

SCADA System

Руководство пользователя

Версия 1.0 — Апрель 2026

Содержание

Предисловие

Быстрый старт за 15 минут

1. Авторизация и регистрация
2. Главная панель (Dashboard)
3. Мнемосхемы (Schemas)
4. Редактор мнемосхем (Schema Builder)
5. Просмотр мнемосхем (Schema Viewer)
6. Устройства (Devices)
7. ПЛК (PLCs)
8. Последовательности (Sequences)
9. Рецепты и переменные
10. События и аварии (Events & Accidents)
11. Опции (Options)
12. Импульсные модули (Pulse Modules)
13. Управление партиями (Batch Management)
14. Производственные линии
15. Тренды (Trends)
16. Аварийные сигналы (Alarms)
17. Математические функции (Math Functions)
18. Точки ввода-вывода (I/O Points)
19. Провизионирование I/O
20. Графические статусы
21. Конфигуратор оборудования (Hardware Configurator)
22. Управление пользователями
23. Управление ролями
24. Журнал активности
25. Настройки (Settings)
26. AI-ассистент
27. Экспорт PLC-проекта
28. Экспорт и импорт проекта
29. Танки (Tanks)

Навигация приложения (Sidebar)

Глобальные элементы интерфейса

Горячие клавиши

Практические примеры

Устранение неполадок и FAQ

Безопасность и лучшие практики

Глоссарий

Приложение A: Справочник WebSocket-каналов

Приложение В: Справочник API (сводная таблица)

Приложение С: Коды ошибок и аварий

Версия документа: 1.0

Дата: Апрель 2026

Язык интерфейса: Мультиязычный (EN, RU, UK, DE, FR, PL, AR и др.)

Предисловие

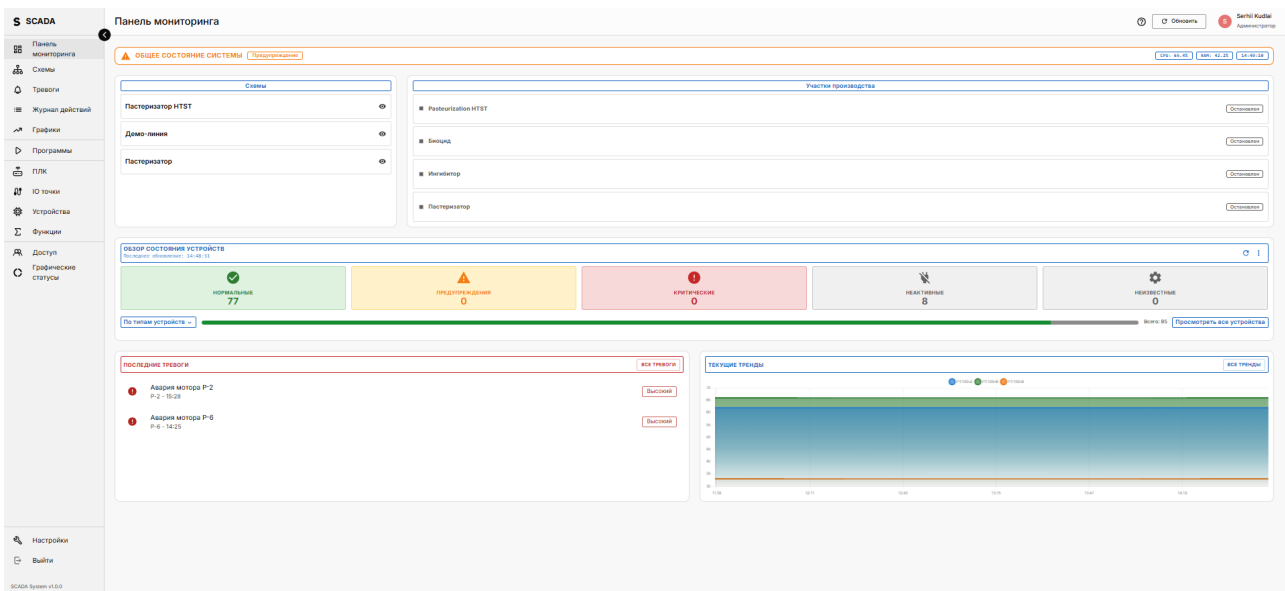
О системе

SCADA System — это современная платформа диспетчерского управления и сбора данных (Supervisory Control and Data Acquisition), разработанная с нуля для нужд промышленной автоматизации нового поколения. Система предназначена для полного цикла управления технологическими процессами — от конфигурирования оборудования и создания мнемосхем до оперативного управления производственными линиями в реальном времени.

В отличие от традиционных SCADA-систем, зародившихся в эпоху десктопных приложений и проприетарных протоколов, данная платформа построена на современном веб-стеке и принципах микросервисной архитектуры. Это означает, что оператор может работать с системой из любого браузера — на промышленном ПК в цеху, на планшете в лаборатории или на ноутбуке из офиса — без установки дополнительного программного обеспечения.

Система охватывает весь жизненный цикл автоматизированного производства:

- Проектирование — визуальный конструктор мнемосхем, конфигурирование устройств, создание алгоритмов управления (последовательностей) и рецептов
- Развёртывание — автоматическая генерация PLC-кода (Siemens SCL/XML), провизионирование точек ввода-вывода, деплой через TIA Portal Agent
- Эксплуатация — мониторинг процесса в реальном времени, управление оборудованием, трендовый анализ, управление партиями
- Обслуживание — журнал активности, аварийные сигналы, диагностика, автоматические обновления, резервное копирование



Коллаж из 4 основных экранов системы: Dashboard (верхний левый), Schema Viewer с работающей схемой (верхний правый), Trends с графиками (нижний левый), Sequence Editor (нижний правый). Демонстрирует широту функциональности.

Для кого предназначена система

| Роль | Как использует систему |
|------------------------------|--|
| Инженер-проектировщик АСУ ТП | Создаёт мнемосхемы, настраивает устройства и последовательности, конфигурирует PLC, генерирует код для TIA Portal |
| Инженер-технолог | Разрабатывает производственные рецепты, настраивает параметры процесса, определяет аварийные пороги |
| Оператор производственной | Наблюдает за мнемосхемой в реальном времени, управляет оборудованием (пуск/стоп, уставки), работает с партиями и рецептами |
| Начальник смены / Мастер | Контролирует состояние всех линий через Dashboard, анализирует тренды, работает с журналом аварий |
| Системный администратор | Управляет пользователями и ролями, обновляет систему, настраивает подключения к PLC и сети |
| Руководство предприятия | Просматривает отчёты, тренды эффективности, статусы производственных линий из любого места через браузер |

Основные преимущества

1. Полностью веб-ориентированная платформа

Система работает в любом современном браузере без установки клиентского ПО. Это принципиально отличает её от классических SCADA-систем (WinCC, Citect, InTouch), которые требуют установки толстого клиента на каждую рабочую станцию, привязаны к конкретной ОС (как правило, Windows) и нуждаются в покупке отдельных лицензий для каждого рабочего места.

- Доступ из любой точки сети (интранет или через VPN)
- Работа на любом устройстве: ПК, планшет, промышленный терминал
- Нет привязки к операционной системе
- Мгновенное обновление интерфейса для всех пользователей одновременно

2. Обновление данных в реальном времени

Двухуровневая архитектура WebSocket обеспечивает мгновенную доставку изменений всем подключённым клиентам:

- Rust WebSocket сервер (порт 8001) — высокопроизводительная доставка массовых обновлений устройств (до тысяч параметров в секунду). Написан на Rust для минимальной задержки и максимальной пропускной способности.
- Django Channels WebSocket — двунаправленное взаимодействие для команд управления (пуск/стоп мотора, открытие/закрытие клапана, изменение уставки PID).

Дельта-обновления передают только изменившиеся поля, а пакетная отправка (batching каждые 250 мс) минимизирует сетевую нагрузку при обновлении сотен параметров одновременно.

3. Интегрированная генерация PLC-кода

Уникальная функция, не имеющая аналогов в большинстве SCADA-систем: из конфигурации устройств, последовательностей, рецептов и аварий система автоматически генерирует полноценный проект для Siemens TIA Portal:

- 11 функциональных блоков (SCL) для каждого типа устройства (Motor FB, Valve FB, PID FB, AI/AO/DI/DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB — автомат состояний для каждой последовательности
- Event / Accident FBs — обработчики событий и аварий
- Data Blocks — экземплярные БД устройств и рецептов
- OB1 Main — главная программа с вызовом всех FB в правильном порядке
- XML Tag Tables — таблицы тегов в формате SimaticML для прямого импорта в TIA Portal

Это сокращает время развёртывания новой производственной линии с недель до часов: инженер конфигурирует процесс в SCADA, нажимает «Генерировать», загружает ZIP в TIA Portal — и PLC готов к работе.

4. Визуальный конструктор мнемосхем

Мощный редактор мнемосхем на базе Konva.js (HTML5 Canvas) заменяет необходимость использования отдельных SCADA-графических редакторов:

- Drag-and-drop размещение устройств, трубопроводов, текста, изображений
- Автоматическая визуализация состояния устройств (цвет, анимация потока)
- Поддержка фоновых изображений (подложка P&ID чертежа)
- Конструктор танков с выбором формы крышек и автоматическим размещением сенсоров
- Многоязычные подписи и переводы
- Неограниченное масштабирование и панорамирование
- Множественное выделение и групповые операции

5. Гибкая система рецептов и последовательностей

Система рецептов реализует стандарт ISA-88 (Batch Control), позволяя разделять:

- Последовательность (Sequence) — алгоритм: порядок шагов и условия переходов
- Рецепт шагов (Recipe) — конкретные значения устройств для каждого шага
- Производственный рецепт (Production Recipe) — набор параметров для конкретного продукта

Одна последовательность может использоваться с разными рецептами для производства разных продуктов на одном оборудовании. Переменные рецепта позволяют параметризовать процесс без изменения алгоритма.

6. Управление партиями (Batch Management)

Полноценная система управления производственными партиями:

- Создание партий с привязкой к рецепту и производственной линии
- Отслеживание прохождения партии по этапам линии
- Очередь партий с приоритетами
- Автоматический трансфер между этапами
- История прохождения каждой партии

7. AI-ассистент

Встроенный AI-помощник на базе LLM (через OpenRouter) значительно ускоряет работу с системой:

- Управление через естественный язык: «Включи мотор MOT_001», «Какие устройства в аварии?»
- Автоматический анализ P&ID чертежей: загрузка изображения → распознавание устройств и связей → предложение размещения на мнемосхеме
- Контекстная помощь и объяснения
- Агентный цикл с инструментами (до 15 итераций) для сложных задач

8. Мультиязычность

Полная поддержка 10+ языков интерфейса с возможностью перевода не только UI, но и пользовательского контента:

- Интерфейс: EN, UK, RU, DE, FR, PL, AR (с RTL) и другие
- Имена устройств, последовательностей, шагов, рецептов, опций — с полем name_translations
- Тексты на мнемосхемах — с полем translations

- Названия мнемосхем — с полем name_translations
- Переключение языка «на лету» без перезагрузки

9. Масштабируемая ролевая модель доступа

Гранулярная система разрешений позволяет точно настроить, кто и что может делать:

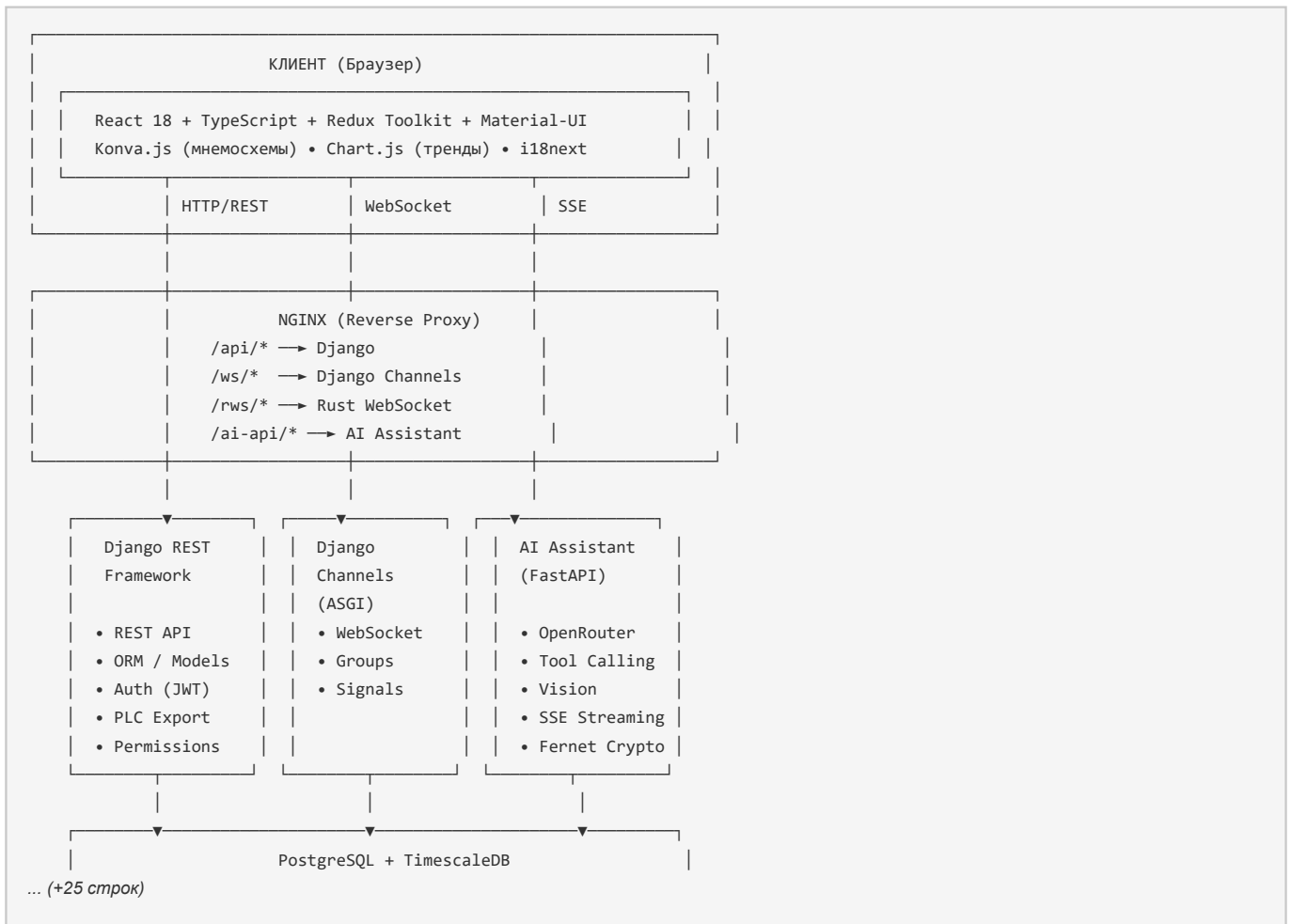
- Роли с матрицей разрешений по 12 категориям и множеству действий
- Управление доступом к отдельным мнемосхемам (per-schema access)
- Журнал всех действий пользователей (аудит)
- IP-фильтрация доступа

10. Промышленная надёжность

- Автоматическое переподключение WebSocket с экспоненциальной задержкой
- Dual WebSocket: Rust для скорости + Django для управления (fallback)
- Полное резервное копирование и восстановление БД
- Экспорт/импорт проектов для переноса между установками
- Система обновлений с rollback и dry-run
- Smoke-тесты после обновлений

Архитектура системы

Система построена на микросервисной архитектуре с чётким разделением ответственности:



Описание компонентов

Frontend (React + TypeScript)

Клиентское приложение, работающее в браузере пользователя. Построено на React 18 с использованием TypeScript для типобезопасности, Redux Toolkit для управления состоянием, Material-UI для компонентов интерфейса. Мнемосхемы отрисовываются через Konva.js (HTML5 Canvas), тренды — через Chart.js. Мультиязычность реализована через i18next с поддержкой 10+ языков, включая RTL (арабский). Взаимодействует с бэкендом через REST API (axios), WebSocket (нативный API) и SSE (для AI-стриминга).

NGINX (Reverse Proxy)

Точка входа для всех запросов. Маршрутизирует трафик между микросервисами:

- /api/* → Django REST API
- /ws/* → Django Channels (WebSocket)
- /rws/* → Rust WebSocket сервер (высокопроизводительный read-only)
- /ai-api/* → AI Assistant (FastAPI)
- /media/* → Статические файлы (иконки, изображения)

Django REST Framework (Backend API)

Ядро бизнес-логики:

- Все CRUD-операции для 40+ типов ресурсов
- Аутентификация и авторизация (JWT + RBAC)
- Управление пользователями, ролями и разрешениями
- Генерация PLC-кода (11 типов функциональных блоков, автомат состояний, Data Blocks, Tag Tables)
- Экспорт/импорт проектов
- Системное обслуживание (обновления, бэкапы, диагностика)

Django Channels (ASGI WebSocket)

Двунаправленное взаимодействие в реальном времени:

- 17+ типов WebSocket-каналов (устройства, схемы, последовательности, партии, аварии, мат. функции, активность, система)
- Redis Channel Layer для group broadcasting
- Обработка команд управления (пуск/стоп устройств, управление партиями)
- Сигналы обновления при изменении моделей Django

AI Assistant (FastAPI)

Автономный микросервис AI-помощника:

- FastAPI на Python 3.11 (порт 8080)
- Интеграция с OpenRouter API (LLM с поддержкой tool calling)
- Агентный цикл-оркестратор (до 15 итераций) с 4 категориями инструментов
- Анализ P&ID изображений через vision-модель
- SSE-стриминг ответов (token, toolcall, toolresult, done)
- Шифрование API-ключей (Fernet/AES)
- JWT-аутентификация (проксируется через Django /api/users/me/)
- Собственные таблицы в общей PostgreSQL БД (префикс ai_assistant_)
- Alembic миграции

PostgreSQL + TimescaleDB

Единая база данных для всех сервисов:

- Основные таблицы (core_*) — устройства, схемы, последовательности, рецепты, роли, пользователи

- Гипертаблицы TimescaleDB — хранение исторических данных (тренды) с автоматическим партиционированием по времени и сжатием
- Таблицы AI-ассистента (ai_assistant_*) — сессии, сообщения, настройки

TimescaleDB обеспечивает производительность запросов к временным рядам на порядки выше стандартного PostgreSQL при больших объёмах данных.

Redis

Коммуникационная шина:

- Channel Layer для Django Channels (группы WebSocket)
- Pub/Sub для уведомлений между Django и Rust Worker
- Кэширование частых запросов

Rust Worker (scada-worker)

Высокопроизводительный сервис обработки в реальном времени, написанный на Rust:

- Sequence Engine (engine.rs) — исполнение последовательностей: отслеживание текущего шага, проверка условий перехода, активация/деактивация устройств, работа с рецептами
- Events Processor — обработка событий последовательностей
- Accident Processor — обнаружение аварийных ситуаций, создание записей аварий
- Tank Volume Calculator — периодический расчёт объёмов танков (каждые 1000 мс) по данным сенсоров уровня и давления
- Math Function Calculator — пересчёт математических функций
- DB Batcher — пакетная запись изменений в PostgreSQL (оптимизация I/O)
- WebSocket Batcher — пакетная отправка обновлений через WebSocket (интервал 250 мс)
- Rust WebSocket Server (порт 8001) — выделенный сервер для read-only стриминга статусов устройств. Значительно быстрее Django Channels для массовых обновлений.

Выбор Rust для этого компонента обусловлен требованиями к минимальной задержке, предсказуемому использованию памяти и способности обрабатывать тысячи обновлений в секунду без garbage collection пауз.

Преимущества по сравнению с традиционными SCADA-системами

| Критерий | Традиционные SCADA | Данная система |
|--------------------|---|--|
| Клиент | Толстый клиент (установка на каждый ПК) | Веб-браузер (ничего не нужно устанавливать) |
| ОС | Только Windows | Любая (Windows, Linux, macOS, Android, iOS) |
| Лицензирование | Per-seat (за каждое рабочее место) | Серверная лицензия (неограниченные клиенты) |
| Обновление | Ручное на каждом ПК | Автоматическое, централизованное (все клиенты сразу) |
| Мобильный доступ | Нет или отдельный модуль | Адаптивный интерфейс из коробки |
| Генерация PLC-кода | Нет (ручное программирование) | Автоматическая из конфигурации SCADA |
| AI-помощник | Нет | Встроенный (анализ P&ID, управление, помощь) |
| Мультиязычность | Ограниченная (интерфейс) | Полная (интерфейс + контент + RTL) |
| Реальное время | Проприетарные протоколы | WebSocket + Rust (sub-second latency) |
| Рецепты | Базовая поддержка | ISA-88 (Sequence + Recipe + Production Recipe) |
| Batch Management | Отдельный модуль (доп. лицензия) | Встроенный |
| Трендовый анализ | Встроенный, ограниченный | TimescaleDB (петабайты данных, автосжатие) |
| Аудит | Базовый | Полный журнал с WebSocket-обновлениями |

| Критерий | Традиционные SCADA | Данная система |
|-------------------------|--|---|
| Бэкап/Восстановление | Внешние инструменты | Встроенный (бэкап, экспорт, импорт, rollback) |
| Стоимость развёртывания | Высокая (лицензии + установка + настройка) | Низкая (Docker, автоматизированное развёртывание) |
| Расширяемость | Закрытая экосистема | Открытый REST API + WebSocket + SSE |

Системные требования

Серверная часть

| Компонент | Минимальные | Рекомендуемые |
|-----------------------|--|---------------------------------------|
| Процессор | 4 ядра | 8+ ядер |
| Оперативная память | 8 ГБ | 16+ ГБ |
| Дисковое пространство | 20 ГБ SSD | 100+ ГБ SSD (для трендов) |
| ОС | Linux (Ubuntu 22.04+) или Windows Server 2019+ | Linux (Ubuntu 22.04 LTS) |
| Docker | Docker Engine 24+ | Docker Engine 24+ с Docker Compose v2 |
| PostgreSQL | 15+ с TimescaleDB | 16+ с TimescaleDB 2.x |
| Redis | 7+ | 7+ |

Клиентская часть (браузер)

| Компонент | Требование |
|-------------------|---|
| Браузер | Chrome 90+, Firefox 90+, Edge 90+, Safari 15+ |
| Разрешение экрана | Минимум 1280×720, рекомендуется 1920×1080+ |
| Сеть | Стабильное подключение к серверу (LAN/VPN) |

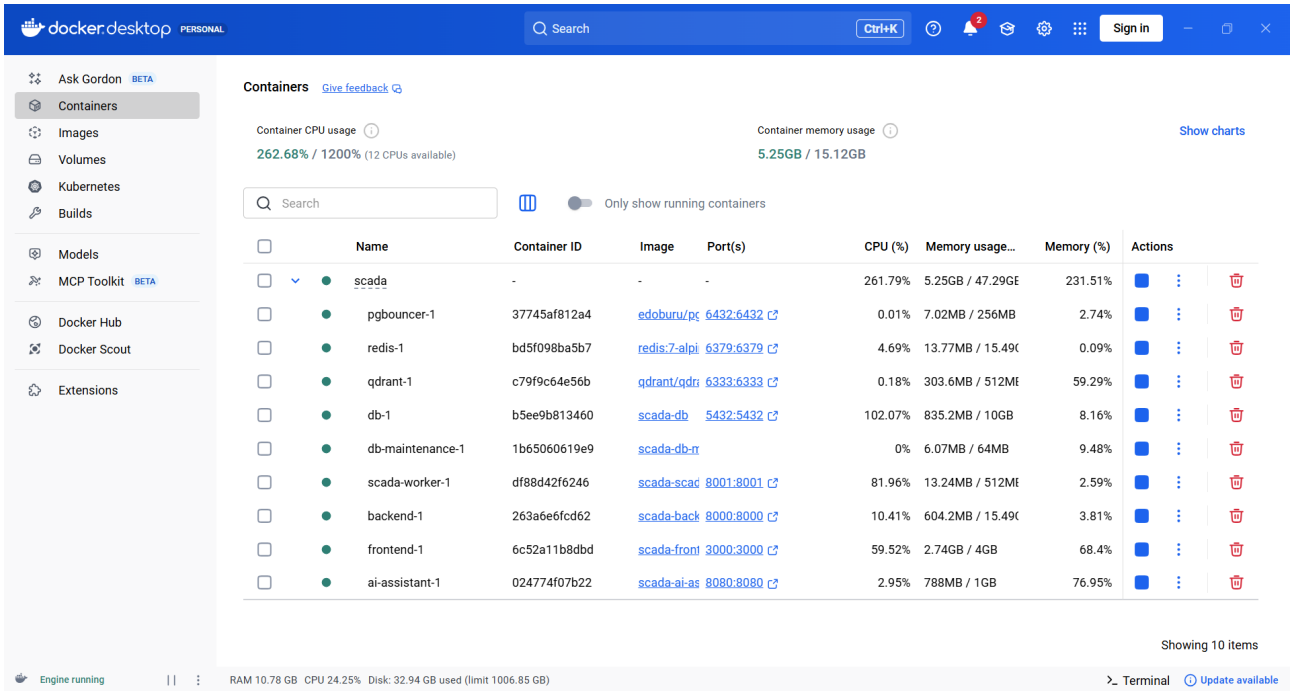
Развёртывание

Система поставляется в виде Docker-контейнеров и разворачивается одной командой:

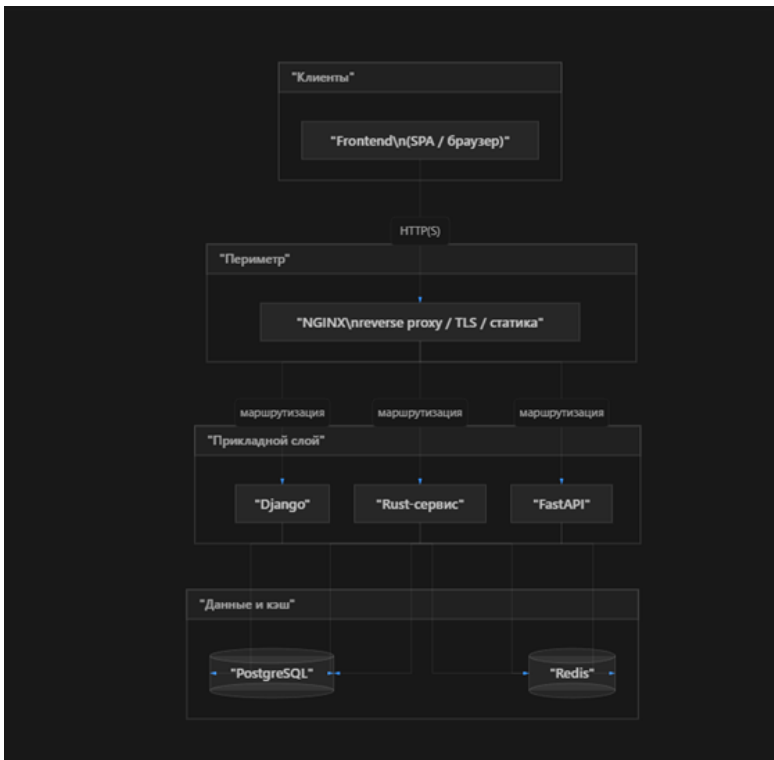
```
docker-compose up -d
```

Состав контейнеров:

- backend — Django REST API + Django Channels
- scada-worker — Rust Worker
- ai-assistant — FastAPI AI сервис
- postgres — PostgreSQL + TimescaleDB
- redis — Redis
- nginx — Reverse Proxy



Терминал с выводом ``docker-compose ps``: все 6 контейнеров в статусе «Up», порты, время работы.



Архитектурная диаграмма системы: блоки компонентов с линиями связи (REST, WebSocket, Redis Pub/Sub, SQL). Цветовая кодировка: синий — Frontend, зелёный — Django, оранжевый — Rust, фиолетовый — AI, серый — инфраструктура.

Быстрый старт за 15 минут

Этот раздел проведёт вас от чистой установки до работающей мнемосхемы с одним мотором и одним клапаном. Если вы впервые запускаете SCADA — начните отсюда.

Шаг 1. Первый вход (2 мин)

- Откройте в браузере адрес установки (например, <http://localhost> или <http://scada.local>).
- На экране входа введите учётные данные администратора, созданные установщиком (по умолчанию admin / пароль, заданный при установке).
- При первом входе система может запросить активацию лицензии — введите ключ и нажмите «Активировать». Без активной лицензии доступ к большинству функций ограничен.
- После успешного входа вы попадёте на Dashboard — главную панель.

Подсказка. Выберите язык интерфейса в левом нижнем углу формы входа — он будет сохранён для вашей учётной записи.

Шаг 2. Создание первого ПЛК (2 мин)

Даже если вы пока работаете только в симуляции, система требует наличия хотя бы одного ПЛК, к которому привязываются устройства.

- В боковом меню откройте PLCs.
- Нажмите «+» в правом верхнем углу.
- Заполните:
 - Name: PLC_DEMO
 - IP Address: 192.168.0.100 (адрес можно поменять позже)
 - Type: Siemens S7-1200 (или любой доступный в списке)
 - Rack / Slot: 0 / 1
- Нажмите «Сохранить». Статус ПЛК будет «Offline» — это нормально для демо.

Шаг 3. Создание первых устройств (3 мин)

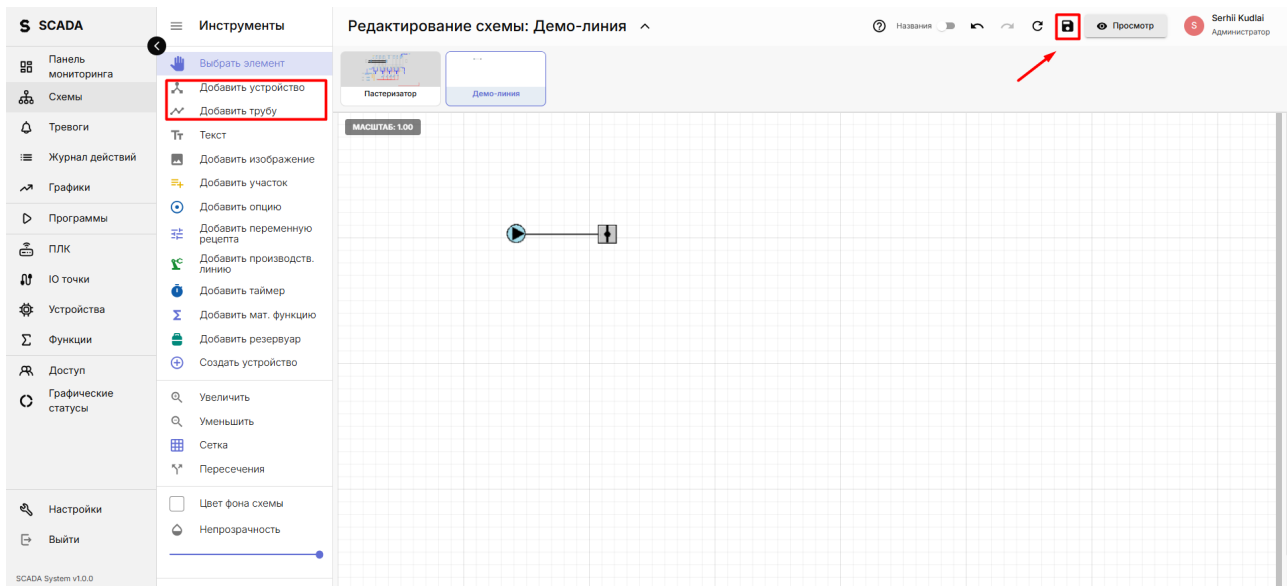
Добавим один мотор и один клапан.

- Перейдите в Devices → «+» → Motor.
- Заполните поля:
 - ID: mot_001
 - Name: Насос подачи
 - PLC: PLC_DEMO
 - Description: «Демонстрационный мотор»
- Нажмите «Сохранить».
- Аналогично создайте клапан («+» → Valve):
 - ID: val_001
 - Name: Клапан линии
 - PLC: PLC_DEMO

Важно. ID устройства — это текстовый идентификатор (mot_001, а не число). Он используется во всех WebSocket-каналах, PLC-экспорте и в условиях переходов. Придерживайтесь единой схемы именования с самого начала.

Шаг 4. Создание первой мнемосхемы (3 мин)

- Откройте Schemas → «+».
- Укажите Name: «Демо-линия» и нажмите «Создать».
- В списке схем нажмите карандаш (Edit) на только что созданной схеме — откроется Schema Builder.
- В левой панели нажмите «Устройство» () и в открывшемся диалоге выберите mot_001 → «Добавить». Мотор появится на холсте — перетащите его мышью.
- Снова «Устройство» → val_001 → «Добавить» — разместите клапан рядом.
- Выберите «Труба» (), кликните на мотор → потом на клапан — между ними появится линия соединения.
- Нажмите Ctrl+S или кнопку «Сохранить» в верхней панели.



Простая демо-схема: один мотор (зелёный кружок) соединён трубой с клапаном (жёлтый треугольник), на заднем плане — сетка холста.

Шаг 5. Запуск просмотра и управление (3 мин)

- В верхней панели редактора нажмите «Просмотр» — вы перейдёте в Schema Viewer.
- Кликните по иконке мотора — откроется роруп управления. Переключите режим на Manual и нажмите Start. Индикатор станет зелёным — мотор «работает».
- Кликните по клапану → Open. Если у трубы включён мониторинг процесса, появится анимация потока.
- Нажмите Stop у мотора — анимация остановится.

Поздравляем — у вас есть первая рабочая мнемосхема с живым управлением.

Шаг 6. Что делать дальше (2 мин)

Теперь у вас есть базовый каркас. Следующие шаги зависят от вашей задачи:

| Задача | Куда идти |
|---|---|
| Автоматизировать процесс (запуск/остановку по условиям) | Раздел 8 «Последовательности» и практический пример «Пастеризатор молока» |

| Задача | Куда идти |
|--|--|
| Настроить рецепты для разных продуктов | Раздел 9 «Рецепты и переменные» |
| Привязать реальные входы/выходы ПЛК | Разделы 18 «I/O Points» и 19 «Провизионирование I/O» |
| Настроить аварийные пороги | Раздел 10 «События и аварии» |
| Сгенерировать проект для TIA Portal | Раздел 27 «Экспорт PLC-проекта» |
| Анализ трендов | Раздел 15 «Тренды» |

Подсказка. На любой странице в шапке есть кнопка «?» — она запускает интерактивный обучающий тур с подсветкой элементов.

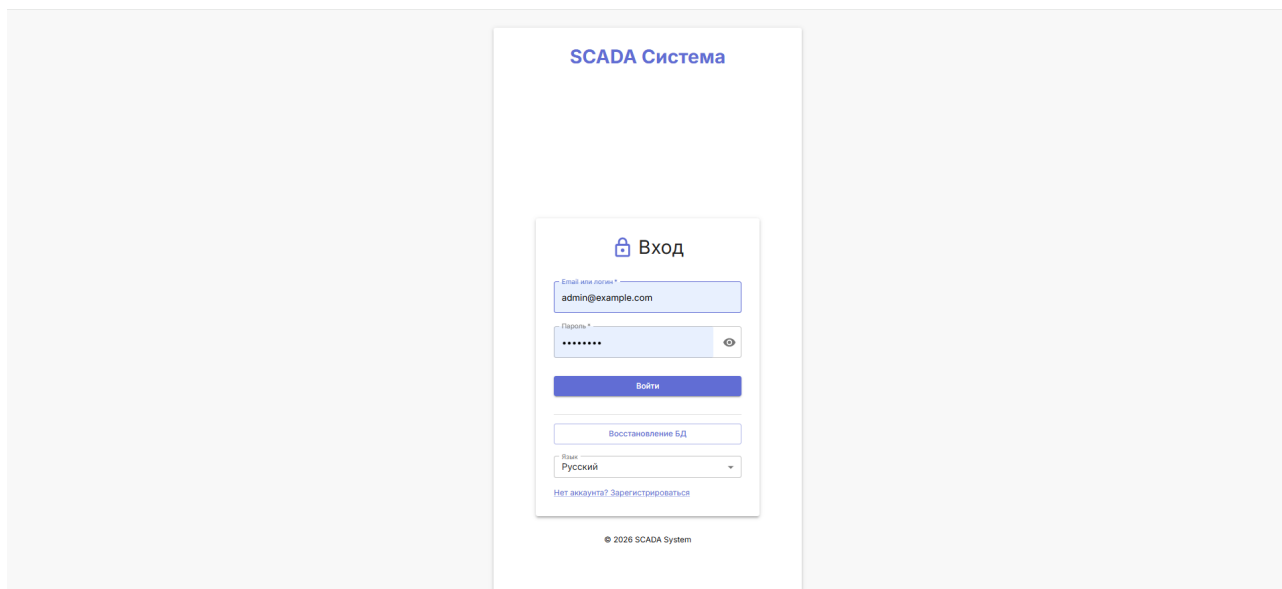
1. Авторизация и регистрация

Для пользователя

Экран входа — первое, что вы видите при открытии системы.

Элементы экрана:

| Элемент | Описание |
|---------------------------------|---|
| Поле «Имя пользователя / Email» | Введите ваш логин или электронную почту |
| Поле «Пароль» | Введите пароль. Иконка справа переключает видимость пароля |
| Выбор языка | Выпадающий список для выбора языка интерфейса (сохраняется после входа) |
| Кнопка «Войти» | Выполняет вход в систему |
| Ссылка «Забыли пароль?» | Переход к восстановлению пароля |
| Ссылка «Регистрация» | Переход к форме регистрации нового аккаунта |



Страница входа: поля «Username/Email» и «Password» (с иконкой глаза), выпадающий список языка, кнопка «Войти», ссылки «Забыли пароль?» и «Регистрация». Тёмная/светлая тема в зависимости от настроек.

Экран регистрации:

| Элемент | Описание |
|-----------------------------|---------------------------------|
| Поле «Имя пользователя» | Уникальное имя для входа |
| Поле «Email» | Электронная почта |
| Поле «Пароль» | Пароль с индикатором надёжности |
| Поле «Подтвердите пароль» | Повторный ввод пароля |
| Поле «Полное имя» | Имя и фамилия |
| Кнопка «Зарегистрироваться» | Создание аккаунта |

Форма регистрации: поля Username, Email, Password (с индикатором надёжности), Confirm Password, Full Name, кнопка «Зарегистрироваться».

Техническая документация

API аутентификации:

| Метод | Endpoint | Описание | Тело запроса |
|-------|--------------------------|-----------------------|---|
| POST | /api/auth/login | Вход в систему | {"username": "...", "password": "..."} |
| POST | /api/auth/register | Регистрация | {"username": "...", "email": "...", "password": "...", "first_name": "...", "last_name": "..."} |
| GET | /api/auth/me | Текущий пользователь | — |
| POST | /api/auth/token/refresh/ | Обновление JWT токена | {"refresh": "..."} |

Ответ при входе:

```
{
  "access": "eyJhbGciOi...",
  "refresh": "eyJhbGciOi...",
  "user": {
    "id": 1,
    "username": "admin",
    "email": "admin@example.com",
    "roles": [...]
  }
}
```

Авторизация: Все защищённые endpoints требуют заголовок Authorization: Bearer .

2. Главная панель (Dashboard)

Для пользователя

Dashboard — центральная страница системы, отображающая общее состояние.

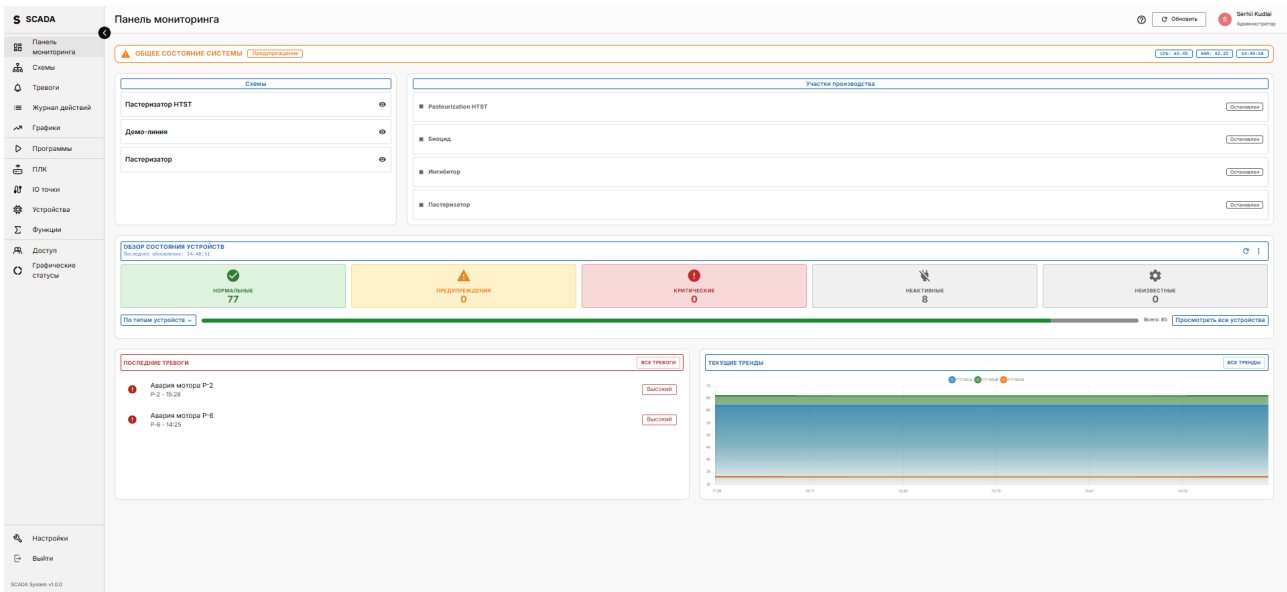
Элементы экрана:

| Элемент | Описание |
|-----------------------------|--|
| Карточки статистики | Краткая сводка: количество устройств, активных схем, аварий, последовательностей |
| Обзор состояния | Визуальные индикаторы состояния устройств и системы |
| Список аварий | Последние аварийные сигналы с цветовой индикацией серьезности |
| Мини-тренды | Миниатюрные графики ключевых параметров |
| Статусы последовательностей | Текущий шаг и прогресс каждой запущенной последовательности |
| Системные метрики | CPU, RAM, Uptime, температура, дисковое пространство |
| Кнопки быстрого доступа | Быстрый переход к основным функциям |

Кнопки в шапке:

| Кнопка | Описание |
|-----------------|--|
| Обновить | Обновить данные на Dashboard |
| Тур | Запуск интерактивного обучающего тура по элементам Dashboard |
| Экспорт проекта | Открывает диалог экспорта всего проекта |
| Импорт проекта | Открывает диалог импорта проекта из файла |

Полный экран Dashboard: карточки статистики сверху, блок аварий слева, мини-тренды справа, статусы последовательностей внизу.
Боковая панель навигации слева.



Блок системных метрик: CPU usage (%), RAM usage (%), Disk usage, Uptime, Network stats. Каждый параметр с цветovým индикатором (зелёный/жёлтый/красный).

Техническая документация

API системных метрик:

| Метод | Endpoint | Описание |
|-------|----------------------------|----------------------------|
| GET | /api/system/info/ | Информация о системе |
| GET | /api/system/health-status/ | Состояние здоровья системы |

WebSocket: Системные метрики (стриминг каждые 2 сек)

Подключение: ws://ws/system

```
{
  "type": "system_metrics",
  "cpu": 45.2,
  "memory": 67.8,
  "disk": 34.1,
  "network": {"rx": 1024, "tx": 512},
  "uptime": 86400,
  "temperature": 52.0,
  "lastUpdate": "2026-04-07T12:00:00Z"
}
```

WebSocket: Статус ПЛК (стриминг каждые 2 сек)

```
{
  "type": "plc_status",
  "is_active": true,
  "status": "running",
  "service_running": true,
  "timestamp": "2026-04-07T12:00:00Z"
}
```

3. Мнемосхемы (Schemas)

Для пользователя

Мнемосхемы — визуальные диаграммы вашего производственного процесса. Это главный инструмент для мониторинга и управления.

