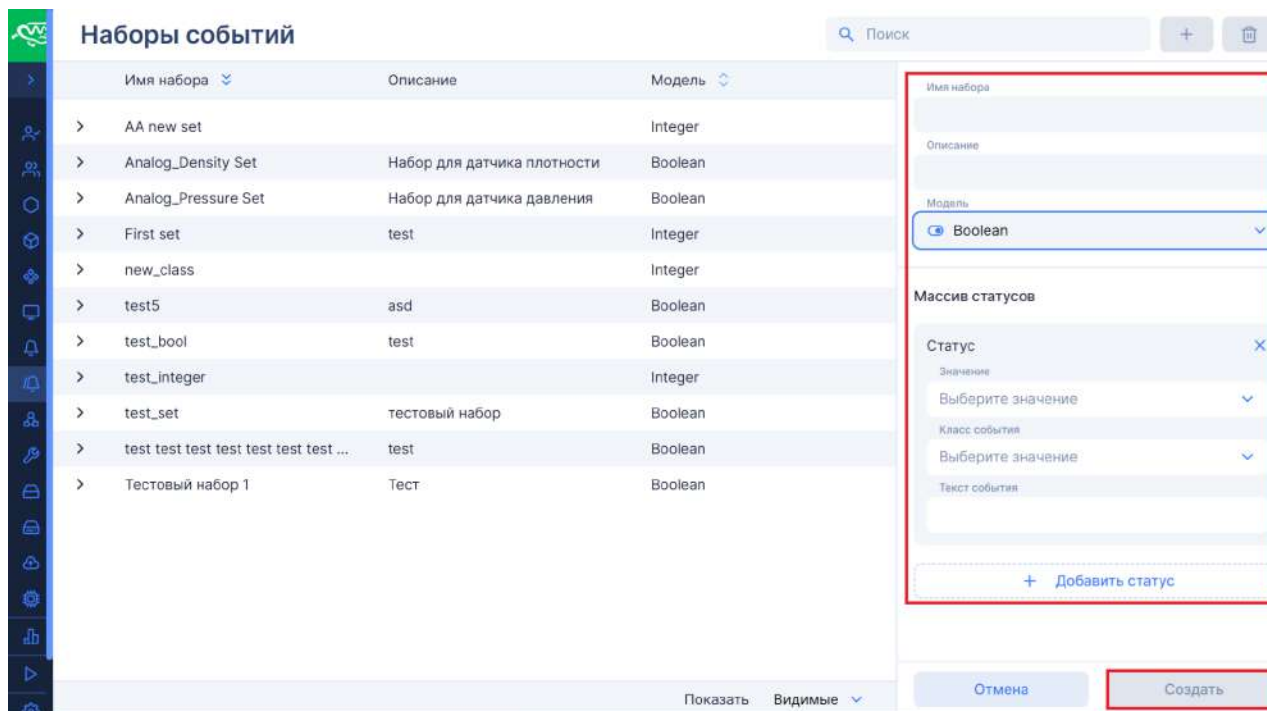


Имя набора	Описание	Модель
AA new set		Integer
Analog_Density Set	Набор для датчика плотности	Boolean
Analog_Pressure Set	Набор для датчика давления	Boolean
First set	test	Integer
new_class		Integer
test5	asd	Boolean
test_bool	test	Boolean
test_integer		Integer

Рис. 3.201. Наборы событий

Для того чтобы создать новый набор событий необходимо воспользоваться кнопкой «Создать» в правом верхнем углу страницы (рис. 3.201).

После чего справа откроется вкладка (рис. 3.202) в которой необходимо будет ввести имя, описание, выбрать модель (Integer или Boolean) и добавить статус в массив статусов (нажатием кнопки «+ Добавить статус»), задав параметры статуса (значение, класс события, текст события). Доступные значения зависят от выбранной модели. Например, для Boolean - true/false, а для Integer значение задается вручную строкой.



Имя набора	Описание	Модель
AA new set		Integer
Analog_Density Set	Набор для датчика плотности	Boolean
Analog_Pressure Set	Набор для датчика давления	Boolean
First set	test	Integer
new_class		Integer
test5	asd	Boolean
test_bool	test	Boolean
test_integer		Integer
test_set	тестовый набор	Boolean
test test test test test test ...	test	Boolean
Тестовый набор 1	Тест	Boolean

Имя набора	Описание	Модель
test_set	тестовый набор	Boolean

Массив статусов						
<table border="1"> <thead> <tr> <th>Статус</th> </tr> </thead> <tbody> <tr> <td>Значение</td> </tr> <tr> <td>Выберите значение</td> </tr> <tr> <td>Класс события</td> </tr> <tr> <td>Выберите значение</td> </tr> <tr> <td>Текст события</td> </tr> </tbody> </table>	Статус	Значение	Выберите значение	Класс события	Выберите значение	Текст события
Статус						
Значение						
Выберите значение						
Класс события						
Выберите значение						
Текст события						
+ Добавить статус						

Показать Видимые

Отмена Создать

Рис. 3.202. Параметры набора событий

Наборы событий можно отсортировать по имени и модели. Для этого необходимо воспользоваться соответствующей кнопкой напротив каждого из параметров (рис. 203).



Рис. 3.203. Фильтрация наборов событий

Также доступен поиск наборов событий по ключевым символам из названия и описания (рис. 3.203).

Пользователь может раскрыть массив статусов у набора событий, не нажимая на сам набор. Для этого необходимо нажать на кнопку «>», после чего отобразятся статусы (Рис. 3.204).

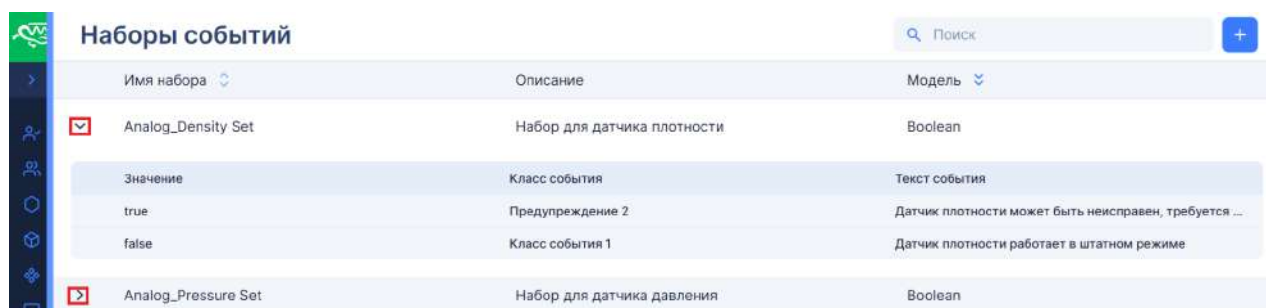


Рис. 3.204. Раскрытие массива статусов

Для того чтобы удалить набор событий, необходимо нажать на набор, после чего на кнопку «Удалить» в верхнем крайнем правом углу страницы (рис. 3.205).

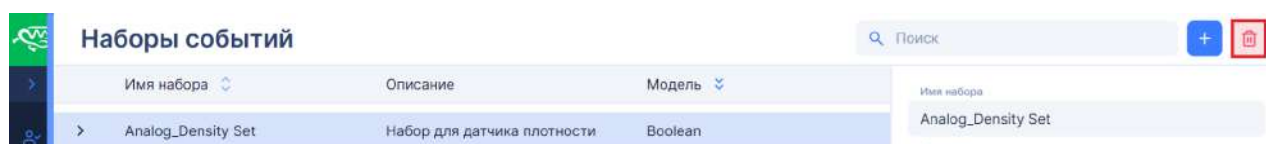


Рис. 3.205. Удаление набора событий

3.11. Пользователи и роли

Платформа предоставляет возможность создавать и редактировать профили пользователей и их роли, а также управлять статусами пользователей (активен/приостановлен) и наделить роль соответствующими правами. В Платформе должен быть определён как минимум один профиль пользователя - «Администратор», который имеет максимальные права. Также может быть задан профиль пользователя «Бесправный» с отсутствием прав.

3.11.1. Настройка ролей

Для создания новой роли необходимо перейти на вкладку «Роли» нажав соответствующую кнопку в боковом меню и затем нажать на «+» в правом верхнем углу страницы (рис. 3.201).



Рис. 3.201. Добавление роли

В открывшийся вкладке справа в полях ввода необходимо ввести название роли (обязательный параметр), её описание, выбрать стартовую мнемосхему и список прав, предоставляемых роли. По окончании нажать «Создать» для сохранения созданной роли (рис. 3.202).

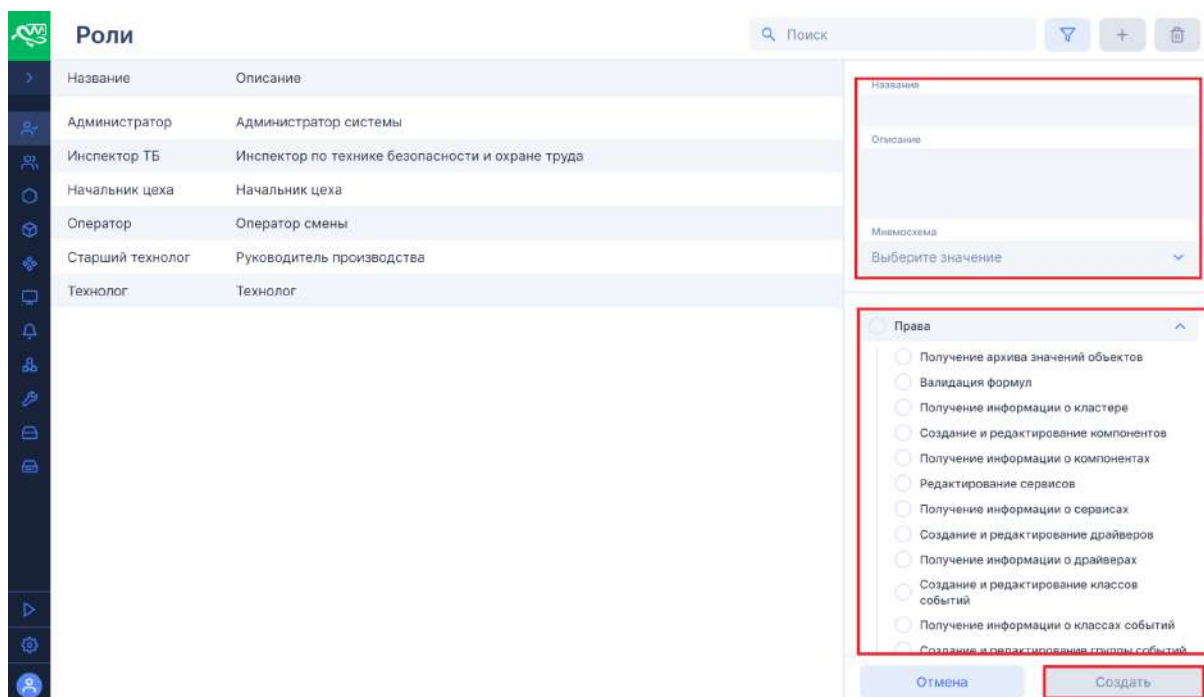


Рис. 3.202. Параметры новой роли

Для того чтобы сконфигурировать выбранную роль, нужно выбрать её из списка, после чего справа появится вкладка в которой можно изменить название роли, её описание и переопределить права. По окончании для сохранения внесенных изменений нажать на кнопку «Сохранить» (рис. 3. 203).

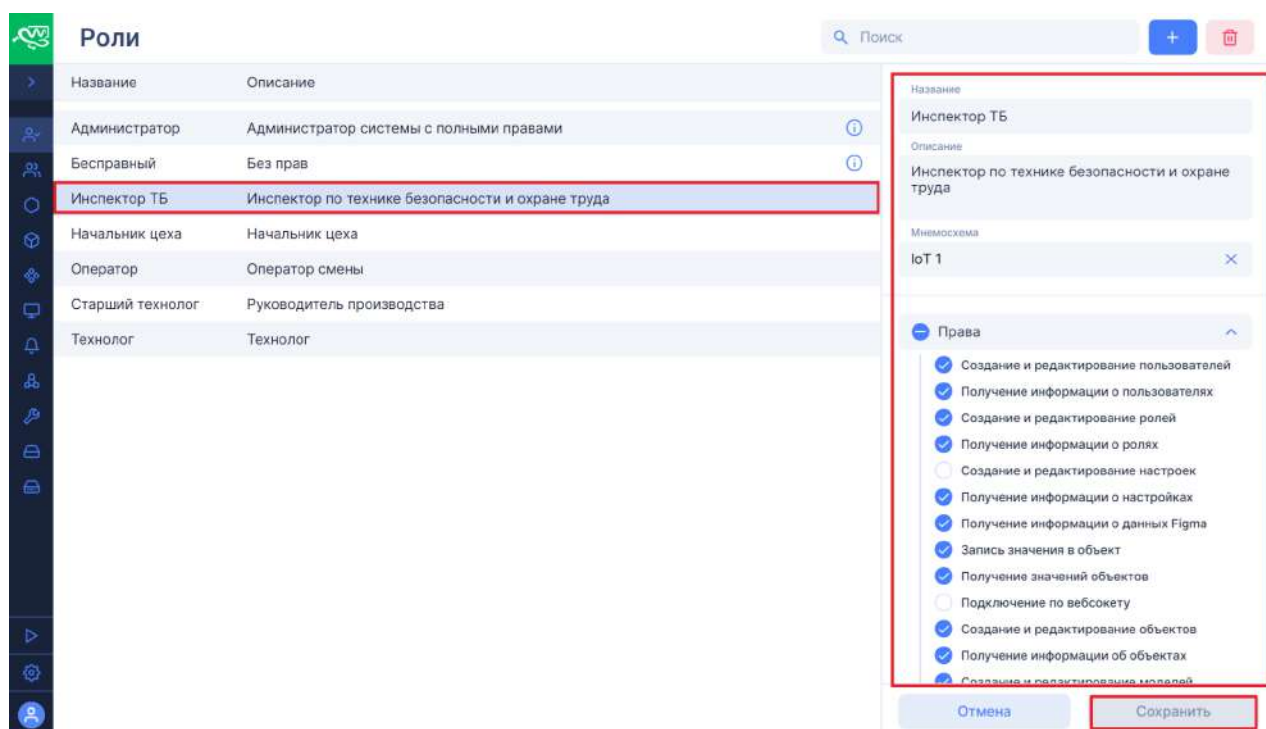


Рис. 3.203. Конфигурирование роли

Для удаления роли, нужно выбрать её из списка, затем нажать на кнопку удаления в правом верхнем углу и подтвердить это действие нажав «Удалить» в модульном окне (рис. 3. 204).



Рис. 3.204. Подтверждение удаления

Некоторые роли могут иметь ограничения. Об этом символизирует соответствующий значок при наведении на который выходит предупреждение. Подобное предупреждение также отображается и на боковой вкладке, при нажатии на роль (рис. 3.205).



Рис. 3.205. Предупреждение об ограничениях

Пользователь имеет возможность продублировать роль, нажав на соответствующую кнопку (рис. 3.206).



Рис. 3.206. Дублирование роли

3.11.2. Настройка пользователей

Система позволяет создавать и конфигурировать пользователей.

Для создания нового пользователя необходимо нажать на кнопку «Создать» в правом верхнем углу. Откроется вкладка справа, в которой необходимо будет ввести данные нового пользователя (имя, логин, email), назначить ему роль и выбрать мнемосхему, которая будет отображаться при входе пользователя в систему (рис. 3.207).

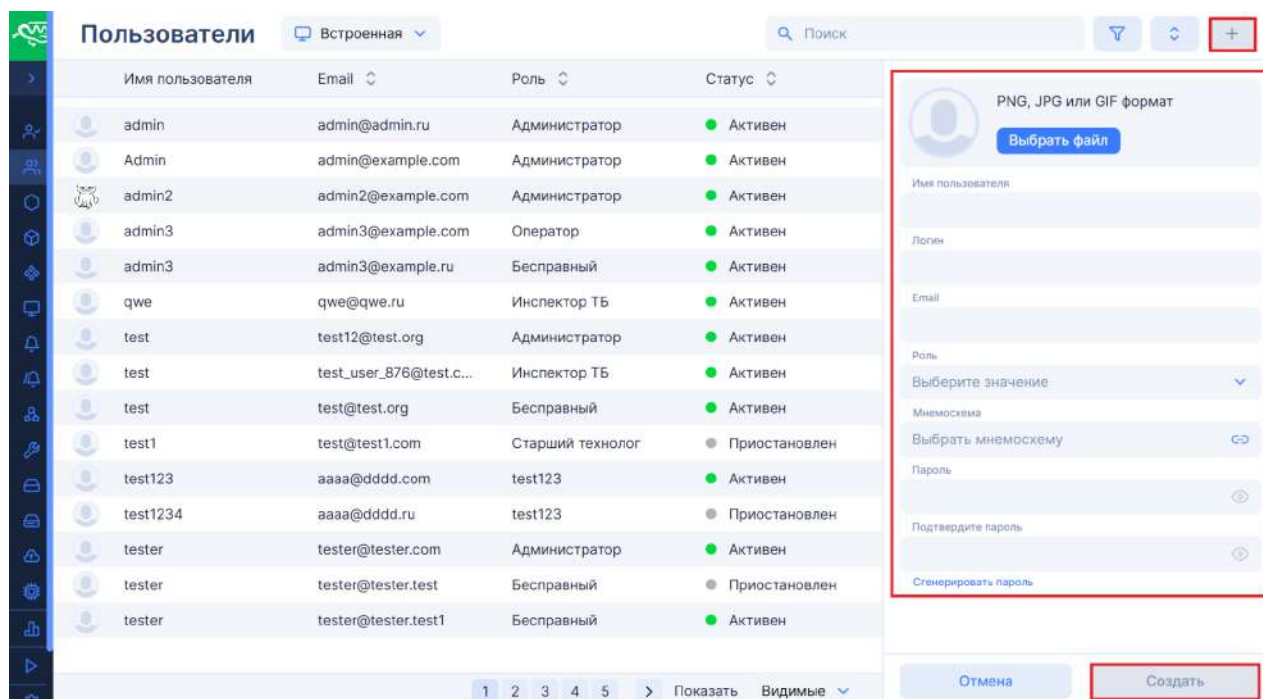


Рис. 3.207. Создание нового пользователя

Далее нужно задать пароль (система позволяет сгенерировать новый пароль автоматически, для этого необходимо нажать на кнопку «Сгенерировать временный пароль» под формой ввода пароля) и выбрать «Систему управления пользователями». По окончании сохранить параметры нового пользователя кнопкой «Создать» (рис. 3.207).

В разделе «Настройки» у пользователя есть возможность задать параметры для создания паролей пользователей (рис. 3.208).

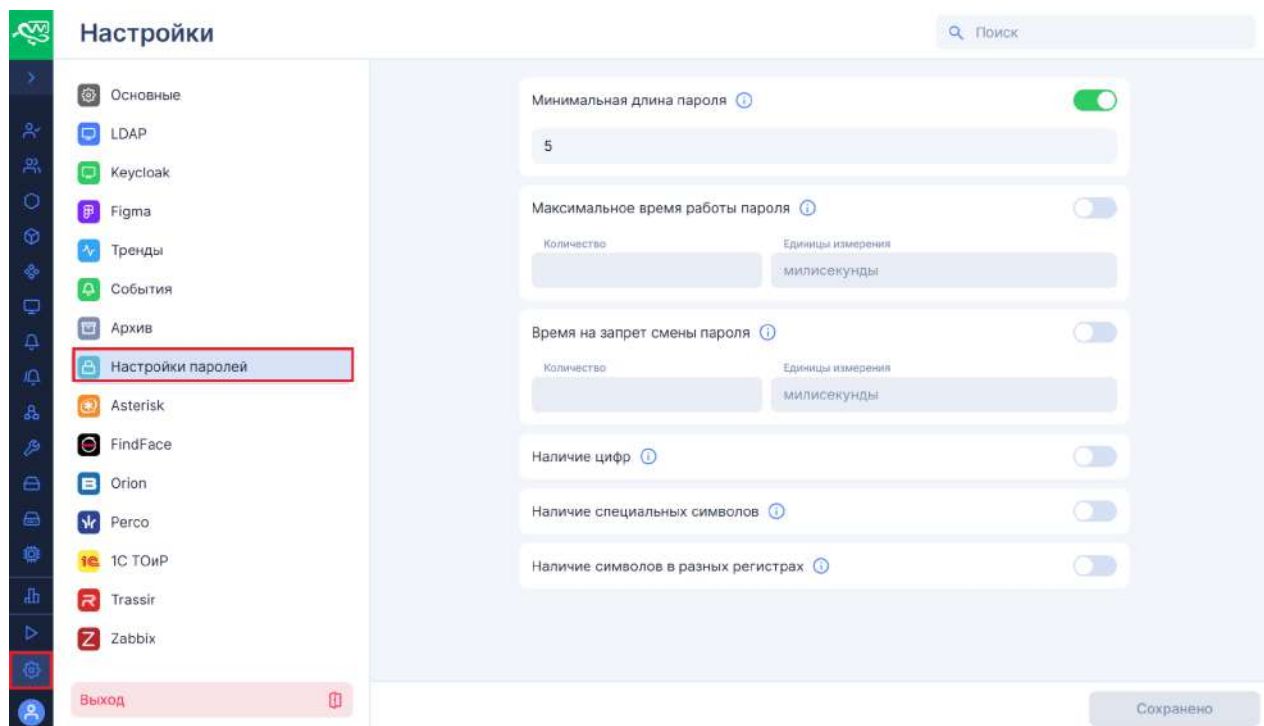


Рис. 3.208. Настройки паролей

Настройки паролей позволяют выбрать его минимальную длину, максимальное время работы, время на запрет его смены, наличие цифр и специальных символов, наличие символов в разных регистрах.

В случае, если введенный пароль не будет соответствовать какой-либо из настроек, то пользователь увидит следующее сообщение (рис. 3. 209).

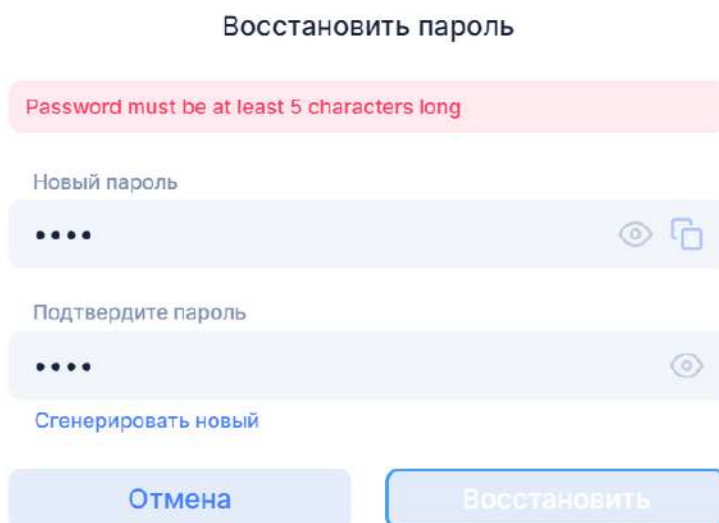


Рис. 3.209. Сообщение о несоответствие настройкам

Для редактирования пользователей, необходимо нажать на пользователя из списка, после чего в открывшейся вкладке справа внести необходимые параметры и нажать «Сохранить» (рис. 3.210).

Пользователи Встроенная

Имя пользователя Email Роль Статус

admin	admin@admin.ru	Администратор	Активен
Admin	admin@example.com	Администратор	Активен
admin2	admin2@example.com	Администратор	Активен
admin3	admin3@example.com	Оператор	Активен
admin3	admin3@example.ru	Бесправный	Активен
qwe	qwe@qwe.ru	Инспектор ТБ	Активен
test	test12@test.org	Администратор	Активен
test	test_user_876@test.c...	Инспектор ТБ	Активен
test	test@test.org	Бесправный	Активен
test1	test@test1.com	Старший технолог	Приостановлен
test123	aaaa@dddd.com	test123	Активен
test1234	aaaa@dddd.ru	test123	Приостановлен
tester	tester@tester.com	Администратор	Активен
tester	tester@tester.test	Бесправный	Приостановлен
tester	tester@tester.test1	Бесправный	Активен

Правая панель редактирования:

- Имя пользователя: admin
- Логин: Aadmin
- Email: admin@admin.ru
- Роль: Администратор
- Мнемосхема: Выбрать мнемосхему
- Пароль: Восстановить пароль
- Система управления пользователями: Встроенная
- Токены: Отмена, Сохранить

Рис. 3.210. Конфигурирование пользователя

Пользователю может быть присвоено изображение (аватар) в формате PNG, JPG, GIF. Для этого нужно нажать кнопку «Выбрать файл» (рис. 3.211) и загрузить соответствующее изображение.



Рис. 3.211. Загрузка аватара пользователя

Пользователь может создавать токены (рис. 3.212) с помощью которых выбирать наделяемые ему права (доступный набор прав зависит от роли пользователя) в системе и срок их действия.

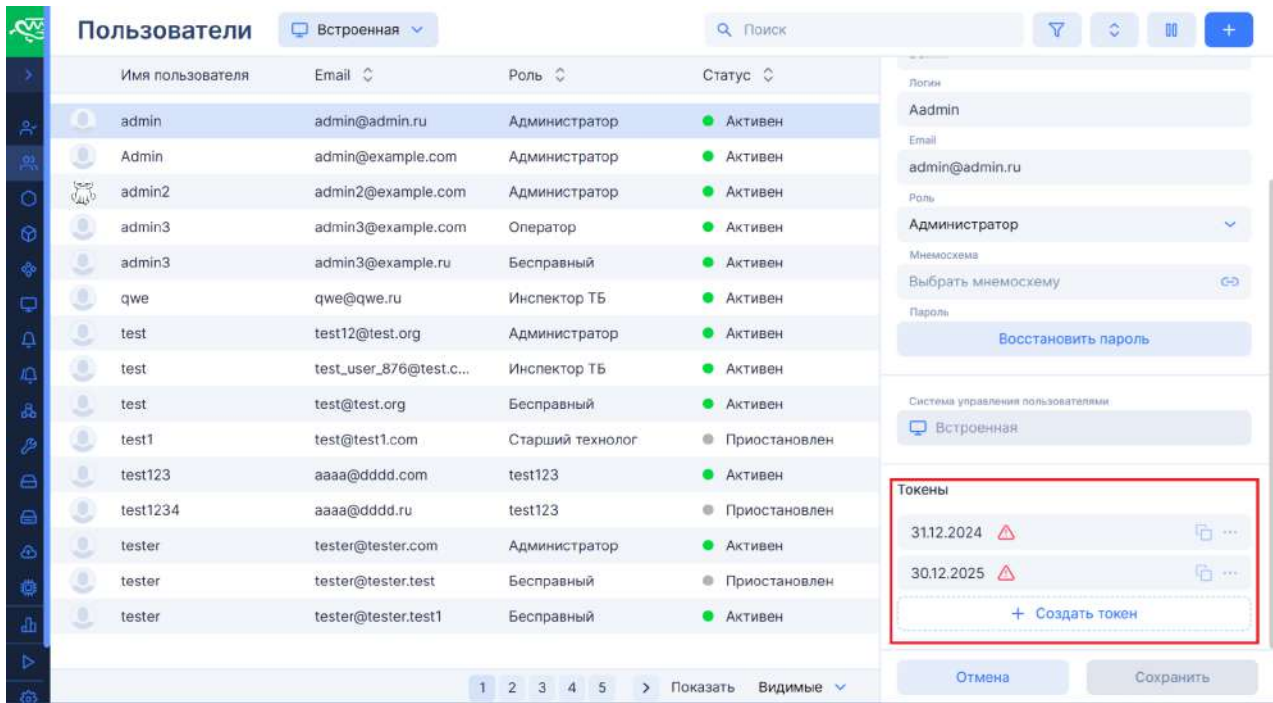


Рис. 3. 212. Пользовательские токены

Токены также можно копировать нажатием соответствующей кнопки (рис. 3.213).

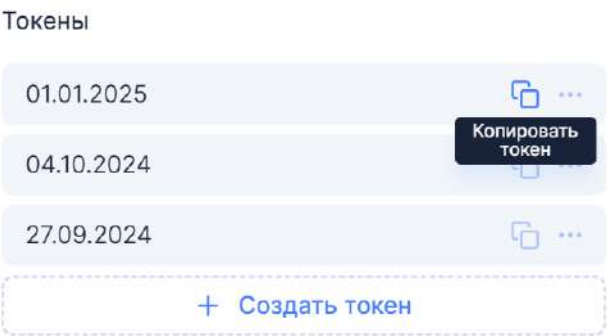


Рис. 3.213. Копирование токена

В поле «Токены» будет отображаться дата до которой действует каждый из них, при наведение на которую отобразится сам токен (рис. 3.214).

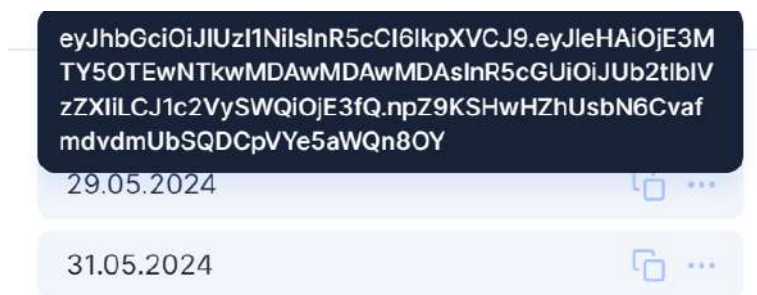


Рис. 3.214. Отображение токена

Наделяемые права можно будет отредактировать, а срок действия устанавливается единоразово при создании токена и больше не изменяется.

Напротив токена с истекшим сроком действия будет отображаться соответствующий значок при наведении на который высветится надпись «Срок токена истек» (рис. 3.215).

