


УТВЕРЖДАЮ

Генеральный директор

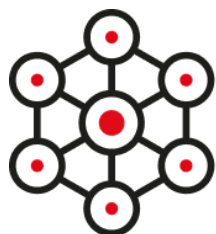
ООО «ЦР»

 Земсков С.А.

М.П.

« 06 » _____ ноября _____ 2025 г.

Распределенная система управления




МультиКонтроль

Описание программы

Москва, 2025 г.

Лист СОГЛАСОВАНИЯ

ФИО	Должность	Организация	Подпись
Земсков С.А.	Генеральный директор	ООО «ЦР»	

Оглавление

Перечень сокращений.....	4
Введение.....	5
1. Описание, назначение и цели создания Системы.....	6
1.1 Общее описание Системы.....	6
1.2. Область применения Системы.....	7
1.3. Назначение Системы.....	8
1.4. Цели создания Системы.....	8
2. Архитектура и функциональные возможности СВК РСУ.....	9
2.1 Цели и задачи автоматизации.....	9
2.2 Среда выполнения.....	10
2.2.2 Функциональные характеристики.....	11
2.2.3 Принципы работы РСУ.....	12
2.2.4 Обработка данных и вычисления.....	13
2.2.5 Управление узлами и отказоустойчивость.....	14
2.2.6 Репликация и синхронизация.....	15
2.2.7 Интеграция и диагностика.....	16
2.3.Среда конфигурирования.....	19
2.3.1 Реализация редактора конфигурации контроллера РСУ.....	19
2.3.2 Создание подсистемы РСУ.....	19
2.3.3 Добавление серверного шкафа в подсистему РСУ.....	21
2.3.4 Добавление крейта в серверный шкаф и работа с ним в визуальном редакторе.....	23
2.3.5 Добавление контроллера в крейт.....	28
2.3.6 Конфигурация контроллера РСУ.....	33
2.3.7 Визуальный редактор конфигуратора.....	41

Перечень сокращений

В настоящем документе применяются термины и сокращения, приведенные с соответствующими определениями и расшифровкой в таблице 1.

Таблица 1. Перечень терминов и сокращений

Сокращение	Описание
АСУ ТП	Автоматизированная система управления технологическими процессами
СВК РСУ	Среда выполнения и конфигурирования распределенных систем управления
ППИ	Прикладной программный интерфейс
SCADA	Supervisory Control and Data Acquisition
МЭК 61131-3	Международный стандарт 61131-3
ST	Structured Text
IL	Instruction List
LD	Ladder Diagram
FBD	Function Block Diagram
SFC	Sequential Function Chart
GbE	Gigabit Ethernet
САПР	Система автоматизированного проектирования
AI	Analog Input
AO	Analog Output
Digital Input	DI
DO	Digital Output
OPC	OLE for Process Control
ARM	Advanced RISC Machine
RISC-V	Reduced Instruction Set Computer V
MIPS	Microprocessor without Interlocked Pipelined Stages
Комдив	Компьютерная дивизия
Lua	Lua (язык программирования)
Cron	Command Run On

Введение

Настоящее описание программы (ОП) посвящено программному обеспечению среды выполнения и конфигурирования контроллеров автоматизированных систем управления технологическими процессами (АСУ ТП) для построения распределенных систем управления (СВК РСУ).

СВК РСУ представляет собой инновационный комплекс, ориентированный на автоматизацию сложных технологических процессов в крупных непрерывных производствах, таких как атомные и тепловые электростанции. Разработанное на базе контроллеров производства ООО «Электромеханика Атом», это ПО обеспечивает интеграцию локальной и распределенной логики управления, поддержку кластерных режимов, репликацию и синхронизацию данных в реальном времени, а также диагностику и конфигурирование систем с учетом строгих требований к надежности и совместимости.

Документ подготовлен в рамках Договора № 090125ЛТА/ЦР от 09 января 2025 г. между Заказчиком (ООО «ЦР») и Исполнителем (ООО «ЛТА»), с учетом плановых сроков разработки (с 09.01.2025 по 30.08.2025). Он детализирует структуру, функциональные характеристики и принципы работы СВК РСУ, опираясь на стандарты ГОСТ 19-й и 34-й серий, и служит основой для дальнейшей эксплуатации, испытаний и модернизации системы. Описание подчеркивает потенциал СВК как открытой платформы, способной к расширению для поддержки новых архитектур и языков программирования, что делает ее ключевым инструментом в импортозамещении и цифровизации промышленных объектов.

1. Описание, назначение и цели создания Системы

1.1 Общее описание Системы

СВК РСУ (Среда выполнения и конфигурирования распределенных систем управления) представляет собой комплексное программное решение, разработанное для автоматизации технологических процессов крупных непрерывных производств.

Система базируется на контроллерах производства ООО «Электромеханика Атом» и включает в себя две основные подсистемы: **среду выполнения** и **среду конфигурирования**.

Среда выполнения представляет собой ключевой компонент системы, обеспечивающий непосредственное управление технологическими процессами и выполнение алгоритмической логики.

Основные функции среды выполнения:

- Управление локальными и глобальными логическими переменными;
- Обработка данных портов ввода/вывода;
- Преобразование сигналов в требуемый формат;
- Выполнение вычислительных сценариев;
- Диагностика состояния контроллера.

Среда конфигурирования позволяет настраивать логику контроллеров, привязывать переменные к физическим портам ввода/вывода, задавать правила репликации, синхронизации и доступа к данным. Это включает визуальный редактор на базе SCADA «Лацерта», что упрощает проектирование систем для инженеров.

1.2. Область применения Системы

Система применима в сценариях, где требуется:

- **Локальное управление:** Выполнение логики на одном контроллере, включая чтение данных с портов ввода/вывода, их преобразование и выполнение вычислительных сценариев с использованием языка МЭК 61131-3 (ST) или современных встраиваемых языков сценариев;
- **Распределенное управление:** Координация работы нескольких контроллеров в кластерном режиме, поддержка параллельной и конвейерной обработки данных, а также балансировка вычислительной нагрузки между узлами. Это делает систему применимой в сетях с высокоскоростными интерконнектами (0.1–100 GbE) или при взаимодействии узлов на значительном удалении;
- **Интеграция с SCADA:** СВК РСУ обеспечивает взаимодействие с SCADA «Лацерта» (реестровая запись №17502 от 02.05.2023) через шлюзовые серверы или напрямую, что делает ее применимой в системах, требующих централизованного мониторинга и управления;
- **Высокая надежность и отказоустойчивость:** Система устойчива к отказам отдельных узлов и разделению сети, что критично для объектов, где остановка процессов недопустима (например, в энергетике).

1.3. Назначение Системы

Система предназначена для создания единой среды, обеспечивающей выполнение и конфигурирование логики управления в контроллерах АСУ ТП (автоматизированных системах управления технологическими процессами). Это ПО ориентировано на построение распределенных систем управления (PCY), где логика управления не ограничивается одним устройством, а распределяется по сети контроллеров.

Система разработана специально для контроллеров производства ООО «Электромеханика Атом», что делает ее адаптированной к оборудованию этого производителя. Она служит основой для создания проектов АСУ ТП, где требуется интеграция локальной (на уровне одного контроллера) и глобальной (на уровне всей сети) логики.

1.4. Цели создания Системы

Цели создания Системы:

- Автоматизация технологических процессов: создание программного обеспечения для автоматизации технологических процессов крупных непрерывных производств (таких как атомные и тепловые электростанции) со сложной распределенной логикой управления;
- Разработка средств для проектов АСУ ТП: формирование инструментов разработки и эксплуатации проектов автоматизированных систем управления технологическими процессами (АСУ ТП), включая распределенные системы управления (PCY), на базе контроллеров производства ООО «Электромеханика Атом»;
- Создание среды выполнения и конфигурирования: Обеспечение необходимости в разработке специализированной среды для выполнения и конфигурирования контроллеров ООО «Электромеханика Атом», позволяющей строить распределенные системы управления с поддержкой локальной и глобальной логики, репликации данных, синхронизации и диагностики.

2. Архитектура и функциональные возможности СВК РСУ

2.1 Цели и задачи автоматизации

Основные задачи создания Системы:

- **Автоматизация технологических процессов:** Создание программного обеспечения для автоматизации технологических процессов крупных непрерывных производств (таких как атомные и тепловые электростанции) со сложной распределенной логикой управления;
- **Разработка средств для проектов АСУ ТП:** Формирование инструментов разработки и эксплуатации проектов автоматизированных систем управления технологическими процессами (АСУ ТП), включая распределенные системы управления (РСУ), на базе контроллеров производства ООО «Электромеханика Атом»;
- **Создание среды выполнения и конфигурирования:** Обеспечение необходимости в разработке специализированной среды для выполнения и конфигурирования контроллеров ООО «Электромеханика Атом», позволяющей строить распределенные системы управления с поддержкой локальной и глобальной логики, репликации данных, синхронизации и диагностики.