Элементы экрана списка:

| Элемент | Описание |
|--------------------|--|
| Список/сетка схем | Карточки или строки таблицы с мнемосхемами |
| Переключатель вида | Сетка / Список (иконки справа сверху) |
| Кнопка «+» | Создание новой мнемосхемы |
| Поиск | Поле поиска по названию |
| Кнопка обновления | Перезагрузка списка |

Действия для каждой схемы:

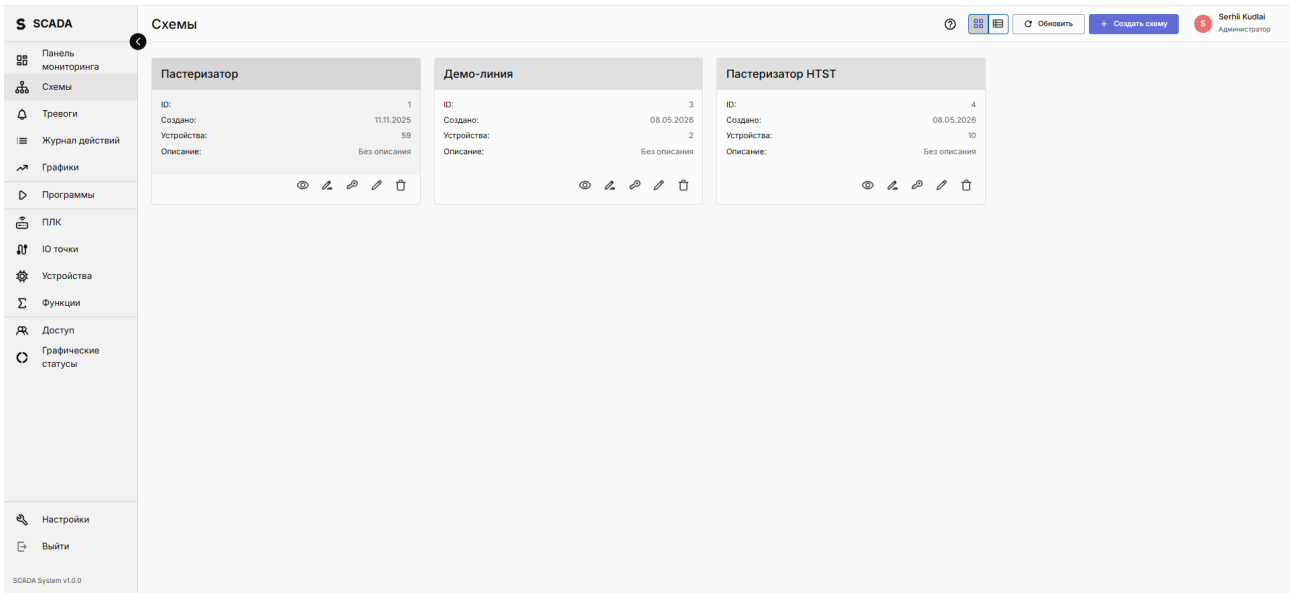
| Кнопка | Иконка | Описание |
|---------------------|--------|---|
| Просмотр | | Открывает схему в режиме просмотра (SchemaViewer) |
| Редактировать | | Открывает схему в редакторе (SchemaBuilder) |
| Удалить | | Удаление схемы (с подтверждением) |
| Управление доступом | | Настройка доступа пользователей к схеме |
| Переименовать | | Изменение названия и переводов названия |
| Настройки | | Настройка параметров схемы (описание, переводы) |

The screenshot displays the 'Схемы' (Schemas) page in the SCADA System. The main content is a table listing schemas with the following data:

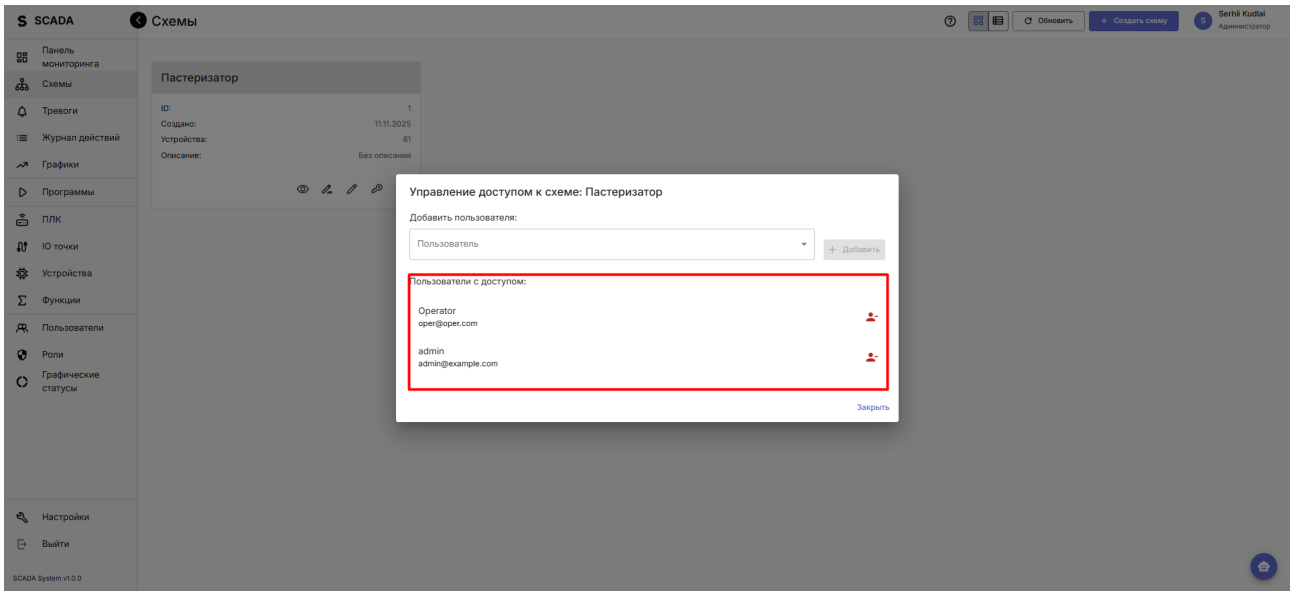
| ID | Имя | Описание | Создано | Устройства | Действия |
|----|-------------------|----------|------------|------------|----------|
| 1 | Пастеризатор | - | 11.11.2025 | 59 | |
| 3 | Демо-линия | - | 08.05.2026 | 2 | |
| 4 | Пастеризатор HTST | - | 08.05.2026 | 10 | |

The interface includes a sidebar with navigation items like 'Панель мониторинга', 'Схемы', 'Тревоги', 'Журнал действий', 'Графики', 'Программы', 'ПЛК', 'Ю точки', 'Устройства', 'Функции', 'Доступ', 'Графические статусы', 'Настройки', and 'Выйти'. The top right corner shows the user 'Serhiy Kudral' as Administrator and a '+ Создать схему' button.

Страница списка мнемосхем в виде сетки карточек. Каждая карточка показывает превью схемы, название, дату изменения, кнопки действий (просмотр, редактирование, удаление). Кнопка «+» в правом верхнем углу.



Та же страница, но в табличном виде: колонки «Название», «Описание», «Дата создания», «Последнее изменение», «Действия».



Диалог управления доступом: список пользователей с чекбоксами, кнопки «Добавить» и «Сохранить».

Техническая документация

API мнемосхем:

| Метод | Endpoint | Описание | Параметры |
|--------|--------------------------------|---------------------------------|---|
| GET | /api/schemas/ | Список всех схем | ?search=...&ordering=... |
| POST | /api/schemas/ | Создание схемы | {"name": "...", "description": "...", "name_translations": {...}} |
| GET | /api/schemas/{id}/ | Получение схемы | — |
| PUT | /api/schemas/{id}/ | Обновление схемы | Полный объект схемы |
| PATCH | /api/schemas/{id}/ | Частичное обновление | {"name": "...", "description": "...", "name_translations": {...}} |
| DELETE | /api/schemas/{id}/ | Удаление схемы | — |
| GET | /api/schemas/{id}/user_access/ | Список пользователей с доступом | — |

| Метод | Endpoint | Описание | Параметры |
|--------|--------------------------------|-----------------|----------------|
| POST | /api/schemas/{id}/user_access/ | Добавить доступ | {"user_id": 1} |
| DELETE | /api/schemas/{id}/user_access/ | Убрать доступ | {"user_id": 1} |

4. Редактор мнемосхем (Schema Builder)

Для пользователя

Редактор мнемосхем — мощный визуальный инструмент для создания и редактирования технологических диаграмм. Работает на холсте (canvas) с функциями перетаскивания, масштабирования и поворота элементов.

Левая панель инструментов

| Кнопка | Иконка | Описание |
|------------------------|----------|--|
| Выделение | | Режим выбора и перемещения элементов |
| Устройство | | Добавить устройство из каталога на схему |
| Труба/Сигнал | | Нарисовать трубопровод (соединение между устройствами) |
| Текст | T | Добавить текстовую аннотацию на схему |
| Изображение | | Добавить фоновое изображение (подложку P&ID) |
| Последовательность | | Привязать последовательность к схеме |
| Опция | | Добавить кнопку опции последовательности |
| Переменная рецепта | | Добавить отображение переменной рецепта |
| Производственная линия | | Добавить производственную линию |
| Мат. функция | <i>f</i> | Добавить математическую функцию |
| Танк | | Открыть конструктор танка (отдельная кнопка) |
| Создать устройство | | Быстрое создание нового устройства |

Верхняя панель инструментов

| Кнопка | Описание |
|-----------------------|---|
| Сохранить | Сохраняет текущее состояние схемы |
| Отменить (Ctrl+Z) | Отмена последнего действия |
| Повторить (Ctrl+Y) | Повтор отменённого действия |
| Просмотр | Переключение в режим SchemaViewer |
| Переключатель «Имена» | Показать/скрыть названия устройств на схеме |
| Название схемы | Отображение текущего имени схемы |

Управление холстом

| Кнопка | Описание |
|-------------------|------------------------------------|
| Увеличить (+) | Увеличение масштаба |
| Уменьшить (-) | Уменьшение масштаба |
| Сетка | Включить/отключить сетку на холсте |
| Цвет фона | Изменить цвет фона схемы |
| Прозрачность фона | Ползунок прозрачности фона |

| Кнопка | Описание |
|----------|----------------------------------|
| Свернуть | Свернуть/развернуть левую панель |

Правая панель свойств

Панель отображается при выборе элемента на схеме и зависит от типа выбранного элемента:

Для устройства (DevicePropertyPanel):

| Свойство | Описание |
|----------------------|-----------------------------------|
| Позиция X, Y | Координаты на холсте |
| Угол поворота | Поворот элемента (0-360°) |
| Размер W, H | Ширина и высота |
| Цвет | Цвет элемента (палитра) |
| Зеркальное отражение | Отразить по горизонтали/вертикали |
| Показывать имя | Вкл/выкл подпись устройства |
| Графический статус | Привязка к шаблону статусов |
| Конфигурация | Устройство-специфичные параметры |

Для трубопровода (PipePropertyPanel):

| Свойство | Описание |
|---------------------|------------------------------|
| Название | Имя трубопровода |
| Цвет | Цвет линии |
| Толщина | Толщина линии (px) |
| Видимость имени | Показывать/скрывать название |
| Мониторинг процесса | Включить анимацию потока |

Для текста (TextPropertyPanel):

| Свойство | Описание |
|-----------------|-------------------------------|
| Текст | Содержимое |
| Размер шрифта | Размер текста |
| Цвет | Цвет текста |
| Жирный / Курсив | Стилизация |
| Переводы | Мультиязычные переводы текста |

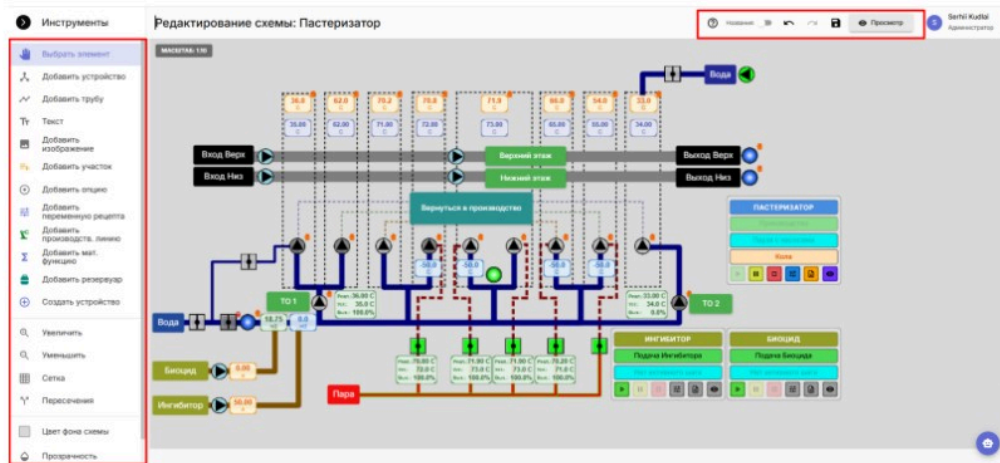
Для последовательности (SequencePropertyPanel):

| Свойство | Описание |
|-------------|---|
| Позиция | Координаты |
| Размер | Ширина × Высота |
| Отображение | Настройки отображения (имя, статус, шаги) |

Диалоги редактора

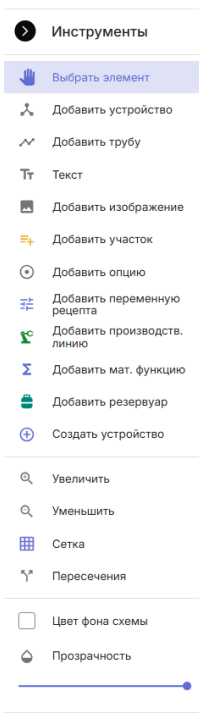
| Диалог | Описание |
|-------------------------|---|
| DeviceDialog | Выбор устройства из каталога для размещения на схеме. Фильтрация по типу. |
| PipeDialog | Настройка имени, типа и видимости трубопровода |
| AddSequenceDialog | Выбор последовательности для привязки к схеме |
| AddOptionDialog | Выбор опции последовательности |
| AddProductionLineDialog | Выбор производственной линии |
| AddRecipeVariableDialog | Выбор переменной рецепта |
| MathFunctionDialog | Выбор математической функции |
| TankConstructorDialog | Конструктор танка: выбор устройства, типы крышек (FLAT/DOME/DISH/CONE), размеры, превью SVG, авто-размещение сенсоров |
| ImageDialog | Загрузка фонового изображения |
| TextDialog | Добавление/редактирование текста |

| | |
|-----------------------|---|
| TankConstructorDialog | Конструктор танка: выбор устройства, типы крышек (FLAT/DOME/DISH/CONE), размеры, превью SVG, авто-размещение сенсоров |
| ImageDialog | Загрузка фонового изображения |
| TextDialog | Добавление/редактирование текста |

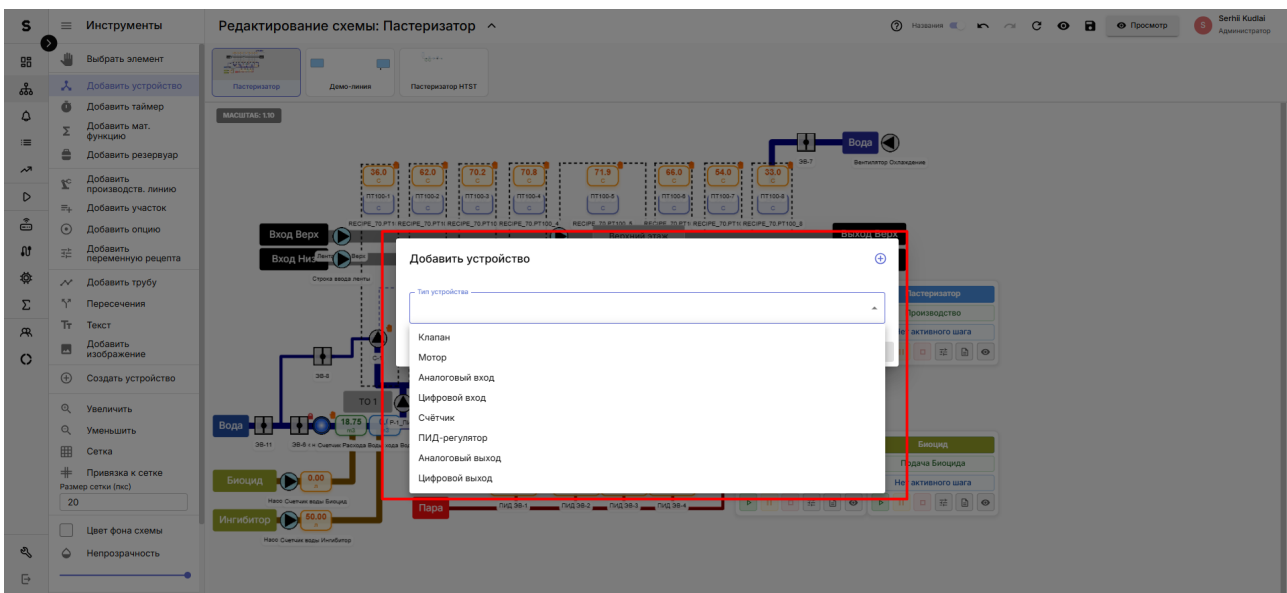


Полный экран редактора: левая панель инструментов с иконками, верхняя панель с кнопками «Сохранить», «Отменить», «Повторить», центральный холст со схемой (устройства, трубы, тексты), правая панель свойств выбранного элемента.

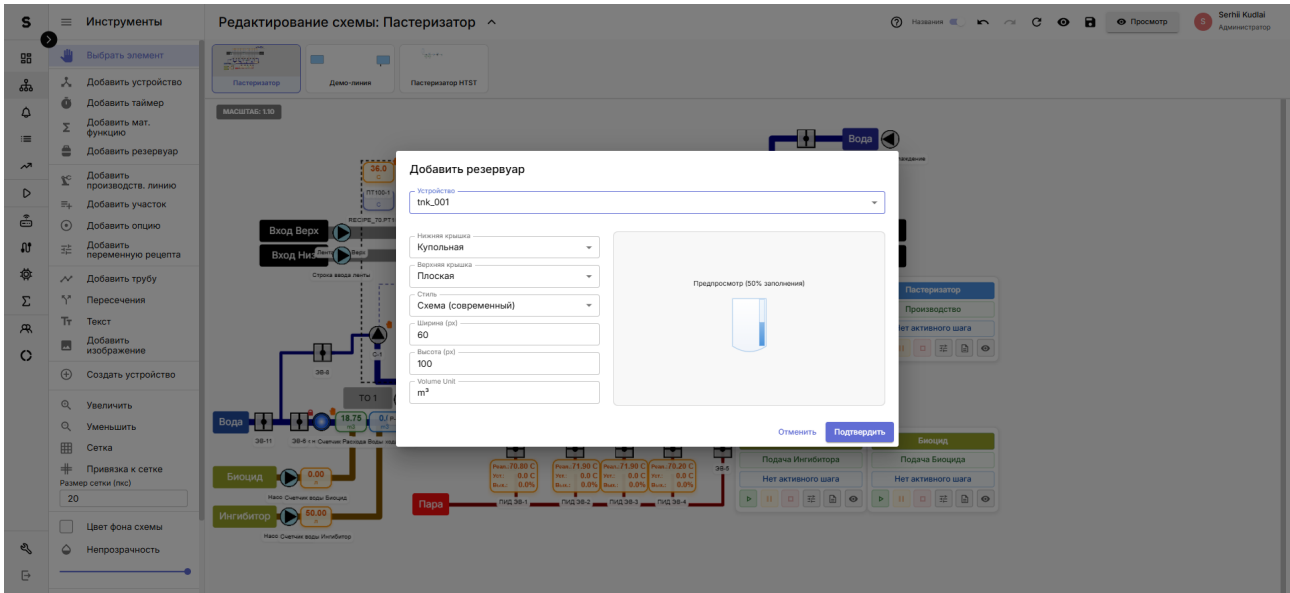
Полный экран редактора: левая панель инструментов с иконками, верхняя панель с кнопками «Сохранить», «Отменить», «Повторить», центральный холст со схемой (устройства, трубы, тексты), правая панель свойств выбранного элемента.



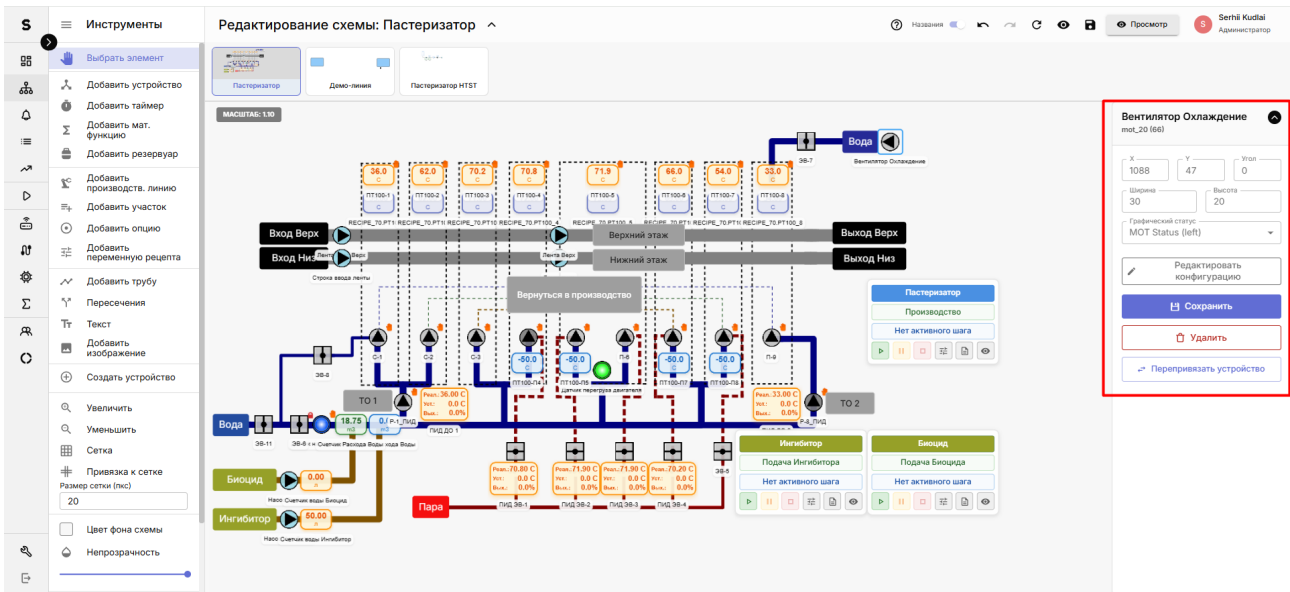
Крупный план левой панели инструментов: каждая кнопка подписана (Выделение, Устройство, Труба, Текст, Изображение, Последовательность, Опция, Переменная рецепта, Производственная линия, Мат. функция, Танк).



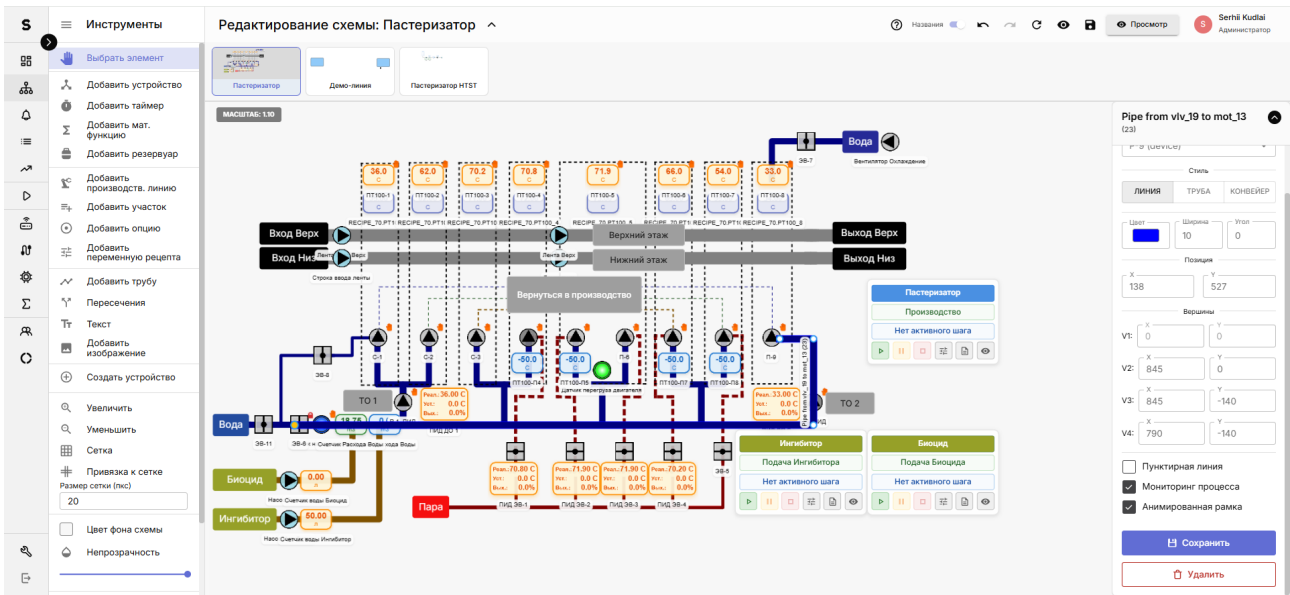
Диалог выбора устройства: список устройств с фильтрацией по типу (Motor, Valve, AI, AO, DI, DO, PID, Counter, Timer, Tank, COS), поле поиска, кнопка «Добавить».



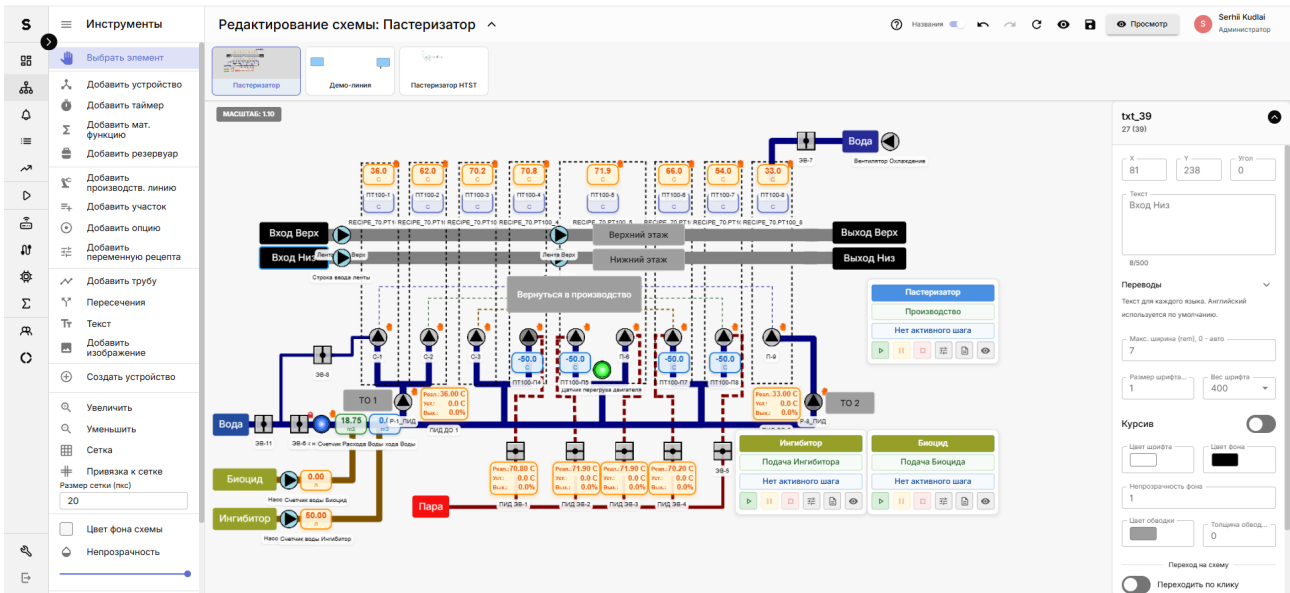
Конструктор танка: выпадающие списки типа верхней и нижней крышки (FLAT, DOME, DISH, CONE), поля ширины/высоты, SVG-превью формы танка, выбор устройства танка.



Правая панель свойств при выборе устройства: поля Position X/Y, Angle, Width/Height, Color picker, переключатели Mirror/Show Name.



Правая панель свойств при выборе трубы: поле Name, Color picker, Thickness slider, Show Name toggle, Process Monitoring toggle.



Правая панель свойств при выборе текста: поле Text, Font Size, Color, Bold/Italic toggles, блок Translations с полями для каждого языка.

Техническая документация

API элементов схемы:

| Метод | Endpoint | Описание |
|-------|---------------------------------------|------------------|
| POST | /api/schemas/{id}/add_motor/ | Добавить мотор |
| POST | /api/schemas/{id}/add_valve/ | Добавить клапан |
| POST | /api/schemas/{id}/add_analog_input/ | Добавить AI |
| POST | /api/schemas/{id}/add_analog_output/ | Добавить АО |
| POST | /api/schemas/{id}/add_digital_input/ | Добавить DI |
| POST | /api/schemas/{id}/add_digital_output/ | Добавить DO |
| POST | /api/schemas/{id}/add_pid_controller/ | Добавить PID |
| POST | /api/schemas/{id}/add_counter/ | Добавить счётчик |

| Метод | Endpoint | Описание |
|-------|---|-----------------------------|
| POST | /api/schemas/{id}/add_variable_speed_drive/ | Добавить ЧП |
| POST | /api/schemas/{id}/add_sequence/ | Добавить последовательность |
| POST | /api/schemas/{id}/add_option/ | Добавить опцию |
| POST | /api/schemas/{id}/add_production_line/ | Добавить линию |
| POST | /api/schemas/{id}/add_recipe_variable/ | Добавить переменную |
| POST | /api/schemas/{id}/add-math-function/ | Добавить мат. функцию |
| POST | /api/schemas/{id}/bulk_add_schema_devices/ | Массовое добавление |
| PATCH | /api/schemas/{id}/update_positions/ | Массовое обновление позиций |
| PATCH | /api/schemas/{id}/devices/{deviceId}/ | Обновить элемент |
| DELET | /api/schemas/{id}/remove_device/ | Удалить элемент |

E

Тело запроса добавления устройства:

```
{
  "device_id": "mot_001",
  "content_type": "motor",
  "position_x": 100,
  "position_y": 200,
  "angle": 0,
  "configuration": {
    "color": "#FF0000",
    "mirror_x": false,
    "mirror_y": false,
    "show_name": true,
    "tank_visual": {
      "bottom_type": "DOME",
      "top_type": "FLAT",
      "width": 60,
      "height": 100
    }
  }
}
```

API трубопроводов:

| Метод | Endpoint | Описание |
|--------|----------------------------|-----------------|
| GET | /api/pipes/?schema_id={id} | Все трубы схемы |
| POST | /api/pipes/ | Создать трубу |
| PATCH | /api/pipes/{id}/ | Обновить трубу |
| DELETE | /api/pipes/{id}/ | Удалить трубу |

5. Просмотр мнемосхем (Schema Viewer)

Для пользователя

Режим просмотра предоставляет живую визуализацию процесса с реальным временем обновления данных.

Возможности:

| Функция | Описание |
|------------------------|---|
| Живой статус устройств | Цвета и анимации обновляются в реальном времени |
| Нажатие на устройство | Открывает рорип с управлением и информацией |
| Масштабирование | Колесо мыши или кнопки «+» / «-» |
| Панорамирование | Зажатие правой кнопки мыши + перемещение |
| Полоса аварий | Сверху/снизу экрана — текущие активные аварии |
| Переключатель имён | Показать/скрыть подписи устройств |

Рорип управления устройством

При нажатии на устройство открывается всплывающее окно с вкладками:

Вкладка «Управление» (Control):

Для каждого типа устройства:

| Тип | Элементы управления |
|-----------------------|--|
| Мотор (Motor) | Кнопки Пуск/Стоп, переключатель Авто/Ручной, управление частотой |
| Клапан (Valve) | Кнопки Открыть/Закрыть, Авто/Ручной, %, расход |
| Аналоговый вход (AI) | Авто/Ручной, ввод значения, индикатор шкалы |
| Аналоговый выход (AO) | Авто/Ручной, ввод значения, Min/Max |
| Дискретный вход (DI) | Авто/Ручной, индикатор ON/OFF, блокировка |
| Дискретный выход (DO) | Авто/Ручной, кнопки ON/OFF, блокировка |
| PID-регулятор | Уставка, Авто/Ручной, P/I/D параметры, ошибка и выход |
| Таймер | Пуск/Стоп/Сброс, установка длительности |
| Счётчик | Сброс, текущее значение, установка значения |
| Танк (Tank) | Уровень заполнения, объём, данные сенсоров |

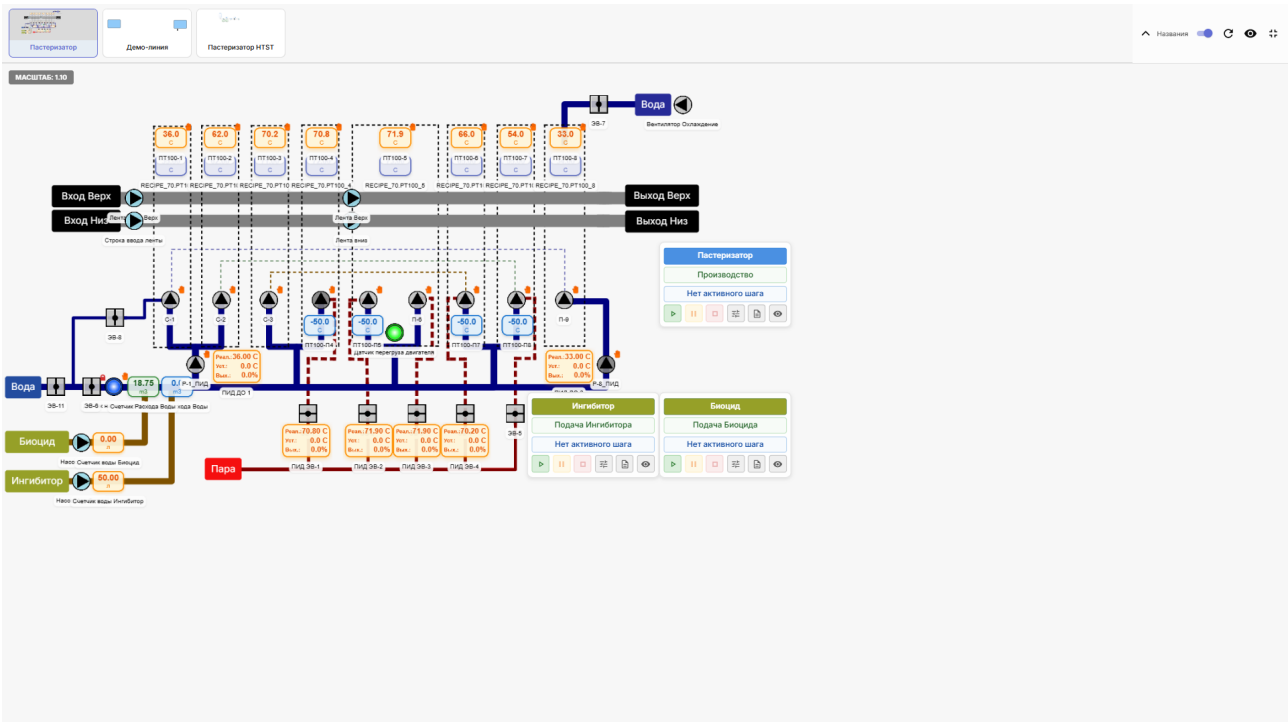
Вкладка «Статус» (Status): Текущее состояние в реальном времени.

Вкладка «Таблица» (Table): Табличное представление параметров.

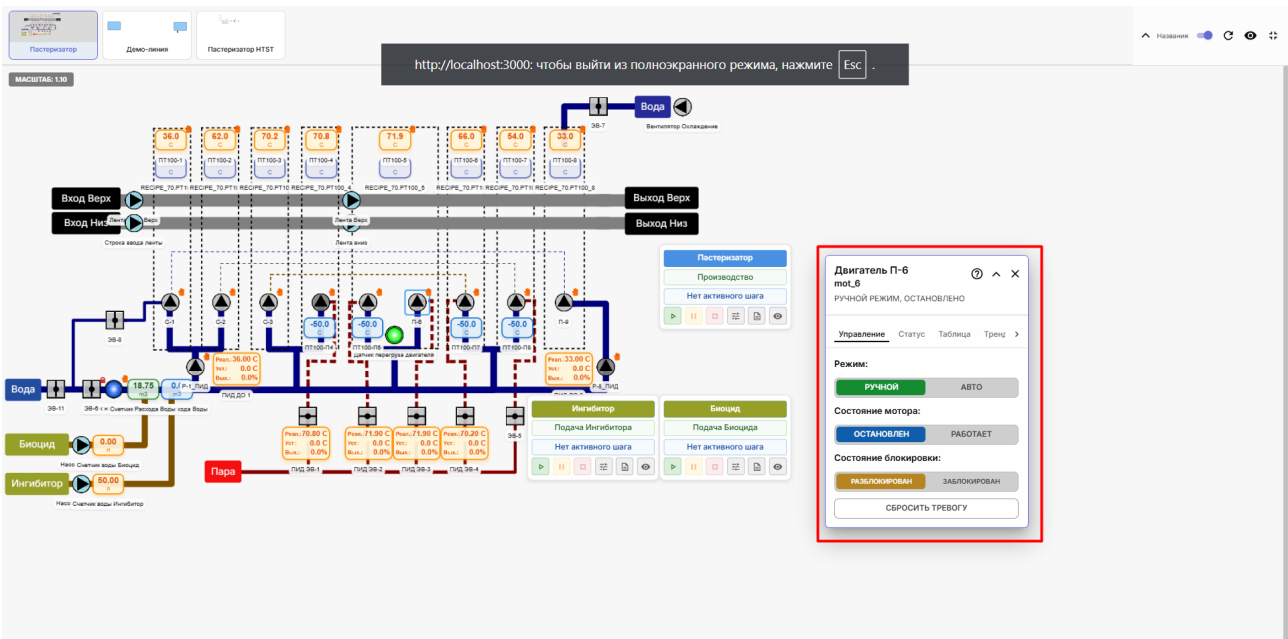
Вкладка «Тренд» (Trend): Миниатюрный график истории значений.

Вкладка «Логи» (Logs): Журнал событий устройства.

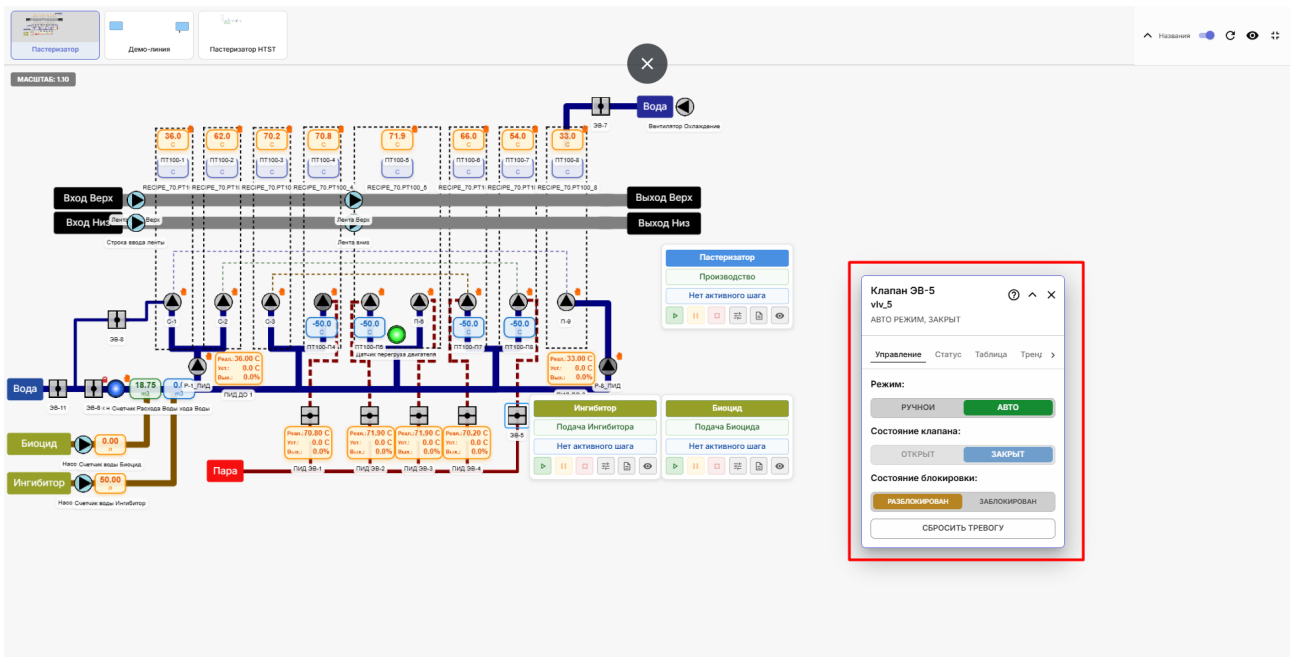
Вкладка «Условия перехода» (Transition Conditions): Активные условия последовательности (если устройство участвует).



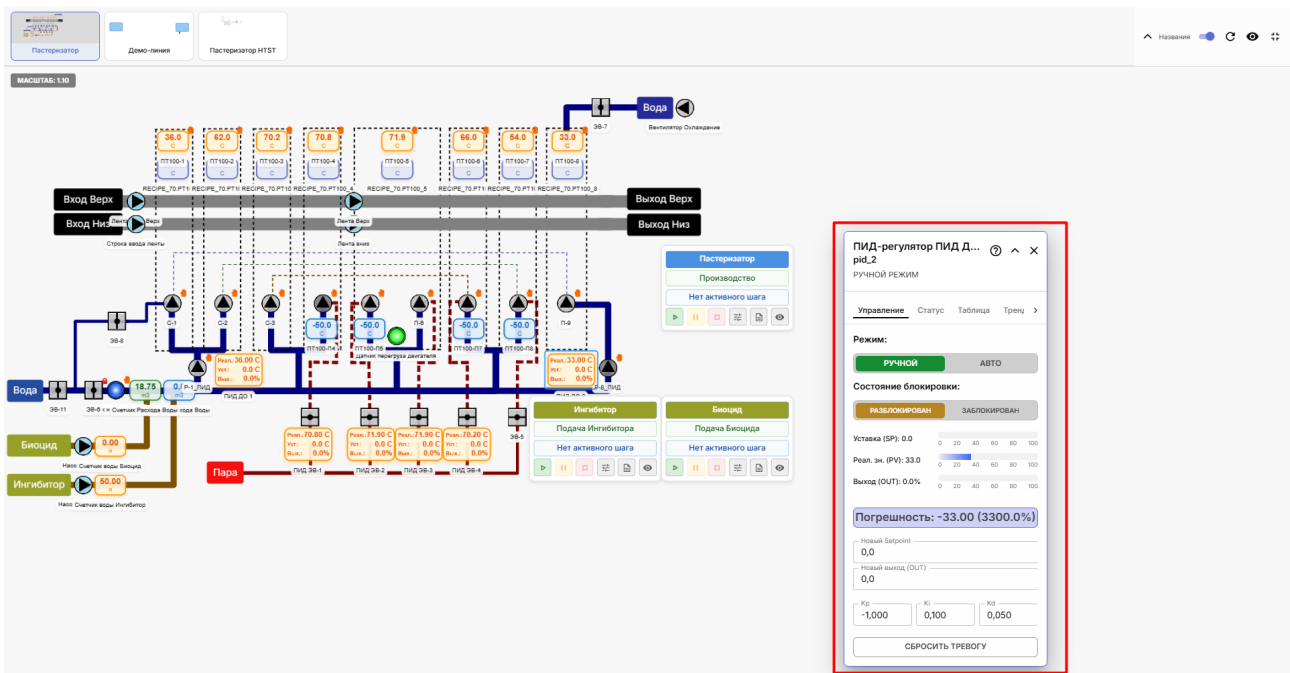
Полный экран просмотра схемы: цветные устройства с анимацией потока по трубам, блок последовательности с текущим шагом, полоса аварий сверху.