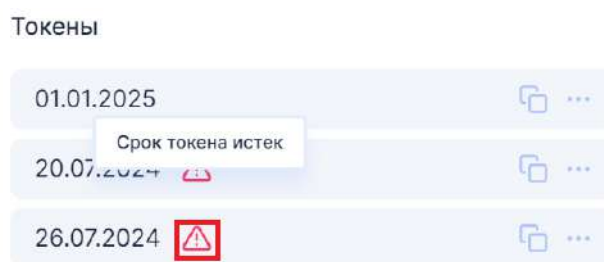


Рис. 3.215. Индикация токена с истекшим сроком действия

Для того чтобы отредактировать токен необходимо нажать на «...» напротив токена (Также имеется возможность удалить токен нажатием «Удалить»), затем на «Изменить», после чего откроется модальное окно редактора (рис. 3.216, 3.217).

Токены

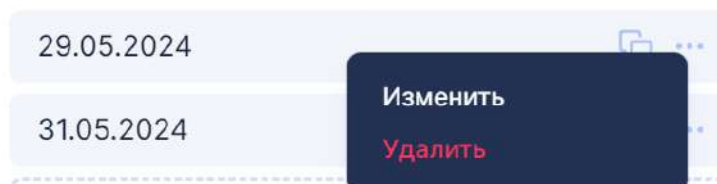


Рис. 3.216. Редактирование/Удаление пользовательского токена

Редактирование токена

Токен

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleH
AiOiE3MTY5OTEwNTkwMDAwMDAwMDAsInR
5cGUOiJUb2t1bIVzZXIIiLCJ1c2VySWQiOiE3fQ.
npZ9KSHwHZhUsbN6CvafmdvdmUbSQDCpV
Ye5aWQn8OY

Дата действия токена

29.05.2024

Права

☒ Получение архива значений объектов
☒ Валидация формул
☐ Получение информации о кластере
☐ Создание и редактирование компонентов
☐ Получение информации о компонентах
☐ Редактирование сервисов
☐ Получение информации о сервисах
☐ Создание и редактирование драйверов
☐ Получение информации о драйверах
☐ Создание и редактирование классов событий

Отмена

Сохранить

Рис. 3.217. Редактирование пользовательского токена

Пользователь может управлять статусом пользователей. Для того чтобы приостановить статус пользователя, необходимо выбрать пользователя в списке и после чего нажать на появившуюся кнопку остановки в правом верхнем углу (рис. 3.218). Данная возможность отсутствует у роли текущего пользователя под которым была осуществлена авторизация.

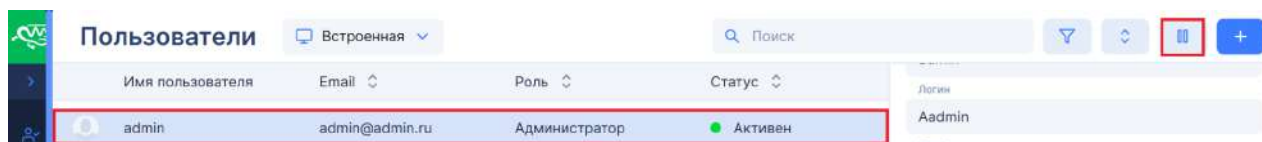


Рис. 3.218. Приостановка статуса пользователя

Аналогично будет работать и запуск статуса пользователя. Для того чтобы вновь запустить статус пользователя нужно выбрать пользователя в списке и нажать на кнопку запуска (рис. 3.219).



Рис. 3.219. Запуск статуса пользователя

Список пользователей можно отфильтровать по их статусу (Активен/Приостановлен) и роли. Для этого следует нажать на кнопку фильтра на панели параметров (рис. 3.220).

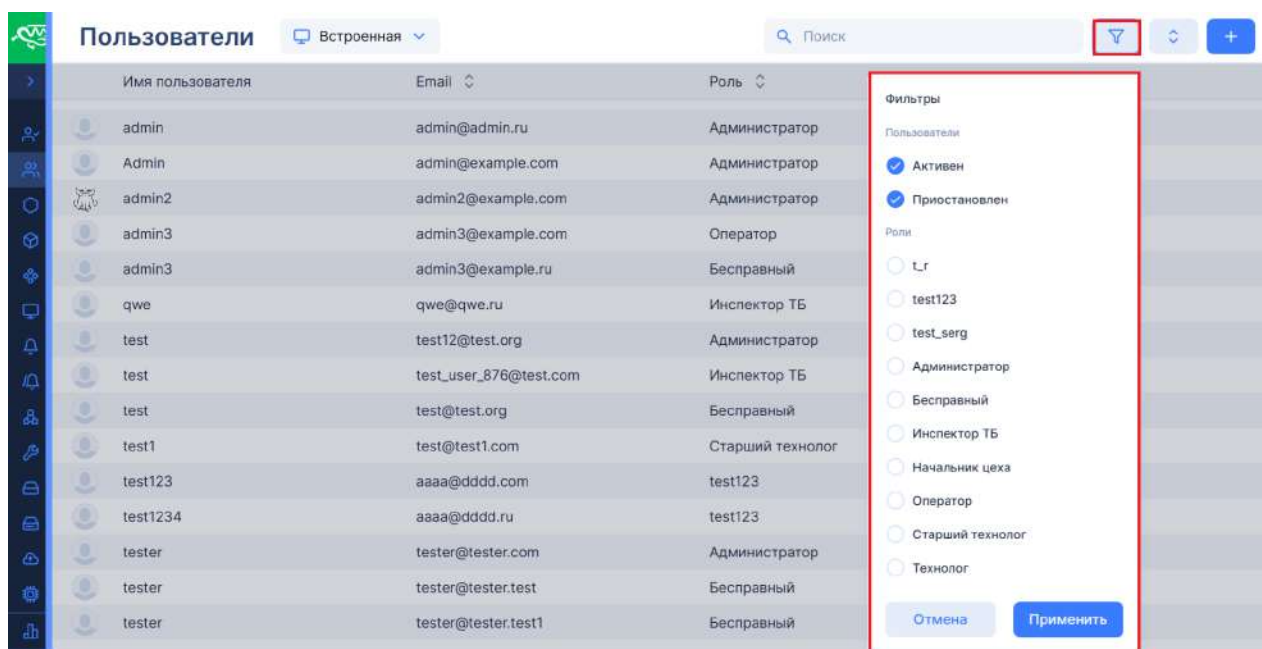


Рис. 3.220. Фильтрация пользователей

Список пользователей также можно отсортировать по Email, роли, статусу. Для этого нужно нажать на кнопку сортировки в верхней панели и нажать кнопку «+ Добавить сортировку» (рис. 3.221).



Рис. 3.221. Сортировка списка пользователей

Либо выбрать в окне параметра сортировки (рис. 3.222).

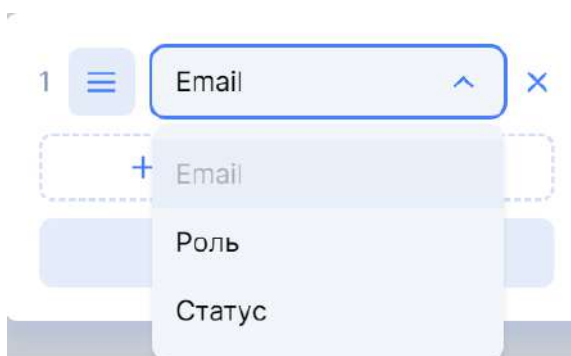


Рис. 3.222. Сортировка через окно параметров

Приоритет сортировки можно менять между собой местами с помощью перетаскивания (Drag-and-drop). Для этого необходимо зажать соответствующую пиктограмму напротив выбранного параметра и оттащить его вниз или вверх, изменив расположение (рис. 3.223).

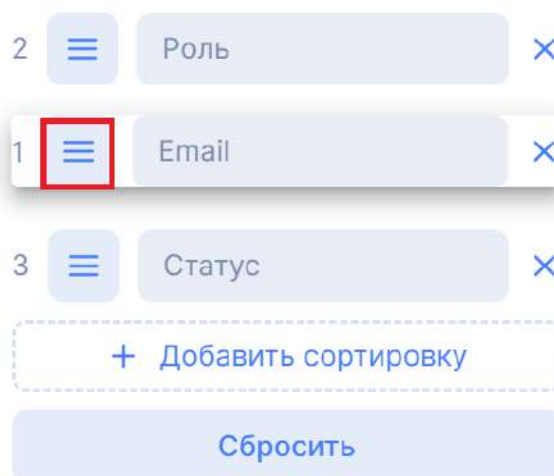


Рис. 3.223. Изменение приоритета сортировки

Также можно воспользоваться сортировкой вручную напротив каждого из параметров (рис. 3.224).



Рис. 3.224. Сортировка параметров вручную

3.12. Личный кабинет

Личный кабинет позволяет пользователю редактировать его данные. В нём можно добавить аватар пользователя, изменить его имя, логин, email, изменить пароль (Или автоматически сгенерировать новый пароль), также выбрать мнемосхему, создать и отредактировать пользовательские токены

Для перехода в раздел Личный кабинет необходимо нажать на иконку юзерпика в левом боковом меню (рис. 3.219).

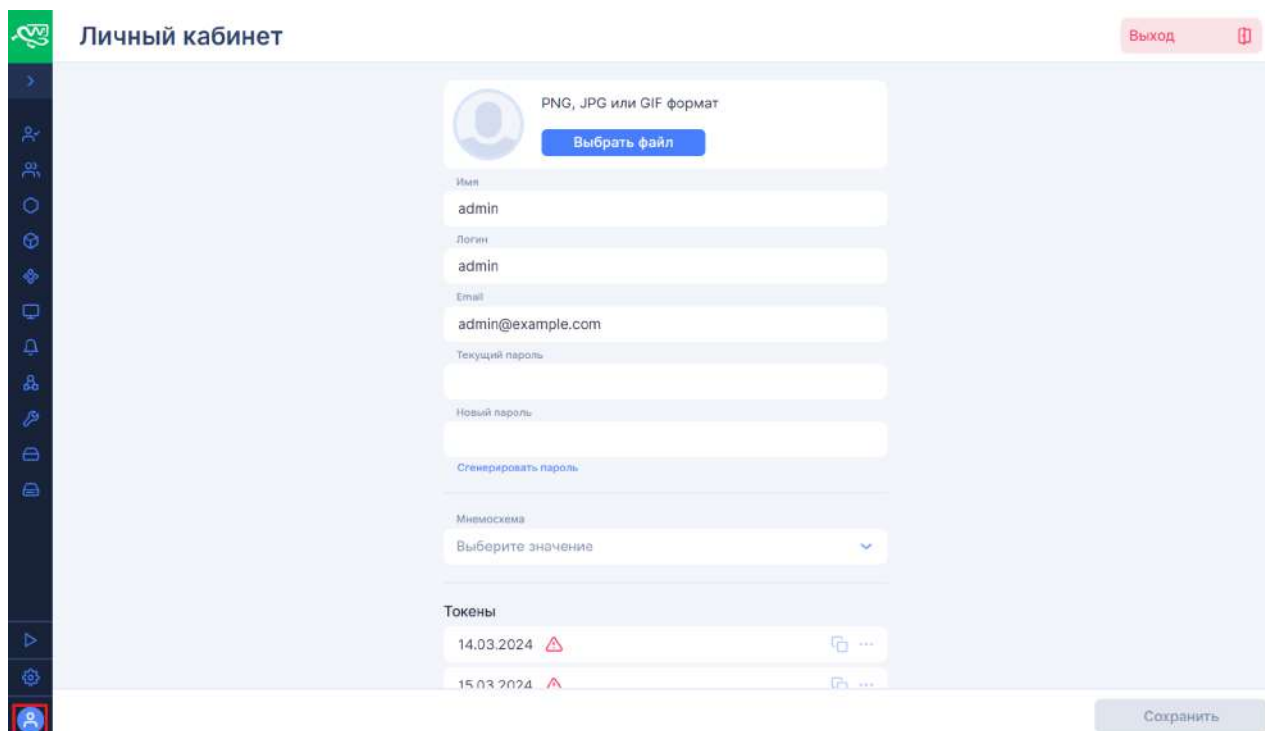


Рис. 3.219. Личный кабинет

3.13. Статистика

Данный раздел платформы предоставляет пользователю обширную информацию о состоянии памяти (доступный и занятый объем), распределении данных в архиве, а также о проекте (количество объектов, моделей, событий, пользователей, мнемосхем, драйверов, задействованных в Системе).

Для перехода на страницу Статистика необходимо воспользоваться боковой панелью меню (рис. 3.220).

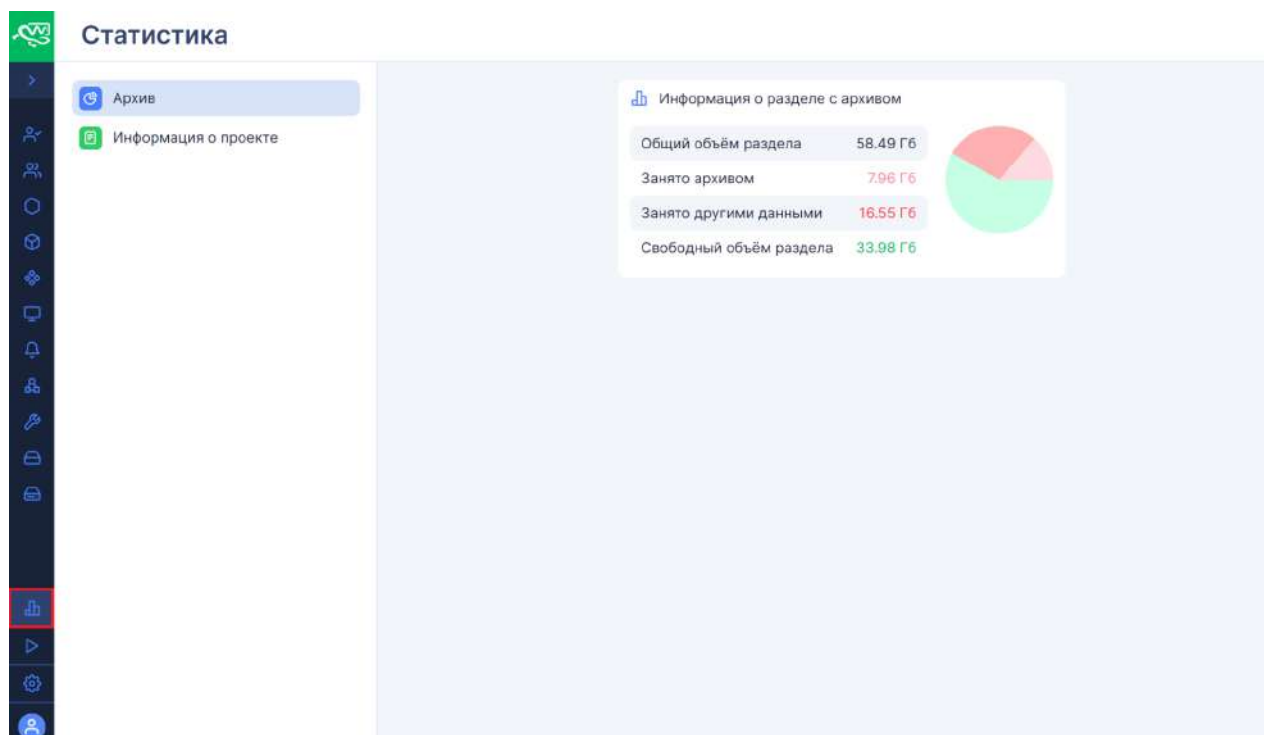


Рис. 3.220. Статистика

Данная страница включает в себя два раздела:

- 1) Архив, в котором находится информация о разделе - общий объем памяти, занятая память, свободная память;
- 2) Информация о проекте - количество объектов, моделей, событий, пользователей, мнемосхем, драйверов, задействованных в Системе.

Чтобы отобразить данные в разделе «Информация о проекте», необходимо нажать на соответствующую кнопку «>» напротив нужного параметра (рис. 3.221).

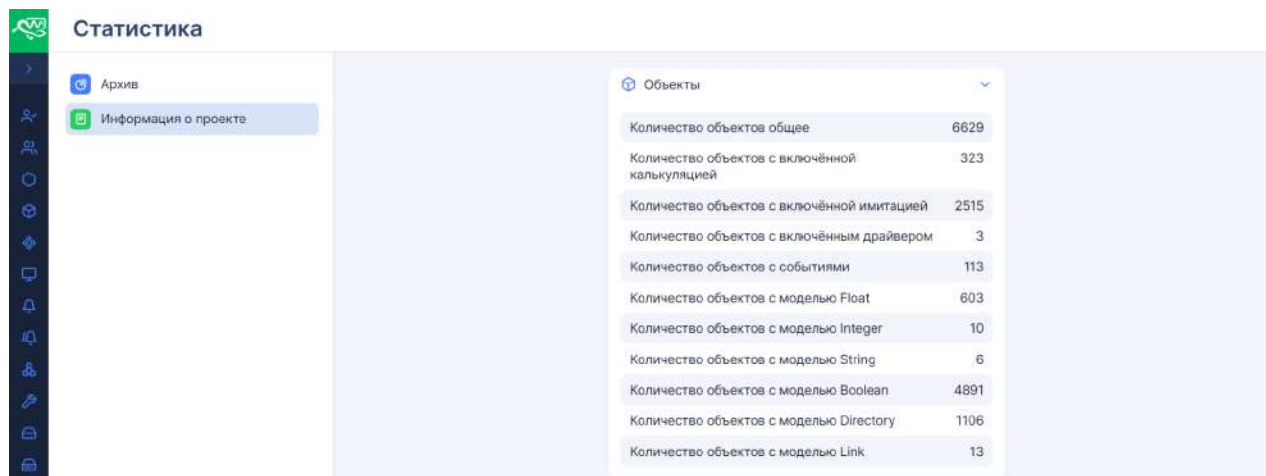


Рис. 3.221. Отображение данных

3.13.1. Настройка архива

Архиву можно задать определенные параметры через подраздел «Архив» раздела бокового меню «Настройки» (рис. 3.222).

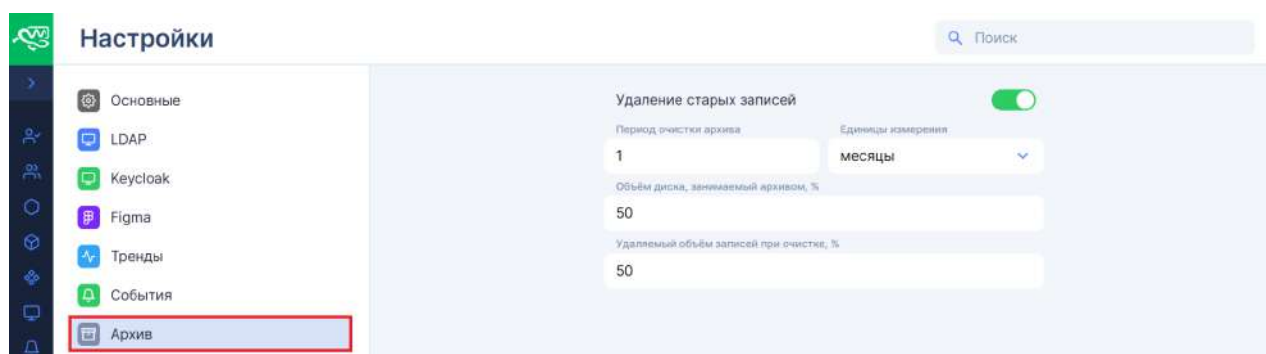


Рис. 3.222. Настройки архива

В настройках Архива пользователь может задать:

1. Период очистки архива (по секундам, минутам, часам, дням и тд.);
2. Объем диска занимаемый архивом;
3. Удаляемый объем записей при очистке.

А также воспользоваться функцией «Удаление старых записей» для удаления данных из архива по тем настройкам, которые заданы.

3.14. Резервное копирование

В Платформе присутствует возможность резервного копирования. Это подразумевает резервное копирование всей метаданных (модели, объекты, компоненты, мнемосхемы, классы событий, пользователи и т.д).

Пользователь может создавать и удалять резервные копии (рис. 3.223). При создании копии также можно указать комментарий (без возможности его редактирования).

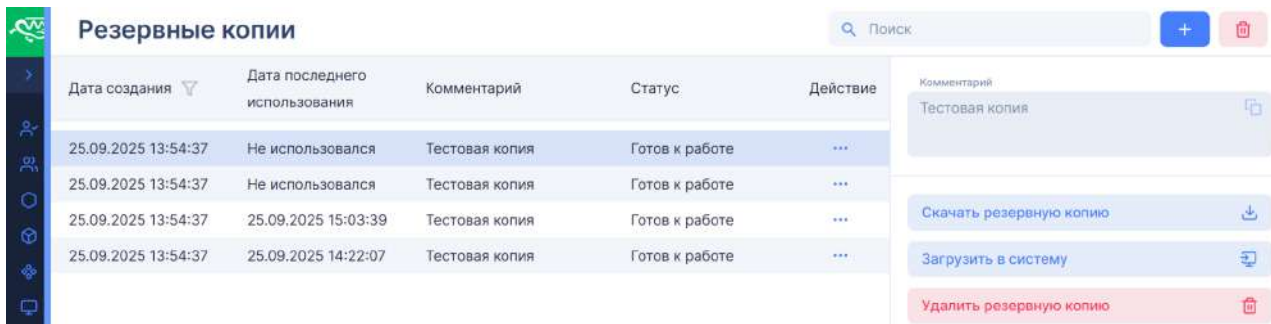


Рис. 3.223. Действия с резервными копиями

Резервная копия будет отображать информацию о дате создания, дате последнего использования, комментарий и статус, относящийся к его загрузке, созданию и удалению (рис. 3.224).

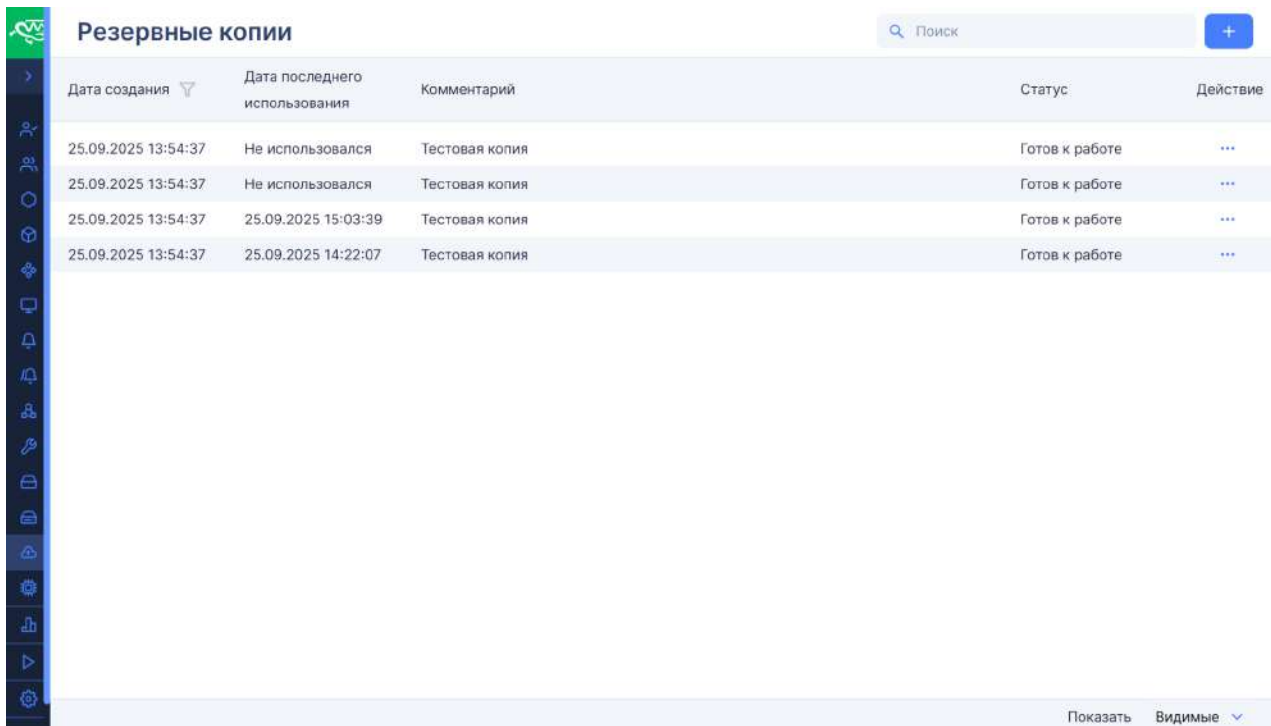


Рис. 3.224. Отображаемая информация

Для удобства в навигации реализован поиск по содержимому комментария, фильтрация по дате и времени создания, количество отображаемых копий на страницу (рис. 3.225).

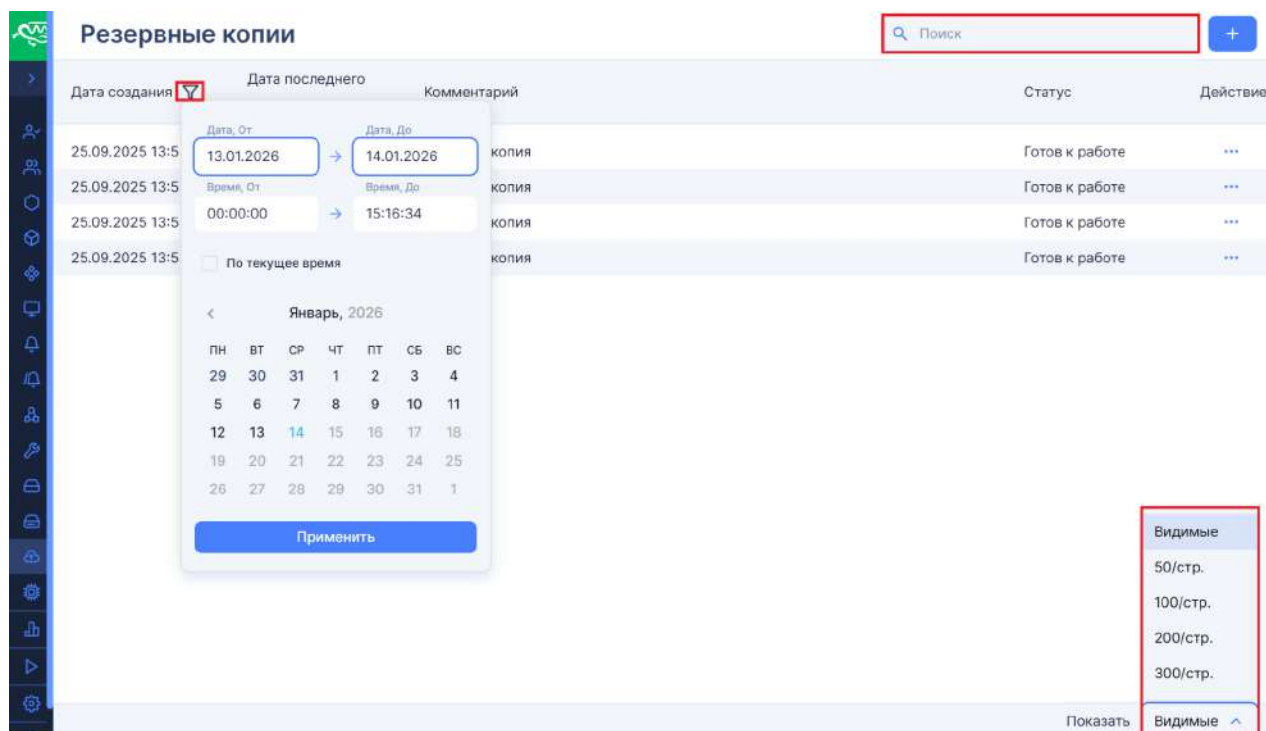


Рис. 3.225. Фильтрация списка бэкапов

Для того чтобы скачать резервную копию, загрузить ее в Систему или удалить, необходимо нажать на кнопку «...» в колонке «Действие» (рис. 3.226).

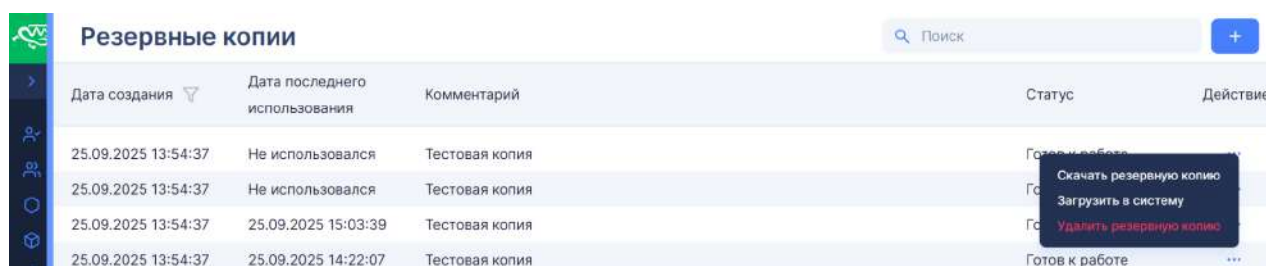


Рис. 3.226. Выбор действия

Либо нажать на саму копию, открыв боковую панель конфигурации (рис. 3.227).

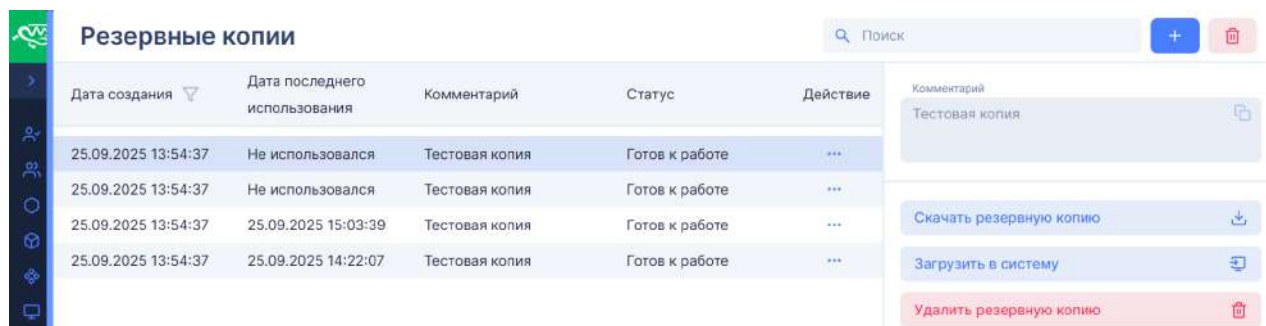


Рис. 3.227. Панель параметров

Для того чтобы создать или загрузить резервную копию из файла, необходимо нажать на соответствующую кнопку «+» в крайнем правом углу панели (рис. 3.228).