2.2 Среда выполнения

Среда выполнения программного обеспечения СВК РСУ представляет собой кроссплатформенную подсистему, обеспечивающую выполнение локальной и распределенной логики управления на контроллерах ООО «Электромеханика Атом» (архитектура ARM) и серверах (архитектура x86) под управлением минимизированной ОС на основе Linux. Она реализует функции управления локальными и глобальными логическими переменными, чтение и преобразование сигналов с портов ввода/вывода, выполнение вычислительных сценариев на языках МЭК 61131-3 (ST) и современных встраиваемых языках сценариев, а также диагностику состояния контроллеров с передачей данных через формализованный прикладной программный интерфейс (ППИ).

В распределенных системах управления (PCU) среда выполнения поддерживает кластерный режим, координируя процессорные модули и шлюзовые серверы для параллельных и конвейерных вычислений, балансировки нагрузки и работы с узлами в высокоскоростных интерконнектах (0.1–100 GbE) или на удалении с высокими задержками. Она обеспечивает самоидентификацию узлов, репликацию (полную и частичную, включая шардинг), синхронизацию данных (до 1 млн синхронизаций/с, частота до 100 мс) в синхронном и асинхронном режимах, управление избыточностью, разграничением доступа и мажоритарной логикой (например, 2/3). Среда поддерживает доступ к удаленным переменным с кэшированием, минимизацию периода неконсистентности с использованием временных меток и диагностику параметров (загрузка процессора, сети, задержки, качество данных).

Среда выполнения устойчива к отказам узлов и разделению сети, обеспечивая высокую доступность и управляемую консистентность данных. Она интегрируется с SCADA «Лацерта» (реестр №17502 от 02.05.2023) через шлюзы или напрямую, поддерживает автоматизацию сборки прошивок и перспективное расширение для архитектур RISC-V, MIPS, Комдив, а также языков C и МЭК 61131-3 (IL, LD, FBD, SFC), что делает ее универсальной для автоматизации крупных непрерывных производств, таких как атомные и тепловые электростанции.

2.2.2 Функциональные характеристики

Среда выполнения отвечает за исполнение алгоритмической логики на узлах контроллеров и поддерживает как локальные, так и глобальные логические переменные.

Ключевые функции среды выполнения включают:

- Управление локальными логическими переменными на уровне отдельного контроллера;
- Управление глобальными логическими переменными на уровне всей РСУ;
- Чтение данных с портов ввода физических устройств;
- Преобразование сигналов портов ввода/вывода в требуемый формат данных и их запись в логические переменные (локальные или глобальные);
- Выполнение вычислительных сценариев с использованием логических переменных;
- Преобразование значений логических переменных в формат для вывода;
- Запись данных в порты вывода.

Логика работы контроллеров поддерживается на базе языка МЭК 61131-3 (ST) для локальной логики одного контроллера, а также на базе современного встраиваемого языка сценариев (например, Lua) для локальной и распределенной логики на нескольких контроллерах. Встраиваемое специальное ПО (СПО) обеспечивает диагностику контроллера с передачей информации о состоянии во внешнюю систему через формализованный прикладной программный интерфейс (ППИ).

2.2.3 Принципы работы PCS

В распределенном режиме (PCS) среда выполнения поддерживает кластерный режим работы контроллеров, объединенных в сеть (рис.1).

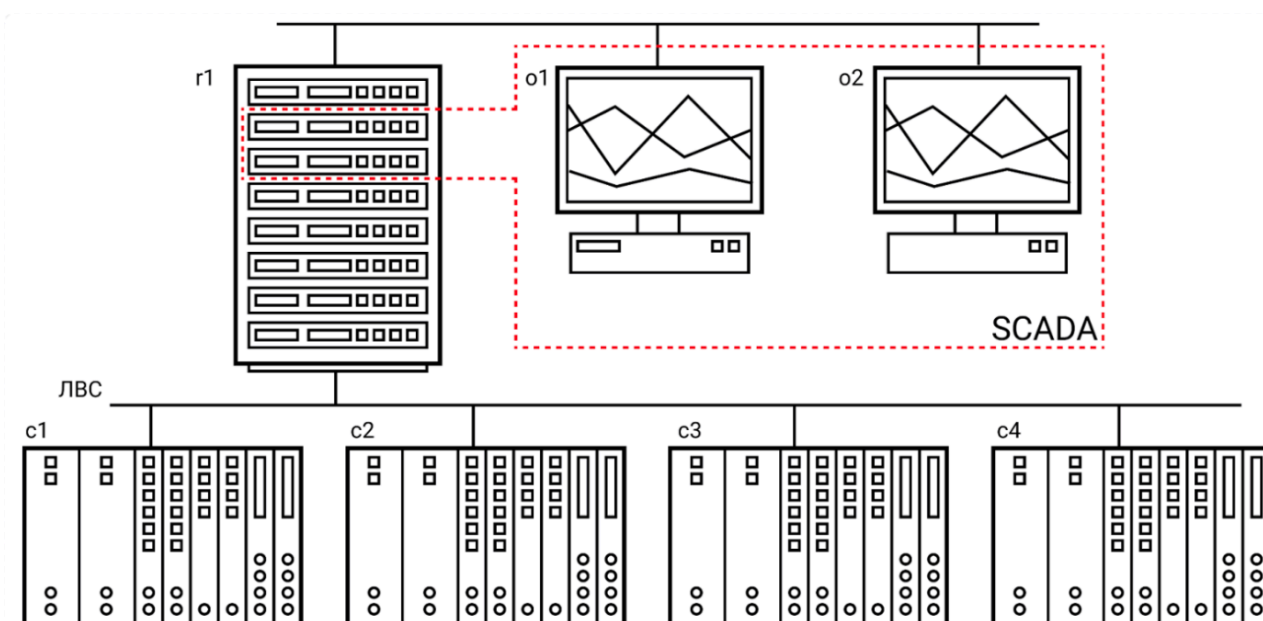


Рис. 1. Схема ACS ТП с PCS

Кластер является программно-определяемым и обеспечивает:

- Координацию среды выполнения на процессорных модулях контроллера и шлюзовых серверных устройствах;
- Обслуживание алгоритмов параллельных и конвейерных вычислений;
- Управление балансировкой вычислительной нагрузки на узлы кластера.

PCS поддерживает работу с процессорными модулями, расположенными как в высокоскоростном надежном интерконнекте (с низкими задержками), так и на значительном удалении (с высокими задержками). Система обслуживает узлы контроллеров (c1–c4), взаимодействующие по выделенному высокоскоростному интерконнекту или аппаратной шине данных (Рис. 1). Шлюзовые серверы координируют узлы кластера и обеспечивают доступ SCADA «Лацрта» (o1, o2) к данным системы. Возможен прямой доступ SCADA к контроллерам через процессорные модули, минуя шлюзы, при достаточной производительности.

Функционал РСУ включает:

- Самоидентификацию узлов кластера;
- Обслуживание логических переменных на уровне узла;
- Репликацию и синхронизацию значений логических переменных между узлами;
- Управление избыточностью переменных;
- Поддержку синхронных и асинхронных режимов;
- Управление разграничением доступа к переменным;
- Идентификацию деградации узлов и их исключение из работы;
- Подписку на изменения переменных узлов;
- Выполнение алгоритмов с параллельной (рис. 3.2 ТЗ) и конвейерной (рис. 3.3 ТЗ) обработкой данных;
- Мажоритарную логику (например, 2/3, 3/4) как разновидность параллельной логики.

2.2.4 Обработка данных и вычисления

Работа РСУ включает управление глобальными логическими переменными, распределенными между узлами, с использованием алгоритмов параллельных и конвейерных вычислений, как изображено на предлагаемых схемах (рис. 2 и 3).

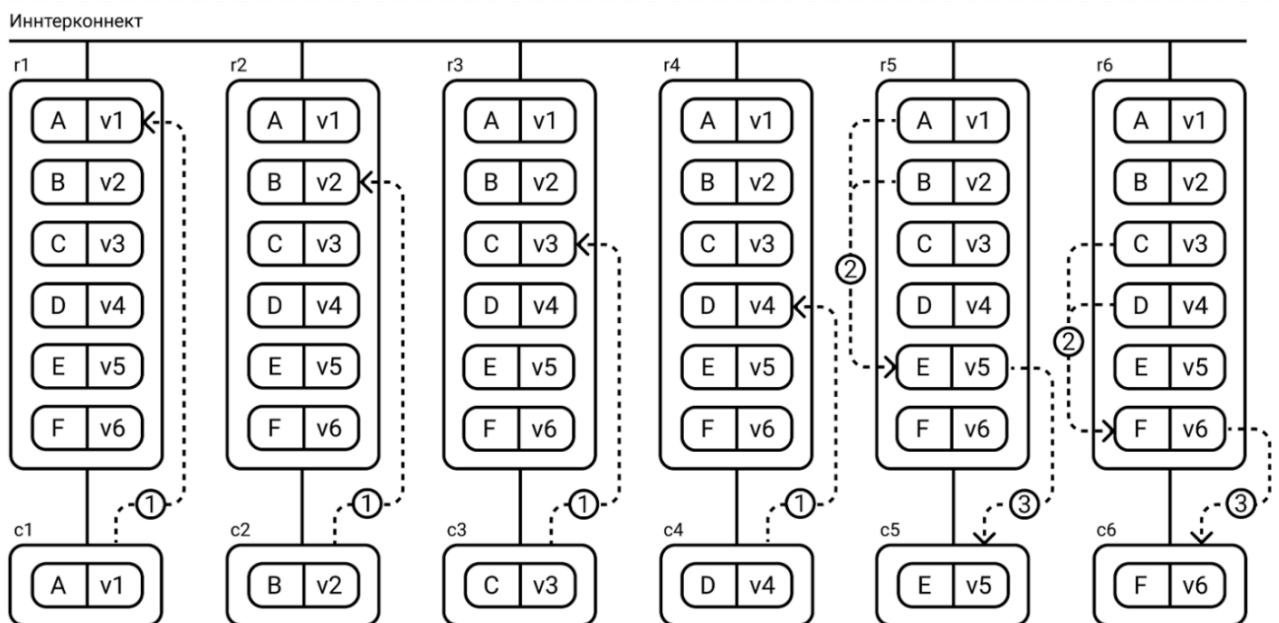


Рис. 2. Параллельный режим работы РСУ



Рис. 3. Конвейерный режим работы РСУ

Среда выполнения обеспечивает чтение сигналов с портов ввода физических устройств, их преобразование в требуемый формат и запись в переменные, а также выполнение вычислительных сценариев на языках МЭК 61131-3 (Structured Text, ST) и современных встраиваемых языках сценариев. Результаты вычислений преобразуются в формат для вывода и записываются в порты вывода, обеспечивая непрерывное управление процессами.