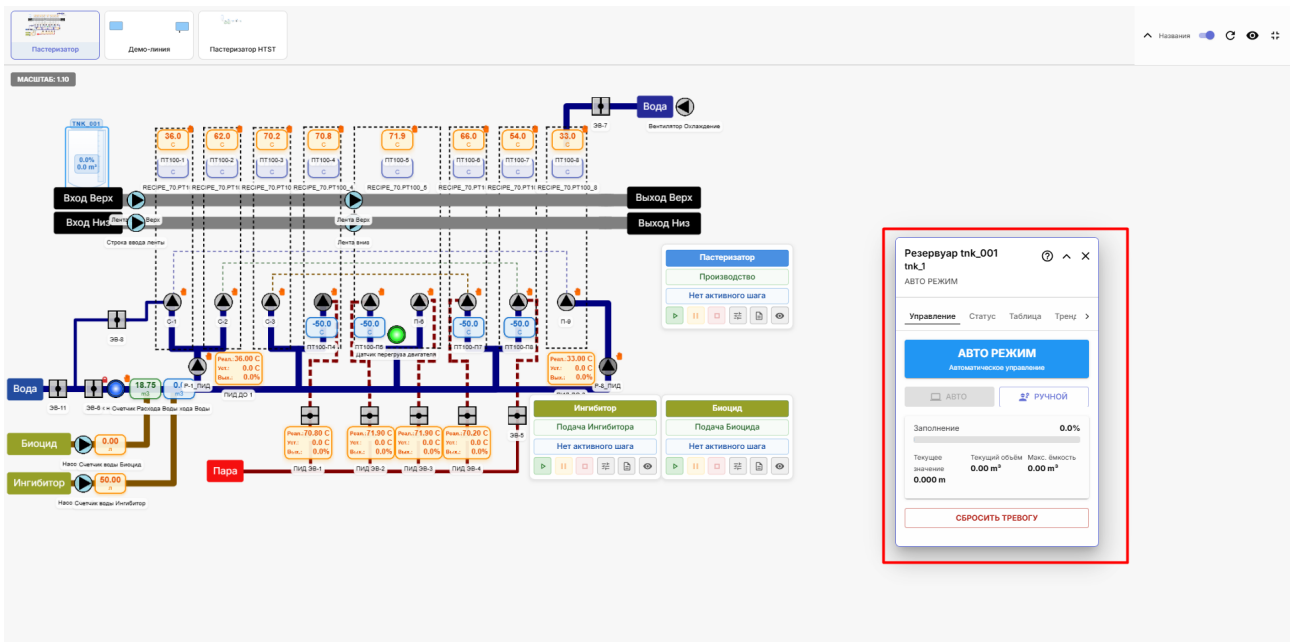
Поряд мотора: вкладка Control с кнопками Start/Stop, переключатель Auto/Manual, частота. Зелёный индикатор — мотор работает.



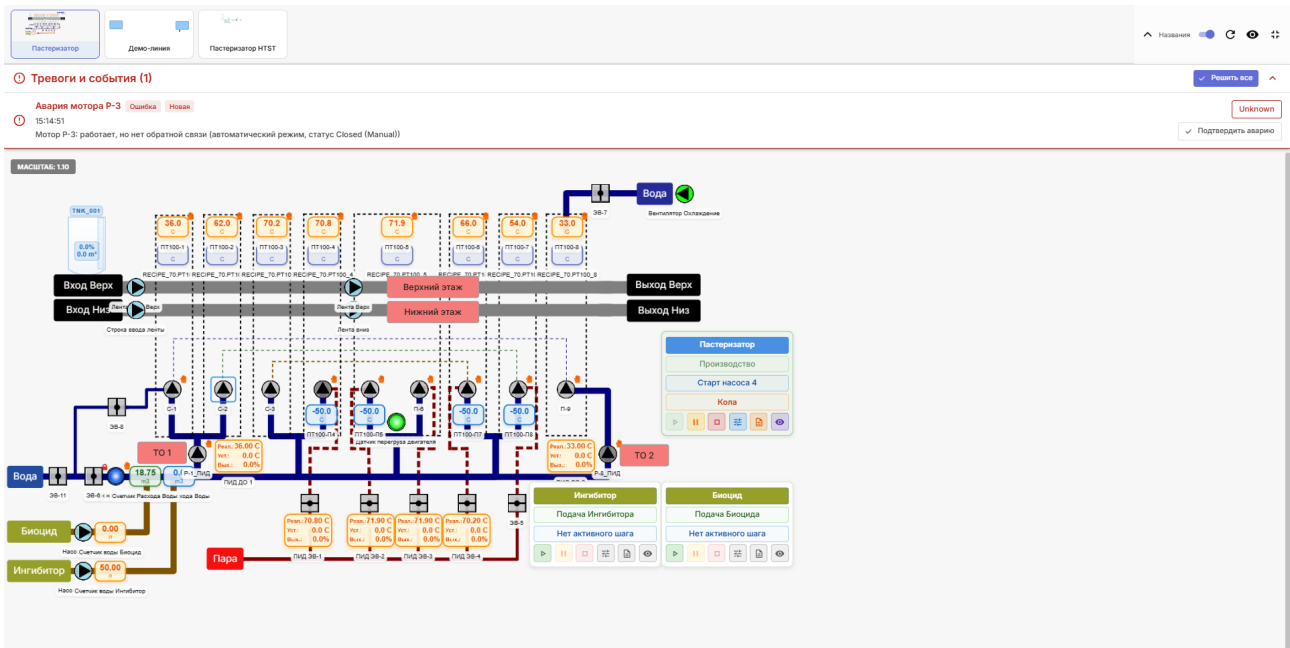
Рорир клапана: кнопки Open/Close, слайдер процента открытия, режим Auto/Manual.



Рорир PID: поля Setpoint, P, I, D, индикатор Error и Output, переключатель Auto/Manual.



Рорир танка: визуализация уровня заполнения, таблица сенсоров (уровень, давление), объём.



Полоса аварий сверху экрана: красные/жёлтые сообщения с timestamp, серьёзностью и именем устройства.

Техническая документация

WebSocket подключения для просмотра схемы:

| WebSocket путь | Описание |
|--------------------------------|------------------------------------|
| ws/schema/{schemald}/devices | Все устройства на схеме (реалтайм) |
| ws/schema/{schemald}/pipes | Все трубопроводы (анимация потока) |
| ws/schema/{schemald}/sequences | Все последовательности на схеме |
| ws/schema/{schemald}/batches | Партии и производственные линии |
| ws/schema/{schemald}/accidents | Аварии для данной схемы |

Формат события обновления устройств:

```
{
  "type": "schema_devices_status",
  "devices": {
    "mot_001": {
      "device": "mot_001",
      "current_status": {
        "value": 1,
        "is_running": true,
        "mode": "auto",
        "frequency": 50.0,
        "interlock": false
      },
      "status_icon": "/media/icons/motor_running.svg",
      "position_x": 100,
      "position_y": 200,
      "angle": 0,
      "updated_at": "2026-04-07T12:00:00Z"
    }
  }
}
```

Дельта-обновления (только изменённые поля):

```
{
  "type": "device_status_update",
  "device": "mot_001",
  "delta": true,
  "changes": {
    "frequency": 45.0
  }
}
```

API управления устройством:

| Метод | Endpoint | Описание | Тело |
|-------|---|---------------------|---------------------------|
| POST | /api/{device-type}/{id}/set_mode/ | Авто/Ручной | {"mode": "manual"} |
| POST | /api/{device-type}/{id}/set_manual_value/ | Установить значение | {"value": 50.0} |
| POST | /api/{device-type}/{id}/set_simulation/ | Симуляция | {"simulation": true} |
| POST | /api/{device-type}/{id}/set_interlock/ | Блокировка | {"interlock": true} |
| POST | /api/{device-type}/{id}/update_status/ | Обновить статус | {"current_status": {...}} |

WebSocket команды (через фронтенд websocketService):

```
{
  "type": "device_command",
  "device": "mot_001",
  "command": "start",
  "params": {},
  "options": {}
}
```

Ответ:

```
{
  "type": "command_response",
  "device": "mot_001",
  "command": "start",
  "status": "success",
  "data": {...},
  "timestamp": "2026-04-07T12:00:00Z"
}
```

6. Устройства (Devices)

Для пользователя

Страница управления устройствами позволяет создавать, редактировать и удалять все типы устройств.

Типы устройств:

| Тип | Код | Описание |
|------------------------|-----|--|
| Клапан | VLV | Запорная/регулирующая арматура |
| Мотор | MOT | Электродвигатели |
| Аналоговый вход | AI | Аналоговые датчики (температура, давление, расход) |
| Аналоговый выход | AO | Аналоговые выходы (4-20 mA, 0-10V) |
| Дискретный вход | DI | Сигналы вкл/выкл (концевые, готовности) |
| Дискретный выход | DO | Дискретные команды |
| Координационный сигнал | COS | Межсистемные координационные сигналы |
| PID-регулятор | PID | Контур автоматического регулирования |
| Счётчик | CNT | Счётчики импульсов |
| Танк | TNK | Ёмкости с расчётом объёма |
| Таймер | TMR | Программные таймеры |

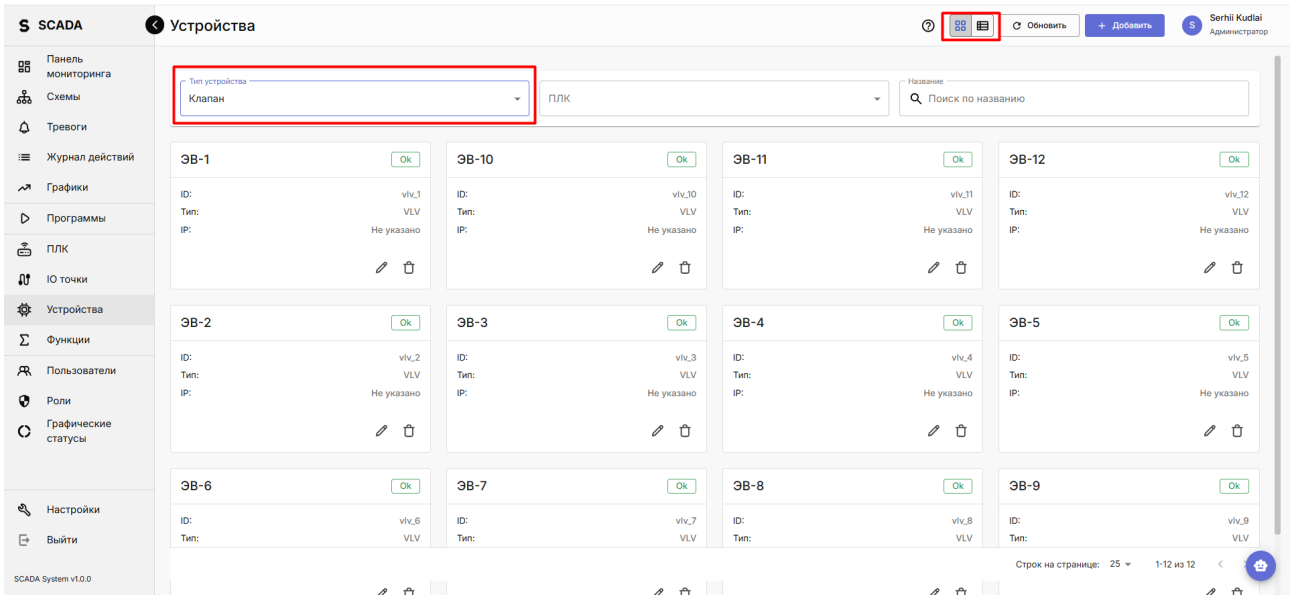
Элементы экрана:

| Элемент | Описание |
|--------------------|-----------------------------------|
| Переключатель вида | Сетка / Таблица |
| Кнопка «+» | Создание нового устройства |
| Фильтр по типу | Выпадающий список типов устройств |
| Поиск | Поиск по названию |
| Обновить | Перезагрузка списка |

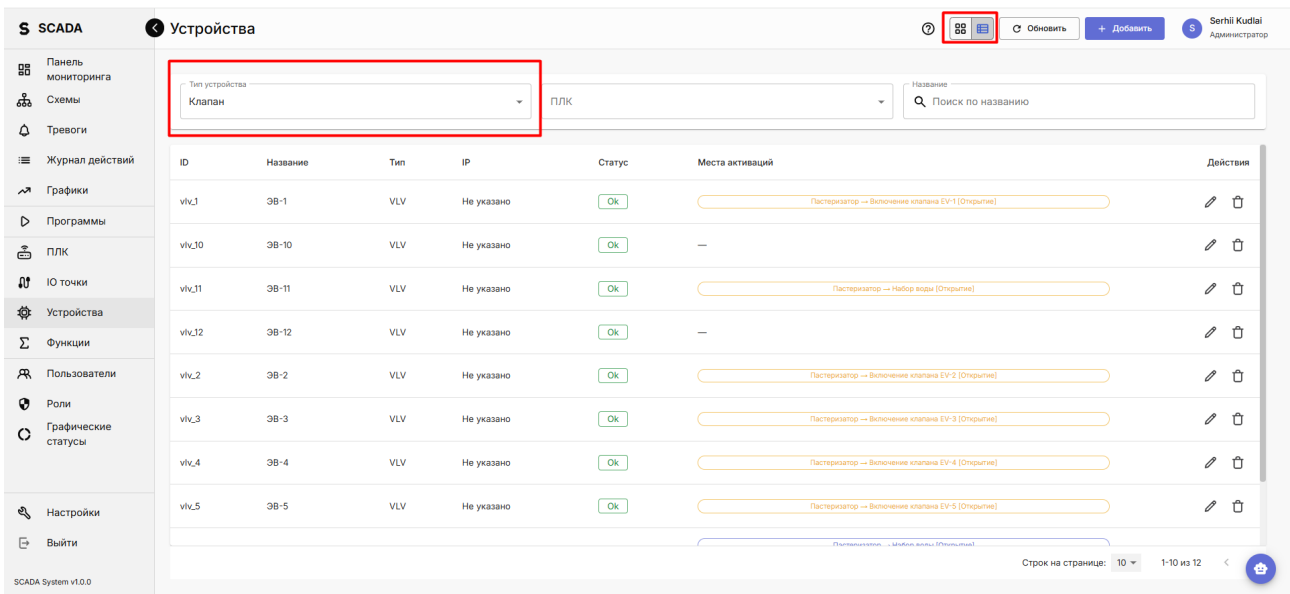
Форма устройства (создание/редактирование):

| Поле | Описание |
|-----------------------|--|
| ID устройства | Уникальный идентификатор (например, mot_001) |
| Название | Человекочитаемое название |
| Переводы названий | Мультиязычные названия |
| Тип | Выбор типа устройства |
| Описание | Текстовое описание |
| Значение по умолчанию | Начальное значение |
| Единица измерения | Инженерная единица |
| Min / Max | Диапазон допустимых значений |
| Масштабирование | Параметры преобразования сигнала |
| Кнопка «Сохранить» | Сохранение |
| Кнопка «Отмена» | Возврат без сохранения |

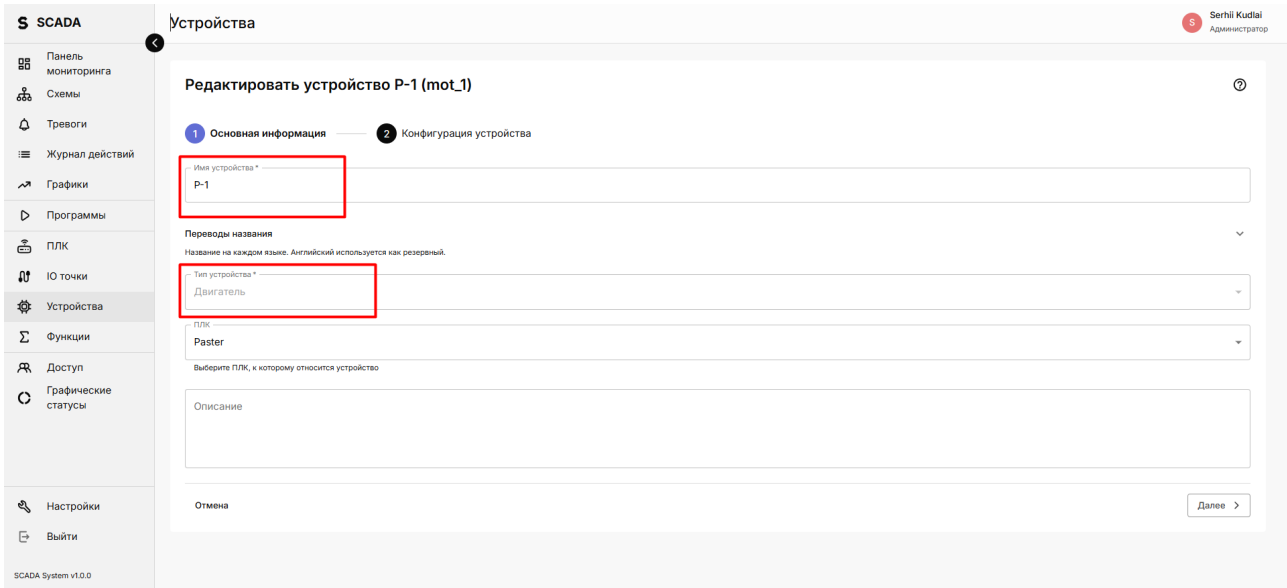
| Поле | Описание |
|------------------|---|
| Кнопка «Удалить» | Удаление устройства (в режиме редактирования) |



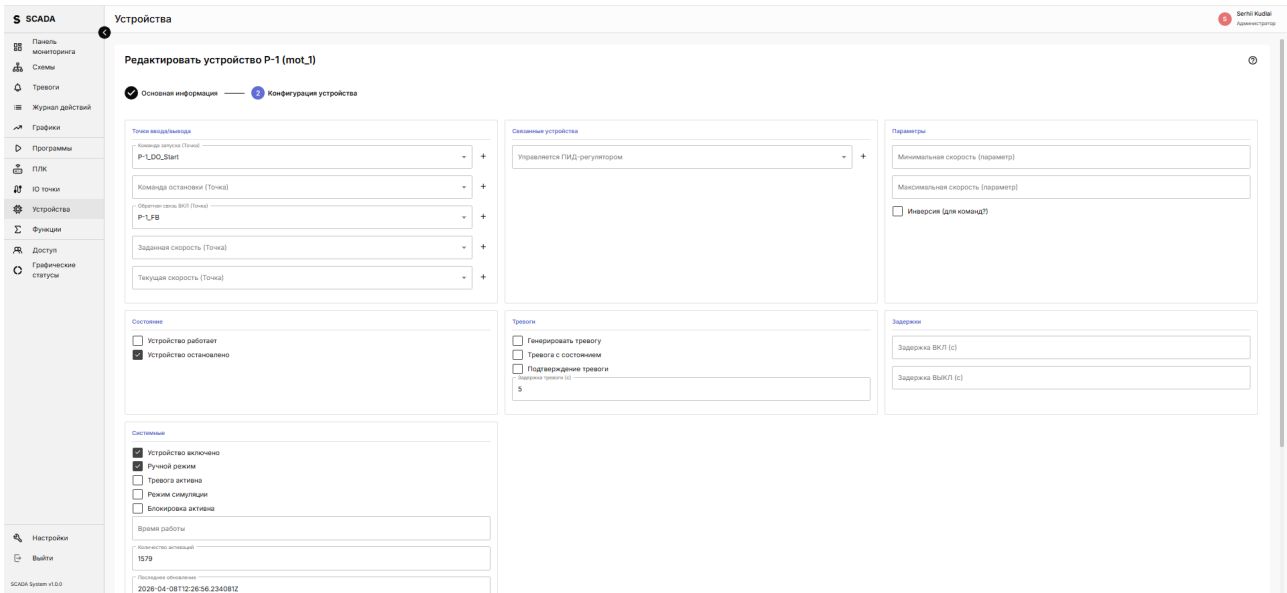
Страница устройств в виде сетки: карточки с иконками устройств (мотор, клапан и т.д.), названием, типом, значением. Фильтр по типу сверху.



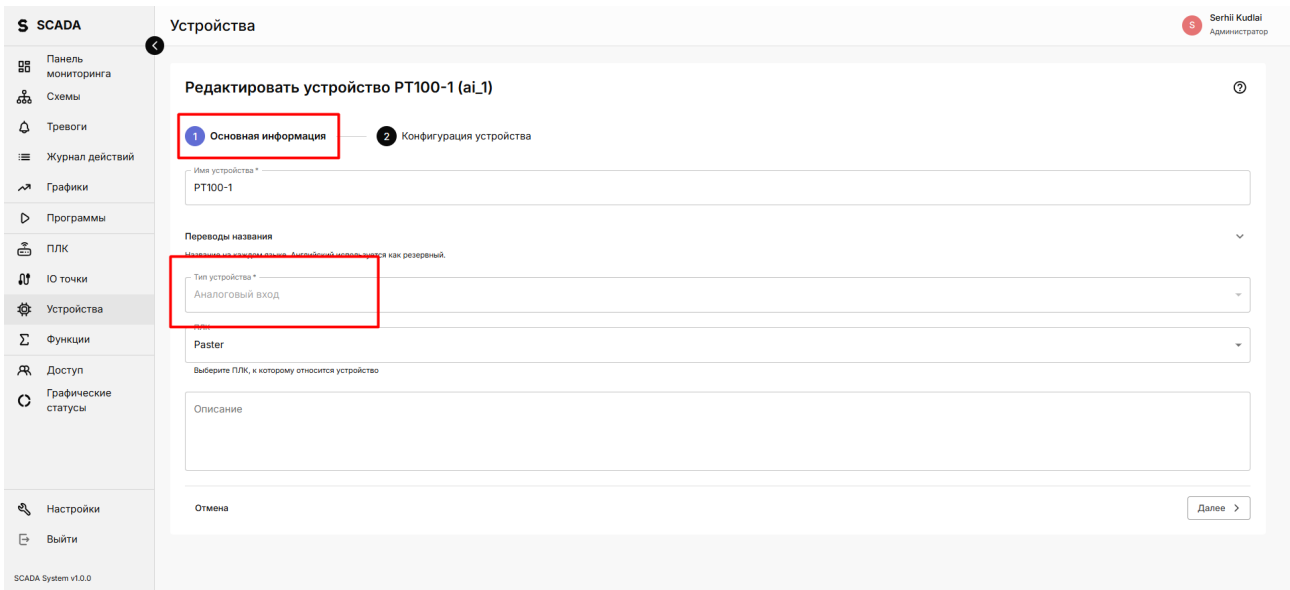
Табличный вид: колонки ID, Название, Тип, Значение по умолчанию, Единица, Действия.



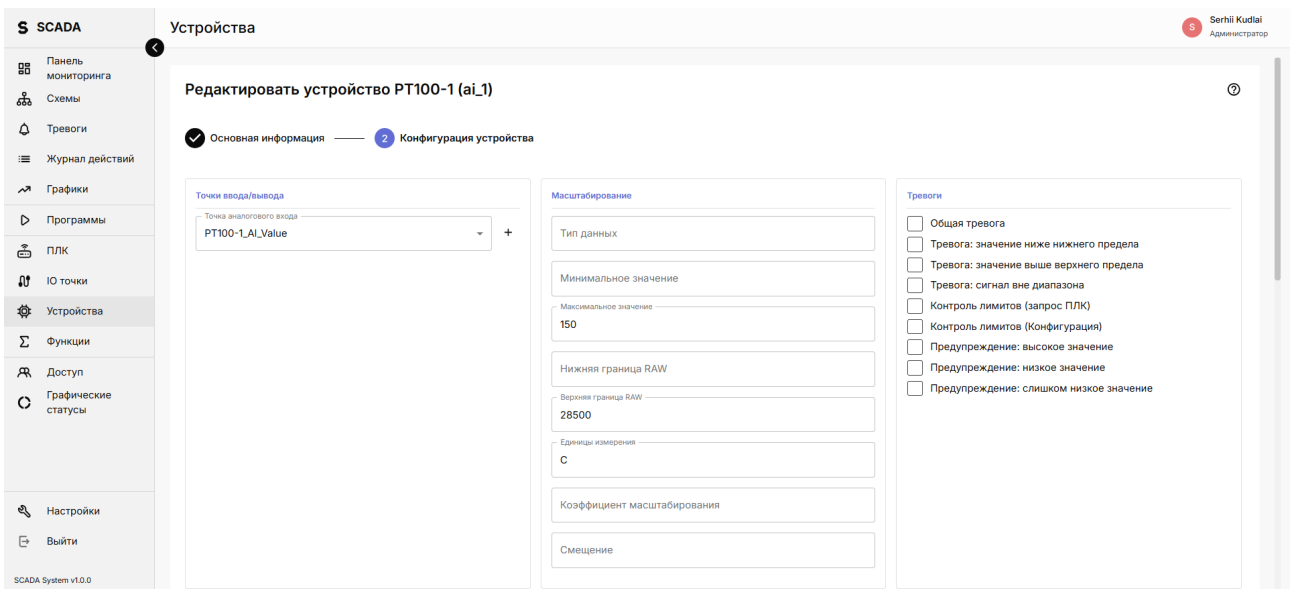
Форма создания мотора: основные поля — ID, Name, PLC, Description, тип устройства.



Форма мотора: min/max speed, задержки, generate_alarm, кнопки сохранения.



Форма аналогового входа: Min/Max, Units, Scaling, Raw Min/Max.



Форма AI: пороги предупреждений, значение по умолчанию и остальные поля.

Техническая документация

API устройств (общий для всех типов):

| Метод | Endpoint | Описание |
|--------|---------------------------|---------------------------|
| GET | /api/devices/all_devices/ | Все устройства всех типов |
| GET | /api/{device-type}/ | Список устройств типа |
| POST | /api/{device-type}/ | Создание устройства |
| GET | /api/{device-type}/{id}/ | Получение устройства |
| PUT | /api/{device-type}/{id}/ | Обновление |
| PATCH | /api/{device-type}/{id}/ | Частичное обновление |
| DELETE | /api/{device-type}/{id}/ | Удаление |

Типы endpoint'ов:

- /api/analog-inputs/
- /api/analog-outputs/
- /api/digital-inputs/
- /api/digital-outputs/
- /api/motors/
- /api/valves/
- /api/pid-controllers/
- /api/counters/
- /api/timers/
- /api/coordination-signals/
- /api/variable-speed-drives/

Дополнительные действия (для каждого типа):

| Метод | Endpoint | Описание |
|-------|-------------------------------------|--|
| POST | /api/{id}/set_mode/ | Переключить Авто/Ручной |
| POST | /api/{id}/set_simulation/ | Вкл/выкл симуляцию |
| POST | /api/{id}/set_interlock/ | Установить/снять блокировку |
| POST | /api/{id}/update_status/ | Обновить текущий статус |
| POST | /api/{id}/set_manual_value/ | Установить ручное значение |
| POST | /api/{id}/trigger_websocket_update/ | Принудительно обновить WS (AllowAnonymous) |

WebSocket: Обновление устройства

Подключение: ws://ws/device/{device_id}

```
{
  "type": "device_status_update",
  "device": "ai_001",
  "data": {
    "current_status": {
      "value": 23.5,
      "mode": "auto",
      "simulation": false,
      "interlock": false,
      "quality": "good"
    },
    "status_icon": null,
    "updated_at": "2026-04-07T12:00:00Z"
  }
}
```

7. ПЛК (PLCs)

Для пользователя

Страница управления программируемыми логическими контроллерами.

Элементы экрана:

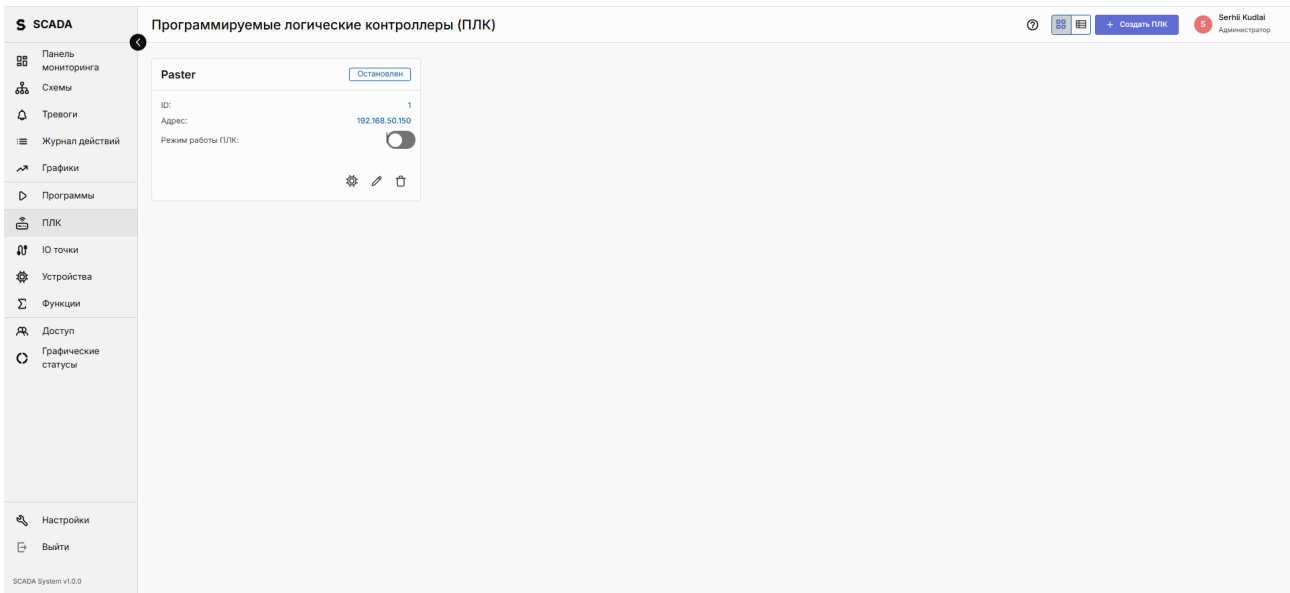
| Элемент | Описание |
|--------------------------|---|
| Список ПЛК | Карточки/таблица с ПЛК |
| Индикатор статуса | Зелёный — онлайн, серый — оффлайн, красный — ошибка |
| Кнопка «+» | Добавить новый ПЛК |
| Кнопка «Hardware Config» | Перейти в конфигуратор оборудования |
| Переключатель Run Mode | Вкл/выкл режим выполнения ПЛК |

Информация на карточке ПЛК:

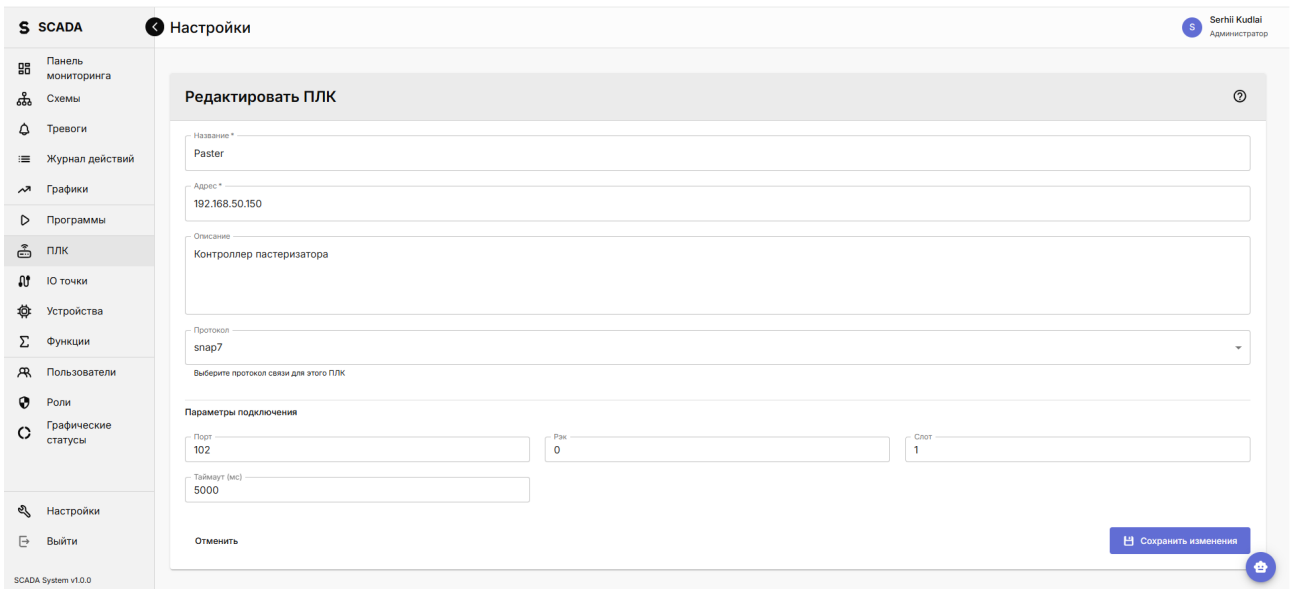
| Поле | Описание |
|----------------------|--|
| Название | Имя ПЛК |
| Протокол | Тип связи (Siemens S7, Modbus, OPC UA) |
| Статус | Online / Offline |
| Время цикла | Скорость опроса |
| IP-адрес | Сетевой адрес |
| Количество устройств | Кол-во привязанных устройств |

Форма ПЛК:

| Поле | Описание |
|--------------------------|-------------------------|
| Название | Имя ПЛК |
| Описание | Текстовое описание |
| Протокол | Выбор протокола связи |
| IP / Порт | Сетевые настройки |
| Rack / Slot | Для Siemens S7 |
| Кнопка «Тест соединения» | Проверка связи с ПЛК |
| Кнопка «Сохранить» | Сохранение конфигурации |



Список ПЛК: карточки с индикатором статуса (зелёный/серый), названием, протоколом, IP-адресом, переключателем Run Mode, кнопкой Hardware Config.



Форма настройки ПЛК: поля Name, Protocol dropdown, IP Address, Port, Rack, Slot, кнопка Test Connection.

Техническая документация

API ПЛК:

| Метод | Endpoint | Описание |
|--------|------------------------------|----------------------|
| GET | /api/plcs/ | Список ПЛК |
| POST | /api/plcs/ | Создание ПЛК |
| GET | /api/plcs/{id}/ | Получение ПЛК |
| PUT | /api/plcs/{id}/ | Обновление |
| DELETE | /api/plcs/{id}/ | Удаление |
| GET | /api/plcs/{id}/status/ | Статус ПЛК |
| POST | /api/plcs/{id}/set-run-mode/ | Переключить Run Mode |

| Метод | Endpoint | Описание |
|-------|-------------------|-----------------|
| GET | /api/plcs/status/ | Статус всех ПЛК |

Симуляция ПЛК:

| Метод | Endpoint | Описание |
|-------|-----------------------------|-----------------------|
| POST | /api/plc/simulation/run/ | Запустить симуляцию |
| GET | /api/plc/simulation/status/ | Статус симуляции |
| POST | /api/plc/service/start/ | Запустить сервис ПЛК |
| POST | /api/plc/service/stop/ | Остановить сервис ПЛК |

8. Последовательности (Sequences)

Для пользователя

Последовательности определяют порядок работы производственного оборудования — шаги, переходы между ними и условия переключения.

Элементы экрана:

| Элемент | Описание |
|----------------------------|---|
| Список последовательностей | Карточки/таблица с фильтрацией |
| Кнопка «+» | Создать новую последовательность |
| Индикатор статуса | Зелёный — работает, серый — остановлена, жёлтый — пауза |
| Фильтр по статусу | Все / Работающие / Остановленные |
| Поиск | Поиск по имени |

Редактор последовательности (вкладки):

| Вкладка | Описание |
|--------------------------|---|
| Свойства | Общие настройки (имя, описание, привязка устройств) |
| Шаги | Список шагов с порядком выполнения |
| Условия перехода | Настройка условий перехода между шагами |
| События | Обработка событий (Events) |
| Аварии | Обработка аварий (Accidents) |
| Опции | Настройка опций |
| Переменные рецепта | Определение переменных |
| Производственные рецепты | Рецепты для данной последовательности |
| Импульсные модули | Управление импульсными модулями |
| Партии | Управление партиями (Batches) |

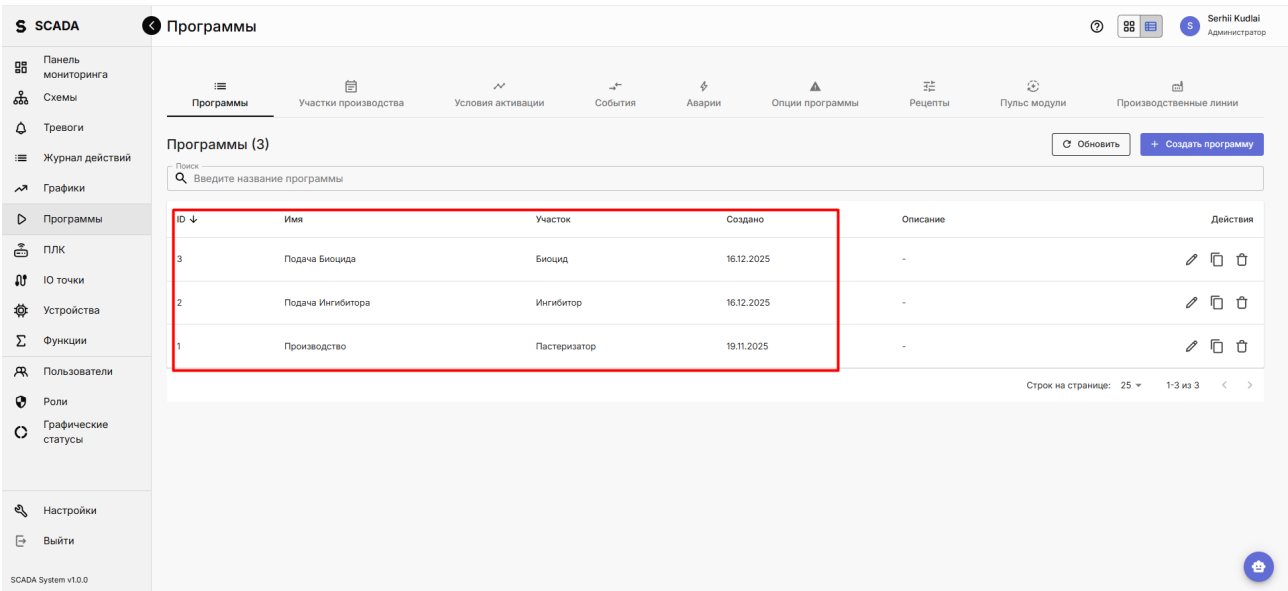
Настройка шага:

| Поле | Описание |
|--------------------|--|
| Порядковый номер | Номер шага в последовательности |
| Название | Имя шага (+ переводы) |
| Описание | Текстовое описание |
| Устройства | Привязанные устройства и их значения на этом шаге |
| Следующие шаги | JSON-определение переходов: {"target_step_order": [condition_id, ...]} |
| Действие с партией | NONE / TRANSFER / COMPLETE |

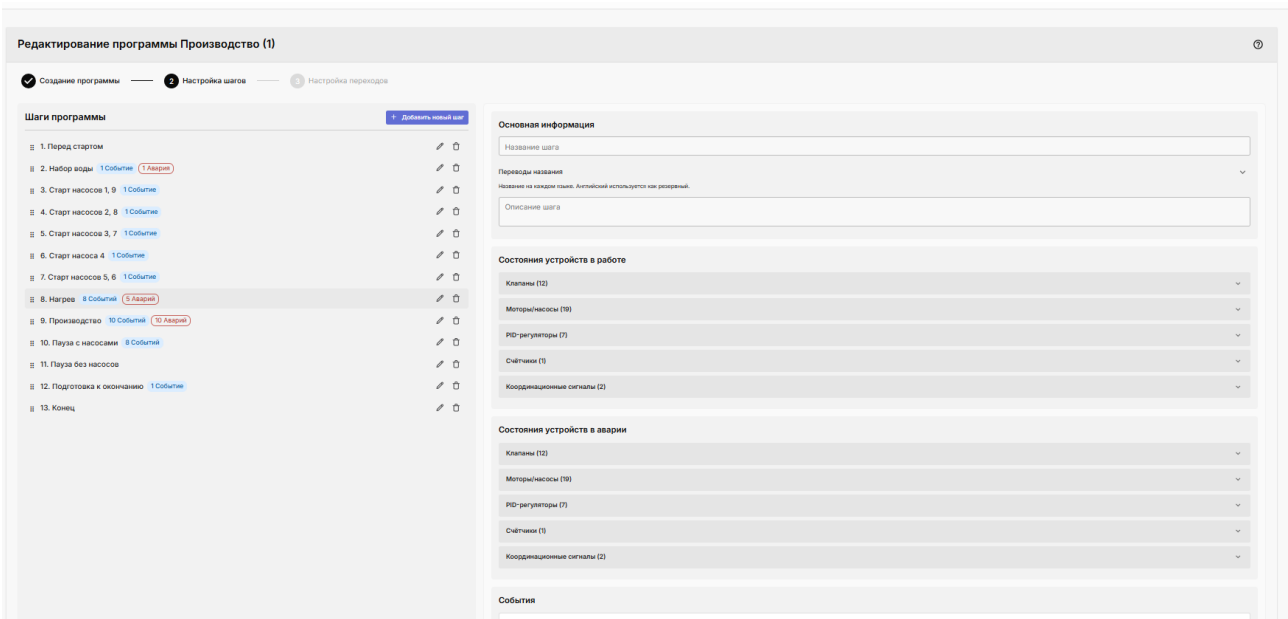
Условие перехода (Transition Condition):

| Поле | Описание |
|------------|--|
| Тип | Device (по значению устройства), Timer (по времени), Option (по опции) |
| Устройство | Какое устройство отслеживать (для Device) |

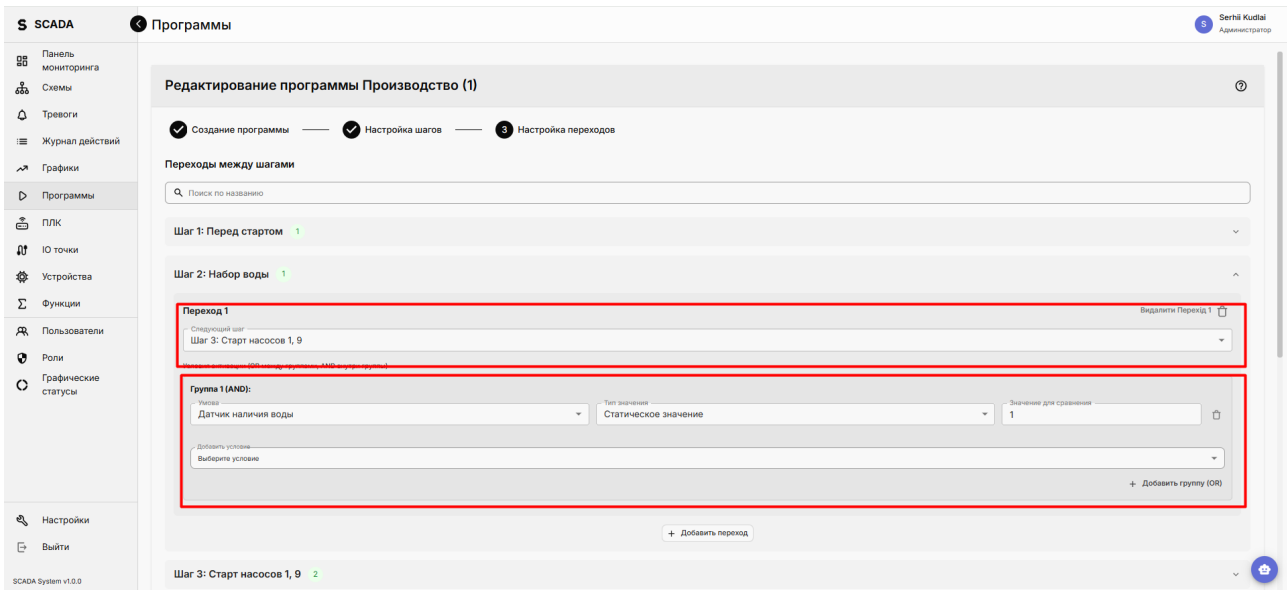
| Поле | Описание |
|--------------|---|
| Оператор | ==, !=, >, =, <= |
| Значение | Целевое значение для сравнения |
| Длительность | Время (для Timer) |
| Симуляция | Включить/выключить симуляцию условия (для тестирования) |



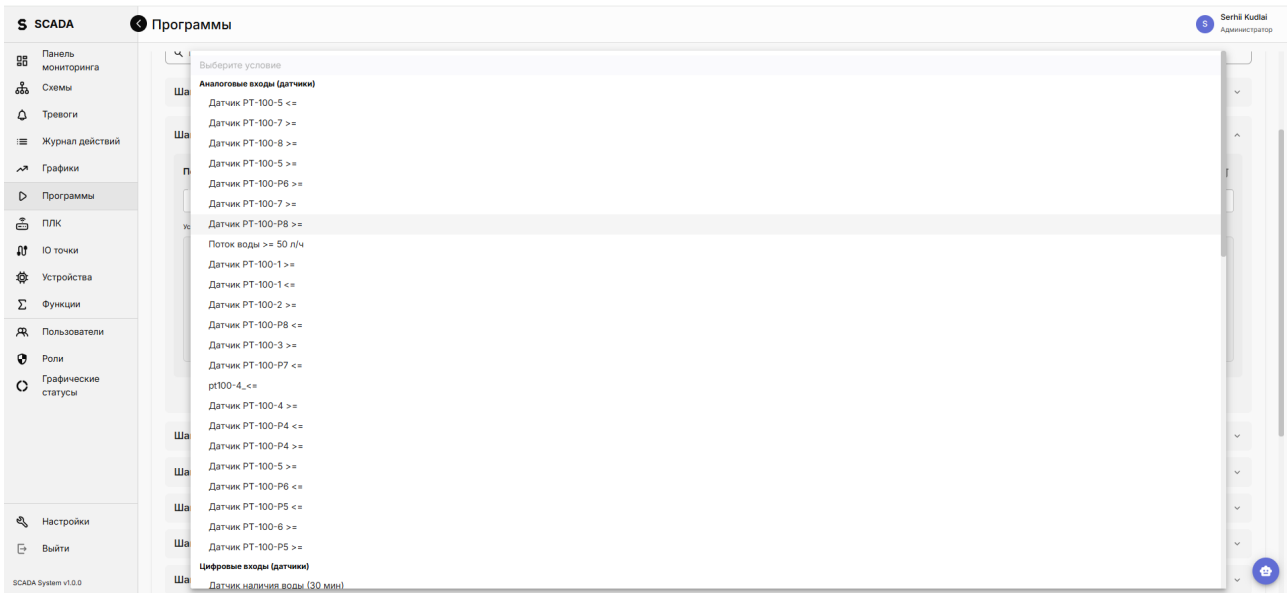
Список последовательностей: карточки с названием, индикатором статуса (зелёный/серый), текущим шагом, кнопками Edit/Delete.