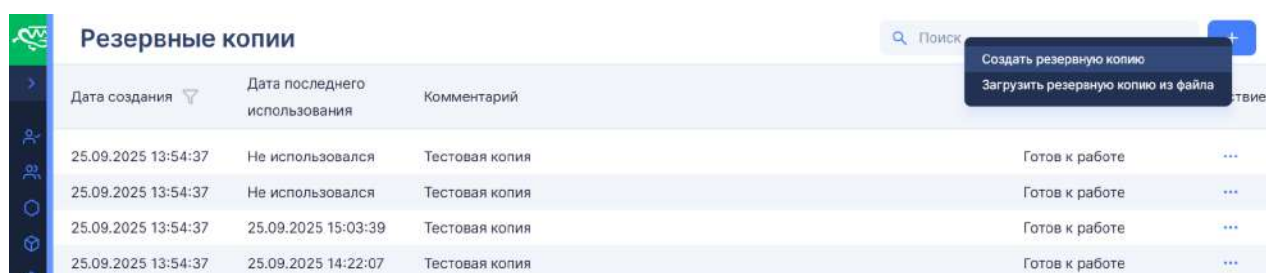


Рис. 3. 228. Создание/загрузка резервной копии из файла

Резервное копирование также доступно по заданному расписанию (периоду). Для этого следует перейти в соответствующий подраздел в разделе «Настройки» бокового меню. После чего нажать «+ Добавить новый период» и задать его вручную, либо через модальное окно (нажав на пиктограмму в окне ввода). При добавлении периода можно также добавить комментарий. По завершении настройки нажать «Сохранить».

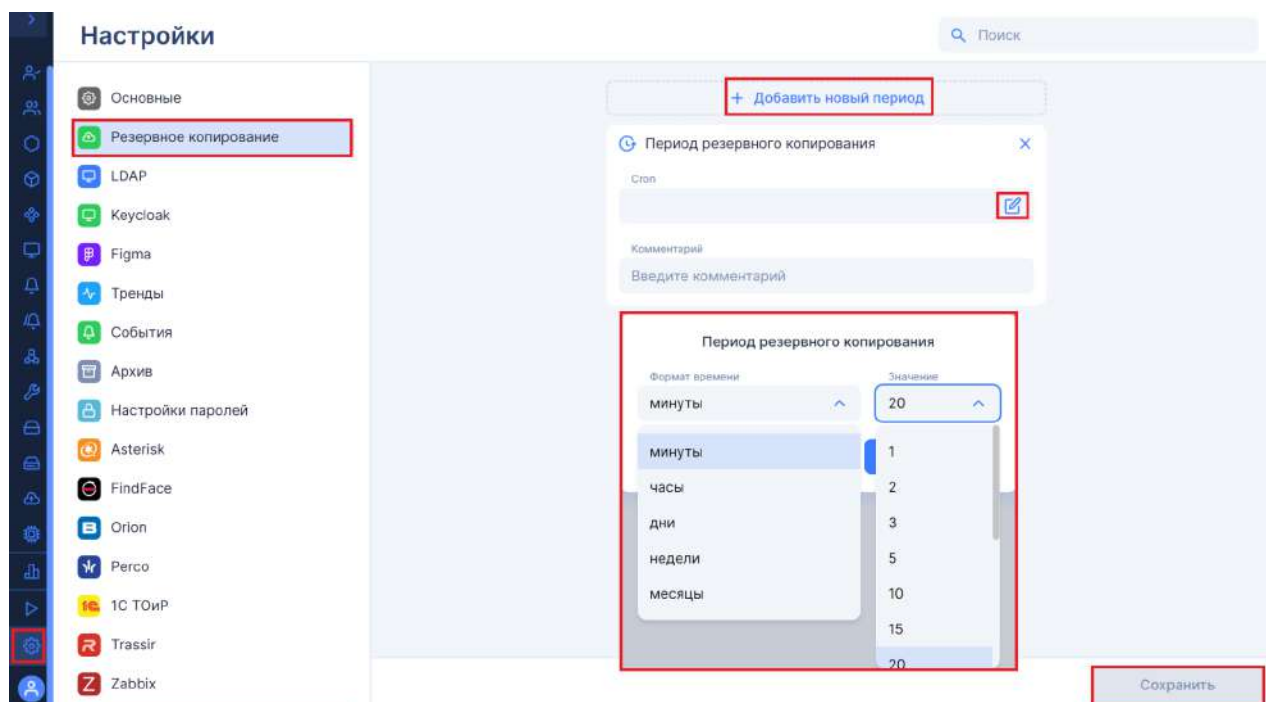


Рис. 3.229. Периоды резервного копирования

После сохранения резервные копии будут отображаться в соответствующем разделе бокового меню (рис. 3.330).

Дата создания	Дата последнего использования	Комментарий	Статус	Действие
25.02.2026 21:41:46	Не использовался	test schedule	Готов к работе	...
25.02.2026 21:21:41	Не использовался	test schedule	Готов к работе	...
25.02.2026 21:01:29	Не использовался	test schedule	Готов к работе	...
25.02.2026 20:41:24	Не использовался	test schedule	Готов к работе	...
25.02.2026 20:21:32	Не использовался	test schedule	Готов к работе	...
25.02.2026 20:01:51	Не использовался	test schedule edit	Готов к работе	...
25.02.2026 19:41:37	Не использовался	test schedule	Готов к работе	...

Рис. 3.330. Резервные копии

4. Среда исполнения

Данный раздел описывает работу среды исполнения, которая является не только рабочим местом оператора, предоставляя визуальную информацию, текущие и архивные данные о состоянии управляемой технологической системы в реальном времени, но и предоставляет возможности "тонкой" настройки Платформы. При этом нет необходимости в прерывании технологических процессов, поскольку изменения в конфигурации системы могут производиться что называется, «на лету».

В режиме исполнения пользователь имеет возможность анализа управляемой системы на основе таблиц и/или графиков, отражающих как текущие значения показателей, так и их изменения во времени, что позволяет отслеживать тенденцию изменения состояния наиболее критических элементов системы.

Механизм авторизованного распределения доступа, реализованный в Платформе, позволяет предотвратить несанкционированное вмешательство в работу системы со стороны пользователей, не обладающих соответствующей подготовкой и квалификацией, что минимизирует влияние т.н. "человеческого фактора" на работу системы. Тем не менее, это не означает полную замену человека машиной, а лишь выдвигает определенные требования к подготовке персонала.

В процессе пусконаладки и дальнейшей эксплуатации Платформы применительно к конкретной технологической системе, могут возникать дополнительные требования к функциональности элементов Платформы. В ряде случаев эти требования могут быть реализованы силами пользователей, без непосредственного привлечения службы технической поддержки Разработчика. Тем не менее, такие изменения в конфигурацию и/или логику работы Платформы должны тщательным образом документироваться с тем, чтобы в случае некорректных изменений была возможность "откатить" назад.

Ниже будет последовательно описан процесс работы в среде исполнения Платформы и доступные в этой среде функциональные возможности.

4.1. Авторизация

Для того чтобы авторизоваться в системе и начать работу с Платформой, необходимо ввести свои учетные данные (логин и пароль) в соответствующие поля и нажать на кнопку «Войти» (рис. 4.1).

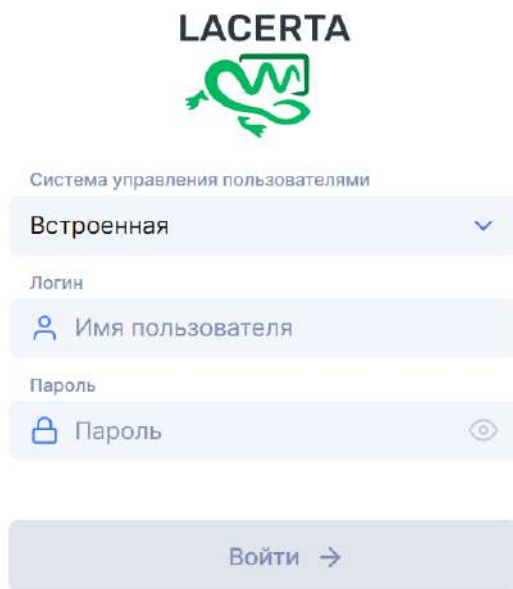


Рис. 4.1. Экран входа в систему

При вводе символов пароля у пользователя есть возможность сделать их видимыми, для этого нужно в форме ввода пароля нажать на значок в виде глаза (рис. 4.2)

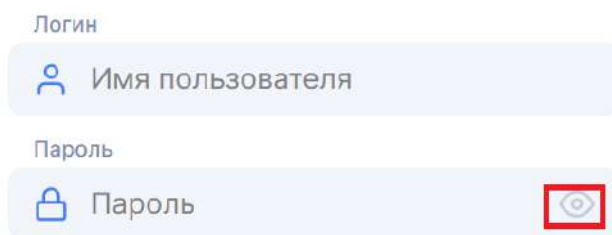
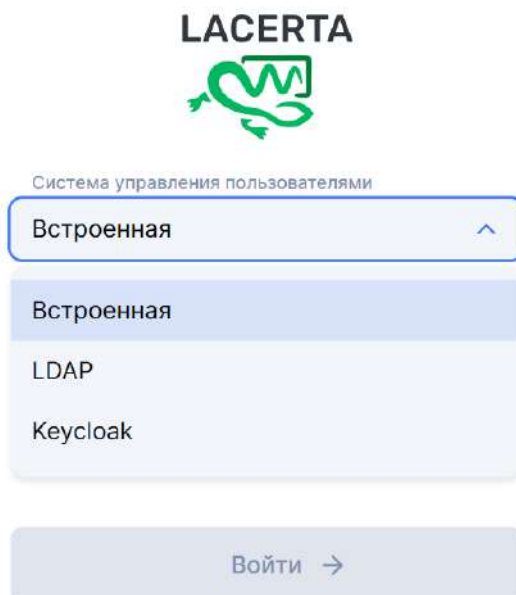


Рис. 4.2. Видимость символов пароля

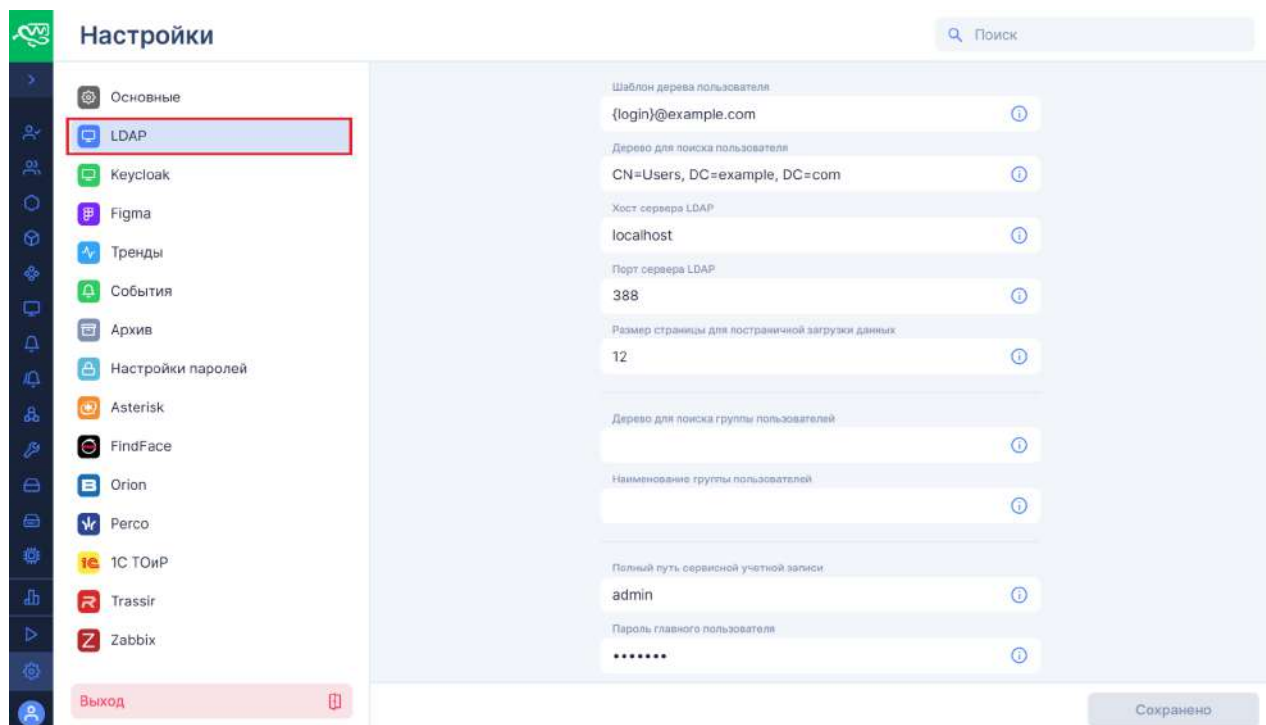
Также при авторизации присутствует возможность выбрать систему управления пользователями (рис. 4.3).



The screenshot shows the LACERTA logo at the top. Below it, the text "Система управления пользователями" is displayed. A dropdown menu is open, showing three options: "Встроенная" (selected), "LDAP", and "Keycloak". Below the dropdown is a button labeled "Войти" with a right-pointing arrow.

Рис. 4.3. Система управления пользователями

Данные системы имеют свой набор параметров и настроек интеграции. Для того чтобы перейти к их настройке, необходимо открыть раздел «Настройки» в боковом меню и выбрать соответствующую систему (рис. 4.4-4.5).



The screenshot shows the "Настройки" (Settings) page in the LACERTA application. The left sidebar contains a list of settings categories: "Основные", "LDAP" (highlighted with a red box), "Keycloak", "Figma", "Тренды", "События", "Архив", "Настройки паролей", "Asterisk", "FindFace", "Orion", "Perco", "1C TOIP", "Trassir", and "Zabbix". The main content area displays the LDAP configuration settings, including fields for "Шаблон дерева пользователя" (set to {login}@example.com), "Дерево для поиска пользователя" (set to CN=Users, DC=example, DC=com), "Хост сервера LDAP" (set to localhost), "Порт сервера LDAP" (set to 388), "Размер страницы для постраничной загрузки данных" (set to 12), "Дерево для поиска группы пользователей", "Наименование группы пользователей", "Полный путь сервисной учетной записи" (set to admin), and "Пароль главного пользователя" (masked with dots). A "Сохранено" (Saved) button is visible at the bottom right.

Рис. 4.4. Настройка интеграции с системой LDAP

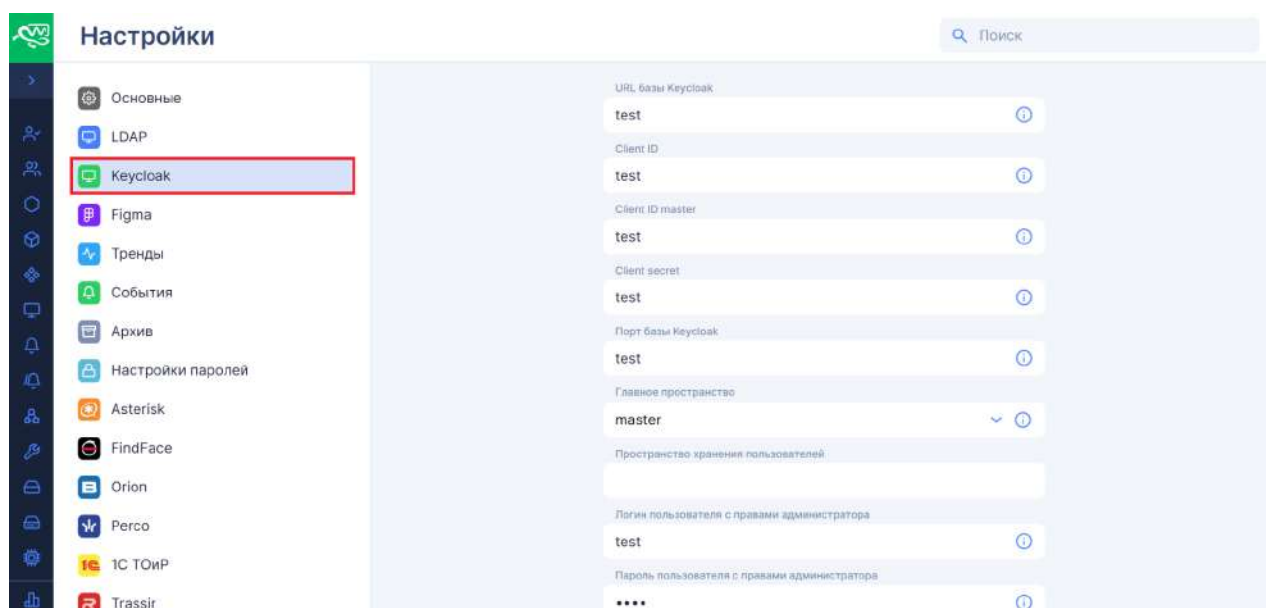


Рис. 4.5. Настройка интеграции с системой Keycloak

Некоторые параметры могут иметь примеры и описание. Для того чтобы отобразить данную подсказку необходимо навести курсор мыши на пиктограмму в углу поля заполнения (рис. 4.6).

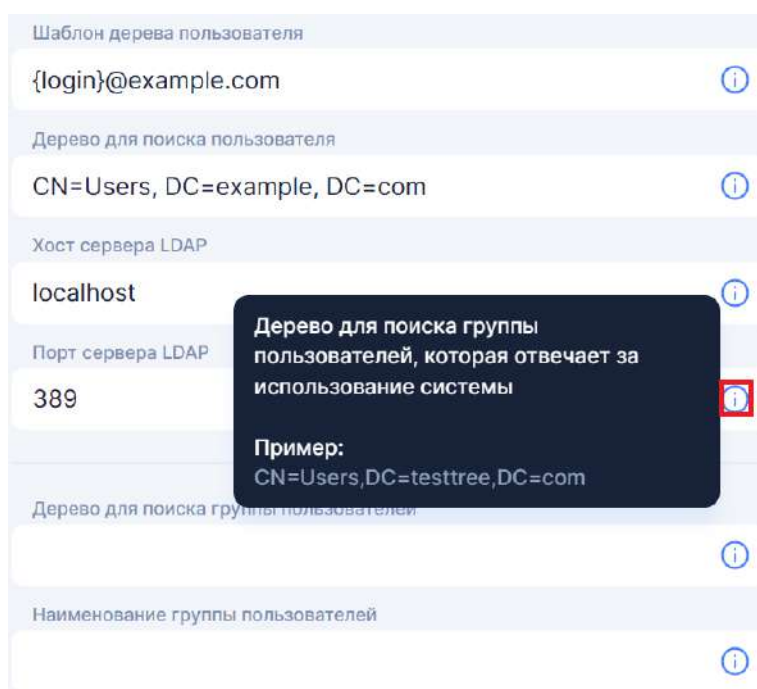


Рис. 4.6. Отображение подсказки

4.2. Стартовый экран и навигация

После успешной авторизации пользователя в Системе (разд. 4.1) открывается стартовый экран с приветствием. Через него можно создать мнемосхему либо импортировать ее из редактора Figma, воспользовавшись соответствующими кнопками (рис. 4.7).

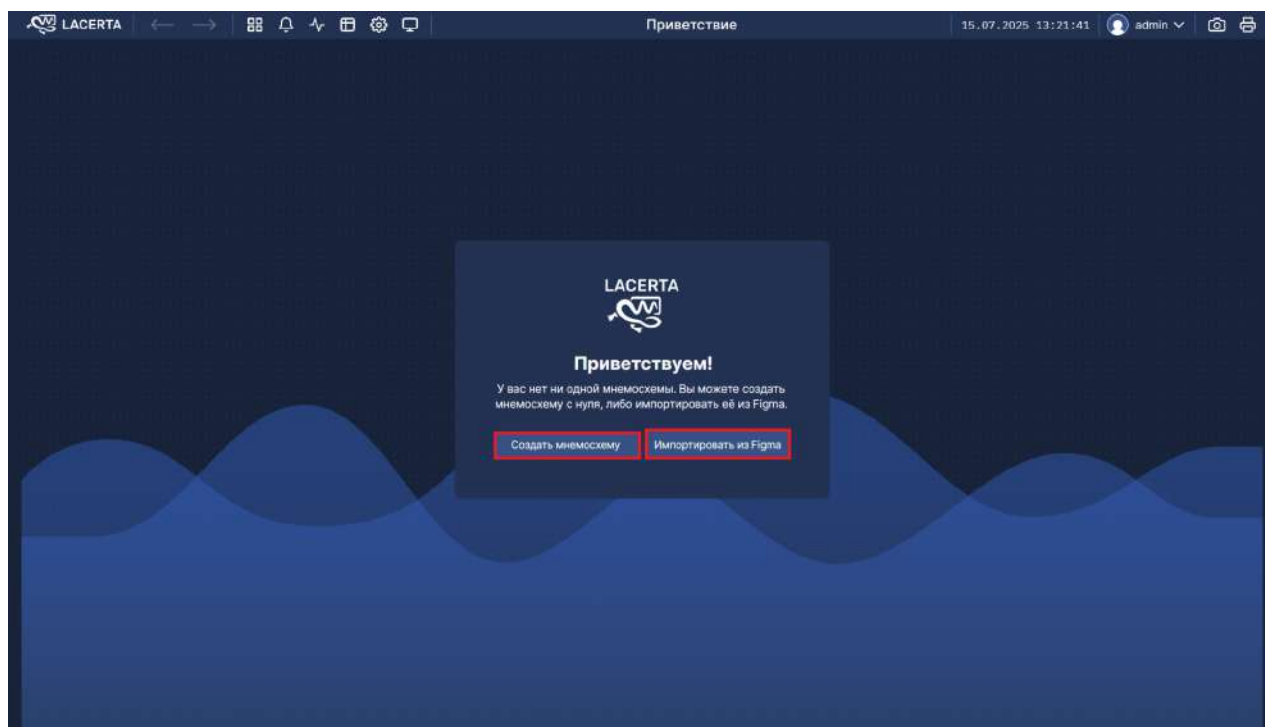


Рис. 4.7. Стартовый экран после авторизации

После создания мнемосхемы (или ее импорта) и запуска в режиме рантайм появляется навигация в виде кнопок и соответствующей информации в верхней части страницы.

Пример организации навигации по мнемосхемам (рис.4.8).

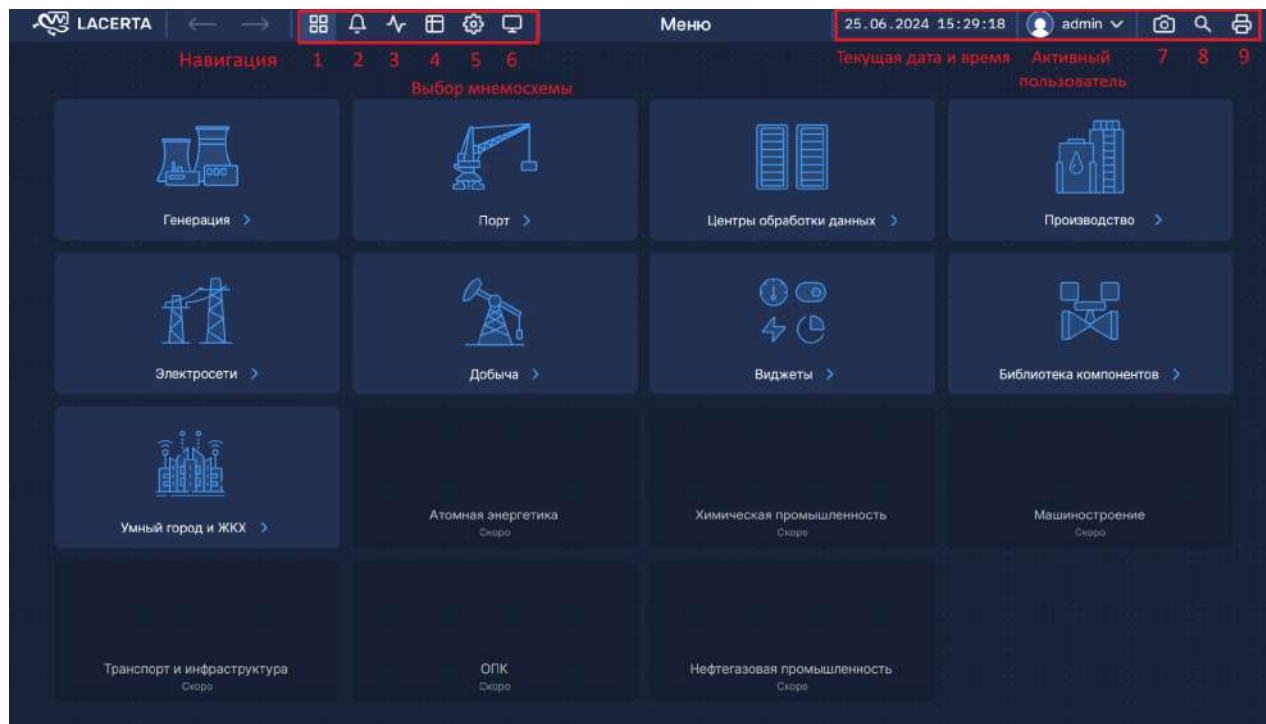


Рис. 4.8. Пример навигации в мнемосхемах

В верхней части экрана расположены кнопки главного меню. Цифрами на рис. 4.3 обозначены:

1. Переход (возврат) на стартовый экран (также можно использовать кнопку «Лацерта» в левой верхней области экрана);
2. Переход на вкладку «События»;
3. Переход на вкладку «Тренды»;
4. Переход на вкладку «Таблицы»;
5. Переход в меню «Настройки» (среда разработки и отладки);
6. Переход на вкладку «VNC»;
7. Сделать снимок экрана;
8. Распечатать образ экрана.

В верхней части экрана отображается информация о текущей дате и времени, а также о пользователе сеанса. В этом разделе главного меню можно завершить работу («Выход», либо перейти в раздел «Настройки» (среду разработки) воспользовавшись выпадающим списком (рис. 4.9)

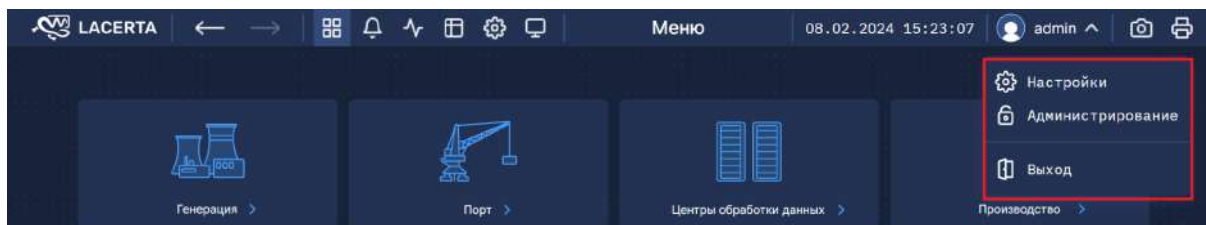


Рис. 4.9. Настройки пользователя

Через поиск по объектам на стартовом экране (обозначен цифрой 8 на рис. 4.8) у пользователя имеется возможность открыть мнемосхему в рантайме или перейти в ее редактор.

В случае если модель объекта в поисковой выдаче соответствует модели, добавленной в какую-либо мнемосхему (мнемосхемы) в качестве аргумента, то ссылка на такую мнемосхему (мнемосхемы) отобразится в поиске (рис. 4.10.).

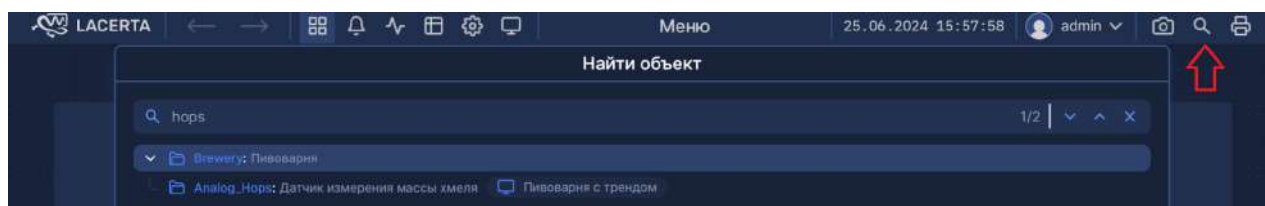


Рис. 4.10. Поиск по объектам

Кликом на заголовок мнемосхемы можно перейти в мнемосхему в рантайме (открыть) или в редактор мнемосхемы (изменить, рис. 4.11.).

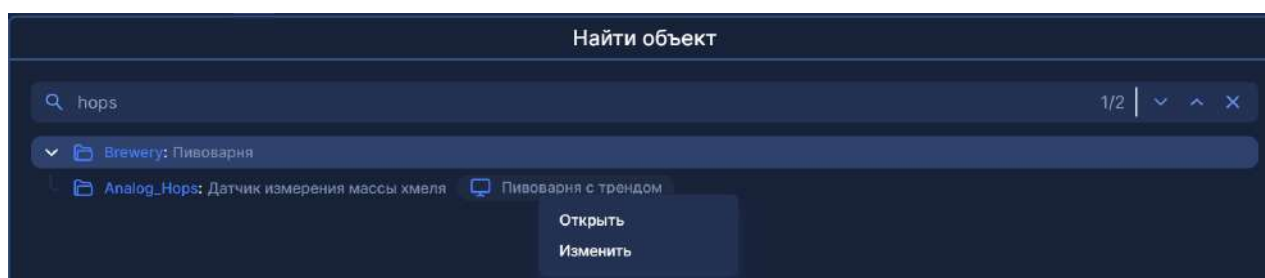


Рис. 4.11. Переход в мнемосхему

4.3. Мнемосхемы и виджеты



Рис. 4.11. Мнемосхема (турбина)

На рис. 4.11. приведён пример мнемосхемы турбины тепловой электростанции. Цифрами на мнемосхеме обозначены её основные элементы:

- 1 - Виджеты (в данном примере - тахометр, манометр, термометр, мощность генерации, напряжение на шине и выходе генератора);
- 2 - Статический элемент - показатель уровня масла ;
- 3 - Динамический элемент - регулятор числа оборотов турбины;
- 4 - фрагмент, содержащий показания датчика давления масла в трансмиссии (заданное и фактическое), а также органы управления.