2.2.5 Управление узлами и отказоустойчивость

Среда выполнения управляет балансировкой вычислительной нагрузки между узлами, включая процессорные модули с частичной репликацией и шлюзовые серверы с полной репликацией, как показано на предложенной схеме (рис. 4).

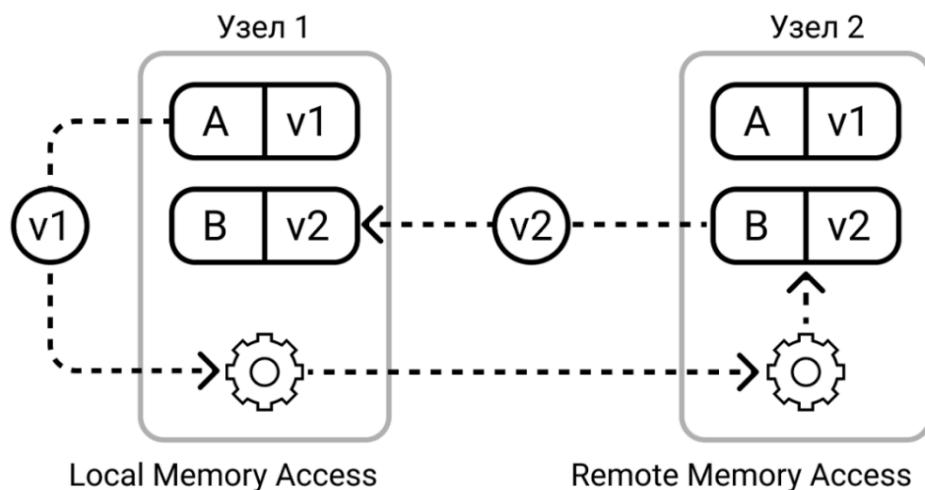


Рис.4. Доступ к удаленной переменной

Она поддерживает разграничение доступа к переменным, идентификацию деградации узлов и их исключение из работы, а также реализацию мажоритарной логики (например, 2/3 или 3/4) для обеспечения корректности результатов. РСУ устойчива к отказам отдельных узлов и разделению сети, сохраняя высокую доступность и управляемую консистентность данных.

2.2.6 Репликация и синхронизация

РСУ под управлением среды выполнения поддерживает репликацию данных, включая полную и частичную (шардинг), а также копирование переменных между узлами для обеспечения избыточности и доступности, как описано в контексте предлагаемых схем (рис. 5 и 6).

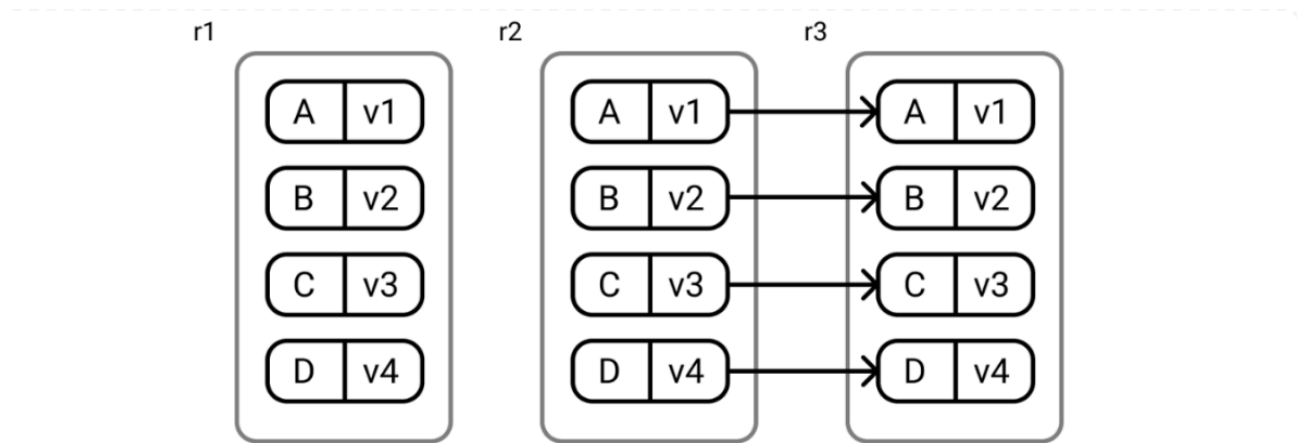


Рис. 5. Репликация

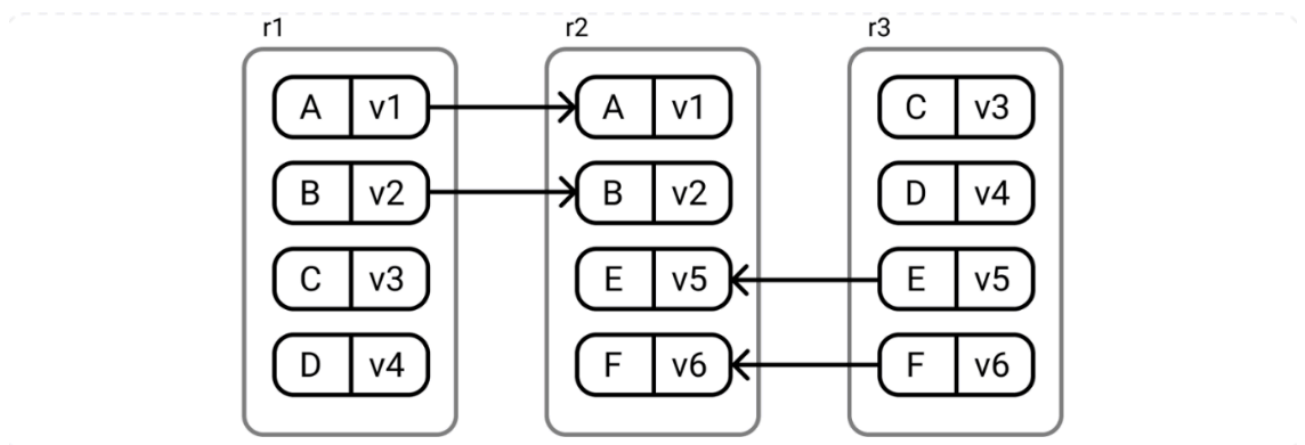


Рис. 6. Сложная схема репликации

Синхронизация реплицированных переменных обеспечивает актуальность и целостность данных с минимальными задержками, без значительного ухудшения производительности. Она осуществляется в синхронном (с заданной периодичностью, например, каждые 100 мс) и асинхронном (по событиям) режимах, с параметрами:

- До 10 000 синхронизаций в секунду на один процессорный модуль;
- До 1 млн синхронизаций в секунду в нормальном режиме эксплуатации;
- До 5 млн синхронизаций в секунду для серверного интерконнекта.

Консистентность данных реализуется в режиме «строгой консистентности в конечном счете» с использованием временных меток, минимизацией периода неконсистентности и диагностикой по заданным периодам (рис. 7). Приоритезация данных осуществляется в конфигурационном и автоматическом режимах, с учетом типов узлов (процессорные модули с частичной репликацией и шлюзовые серверы с полной репликацией).