Редактор шагов: нумерованный список шагов, для каждого — название, привязанные устройства, стрелки переходов, кнопки Add/Delete Step.



Настройка условий перехода: таблица с колонками Type, Device, Operator, Value, Duration, Simulation toggle.



Диалог создания условия перехода: выбор типа (Device/Timer/Option), выбор устройства из dropdown, оператор сравнения, поле значения.

Техническая документация

API последовательностей:

| Метод | Endpoint | Описание |
|-------|--------------------------------------|----------------------------|
| GET | /api/sequences/ | Список последовательностей |
| POST | /api/sequences/ | Создание |
| GET | /api/sequences/{id}/ | Получение |
| PUT | /api/sequences/{id}/ | Обновление |
| DELET | /api/sequences/{id}/ | Удаление |
| GET | /api/sequences/{id}/steps/ | Список шагов |
| POST | /api/sequences/{id}/add_step/ | Добавить шаг |
| POST | /api/sequences/{id}/set_steps_order/ | Изменить порядок шагов |

| Метод | Endpoint | Описание |
|-------|--|------------------|
| POST | /api/sequences/{id}/add_transition_conditions/ | Добавить условия |

API шагов:

| Метод | Endpoint | Описание |
|--------|---------------------------|-------------|
| GET | /api/sequence-steps/ | Все шаги |
| POST | /api/sequence-steps/ | Создать шаг |
| PUT | /api/sequence-steps/{id}/ | Обновить |
| DELETE | /api/sequence-steps/{id}/ | Удалить |

API условий перехода:

| Метод | Endpoint | Описание |
|--------|--|----------------------|
| GET | /api/transition-conditions/ | Все условия |
| POST | /api/transition-conditions/ | Создать |
| PUT | /api/transition-conditions/{id}/ | Обновить |
| DELETE | /api/transition-conditions/{id}/ | Удалить |
| POST | /api/transition-conditions/{id}/simulate/ | Симуляция условия |
| POST | /api/transition-conditions/clear_simulation/ | Сброс симуляции шага |

WebSocket: Статус последовательности

Подключение: ws://ws/sequence/{sequence_id}

```
{
  "type": "sequence_status",
  "id": 1,
  "name": "Production A",
  "is_running": true,
  "is_pause": false,
  "is_stop": false,
  "active_step": 3,
  "active_step_data": {
    "id": 3,
    "order": 3,
    "name": "Heating",
    "name_translations": {"ru": "Нагрев"},
    "description": "..."
  },
  "previous_step": 2,
  "current_step_start_time": "2026-04-07T11:30:00Z",
  "monitored_devices": {
    "ai_001": {"value": 72.3, "mode": "auto"}
  },
  "activated_devices": {
    "mot_001": {"value": 1, "is_running": true}
  },
  "conditions": [
    {
      "id": 10,
      "type": "device",
      "device_id": "ai_001",
      "operator": ">=",
      "value": 80.0,
    }
  ]
}
```

... (+9 строк)

9. Рецепты и переменные

Для пользователя

Рецепты определяют конкретные значения параметров устройств на каждом шаге последовательности. Переменные рецепта позволяют параметризовать эти значения.

Типы рецептов:

| Тип | Описание |
|---|---|
| Recipe (Рецепт шагов) | Определяет значения устройств для каждого шага последовательности |
| Production Recipe (Производственный рецепт) | Набор переменных для конкретного производственного задания |

Форма рецепта шагов:

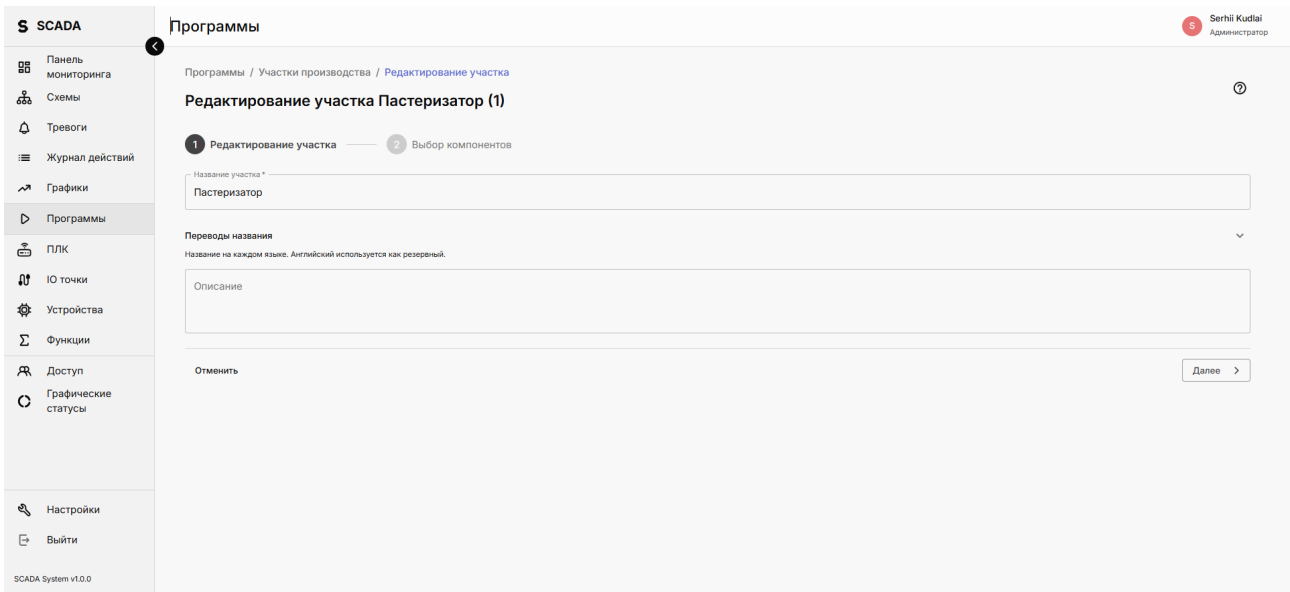
| Поле | Описание |
|--------------------|--|
| Название | Имя рецепта (+ переводы) |
| Последовательность | Привязка к последовательности |
| Шаги | Для каждого шага — набор «устройство → значение» |

Форма производственного рецепта:

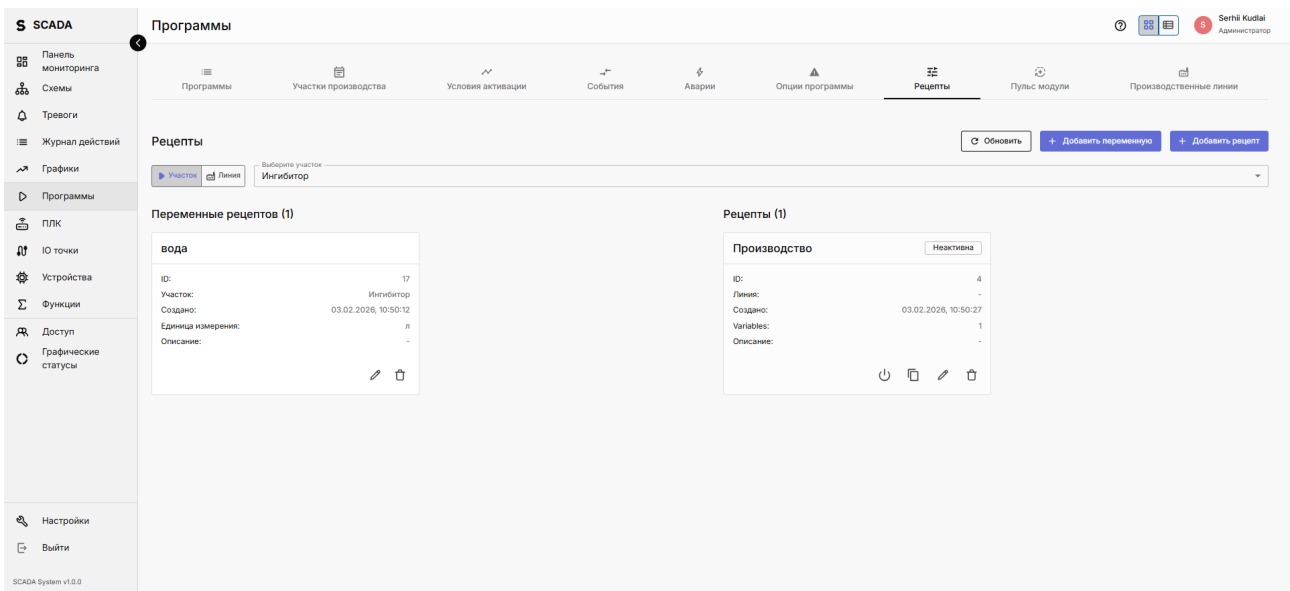
| Поле | Описание |
|------------|---|
| Название | Имя рецепта |
| Привязка | К последовательности ИЛИ к производственной линии (XOR) |
| Переменные | Список переменных со значениями |

Переменные рецепта:

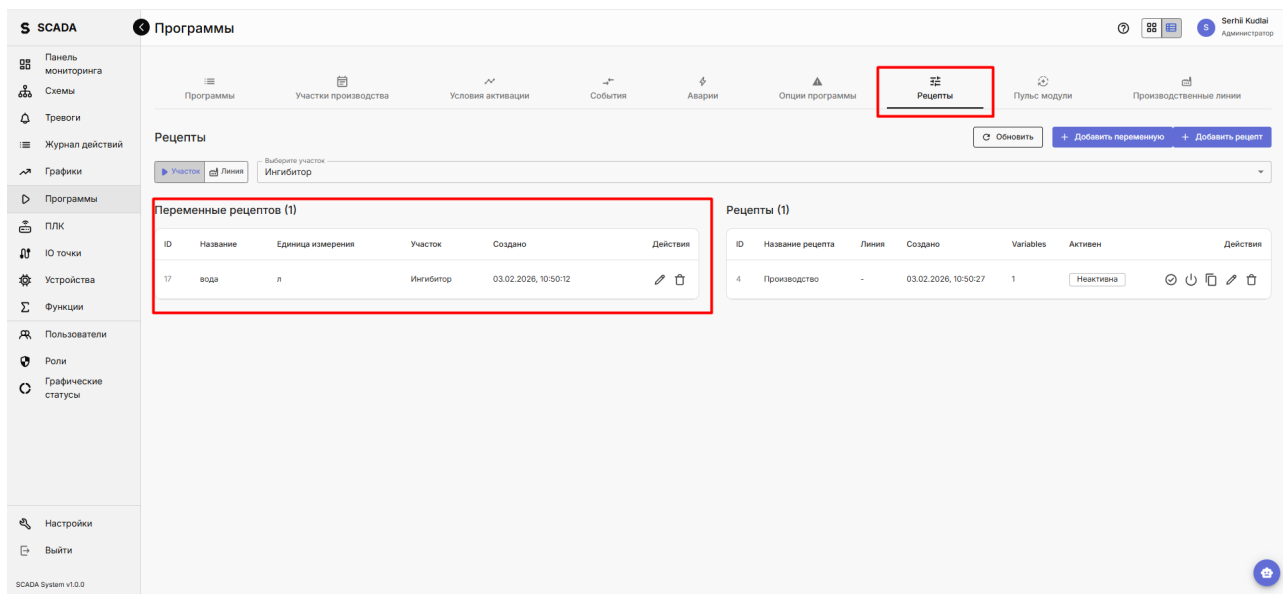
| Поле | Описание |
|-----------------------|---|
| Название | Имя переменной (+ переводы) |
| Привязка | К последовательности ИЛИ к производственной линии (XOR) |
| Тип | Числовой, текстовый |
| Значение по умолчанию | Начальное значение |
| Min / Max | Ограничения |



Мастер создания рецепта: шаг 1 — имя и последовательность, шаг 2 — для каждого шага таблица «устройство → значение».



Управление производственными рецептами: таблица с рецептами, кнопки Create/Edit/Delete/Duplicate, переключатель Sequence/Production Line.



Управление переменными: список переменных с именем, типом, значением по умолчанию, Min/Max. Кнопки добавления/удаления.

Техническая документация

API рецептов:

| Метод | Endpoint | Описание |
|--------|--------------------|-----------------------|
| GET | /api/recipes/ | Список рецептов шагов |
| POST | /api/recipes/ | Создать рецепт |
| PUT | /api/recipes/{id}/ | Обновить |
| DELETE | /api/recipes/{id}/ | Удалить |

API производственных рецептов:

| Метод | Endpoint | Описание | Параметры |
|-------|--|------------------|--------------------------------------|
| GET | /api/production-recipes/ | Все рецепты | ?sequence=ID или ?production_line=ID |
| POST | /api/production-recipes/ | Создать | — |
| PUT | /api/production-recipes/{id}/ | Обновить | — |
| DELET | /api/production-recipes/{id}/ | Удалить | — |
| POST | /api/production-recipes/{id}/select-production-recipe/ | Выбрать активный | — |

API переменных:

| Метод | Endpoint | Описание | Параметры |
|-------|--|----------------|--------------------------------------|
| GET | /api/production-recipe-variables/ | Все переменные | ?sequence=ID или ?production_line=ID |
| POST | /api/production-recipe-variables/ | Создать | — |
| PUT | /api/production-recipe-variables/{id}/ | Обновить | — |
| DELET | /api/production-recipe-variables/{id}/ | Удалить | — |

E

10. События и аварии (Events & Accidents)

Для пользователя

События (Events) — условия, которые выполняют действия во время работы последовательности (например, переключение при достижении значения).

Аварии (Accidents) — критические условия, останавливающие процесс при возникновении опасной ситуации.

Форма события:

| Поле | Описание |
|------------|--------------------------------|
| Название | Имя события (+ переводы) |
| Тип | Тип срабатывания |
| Устройство | Отслеживаемое устройство |
| Условие | Оператор сравнения и значение |
| Действие | Что выполнить при срабатывании |

Форма аварии:

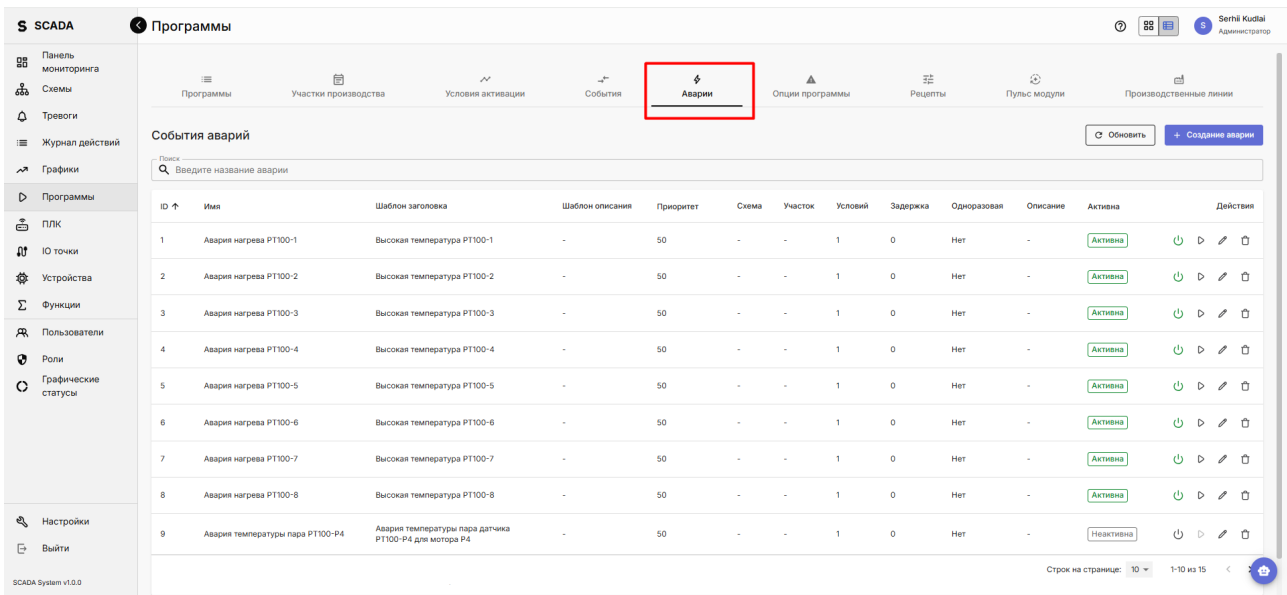
| Поле | Описание |
|-------------|-----------------------------------|
| Название | Имя аварии (+ переводы) |
| Серьёзность | INFO / WARNING / ERROR / CRITICAL |
| Устройство | Отслеживаемое устройство |
| Условие | Оператор и значение |
| Блокировка | Какие устройства заблокировать |

События (12)

| ID ↑ | Имя | Тип действия | Участок | Условия активации | Статус | Действия |
|------|--------------------------|-------------------------|--------------|---------------------------------------|---------|----------|
| 1 | Набор воды | Активировать устройства | Пастеризатор | Датчик наличия воды | Активна | ⏻ ⏸ ⏹ |
| 2 | Включение верхнего этажа | Активировать устройства | Пастеризатор | Селектор Пастеризатора "Верхний этаж" | Активна | ⏻ ⏸ ⏹ |
| 3 | Включение нижнего этажа | Активировать устройства | Пастеризатор | Селектор Пастеризатора "Нижний этаж" | Активна | ⏻ ⏸ ⏹ |
| 4 | Включение клапана EV-1 | Активировать устройства | Пастеризатор | Датчик PT-100-P4 <= | Активна | ⏻ ⏸ ⏹ |
| 5 | Включение клапана EV-3 | Активировать устройства | Пастеризатор | Датчик PT-100-P6 <= | Активна | ⏻ ⏸ ⏹ |
| 6 | Включение клапана EV-2 | Активировать устройства | Пастеризатор | Датчик PT-100-P5 <= | Активна | ⏻ ⏸ ⏹ |
| 7 | Включение клапана EV-4 | Активировать устройства | Пастеризатор | Датчик PT-100-P7 <= | Активна | ⏻ ⏸ ⏹ |
| 8 | Включение клапана EV-5 | Активировать устройства | Пастеризатор | Датчик PT-100-P8 <= | Активна | ⏻ ⏸ ⏹ |
| 9 | Активация вентилятора | Активировать устройства | Пастеризатор | Датчик наличия воды (30 мин) | Активна | ⏻ ⏸ ⏹ |
| 10 | Активация TO 2 | Активировать устройства | Пастеризатор | Селектор Теплообменника Зона 2 | Активна | ⏻ ⏸ ⏹ |

Строк на странице: 10 < Назад 1 2 Далее > 1-10 из 12

Управление событиями: таблица с событиями, колонки Name, Device, Condition, Action, кнопки Add/Edit/Delete.



Управление авариями: таблица с цветовой индикацией серьёзности (INFO-синий, WARNING-жёлтый, ERROR-красный, CRITICAL-тёмно-красный).

Техническая документация

API событий:

| Метод | Endpoint | Описание |
|--------|------------------------------|-----------------------------|
| GET | /api/events/ | Список событий |
| POST | /api/events/ | Создать |
| PUT | /api/events/{id}/ | Обновить |
| DELETE | /api/events/{id}/ | Удалить |
| POST | /api/events/{id}/set_active/ | Активировать/деактивировать |

API аварий:

| Метод | Endpoint | Описание |
|-------|----------------------------------|----------------------------|
| GET | /api/accident-events/ | Список определений аварий |
| POST | /api/accident-events/ | Создать определение |
| GET | /api/accidents/ | Список произошедших аварий |
| POST | /api/accidents/{id}/acknowledge/ | Подтвердить аварию |
| POST | /api/accidents/{id}/resolve/ | Разрешить аварию |
| POST | /api/accidents/resolve-all/ | Разрешить все |
| GET | /api/accidents/active_events/ | Текущие активные аварии |

WebSocket: Аварии

Подключение: ws://ws/accidents

```
{
  "type": "accident_created",
  "accident_id": 42,
  "title": "Высокая температура",
  "description": "AI_001 превысил порог 95°C",
  "status": "NEW",
  "severity_level": "CRITICAL",
  "device_id": "ai_001",
  "device_name": "Temperature Sensor 1",
  "schema_id": 1,
  "occurred_at": "2026-04-07T12:00:00Z"
}
```

11. Опции (Options)

Для пользователя

Опции — кнопки/переключатели, доступные оператору во время работы последовательности. Позволяют выбирать альтернативные пути выполнения.

Форма опции:

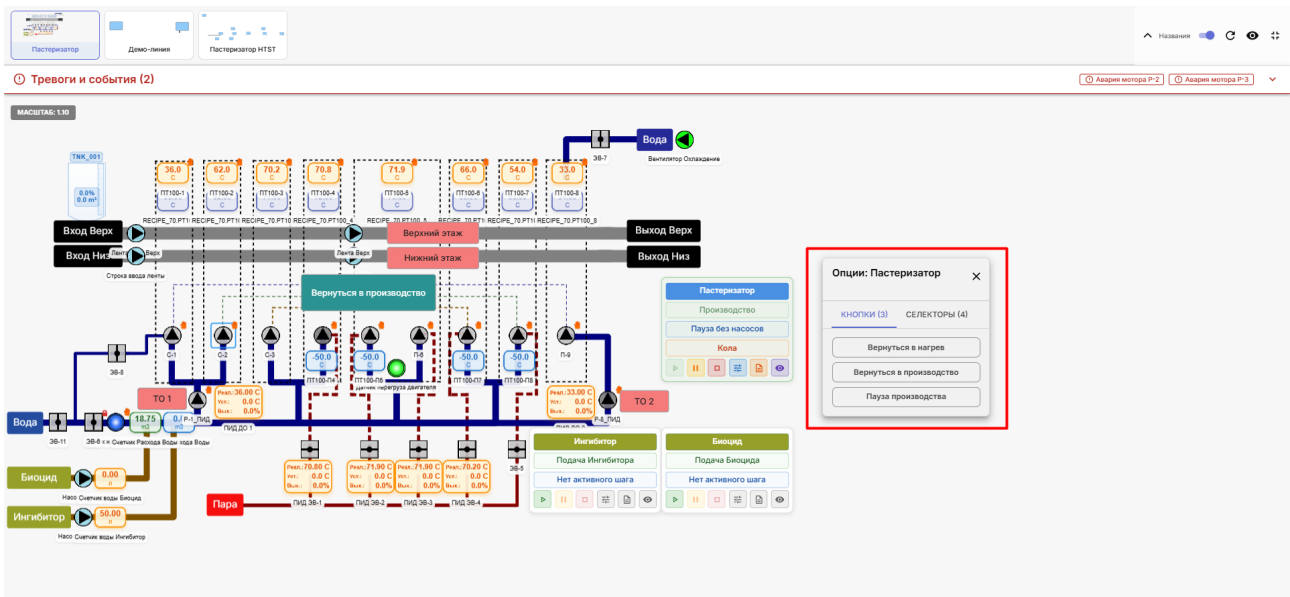
| Поле | Описание |
|--------------------|-------------------------------|
| Название | Имя опции (+ переводы) |
| Последовательность | Привязка к последовательности |
| Описание | Текстовое описание |

На схеме (Schema Viewer):

- Опция отображается как кнопка
- Нажатие переключает опцию (вкл/выкл)
- Цвет кнопки меняется в зависимости от состояния

| ID | Имя | Описание | Тип | Участок | Активировано | Условия | Доступность | Статус | Действия |
|----|--------------------------|----------|------------------|--------------|----------------------|---------|-------------|-----------|------------|
| 1 | Пауза производства | - | Кнопка (триггер) | Пастеризатор | 31.03.2026, 13:43:31 | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 2 | Верхний этаж | - | Селектор | Пастеризатор | - | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 3 | Нижний этаж | - | Селектор | Пастеризатор | - | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 4 | Вернуться в производство | - | Кнопка (триггер) | Пастеризатор | 09.04.2026, 14:01:30 | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 5 | Вернуться в нагрев | - | Кнопка (триггер) | Пастеризатор | 24.03.2026, 15:44:42 | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 6 | ТО 1 | - | Селектор | Пастеризатор | - | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |
| 7 | ТО 2 | - | Селектор | Пастеризатор | - | 1 | Доступна | Неактивна | ⏻ ⏪ ⏩ ⏹ 🗑️ |

Управление опциями: список опций с названием, последовательностью, описанием, кнопками Edit/Delete.



Кнопка опции на схеме: прямоугольная кнопка с названием, зелёная — активна, серая — неактивна.

Техническая документация

API опций:

| Метод | Endpoint | Описание |
|--------|-----------------------------|--------------|
| GET | /api/sequence-options/ | Список опций |
| POST | /api/sequence-options/ | Создать |
| PUT | /api/sequence-options/{id}/ | Обновить |
| DELETE | /api/sequence-options/{id}/ | Удалить |

12. Импульсные модули (Pulse Modules)

Для пользователя

Импульсные модули генерируют циклические последовательности включений/выключений устройств (например, циклическая подача воды).

Структура:

| Уровень | Описание |
|-----------------|---|
| Модуль | Контейнер для шагов |
| Шаг | Один этап цикла |
| Фаза | Временной интервал внутри шага (вкл/выкл) |
| Устройство фазы | Какое устройство включается/выключается |

Управление импульсными модулями: иерархический список — модуль → шаги → фазы → устройства. Кнопки Add/Edit/Delete на каждом уровне.

Техническая документация

API импульсных модулей:

| Метод | Endpoint | Описание |
|-------|--------------------------------|-----------------|
| GET | /api/pulse-modules/ | Список модулей |
| POST | /api/pulse-modules/ | Создать |
| POST | /api/pulse-modules/{id}/start/ | Запустить |
| POST | /api/pulse-modules/{id}/stop/ | Остановить |
| GET | /api/pulse-module-steps/ | Шаги модуля |
| GET | /api/pulse-module-step-phases/ | Фазы шага |
| GET | /api/pulse-module-devices/ | Устройства фазы |

13. Управление партиями (Batch Management)

Для пользователя

Система управления партиями отслеживает производственные порции (batches) на протяжении всего процесса.

Элементы панели партий:

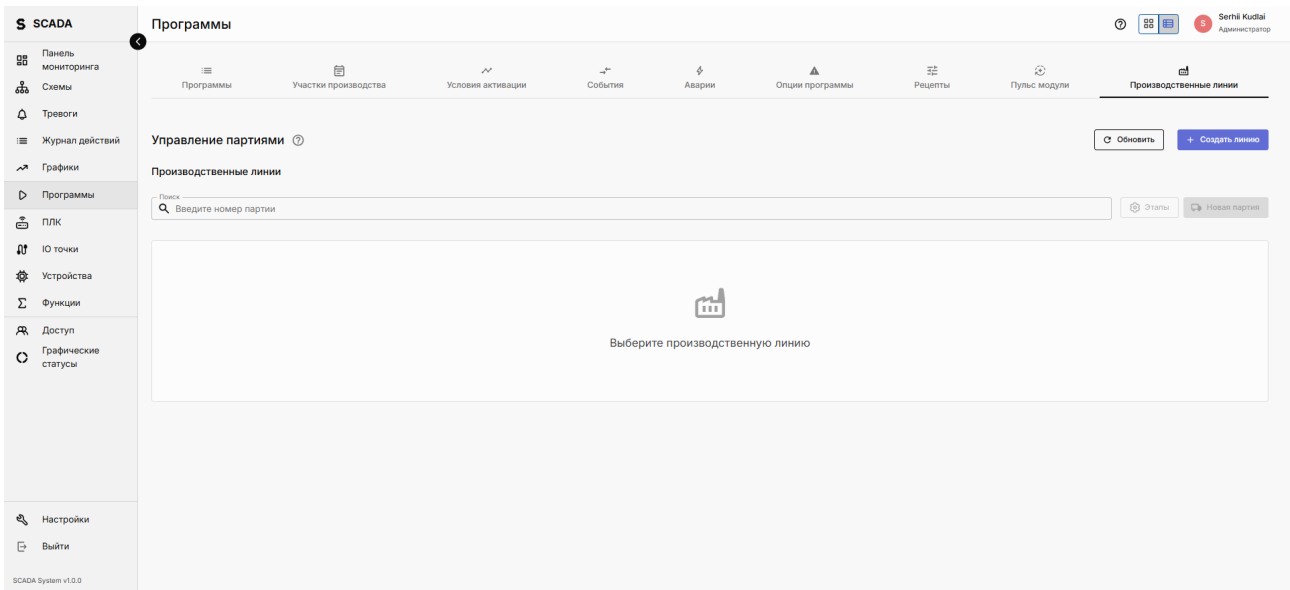
| Элемент | Описание |
|----------------|--|
| Создать партию | Кнопка создания новой партии |
| Таблица партий | Список с номером, рецептом, статусом, прогрессом |
| Статус партии | Pending / Queued / In Progress / Paused / Completed / Failed / Cancelled |
| Очередь | Порядок обработки партий |

Действия с партией:

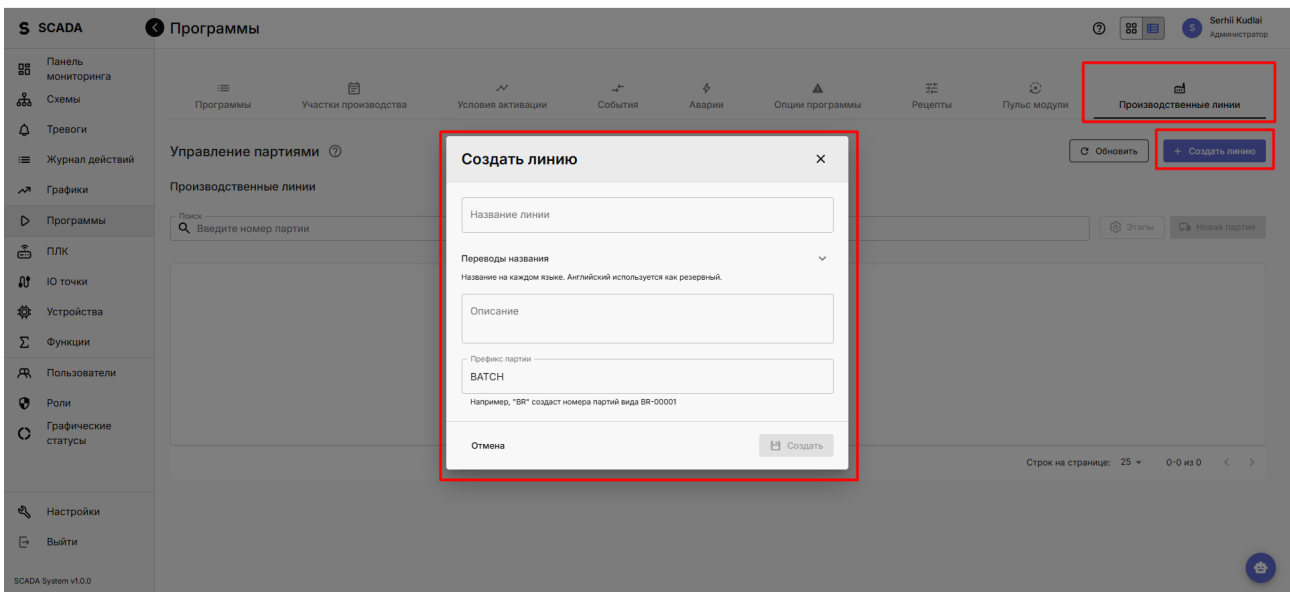
| Кнопка | Описание |
|--------|-----------------------------|
| Play | Запуск/возобновление партии |
| Pause | Приостановка |
| Stop | Полная остановка |
| Skip | Пропуск текущего шага |
| Cancel | Отмена партии |
| Delete | Удаление из списка |

Диалог создания партии:

| Поле | Описание |
|------------------------|--------------------------------|
| Производственная линия | Выбор линии |
| Рецепт | Выбор рецепта |
| Номер партии | Уникальный номер (BAT-001) |
| Авто-старт | Запустить сразу после создания |



Панель управления партиями: таблица с партиями, цветовая индикация статуса (зелёный — In Progress, серый — Pending, синий — Completed, красный — Failed), кнопки управления.



Диалог создания: выбор линии, рецепта, ввод номера, чекбокс «Авто-старт», кнопки «Создать» и «Отмена».

Техническая документация

API партий:

| Метод | Endpoint | Описание |
|-------|--|-----------------------------|
| GET | /api/batches/ | Все партии |
| POST | /api/production-lines/{id}/create_batch/ | Создать партию |
| POST | /api/batches/{id}/start/ | Запустить |
| POST | /api/batches/{id}/pause/ | Пауза |
| POST | /api/batches/{id}/resume/ | Продолжить |
| POST | /api/batches/{id}/cancel/ | Отменить |
| POST | /api/batches/{id}/transfer/ | Перевести на следующий этап |

| Метод | Endpoint | Описание |
|-------|----------------------------|--------------------------|
| GET | /api/batches/{id}/history/ | История этапов |
| GET | /api/batch-queue/ | Очередь партий |
| POST | /api/batch-queue/reorder/ | Изменить порядок очереди |

WebSocket: Статус партий

Подключение: `ws://ws/schema/{schemaid}/batches`

```
{
  "type": "batch_status",
  "batch_id": "uuid-...",
  "batch_number": "BAT-001",
  "status": "IN_PROGRESS",
  "current_stage": "Mixing",
  "progress": 65,
  "production_line_id": "uuid-...",
  "recipe_name": "Recipe A",
  "updated_at": "2026-04-07T12:00:00Z"
}
```

WebSocket команды (Client → Server):

```
{
  "type": "batch_command",
  "command": "start",
  "batch_id": "uuid-...",
  "production_line_id": "uuid-..."
}
```

14. Производственные линии

Для пользователя

Производственная линия объединяет несколько последовательностей в единый производственный маршрут с этапами.

Элементы:

| Поле | Описание |
|----------------|--|
| Название | Имя линии (+ переводы) |
| Этапы | Список этапов (Production Line Stages), каждый привязан к последовательности |
| Порядок этапов | Определяет маршрут прохождения партии |
| Статус | Активна / Неактивна |

Управление линиями: таблица с названием, количеством этапов, статусом, кнопками Edit/Delete. Детальный вид — список этапов с перетаскиванием для изменения порядка.

Техническая документация

API производственных линий:

| Метод | Endpoint | Описание |
|--------|--|----------------|
| GET | /api/production-lines/ | Список линий |
| POST | /api/production-lines/ | Создать |
| PATCH | /api/production-lines/{id}/ | Обновить |
| DELETE | /api/production-lines/{id}/ | Удалить |
| POST | /api/production-lines/{id}/activate/ | Активировать |
| POST | /api/production-lines/{id}/deactivate/ | Деактивировать |
| GET | /api/production-line-stages/ | Все этапы |

| Метод | Endpoint | Описание |
|--------|-----------------------------------|--------------|
| POST | /api/production-line-stages/ | Создать этап |
| PATCH | /api/production-line-stages/{id}/ | Обновить |
| DELETE | /api/production-line-stages/{id}/ | Удалить |

15. Тренды (Trends)

Для пользователя

Тренды — визуализация исторических данных с устройств в виде графиков.

Элементы экрана:

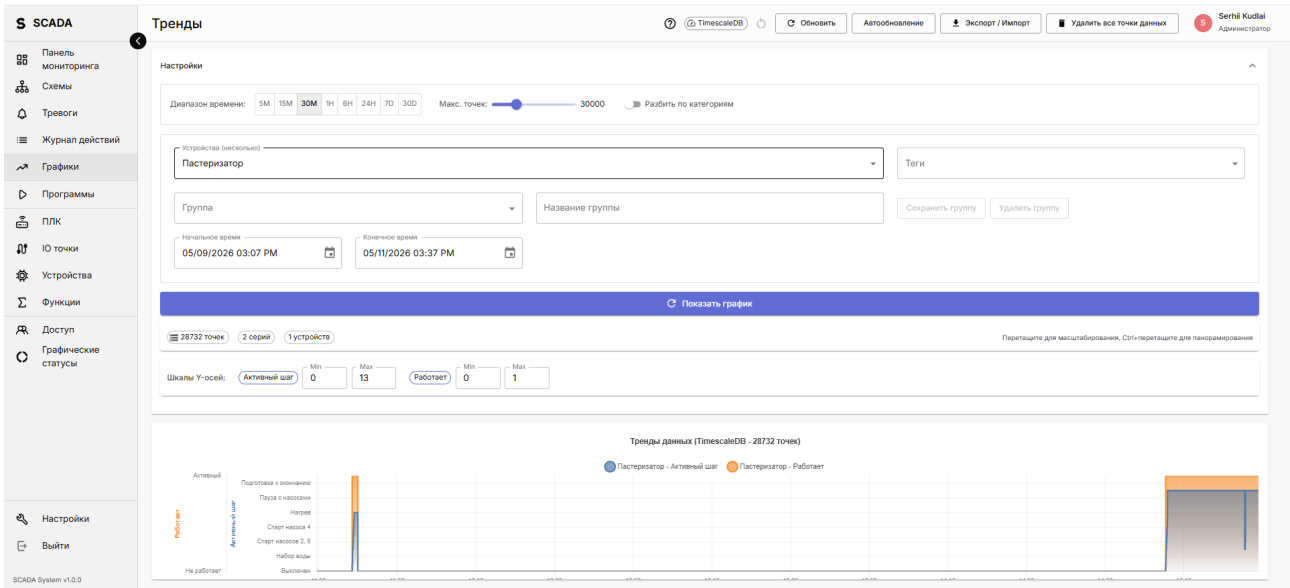
| Элемент | Описание |
|-------------------------|--|
| Выбор устройств | Мультиселект устройств/параметров для графика |
| Временной диапазон | Пресеты: 5 мин, 15 мин, 30 мин, 1 час, 6 часов, 24 часа, 7 дней, 30 дней |
| Пользовательский период | Выбор произвольного диапазона дат/времени |
| График | Линейный график Chart.js с масштабированием |
| Группы устройств | Сохранение/загрузка наборов устройств |
| Хранилище | Информация об использовании дискового пространства |

Взаимодействие с графиком:

| Действие | Описание |
|-----------------------|-------------------------------------|
| Drag (перетаскивание) | Выделение области для увеличения |
| Scroll (колесо мыши) | Масштабирование |
| ПКМ + перетаскивание | Панорамирование |
| Наведение | Всплывающая подсказка со значениями |
| Reset zoom | Сброс масштаба к исходному |

Кнопки:

| Кнопка | Описание |
|---------------------|---|
| Добавить устройства | Открывает мультиселект устройств |
| Сохранить группу | Сохраняет текущий набор устройств |
| Загрузить группу | Выбор сохранённой группы |
| Удалить группу | Удаление сохранённой группы |
| Скачать | Экспорт данных (CSV/JSON) |
| Скачать график | Экспорт изображения графика |
| Удалить данные | Удаление точек данных (суперпользователь) |
| Обновить | Принудительное обновление данных |



Полный экран трендов: сверху — выбор устройств (multi-select chips), выбор временного диапазона (кнопки 5m/15m/1h/6h/24h/7d/30d и Date Time Picker). По центру — линейный график с несколькими линиями разных цветов. Справа — легенда.