В рамках Платформы с помощью мнемосхемы в реальном времени решаются следующие задачи:

- наглядное отображение технологического процесса (процессов) управляемой системы;
- обеспечение интерактивного взаимодействия между оператором и управляемой системой (человеко-машинный интерфейс);
- возможность непосредственного управления технологической системой и её отдельными элементами;
- отображение и локализация событий (тревог, критических и аварийных ситуаций).
- возможность обращения к элементам мнемосхемы (объектам) для получения оперативной информации и/или непосредственного управления объектом.

Отображение стартовой мнемосхемы в рантайме происходит по следующему алгоритму:

- Приоритетно настройка берется с Пользователя;
- Если на Пользователе не задана, берется с Роли;
- Если не задана на Роли, берется глобальная из раздела «Настройки».

4.3.1. Композитная мнемосхема

Композитная мнемосхема (составная мнемосхема) состоит из нескольких блоков. На данный момент, функционируют следующие виды блоков (рис. 4.12) - mнemo (мнемосхема), iframe, script (скрипт).

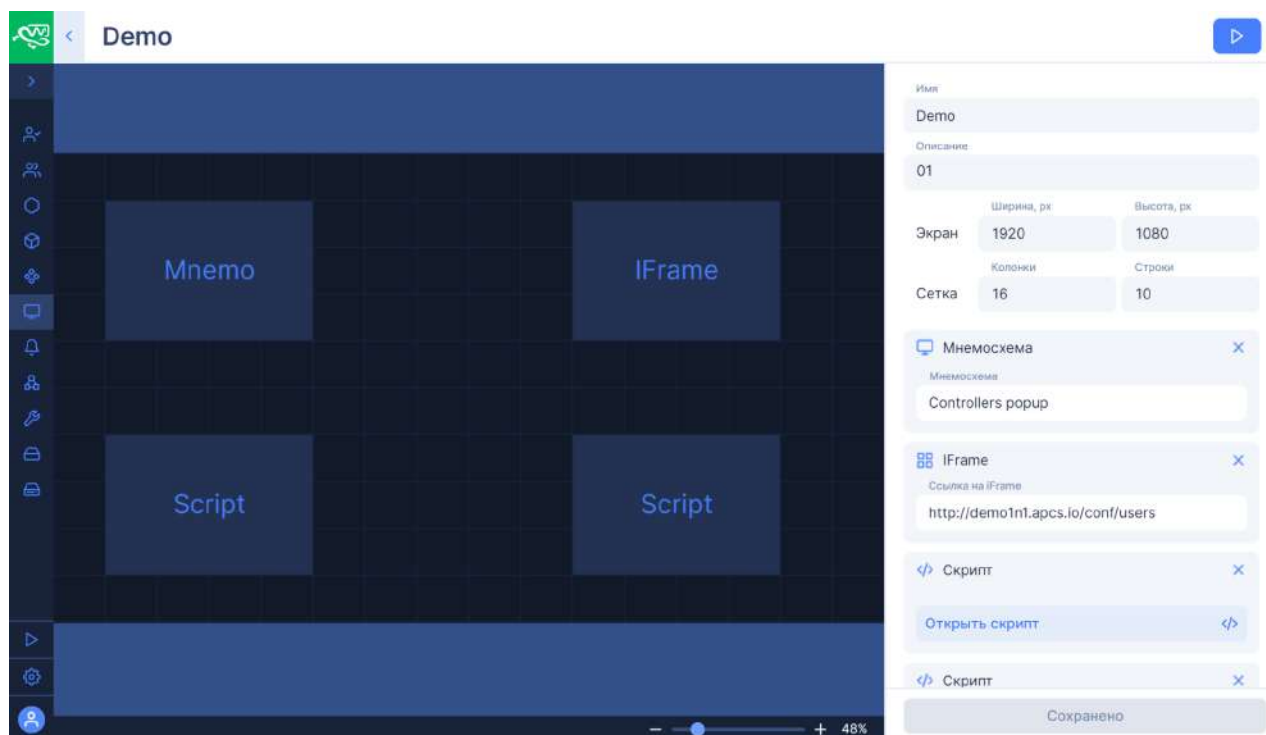


Рис. 4.12. Композитная мнемосхема

Для создания блока необходимо зажать курсор и растянуть блок на необходимый размер (рис. 4.13).

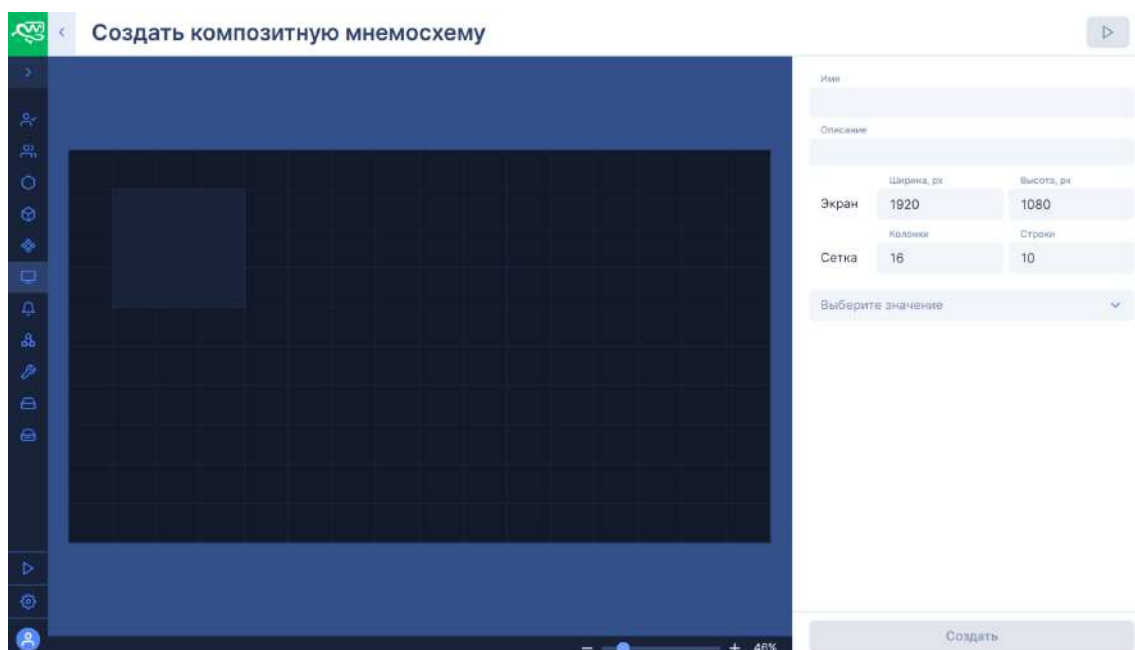


Рис. 4.13. Создание блока

Далее выбрать нужный вид блока из представленных и задать его параметры (рис. 4.14).

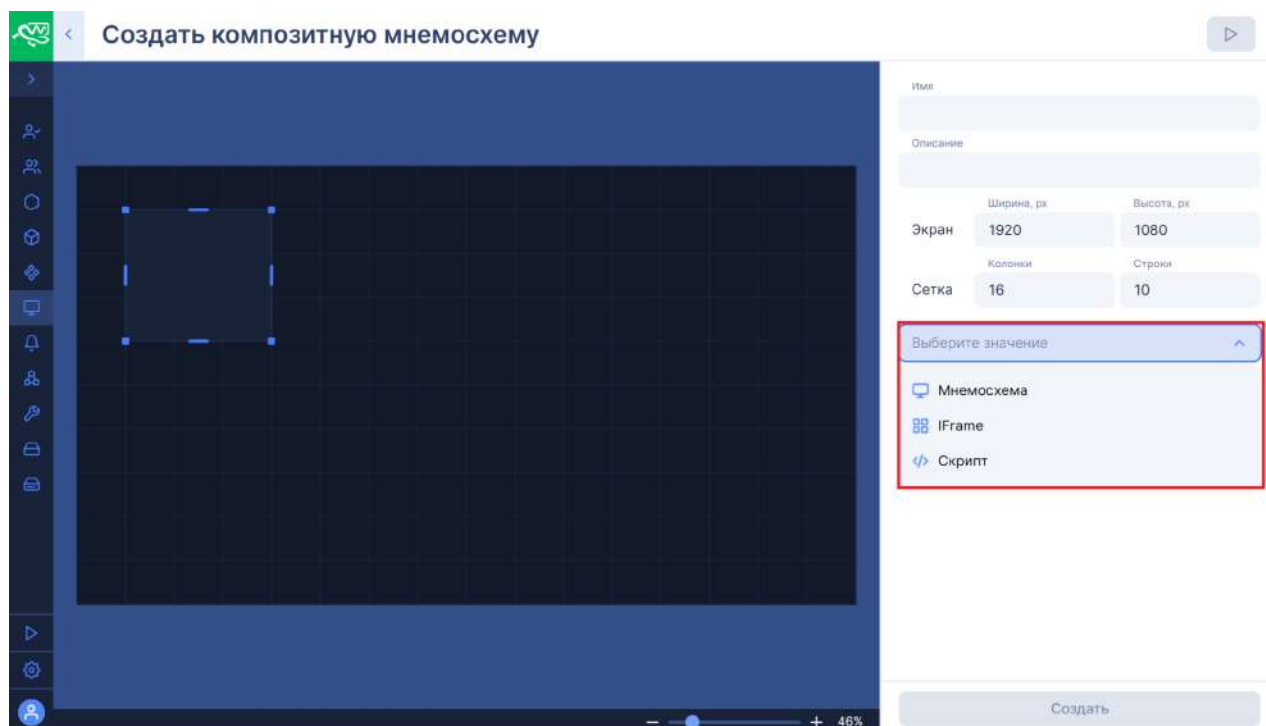


Рис. 4.14. Выбор значения

По завершении нажать сохранить для создания композитной мнемосхемы (рис. 4.15).

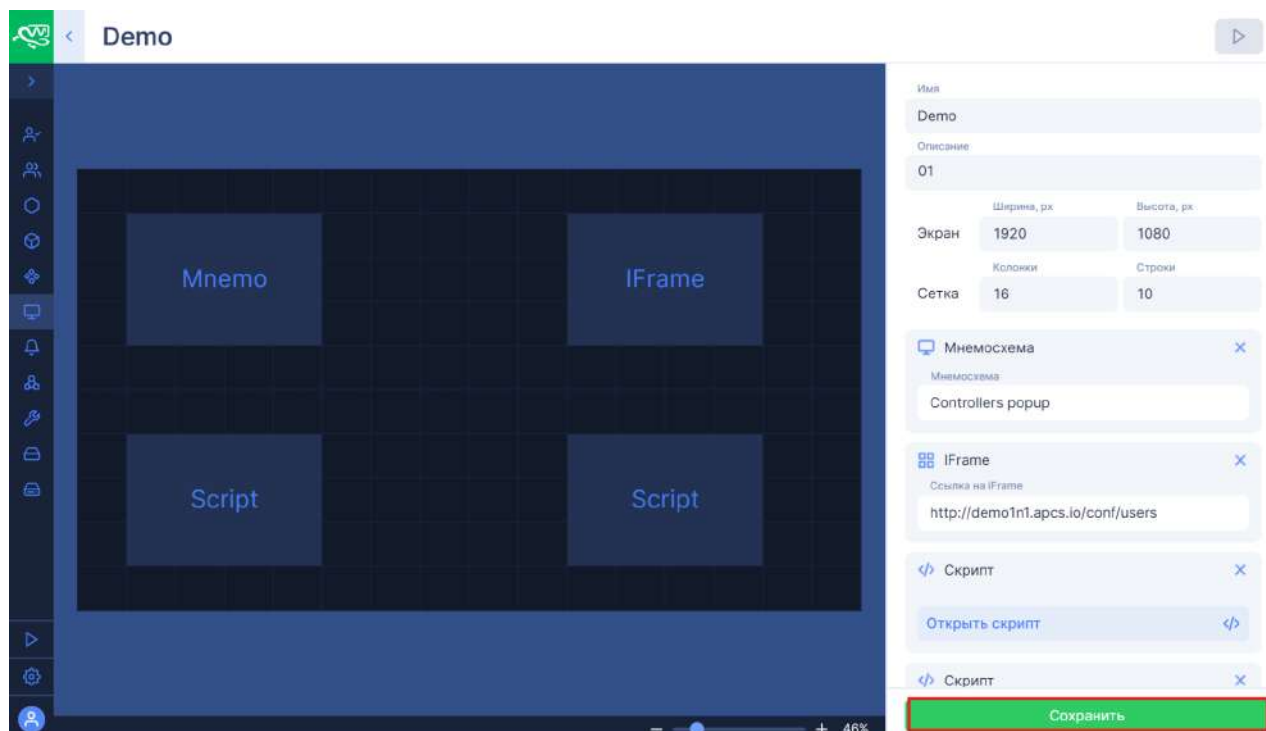


Рис. 4.15. Создание композитной мнемосхемы

Перемещение блоков осуществляется путем зажатия курсора на блоке и его перетягивания в нужное место. Для того чтобы удалить блок нужно кликнуть на него и после этого нажать на клавишу «Backspace».

Настройки также позволяют задать пользователем параметры экрана и сетки (рис. 4.16).

The screenshot displays a configuration window for a block named 'iFrame&Mnemo'. The window has a light blue header with the block name. Below the header, there are two sections: 'Имя' (Name) and 'Описание' (Description), both containing the text 'iFrame&Mnemo'. The main configuration area is divided into two rows. The first row, labeled 'Экран' (Screen), contains two input fields: 'Ширина, px' (Width, px) with the value '1920' and 'Высота, px' (Height, px) with the value '1080'. The second row, labeled 'Сетка' (Grid), contains two input fields: 'Колонки' (Columns) with the value '16' and 'Строки' (Rows) with the value '9'. These four input fields are enclosed in a red rectangular box. Below the configuration area, there are three expandable sections. The first section, titled 'Мнемо' (Mnemonic), has a sub-label 'Ссылка на мнемосхему' (Link to mnemonic) and a text input field containing 'IoT 1'. The second section, titled 'iFrame', has a sub-label 'Ссылка на iFrame' (Link to iFrame) and a text input field containing '1'. The third section is titled 'iFrame' and is currently collapsed. At the bottom of the window is a green button labeled 'Сохранить' (Save).

	Ширина, px	Высота, px
Экран	1920	1080
	Колонки	Строки
Сетка	16	9

Рис. 4.16. Параметры блока

Для старта композитной мнемосхемы в рантайме необходимо нажать на кнопку «Запуск» (рис. 4.17).

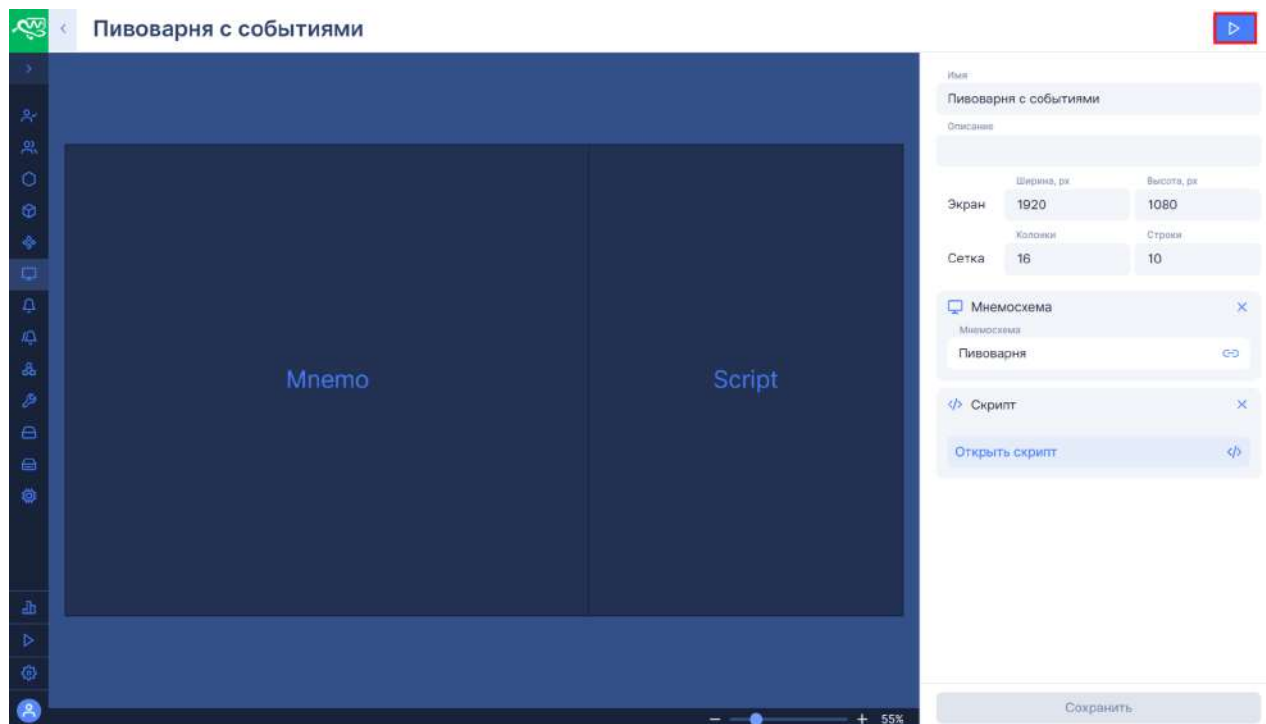


Рис. 4.17. Кнопка запуска мнемосхемы

После запуска композитная мнемосхема отобразится в подобном виде (рис. 4.18).

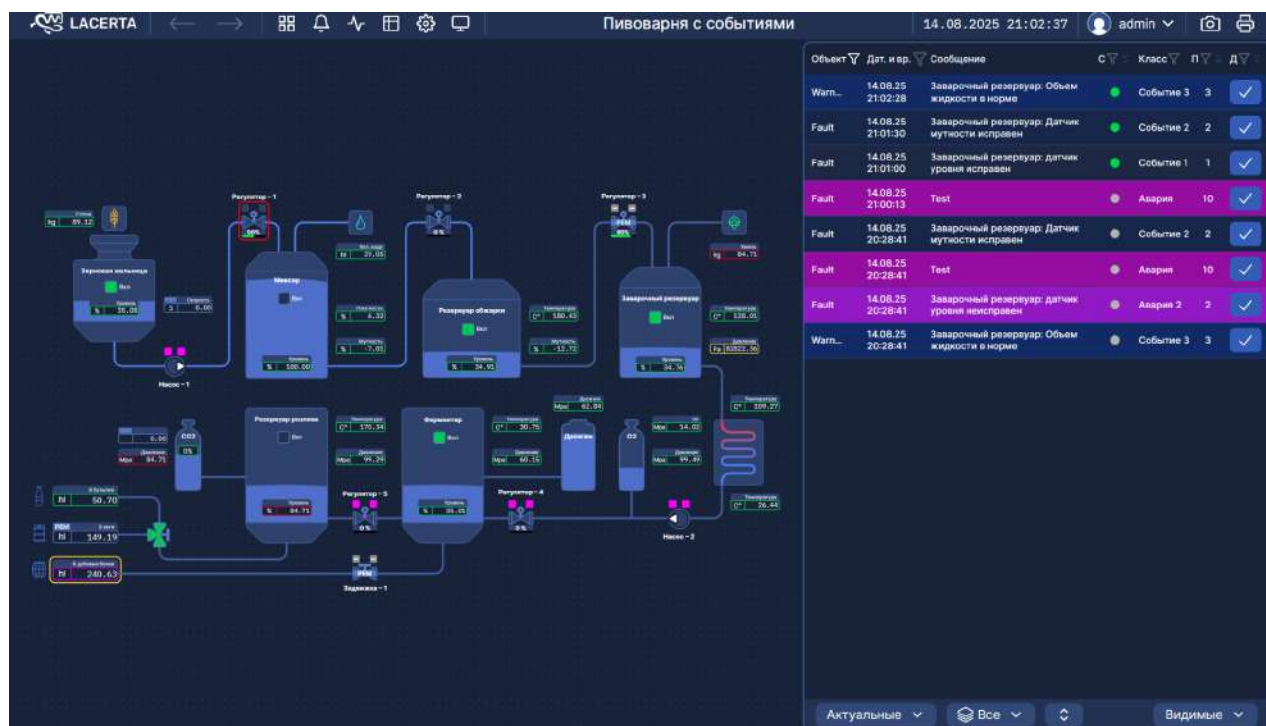


Рис. 4.18. Композитная мнемосхема в режиме рантайма

На примере (рис. 4.18) представлена композитная мнемосхема с виджетом событий. Для отображения данного виджета был использован блок «Script» и функция `showEvents(element, class_tags, object_ids, limit, sorting)` с параметрами `(element, null, [160, 162, 155, 9966, 10065, 165])`, где `null` - необходим, поскольку мы пропускаем параметр `class_tags`, `[160, 162, 155, 9966, 10065, 165]` - массив `id` объектов (рис. 4.19).

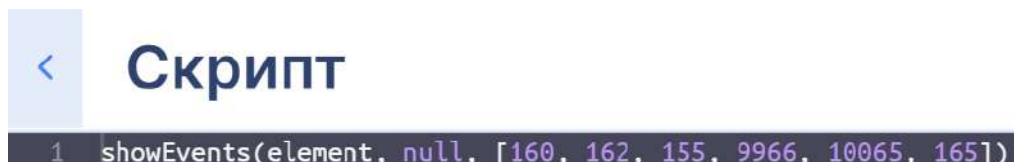


Рис. 4.19. Параметры функции в редакторе

Композитная мнемосхема может состоять из нескольких блоков «Script» с возможностью отображать сразу несколько виджетов (рис. 4.20).

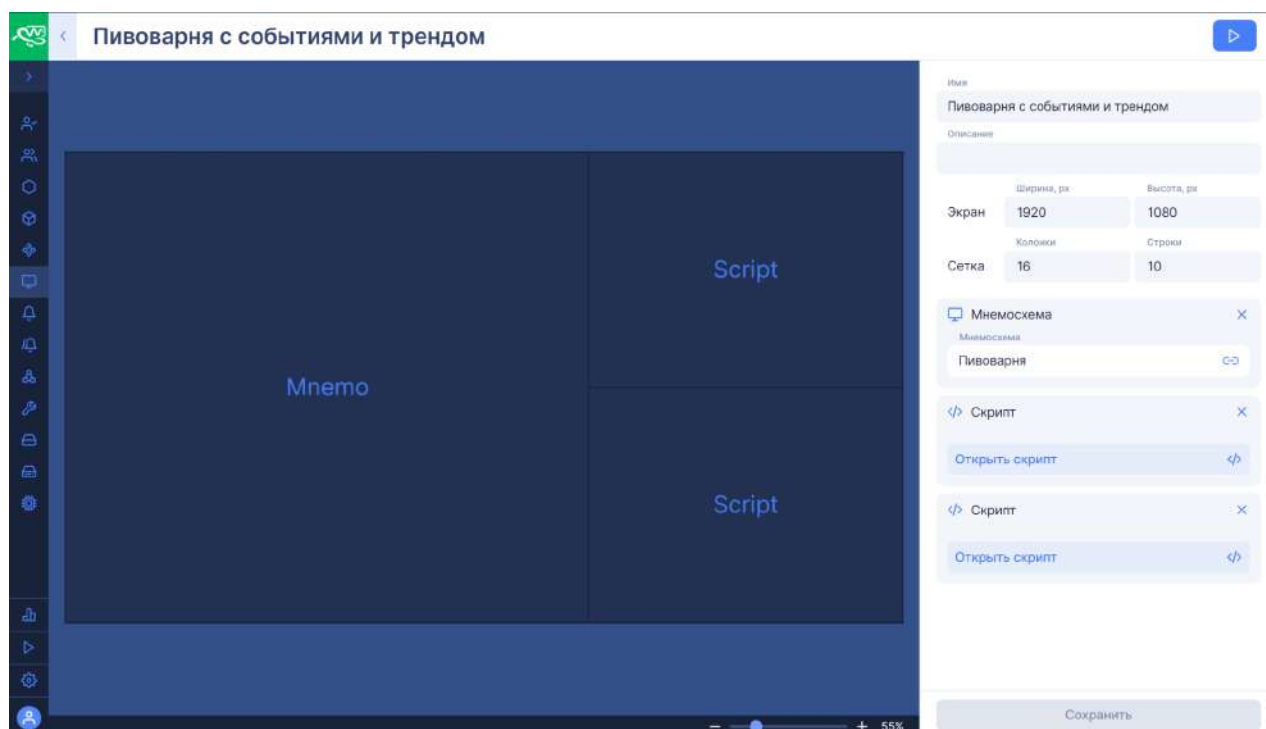


Рис. 4.20. Пример мнемосхемы с несколькими блоками «Script»

Пример запуска такой мнемосхемы режиме рантайм (рис. 4.21).

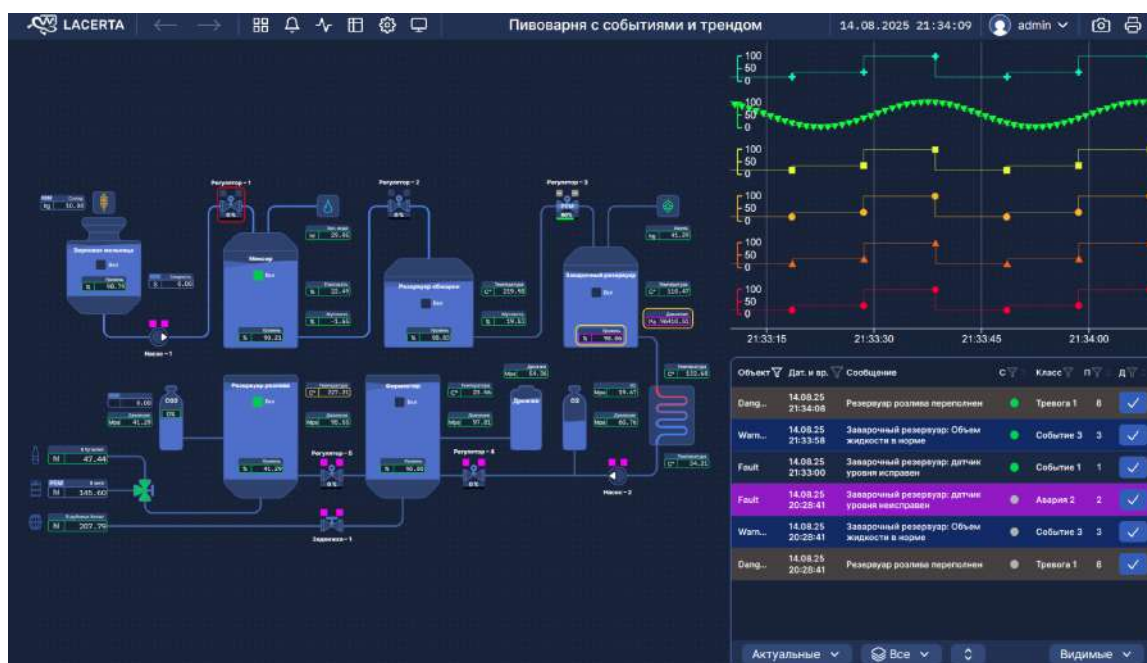


Рис. 4.21. Запуск мнемосхемы в режиме рантайм

На примере (рис. 4.21) представлена мнемосхема с виджетом событий и виджетом тренда. Для отображения виджета тренда был использован блок «Script» и функция `showTrend(element, objects, selectionId, type, date_start, date_end, interval)`, с параметрами `(element, null, 265, 1)`, где `null` - необходим, поскольку мы пропускаем параметр `objects`, `265` - id выборки, `1` - тип графика, которому соответствует "Ступенчатая" (рис. 4.22).

< Скрипт

```
1 showTrend(element, null, 265, 1)
```

Рис. 4.22. Параметры функция в редакторе

Для отображения виджета событий был использован блок «Script» и функция `showEvents(element, class_tags, object_ids, limit, sorting)` с параметрами `(element, null, [10071, 155, 9966])`, где `null` - необходим, поскольку мы пропускаем параметр `class_tags`, `[10071, 155, 9966]` - массив id объектов (рис. 4.23).

< Скрипт

```
1 showEvents(element, null, [10071, 155, 9966])
```

Рис. 4.23. Параметры функция в редакторе

Композитная мнемосхема может иметь вертикальное расположение блоков мнемосхем с ориентировкой под планшет (рис. 4.24).

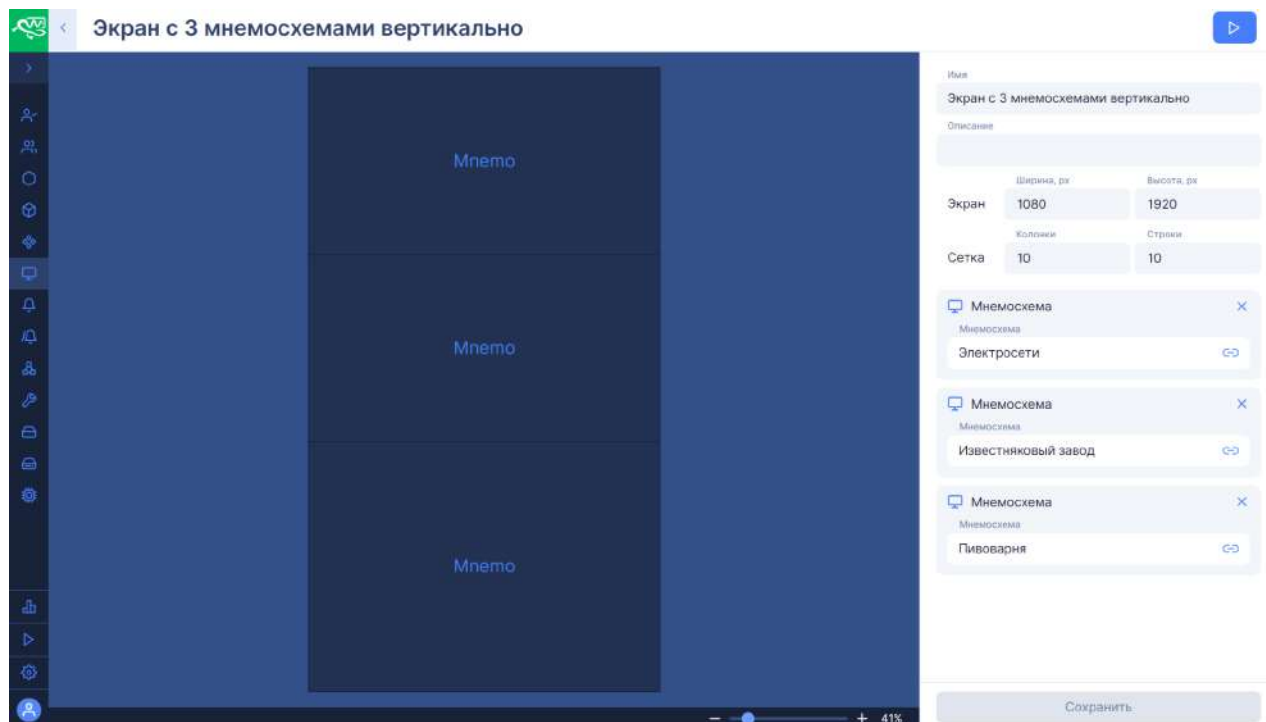


Рис. 4.24. Пример вертикального расположения мнемосхем

Пример вертикальной мнемосхемы в режиме рантайм (рис. 4.25).