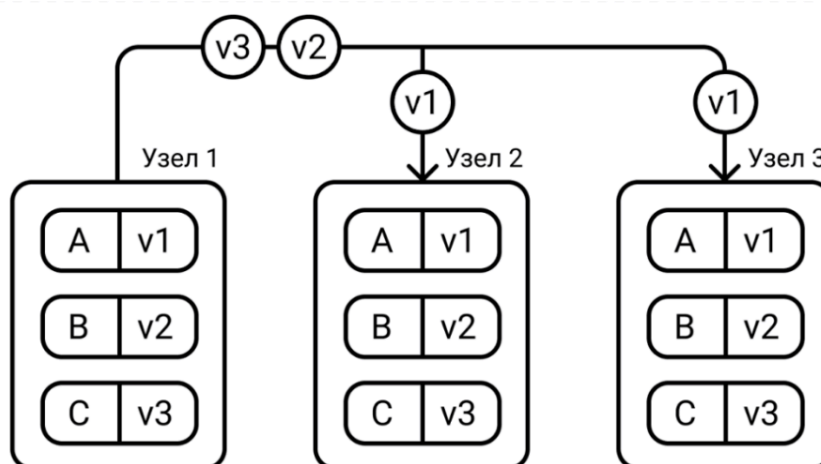


Рис. 7. Схема синхронизации данных

2.2.7 Интеграция и диагностика

PCU интегрируется с SCADA «Лацрта» (реестр №17502 от 02.05.2023) через шлюзовые серверы или напрямую, предоставляя доступ к данным и событийной модели. Диагностика включает мониторинг загрузки процессора и сетевых соединений, измерение задержек синхронизации, анализ времени обработки алгоритмов и оценку качества данных, с сохранением информации об аварийных ситуациях для последующего анализа. Это обеспечивает оперативное обнаружение и устранение неисправностей в реальном времени.

В качестве примера работы Среды выполнения будет приведен следующий распределенный алгоритм (рис. 8).

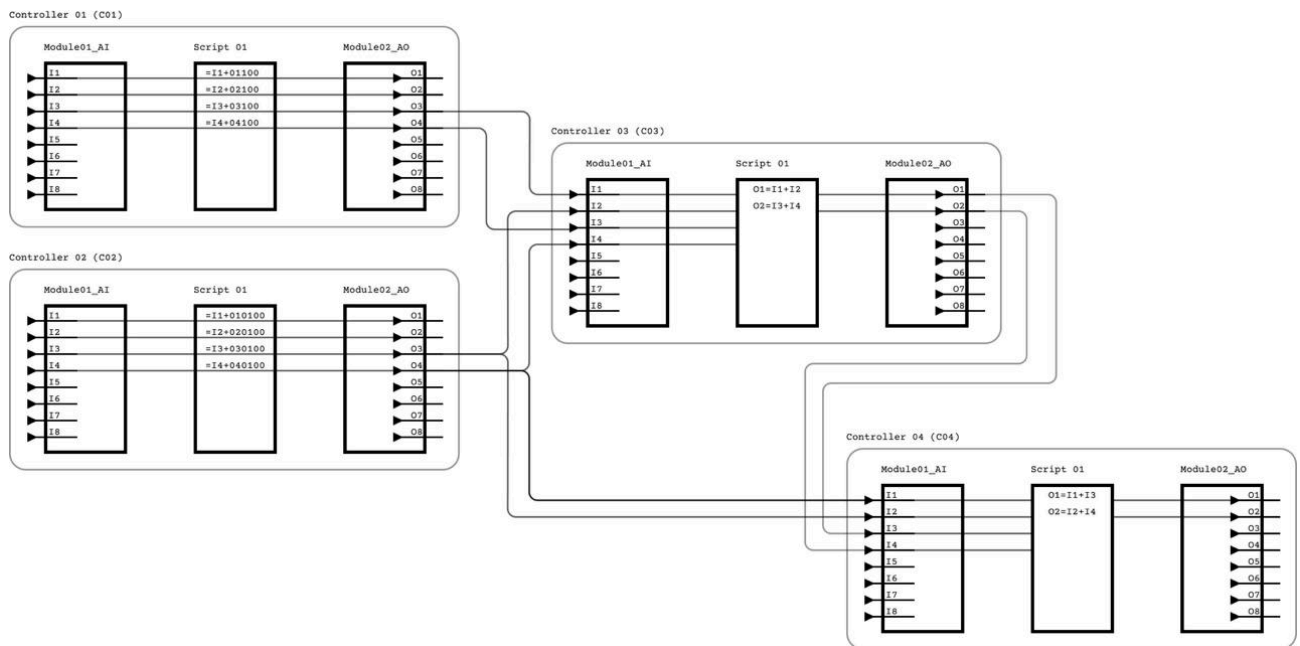


Рис. 8. Схема алгоритма

Общая структура

Система управления построена на базе четырёх контроллеров (C01–C04), каждый из которых включает:

- Модуль AI (ввода): принимает аналоговые сигналы I1–I8;
- Сценарий Script 01: выполняет вычисления и преобразования сигналов;
- Модуль АО (вывода): формирует выходные сигналы O1–O8.

Контроллеры соединены между собой: часть выходов одних контроллеров используется в качестве входов для других.

1. Алгоритм контроллера C01 и C02:

На вход подаются сигналы I1–I4.

В сценарии Script 01 выполняются вычисления:

$$O1 = I1 + 01100$$

$$O2 = I2 + 02100$$

$$O3 = I3 + 03100$$

$$O4 = I4 + 04100$$

Сигналы O1–O4 формируются на выходах модуля АО и передаются:

- локально на исполнительные устройства,
- в контроллеры C03 и C04 для дальнейшей обработки.

2. Алгоритм контроллера C03:

На вход подаются выходные сигналы от контроллеров C01 и C02.

В сценарии Script 01 выполняются операции суммирования:

$$O1 = I1 + I2$$

$$O2 = I3 + I4$$

Полученные результаты формируются на выходах модуля АО и подаются в контроллер C04.

3. Алгоритм контроллера C04

На вход поступают сигналы от контроллеров C01, C02 и C03.

В сценарии Script 01 выполняются вычисления:

$$O1 = I1 + I3$$

$$O2 = I2 + I4$$

Итоговые результаты формируются на выходах модуля АО и подаются в исполнительные устройства.

Итоговый принцип работы

Контроллеры C01 и C02 формируют промежуточные выходные сигналы, прибавляя к каждому входному значению постоянные коэффициенты (смещения).

Контроллер C03 получает данные от C01 и C02, суммирует их попарно и формирует новые выходные сигналы.

Контроллер C04 объединяет входные сигналы от всех предыдущих контроллеров и выполняет дополнительные операции суммирования, формируя итоговые выходные управляющие воздействия.

Таким образом, система реализует многоуровневую обработку сигналов по следующему принципу:

Входные сигналы → корректировка коэффициентами → промежуточные выходы → суммирование → финальные управляющие сигналы.

2.3.Среда конфигурирования

2.3.1 Реализация редактора конфигурации контроллера PCY

Подсистема PCY представляет собой структурный элемент в САПР PCY, предназначенный для организации и управления конфигурацией распределенной системы управления.

Подсистема позволяет добавлять серверные шкафы, крейты и другие модули, а также удалять их с опцией удаления вложенных элементов через модальное окно. В контексте среды конфигурирования подсистема служит контейнером для визуального редактирования и настройки компонентов PCY, включая графические примитивы модулей.

2.3.2 Создание подсистемы PCY

Для создания подсистемы в среде проектирования (САПР) PCY необходимо воспользоваться кнопкой «Создать» (рис.9).

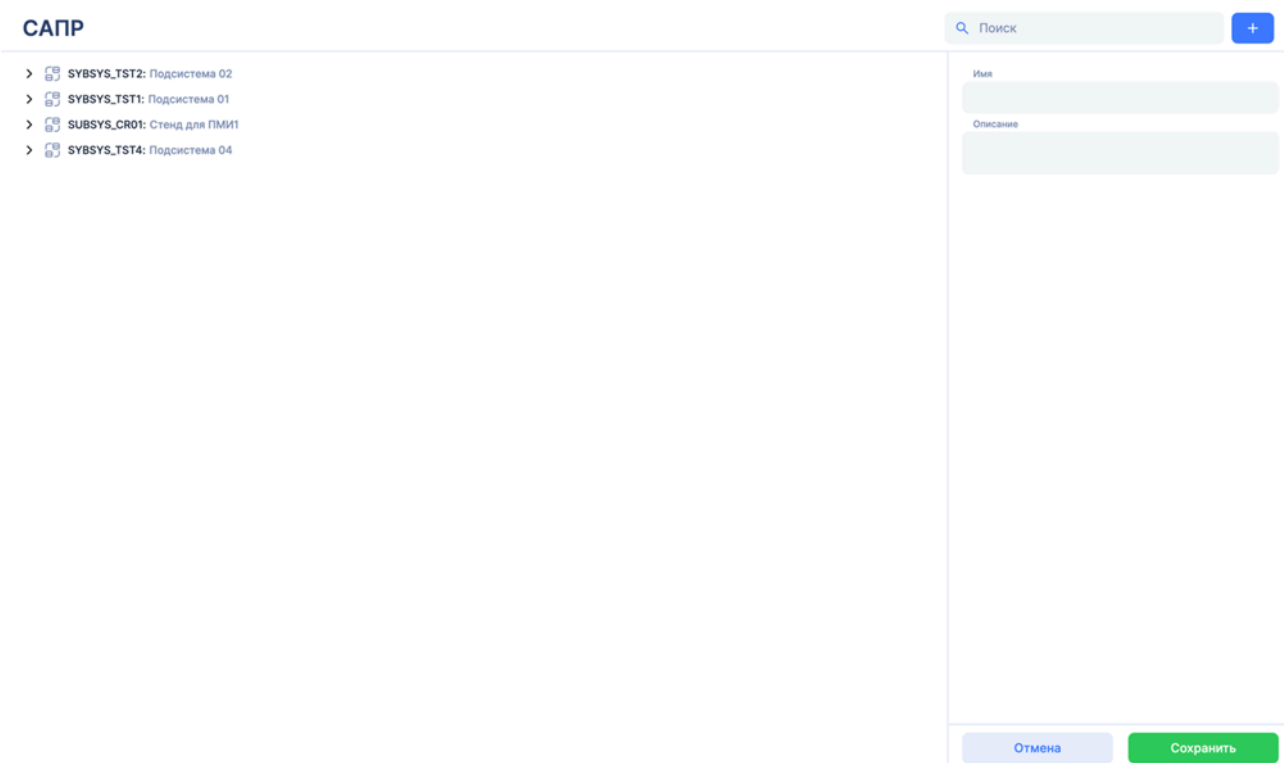


Рис. 9. Интерфейс создания подсистемы PCY

После заполнения полей «Имя», «Описание» и нажатия кнопки «Сохранить» подсистема отобразится в общем списке подсистем (рис. 10). В дальнейшем параметры могут быть отредактированы.