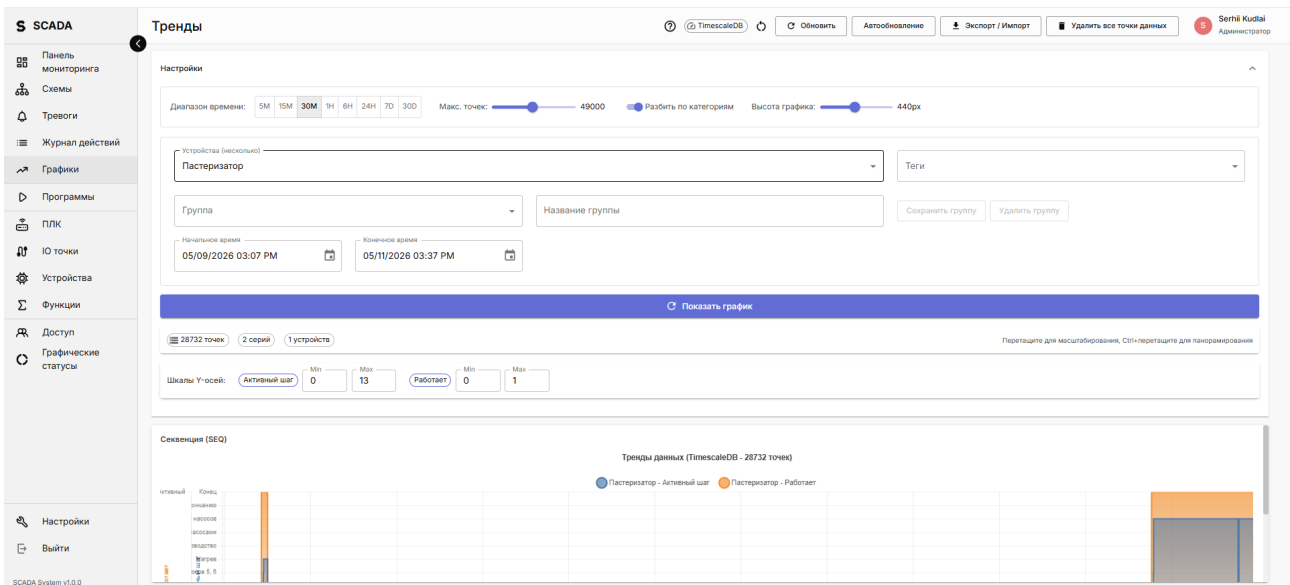
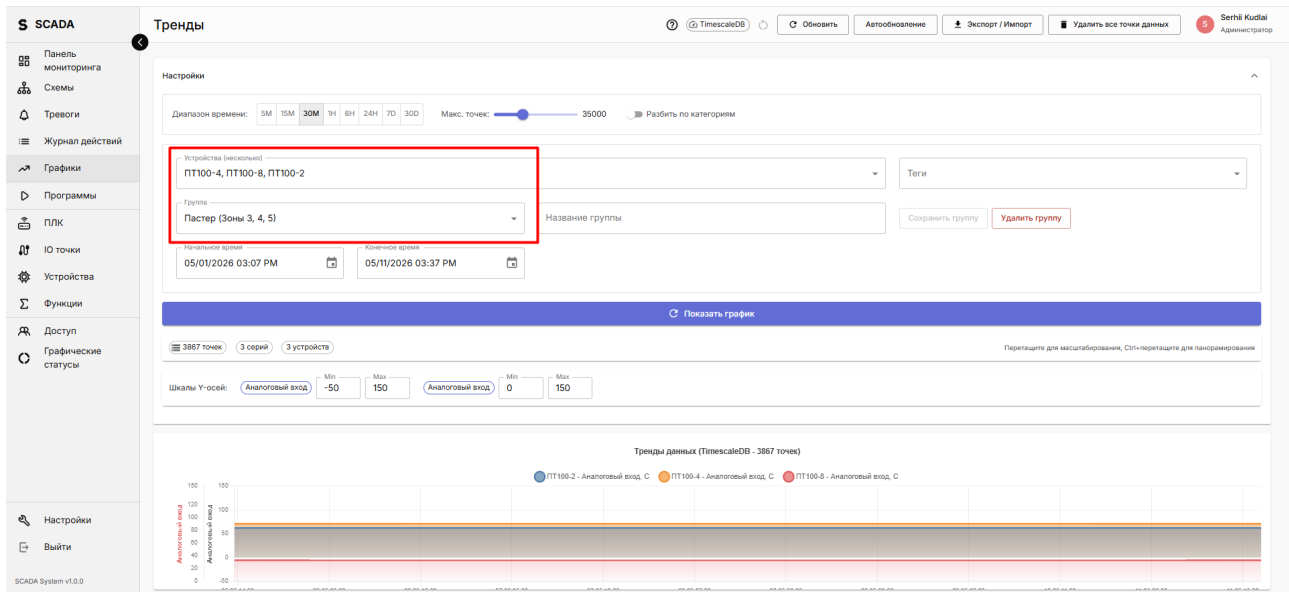


График после масштабирования: выделенная область увеличена, виден Reset zoom кнопка.



Dropdown сохранённых групп устройств: «Группа 1 — Температуры», «Группа 2 — Давления», кнопки Save/Load/Delete.

Техническая документация

API трендов (TimescaleDB):

| Метод | Endpoint | Описание | Параметры |
|-------|-----------------------------------|--------------------|---|
| GET | /api/graphs/graph/ | Данные графика | ?start_time=ISO&end_time=ISO&device_ids=id1,id2 |
| GET | /api/graphs/available-parameters/ | Доступные | — |
| GET | /api/graphs/timescale-status/ | Статус TimescaleDB | — |
| DELET | /api/graphs/delete-all/ | Удалить все данные | — |
| GET | /api/data-points/ | Точки данных | ?limit=100&offset=0 |

Данные экспорта:

| Метод | Endpoint | Описание |
|-------|---------------------------------|-------------------|
| POST | /api/project/export-graph-data/ | Экспорт графиков |
| POST | /api/project/import-graph-data/ | Импорт графиков |
| GET | /api/project/graph-data-stats/ | Статистика данных |

16. Аварийные сигналы (Alarms)

Для пользователя

Централизованный журнал всех аварийных событий в системе.

Элементы таблицы:

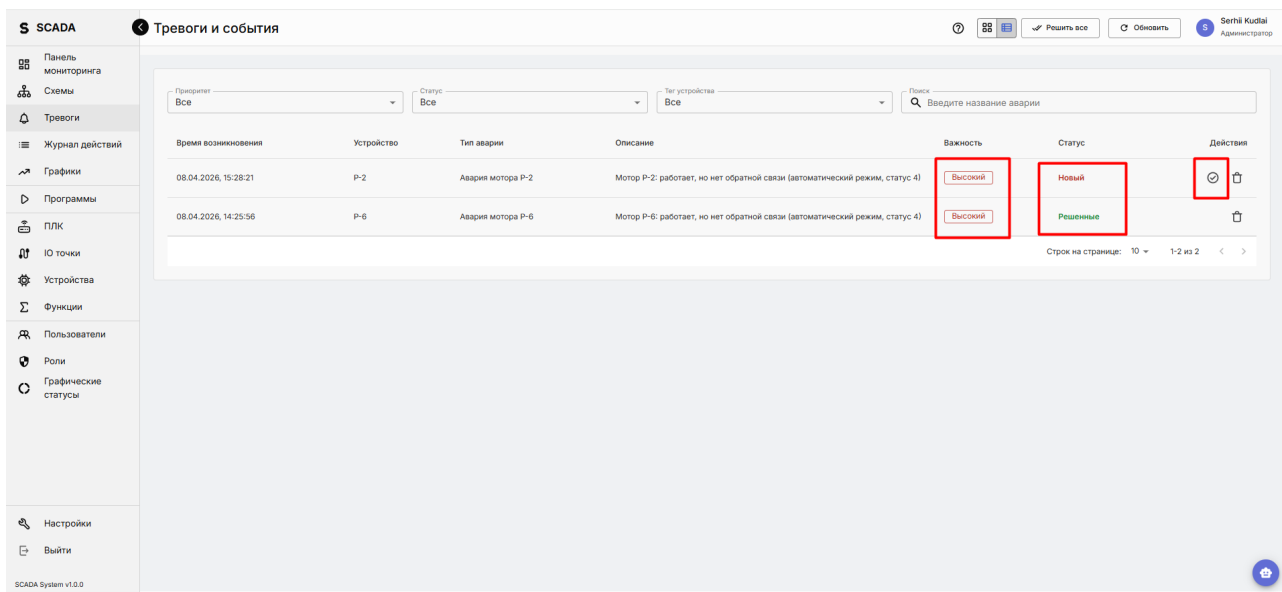
| Колонка | Описание |
|-------------|------------------------------------|
| Время | Дата и время возникновения |
| Серьёзность | Error (красный) / Warning (жёлтый) |
| Сообщение | Описание аварии |
| Устройство | Источник |
| Статус | Active / Acknowledged / Resolved |
| Действия | Кнопки управления |

Фильтры:

| Фильтр | Описание |
|----------------|--|
| По серьёзности | Warning / Error / All |
| По статусу | Active / Acknowledged / Resolved / All |
| По устройству | Выбор устройства |
| Поиск | Текстовый поиск |

Кнопки:

| Кнопка | Описание |
|-------------|----------------------------------|
| Acknowledge | Подтвердить (принять к сведению) |
| Resolve | Разрешить (проблема устранена) |
| Resolve All | Разрешить все аварии |
| Delete | Удалить запись |
| Refresh | Обновить список |



Страница аварий: таблица с цветовой индикацией серьёзности (красная полоса слева — Error, жёлтая — Warning). Фильтры сверху. Кнопки Acknowledge/Resolve для каждой строки.

Техническая документация

API аварий: См. раздел 10.

WebSocket: Подключение ws/accidents — получение новых аварий и обновлений статуса в реальном времени.

17. Математические функции (Math Functions)

Для пользователя

Математические функции позволяют создавать вычисляемые значения на основе данных с устройств и переменных рецепта.

Элементы экрана:

| Элемент | Описание |
|----------------|--------------------------|
| Список функций | Таблица/карточки функций |
| Кнопка «+» | Создать функцию |
| Поиск | По названию |

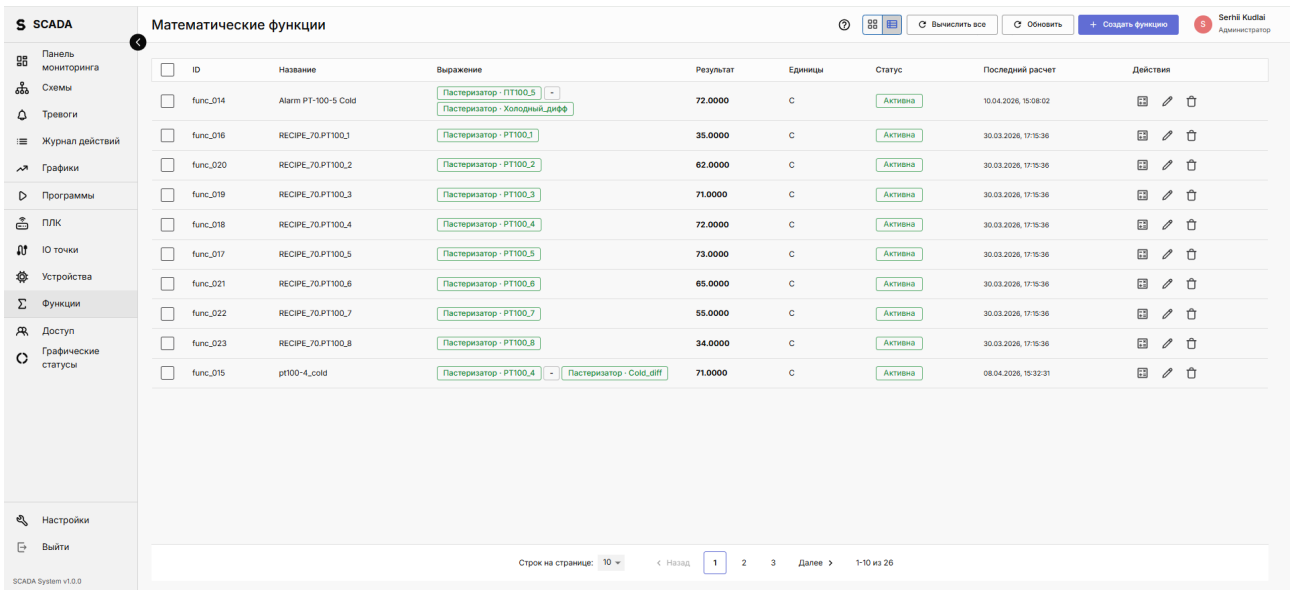
Редактор выражений (блочный):

Выражение строится из блоков, которые добавляются последовательно:

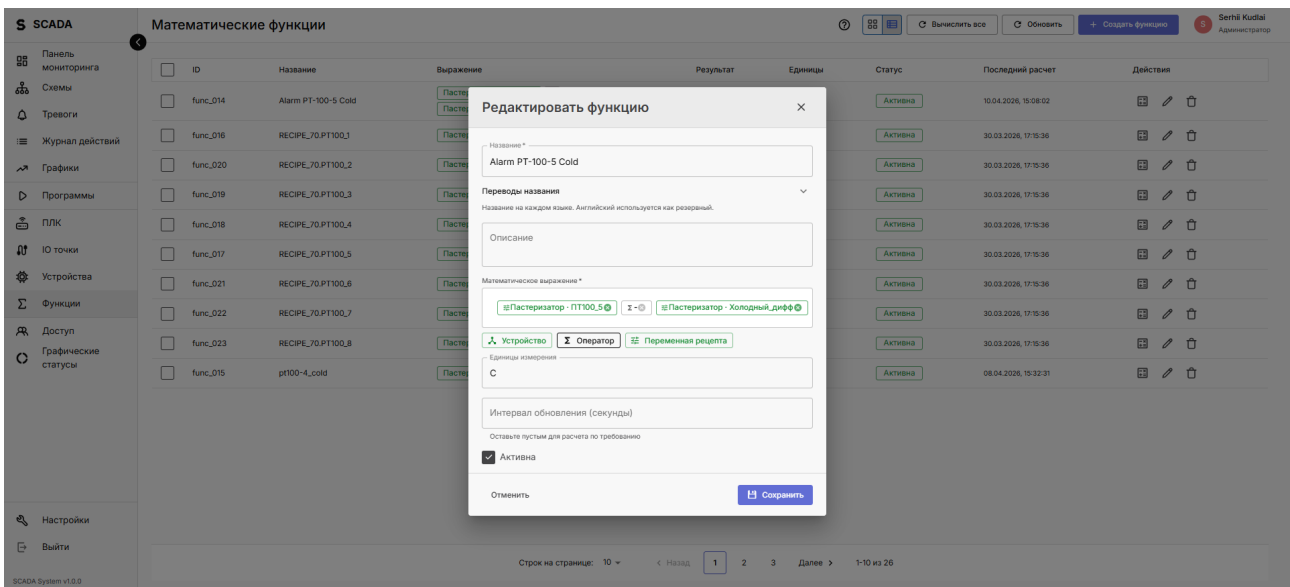
| Тип блока | Описание |
|--------------------|---------------------------------|
| Число | Константное числовое значение |
| Устройство | Текущее значение устройства |
| Переменная рецепта | Значение переменной |
| Оператор | + , - , × , ÷ , % , ^ |
| Скобки | (и) для группировки |
| Функция | sqrt, log10, round, floor, ceil |

Кнопки:

| Кнопка | Описание |
|---------------|---------------------------------|
| Добавить блок | Добавить новый блок в выражение |
| Удалить блок | Удалить блок из выражения |
| Проверить | Валидация синтаксиса |
| Тест | Вычислить с текущими значениями |
| Сохранить | Сохранение функции |



Список функций: таблица с колонками Name, Expression, Result, Unit, кнопки Edit/Delete/Test.



Редактор выражений: ряд блоков — [A1_001] [x] [2.5] [+][sqrt] [(] [A1_002] [)]. Кнопки добавления блоков снизу. Результат теста справа.

Техническая документация

API математических функций:

| Метод | Endpoint | Описание |
|--------|---|----------------|
| GET | /api/math-functions/ | Список функций |
| POST | /api/math-functions/ | Создать |
| PUT | /api/math-functions/{id}/ | Обновить |
| DELETE | /api/math-functions/{id}/ | Удалить |
| POST | /api/math-functions/{id}/calculate/ | Вычислить |
| POST | /api/math-functions/{id}/validate_expression/ | Валидация |
| POST | /api/math-functions/{id}/test_expression/ | Тест |
| POST | /api/math-functions/validate_all/ | Валидация всех |

WebSocket: Обновление мат. функций

Подключение: ws://ws/math-functions

```
{
  "type": "math_function_update",
  "function_id": 5,
  "name": "Total Flow",
  "expression": "ai_001 * 2.5 + sqrt(ai_002)",
  "result": 157.3,
  "units": "L/h",
  "last_calculated": "2026-04-07T12:00:00Z",
  "calculation_error": null,
  "is_active": true
}
```

18. Точки ввода-вывода (I/O Points)

Для пользователя

I/O Points определяют связь между физическими сигналами ПЛК и логическими устройствами SCADA.

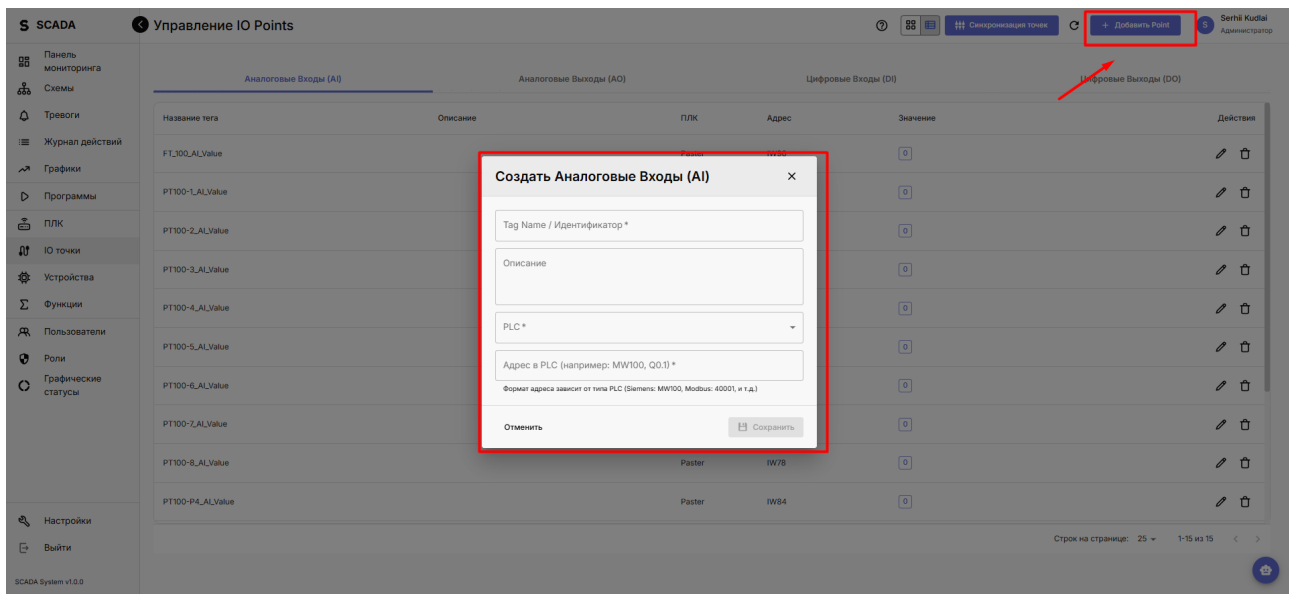
Вкладки:

| Вкладка | Описание |
|-----------------|-----------------------------------|
| Analog Inputs | Аналоговые входы (4-20 mA, 0-10V) |
| Analog Outputs | Аналоговые выходы |
| Digital Inputs | Дискретные входы |
| Digital Outputs | Дискретные выходы |

Форма I/O точки (IoPointDialog):

| Поле | Описание |
|-----------------|------------------------------------|
| ПЛК | Выбор контроллера |
| Номер/Адрес | Адрес в ПЛК (IW0, QW0, IO.0, Q0.0) |
| Устройство | Привязка к логическому устройству |
| Масштабирование | Min/Max (для аналоговых) |
| Тип данных | INT, REAL, BOOL |
| Единица | Инженерная единица |

Страница I/O Points: 4 вкладки (AI, AO, DI, DO), таблица с колонками PLC, Address, Device, Scale, Unit, кнопки Add/Edit/Delete.



Диалог создания I/O точки: dropdown ПЛК, поле адреса, dropdown устройства, поля Min/Max Scale, Unit.

Техническая документация

API I/O точек:

| Метод | Endpoint | Описание |
|-------|------------------------------|------------|
| GET | /api/points/analog-inputs/ | AI точки |
| POST | /api/points/analog-inputs/ | Создать AI |
| GET | /api/points/analog-outputs/ | АО точки |
| POST | /api/points/analog-outputs/ | Создать АО |
| GET | /api/points/digital-inputs/ | DI точки |
| POST | /api/points/digital-inputs/ | Создать DI |
| GET | /api/points/digital-outputs/ | DO точки |
| POST | /api/points/digital-outputs/ | Создать DO |

Для всех типов: стандартные CRUD операции (GET, POST, PUT, PATCH, DELETE) + /validate/ и /test/.

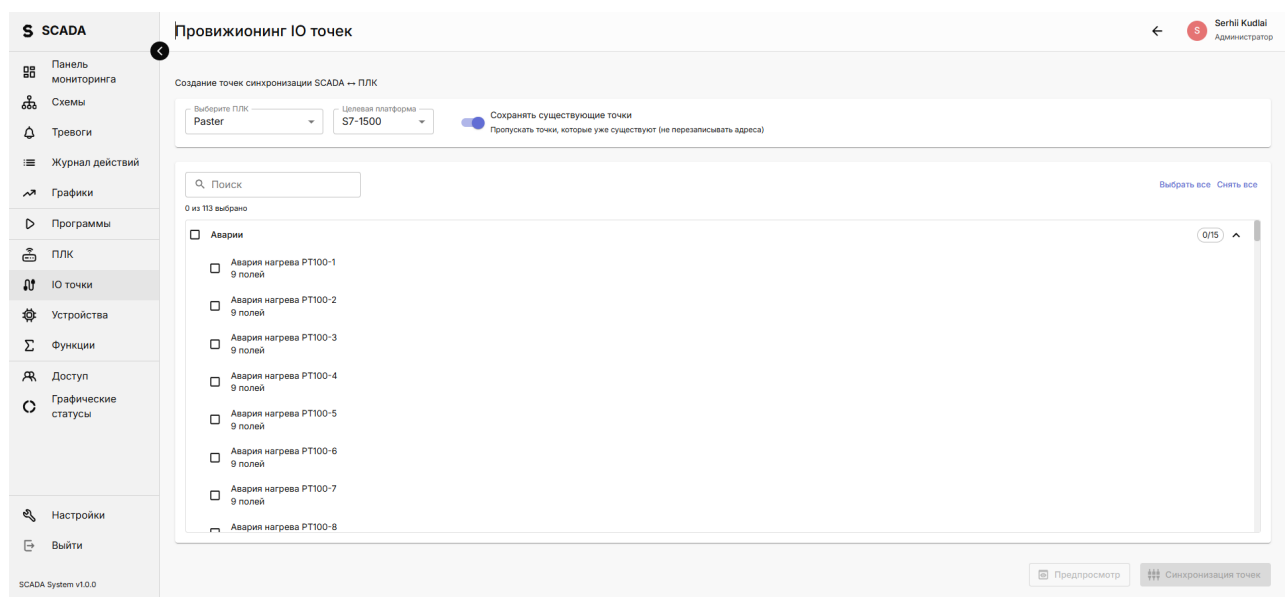
19. Провизионирование I/O

Для пользователя

Массовое создание I/O точек на основе конфигурации устройств.

Элементы экрана:

| Элемент | Описание |
|------------------|--|
| Выбор ПЛК | Целевой контроллер |
| Тип ПЛК | S7-1200 или S7-1500 |
| Сохранить адреса | Чекбокс — не перезаписывать существующие адреса |
| Список сущностей | Чекбоксы для каждого типа (моторы, клапаны, PID, AI, AO, DI, DO, COS, счётчики, таймеры, танки, последовательности, события, аварии) |
| Превью | Предварительный просмотр результата |
| Провизионировать | Запуск создания точек |



Страница провизионирования: dropdown ПЛК, radio S7-1200/S7-1500, список сущностей с чекбоксами, кнопки Preview и Provision.

Техническая документация

API провизионирования:

| Метод | Endpoint | Описание |
|-------|---|--------------------|
| POST | /api/plc-export/provision-io-points/preview/ | Превью |
| POST | /api/plc-export/provision-io-points/ | Выполнить |
| GET | /api/plc-export/provision-io-points/entities/ | Доступные сущности |

20. Графические статусы

Для пользователя

Графические статусы — шаблоны иконок для отображения состояния устройств на мнемосхемах.

Элементы:

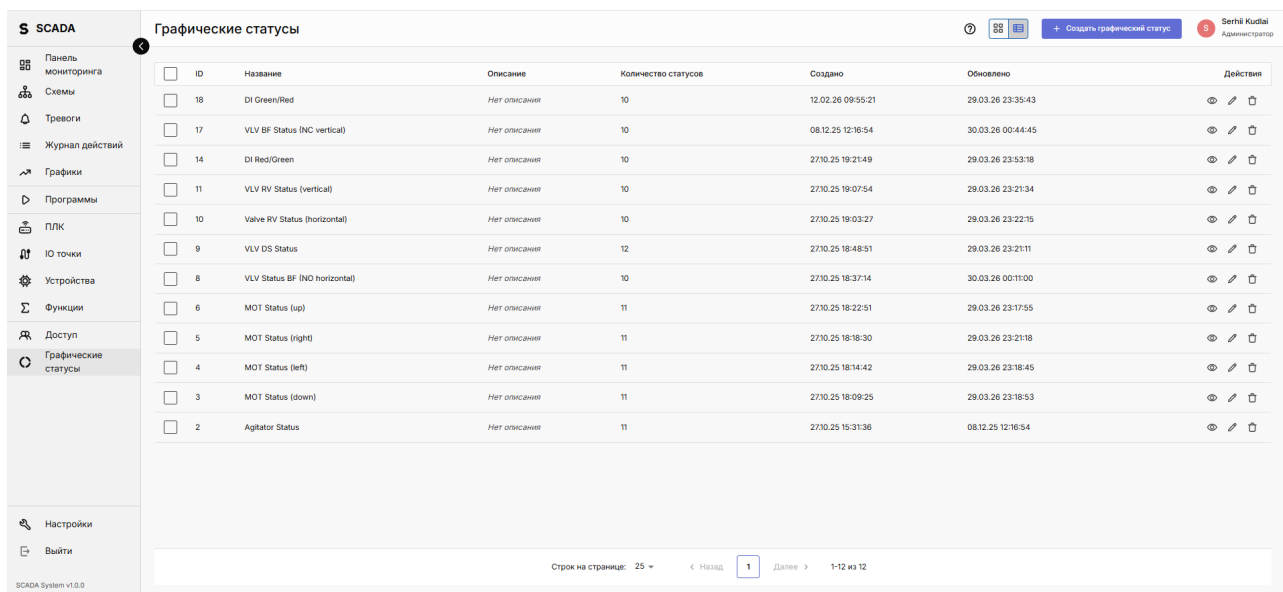
| Элемент | Описание |
|-----------------|------------------|
| Список шаблонов | Карточки/таблица |
| Кнопка «+» | Создать шаблон |

Мастер создания:

- Шаг 1: Имя, описание, тип устройства
- Шаг 2: Статусы и иконки
- Код статуса → Иконка (URL или загрузка)
- Галерея доступных иконок
- Загрузка своих иконок

| Кнопка | Описание |
|----------------------|-----------------------------|
| Добавить статус | Новая строка «код → иконка» |
| Удалить статус | Удалить строку |
| Загрузить иконку | Upload файла иконки |
| Галерея | Выбор из ранее загруженных |
| Предыдущий/Следующий | Навигация по мастеру |
| Сохранить | Сохранение шаблона |

Список графических статусов: карточки с превью иконок, название шаблона, тип устройства.



Мастер создания: слева — список статусов (код + превью иконки), справа — галерея иконок для выбора.

Техническая документация

API графических статусов:

| Метод | Endpoint | Описание |
|--------|--------------------------------------|------------------------------|
| GET | /api/graphical-statuses/ | Список шаблонов |
| POST | /api/graphical-statuses/ | Создать |
| PUT | /api/graphical-statuses/{id}/ | Обновить |
| DELETE | /api/graphical-statuses/{id}/ | Удалить |
| POST | /api/graphical-statuses/upload-icon/ | Загрузить иконку (multipart) |
| GET | /api/graphical-statuses/list-icons/ | Список иконок |

21. Конфигуратор оборудования (Hardware Configurator)

Для пользователя

Визуальный редактор аппаратной конфигурации ПЛК для развёртывания в TIA Portal.

Вкладки:

| Вкладка | Описание |
|-------------------|---------------------------------------|
| Rack View | Визуальный редактор стойки с модулями |
| PROFINET Topology | Сетевая топология устройств |
| Address Table | Таблица адресов I/O |

Редактор стойки (Rack Editor):

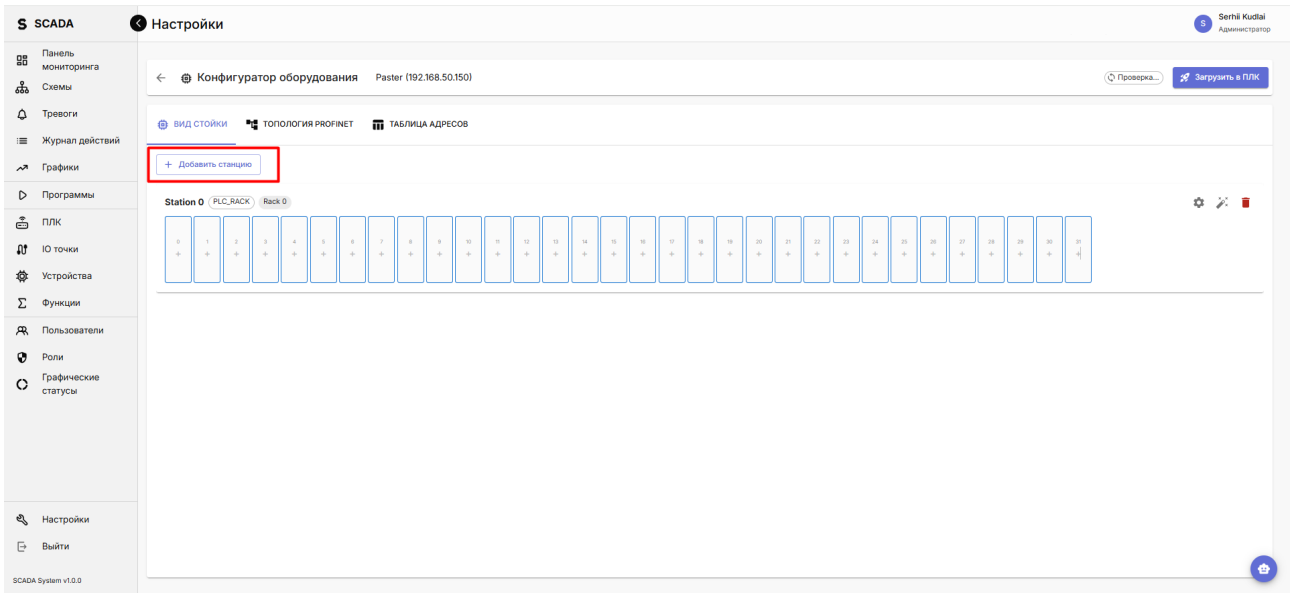
| Элемент | Описание |
|-----------------|--|
| Слоты стойки | Визуальное представление модулей в стойке |
| Каталог модулей | Боковая панель с доступными модулями (drag-drop) |
| Панель свойств | Конфигурация выбранного модуля |
| Добавить модуль | Перетаскивание из каталога |
| Удалить модуль | Удаление из слота |

Кнопки панели инструментов:

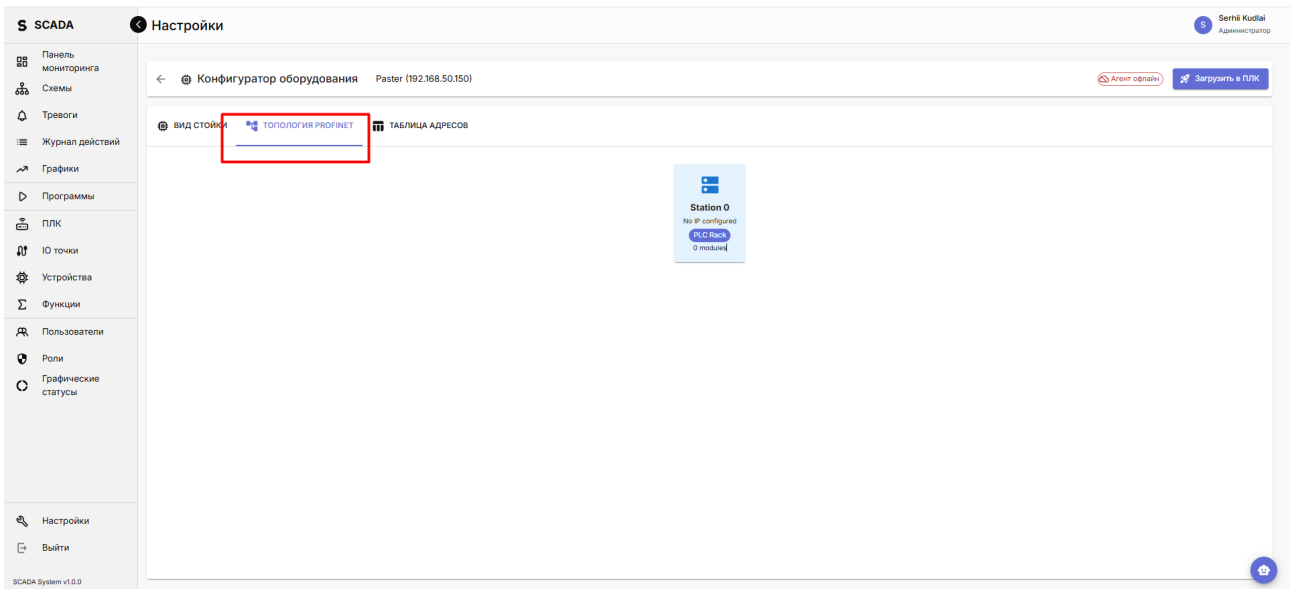
| Кнопка | Описание |
|------------------|--|
| Deploy | Развернуть конфигурацию в TIA Portal |
| Download | Скачать конфигурацию |
| Import GSD | Импорт файла описания устройства (GSD/GSDML) |
| TIA Agent Status | Индикатор подключения к TIA Agent |
| Back | Возврат к списку ПЛК |

Диалоги:

| Диалог | Описание |
|---------------------|---|
| TiaDeployDialog | Развёртывание: выбор целевой системы, прогресс, статус |
| GsdImportDialog | Импорт GSD: загрузка файла, превью модулей, подтверждение |
| NetworkConfigDialog | Настройка сети: IP, маска, шлюз, PROFINET параметры |



Редактор стойки: визуальная стойка с модулями в слотах, каталог модулей слева (выдвижная панель), панель свойств справа, кнопка Deploy в тулбаре.



Топология PROFINET: сетевая диаграмма с узлами устройств и связями между ними.

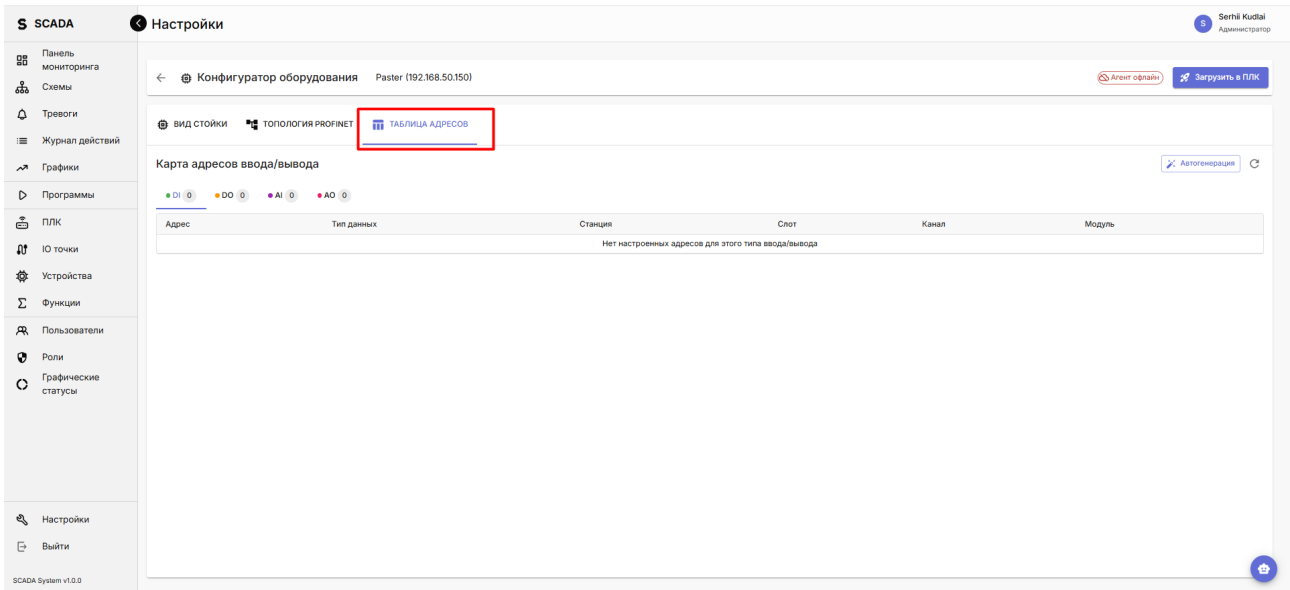
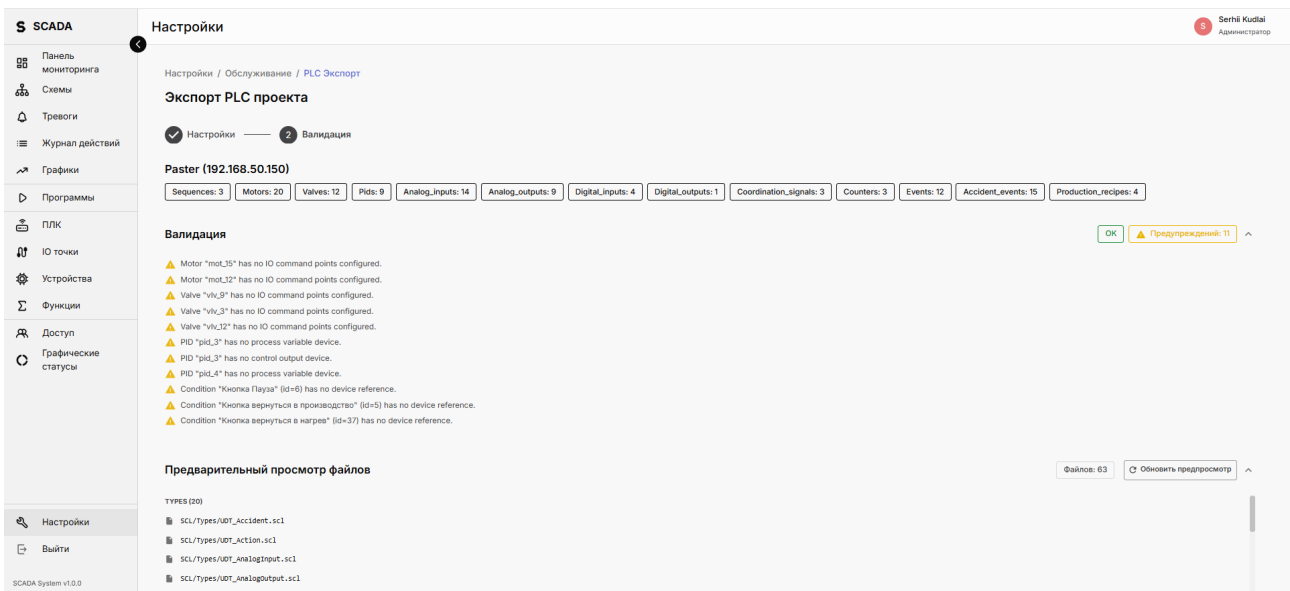


Таблица адресов: колонки Module, Slot, Channel, Address, Type (I/O), Device.



Диалог развёртывания: выбор TIA Agent, прогресс-бар, лог операций, кнопки Deploy/Cancel.

Техническая документация

API конфигуратора:

| Метод | Endpoint | Описание |
|-------|--|-------------------------|
| GET | /api/hw-catalog/ | Каталог модулей |
| POST | /api/hw-catalog/import-gsd/ | Импорт GSD (multipart) |
| GET | /api/hw-stations/ | Станции |
| POST | /api/hw-stations/ | Создать станцию |
| POST | /api/hw-stations/{id}/auto-assign-addresses/ | Авто-назначение адресов |
| GET | /api/hw-slots/ | Слоты |
| POST | /api/hw-slots/ | Создать слот |
| POST | /api/hw-slots/bulk-update/ | Массовое обновление |

| Метод | Endpoint | Описание |
|-------|--|-------------------------|
| GET | /api/hw-network/ | Сетевая конфигурация |
| POST | /api/tia/deploy/ | Развернуть в TIA Portal |
| GET | /api/tia-deploy-jobs/{id}/progress/ | Прогресс развёртывания |
| GET | /api/tia-agent-config/ | Конфигурации TIA Agent |
| GET | /api/tia-agent-config/{id}/check-connection/ | Тест соединения |

22. Управление пользователями

Для пользователя

Создание, редактирование и удаление учётных записей пользователей.