Рис. 4.25. Пример мнемосхемы с вертикальным расположением блоков

В композитную мнемосхему можно расположить большое количество блоков. Для наглядности пример композитной мнемосхемы с горизонтальным расположением сразу 6 блоков мнемосхем (рис. 4.26).

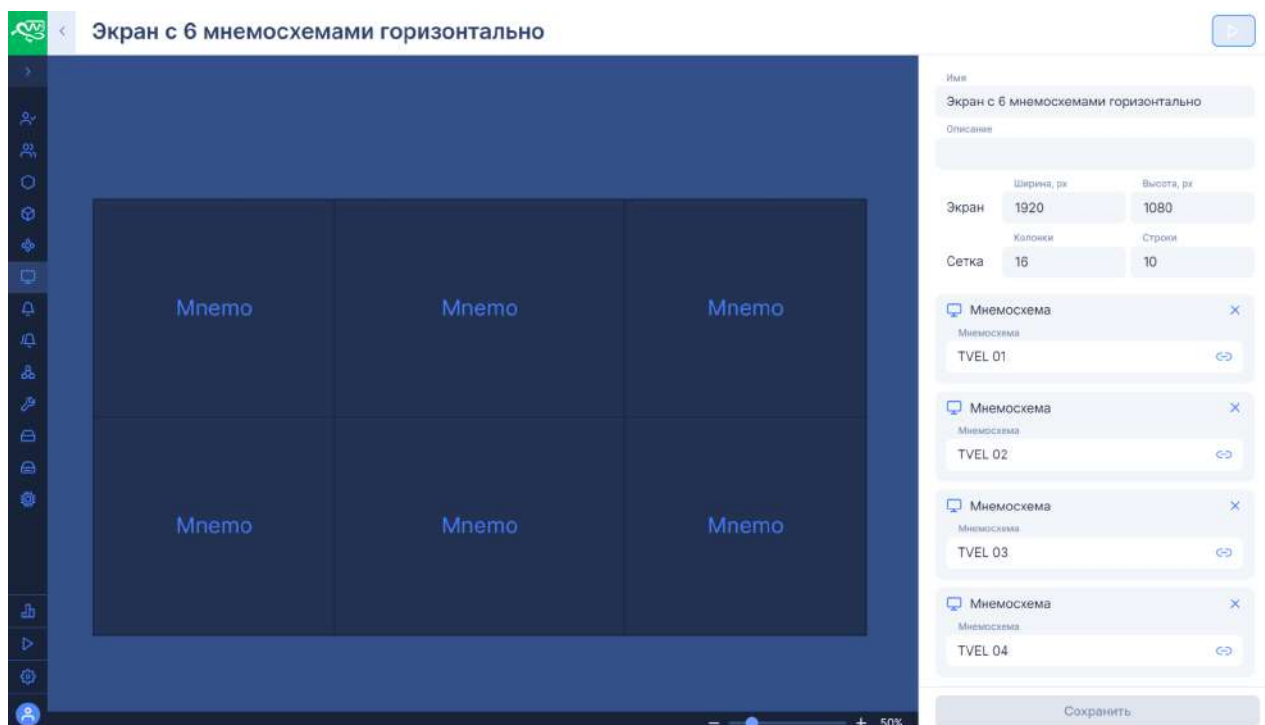


Рис. 4.26. Пример композитной мнемосхемы с 6 блоками

Пример запуска такой мнемосхемы в режиме рантайм (рис. 4.27).

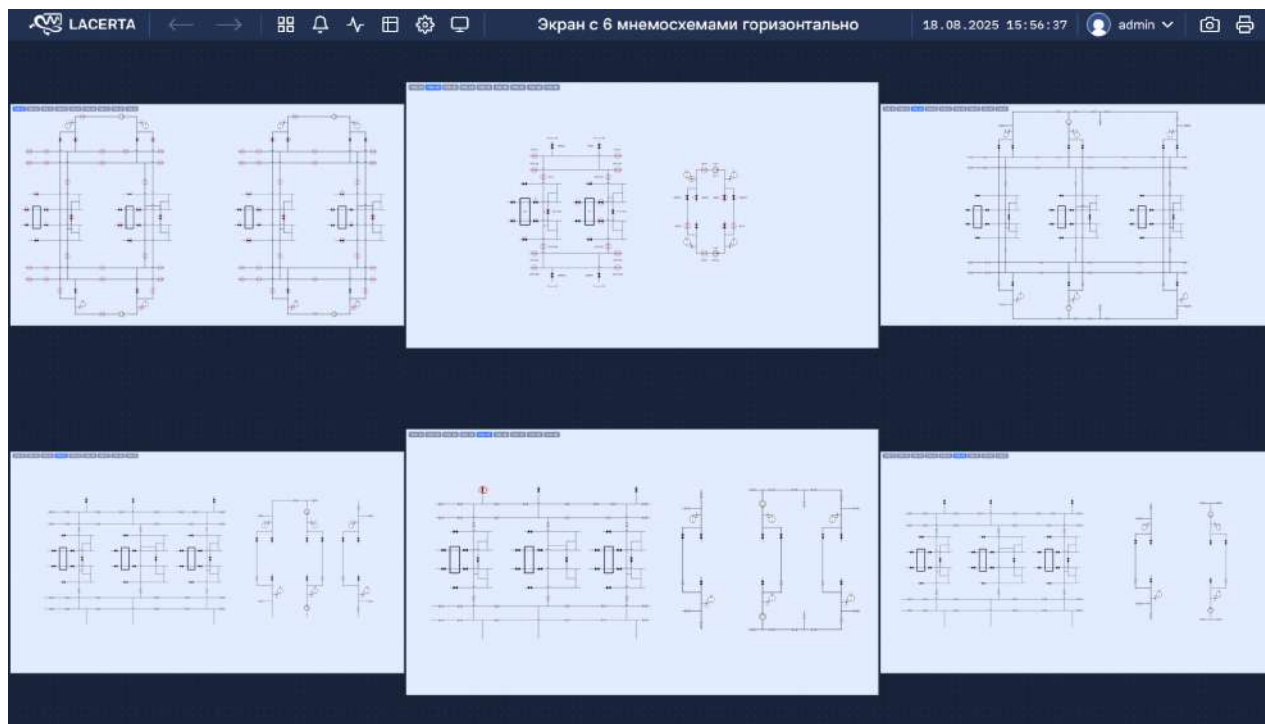


Рис. 4.27. Запуск мнемосхемы в режиме рантайм

4.3.2. Состояния компонентов

Мнемосхема может отображать одно, или же совокупность векторных изображений, а также состоять из нескольких заранее подготовленных компонентов, свойства которых (цвет, видимость, прозрачность и др.) могут изменяться согласно заданному сценарию пользователя, отображая тем самым состояние управляемой системы. Перечень векторных изображений и сценарии их поведения определяются на этапе проектирования и разработки Системы.

Пример вариантов отображения компонентов мнемосхемы приведен на рис. 4.19.



Рис. 4.19. Примеры компонентов в зависимости от состояния объекта

Таким образом, из компонентов возможна сборка панелей управления и/или отображение информации (dashboard) из готовых элементов (рис. 4.20).



Рис. 4.20. Пример мнемосхемы (dashboard)

4.3.3. Состояние тревоги

Возникающие состояния «Тревога» отображаются на мнемосхеме путём изменения компонентов (цвет, форма) (рис. 4.21).

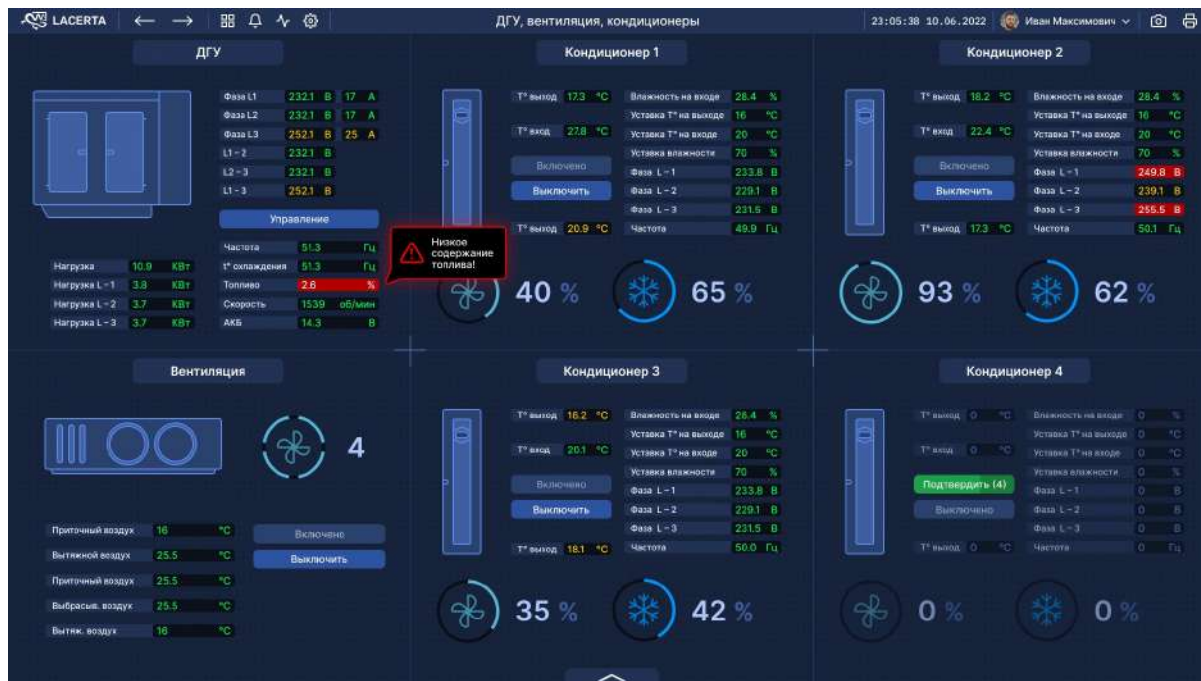


Рис. 4.21. Пример мнемосхемы

Цвет и форма компонентов при различных тревогах обозначаются индивидуально через привязку компонентов к моделям, где в скрипте описывается поведение компонента в зависимости от значений модели.

4.4. Таблицы

Данные Платформы могут отображаться в табличном виде. Для того чтобы перейти в раздел таблиц, нужно выбрать соответствующий пункт главного меню (Рис. 4.22).



Рис. 4.22. Переход в Таблицы

После перехода в раздел «Таблицы» появляется стартовый экран, нажатием на который открывается форма выбора сигналов объектов (рис. 4.23).

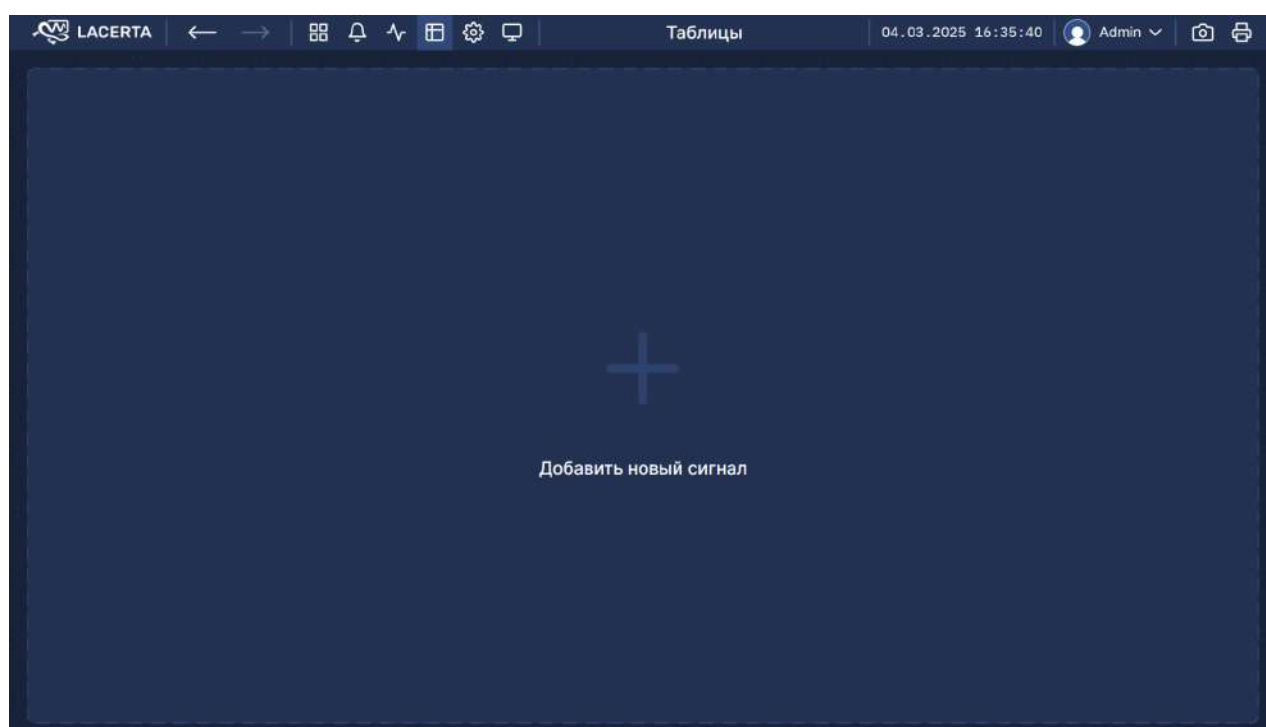


Рис. 4.23. Стартовый экран

У пользователя есть возможность выбрать сигналы самостоятельно либо воспользоваться уже готовыми выборками (рис. 4.24).

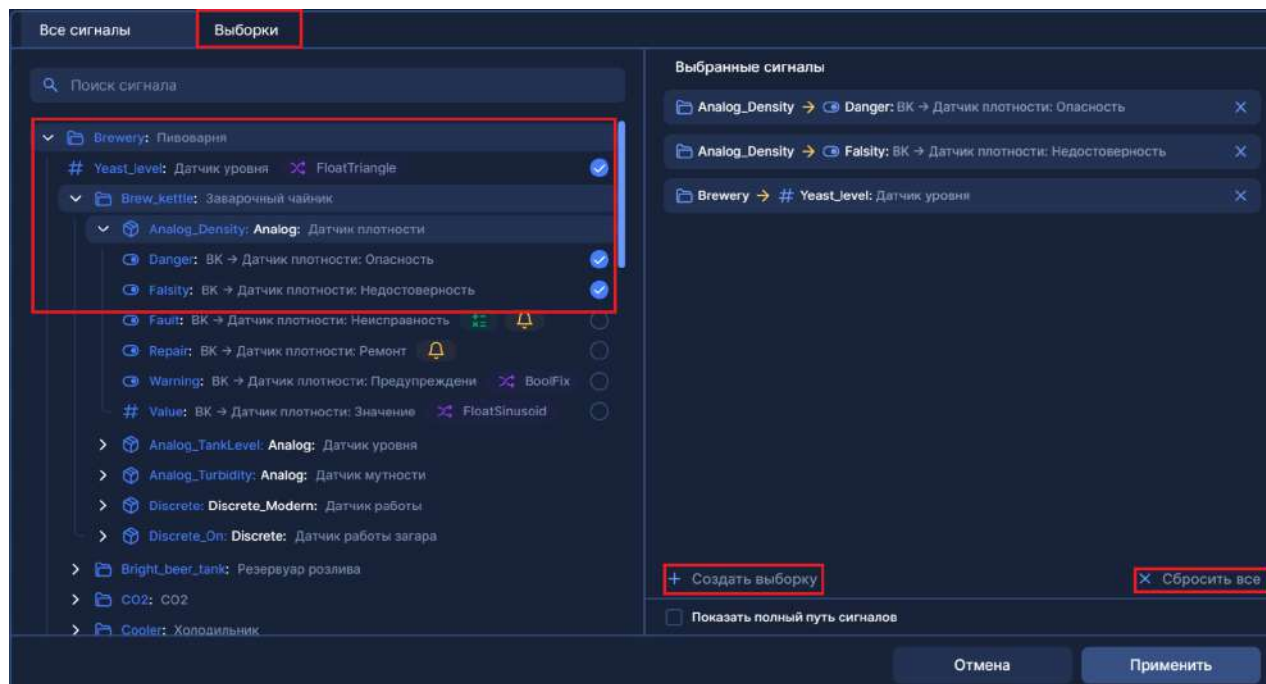


Рис. 4.24. Выбор сигналов объектов

Для выбора сигналов необходимо отметить их галочкой, при этом они автоматически отобразятся в списке выбранных сигналов (рис. 4.24).

Выбранные сигналы можно сохранить как выборку или сбросить их. Для этого необходимо воспользоваться соответствующими кнопками в форме выбора сигналов (рис. 4.24).

Отметив флажок «Показать полный путь сигналов» можно проверить корректность выбора (рис.4.25).

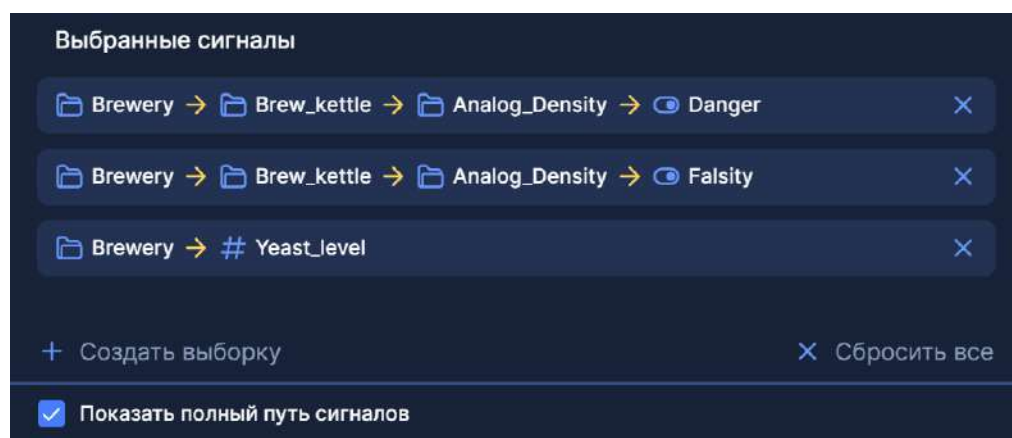


Рис. 4.25. Флажок показа полного пути сигнала

После нажатия кнопки «+ Создать выборку» нужно задать ее название в произвольной форме и нажать «Создать» (рис. 4.26).

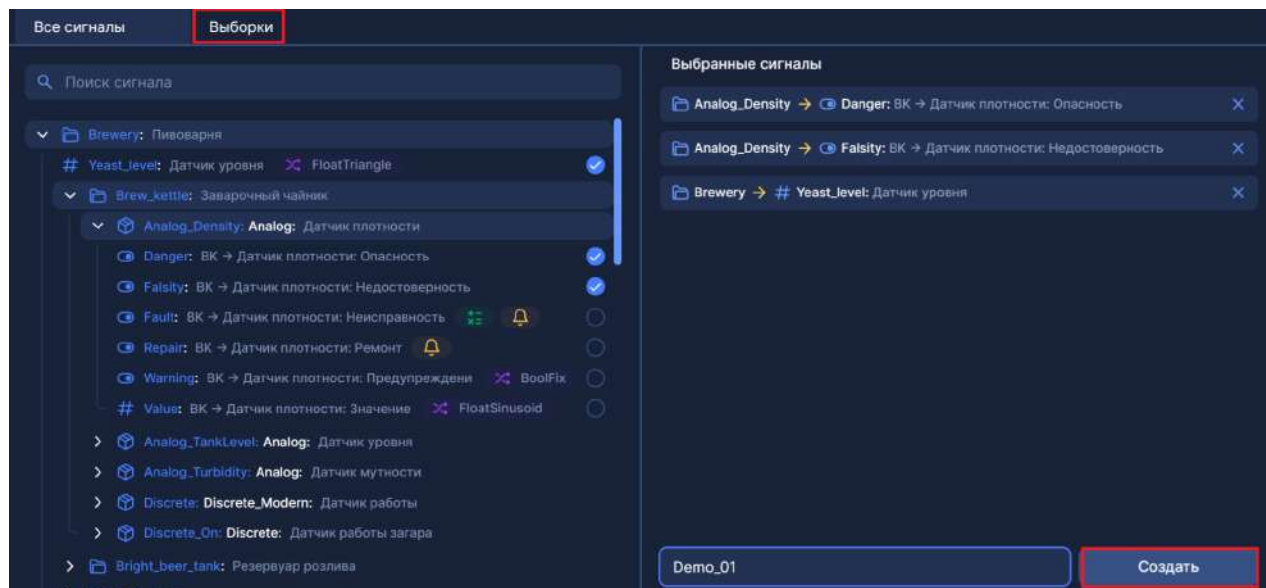


Рис. 4.26. Создание выборки

Созданные пользователем выборки будут находиться во вкладке «Выборки» (рис.4.27).

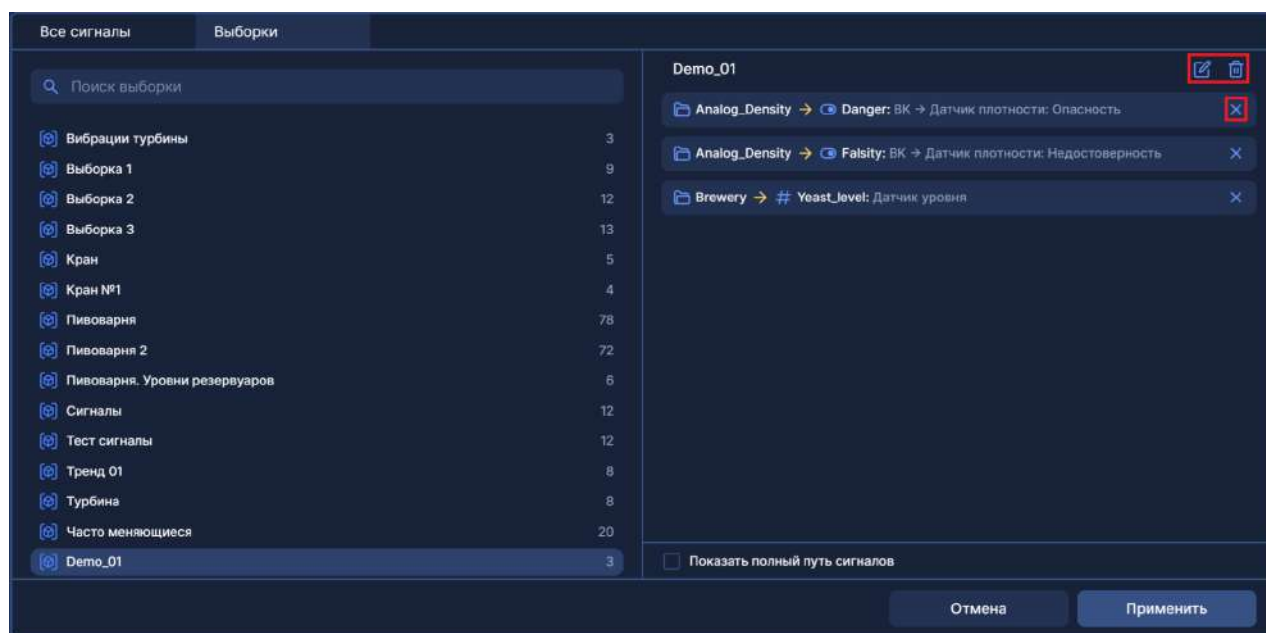


Рис. 4.27. Вкладка «Выборки»

Пользователь может отредактировать и удалить выборку путем нажатия соответствующей кнопки (рис. 4.27).

Для удаления сигналов из выборки необходимо нажать на кнопку редактирования выборки и затем нажать на крестик напротив соответствующего сигнала (рис. 4.27).

Таким же образом можно будет и добавить новые сигналы в выборку, после нажатия кнопки редактирования выборки откроется список сигналов доступных для выбора (рис. 4.28).

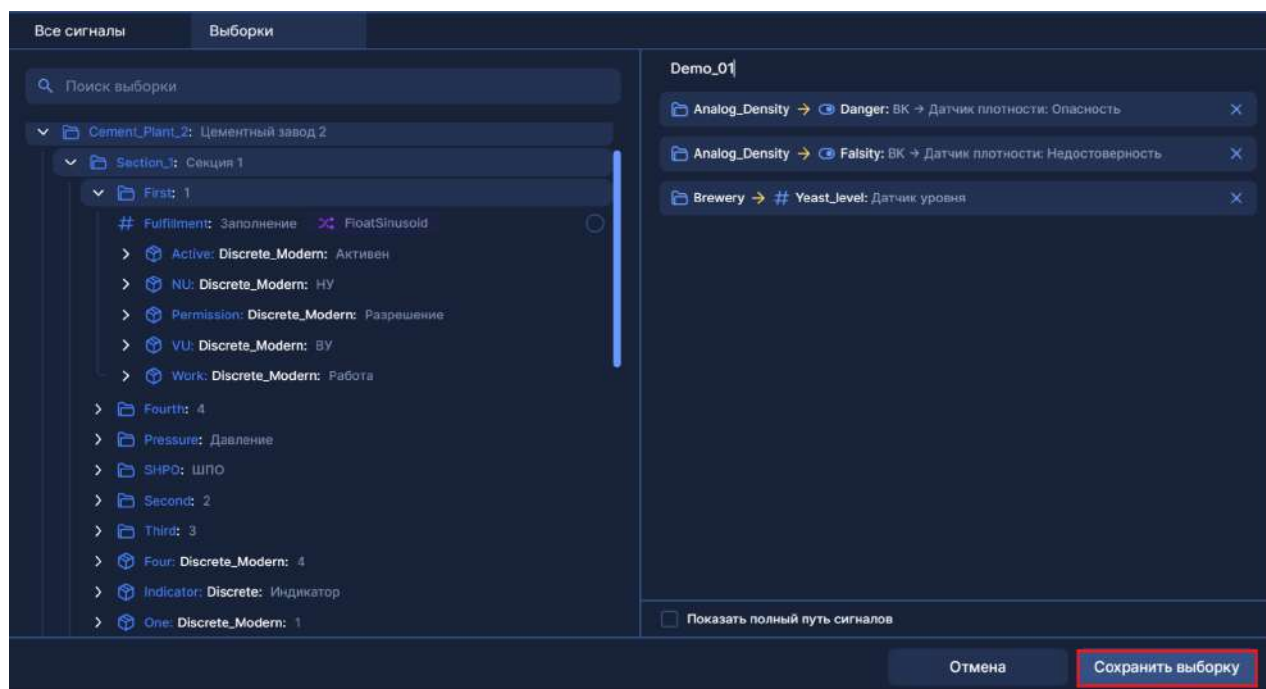


Рис. 4.28. Добавление новых сигналов в выборку

По окончании нажать «Сохранить выборку» для сохранения параметров выборки (рис. 4.28).

Пользователь может убрать сигналы из определенной выборки без ее прямого редактирования, тем самым оставив только необходимые в данный момент сигналы и не затронув заданные параметры выборки. Для этого необходимо нажать на нужную выборку и кликом на крестик убрать ненужные сигналы (Рис. 4.29).

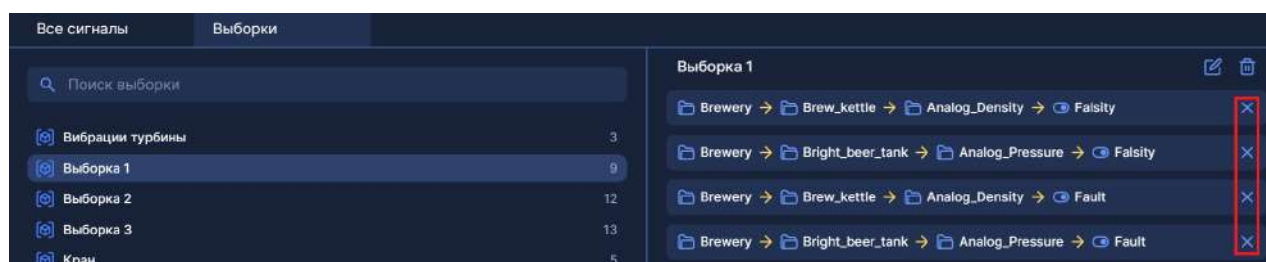


Рис. 4.29. Удаление сигналов без прямого редактирования

Для того чтобы переименовать выборку нужно воспользоваться кнопкой редактирования выборки (рис 4.27), затем кликнуть на имя выборки и задать новое имя (рис. 4.30).

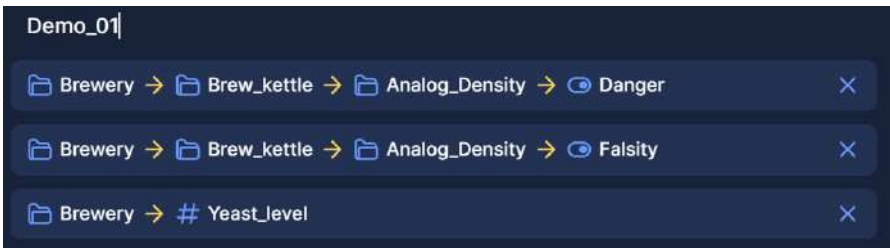


Рис. 4.30. Переименование выборки

После выбора сигналов откроется таблица в которой будет отображаться объект, его описание, значение, дата и время (рис. 4.31).