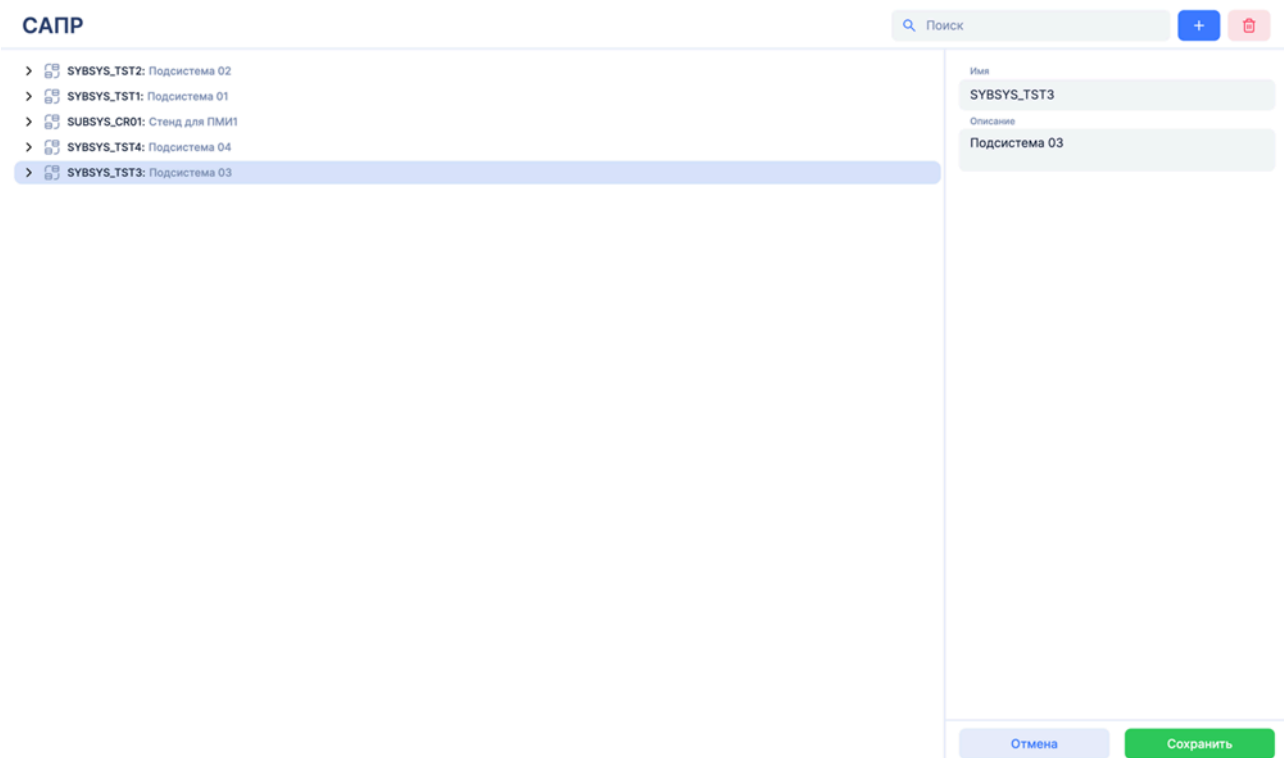


Рис. 10. Интерфейс просмотра и редактирования подсистемы PCY

Для удаления созданной ранее подсистемы необходимо воспользоваться кнопкой «Удалить». В модальном окне удаления подсистемы при необходимости можно активировать чекбокс «Удалить все вложенные папки» (рис.11).

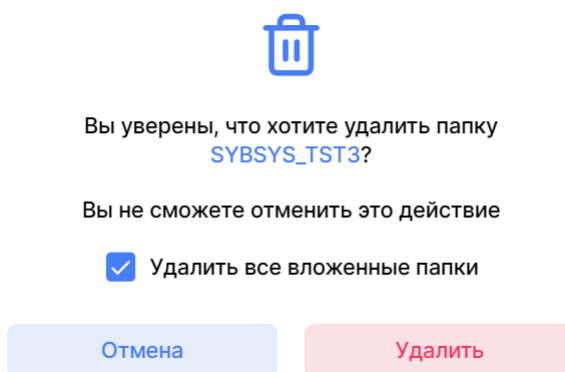


Рис. 11. Модальное окно удаления подсистемы PCY

2.3.3 Добавление серверного шкафа в подсистему РСУ

Для добавления серверного шкафа необходимо открыть отдельное окно кнопкой «+ Новый шкаф». После заполнения значений в полях «Имя», «Описание» и «Размер» следует нажать кнопку «Сохранить» (рис. 12).

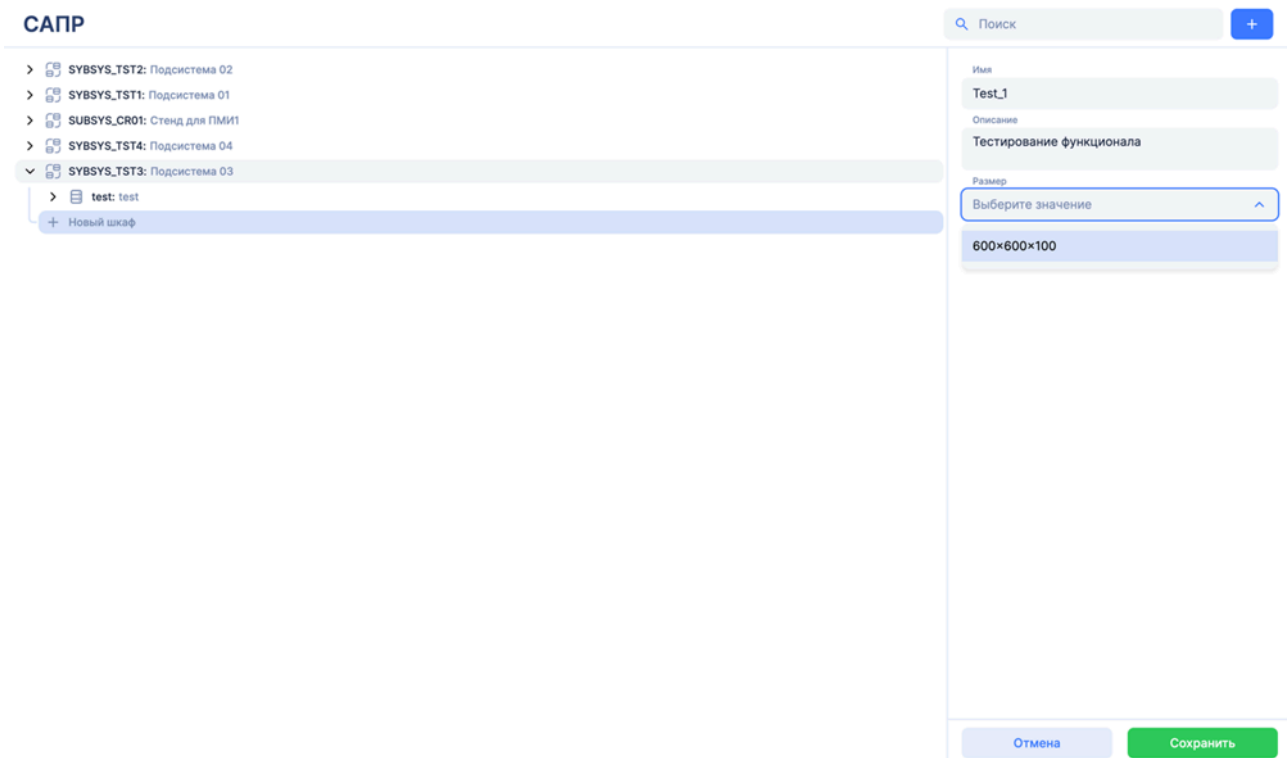


Рис. 12. Интерфейс добавления серверного шкафа в подсистему РСУ

Добавленный серверный шкаф отобразится в подсистеме РСУ. Поля «Имя», «Описание» и «Размер» могут быть отредактированы в дальнейшем (рис.13).



Рис.13. Интерфейс просмотра и редактирования серверного шкафа

Для открытия визуального редактора конфигурации необходимо воспользоваться кнопкой «Открыть визуализацию» (рис. 13).

Серверный шкаф в визуальном редакторе будет содержать те же поля, что и в подсистеме PCY («Имя», «Описание» и «Размер», рис. 14).