Таблица пользователей:

| Колонка | Описание |
|------------------|-------------------|
| Имя пользователя | Login |
| Полное имя | Имя и фамилия |
| Email | Электронная почта |
| Роли | Назначенные роли |
| Последний вход | Дата и время |
| Статус | Active / Inactive |
| Действия | Edit / Delete |

Форма пользователя:

| Поле | Описание |
|---------------|-------------------------------|
| Username | Уникальный логин |
| Email | Электронная почта |
| Пароль | Пароль (при создании) |
| Подтверждение | Повторный ввод пароля |
| Имя / Фамилия | Реальное имя |
| Аватар | Загрузка фото профиля |
| Роли | Мультиселект ролей |
| Статус | Переключатель Active/Inactive |

| Кнопка | Описание |
|----------------|---|
| Сохранить | Сохранение изменений |
| Отмена | Возврат без сохранения |
| Удалить | Удаление аккаунта (только в редактировании) |
| Сменить пароль | Изменить пароль пользователя |

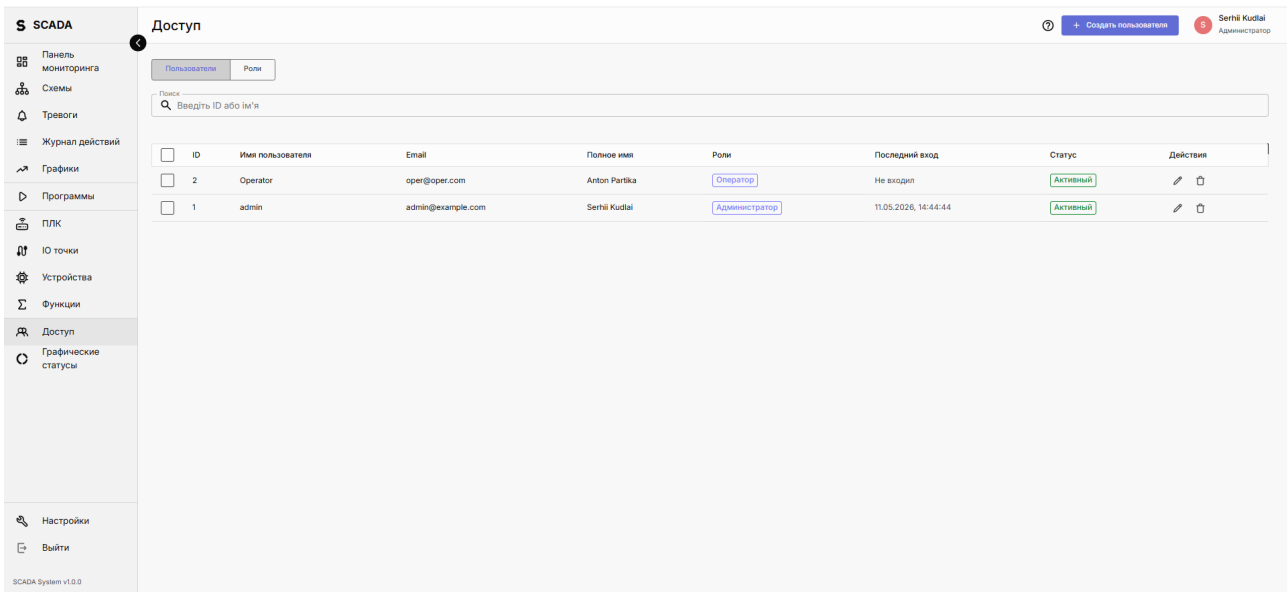
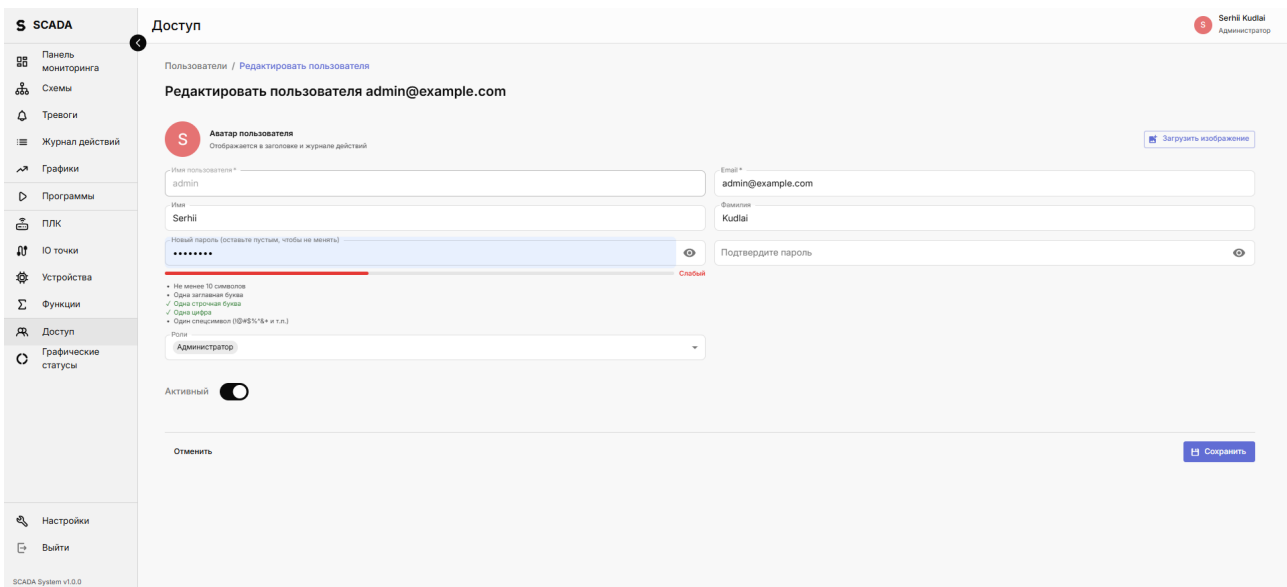


Таблица пользователей с колонками, аватарами, цветовыми бейджками ролей, индикатором статуса.



Форма создания/редактирования: поля username, email, password, role selector, avatar upload, status toggle.

Техническая документация

API пользователей:

| Метод | Endpoint | Описание |
|--------|--------------------------------|------------------------------|
| GET | /api/users/ | Список пользователей |
| POST | /api/users/ | Создать |
| GET | /api/users/{id}/ | Получить |
| PUT | /api/users/{id}/ | Обновить |
| DELETE | /api/users/{id}/ | Удалить |
| GET | /api/users/me/ | Текущий пользователь |
| POST | /api/users/{id}/set_password/ | Сменить пароль |
| POST | /api/users/{id}/upload-avatar/ | Загрузить аватар (multipart) |

| Метод | Endpoint | Описание |
|--------|------------------------|-------------------|
| GET | /api/users/{id}/roles/ | Роли пользователя |
| POST | /api/users/{id}/roles/ | Назначить роль |
| DELETE | /api/users/{id}/roles/ | Снять роль |

23. Управление ролями

Для пользователя

Ролевая модель управления доступом. Каждая роль определяет набор разрешений.

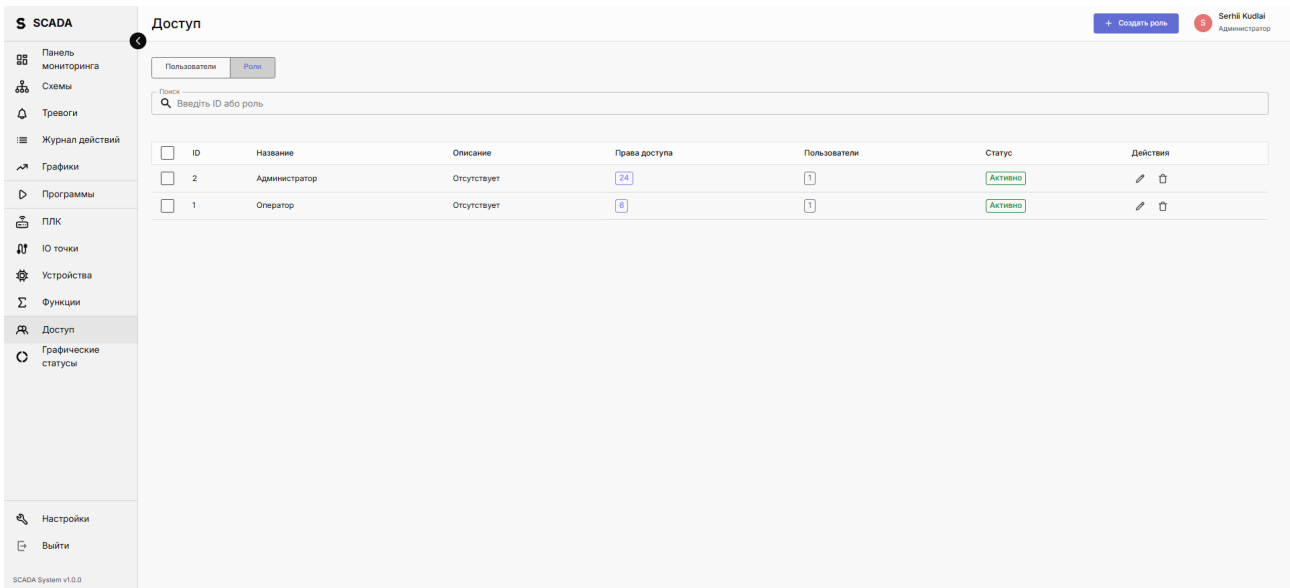
Таблица ролей:

| Колонка | Описание |
|---------------|---------------------------------------|
| Название | Имя роли |
| Описание | Текстовое описание |
| Пользователей | Количество пользователей с этой ролью |
| Действия | Edit / Delete |

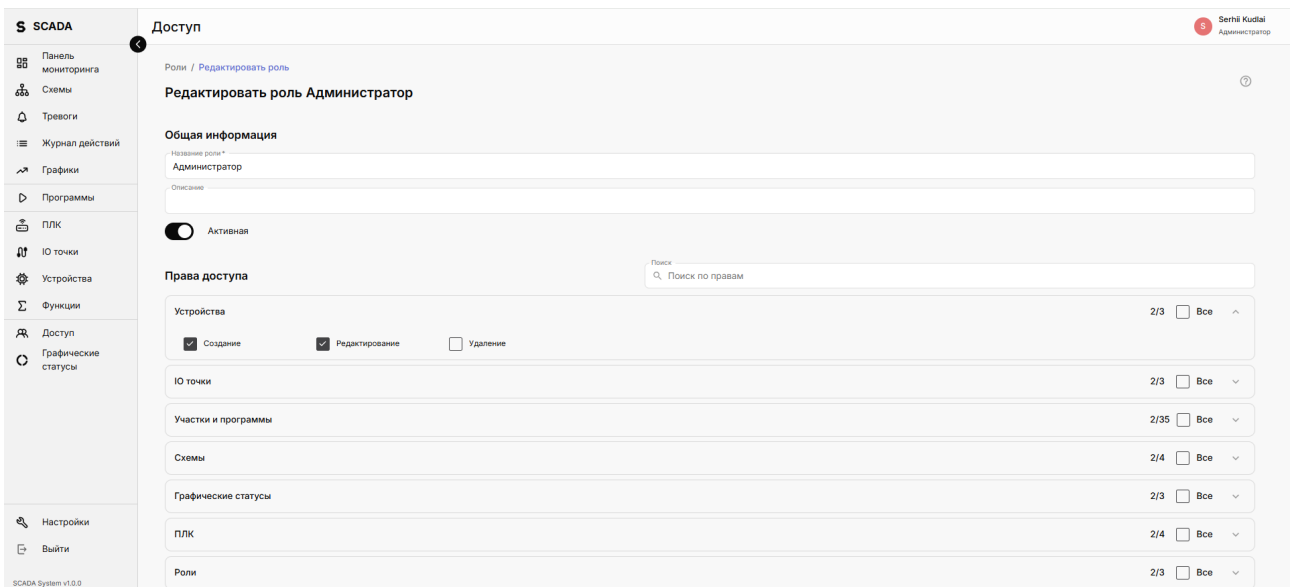
Матрица разрешений (в форме роли):

| Категория | Разрешения |
|---------------------|---|
| Устройства | Создание, Просмотр, Редактирование, Удаление |
| ПЛК | Создание, Просмотр, Редактирование, Удаление, Hardware Config |
| Последовательности | Создание, Просмотр, Редактирование, Удаление, Дублирование |
| Мнемосхемы | Создание, Просмотр, Редактирование, Удаление |
| I/O Points | Создание, Просмотр, Редактирование, Удаление |
| Пользователи | Создание, Просмотр, Редактирование, Удаление |
| Роли | Создание, Просмотр, Редактирование, Удаление |
| Аварии | Подтверждение, Разрешение, Удаление |
| Тренды | Просмотр, Экспорт, Удаление данных |
| Графические статусы | Создание, Просмотр, Редактирование, Удаление |
| Управление | Подключения, Лицензия, Обновления |

| Кнопка | Описание |
|--------------|------------------------|
| Select All | Выбрать все разрешения |
| Deselect All | Снять все разрешения |
| Сохранить | Сохранить роль |



Список ролей: таблица с названием, описанием, количеством пользователей, кнопками Edit/Delete.



Форма роли: имя, описание, матрица чекбоксов по категориям (Devices, PLCs, Sequences, Schemas...). Кнопки Select All/Deselect All.

Техническая документация

API ролей:

| Метод | Endpoint | Описание |
|--------|------------------------------|----------------------|
| GET | /api/roles/ | Список ролей |
| POST | /api/roles/ | Создать |
| GET | /api/roles/{id}/ | Получить |
| PATCH | /api/roles/{id}/ | Обновить |
| DELETE | /api/roles/{id}/ | Удалить |
| GET | /api/roles/{id}/permissions/ | Разрешения роли |
| GET | /api/permissions/categories/ | Категории разрешений |
| GET | /api/permissions/actions/ | Действия разрешений |

24. Журнал активности

Для пользователя

Аудит всех действий пользователей в системе.

Элементы экрана:

| Колонка | Описание |
|--------------|---|
| Время | Дата и время действия |
| Пользователь | Кто выполнил |
| Действие | Тип операции (create, update, delete, login...) |
| Описание | Детали действия |
| IP-адрес | С какого адреса |

Фильтры:

| Фильтр | Описание |
|-----------------|--------------------|
| По пользователю | Выбор пользователя |
| По действию | Тип операции |
| По дате | Диапазон дат |
| Поиск | Текстовый поиск |

Статистика:

- Общее количество действий
- Сводка по пользователям
- Сводка по типам действий

SCADA System v1.0.0

Журнал действий пользователей

Всего действий (7 дней) 54

Наиболее частые действия: Создание устройства: 11 | Удаление устройства: 1 | Устройство активировано: 1 | Изменение состояния устройства: 3 | Создание участка: 1

Поиск по описанию: Корневая: Все | Тип действия: Все | Период: 11.05.2025 - 11.05.2026

| Дата/Время | Пользователь | Тип действия | Описание | Объект | IP адрес |
|---------------------|--------------|--------------------------------|---|------------------------------|------------|
| 11.05.2026 15:34:46 | admin | Изменение состояния устройства | Изменено состояние C-2 на Ручной | Двигатель: C-2 | - |
| 11.05.2026 15:24:17 | admin | Обновление программы | Обновлена программа Производство | Программа: Производство | 172.18.0.1 |
| 11.05.2026 15:15:02 | admin | Изменение состояния устройства | Изменено состояние C-2 на Авто | Двигатель: C-2 | - |
| 11.05.2026 15:14:46 | admin | Изменение состояния устройства | Изменено состояние C-3 на Авто | Двигатель: C-3 | - |
| 11.05.2026 15:14:46 | Анонимный | Устройство активировано | Активировано устройство C-3 (состояние: Работает) | Двигатель: C-3 | - |
| 11.05.2026 15:14:23 | admin | Запуск участка | Запущен участок Пастеризатор | Участок: Пастеризатор | - |
| 11.05.2026 15:11:54 | admin | Обновление схемы | Обновлена схема Пастеризатор | Схема: Пастеризатор | 172.18.0.1 |
| 11.05.2026 12:25:08 | admin | Обновление участка | Обновлен участок Пастеризатор | Участок: Пастеризатор | 172.18.0.1 |
| 11.05.2026 11:49:20 | admin | Остановка участка | Остановлен участок Pasteurization HTST | Участок: Pasteurization HTST | - |
| 11.05.2026 11:49:00 | admin | Остановка участка | Остановлен участок Пастеризатор | Участок: Пастеризатор | - |
| 11.05.2026 11:47:34 | admin | Запуск участка | Запущен участок Пастеризатор | Участок: Пастеризатор | - |
| 08.05.2026 14:36:23 | admin | Обновление программы | Обновлена программа Пастеризатор | Программа: Пастеризатор | 172.18.0.1 |
| 08.05.2026 14:36:15 | admin | Обновление участка | Обновлен участок Pasteurization HTST | Участок: Pasteurization HTST | 172.18.0.1 |
| 08.05.2026 14:35:50 | admin | Смена шага | Шаг изменён на 2 для участка Pasteurization HTST | Участок: Pasteurization HTST | 172.18.0.1 |
| 08.05.2026 14:35:44 | admin | Запуск участка | Запущен участок Pasteurization HTST | Участок: Pasteurization HTST | - |
| 08.05.2026 14:35:43 | admin | Обновление участка | Обновлен участок Pasteurization HTST | Участок: Pasteurization HTST | 172.18.0.1 |

Строк на странице: 25 | Назад | 1 | 2 | 3 | 4 | 5 | Далее | 1-25 из 113

Журнал активности: таблица с фильтрами сверху, статистика справа (графики по типам действий и по пользователям). Пагинация снизу.

Техническая документация

API журнала активности:

| Метод | Endpoint | Описание | Параметры |
|--------|-------------------------------------|------------------|---|
| GET | /api/user-activity-logs/ | Список логов | ?user_id=X&action_type=X&limit=100&offset=0 |
| GET | /api/user-activity-logs/recent/ | За последние 24ч | — |
| GET | /api/user-activity-logs/statistics/ | Статистика | — |
| DELETE | /api/user-activity-logs/clear/ | Очистить (admin) | — |

Е

WebSocket: Журнал в реальном времени

Подключение: ws://ws/activity-log

```
{
  "type": "activity_log_entry",
  "id": 1234,
  "action_type": "device_update",
  "description": "Updated motor MOT_001 mode to manual",
  "user": {"id": 1, "username": "admin"},
  "content_object": {"type": "motor", "id": "mot_001"},
  "metadata": {},
  "created_at": "2026-04-07T12:00:00Z"
}
```

25. Настройки (Settings)

Для пользователя

Страница настроек содержит несколько вкладок:

Вкладка «Общие»

| Параметр | Описание |
|----------------|---|
| Тема | Выбор цветовой схемы (Blue, Green, Orange, Pink, Purple, Red) |
| Тёмный режим | Переключатель Light/Dark |
| Язык | Выбор языка интерфейса (EN, UK, RU, DE, FR, PL, AR...) |
| Версия дизайна | V1 / V2 |
| AI-ассистент | Включить/отключить AI-помощника |

Вкладка «Уведомления»

| Параметр | Описание |
|--------------------|-------------------------------------|
| Email-уведомления | Вкл/выкл |
| Push-уведомления | Вкл/выкл |
| SMS-уведомления | Вкл/выкл |
| Фильтр серьёзности | Минимальный уровень для уведомлений |

Вкладка «Подключения»

| Элемент | Описание |
|-------------------|---------------------------------------|
| Список TIA Agent | Конфигурации подключений к TIA Portal |
| Добавить | Новый TIA Agent (host, port) |
| Тест | Проверка соединения |
| Индикатор статуса | Зелёный/красный |

Вкладка «Лицензия»

| Поле | Описание |
|----------------|---------------------------------|
| Статус | Valid / Invalid / Trial |
| Тип | Lifetime / Trial / Subscription |
| Дней осталось | Обратный отсчёт |
| Компания | Название компании |
| Дата истечения | Дата окончания лицензии |
| ID установки | Уникальный идентификатор |

Вкладка «Обновления»

| Элемент | Описание |
|----------------------|--------------------------------|
| Текущая версия | Номер текущей версии |
| Проверить обновления | Кнопка проверки |
| Скачать обновление | Кнопка загрузки |
| Прогресс | Индикатор прогресса обновления |
| Авто-обновление | Переключатель |

Вкладка «Обслуживание»

| Элемент | Описание |
|----------------------|--|
| База данных | Операции обслуживания БД |
| Кэш | Очистка кэша |
| Логи | Управление логами |
| Здоровье системы | Проверки состояния |
| Бэкап/Восстановление | Резервное копирование и восстановление |
| Очистка диска | Удаление временных файлов |
| Smoke-тест | Проверка работоспособности |

The screenshot shows the 'Настройки' (Settings) page for SCADA System. The 'Общие' (General) tab is active, showing the following settings:

- Темная тема** (Dark Mode):
- AI Ассистент** (AI Assistant):
- Цветовая тема** (Color Theme): Blue Gray
- Версия дизайна** (Design Version): v1 — Current v2 — New Design
- Язык** (Language):

Информация о системе (System Information):

- Версия:** 1.0.0 (8782a77)
- Ветка:** danil/instruction-fix
- Последний коммит:** Refactor list page layouts with consistent structure (10.05.2026, 21:23:26)
- Разработчик:** SCADA Team
- Контакты:** support@scada-system.com

Buttons: [Development](#), [Сохранить](#) (Save)

Вкладка General: переключатель Dark Mode, Color Picker (6 цветов), dropdown языка, переключатель Design V1/V2, toggle AI Assistant.

SCADA Настройки Serhi Kudal
Администратор

Общие Уведомления **Подключения** Лицензии Обновления Обслуживание

Настройки API и MQTT

API URL: MQTT Бранд:

TIA Portal Агент

Настройка подключения к TIA Portal Agent для развёртывания проектов ПЛК

Имя агента: URL агента:

API ключ (необязательно): Версия TIA Portal:

Оставьте пустым для сохранения текущего ключа

Директория проекта на агенте: URL обратного вызова бранда:

URL, по которому агент обращается к бранду (для прогресса развёртывания)

Активен

Последний heartbeat: 29.03.2026, 14:25:48

SCADA System v1.0.0

Вкладка Connections: список TIA Agent с host/port, кнопки Add/Edit/Delete/Test, индикаторы статуса.

SCADA Настройки Serhi Kudal
Администратор

Общие Уведомления Подключения **Лицензии** Обновления Обслуживание

Статус лицензии

Тип лицензии: Пожизненная лицензия **Статус:** Активна

Действительна до: Пожизненная **Дней осталось:** Пожизненная

ID установки: 487e31f8e9992e212a79a288cc880aac6

Активировать новую лицензию

Введите лицензионный ключ

Введите лицензионный ключ

Для продолжения работы после окончания пробного периода необходимо ввести лицензионный ключ. Ключ можно получить в технической поддержке.

Информация о системе

Версия: 1.0.0 (67322877)

Ветка: danil/instruction-fix

Последний коммит: Refactor list page layouts with consistent structure (10.05.2026, 21:23:26)

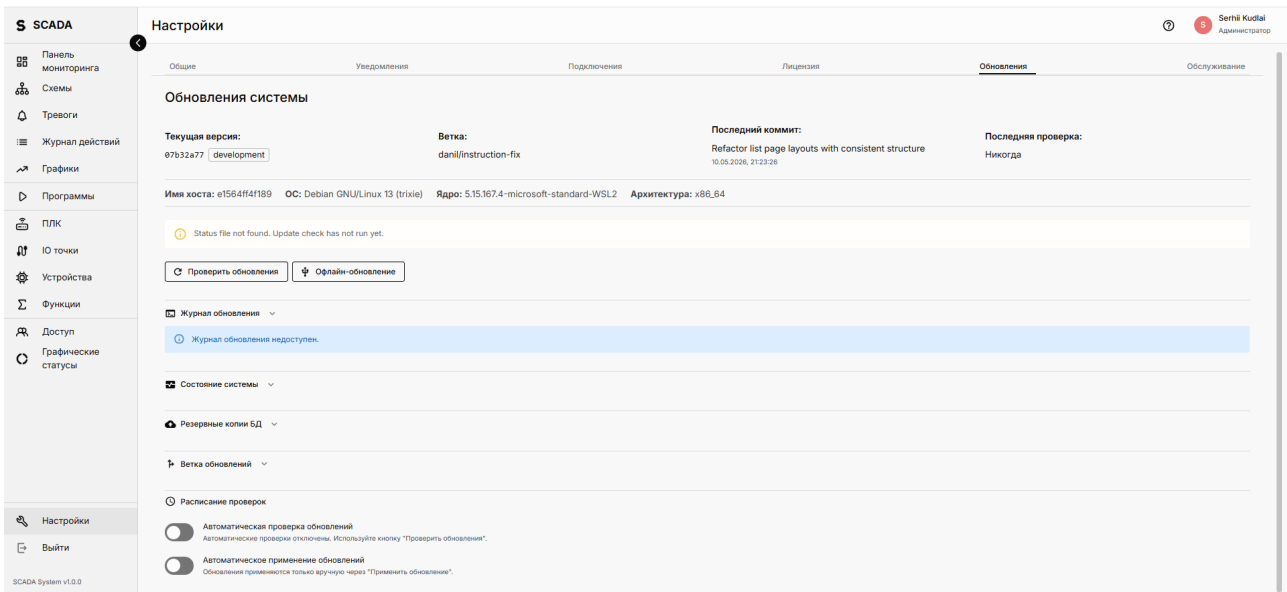
Разработчик: SCADA Team

Контакты: support@scada-system.com

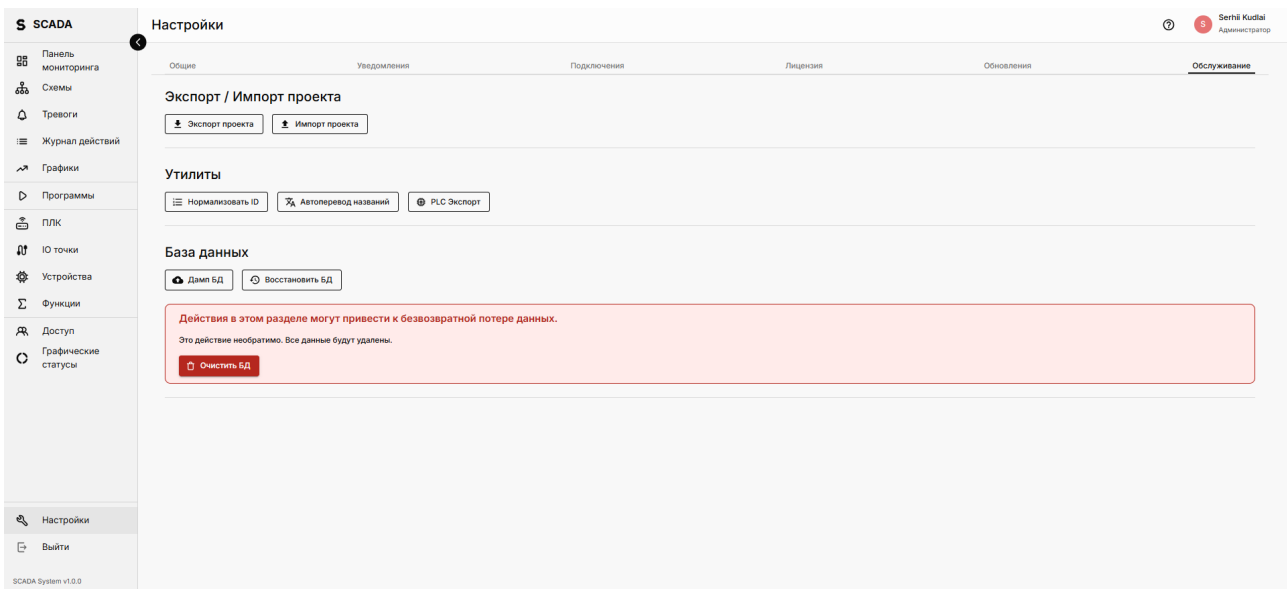
Development

SCADA System v1.0.0

Вкладка License: статус лицензии (зелёный бейдж), тип, дней осталось, компания, дата истечения.



Вкладка Updates: текущая версия, кнопка Check for Updates, прогресс-бар, логи обновления.



Вкладка Maintenance: кнопки Backup DB, Restore DB, Clear Cache, Disk Cleanup, Smoke Test с описаниями.

Техническая документация

API настроек пользователя:

| Метод | Endpoint | Описание |
|-------|----------------------|-------------------------|
| GET | /api/user/language/ | Получить язык |
| POST | /api/user/language/ | Установить язык |
| GET | /api/user/dark-mode/ | Получить тёмный режим |
| POST | /api/user/dark-mode/ | Установить тёмный режим |

API лицензии:

| Метод | Endpoint | Описание |
|-------|------------------------|-----------------|
| GET | /api/license/status/ | Статус лицензии |
| POST | /api/license/activate/ | Активация |
| POST | /api/license/generate/ | Генерация ключа |

API обновлений:

| Метод | Endpoint | Описание |
|-------|--|--------------------------|
| GET | /api/system/update-status/ | Статус обновления |
| POST | /api/system/update-check/ | Проверка обновлений |
| POST | /api/system/update-apply/ | Применить обновление |
| GET | /api/system/update-progress/ | Прогресс |
| POST | /api/system/update-cancel/ | Отмена |
| POST | /api/system/update-rollback/ | Откат |
| GET | /api/system/update-history/ | История обновлений |
| GET | /api/system/update-backups/ | Резервные копии |
| POST | /api/system/update-backups/{filename}/restore/ | Восстановление из бэкапа |
| POST | /api/system/disk-cleanup/ | Очистка диска |
| GET | /api/system/smoke-test/ | Smoke-тест |
| GET | /api/system/diagnostics/ | Диагностика (скачивание) |

WebSocket: Уведомления об обновлениях

```
{
  "type": "system_update",
  "event": "update_progress",
  "progress_percent": 45,
  "current_step": "Applying migrations...",
  "status": "in_progress"
}
```

26. AI-ассистент

Для пользователя

Встроенный AI-помощник для управления SCADA через естественный язык.

Элементы:

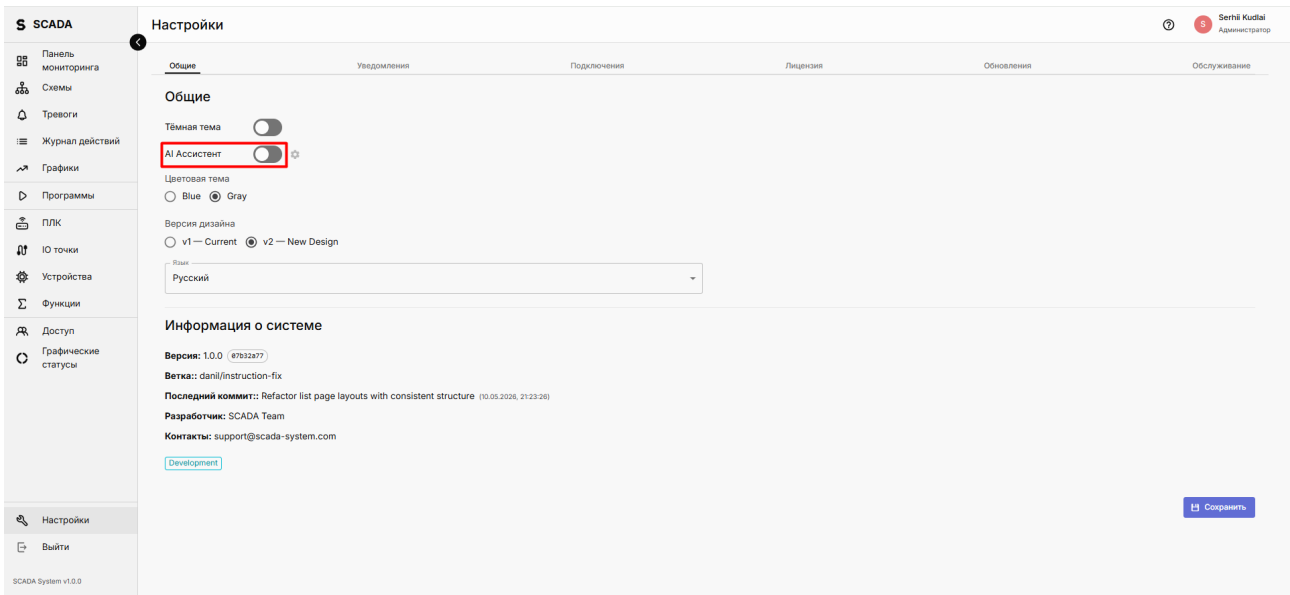
| Элемент | Описание |
|-------------------|--|
| Кнопка AI (FAB) | Плавающая кнопка в правом нижнем углу экрана |
| Панель чата | Выдвижная/перетаскиваемая панель |
| Список сессий | Боковая панель с историей бесед |
| Поле ввода | Текстовое поле для сообщений |
| Кнопка отправки | Отправить сообщение |
| Кнопка прикрепить | Загрузить изображение (P&ID чертёж) |
| Настройки | Конфигурация AI (модель, API ключ) |

Возможности AI:

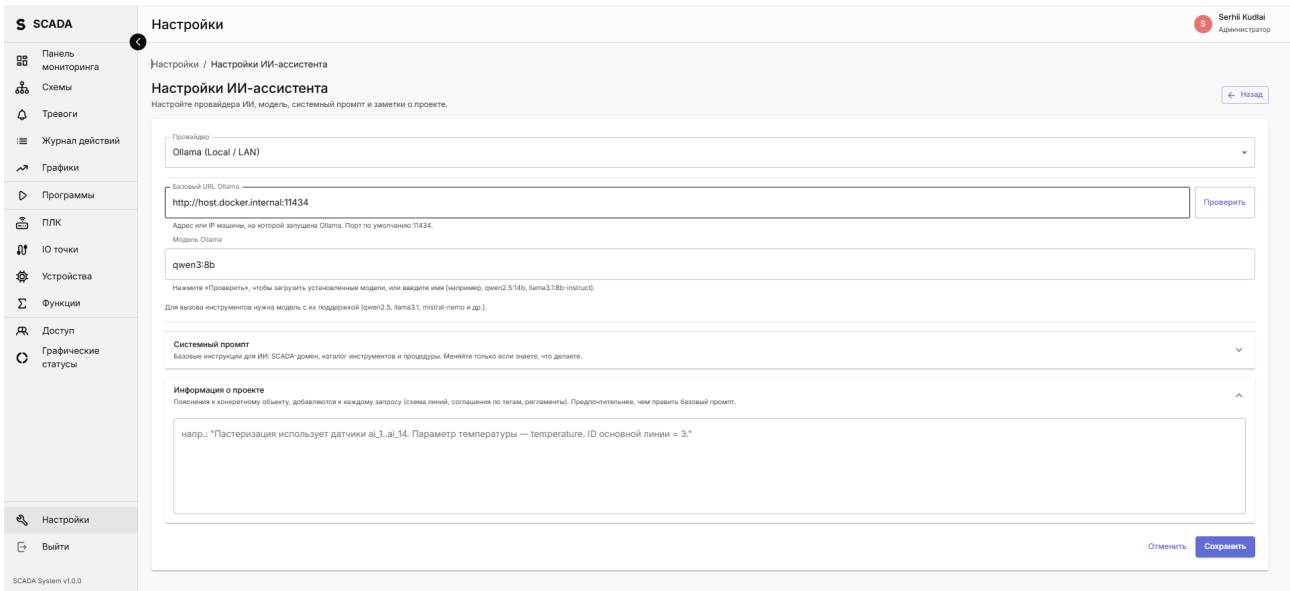
| Функция | Описание |
|-------------------------|---|
| Управление устройствами | «Включи мотор MOT_001» |
| Запрос информации | «Какие устройства сейчас в аварии?» |
| Создание элементов | «Создай новый аналоговый вход AI_010» |
| Анализ P&ID | Загрузка изображения → распознавание элементов → предложение размещения |
| Помощь | Ответы на вопросы о системе |

Настройки AI (AiSettingsDialog):

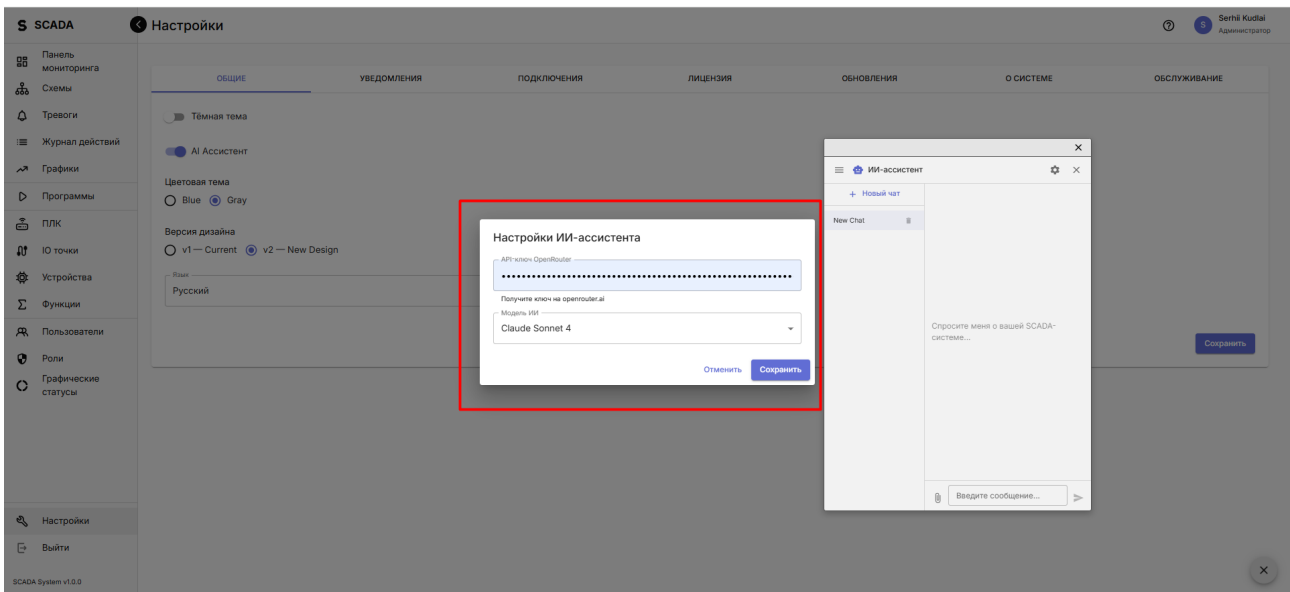
| Поле | Описание |
|-------------|--|
| API ключ | Ключ OpenRouter (шифруется AES/Fernet) |
| Модель | Выбор LLM модели |
| Температура | Креативность ответов (0-1) |
| Max tokens | Максимальная длина ответа |



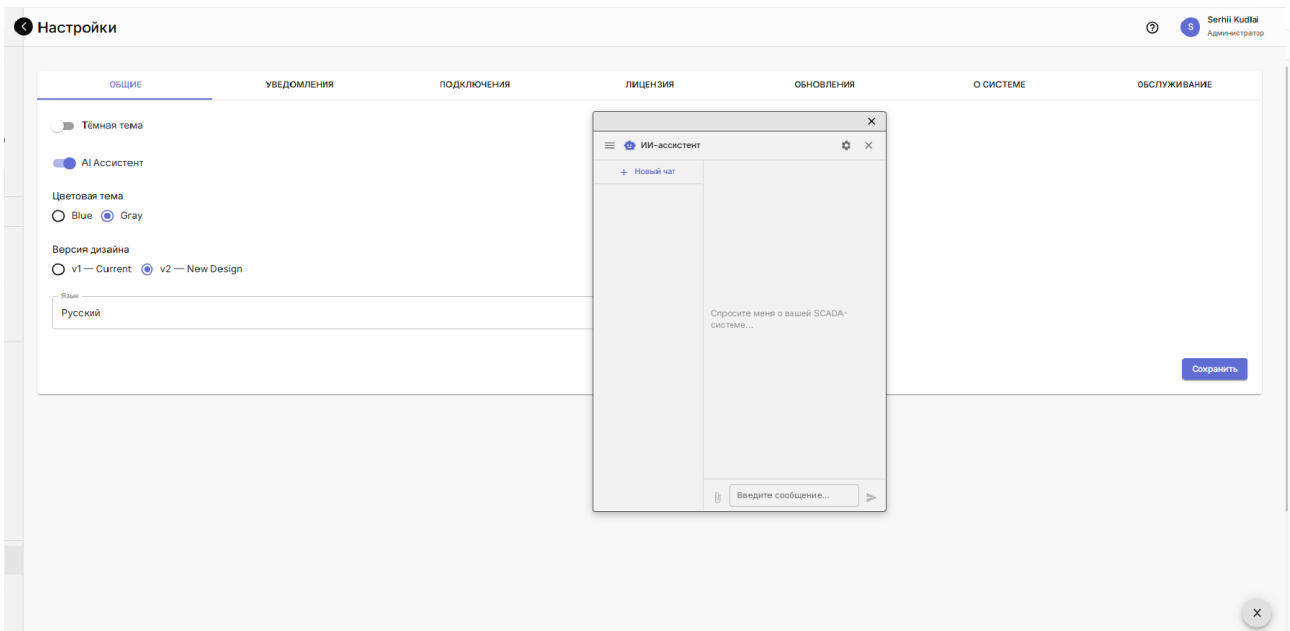
Плавающая кнопка AI в правом нижнем углу: круглая, с иконкой робота/чата.



Панель чата AI: слева — список сессий, справа — переписка (сообщения пользователя и ответы AI), внизу — поле ввода с кнопками Send и Attach.



Настройки AI: поле API Key (замаскировано), dropdown Model, slider Temperature, поле Max Tokens.



Анализ P&ID: загруженное изображение слева, справа — распознанные элементы (устройства, трубы) со структурированным предложением, кнопка «Применить».

Техническая документация

AI API (FastAPI microservice, BaseURL: `ai-api/v1`):

| Метод | Endpoint | Описание |
|--------|-------------------------|----------------------------------|
| POST | /sessions | Создать сессию |
| GET | /sessions | Список сессий |
| DELETE | /sessions/{id} | Удалить сессию |
| PATCH | /sessions/{id} | Переименовать |
| GET | /sessions/{id}/messages | Сообщения сессии |
| POST | /chat/{id}/messages | Отправить сообщение (SSE stream) |

| Метод | Endpoint | Описание |
|-------|-----------------|--------------------------------|
| POST | /images/analyze | Анализ изображения (multipart) |
| POST | /images/apply | Применить предложение |
| GET | /settings | Получить настройки |
| PATCH | /settings | Обновить настройки |
| GET | /health | Проверка здоровья |

SSE события (при stream=true):

| Event | Описание |
|-------------|--|
| token | Токен текста ответа (стриминг) |
| tool_call | AI вызывает инструмент (devicetools, schematools, query_tools) |
| tool_result | Результат вызова инструмента |
| error | Ошибка |
| done | Ответ завершён |

Формат SSE:

```

event: token
data: {"content": "Мотор "}

event: token
data: {"content": "MOT_001 "}

event: tool_call
data: {"name": "get_device_status", "arguments": {"device_id": "mot_001"}}

event: tool_result
data: {"result": {"status": "running", "value": 1}}

event: token
data: {"content": "сейчас работает."}

event: done
data: {}

```

27. Экспорт PLC-проекта

Для пользователя

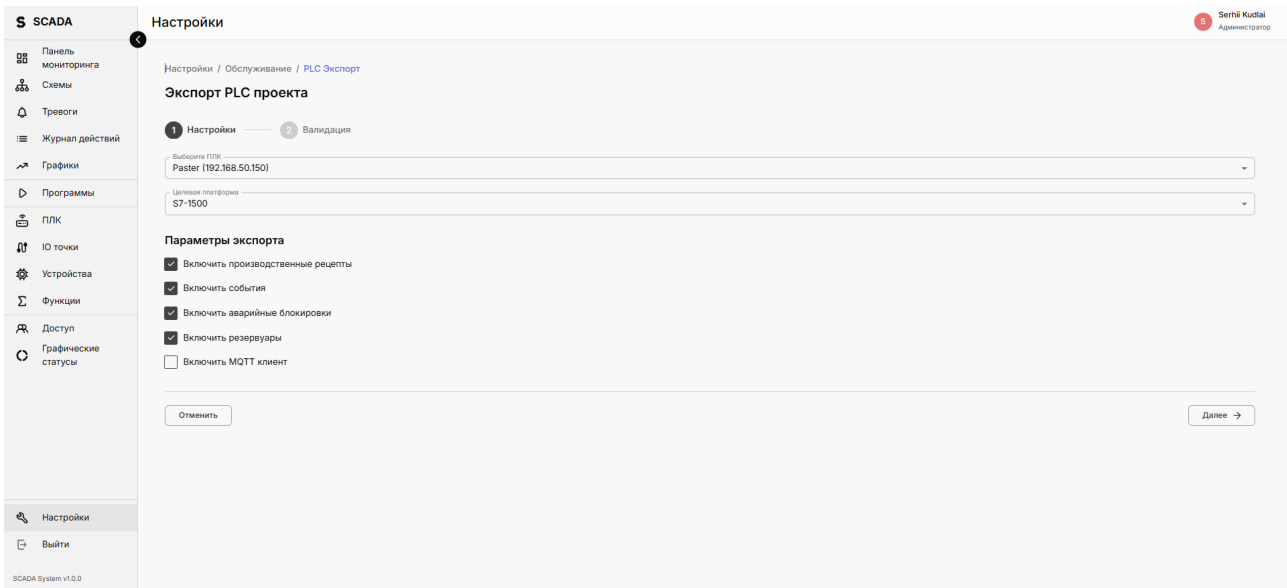
Генерация проекта для Siemens TIA Portal на основе конфигурации SCADA.