LACERTA Таблицы 04.07.2025 13:50:12 admin

1 минута 04.07.2025, 13:28:13 Все значения Выборка 1: 9 Обновить

Объект	Описание	Значение	Дата и время
Brewery > # Yeast_Level	Датчик уровня	82.22	04.07.2025 13:28:14
Brewery > # Yeast_Level	Датчик уровня	82.22	04.07.2025 13:28:14
Brewery > # Yeast_Level	Датчик уровня	82.22	04.07.2025 13:28:14
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:13
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:12
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:12
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	0	04.07.2025 13:28:12

1 2 3 4 5 6 7 ... 43 Показать Видимые

Рис. 4.31. Таблица данных

В таблице можно выбрать интервал времени за который будут выводиться данные (рис. 4.32).

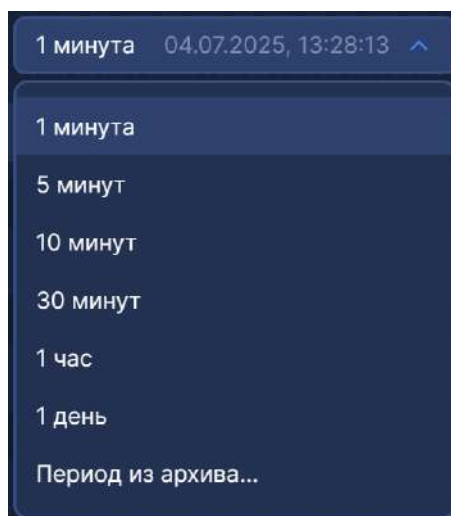


Рис. 4.32. Интервал вывода данных

Или задать свой собственный интервал, нажав на «Период из архива» (рис. 4.33).

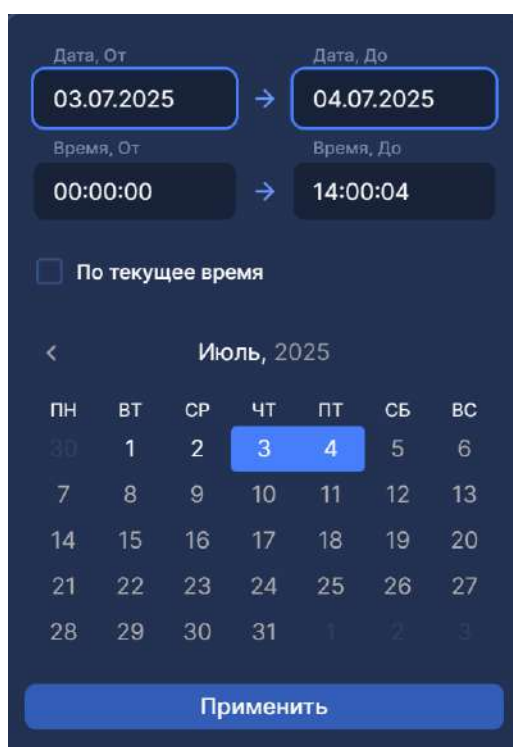


Рис. 4.33. Собственный интервал вывода данных

Помимо временного интервала, можно выбрать и тип таблицы (рис. 4.34).

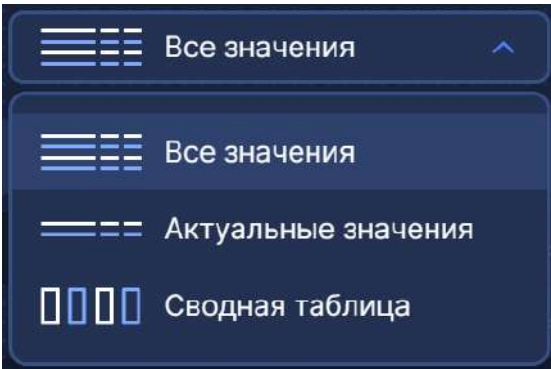


Рис. 4.34. Выбор типа таблицы

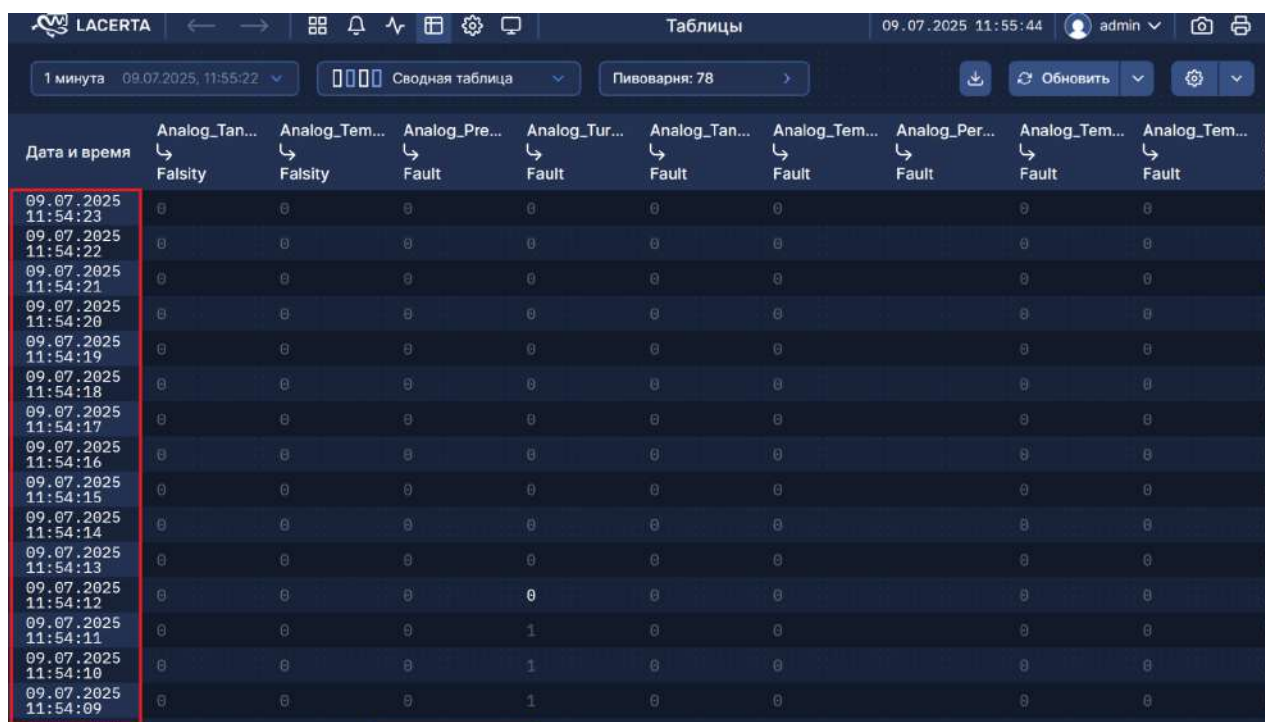
Кроме типа «Все значения» доступен вывод таблицы с актуальными значениями и вывод сводной таблицы (рис. 4.35 и рис. 4.36).

Объект	Описание	Значение	Дата и время
Analog_Pressure > # Value	ВВТ → Датчик давления: Значение	103.28	04.07.2025 14:12:18
Analog_Pressure > Falsity	ВВТ → Датчик давления: Недостоверность	0	04.07.2025 14:12:19
Analog_Pressure > Fault	ВВТ → Датчик давления: Неисправность	0	04.07.2025 14:12:18
Analog_Pressure > Repair	ВВТ → Датчик давления: Ремонт	0	04.07.2025 14:12:18
Analog_Density > Falsity	ВК → Датчик плотности: Недостоверность	1	23.06.2025 17:00:03
Analog_Density > Fault	ВК → Датчик плотности: Неисправность	1	04.07.2025 14:12:19
Analog_Density > Repair	ВК → Датчик плотности: Ремонт		
Discrete_On > Value	ВК → Датчик работы загара: Значение	1	04.07.2025 14:11:52
Brewery > # Yeast_level	Датчик уровня	84.44	04.07.2025 14:12:18

Рис. 4.35. Вывод таблицы с актуальными значениями

Рис.4.36. Вывод сводной таблицы

В сводной таблице присутствует фиксированный столбец с датой и временем в крайнем левом углу для удобства в навигации пользователя (рис. 4.37).



The screenshot shows the LACERTA software interface. At the top, there's a header with the LACERTA logo, navigation arrows, and a 'Таблицы' (Tables) tab. Below the header, there's a filter bar with a time range '1 минута' and a date '09.07.2025, 11:55:22'. A dropdown menu shows 'Сводная таблица' (Summary table). To the right, there's a button 'Пивоварня: 78' and a 'Обновить' (Refresh) button. The main area displays a table with the following columns: 'Дата и время' (Date and time), 'Analog_Tan... Falsity', 'Analog_Tem... Falsity', 'Analog_Pre... Fault', 'Analog_Tur... Fault', 'Analog_Tan... Fault', 'Analog_Tem... Fault', 'Analog_Per... Fault', 'Analog_Tem... Fault', and 'Analog_Tem... Fault'. The 'Дата и время' column is highlighted with a red box, indicating it is a fixed column. The table contains 15 rows of data, all showing '0' for the fault columns.

Дата и время	Analog_Tan... Falsity	Analog_Tem... Falsity	Analog_Pre... Fault	Analog_Tur... Fault	Analog_Tan... Fault	Analog_Tem... Fault	Analog_Per... Fault	Analog_Tem... Fault	Analog_Tem... Fault
09.07.2025 11:54:23	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:22	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:21	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:20	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:19	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:18	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:17	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:16	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:15	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:14	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:13	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:12	0	0	0	0	0	0	0	0	0
09.07.2025 11:54:11	0	0	0	1	0	0	0	0	0
09.07.2025 11:54:10	0	0	0	1	0	0	0	0	0
09.07.2025 11:54:09	0	0	0	1	0	0	0	0	0

Рис. 4.37. Фиксированный столбец

Таблицы могут быть экспортированы в файл формата Excel для дальнейшей обработки. Для экспорта таблиц можно воспользоваться кнопками «Постраничный отчёт» или «Общий отчёт» (рис. 4.37). После выбора формы отчета будет предложено сохранить файл в формате Excel.

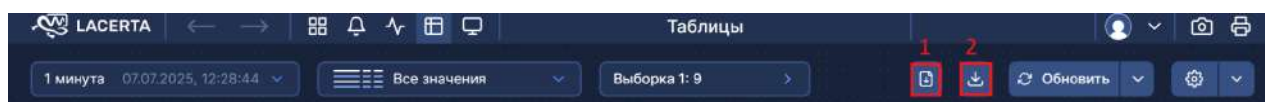


Рис.4.37. Кнопки меню «Таблицы»

Цифрами на рис. 4.37 обозначено:

1. Кнопка «Постраничный отчёт»
2. Кнопка «Общий отчёт»

После сохранения в формате Excel таблицы отобразятся в подобном виде (Рис. 4.38 и 4.39).

[illegible]

Рис. 4.38. Постраничный отчет в формате Excel

В постраничном отчете (рис. 4.38) информация о каждом объекте располагается на отдельных страницах. На странице содержится имя объекта, описание, дата и время, его значение. В нижней части таблицы имеется возможность переключаться между объектами.

[illegible]

Рис. 4.39. Общий отчет в формате Excel

В общем отчете (рис. 4.39) информация об объектах отображена на единой странице. Он также содержит имя объекта, описание, дату и время, его значение.

У сводного типа таблиц также присутствует возможность выгрузки отчета в формат Excel (рис. 4.40).



Рис. 4.40. Выгрузка отчёта в формат Excel

Информация об объектах будет располагаться аналогично информации в таблицах (рис. 4.41).

 The screenshot shows an Excel spreadsheet with data from a summary table report. The data is organized into columns labeled A through S. The first column (A) contains dates from 2025-07-1 to 2025-07-1. The second column (B) contains status values like 'ЛОЖЬ' (False) and 'ИСТИНА' (True). The third column (C) contains numerical values like 85,0536 and 101,364. The fourth column (D) contains status values like 'ЛОЖЬ' and 'ЛОЖЬ'. The fifth column (E) contains status values like 'ЛОЖЬ' and 'ЛОЖЬ'. The sixth column (F) contains status values like 'ЛОЖЬ' and 'ЛОЖЬ'. The seventh column (G) contains status values like 'ЛОЖЬ' and 'ЛОЖЬ'. The eighth column (H) contains numerical values like 83,6087 and 83,6087. The ninth column (I) contains numerical values like 83,6401 and 83,6401. The tenth column (J) contains numerical values like 83,6401 and 83,6401. The eleventh column (K) contains numerical values like 83,6716 and 83,6716. The twelfth column (L) contains numerical values like 83,703 and 83,703. The thirteenth column (M) contains numerical values like 83,7344 and 83,7344. The fourteenth column (N) contains numerical values like 83,7659 and 83,7659. The fifteenth column (O) contains numerical values like 83,7973 and 83,7973. The sixteenth column (P) contains numerical values like 83,7973 and 83,7973. The seventeenth column (Q) contains numerical values like 83,8287 and 83,8287. The eighteenth column (R) contains numerical values like 83,8601 and 83,8601. The nineteenth column (S) contains numerical values like 83,8916 and 83,8916. The twentieth column (T) contains numerical values like 83,923 and 83,923. The twenty-first column (U) contains numerical values like 83,9544 and 83,9544. The twenty-second column (V) contains numerical values like 83,9544 and 83,9544.

Рис. 4.41. Отчет сводной таблицы в формате Excel

Для того чтобы обновить информацию за выбранный период в архив, нужно воспользоваться кнопкой «Обновление запроса периода в архив», предварительно выбрав промежуток, за который ее отобразить (рис.4.38).

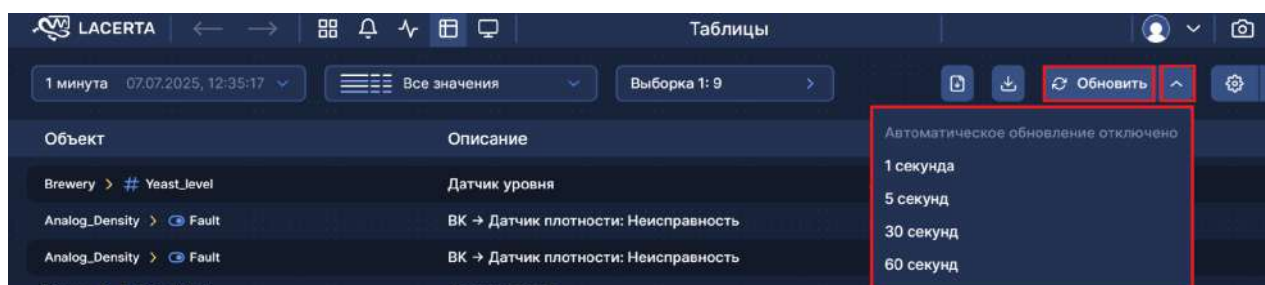


Рис. 4.38. Кнопка обновления запроса периода в архив

Пользователь может выбрать колонки, которые не нужно отображать в таблице. Для этого необходимо нажать на кнопку с пиктограммой «шестеренки» в крайнем правом углу страницы (рис.4.39).

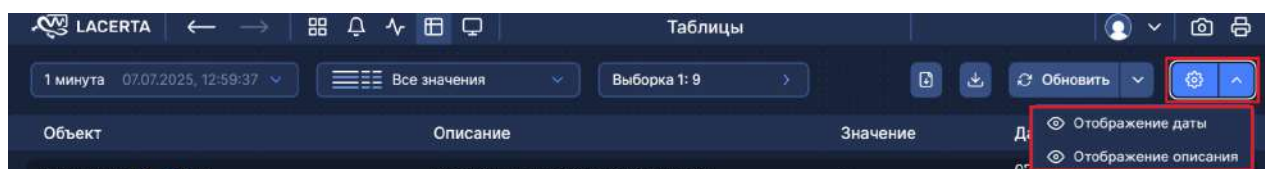


Рис. 4.39. Настройка отображения

При нажатии на объект в таблице пользователя перенаправит на конкретный выбранный объект в разделе меню «Объекты» (рис. 4.40 и 4.41).

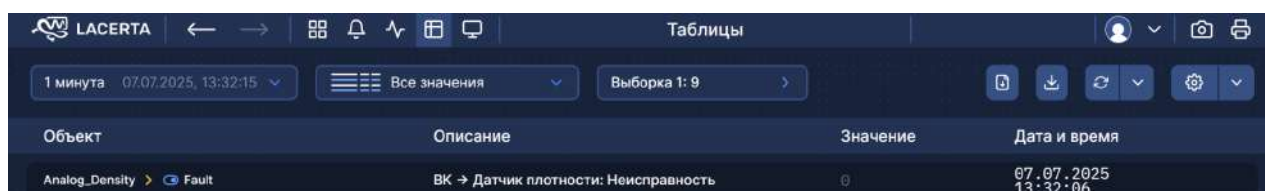


Рис. 4.40. Объект Analog_density

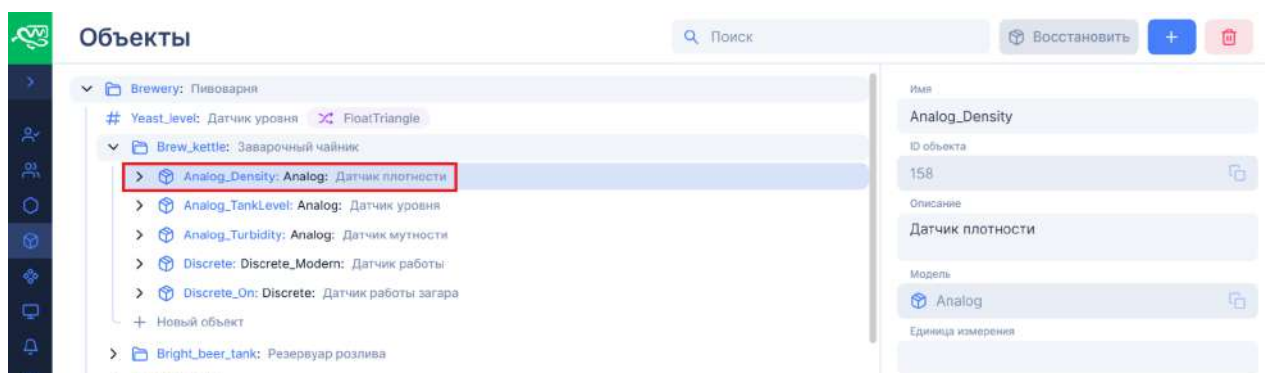


Рис. 4. 41. Раздел меню «Объекты»

Для того чтобы отобразить нужное количество объектов в таблице, необходимо воспользоваться кнопкой «Показать» в правом нижнем углу страницы (рис. 4.42).

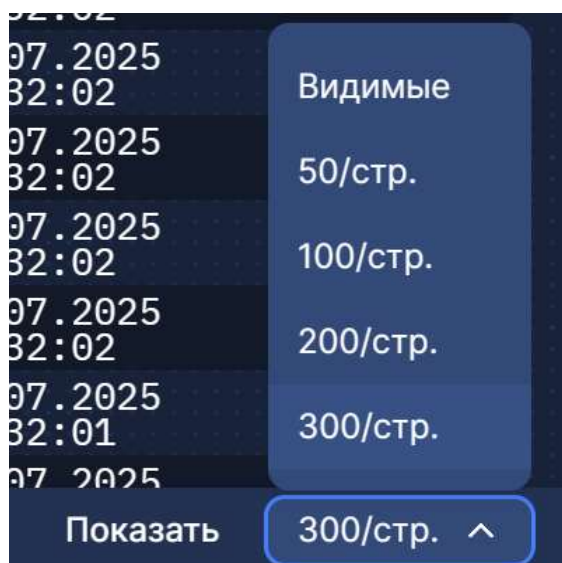


Рис. 4. 42. Кнопка «Показать»

Для переключения между страницами таблицы можно воспользоваться соответствующими кнопками в нижнем правом углу страницы (рис. 4.43).



Рис. 4.43. Переключение между страницами таблицы

4.5. Тренды

Данные Платформы могут быть представлены в виде трендов (графиков). Для перехода в раздел «Тренды» необходимо нажать на соответствующую кнопку в верхнем меню слева. (рис. 4.35).



Рис. 4. 35. Переход в Тренды

После перехода в раздел «Тренды» появляется стартовый экран (рис. 4.36)

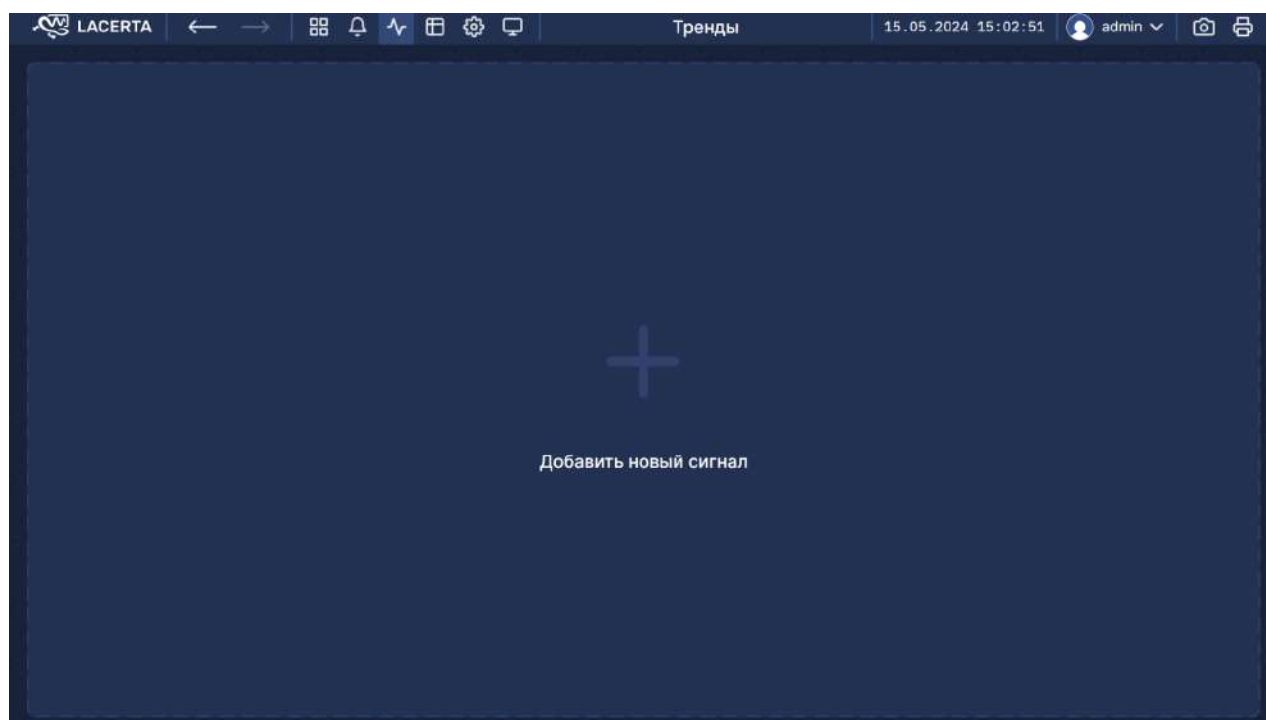


Рис. 4.36. Стартовый экран Тренды

Для построения графиков необходимо нажать на окно «+ Добавить новый сигнал» (рис. 4.36), после чего определить набор отображаемых значений сигналов (от 1 до 12), либо выбрать уже готовую выборку (перейдя на вкладку «Выборки»). По завершении нажать «Применить» (рис. 4.37).

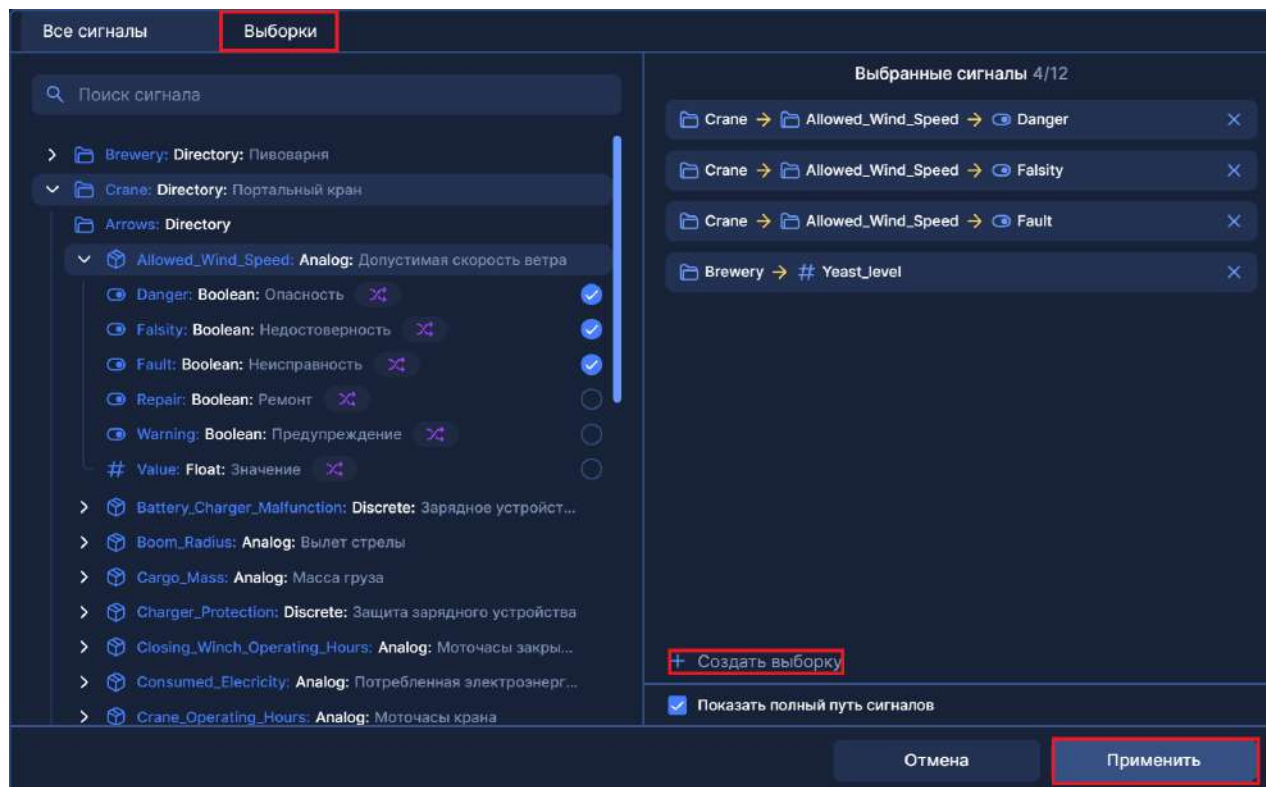


Рис. 4.37. Выбор сигналов

При нажатии флажка «Показать полный путь сигналов» в списке выбора будет показана вся ветвь иерархической структуры объекта (рис. 4.37). Этот способ полезен при работе с объектами, имеющими сложную структуру.

Как и в случае с таблицами, пользователь может сохранить настройки в виде выборки для дальнейшего использования. Для этого нужно нажать на кнопку «+ Создать выборку» (рис. 4.37) и затем задать название выборки. По завершении нажать «Создать» (рис. 4.38). Созданные выборки будут располагаться во вкладке «Выборки» (рис. 4.38).

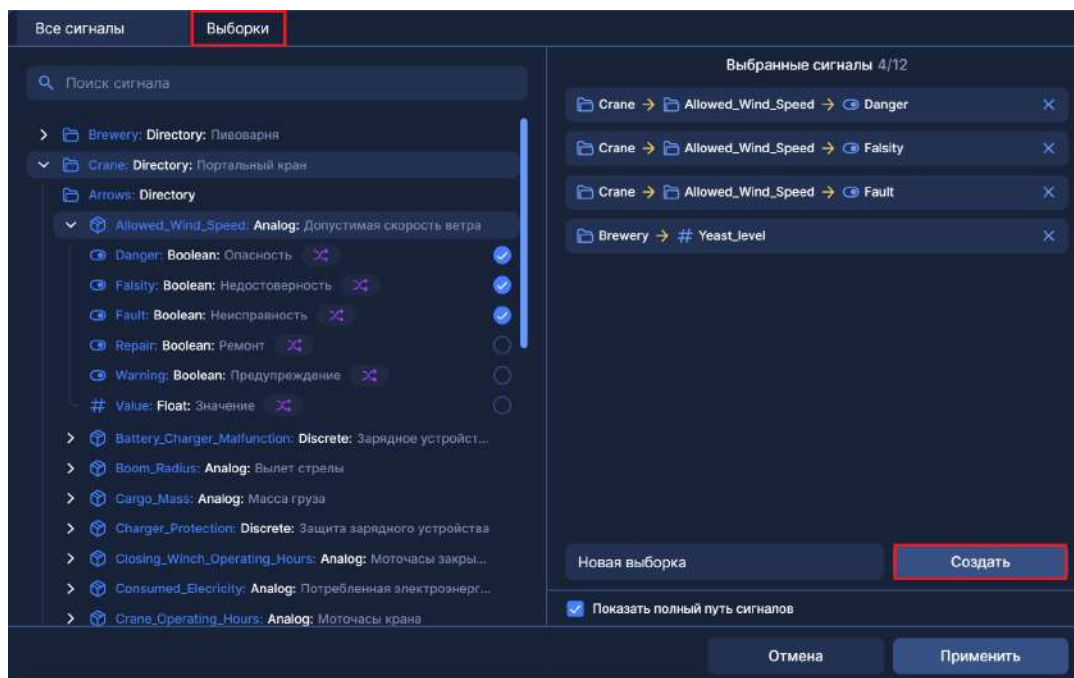


Рис. 4.38. Создание выборки

При попытке пользователем добавления в Тренды выборки с более 12 сигналами Системой выводится предупреждение (рис. 4.39).