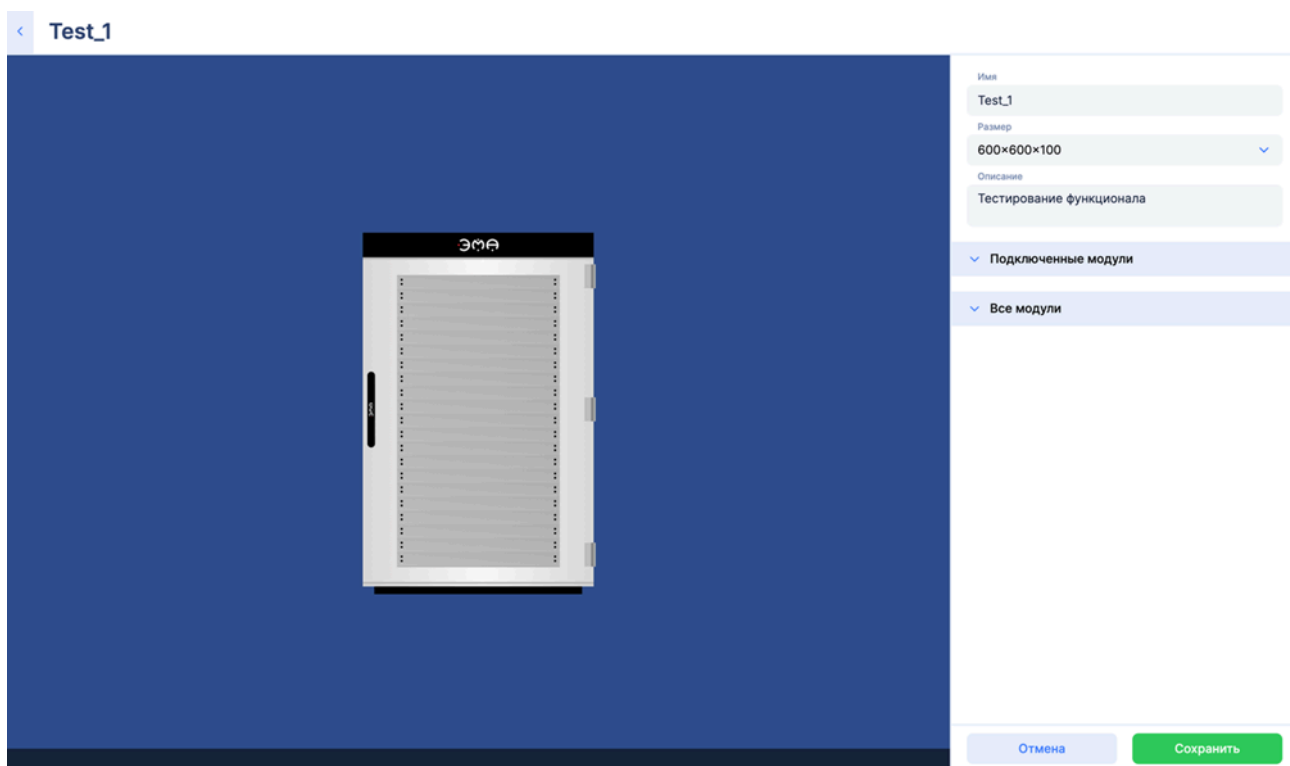


Рис. 14. Визуальный редактор конфигурации контроллера PCY

Визуальный редактор включает список подключенных к данному серверному шкафу модулей, а также список всех модулей в виде графических примитивов (рис. 15).

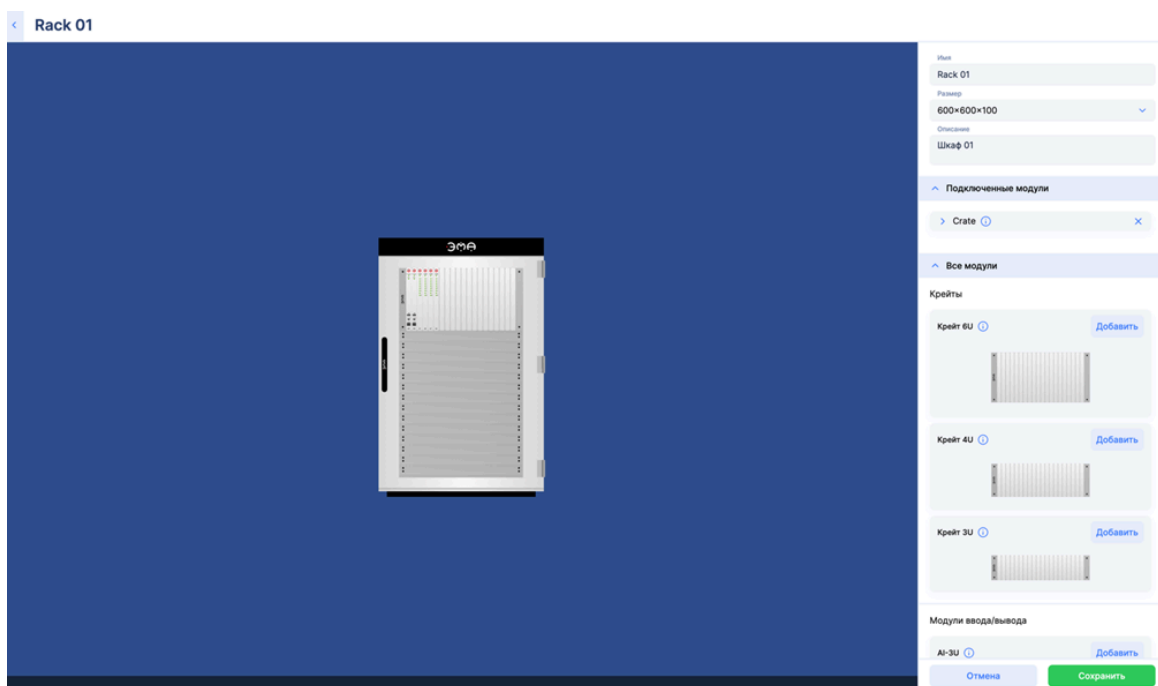


Рис. 15. Пример шкафа с подключенным модулем

2.3.4 Добавление крейта в серверный шкаф и работа с ним в визуальном редакторе

Для добавления крейта необходимо нажать кнопку «Новый крейт», предварительно развернув шкаф, в который требуется добавить крейт (рис.16).

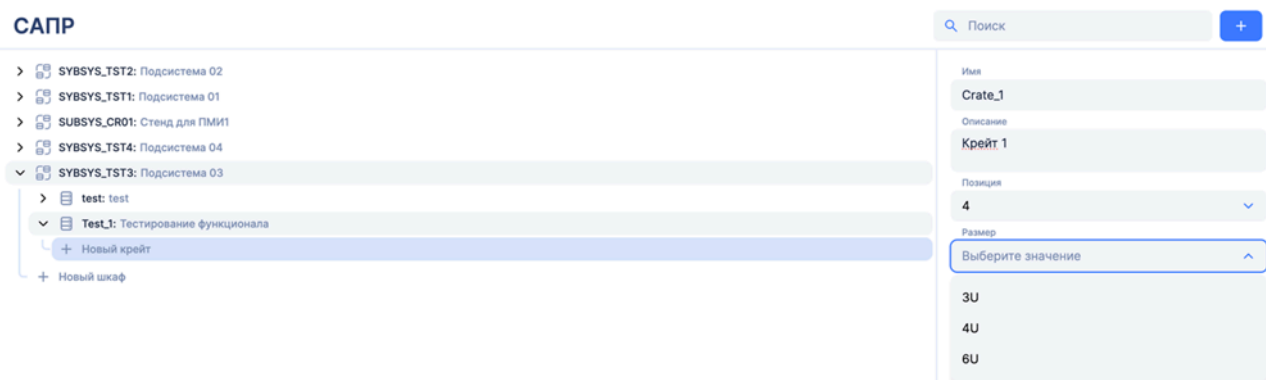


Рис. 16. Интерфейс добавления нового крейта в серверный шкаф

После заполнения полей «Имя», «Описание», выбора позиции и размера крейта в шкафу, а также нажатия кнопки «Сохранить», крейт отобразится под выбранным шкафом (рис.17).