Диалог экспорта (PlcExportDialog):

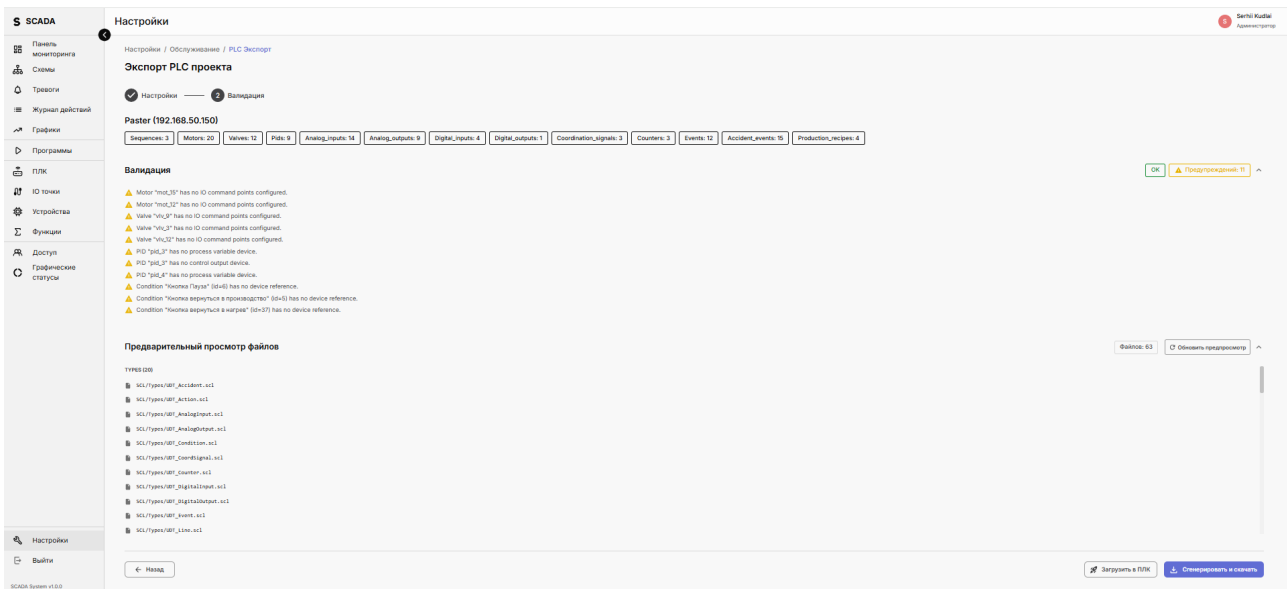
| Элемент | Описание |
|-------------------|--------------------------------------|
| Выбор ПЛК | Целевой контроллер |
| Тип ПЛК | S7-1200 / S7-1500 |
| Включить рецепты | Чекбокс — добавить рецепты |
| Включить события | Чекбокс — добавить события |
| Включить аварии | Чекбокс — добавить аварии |
| Включить танки | Чекбокс — добавить танки |
| Включить MQTT | Чекбокс — добавить MQTT коммуникацию |
| MQTT Broker IP | IP-адрес MQTT брокера |
| MQTT Topic Prefix | Префикс топиков |
| Превью | Предварительный просмотр кода SCL |
| Генерировать | Скачать ZIP-файл с проектом |
| Deploy to TIA | Развернуть через TIA Agent |

Состав генерируемого проекта:

- SCL код функциональных блоков (Motor FB, Valve FB, PID FB, AI FB, AO FB, DI FB, DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB (автомат состояний)
- Event/Accident FBs
- Data Blocks (Instance DBs, Recipe DBs)
- OB1 Main (вызов всех FB)
- XML Tag Tables (SimaticML)



Диалог экспорта: dropdown ПЛК, radio S7-1200/S7-1500, чекбоксы Include (Recipes, Events, Accidents, Tanks, MQTT), поля MQTT config, кнопки Preview/Generate/Deploy.



Превью SCL кода: слева — дерево файлов (FBs, DBs, Tag Tables), справа — содержимое SCL файла с подсветкой синтаксиса.

Техническая документация

API PLC Export:

| Метод | Endpoint | Описание |
|-------|------------------------------|---------------|
| GET | /api/plc-export/preview/ | Превью кода |
| POST | /api/plc-export/validate/ | Валидация |
| POST | /api/plc-export/generate/ | Генерация ZIP |
| GET | /api/plc-export/address-map/ | Карта адресов |

28. Экспорт и импорт проекта

Для пользователя

Полный экспорт/импорт проекта SCADA для переноса между установками.

Диалог экспорта (ExportProjectDialog):

| Элемент | Описание |
|-------------------|--|
| Чекбоксы объектов | Выбор типов для экспорта: Devices, Schemas, Sequences, Recipes, Events, Accidents, Options, Trends, Alarms, Users, Roles |
| Select All | Выбрать все |
| Включить графики | Экспортировать данные трендов |
| Поиск | Фильтрация объектов |
| Экспорт | Скачать файл экспорта |
| Прогресс | Индикатор прогресса |

Кнопки:

| Кнопка | Описание |
|-----------------|-------------------------------------|
| Экспорт проекта | Начать экспорт выбранных объектов |
| Импорт проекта | Загрузить файл → превью → применить |
| Бэкап БД | Создать резервную копию базы данных |
| Восстановить БД | Восстановить из резервной копии |

The screenshot shows the 'Export Project' dialog in the SCADA System. The left sidebar contains navigation icons for various system components. The main content area is titled 'Настройки' (Settings) and 'Экспорт проекта' (Export Project). It features a checkbox labeled 'Включить точки графиков' (Include graph points) which is checked, with a note below it: 'Это может значительно увеличить размер файла' (This can significantly increase the file size). Below this is a section 'Выберите объекты для экспорта' (Select objects for export) with a 'Выбрано: 247' (Selected: 247) indicator and a 'Снять все' (Deselect all) button. The list of objects includes: Схемы (3/3), Последовательности (4/4), Двигатели (21/21), Клапаны (15/15), ПИД-регуляторы (11/11), Преобразователи частоты (0/0), Счётчики (3/3), Таймеры (0/0), Аналоговые входы (17/17), and Аналоговые выходы (9/9).

Диалог экспорта: список типов с чекбоксами, кнопка Select All, toggle «Include Graphs», кнопка Export, прогресс-бар.

Техническая документация

API проекта:

| Метод | Endpoint | Описание |
|-------|---------------------------------|-------------------------------|
| GET | /api/project/export-models/ | Список экспортируемых моделей |
| POST | /api/project/export/ | Экспорт |
| POST | /api/project/import/preview/ | Превью импорта |
| POST | /api/project/import/ | Импорт |
| POST | /api/project/backup-db/ | Бэкап БД |
| POST | /api/project/restore-db/ | Восстановление |
| POST | /api/project/normalize-ids/ | Нормализация ID |
| POST | /api/project/clear-database/ | Очистка БД |
| POST | /api/project/export-graph-data/ | Экспорт графиков |
| POST | /api/project/import-graph-data/ | Импорт графиков |
| GET | /api/project/graph-data-stats/ | Статистика графиков |

29. Танки (Tanks)

Для пользователя

Танки — ёмкости с автоматическим расчётом объёма на основе данных сенсоров.

Параметры танка:

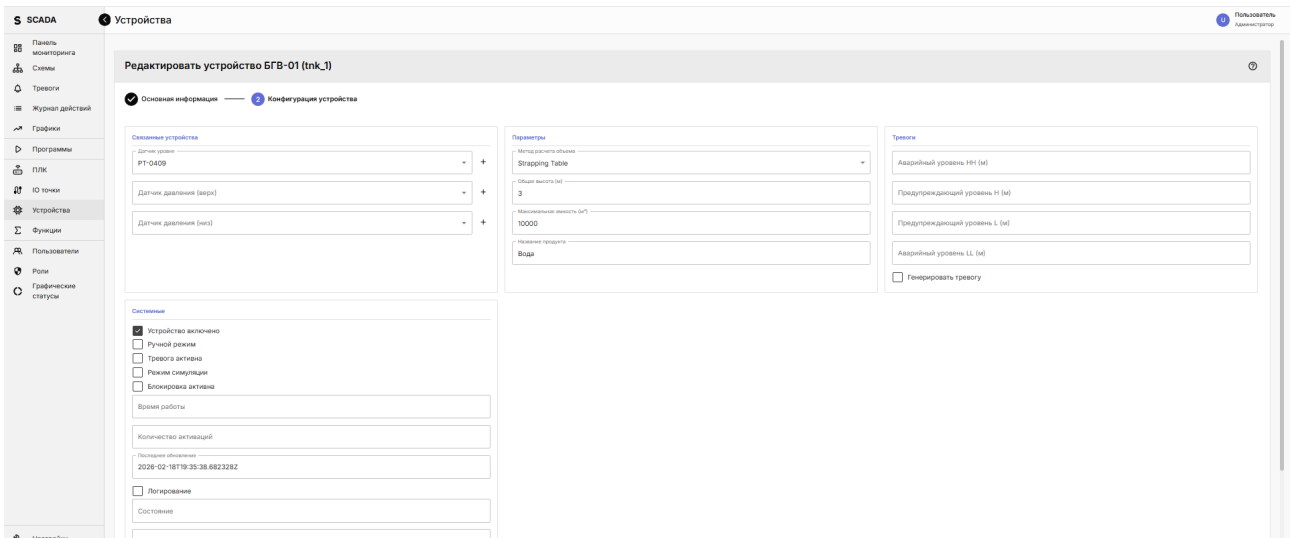
| Поле | Описание |
|----------------------|--|
| Сенсор уровня | Привязка к AI устройству (уровнемер) |
| Сенсор давления верх | Давление в верхней части (опционально) |
| Сенсор давления низ | Давление в нижней части (для ΔP) |
| Аварийные уровни | НН, Н, L, LL — пороги аварийных сигналов |

Методы расчёта объёма:

| Метод | Описание |
|-----------------|--|
| Strapping Table | Таблица градуировки (интерполяция sensor_value → volume) |
| Geometric | Геометрический (секции: цилиндр, конус, купол, тарелка) |

Конструктор танка на схеме:

| Элемент | Описание |
|--------------------------|---|
| Выбор устройства | Dropdown существующих TNK устройств |
| Тип нижней крышки | FLAT / DOME / DISH / CONE |
| Тип верхней крышки | FLAT / DOME / DISH |
| Ширина / Высота | Визуальные размеры |
| SVG Превью | Предварительный просмотр формы танка |
| Авто-размещение сенсоров | Сенсоры размещаются вдоль правой стенки |



Форма танка: привязка сенсоров (Level Sensor dropdown, Pressure Top, Pressure Bottom), пороги аварий (HH, H, L, LL — числовые поля).

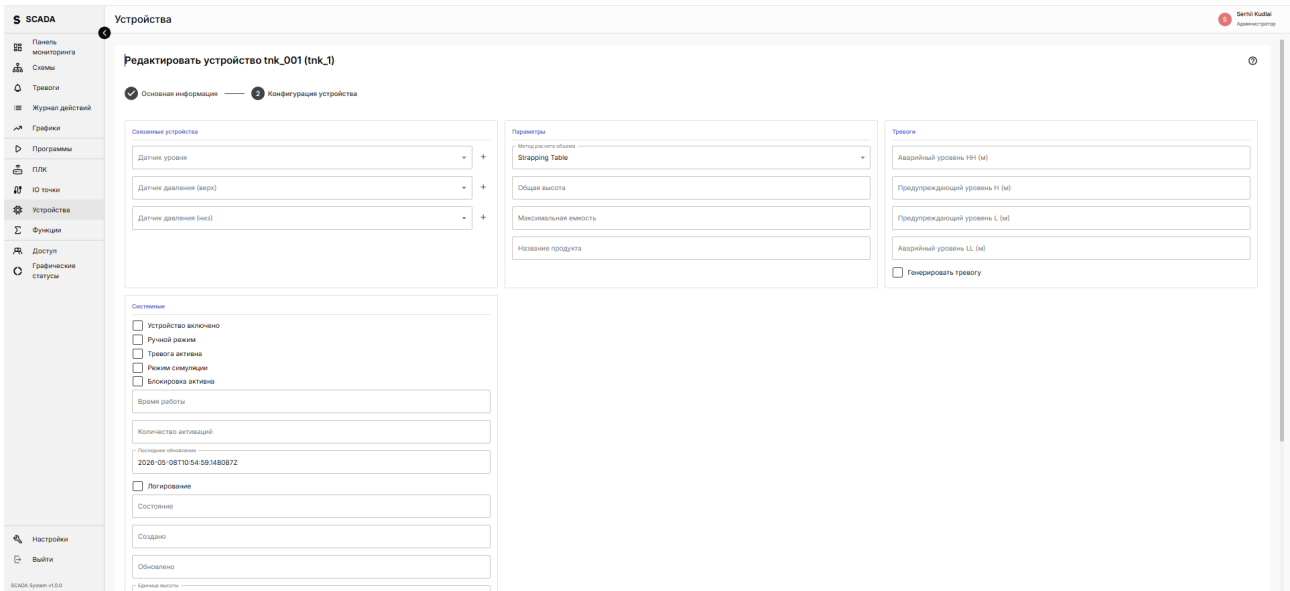
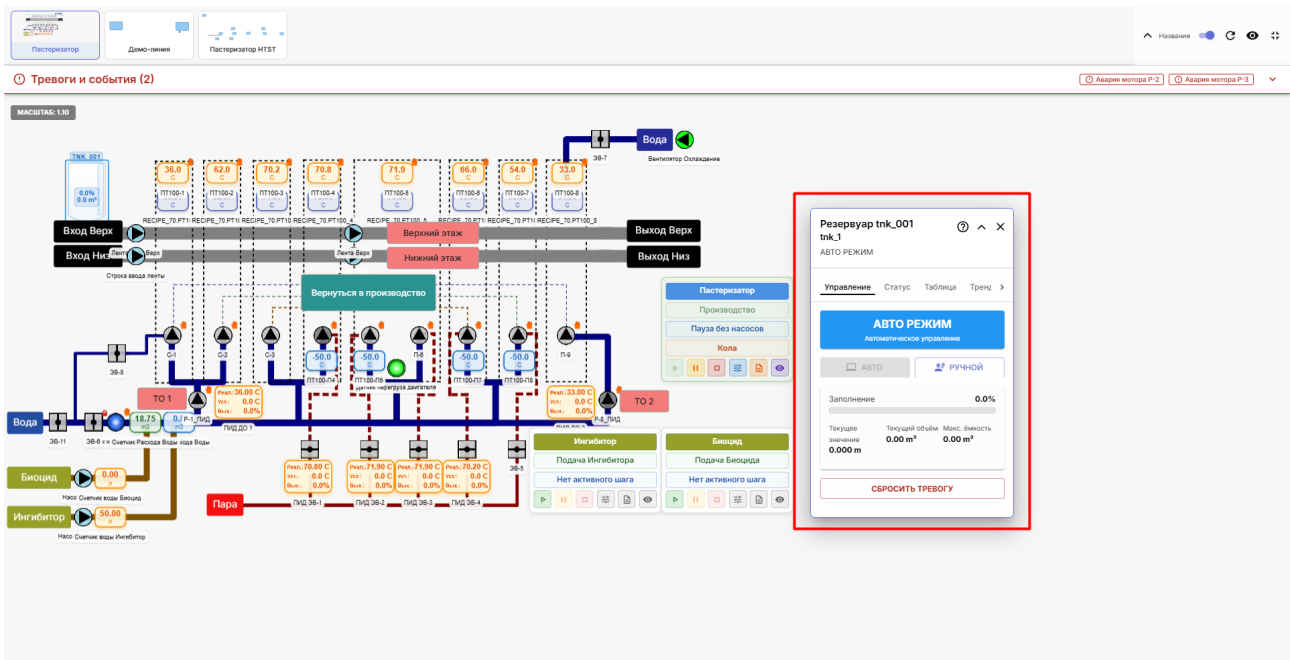


Таблица градуировки: две колонки — Sensor Value и Volume, кнопки Add Row, Delete Row, Import CSV, Export CSV.



Конструктор танка: слева — настройки (устройство, крышки, размеры), справа — SVG превью с выбранной формой.

Техническая документация

API танков:

| Метод | Endpoint | Описание |
|-------|---|------------------------------------|
| GET | /api/tanks/ | Список танков |
| POST | /api/tanks/ | Создать |
| PUT | /api/tanks/{id}/ | Обновить |
| DELET | /api/tanks/{id}/ | Удалить |
| POST | /api/tanks/{id}/calculate_volume/ | Пересчитать объём |
| POST | /api/tanks/{id}/import_strapping_table/ | Импорт таблицы (CSV) |
| GET | /api/tanks/{id}/export_strapping_table/ | Экспорт таблицы |
| GET | /api/tanks/{id}/volume_at_level/ | Объём при уровне (?sensor_value=X) |
| GET | /api/tank-sections/ | Секции (?tank=ID) |
| GET | /api/tank-strapping-entries/ | Строки таблицы (?tank=ID) |

Rust Worker — периодический расчёт (каждые 1000мс):

Файл `scada-worker/src/tasks/tank_volume.rs` — опрашивает сенсоры уровня и давления, рассчитывает объём по градуировочной таблице или геометрии, записывает результат в БД и передаёт через Redis.

Навигация приложения (Sidebar)

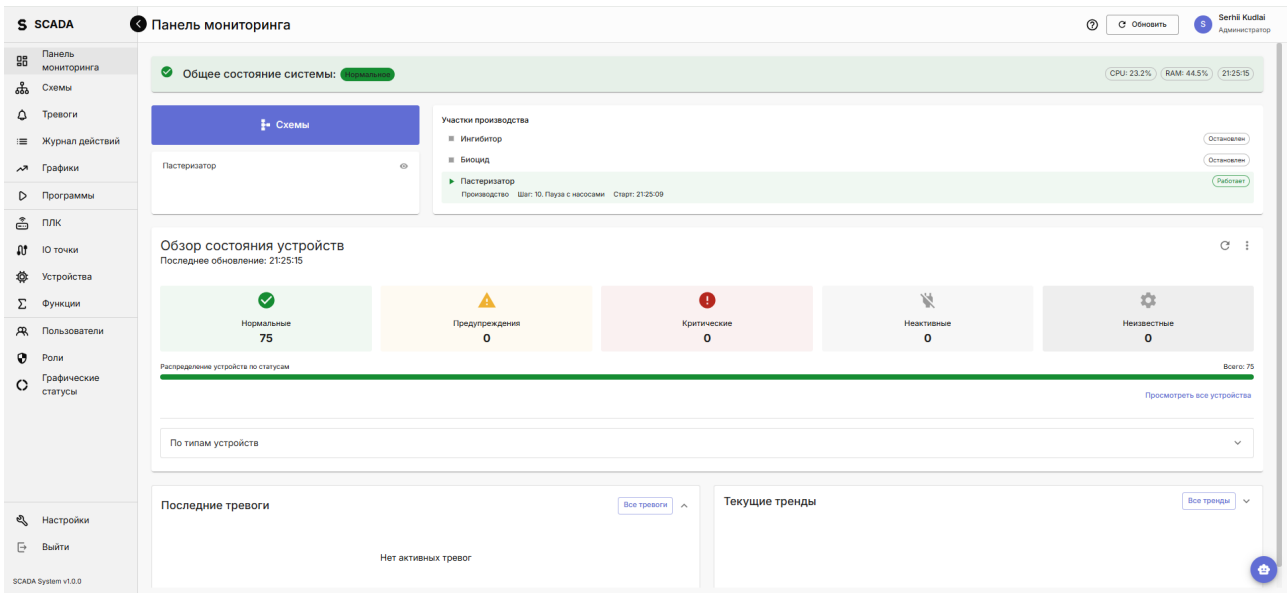
Для пользователя

Боковая панель навигации доступна на всех страницах. Пункты отображаются в зависимости от разрешений пользователя.

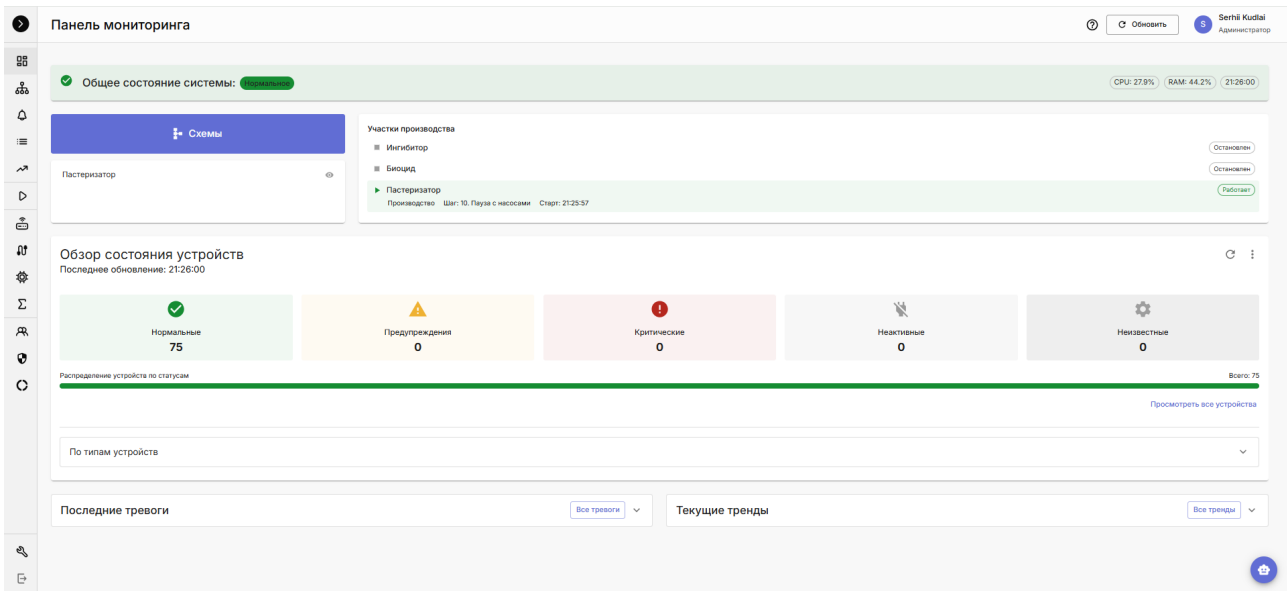
| Пункт меню | Страница | Требуемое разрешение |
|--------------------|--------------------|----------------------|
| Dashboard | Главная панель | — (доступно всем) |
| Schemas | Мнемосхемы | — |
| Alarms | Аварии | — |
| Activity Logs | Журнал активности | — |
| Trends | Тренды | — |
| Sequences | Последовательности | sequences_recipes |
| PLCs | ПЛК | plcs |
| I/O Points | Точки ввода-вывода | io_points |
| Devices | Устройства | devices |
| Functions | Мат. функции | devices |
| Users | Пользователи | users |
| Roles | Роли | roles |
| Graphical Statuses | Граф. статусы | graphical_statuses |
| Settings | Настройки | — (доступно всем) |
| Logout | Выход | — |

Дополнительные элементы боковой панели:

| Элемент | Описание |
|---------------------|----------------------------|
| Аватар пользователя | Отображение фото и имени |
| Свернуть/Развернуть | Уменьшить панель до иконок |
| Номер версии | Текущая версия приложения |



Боковая панель в развёрнутом виде: аватар пользователя сверху, пункты меню с иконками и подписями, кнопка Logout внизу.



Боковая панель в свёрнутом виде: только иконки пунктов меню.

Глобальные элементы интерфейса

Для пользователя

Шапка (Header):

| Элемент | Описание |
|----------------------|--|
| Заголовок страницы | Текущее название раздела |
| Кнопка помощи | Запуск обучающего тура |
| Индикатор обновлений | Бейдж при наличии обновлений |
| Индикатор соединения | Статус WebSocket/MQTT подключения |
| Контекстные действия | Кнопки, специфичные для текущей страницы |

Уведомления (Snackbar):

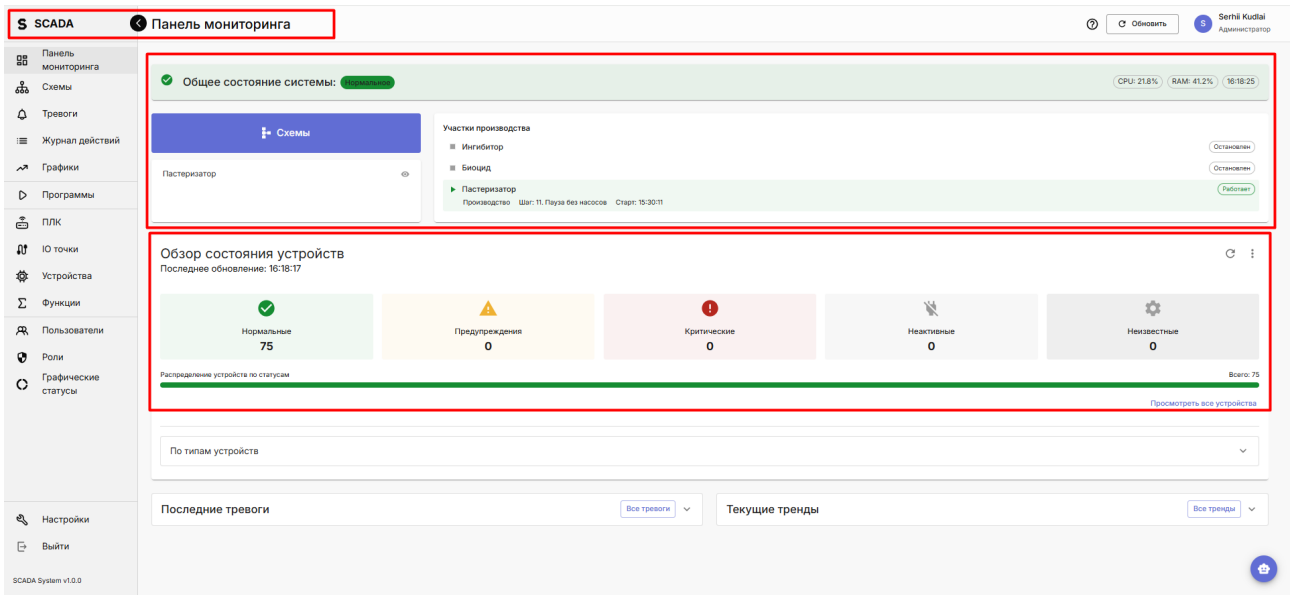
| Тип | Описание |
|-------------------|------------------|
| Success (зелёный) | Операция успешна |
| Error (красный) | Ошибка |
| Warning (жёлтый) | Предупреждение |
| Info (синий) | Информация |

Баннеры:

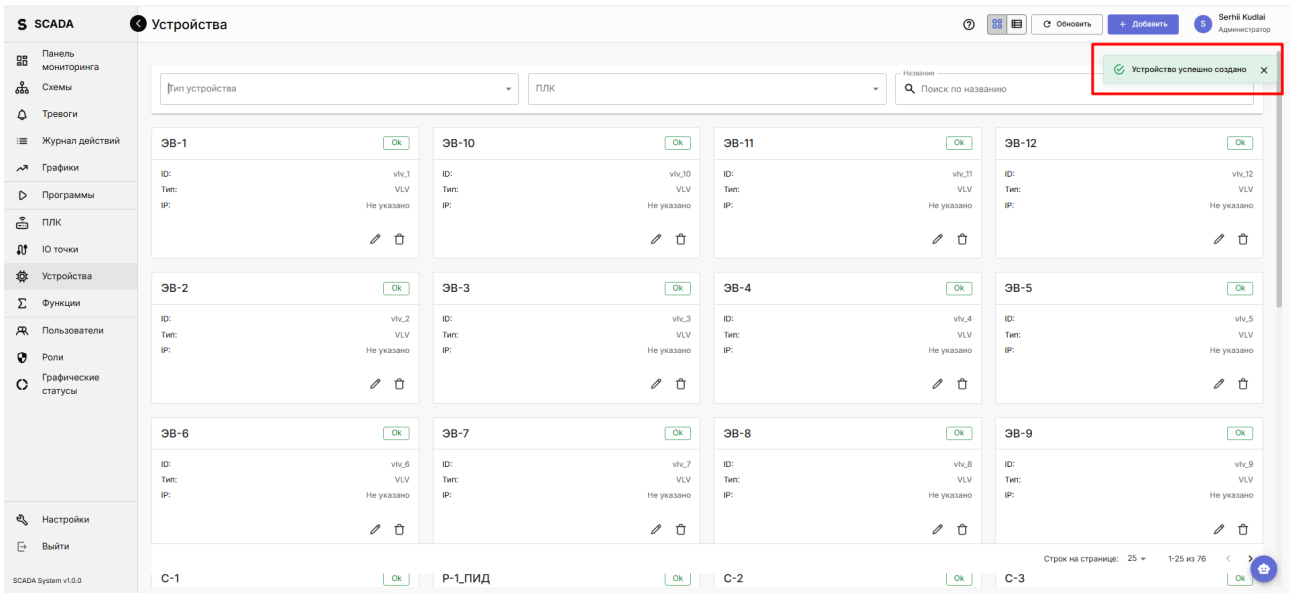
| Баннер | Описание |
|-------------------|---|
| Connection Status | Жёлтый баннер при потере соединения (после 15 сек) |
| Maintenance | Синий баннер при включённом режиме обслуживания |
| Version Mismatch | Оранжевый баннер при несоответствии версий frontend/backend |
| Update Available | Зелёный бейдж в шапке при доступном обновлении |

Обучающие туры (Guided Tours):

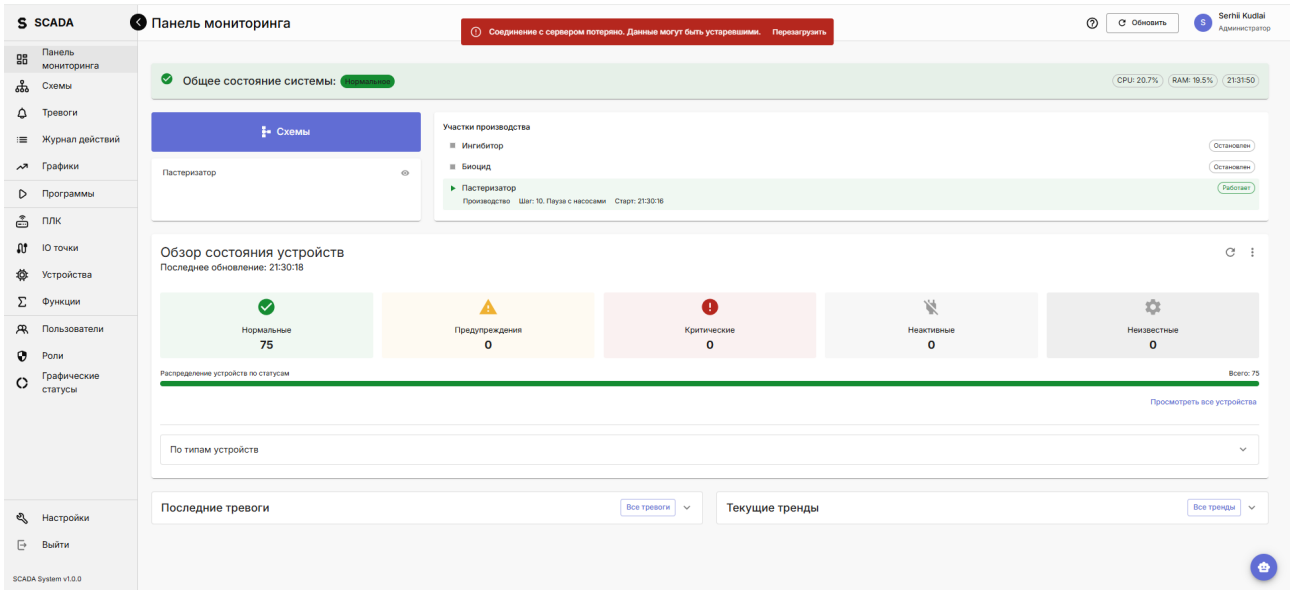
Доступны на каждой странице через кнопку «?» в шапке. Пошаговое интерактивное руководство с подсветкой элементов и описанием.



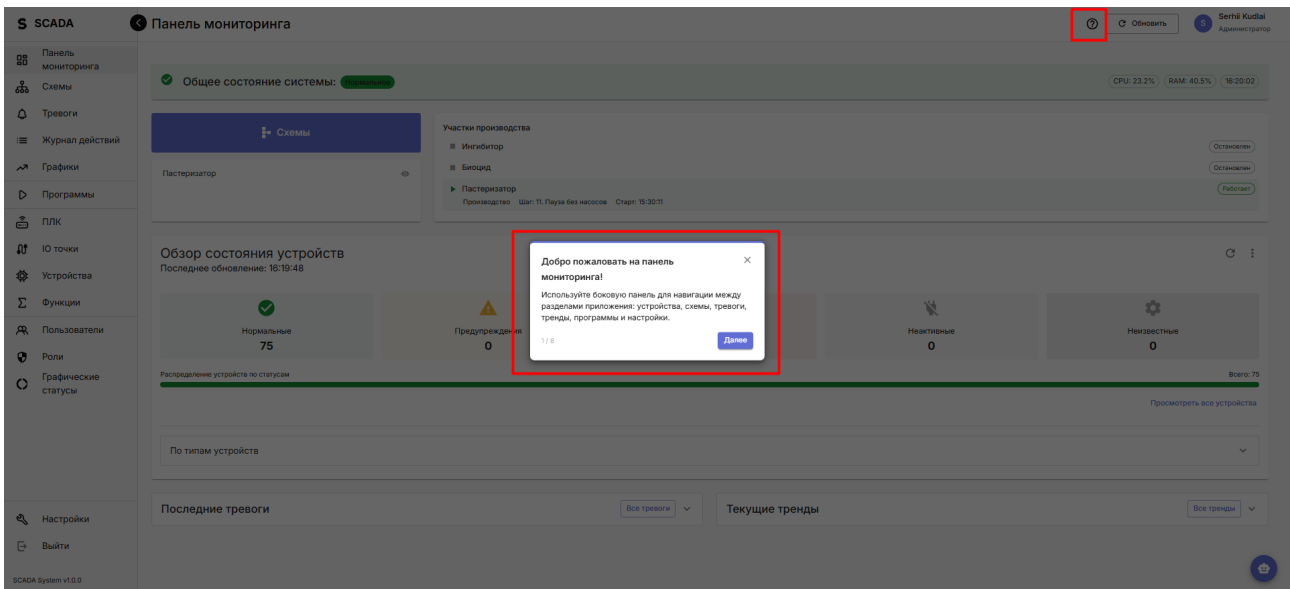
Шапка приложения: заголовок страницы слева, кнопка тура (i), индикатор обновлений (красный бейдж), статус соединения (зелёная/красная точка).



Примеры уведомлений: зелёный «Device saved successfully», красный «Error: Connection lost», жёлтый «Warning: License expires in 7 days».



Жёлтый баннер сверху: «Connection lost. Attempting to reconnect...» с анимацией.



Обучающий тур: подсвеченный элемент с подсказкой, кнопки «Next», «Back», «Skip», прогресс-бар.

Горячие клавиши

| Клавиша | Действие | Где работает |
|-------------|-------------------------------------|---------------------------------|
| Ctrl+Z | Отменить | Schema Builder |
| Ctrl+Y | Повторить | Schema Builder |
| Ctrl+S | Сохранить | Schema Builder |
| Delete | Удалить элемент | Schema Builder |
| Ctrl+A | Выделить все | Schema Builder |
| Ctrl+C | Копировать | Schema Builder |
| Ctrl+V | Вставить | Schema Builder |
| Escape | Отменить выделение / Закрыть диалог | Везде |
| Колесо мыши | Масштабирование | Schema Builder / Viewer, Trends |

Практические примеры

Раздел содержит сквозные сценарии автоматизации, показывающие, как связать воедино устройства, мнемосхему, последовательность, рецепт и события для реальных технологических процессов.

Пример 1. Автоматизация пастеризатора молока

Технологическая задача. Нагрев молока до 75 °С, выдержка 15 секунд, охлаждение до 4 °С. Это классический HTST-процесс (High Temperature Short Time), широко применяемый в молочной промышленности.

Оборудование процесса:

| Устройство | ID | Тип | Назначение |
|---------------------------|---------|----------------|--|
| Насос подачи | mot_00 | Motor | Прокачка молока через теплообменник |
| Клапан подачи пара | val_001 | Valve | Подача пара в нагревательную секцию |
| Клапан охладителя | val_002 | Valve | Подача ледяной воды |
| Датчик t° нагрева | ai_001 | Analog Input | Температура после нагревательной секции |
| Датчик t° выдержки | ai_002 | Analog Input | Температура после секции выдержки |
| Датчик t° на выходе | ai_003 | Analog Input | Температура после охладителя |
| Регулятор нагрева | pid_001 | PID Controller | Поддержание t° нагрева (уставка → val_001) |
| Регулятор охлаждения | pid_002 | PID Controller | Поддержание t° на выходе (уставка → val_002) |
| Уровень в приёмном танке | tnk_001 | Tank | Контроль наполнения |
| 3-ходовой клапан возврата | val_003 | Valve | Возврат недогретого молока на повторный нагрев |

Этап 1. Создание устройств

Создайте все устройства через Devices → «+». Для PID-регулятора pid_001:

```
POST /api/pid-controllers/
{
  "id": "pid_001",
  "name": "Регулятор нагрева",
  "plc": 1,
  "setpoint": 75.0,
  "kp": 2.5,
  "ki": 0.1,
  "kd": 0.05,
  "output_min": 0,
  "output_max": 100
}
```

Для танка с автоматическим расчётом объёма через strapping-таблицу:

```
POST /api/tanks/
{
  "id": "tnk_001",
  "name": "Приёмный танк",
  "plc": 1,
  "volume_method": "STRAPPING",
  "level_sensor": "ai_004",
  "alarm_hh": 95.0,
}
```

```
"alarm_h": 85.0,  
"alarm_l": 15.0,  
"alarm_ll": 5.0  
}
```

Этап 2. Последовательность (Sequence)

Последовательность из 5 шагов описывает полный цикл пастеризации:

| № | Шаг | Назначение | Условие перехода к следующему шагу |
|---|----------|--------------------------|------------------------------------|
| 1 | Idle | Ожидание команды запуска | Опция «Start» (оператор) |
| 2 | Heating | Нагрев до уставки | ai_001 >= 75.0 в течение 5 сек |
| 3 | Holding | Выдержка при t° | Таймер 15 сек |
| 4 | Cooling | Охлаждение до 4 °C | ai_003 <= 4.0 |
| 5 | Complete | Завершение | Возврат на Idle |

Создание последовательности через API:

```
POST /api/sequences/
{
  "name": "Pasteurization HTST",
  "name_translations": {"ru": "Пастеризация HTST", "uk": "Пастеризація HTST"},
  "description": "Нагрев 75°C → выдержка 15с → охлаждение 4°C"
}
```

Добавление шагов:

```
POST /api/sequence-steps/
{
  "sequence": 1,
  "order": 2,
  "name": "Heating",
  "name_translations": {"ru": "Нагрев"},
  "next_steps": {"3": [10]}
}
```

В поле next_steps ключ "3" — порядковый номер целевого шага, значение [10] — список ID условий перехода (TransitionCondition).

Условие перехода «температура ≥ 75°C»:

```
POST /api/transition-conditions/
{
  "id": 10,
  "type": "device",
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">=",
  "value": 75.0,
  "duration_seconds": 5
}
```

Важно. duration_seconds гарантирует, что условие выполняется непрерывно в течение указанного времени — это защищает от ложных переходов при единичных выбросах сигнала.

Этап 3. Производственный рецепт

Вынесем параметры, зависящие от конкретного продукта, в переменные рецепта — чтобы ту же последовательность использовать для разных молочных продуктов (цельное молоко, сливки,

обезжиренное молоко).

Переменные рецепта:

```
POST /api/production-recipe-variables/
{
  "name": "target_heating_temp",
  "name_translations": {"ru": "Целевая t° нагрева"},
  "sequence": 1,
  "default_value": 75.0,
  "min_value": 70.0,
  "max_value": 90.0
}
```

```
POST /api/production-recipe-variables/
{
  "name": "holding_time_sec",
  "name_translations": {"ru": "Время выдержки, сек"},
  "sequence": 1,
  "default_value": 15.0,
  "min_value": 10.0,
  "max_value": 30.0
}
```

Рецепт для цельного молока:

```
POST /api/production-recipes/
{
  "name": "Whole Milk 3.5%",
  "name_translations": {"ru": "Молоко цельное 3.5%"},
  "sequence": 1,
  "variables": {
    "target_heating_temp": 75.0,
    "holding_time_sec": 15.0
  }
}
```

Рецепт для сливок (более жёсткий режим):

```
POST /api/production-recipes/
{
  "name": "Cream 30%",
  "name_translations": {"ru": "Сливки 30%"},
  "sequence": 1,
  "variables": {
    "target_heating_temp": 85.0,
    "holding_time_sec": 20.0
  }
}
```

При активации рецепта уставка PID `pid_001.setpoint` автоматически привязывается к переменной `target_heating_temp`, а таймер шага Holding — к `holding_time_sec`.

Этап 4. Аварийные события

Определим две критические аварии:

1. Перегрев (выше 90 °C):

```
POST /api/accident-events/
{
  "name": "Overheat protection",
  "name_translations": {"ru": "Защита от перегрева"},
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">",
  "value": 90.0,
  "severity_level": 4,
  "action": "STOP_SEQUENCE"
}
```

severity_level: 4 — CRITICAL, немедленное прекращение последовательности.

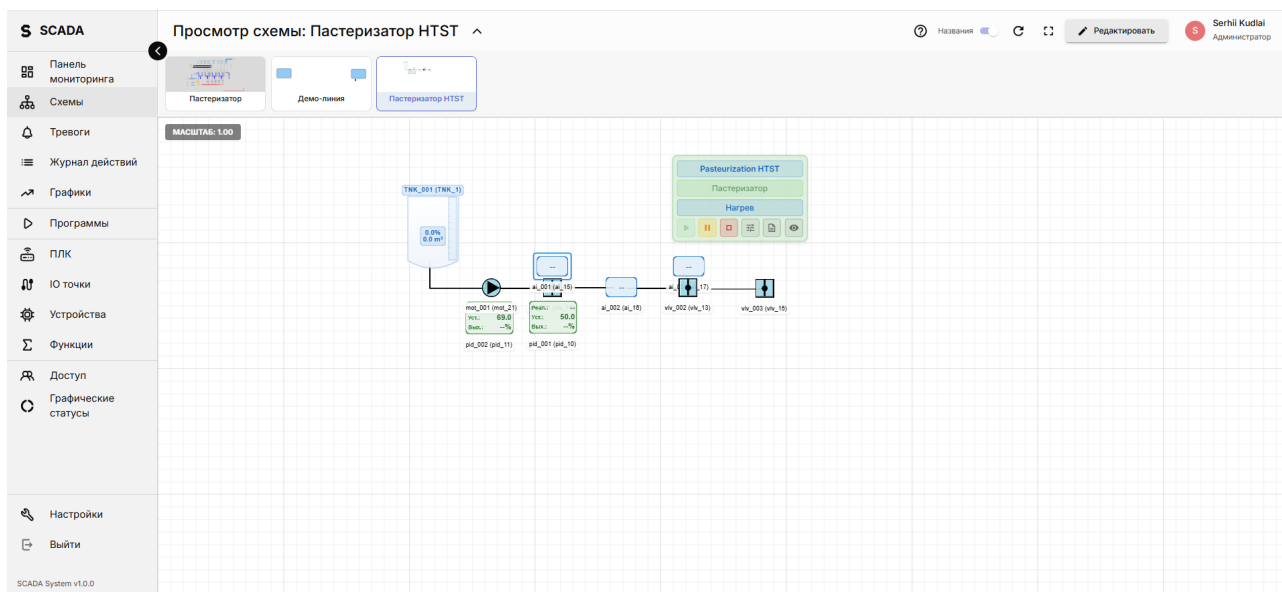
2. Недогрев на выдержке (ниже 73 °C в режиме Holding):

```
POST /api/accident-events/
{
  "name": "Insufficient pasteurization",
  "name_translations": {"ru": "Недостаточная пастеризация"},
  "device_id": "ai_002",
  "content_type": "analoginput",
  "operator": "<",
  "value": 73.0,
  "severity_level": 3,
  "action": "TRIGGER_DIVERT"
}
```

При срабатывании этой аварии система автоматически включит клапан возврата val_003 — недопастеризованное молоко уйдёт на повторный нагрев.