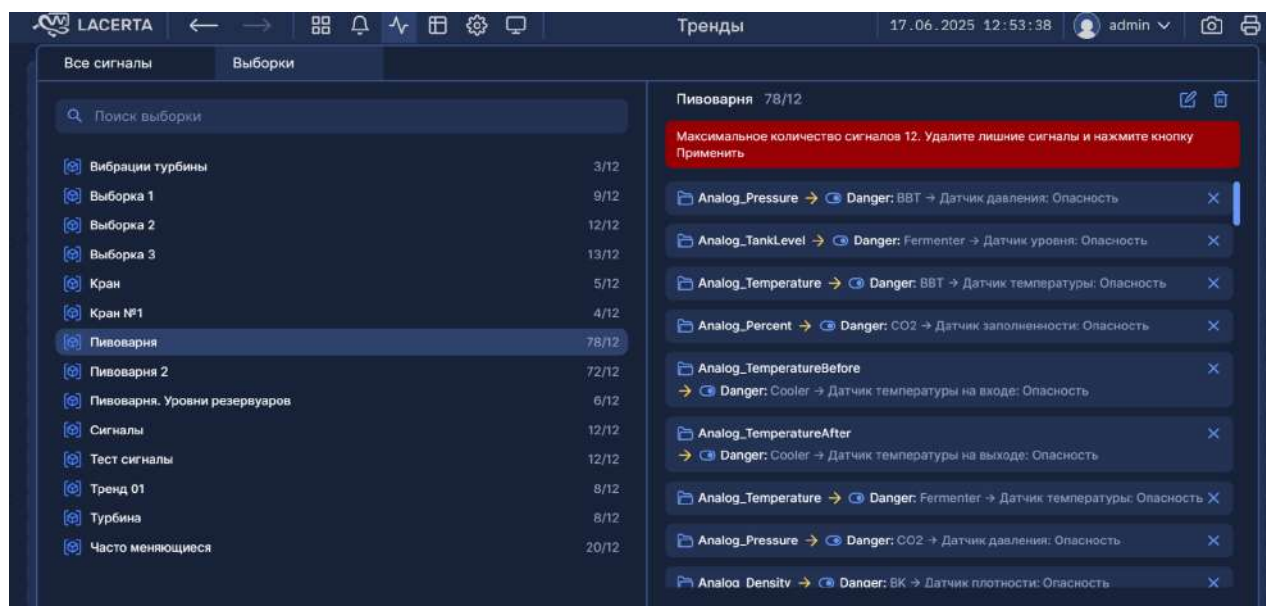


Рис. 4.39. Предупреждение Системы

Пользователь может отредактировать и удалить выборку путем нажатия соответствующей кнопки после клика на выборку (рис. 4.40).

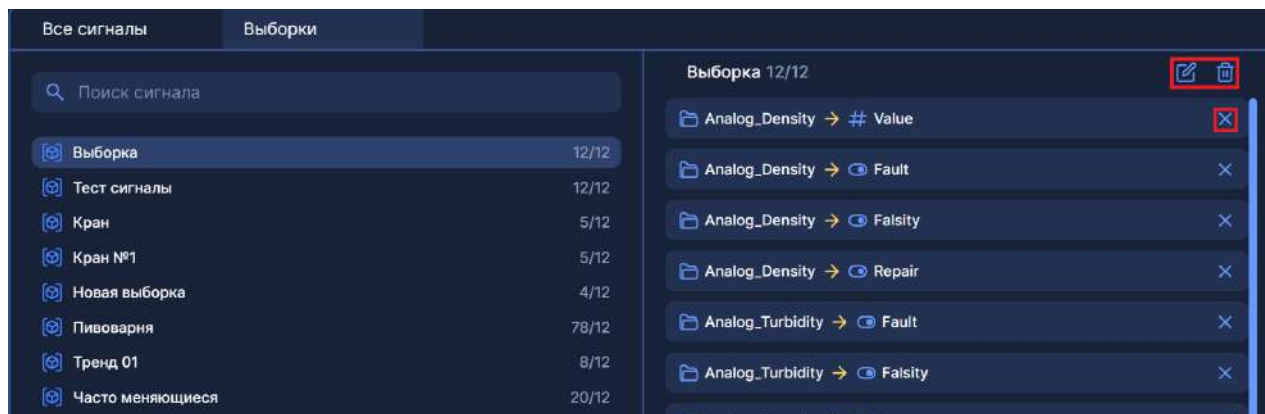


Рис. 4.40. Кнопки редактирования и удаления выборки

Для удаления сигналов из выборки нужно нажать на кнопку редактирования выборки и затем нажать на крестик напротив соответствующего сигнала (рис. 4.40).

По окончании нажать «Сохранить выборку» для сохранения параметров выборки (рис. 4.41).

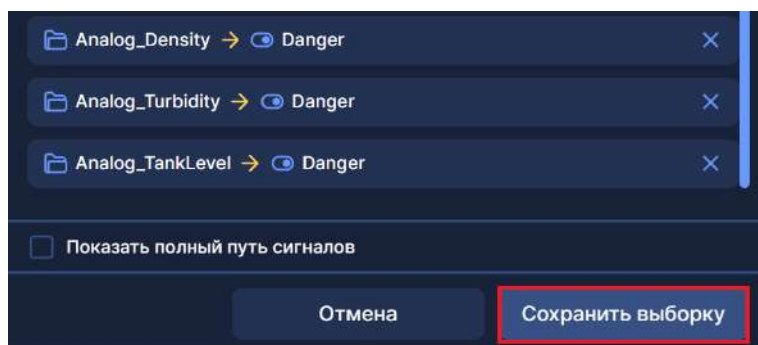


Рис. 4.41. Сохранение выборки

Пользователь имеет возможность убрать сигналы из определенной выборки без ее прямого редактирования, тем самым оставив только необходимые в данный момент сигналы и не затронув заданные параметры выборки. Для этого необходимо нажать выборку и кликом на крестик убрать ненужные сигналы (Рис. 4.40).

По окончании нажать «Применить» для перехода на график с актуальными в данный момент сигналами в выборке (Рис.4.42).

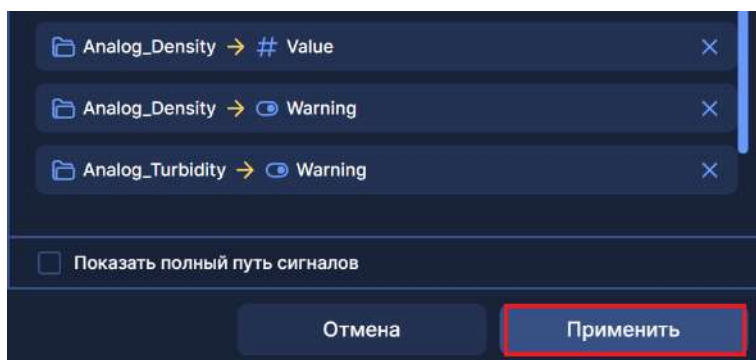


Рис. 4.42. Кнопка перехода на график

Для того чтобы переименовать выборку нужно нажать на кнопку редактирования выборки (рис 4.40), затем кликнуть на имя выборки и задать новое имя (рис. 4.43).

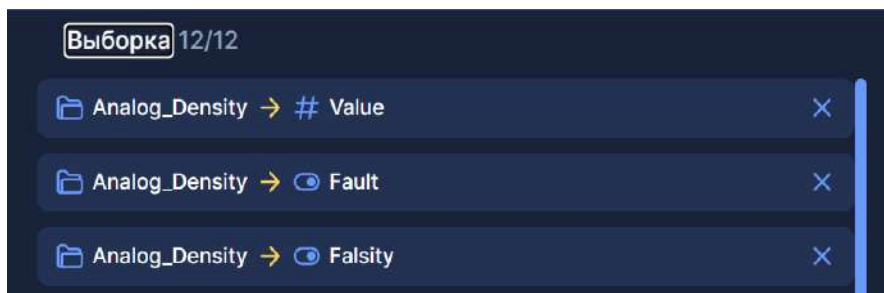


Рис. 4.43. Переименование выборки

После добавления сигналов на главном экране отобразится график трендов (рис. 4.44).



Рис. 4.44. Отображение графика трендов

При наведении на график у трендов отображаются тултипы со значениями (рис. 4.45).

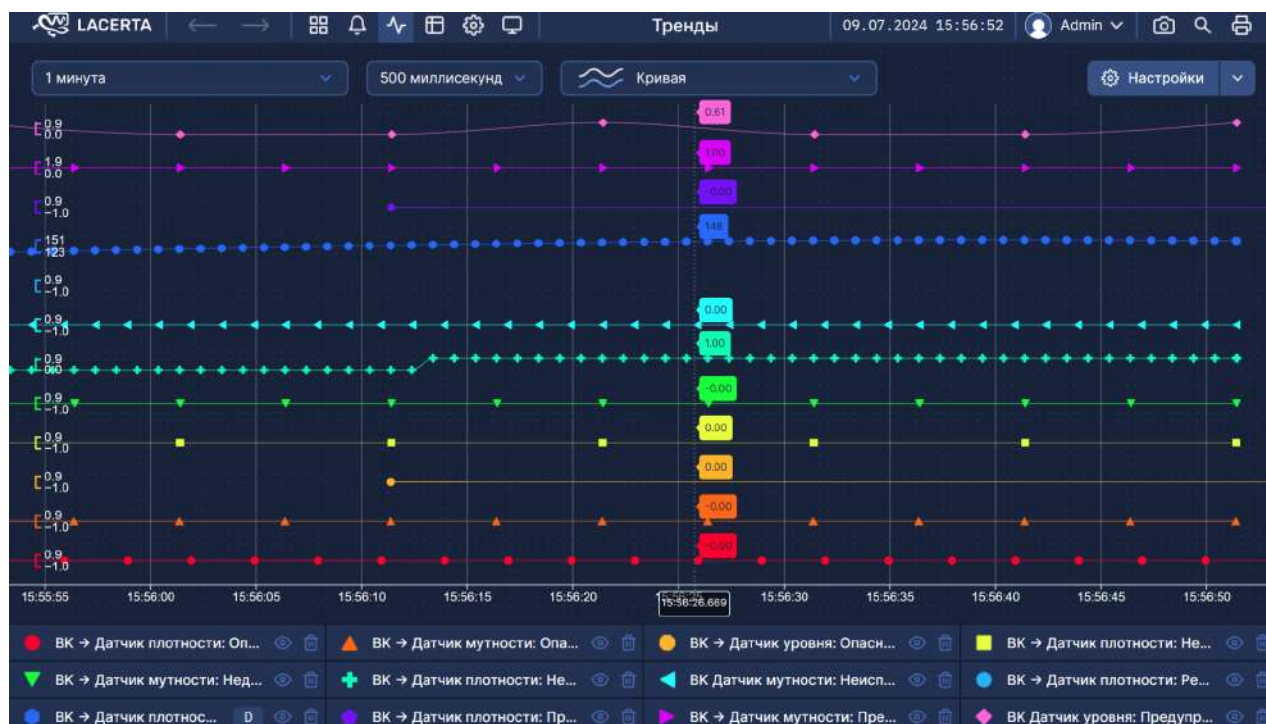


Рис. 4.45. Отображение тултипов

Для отключения маркеров на графике и в нижней панели необходимо нажать на кнопку «Отображение маркеров», развернув параметр «Настройки» (рис. 4.46).

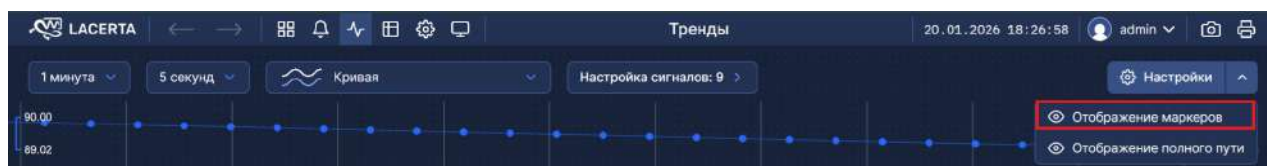


Рис. 4.46. Отключение отображения маркеров

В режиме выключенного отображения маркеров возможно подсвечивание графика при наведении курсора на соответствующий сигнал в нижней панели (рис. 4.47).

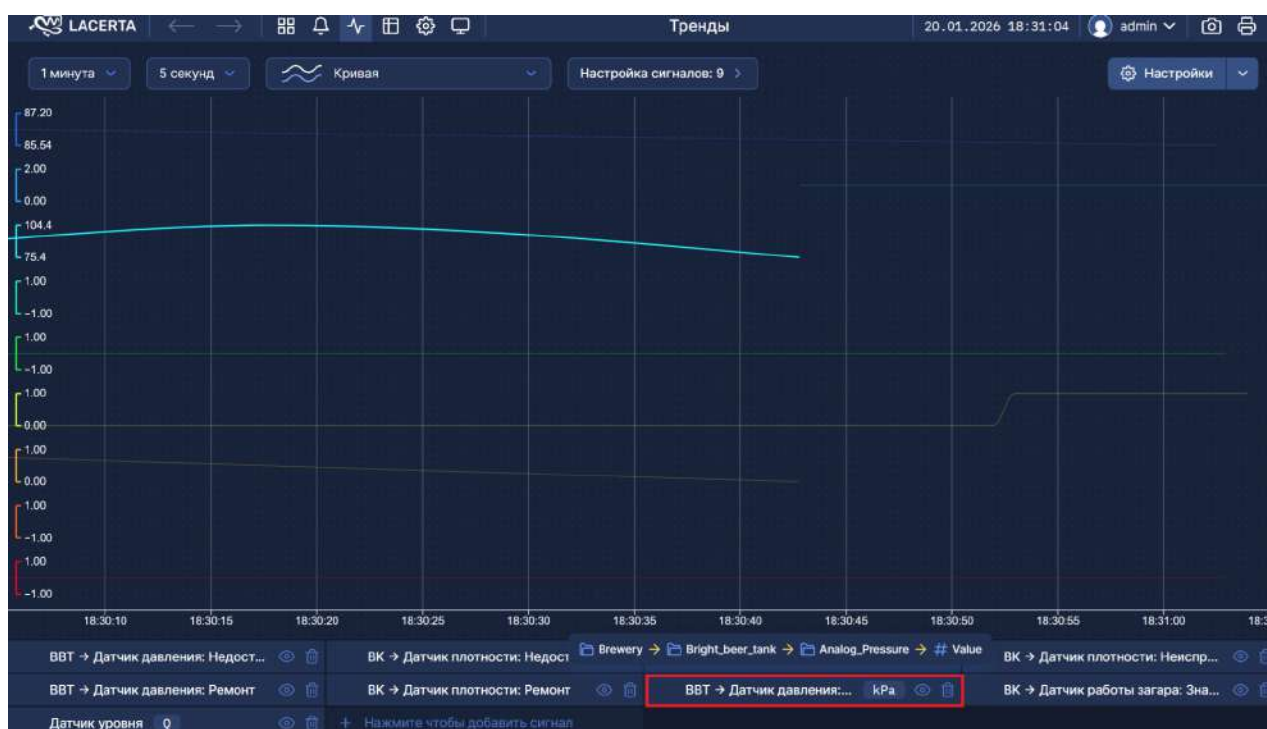


Рис. 4.47. Подсвечивание сигнала на графике

Для отображения полного пути объекта при наведении на сигнал необходимо нажать на кнопку «Отображение полного пути», развернув параметр «Настройки» (рис. 4.48).



Рис. 4. 48. Отображение полного пути объекта

Для того чтобы скрыть тренд на графике нужно нажать на пиктограмму с изображением глаза напротив сигнала в нижней панели (рис. 4.48).

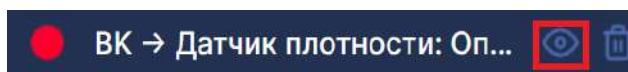


Рис. 4.48. Пиктограмма скрытия сигнала с графика

Для удаления сигнала необходимо нажать на пиктограмму с соответствующим изображением (рис. 4.49).

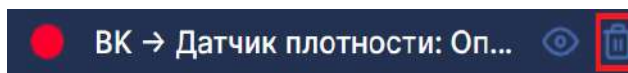


Рис. 4.49. Удаление сигнала

При наведении курсора на сигнал отображается его полный путь (рис. 4.50).



Рис. 4.50. Отображение полного пути сигнала

Нажатие на любую из папок из пути сигнала переводит пользователя в боковой раздел меню «Объекты» (рис. 4. 51).

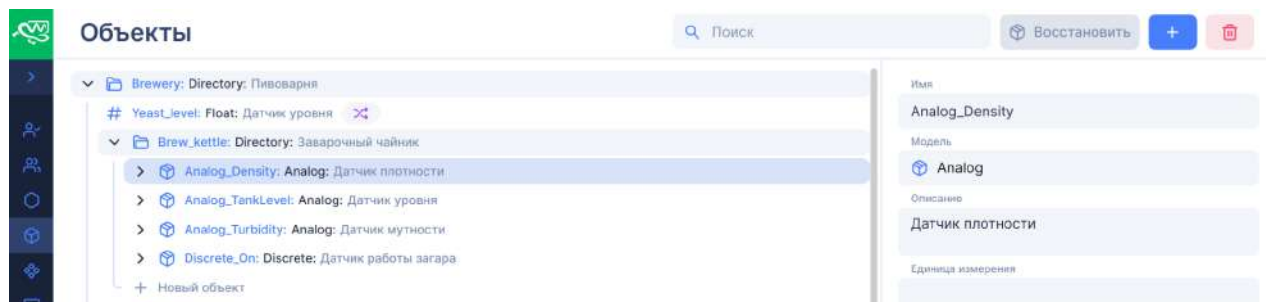


Рис. 4.51. Переход в раздел меню «Объекты»

У пользователя имеется возможность масштабирования участка графика. Для этого необходимо зажать выбранный участок графика и растянуть его на необходимую ширину (рис. 4.52).

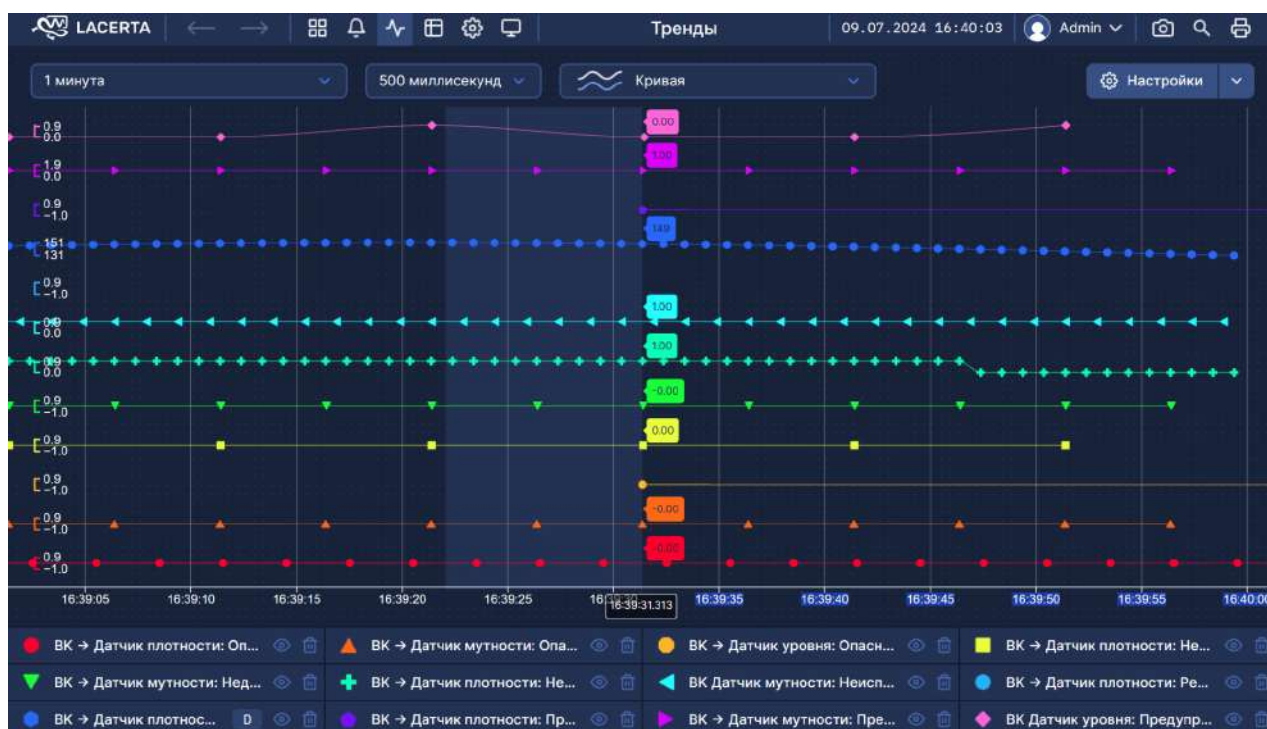


Рис. 4.52. Масштабирование участка графика

Границы выбранного участка будут отображаться в дропдауне периода. Масштабировать можно сколько угодно глубоко. У пользователя есть возможность вернуться к предыдущему периоду либо сбросить период к первоначальному (до масштабирования). Для этого необходимо воспользоваться соответствующими кнопками в окне дропдауна периода (рис. 4.53).



Рис. 4.53. Кнопка возврата к исходным параметрам

Пользователь может задать период (Временной отрезок), временной интервал между значениями и выбрать вид тренда (рис. 4.54).

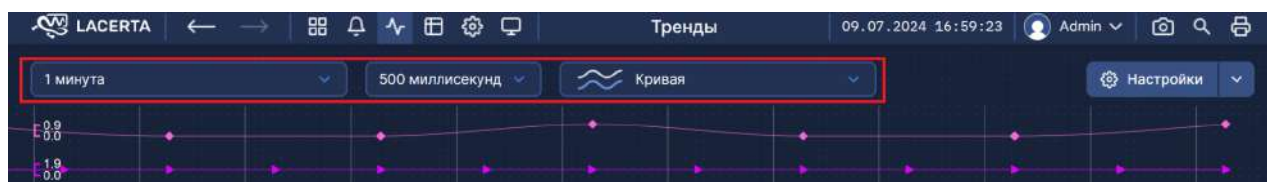


Рис. 4.54. Параметры трендов

Выбранный период (временной отрезок) за который нужно отобразить данные, делится на временные интервалы. В каждом интервале находятся точки минимума и максимума которые отображаются на графике (рис. 4.55).

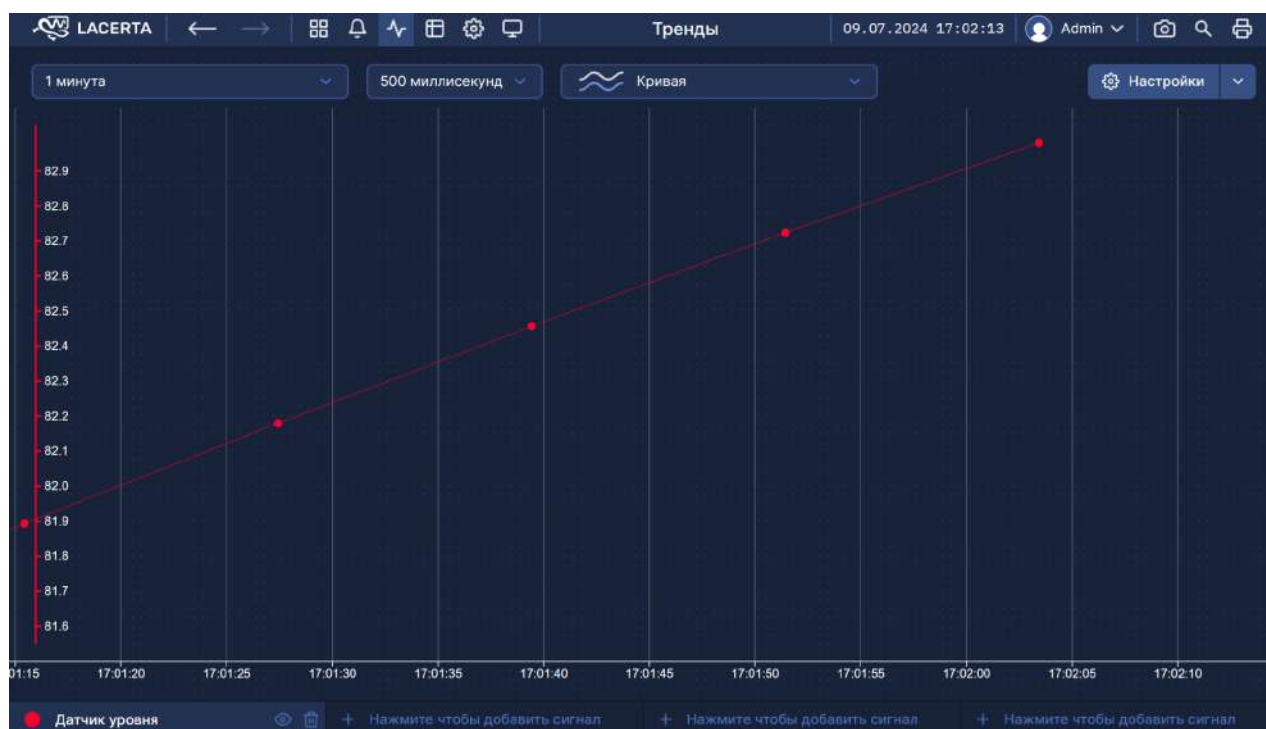


Рис. 4.55 . Временные интервалы

Доступные периоды и интервалы к ним (рис. 4.55) задаются в пункте бокового меню «Настройки» во вкладке «Тренды».

Для того чтобы задать конфигурацию Трендов необходимо перейти на страницу «Настройки» в боковом меню, затем нажать на «Тренды» (рис. 4.56).

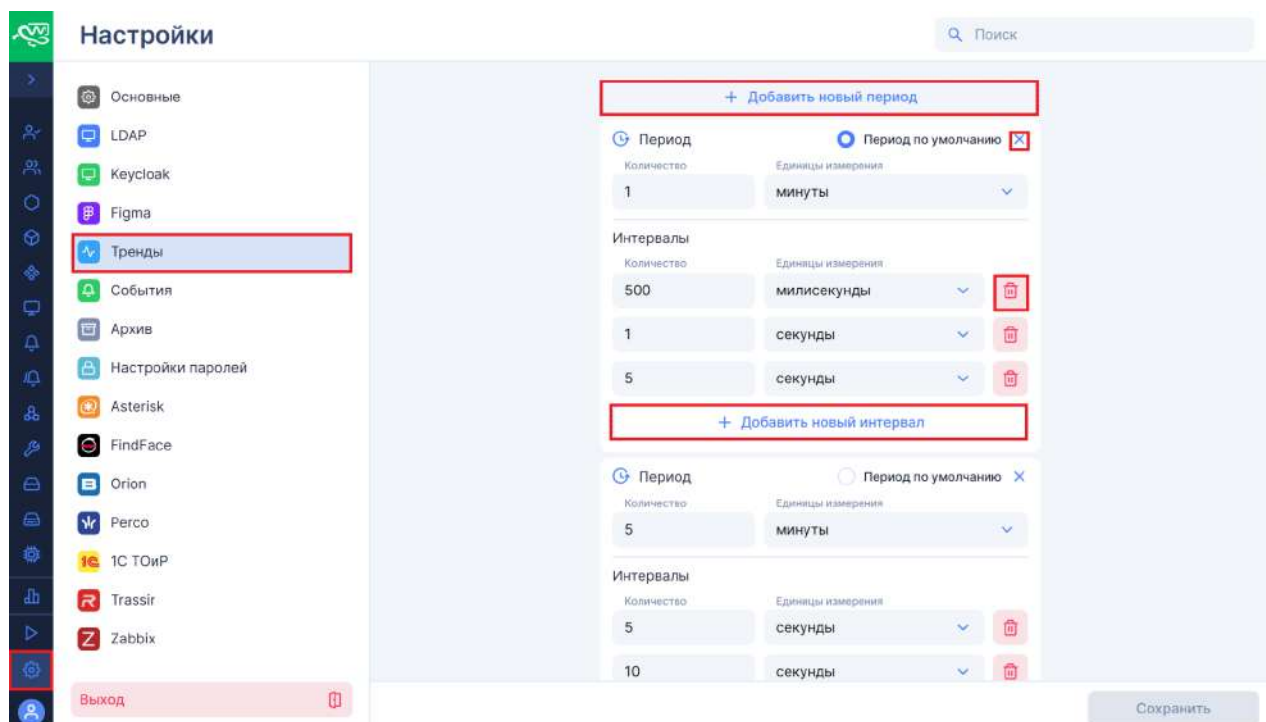


Рис. 4.56. Конфигурация Трендов

Настройка позволяет создавать периоды Трендов и к каждому периоду его собственные интервалы.

Для того чтобы создать новый период, необходимо нажать на кнопку «+ Добавить новый период», после чего к этому периоду можно будет добавить интервалы кнопкой «+ Добавить новый интервал». По окончании нажать на «Сохранить» для сохранения заданной конфигурации (рис. 4.56).

Для того чтобы удалить период (вместе с ним удаляются и заданные интервалы) необходимо нажать на крестик в углу параметров периода (рис. 4.56).

Удалить можно только периоды без значения «Период по умолчанию». Удаление таких периодов возможно только при переносе данного значения на другие периоды.

Для того чтобы удалить интервал, нужно нажать на кнопку удаления, обозначенную соответствующей пиктограммой корзины напротив интервала (рис. 4.56).

Созданные периоды и интервалы к ним будут отображаться на странице Тренды в дропдауне выбора периода и дропдауне выбора интервала (рис. 4.57).

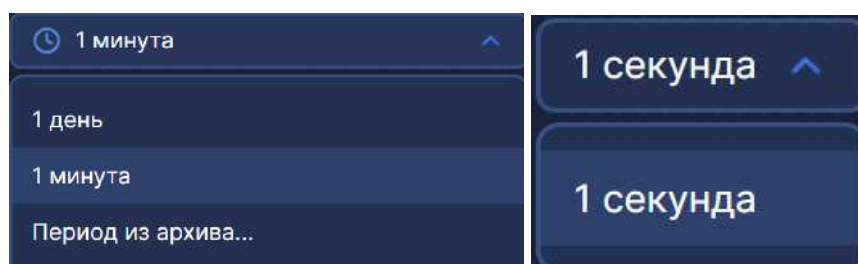


Рис. 4.57. Дропдаун выбора периода/интервала

Период должен быть строго больше интервала, иначе отобразится предупреждение и период нельзя будет сохранить (рис. 4.58).