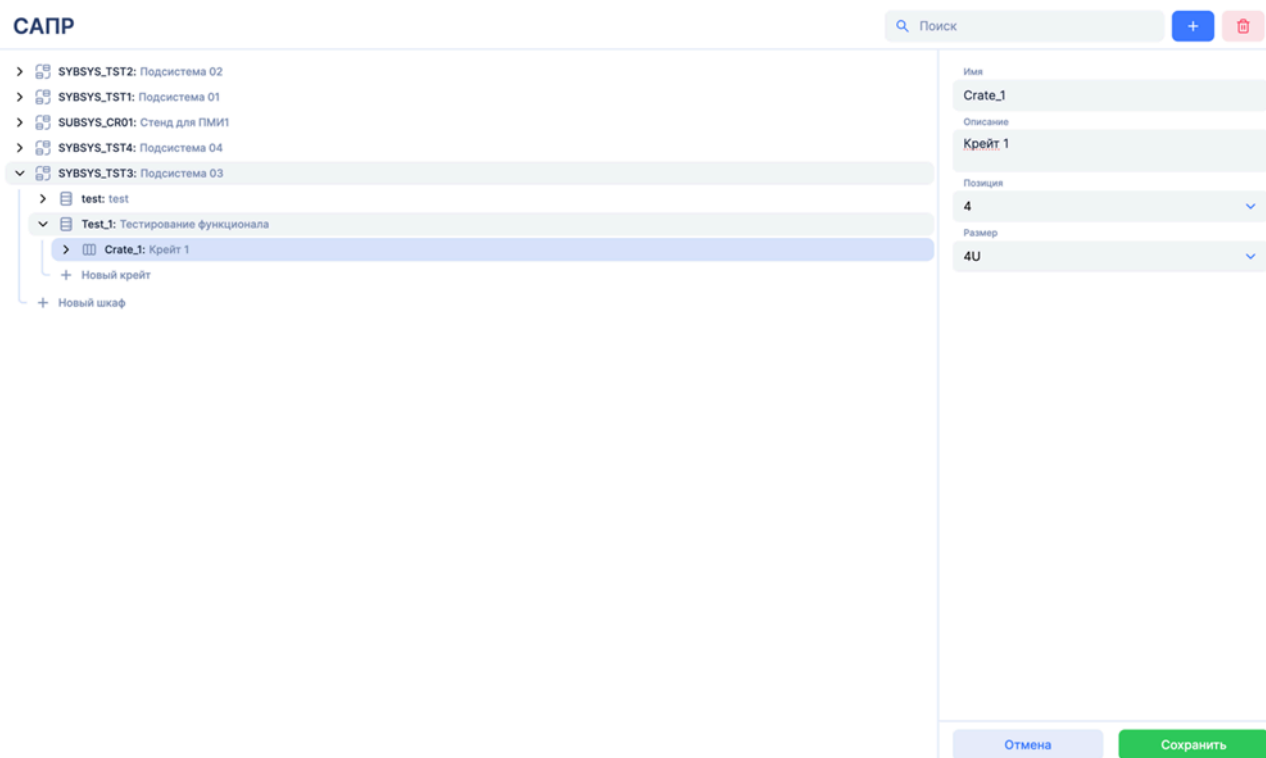


Рис. 17. Интерфейс просмотра и редактирования крейта

Для просмотра добавленного крейта в визуальном редакторе необходимо воспользоваться кнопкой «Открыть визуализацию» в окне просмотра и редактирования серверного шкафа. Добавленный крейт отобразится в серверном шкафу, а также под спойлером «Подключенные модули» (рис.18).

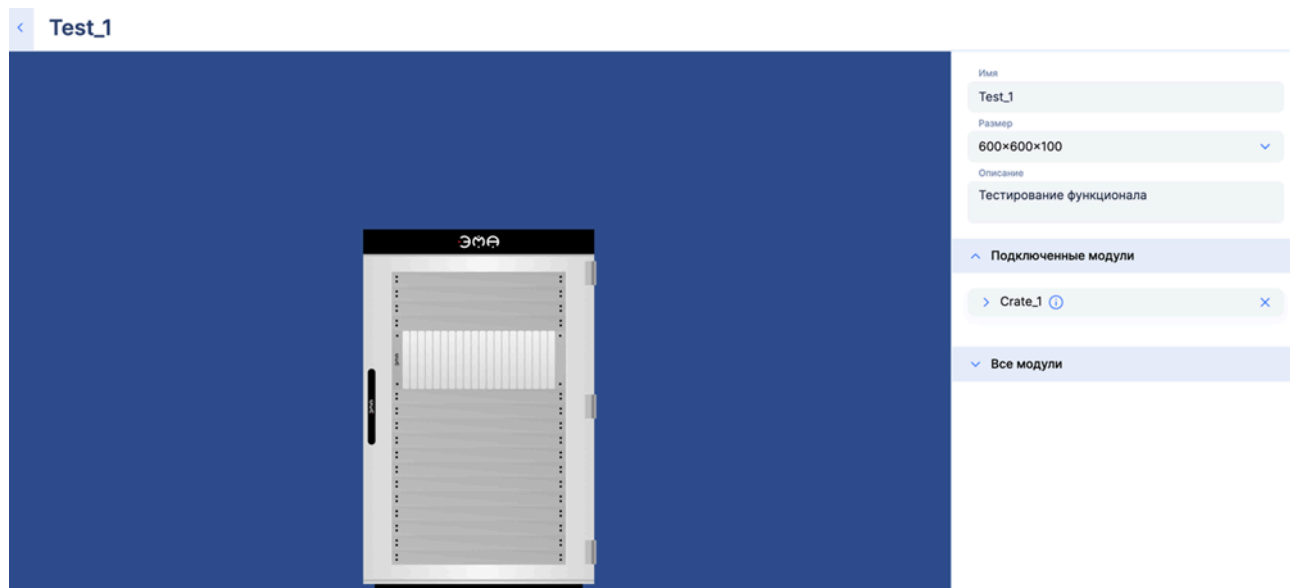


Рис. 18. Визуальный редактор конфигурации контроллера PCY

Новый крейт может быть добавлен и напрямую из визуального редактора. Для этого необходимо выбрать требуемый крейт из списка крейтов, расположенных под спойлером «Все модули» и нажать кнопку «Добавить» (рис.19).

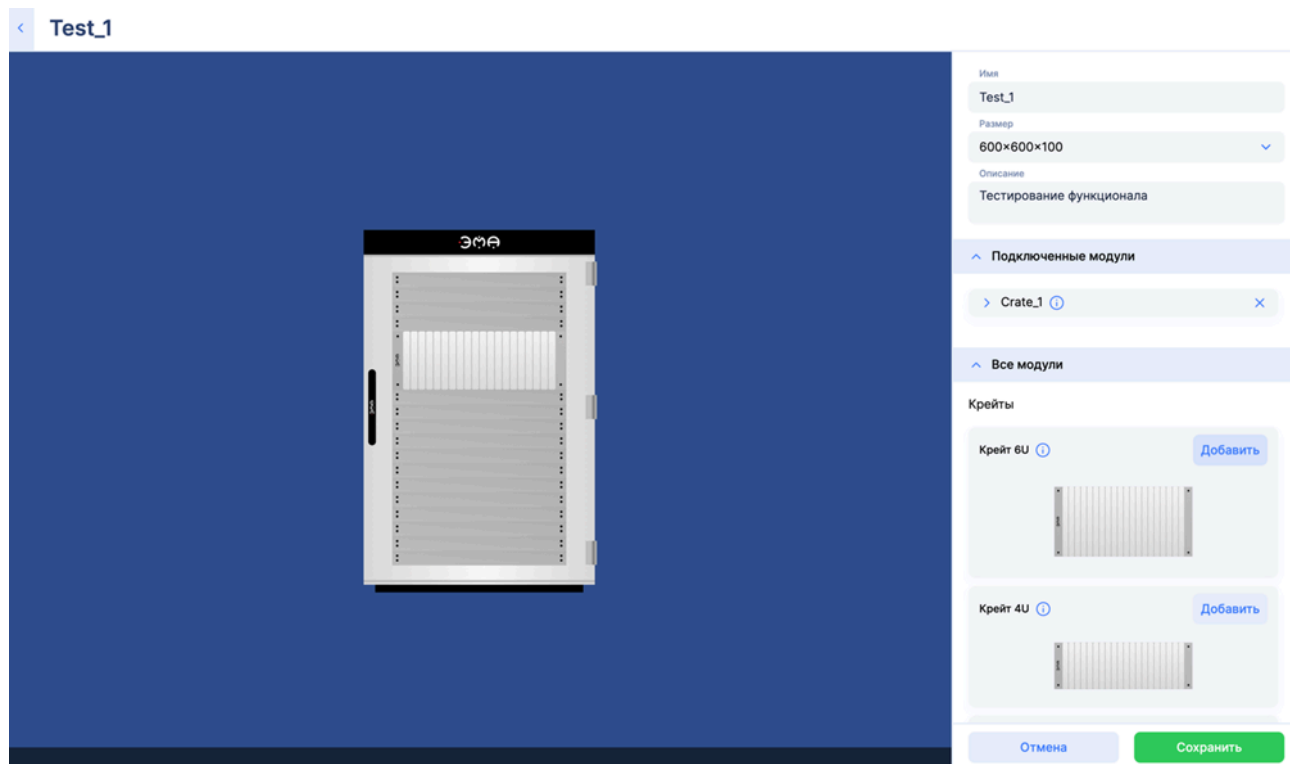


Рис. 19. Добавление крейта в шкаф через визуальный редактор

Выбранный крейт отобразится в серверном шкафу, а также отобразится в «Подключенных модулях» (рис. 20).