Этап 5. Мнемосхема

- Создайте схему «Пастеризатор HTST».
- В Schema Builder разместите:
- Приёмный танк tkn_001 через «Танк» — в конструкторе выберите форму FLAT / DOME и размеры 80×120.
- Насос mot_001, клапаны val_001, val_002, val_003, PID pid_001, pid_002.
- Датчики ai_001, ai_002, ai_003.
- Соедините трубами: танк → насос → нагреватель (с val_001 и ai_001) → секция выдержки (ai_002) → охладитель (val_002, ai_003) → выход.
- Привяжите последовательность: Последовательность → выбрать «Pasteurization HTST».
- Добавьте кнопки-опции «Start», «Pause», «Stop» (отдельный блок «Опция»).



Мнемосхема пастеризатора: слева приёмный танк с уровнем, от него труба идёт к насосу, далее через нагревательную секцию с датчиком температуры и клапаном пара, секцию выдержки, охлаждающую секцию с клапаном хладоносителя. Справа — блок последовательности с подсветкой активного шага «Heating».

Этап 6. Запуск процесса

- Откройте схему в Schema Viewer.
- На блоке последовательности выберите активный рецепт — например, «Whole Milk 3.5%».
- Нажмите Start — последовательность перейдёт со шага Idle на Heating.
- Наблюдайте за процессом в реальном времени:
- Подсветка активного шага (жёлтый → зелёный при успешном переходе).
- Текущее значение t° на датчике в роруп.
- Процент открытия val_001 (управляется PID).
- При достижении 75°C и удержании 5 секунд произойдёт автоматический переход на Holding.
- По истечении 15 секунд — переход на Cooling.
- При t° на выходе $\leq 4^\circ\text{C}$ — Complete → возврат на Idle.

Этап 7. Анализ после партии

- Откройте Trends → создайте график.
- Добавьте трендовые линии: ai_001, ai_002, ai_003, pid_001.output.
- Установите интервал — последние 30 минут.
- Проверьте:
- Форму кривой нагрева (время выхода на режим).
- Стабильность температуры выдержки (колебания должны быть $\leq 0.5^\circ\text{C}$).
- Скорость охлаждения.
- Экспортируйте график в PNG или CSV для отчёта.

Этап 8. Генерация PLC-проекта

- Откройте Dashboard → кнопка «Экспорт PLC» (или Settings → PLC Export).
- В диалоге выберите:
- ПЛК: PLC_DEMO
- Устройства, последовательности, рецепты и аварии — все.
- Нажмите «Preview» — система покажет список генерируемых файлов:
- Motor_FB.scl, Valve_FB.scl, PID_FB.scl — функциональные блоки устройств

- Sequence_Pasteurization_FB.scl — автомат состояний
- Accident_FB.scl — обработчики аварий
- OB1_Main.scl — главная программа
- Tags_Inputs.xml, Tags_Outputs.xml — таблицы тегов для TIA Portal
- «Validate» — проверит корректность конфигурации.
- «Generate» — сформирует ZIP для импорта в TIA Portal.

Пример 2. CIP-мойка с управлением партиями

Технологическая задача. Автоматическая мойка технологического оборудования (танков, трубопроводов) после производственной смены. Многоступенчатый процесс: предварительный ополаск → щелочная мойка → промежуточный ополаск → кислотная мойка → финальный ополаск.

Особенности примера:

- Используется Batch Management — каждый цикл мойки фиксируется как партия.
- Применяется Production Line с этапами (stages).
- Оператор может пропустить отдельные этапы через опции (kurzes Reinigen).
- Демонстрируется использование таймеров и счётчиков.

Этап 1. Оборудование

| ID | Тип | Назначение |
|----------------|--------------|--|
| tnk_cip_01 | Tank | Танк щелочного раствора (NaOH 2%) |
| tnk_cip_02 | Tank | Танк кислотного раствора (HNO 1.5%) |
| tnk_water | Tank | Танк оборотной воды |
| mot_cip_pump | Motor | Циркуляционный насос CIP |
| val_cip_supply | Valve | Клапан подачи в моющий контур |
| val_cip_return | Valve | Клапан обратки |
| val_alkali | Valve | Клапан подачи щёлочи |
| val_acid | Valve | Клапан подачи кислоты |
| val_water | Valve | Клапан подачи воды |
| ai_cip_temp | Analog Input | Температура раствора |
| ai_cip_cond | Analog Input | Электропроводность (контроль концентрации) |
| cnt_cip_flow | Counter | Расходомер на подаче |
| tim_cip_stage | Timer | Таймер текущего этапа |

Этап 2. Производственная линия и этапы

Создаём Production Line с пятью этапами — каждый этап будет иметь свою последовательность.

```
POST /api/production-lines/
{
  "name": "CIP Line 1",
  "name_translations": {"ru": "CIP-линия №1"},
  "description": "Мойка танков производства 1"
}
```

Этапы (ProductionLineStage):

| № | Название | Последовательность | Действие с партией |
|---|--------------------|---------------------------------------|----------------------------|
| 1 | Pre-rinse | seq_cip_rinse (вода 90 сек) | TRANSFER на следующий этап |
| 2 | Alkali wash | seq_cip_alkali (NaOH 20 мин при 75°C) | TRANSFER |
| 3 | Intermediate rinse | seq_cip_rinse | TRANSFER |
| 4 | Acid wash | seq_cip_acid (HNO 15 мин при 65°C) | TRANSFER |
| 5 | Final rinse | seq_cip_rinse | COMPLETE |

Поле `batch_transfer_action` на последнем шаге каждой последовательности управляет переходом:

- TRANSFER — партия передаётся на следующий этап автоматически.
- COMPLETE — партия завершается.
- NONE — партия остаётся на текущем этапе (ручное управление).

Этап 3. Последовательность «Щелочная мойка»

Шаги:

- Fill — наполнение контура щёлочью (`val_alkali = OPEN`, `mot_cip_pump = ON`).
- Heat — нагрев до 75 °C (условие: `ai_cip_temp >= 75.0`).
- Circulate — циркуляция 20 мин (таймер `tim_cip_stage`).
- Drain — слив (`val_cip_return = OPEN` в канализацию).

Рецепт «Щелочная мойка стандартная»:

```
POST /api/production-recipes/
{
  "name": "Alkali wash - standard",
  "production_line": 1,
  "variables": {
    "alkali_temp": 75.0,
    "alkali_duration_min": 20,
    "acid_temp": 65.0,
    "acid_duration_min": 15
  }
}
```

Обратите внимание: рецепт привязан не к последовательности, а к производственной линии (`production_line: 1`). Переменные доступны во всех последовательностях этапов этой линии.

Этап 4. Опции для оператора

Добавим кнопки опций, позволяющие ускорить или модифицировать процесс:

| Опция | Описание | Эффект |
|--------------|------------------------------|--|
| SKIP_ACID | Пропустить кислотную мойку | При активации этап 4 пропускается (переход сразу на этап 5) |
| DOUBLE_RINSE | Двойной финальный ополаск | Этап 5 повторяется дважды |
| LOW_TEMP | Мягкий режим (пониженная t°) | Переменные <code>alkali_temp = 60</code> , <code>acid_temp = 50</code> |

Опции добавляются через Sequence Options и размещаются на мнемосхеме как кнопки.

Этап 5. Создание и запуск партии

- Откройте схему CIP-линии в Schema Viewer.
- На элементе Production Line нажмите «Создать партию» — откроется CreateBatchDialog.

- Заполните:
- Batch name: CIP-2026-04-18-001
- Recipe: «Alkali wash — standard»
- Stage: Pre-rinse (начальный этап)
- Priority: Normal
- Create — партия попадает в очередь.
- Активируйте партию из панели BatchManagementPanel → Start.

API-эквивалент:

```
POST /api/production-lines/1/create_batch/  
{  
  "name": "CIP-2026-04-18-001",  
  "recipe_id": 5,  
  "starting_stage_id": 1,  
  "priority": "NORMAL"  
}
```

Этап 6. Мониторинг партии

В панели BatchManagementPanel отображается:

- Текущий этап (например, «Alkali wash»).
- Прогресс внутри этапа (таймер обратного отсчёта).
- Текущие показания ключевых датчиков (t°, проводимость, расход).
- Общий прогресс партии по всем этапам.

История переходов сохраняется в BatchStageHistory — позже можно восстановить полную хронологию:

```
GET /api/batches/{id}/history/
```

Этап 7. Валидация успешной мойки

По стандартам HACCP/ISO 22000 требуется документально подтвердить факт мойки. Используем аварийное событие в обратную сторону:

```
POST /api/events/  
{  
  "name": "CIP validation - conductivity check",  
  "device_id": "ai_cip_cond",  
  "operator": "<",  
  "value": 0.5,  
  "duration_seconds": 30,  
  "action": "LOG_EVENT"  
}
```

Это событие логирует в журнал момент, когда проводимость финального ополаска упала ниже 0.5 мСм/см в течение 30 секунд — признак полного удаления моющего средства.

Журнал доступен в Activity Logs с фильтром по последовательности/партии и может быть экспортирован в CSV для отчётности.

Чек-лист настройки нового технологического процесса

Используйте как шаблон при настройке любого нового процесса:

1. Проектирование (до ввода в систему):

- Нарисована P&ID схема процесса
- Составлен список устройств с уникальными ID
- Определены все аналоговые и дискретные сигналы
- Описана логика работы (шаги и условия переходов)
- Определены аварийные пороги и действия при срабатывании

2. Настройка в SCADA:

- Создан ПЛК с корректным адресом
- Созданы все устройства с привязкой к ПЛК
- Настроены параметры PID (K_p , K_i , K_d)
- Настроены Tank strapping-таблицы (если применимо)
- Создана мнемосхема с размещением устройств и трубами
- Создана последовательность с шагами
- Созданы условия переходов между шагами
- Созданы переменные рецепта
- Создан хотя бы один рецепт с дефолтными значениями
- Настроены аварийные события с корректными severity

3. Тестирование (до ввода в промышленную эксплуатацию):

- Каждое устройство управляется вручную (Manual mode)
- Условия переходов срабатывают корректно (используйте симуляцию — POST `/api/transition-conditions/{id}/simulate/`)
- Последовательность проходит все шаги end-to-end в режиме Manual
- Аварии срабатывают и останавливают процесс корректно
- Рецепты применяются — значения уставок обновляются
- Тренды пишут данные по всем ключевым переменным
- PLC-экспорт проходит валидацию без ошибок

4. Ввод в эксплуатацию:

- Операторам проведён инструктаж
- Выполнено резервное копирование проекта (POST `/api/project/export/`)
- Назначены роли и разрешения (оператор, мастер, инженер)
- Запущена первая партия на пониженных параметрах для проверки
- Оформлена документация на рабочем месте

Устранение неполадок и FAQ

Раздел содержит типичные проблемы, с которыми сталкиваются пользователи, и способы их решения.

Соединение и аутентификация

Q: Постоянно показывается жёлтый баннер «Connection lost».

Возможные причины:

- Остановился Rust-worker или Django-backend — проверьте: `docker-compose ps`, все 6 контейнеров должны быть в статусе Up.
- Nginx не проксирует WebSocket — откройте DevTools браузера → Network → WS. Если соединение закрывается с кодом 1006 или 502 — проблема в nginx. Проверьте `nginx.conf`: должны быть `proxy_http_version 1.1` и заголовки `Upgrade / Connection`.
- Истёк JWT-токен — выйдите и войдите повторно. В норме токен обновляется автоматически за 1 минуту до истечения.
- Сетевой фаервол блокирует порт 8001 (Rust WebSocket). Система упадёт на fallback Django WebSocket, но с задержкой. Откройте порт.

Q: После обновления появляется оранжевый баннер «Version mismatch».

Frontend и backend развёрнуты разных версий. Обновите браузер с очисткой кэша (Ctrl+Shift+R). Если не помогло — проверьте `docker-compose.yml`: tag образа для frontend должен совпадать с tag для backend и `scada-worker`.

Q: Не могу войти — пишет «Invalid credentials».

- Проверьте раскладку клавиатуры (CAPS LOCK).
- Убедитесь, что используете username, а не email (если настройка требует конкретного).
- Если пароль забыт и SMTP не настроен — войдите на сервер и сбросьте пароль командой: ``bash docker-compose exec backend python manage.py changepassword``
- Проверьте, не заблокирован ли ваш IP в IP Access (администратор может включить whitelist).

Мнемосхема и устройства

Q: Устройство на схеме остаётся серым, значения не обновляются.

- PLC offline — проверьте статус в разделе PLCs. Если статус не Online → устройство не получает данные.
- Устройство не привязано к PLC — откройте карточку устройства, проверьте поле PLC.
- WebSocket канал не подписан — проверьте в DevTools → Network → WS: должно быть подключение `ws/schema/{id}/devices`. Если его нет — обновите страницу.
- У вас нет разрешения на устройство — попросите администратора проверить ваши роли.

Q: Танк показывает объём 0, хотя датчик уровня работает.

- Не настроена strapping-таблица (для метода STRAPPING). Откройте танк → вкладка «Strapping Table» → добавьте минимум 2 точки (`sensor_value` → `volume`).
- Неверно выбран метод расчёта — проверьте поле `volume_method`: STRAPPING или GEOMETRIC. Для геометрического должны быть заданы TankSection (цилиндр, конус и т.п.).
- Rust-worker не обновляет объём — проверьте логи `docker-compose logs scada-worker | grep tank_volume`. По умолчанию расчёт раз в 1 секунду (`TANK_VOLUME_INTERVAL_MS`).

Q: Труба не анимирует поток, хотя мотор включён.

Включите свойство «Мониторинг процесса» (Process Monitoring) у трубы в правой панели свойств Schema Builder. По умолчанию оно выключено.

Q: При перетаскивании устройство «прыгает» обратно.

Это происходит при одновременном WebSocket-обновлении. Если вы в Schema Builder — редактирование блокирует live-updates. Если нет — сохраните изменение (Ctrl+S) сразу после перемещения.

Последовательности и рецепты

Q: Последовательность не переходит на следующий шаг, хотя условие выполнено.

- Условие не привязано к шагу — откройте текущий шаг, проверьте next_steps. Для целевого шага должен быть указан ID условия.
- `duration_seconds` ещё не истёк — условие должно держаться непрерывно указанное время. Посмотрите в роруп условия поле «Continuous for».
- Условие симулируется с другим значением — проверьте флаг is_simulated в WebSocket-данных условия. Сбросьте: POST /api/transition-conditions/clear_simulation/ с {"step_id": }.
- Текущее значение устройства не приходит — устройство offline (см. выше).

Q: Поменял рецепт, но уставка PID не изменилась.

Убедитесь, что:

- Рецепт активирован — POST /api/production-recipes/{id}/select-production-recipe/.
- Переменная рецепта привязана к полю устройства в редакторе последовательности (вкладка «Шаги» → для каждого шага проверьте значения устройств).
- Последовательность не находится в состоянии is_pause = true — pause замораживает все переходы и применения значений.

Q: Хочу протестировать логику без реального оборудования.

- Используйте симуляцию условий перехода — POST /api/transition-conditions/{id}/simulate/ с {"simulated_value": 80.0}. Условие будет оценивать переданное значение вместо реального.
- Для аналоговых входов — включите Simulation mode: POST /api/analog-inputs/{id}/set_simulation/ с {"is_simulated": true, "simulated_value": 75.0}.
- После тестирования обязательно выключите симуляцию — в промышленной эксплуатации это опасно.

Партии (Batches)

Q: Партия «висит» на этапе и не переходит дальше.

- Проверьте batch_transfer_action на последнем шаге последовательности текущего этапа — должно быть TRANSFER (для перехода) или COMPLETE (для завершения).
- Проверьте, что следующий этап существует и активен (is_active: true).
- Проверьте логи Rust-worker — сообщение batch transfer failed укажет причину.
- Ручной трансфер: POST /api/batches/{id}/transfer/ с {"next_stage_id": }.

Q: Очередь партий перемешалась по приоритету.

Сортировка очереди — по приоритету, затем по времени создания. Изменить приоритет: POST /api/batch-queue/{id}/set_priority/ с {"priority": "HIGH"}. Возможные значения: LOW, NORMAL, HIGH, URGENT.

PLC-экспорт

Q: При нажатии «Validate» появляется список ошибок.

Типичные ошибки валидации:

- Device ID contains invalid characters — ID должен быть только [a-z0-9_], без дефисов и пробелов.
- Sequence references missing device — устройство удалено, но ссылка в шаге осталась. Перейдите в редактор последовательности и исправьте.
- PID output_max <= output_min — проверьте границы PID.
- Tank strapping entries must be monotonic — значения sensor_value должны возрастать.
- No PLC assigned to device X — привяжите устройство к ПЛК.

Q: Сгенерированный SCL не компилируется в TIA Portal.

- Проверьте версию TIA Portal — генератор поддерживает TIA Portal v16+.
- Убедитесь, что импортированы обе XML-таблицы тегов (Inputs и Outputs) до импорта SCL.
- Посмотрите лог компиляции TIA Portal — ошибка обычно указывает конкретный FB и строку.

Тренды

Q: График пуст, хотя устройство работает.

- Временной диапазон слишком мал или в будущем — сбросьте на «Last 1 hour».
- Timescale offline — проверьте GET /api/graphs/timescale-status/. Если возвращает disconnected — TimescaleDB не запущен или недоступен.
- Data points не собираются — раздел Data Points → устройство должно иметь активную подписку на сбор.

Q: Тренд сильно тормозит при большом интервале.

Использует даунсемплинг. Если данных слишком много — уменьшите диапазон или интервал агрегации. Для долгосрочного анализа экспортируйте данные в CSV и обработайте вне системы.

AI-ассистент

Q: AI-ассистент возвращает «Authentication failed».

- Проверьте API-ключ OpenRouter в Settings → AI Assistant. Ключи шифруются Fernet при сохранении — ошибка означает либо просроченный ключ, либо повреждение БД.
- Проверьте баланс на OpenRouter — при исчерпании средств запросы отклоняются.
- Проверьте логи: docker-compose logs ai-assistant | tail -50.

Q: AI долго отвечает.

Модель выполняет агентный цикл до 15 итераций с вызовами инструментов. Сложные запросы («проанализируй все аварии за неделю и предложи оптимизацию») могут занимать 30-60 секунд. Для быстрых ответов — короткие и конкретные вопросы.

Производительность

Q: Интерфейс тормозит при работе со схемой с 100+ устройствами.

- Отключите анимацию потока у труб, не используемых для процесса.
- В Settings → Performance включите «Throttle non-critical updates» (обновления не-активных устройств раз в 2 секунды вместо 250 мс).
- Увеличьте RAM Docker (минимум 4 ГБ для комфортной работы).

Q: База данных быстро растёт.

Основной источник — journal активности и исторические тренды. Настройте retention в Settings → Data Management:

- Activity Logs: по умолчанию 90 дней.
- Trend data: настраивается per-device через Data Points.

Безопасность и лучшие практики

Этот раздел содержит рекомендации по безопасной эксплуатации системы. Следование им снижает риск инцидентов на производстве и обеспечивает соответствие стандартам качества (ISO 9001, HACCP, GMP).

Правила работы с режимами Auto / Manual

Manual mode даёт оператору прямое управление устройством в обход последовательностей и аварийных блокировок. Это мощный инструмент — и опасный.

Правильное использование Manual:

- Только для пуска-наладки и диагностики.
- Только на остановленном оборудовании или в специально предусмотренных для этого состояниях.
- С обязательным устным согласованием с начальником смены на работающем производстве.
- С записью в журнал активности (это происходит автоматически, но оператор должен знать).

Неправильное использование Manual:

- Обход защитных условий для ускорения процесса.
- Запуск оборудования в ручном режиме без знания текущего состояния партии.
- Оставление оборудования в Manual после окончания смены.

Правило. По окончании работы с устройством в Manual — всегда возвращайте его в Auto. Система не делает это автоматически.

Симуляция: только для тестов

Симуляция условий, аналоговых входов и таймеров — инструмент для проектировщика, не для оператора.

| Сценарий | Разрешено? |
|--|---|
| Тестирование логики последовательности на стенде | Да |
| Проверка срабатывания аварий в симуляторе | Да |
| «Пропуск» условия во время производства, чтобы ускорить запуск | Категорически нет |
| Симуляция датчика, когда реальный сломался | Нет — исправьте датчик или остановите процесс |

Любая активная симуляция при работающем производстве — потенциальная причина инцидента. Перед запуском промышленной партии запустите проверку: `GET /api/transition-conditions/?is_simulated=true` — список не должен содержать ничего, что касается активной последовательности.

Бэкапы и восстановление

Что бэкапить:

- Конфигурация проекта — устройства, схемы, последовательности, рецепты, аварии. Делается через `POST /api/project/export/` → ZIP-файл.
- База данных — полный дамп PostgreSQL (включает journal, исторические тренды, данные партий). Команда: ``bash docker-compose exec postgres pgdump -U scada scadadb > backup_$(date +%Y%m%d).sql``
- Пользовательские загрузки — volume ai_uploads (изображения для AI-анализа).
- Настройки nginx и docker-compose.yml — храните в Git.

Как часто:

- Конфигурация проекта — перед любым значимым изменением.
- База данных — ежедневно автоматически + перед обновлениями системы.
- Перед обновлением версии SCADA — обязательно оба типа бэкапа.

Проверка восстановления. Раз в квартал разверните бэкап на тестовом стенде и убедитесь, что он корректен. Бэкап, который никогда не проверялся, — это ненадёжный бэкап.

Управление доступом

Принцип минимальных привилегий. Каждый пользователь получает только те разрешения, которые нужны для его работы.

Типовая матрица ролей:

| Роль | Разрешения |
|---------------|--|
| Оператор | Просмотр схем + управление устройствами в Auto + подтверждение аварий. Нет доступа к редактированию. |
| Мастер смены | Всё, что у оператора + запуск/остановка партий + изменение рецептов из списка разрешённых. |
| Технолог | Всё, что у мастера + создание/редактирование рецептов и переменных. |
| Инженер | Всё, что у технолога + редактирование схем, последовательностей, устройств. |
| Администратор | Полный доступ, включая управление пользователями и системные настройки. |

Дополнительно:

- Per-schema access — отдельные схемы можно закрыть от пользователей, не работающих на данном участке.
- IP-фильтрация — ограничьте доступ только с внутренней сети предприятия (диапазон 192.168.0.0/16 или VPN).
- Регулярный аудит — раз в квартал проверяйте список активных пользователей в Users. Удаляйте уволенных сотрудников.

Обновление системы

Последовательность корректного обновления:

- Запланируйте окно — обновление требует остановки производства на 5-15 минут.
- Сделайте бэкап (конфигурация + БД).
- Прочитайте CHANGELOG обновления — могут быть breaking changes в API или миграции.
- Запустите dry-run — система проверит миграции без применения.
- Примените обновление — через Settings → System → Updates.
- Дождитесь smoke-тестов — система автоматически прогонит базовые проверки.
- Если что-то сломалось — используйте Rollback (доступен в течение 24 часов после обновления).

Никогда не обновляйте систему во время активной партии или при наличии нерешённых критических аварий.

Работа с авариями

Что ДЕЛАТЬ при срабатывании аварии:

- Не паниковать. Прочитайте описание аварии в Alarm Bar.
- Перейти к оборудованию, на котором сработала авария.

- Визуально оценить состояние (утечка, перегрев, посторонний шум).
- Если опасности нет — нажать Acknowledge в Alarms.
- Устранить причину.
- Нажать Resolve после устранения.

Что НЕ ДЕЛАТЬ:

- Массово «квитировать» все аварии без разбора (resolve-all — только в крайних случаях по решению мастера).
- Удалять аварийные события из настроек во время работы.
- Снижать severity-level, чтобы избавиться от надоедливого аларма — вместо этого настройте корректные пороги.

Журнал активности

Журнал (Activity Log) — ваша защита и источник истины. Он автоматически фиксирует:

- Входы и выходы пользователей.
- Все команды управления устройствами (Start, Stop, Open, Close).
- Изменения рецептов и последовательностей.
- Срабатывания аварий.
- Запуски/остановки/трансферы партий.

Рекомендации:

- Храните журнал минимум 1 год (требование многих стандартов качества).
- Регулярно экспортируйте в CSV для архива.
- При расследовании инцидента — первым делом смотрите журнал за период ± 1 час вокруг события.

Рекомендации по именованию

Единая конвенция именования сильно упрощает жизнь — особенно при росте проекта.

Устройства:

- Префикс по типу: mot_ (motor), val_ (valve), ai_ (analog input), pid_, tnk_ (tank).
- Нумерация с ведущими нулями: mot_001, mot_002 (до mot_099).
- Группировка по участку: val_tank1_in, val_tank1_out.

Последовательности:

- Префикс процесса: seq_pasteurize_htst, seq_cip_alkali.

Рецепты:

- С именем продукта и параметрами: Milk_Whole_3.5%, Cream_30%_HotFill.

Глоссарий

Справочник терминов, используемых в системе и в данной документации.

Общая автоматизация

SCADA (Supervisory Control And Data Acquisition) — система диспетчерского управления и сбора данных. Класс программных систем для мониторинга и управления технологическими процессами.

ПЛК / PLC (Programmable Logic Controller) — программируемый логический контроллер. Промышленный компьютер, выполняющий логику управления оборудованием в реальном времени.

HMI (Human-Machine Interface) — интерфейс «человек-машина». В данной системе HMI — это Schema Viewer.

P&ID (Piping and Instrumentation Diagram) — технологическая схема с обозначением трубопроводов и контрольно-измерительных приборов.

I/O — Input/Output, точки ввода-вывода. Физические или логические каналы связи между ПЛК и оборудованием.

Tag — тег, именованный сигнал ПЛК. В SCADA соответствует адресу, по которому читается/записывается значение (например, DB10.DBDO для Siemens).

Устройства

AI (Analog Input) — аналоговый вход. Принимает непрерывный сигнал (например, 4-20 мА, 0-10 В) и преобразует в инженерную величину (температура, давление).

AO (Analog Output) — аналоговый выход. Выдаёт непрерывный сигнал для управления (уставка ЧРП, положение клапана).

DI / DO (Digital Input / Output) — дискретный вход/выход. Логический сигнал 0/1 (кнопка, концевик, лампа, пускатель).

PID (Proportional-Integral-Derivative) — ПИД-регулятор. Алгоритм автоматического регулирования с тремя составляющими: пропорциональной (K_p), интегральной (K_i) и дифференциальной (K_d).

Setpoint (SP) — уставка, целевое значение регулируемой величины.

Process Value (PV) — текущее значение регулируемой величины, измеренное датчиком.

VSD (Variable Speed Drive) — частотно-регулируемый привод (ЧРП). Управляет скоростью вращения электродвигателя.

COS (Coordination Signal) — координационный сигнал. Логическая переменная, используемая для межпроцессной синхронизации (например, «линия 1 готова к передаче партии»).

Counter — счётчик (импульсов, событий). Используется для подсчёта расхода, оборотов, циклов.

Interlock — блокировка. Условие, запрещающее выполнение действия (например, «не запускать насос при закрытом клапане»).

Танки

Strapping table (калибровочная таблица) — таблица соответствия показаний датчика уровня объёму жидкости в танке. Используется для танков нестандартной формы.

HH / H / L / LL — High High / High / Low / Low Low. Четыре уровня аварийных порогов для танков: критически высокий, высокий, низкий, критически низкий.

Hydrostatic level — гидростатический расчёт уровня по перепаду давлений (ΔP) между верхним и нижним датчиком давления в танке.

Последовательности и рецепты

Sequence — последовательность. Автомат состояний, описывающий последовательность шагов технологического процесса.

Step — шаг. Одно состояние последовательности.

Transition — переход. Связь между двумя шагами с условием выполнения.

Transition Condition — условие перехода. Тип: Device (по значению устройства), Timer (по таймеру), Option (по опции оператора).

Recipe — рецепт. Конкретные значения устройств для каждого шага последовательности.

Production Recipe — производственный рецепт. Набор переменных для конкретного продукта.

Recipe Variable — переменная рецепта. Параметр, значение которого зависит от типа продукта.

Option — опция. Управляющий сигнал от оператора (кнопка Start / Pause / Skip).

Партии и производственные линии

Batch (партия) — единица продукции, проходящая через производственный процесс. Принцип взят из стандарта ISA-88.

Production Line (производственная линия) — набор оборудования для производства продукции.

Production Line Stage (этап линии) — один этап обработки на линии (например, «мойка» → «сушка» → «упаковка»).

Batch Transfer — передача партии с одного этапа на следующий. Управляется полем batch_transfer_action на шаге последовательности.

Batch Queue — очередь партий, ожидающих запуска на линии.

ISA-88 — международный стандарт для batch control в периодических производствах.

PLC-экспорт и Siemens

TIA Portal — Totally Integrated Automation Portal. Среда разработки Siemens для конфигурирования ПЛК.

SCL (Structured Control Language) — язык программирования ПЛК Siemens (аналог Pascal по стандарту IEC 61131-3).

FB (Function Block) — функциональный блок. Программная единица в ПЛК со своей логикой и экземплярными данными.

DB (Data Block) — блок данных. Структура для хранения переменных в памяти ПЛК.

OB (Organization Block) — организационный блок. Точка входа программы ПЛК.

OB1 — главный циклический OB, вызывается каждый скан-цикл.

SimaticML — XML-формат для импорта/экспорта проектов в TIA Portal.

GSD-файл (Generic Station Description) — описатель аппаратного модуля PROFIBUS/PROFINET.

WebSocket и API

REST API — Representational State Transfer. Протокол HTTP-запросов для CRUD-операций с данными.

WebSocket — двунаправленный протокол поверх TCP, позволяющий push-обновления от сервера к клиенту.

JWT (JSON Web Token) — токен аутентификации. Содержит данные пользователя в подписанном виде.

SSE (Server-Sent Events) — технология односторонней потоковой передачи от сервера к клиенту. Используется AI-ассистентом для потоковых ответов.

Delta update — дельта-обновление. Передача только изменившихся полей объекта вместо полного состояния.

Batching — пакетирование. Группировка нескольких обновлений в одно сообщение для снижения сетевой нагрузки.

Производственные стандарты

HACCP (Hazard Analysis and Critical Control Points) — анализ рисков и критические контрольные точки. Методология обеспечения безопасности пищевой продукции.

GMP (Good Manufacturing Practice) — надлежащая производственная практика. Стандарт управления производством, особенно в фармацевтике и пищевом производстве.

CIP (Cleaning In Place) — мойка на месте. Автоматическая мойка технологического оборудования без разборки.

HTST (High Temperature Short Time) — высокотемпературная кратковременная пастеризация (72-75°C, 15-20 секунд).

ISO 9001 — международный стандарт системы менеджмента качества.

Приложение А: Справочник WebSocket-каналов

| Канал | Описание | Направление |
|--------------------------|---------------------------|-----------------|
| ws/device/{id} | Статус устройства | Server → Client |
| ws/device/{id}/control | Команды устройству | Bidirectional |
| ws/schema/{id}/devices | Все устройства схемы | Server → Client |
| ws/schema/{id}/pipes | Трубопроводы схемы | Server → Client |
| ws/schema/{id}/sequences | Последовательности схемы | Server → Client |
| ws/schema/{id}/batches | Партии схемы | Bidirectional |
| ws/schema/{id}/accidents | Аварии схемы | Server → Client |
| ws/sequence/{id} | Статус последовательности | Server → Client |
| ws/sequences/all | Все последовательности | Server → Client |
| ws/pipe/{id} | Статус трубопровода | Server → Client |
| ws/system | Системные метрики | Server → Client |
| ws/plc/status | Статус ПЛК | Server → Client |
| ws/device-status | Глобальный статус | Server → Client |
| ws/cos/all | Координационные сигналы | Server → Client |
| ws/math-functions | Мат. функции | Server → Client |
| ws/activity-log | Журнал активности | Server → Client |
| ws/accidents | Все аварии | Server → Client |

Rust WebSocket сервер (порт 8001, прокси `/rws/ws/`):

- Read-only для массовых обновлений устройств
- Автоматический fallback на Django WebSocket
- Аутентификация через JWT в query-параме­тре URL

Приложение В: Справочник API (сводная таблица)

| Ресурс | Base URL | CRUD | Доп. действия |
|-----------------------|-----------------------------------|------|--|
| Auth | /api/auth/ | — | login, register, me, refresh |
| Users | /api/users/ | | set_password, upload-avatar, roles |
| Roles | /api/roles/ | | permissions |
| Permissions | /api/permissions/ | | categories, actions |
| Devices (all) | /api/devices/ | — | alldevices, batchwebsocket_update |
| Analog Inputs | /api/analog-inputs/ | | setmode, setsimulation, setinterlock, setmanual_value |
| Analog Outputs | /api/analog-outputs/ | | (аналогично) |
| Digital Inputs | /api/digital-inputs/ | | (аналогично) |
| Digital Outputs | /api/digital-outputs/ | | (аналогично) |
| Motors | /api/motors/ | | (аналогично) |
| Valves | /api/valves/ | | (аналогично) |
| PID Controllers | /api/pid-controllers/ | | (аналогично) |
| Counters | /api/counters/ | | (аналогично) |
| Timers | /api/timers/ | | (аналогично) |
| COS | /api/coordination-signals/ | | (аналогично) |
| VSD | /api/variable-speed-drives/ | | (аналогично) |
| Tanks | /api/tanks/ | | calculatevolume, import/exportstrapping |
| Tank Sections | /api/tank-sections/ | | — |
| Tank Strapping | /api/tank-strapping-entries/ | | — |
| PLCs | /api/plcs/ | | status, set-run-mode |
| Schemas | /api/schemas/ | | add/remove devices, sequences, options, production lines |
| Pipes | /api/pipes/ | | — |
| Sequences | /api/sequences/ | | steps, transitions, positions |
| Sequence Steps | /api/sequence-steps/ | | — |
| Transition Conditions | /api/transition-conditions/ | | simulate, clear_simulation |
| Recipes | /api/recipes/ | | — |
| Production Recipes | /api/production-recipes/ | | select-production-recipe |
| Recipe Variables | /api/production-recipe-variables/ | | — |
| Events | /api/events/ | | set_active |
| Accident Events | /api/accident-events/ | | — |
| Accidents | /api/accidents/ | | acknowledge, resolve, resolve-all |
| Sequence Options | /api/sequence-options/ | | — |
| Production Lines | /api/production-lines/ | | activate, deactivate, create_batch |
| Production Stages | /api/production-line-stages/ | | — |
| Batches | /api/batches/ | | start, pause, resume, cancel, transfer |
| Batch Queue | /api/batch-queue/ | | reorder, set_priority |
| Pulse Modules | /api/pulse-modules/ | | start, stop |

| Ресурс | Base URL | CRUD | Доп. действия |
|--------------------|------------------------------|------|---|
| Math Functions | /api/math-functions/ | | calculate, validate, test |
| I/O Points (AI) | /api/points/analog-inputs/ | | validate, test |
| I/O Points (AO) | /api/points/analog-outputs/ | | (аналогично) |
| I/O Points (DI) | /api/points/digital-inputs/ | | (аналогично) |
| I/O Points (DO) | /api/points/digital-outputs/ | | (аналогично) |
| Graphical Statuses | /api/graphical-statuses/ | | upload-icon, list-icons |
| Graphs | /api/graphs/ | | data, timescale-status |
| Data Points | /api/data-points/ | | — |
| HW Catalog | /api/hw-catalog/ | | import-gsd |
| HW Stations | /api/hw-stations/ | | auto-assign-addresses |
| HW Slots | /api/hw-slots/ | | bulk-update |
| HW Network | /api/hw-network/ | | — |
| TIA Deploy | /api/tia/ | — | deploy, auto-addresses, address-mapping |
| TIA Agent Config | /api/tia-agent-config/ | | check-connection, agent-status |
| Activity Logs | /api/user-activity-logs/ | R/D | recent, statistics, clear |
| IP Access | /api/ip-access/allowed-ips/ | | checkip, myips |
| License | /api/license/ | — | status, activate, generate |
| System | /api/system/ | — | info, health, updates, maintenance |
| Project | /api/project/ | — | export, import, backup, restore |
| PLC Export | /api/plc-export/ | — | preview, validate, generate |
| AI Sessions | /ai-api/v1/sessions | | messages |
| AI Chat | /ai-api/v1/chat | — | messages (SSE) |
| AI Images | /ai-api/v1/images | — | analyze, apply |
| AI Settings | /ai-api/v1/settings | R/U | — |

Приложение С: Коды ошибок и аварий

Справочник кодов и статусов, с которыми взаимодействует пользователь и которые встречаются в журналах, WebSocket-сообщениях и API-ответах.

Уровни серьезности аварий (Severity levels)

Поле severity_level модели AccidentType и accident_event.severity_type.

| Код | Наименование | Цвет | Рекомендуемая реакция |
|--------------|--------------|-----------|---|
| 1 / LOW | Низкий | Голубой | Информационное сообщение. Не требует немедленных действий. Записывается в журнал для статистики. |
| 2 / MEDIUM | Средний | Жёлтый | Требует внимания оператора в ближайшее время. Производство может продолжаться. |
| 3 / HIGH | Высокий | Оранжевый | Требует немедленного внимания. Возможна деградация качества продукта или риск повреждения оборудования. |
| 4 / CRITICAL | Критический | Красный | Немедленная остановка последовательности. Возможен риск для людей или оборудования. |

L

Типы серьезности событий (`SeverityType` для Events):

| Код | Наименование | Применение |
|---------|----------------|---|
| INFO | Информационное | Статусные сообщения (старт, завершение этапа) |
| WARNING | Предупреждение | Отклонение от нормы без угрозы |
| ERROR | Ошибка | Критическая проблема, требующая вмешательства |

Статусы аварий (Accident Status)

Поле status модели Accident. Жизненный цикл аварии:

NEW → ACKNOWLEDGED → IN_PROGRESS → RESOLVED → CLOSED

| Код | Наименование | Описание |
|--------------|--------------|---|
| NEW | Новая | Только что возникла, не подтверждена. Отображается в Alarm Bar. |
| ACKNOWLEDGED | Подтверждена | Оператор подтвердил получение (POST /api/accidents/{id}/acknowledge/). Убирается из топ-приоритета, но остаётся активной. |
| IN_PROGRESS | В работе | Ведётся устранение причины. |
| RESOLVED | Устранена | Причина устранена (POST /api/accidents/{id}/resolve/). |
| CLOSED | Закрыт | Финальный статус после RESOLVED + определённое время (для отчётности). |

D a

Статусы последовательностей

Поля модели Sequence:

| Поле | Тип | Значение | Описание |
|------------|------|----------|---|
| is_running | bool | true | Последовательность активна, шаги переключаются. |

| Поле | Тип | Значение | Описание |
|----------|------|----------|---|
| is_pause | bool | true | Приостановлена, переходы заморожены, устройства удерживают текущее состояние. |
| is_stop | bool | true | Остановлена, все устройства возвращены в безопасное состояние. |

Комбинации:

- is_running=true, is_pause=false, is_stop=false — работа.
- is_running=true, is_pause=true, is_stop=false — на паузе (можно возобновить).
- is_running=false, is_pause=false, is_stop=true — полностью остановлена.

Статусы партий (Batch Status)

| Код | Наименование | Описание |
|-----------|----------------|-----------------------------------|
| PENDING | В ожидании | В очереди, не запущена |
| RUNNING | Выполняется | Активная партия |
| PAUSED | Приостановлена | Временно остановлена оператором |
| COMPLETED | Завершена | Успешно прошла все этапы |
| CANCELLED | Отменена | Прервана вручную |
| FAILED | Ошибка | Прервана из-за критической аварии |

Приоритеты (Priority):

| Код | Порядок в очереди |
|--------|-----------------------|
| URGENT | Обрабатывается первой |
| HIGH | 2 |
| NORMAL | 3 (по умолчанию) |
| LOW | 4 |

Режимы устройств (Device Mode)

Применимы к моторам, клапанам, PID, AI/AO/DI/DO:

| Код | Описание |
|------------|--|
| AUTO | Управление через последовательность / логику системы |
| MANUAL | Прямое управление оператором |
| SIMULATION | Симулированное значение (только для тестирования) |
| INTERLOCK | Заблокировано блокировкой (нельзя менять значение) |

Коды ошибок API

Типичные HTTP-коды и их значение в контексте системы:

| Код | Значение | Типичная причина |
|-------------|----------|---------------------|
| 200 OK | Успех | Запрос выполнен |
| 201 Created | Создано | Создан новый объект |

| Код | Значение | Типичная причина |
|-------------------------|-----------------|--|
| 204 No Content | Успех без тела | Удаление |
| 400 Bad Request | Неверный запрос | Валидация не прошла. Тело ответа содержит {"field": ["error msg"]} |
| 401 Unauthorized | Не авторизова | JWT-токен отсутствует или невалиден |
| 403 Forbidden | Доступ запрещён | Аутентифицирован, но нет прав на действие |
| 404 Not Found | Не | ID не существует |
| 409 Conflict | Конфликт | Например, попытка создать устройство с существующим ID |
| 423 Locked | Заблокировано | Объект заблокирован другим процессом (например, сохранение схемы во время запущенной последовательности) |
| 500 Internal Error | Ошибка сервера | Баг или исключение. Проверьте логи docker-compose logs backend |
| 502 Bad Gateway | Ошибка прокси | Nginx не может достучаться до backend / Rust worker |
| 503 Service Unavailable | Недоступно | Режим обслуживания или перегрузка |

Коды ошибок WebSocket

При закрытии WebSocket-соединения могут возвращаться коды:

| Код | Описание |
|------|--|
| 1000 | Нормальное закрытие |
| 1001 | Сервер уходит (перезапуск) |
| 1006 | Аномальное закрытие (потеря связи). Frontend автоматически переподключается. |
| 1008 | Нарушение политики (например, невалидный JWT в query-параметре) |
| 1011 | Внутренняя ошибка сервера |
| 4001 | Custom: Authentication failed |
| 4003 | Custom: Forbidden (нет прав на канал) |
| 4004 | Custom: Channel not found |

Типы условий перехода (Transition Condition Type)

| Код | Описание |
|--------|--|
| device | По значению устройства (сравнение с порогом) |
| timer | По длительности (после входа в шаг) |
| option | По нажатию опции оператором |

Операторы сравнения (для type='device'):

| Символ | Значение |
|--------|----------|
| == | Равно |
| != | Не равно |
| > | Больше |

| Символ | Значение |
|--------|------------------|
| < | Меньше |
| >= | Больше или равно |
| <= | Меньше или равно |

Действия с партией на шаге (batch_transfer_action)

| Код | Описание |
|----------|--|
| NONE | Без действия, партия остаётся на текущем этапе |
| TRANSFER | Передать партию на следующий этап линии |
| COMPLETE | Завершить партию (финальный статус) |

Типы форм крышек танков

Для поля SchemaDevice.configuration.tank_visual:

Нижние (`bottom_type`):

| Код | Описание |
|------|---------------------------------|
| FLAT | Плоское |
| DOME | Сферическое (выпуклое вниз) |
| DISH | Торосферическое (эллиптическое) |
| CONE | Коническое |

Верхние (`top_type`):

| Код | Описание |
|------|-----------------|
| FLAT | Плоское |
| DOME | Сферическое |
| DISH | Торосферическое |

Документ создан автоматически на основе анализа исходного кода. Для заполнения скриншотов: откройте приложение, перейдите на указанный экран и сделайте скриншот с описанными элементами.