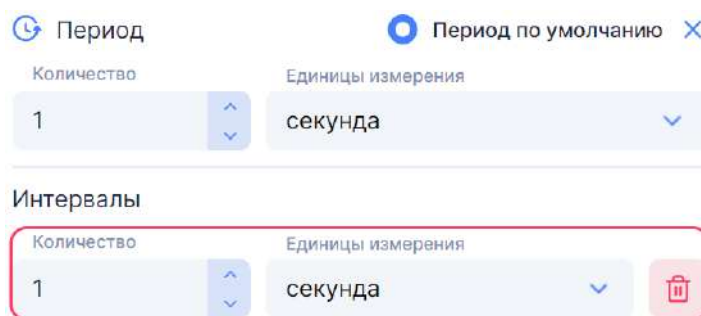


Рис. 4.58. Предупреждение Системы

Такое же предупреждение отобразится и при попытке создания одинаковых периодов (рис. 4.59)

The image shows two identical, vertically stacked panels, each representing a 'Период' (Period) configuration. Each panel has a title bar with a clock icon and the word 'Период', and a close button 'X'. Below the title bar, there are two radio buttons: 'Период' (selected) and 'Период по умолчанию'. Underneath, there are two input fields: 'Количество' (Quantity) with the value '1' and 'Единицы измерения' (Units of measurement) with the value 'секунда'. At the bottom of each panel, there is a section titled 'Интервалы' (Intervals) containing a dashed box with a blue plus icon and the text '+ Добавить новый интервал'.

Рис. 4.59. Предупреждение Системы

Аналогичное предупреждение будет и при создании одинаковых интервалов в рамках одного периода (рис. 4.60).

The image shows a single 'Период' (Period) configuration panel. It has the same title bar and radio buttons as in the previous figure. The 'Количество' (Quantity) field has the value '1' and the 'Единицы измерения' (Units of measurement) field has the value 'секунда'. Below the 'Интервалы' (Intervals) section, there are two identical rows. Each row has a 'Количество' (Quantity) field with the value '1' and an 'Единицы измерения' (Units of measurement) field with the value 'миллисекунда'. To the right of each 'Единицы измерения' field is a red trash can icon, indicating that these intervals can be deleted.

Рис. 4.60. Предупреждение Системы

У пользователя существует возможность отображения периода по умолчанию в Трендах. Для того того нужно кликнуть на чекбокс «Период по умолчанию» после чего нажать «Сохранить» (рис. 4.61).

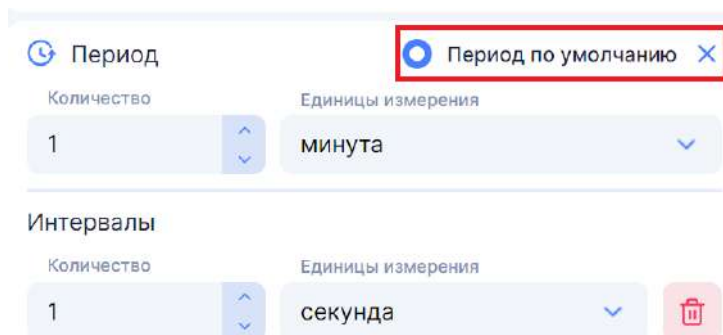


Рис. 4.61. Чекбокс «Период по умолчанию»

Теперь на странице Тренды в дропдауне выбора периода всегда будет отображаться заданный период по умолчанию (рис. 4.62)

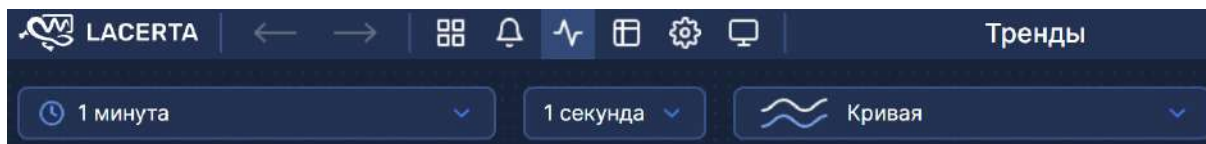


Рис. 4.62. Период по умолчанию

4.6. События

События являются динамическими объектами платформы и создаются как реакция на изменения значения переменных (разд. 2.4).

Для того чтобы перейти в раздел «События» необходимо нажать на соответствующую кнопку в верхнем меню страницы (рис. 4.63).



Рис. 4.63. Панель кнопок меню

После чего откроется список текущих событий (рис. 4.64).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Warning	24.04.2026 14:28:02	Заварочный резервуар: Объем жидкости в норме	Активное	Событие 3	3	Квитировать
Warning	24.04.2026 14:27:15	Резервуар розлива: Температура превысила значение в 220 °C	Активное	Предупреждение 3	3	Квитировать
Fault	24.04.2026 14:27:01	Заварочный резервуар: Датчик мутности неисправен	Активное	Тревога 2	6	Квитировать
Value	24.04.2026 14:26:22	Резервуар розлива: выключен	Активное	Событие 2	2	Квитировать
Fault	24.04.2026 14:26:20	Заварочный резервуар: датчик уровня исправен	Активное	Событие 1	1	Квитировать
Fault	24.04.2026 14:25:32	Test	Не активное	Авария	10	Квитировать
Repair	24.04.2026 14:24:42	CO2: Датчик давления в ремонте	Активное	Событие 3	3	Квитировать
Danger	22.04.2026 20:48:08	Резервуар розлива: переполнен	Не активное	Тревога 1	8	Квитировать
Fault	01.04.2026 10:52:13	Test	Не активное	Авария	10	Квитировать
Repair	23.03.2026 14:38:48	CO2: Датчик давления в ремонте	Не активное	Событие 3	3	Квитировать
Danger	23.03.2026 14:30:42	Резервуар розлива: переполнен	Не активное	Тревога 1	8	Квитировать
Fault	23.03.2026 14:30:29	Заварочный резервуар: датчик уровня исправен	Не активное	Событие 1	1	Квитировать
Fault	23.03.2026 14:30:29	Заварочный резервуар: Датчик мутности неисправен	Не активное	Тревога 2	6	Квитировать
Warning	23.03.2026 14:30:29	Резервуар розлива: Температура превысила значение в 220 °C	Не активное	Предупреждение 3	3	Квитировать
Warning	23.03.2026 14:30:29	Заварочный резервуар: Объем жидкости в норме	Не активное	Событие 3	3	Квитировать

Рис. 4.64. Страница «События»

Список событий отображается в реальном времени и содержит:

- объекты;
- дату и время наступления события;
- сообщение, определённое пользователем для данного события (разд. 3.9.1);
- состояние события (активное/неактивное);
- класс события (значение, определенное пользователем в настройках событий);
- приоритет события (значение, определенное пользователем в настройках событий, от 1 до 10);
- кнопку действия (возможность квитирования «подтверждения» события).

Для подтверждения получения информации о событии, нужно нажать на кнопку «Квитировать» в соответствующей строке события (рис. 4.64).

После квитирования события вместо кнопки "Квитировать" выводится имя пользователя и время квитирования события (рис. 4.65).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Warning	24.04.2026 14:29:02	Заварочный резервуар: Объем жидкости в норме	Активное	Событие 3	3	admin 24.04.2026 14:29:19

Рис. 4.65. Квитирование события

В списке событий присутствуют и те события, которые не имеют возможности квитирования (рис. 4.66).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Wheel_mode	24.04.2026 14:34:45	Кран 04: Карусельный режим колес	Активное	Класс события 1	2	

Рис. 4.66. Событие без возможности квитирования

Пользователь может отфильтровать список событий по их объектам, дате и времени, состоянию, классу, приоритету и возможности квитирования. Для этого необходимо нажать на кнопку фильтра напротив каждого параметра (рис. 4.67).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие

Рис. 4.67. Фильтрация списка событий

Для того чтобы отфильтровать список событий по объектам нужно нажать на кнопку фильтра и после этого на «+ Добавить объект» (рис. 4.68).

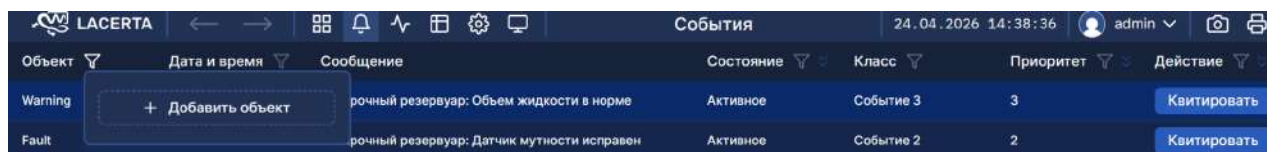


Рис. 4.68. Фильтрация событий по объектам

Откроется модальное окно в котором необходимо выбрать объект и нажать кнопку «Применить» (рис. 4.69).

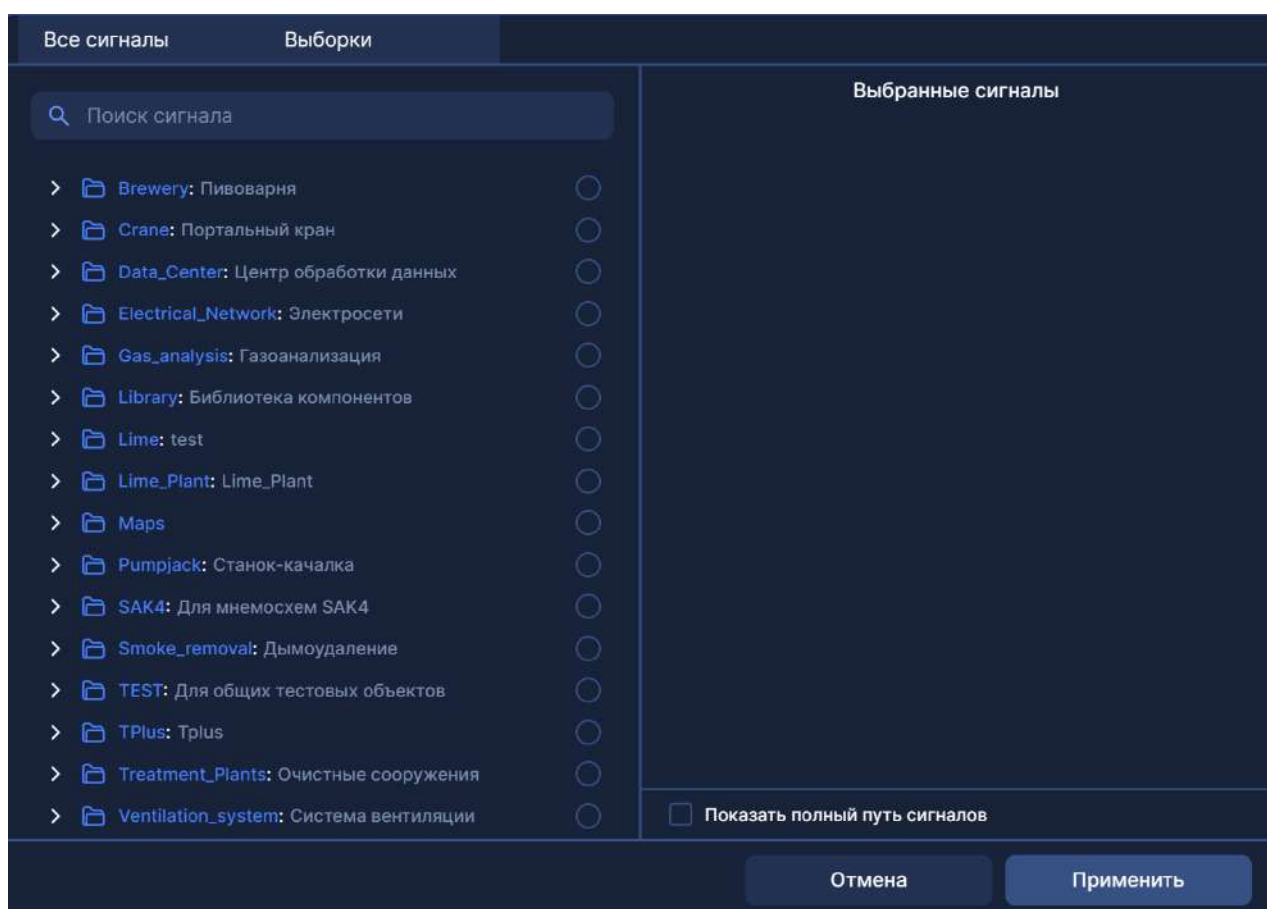


Рис. 4.69. Окно выбора объекта

После этого отобразятся только события с выбранным объектом из фильтра (рис. 4.70)

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Warning	24.04.2026 14:40:02	Заварочный резервуар: Объем жидкости в норме	Активное	Событие 3	3	Квитировать
Value	24.04.2026 14:39:42	Резервуар розлива выключен	Активное	Событие 2	2	Квитировать
Fault	24.04.2026 14:39:40	Заварочный резервуар: датчик уровня исправен	Активное	Событие 1	1	Квитировать
Warning	24.04.2026 14:39:15	Резервуар розлива: Температура превысила значение в 220 °C	Активное	Предупреждение 3	3	Квитировать
Fault	24.04.2026 14:39:01	Заварочный резервуар: Датчик мутности неисправен	Активное	Тревога 2	6	Квитировать
Fault	24.04.2026 14:38:52	Test	Не активное	Авария	10	Квитировать
Repair	24.04.2026 14:24:42	CO2: Датчик давления в ремонте	Не активное	Событие 3	3	Квитировать
Danger	22.04.2026 20:48:08	Резервуар розлива переполнен	Не активное	Тревога 1	8	Квитировать
Fault	01.04.2026 10:52:13	Test	Не активное	Авария	10	Квитировать
Repair	23.03.2026 14:38:48	CO2: Датчик давления в ремонте	Не активное	Событие 3	3	Квитировать
Danger	23.03.2026 14:30:42	Резервуар розлива переполнен	Не активное	Тревога 1	8	Квитировать
Fault	23.03.2026 14:30:29	Заварочный резервуар: датчик уровня исправен	Не активное	Событие 1	1	Квитировать
Fault	23.03.2026 14:30:29	Заварочный резервуар: Датчик мутности неисправен	Не активное	Тревога 2	6	Квитировать
Warning	23.03.2026 14:30:29	Резервуар розлива: Температура превысила значение в 220 °C	Не активное	Предупреждение 3	3	Квитировать
Warning	23.03.2026 14:30:29	Заварочный резервуар: Объем жидкости в норме	Не активное	Событие 3	3	Квитировать

Рис.4.70. Фильтрация событий по объектам

Для того чтобы перегруппировать события по их состоянию, приоритету, действию (например, отобразить первыми в списке событий - события по их приоритету) следует воспользоваться соответствующей кнопкой напротив каждого из параметров (рис. 4.71).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Repair	24.04.2026 14:41:22	CO2: Датч	Не активное	Событие 3	3	Квитировать
Repair	23.03.2026 14:38:48	CO2: Датч	Не активное	Событие 3	3	Квитировать
Warning	23.03.2026 14:30:29	Резервуар	Не активное	Предупреждение 3	3	Квитировать
Warning	23.03.2026 14:30:29	Заварочн	Не активное	Событие 3	3	Квитировать
Value	23.03.2026 14:30:28	Заварочн	Активное	Событие 2	2	Квитировать
Value	24.04.2026 14:46:22	Резервуар	Не активное	Событие 2	2	Квитировать
Value	23.03.2026 14:30:28	Резервуар	Не активное	Событие 2	2	Квитировать

Рис. 4.71. Перегруппировка (сортировка) событий

Либо воспользоваться кнопкой сортировки на нижней панели настроек, выбрав порядок сортировки каждого параметра (рис. 4.72).

Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
Repair	24.04.2026 14:41:22	CO2: Датч	Не активное	Событие 3	3	Квитировать
Repair	23.03.2026 14:38:48	CO2: Датч	Не активное	Событие 3	3	Квитировать
Warning	23.03.2026 14:30:29	Резервуар	Не активное	Предупреждение 3	3	Квитировать
Warning	23.03.2026 14:30:29	Заварочн	Не активное	Событие 3	3	Квитировать
Value	23.03.2026 14:30:28	Заварочн	Активное	Событие 2	2	Квитировать
Value	24.04.2026 14:46:22	Резервуар	Не активное	Событие 2	2	Квитировать
Value	23.03.2026 14:30:28	Резервуар	Не активное	Событие 2	2	Квитировать

Рис. 4. 72. Сортировка событий

Для того чтобы отобразить только актуальные/архивные события или отфильтровать события по их группам необходимо воспользоваться элементами управления списком в нижнем левом углу страницы (рис. 4.72).

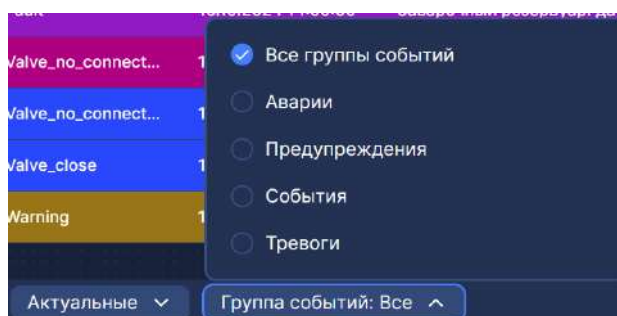


Рис. 4.72. Выбор класса отображаемых событий

Для редактирования количества событий на странице нужно нажать на кнопку в крайнем правом нижнем углу страницы и выбрать необходимое количество отображаемых событий (рис. 4.73).

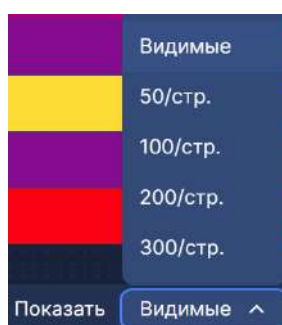


Рис. 4.73. Выбор количества событий на страницу

Для перехода между страницами событий необходимо выбрать номер соответствующей страницы в правом нижнем углу либо воспользоваться кнопкой стрелки (при переходе на вторую и последующую страницу отобразится кнопка стрелки в обратном направлении, рис. 4.74).



Рис. 4.74. Переход между страницами

4.7. Тревоги

В платформе может быть определено событие типа «Тревога» (разд. 2.4.1), для чего необходимо создать соответствующий класс события.

Для этого нужно перейти в «Классы событий» нажатием на соответствующую кнопку в боковом меню, после чего нажать на кнопку «+» в правом верхнем углу страницы, затем задать параметры и выбрать в пункте «Группы» - «Тревоги». По окончании нажать кнопку «Создать» для сохранения класса событий (рис. 4.75).

Классы событий

Поиск

Имя класса	Основной	Квотируемый	Можно квитиовать	Звук	Приоритет
Авария	Цвет события	Цвет события	Да	Нет	10
Авария 1	Цвет события	Цвет события	Да	Нет	1
Авария 2	Цвет события	Цвет события	Да	Нет	2
Авария 3	Цвет события	Цвет события	Нет	Нет	3
Авария С...	Цвет события	Цвет события	Да	Нет	10
Авария С...	Цвет события	Цвет события	Да	Нет	10
Авария С...	Цвет события	Цвет события	Да	Нет	10
Атест			Нет	Нет	1
Класс соб...	Цвет события	Цвет события	Да	Нет	4
Класс соб...	Цвет события	Цвет события	Нет	Нет	2
Класс соб...	Цвет события	Цвет события	Да	Нет	5
Класс соб...	Цвет события	Цвет события	Нет	Нет	1
Предупр.	Цвет события	Цвет события	Да	Нет	1
Пред.	Цвет события	Цвет события	Да	Нет	2
Предупр.	Цвет события	Цвет события	Да	Нет	8

Имя класса

Текст #ffffff Фон #ffffff

Звук Выберите значение

Приоритет Выберите значение

Можно квитиовать Выкл

Группы

☐ Аварии

☐ Предупреждения

☒ Тревоги

+ Добавить группу

Отмена Создать

Рис. 4.75. Создание класса событий

Также для событий типа «Тревоги» возможно применение квитирования для этого в параметрах необходимо включить кнопку «Можно подтвердить» выбрав параметры выводимого текста (рис.4.76).

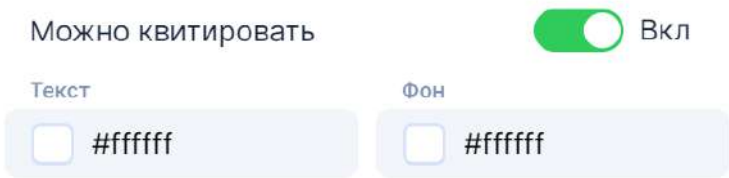


Рис.4.76. Кнопка включения квитирования

4.7.1. Таблица

Для того чтобы отобразить в таблице группу событий «Тревоги» нужно нажать на кнопку «Группы событий» в левом нижнем углу страницы и выбрать «Тревоги» (рис. 4.77).

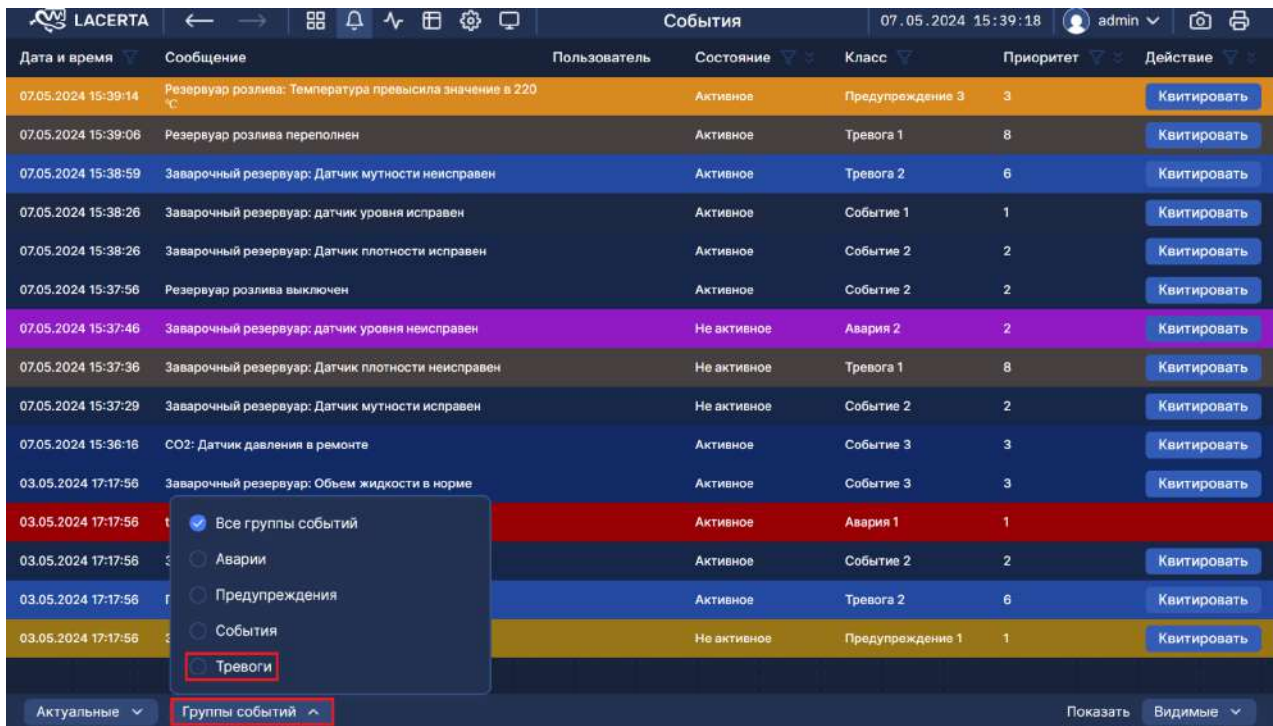
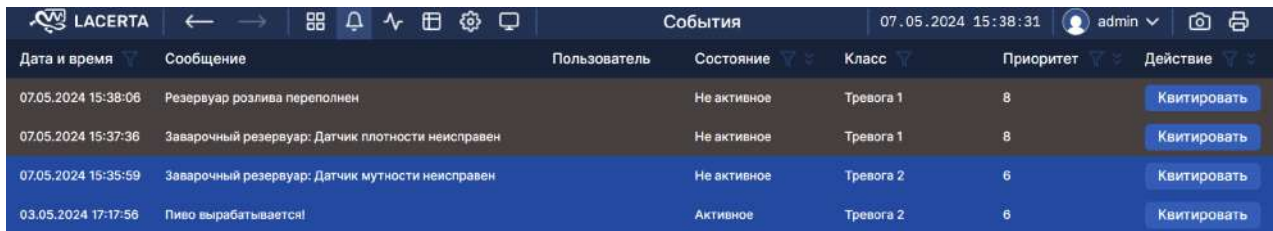


Рис. 4.77. Тревоги в таблице

После чего в таблице отобразятся только события типа «Тревоги» (рис. 4.78).



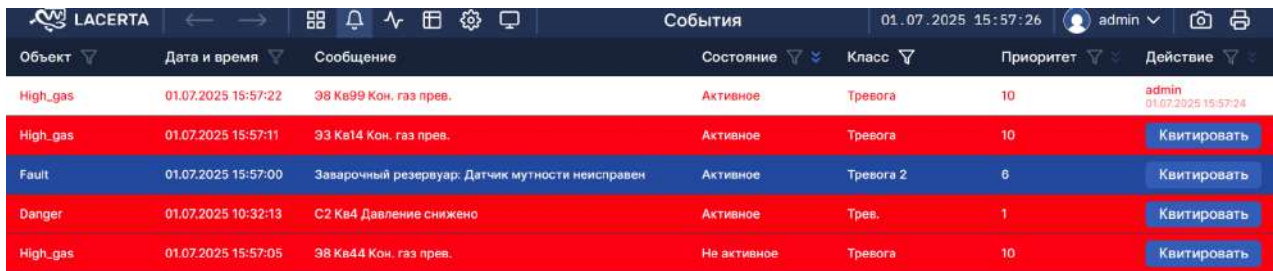
Дата и время	Сообщение	Пользователь	Состояние	Класс	Приоритет	Действие
07.05.2024 15:38:06	Резервуар розлива переполнен		Не активное	Тревога 1	8	Квитировать
07.05.2024 15:37:36	Заварочный резервуар: Датчик плотности неисправен		Не активное	Тревога 1	8	Квитировать
07.05.2024 15:35:59	Заварочный резервуар: Датчик мутности неисправен		Не активное	Тревога 2	6	Квитировать
03.05.2024 17:17:56	Пиво вырабатывается!		Активное	Тревога 2	6	Квитировать

Рис. 4.78. Отображение событий типа «Тревоги»

4.7.2. Квитирование

Для событий типа «Тревоги» возможно применение квитирования - подтверждения пользователем получения информации.

Для подтверждения получения информации о событии (при условии включенной кнопки «Можно подтвердить» в параметрах, в ином случае событий будет квитировано Системой автоматически), нужно нажать на кнопку «Квитировать» в соответствующей строке события. После нажатия напротив события отобразится имя пользователя и дата/время квитирования (рис. 4.79).



Объект	Дата и время	Сообщение	Состояние	Класс	Приоритет	Действие
High_gas	01.07.2025 15:57:22	Э8 Кв99 Кон. газ прев.	Активное	Тревога	10	admin 01.07.2025 15:57:26
High_gas	01.07.2025 15:57:11	Э3 Кв14 Кон. газ прев.	Активное	Тревога	10	Квитировать
Fault	01.07.2025 15:57:00	Заварочный резервуар: Датчик мутности неисправен	Активное	Тревога 2	6	Квитировать
Danger	01.07.2025 10:32:13	С2 Кв4 Давление снижено	Активное	Тревог.	1	Квитировать
High_gas	01.07.2025 15:57:05	Э8 Кв44 Кон. газ прев.	Не активное	Тревога	10	Квитировать

Рис.4.79. Тревоги, квитирование

Приложение 1. Термины и определения

Термин	Определение
Виджет	вспомогательное графическое приложение, предназначенное для выполнения определённых задач
Мультитенантность	элемент архитектуры программного обеспечения, где единый экземпляр приложения обслуживает множество клиентов

Приложение 2. Принятые сокращения

Сокращение	Значение
АДА	архивные данные
АСУ	автоматизированная система управления
ИСП	интегрированная среда разработки
МДА	метаданные
ОДА	оперативные данные
ППИ	прикладной программный интерфейс
РВ	реальное время
СРИ	среда исполнения
СУБД	система управления базой (базами) данных
ЦОД	центр обработки данных
ICMP	ICMP (Internet Control Message Protocol) - протокол третьего уровня модели OSI, который используется для диагностики проблем со связностью в сети, который помогает определить сможет ли достичь пакет адреса назначения в установленные временные рамки.
IEC 61850	Основной протокол передачи данных в системах автоматизации электроподстанций (устройства релейной защиты, анализаторы качества электроэнергии и другие устройства.
Modbus TCP	протокол Modbus RTU с интерфейсом TCP, который работает в сети Ethernet.
MQTT	MQTT (Message queuing telemetry transport) - упрощённый сетевой протокол, работающий поверх TCP/IP, ориентированный на обмен сообщениями между устройствами по принципу издатель-подписчик.
NTP	NTP (Network Time Protocol) – сетевой протокол для синхронизации системного времени компьютера через сетевое соединение.
OPC UA	OPC UA (Unified Architecture) – современный стандарт, описывающий передачу данных в промышленных сетях. Он обеспечивает защищенную и надежную коммуникацию между устройствами, являясь при этом аппаратно- и

Сокращение	Значение
	платформо-независимым, что позволяет обеспечить обмен данными между устройствами с разными операционными системами.
REST	REST (Representational State Transfer) – способ создания API с помощью протокола HTTP. На русском его называют «передачей состояния представления»
S7comm	Реализация протокола Profiinet для серии контроллеров S7 300-1500
SNMP	SNMP (Simple Network Management Protocol) – протокол прикладного уровня, он делает возможным обмен данными между сетевыми устройствами.
SVG	Scalable Vector Graphic (масштабируемая векторная графика)