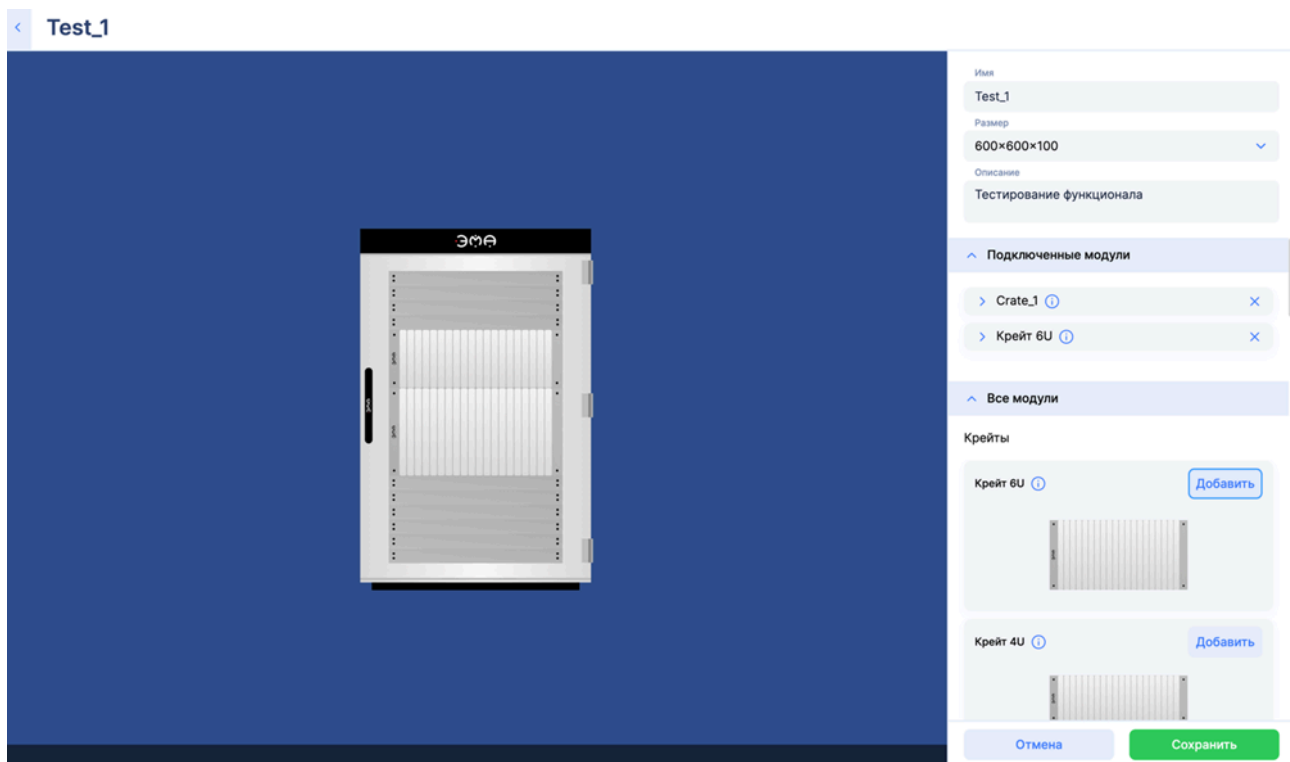


Рис. 20. Добавление крейта в шкаф через визуальный редактор

После нажатия кнопки «Сохранить» крейт отобразится в списке крейтов серверного шкафа (рис. 21).

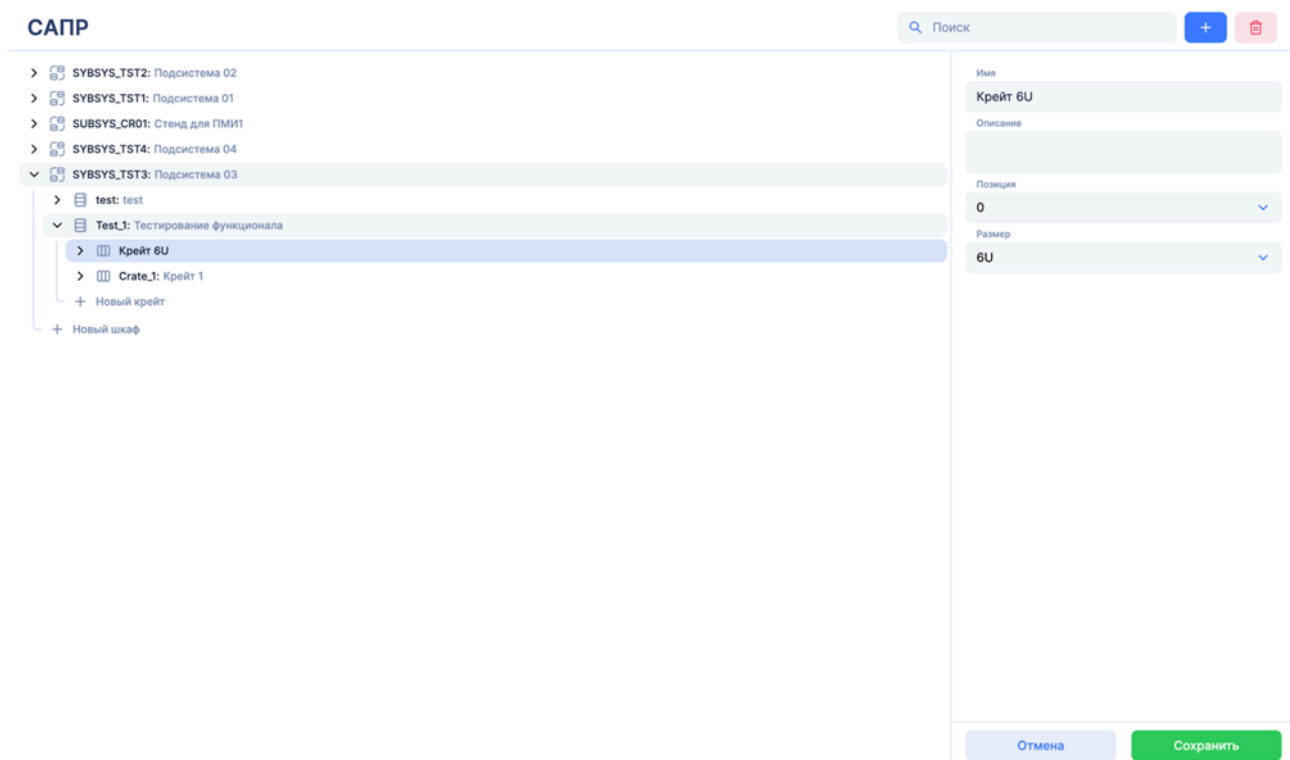


Рис. 21. Добавление крейта в шкаф через визуальный редактор

Перемещать (изменять позицию) крейта в серверном шкафу можно напрямую из визуального редактора при помощи метода «drag-and-drop». Для этого необходимо захватить элемент, нажав на кнопку «+», расположенную в правом верхнем углу крейта, и переместить крейт в желаемое место (рис. 22).

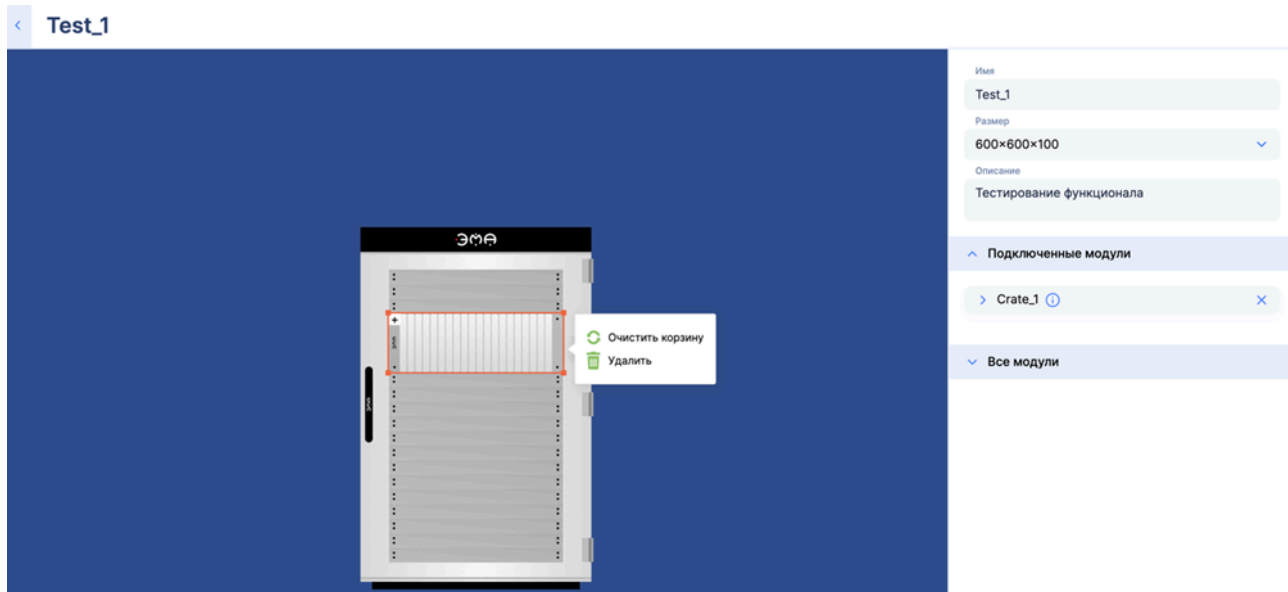


Рис. 22. Перемещение (изменение позиции) крейта через визуальный редактор
В результате изменения позиции крейт отобразится в новом месте (рис. 23).

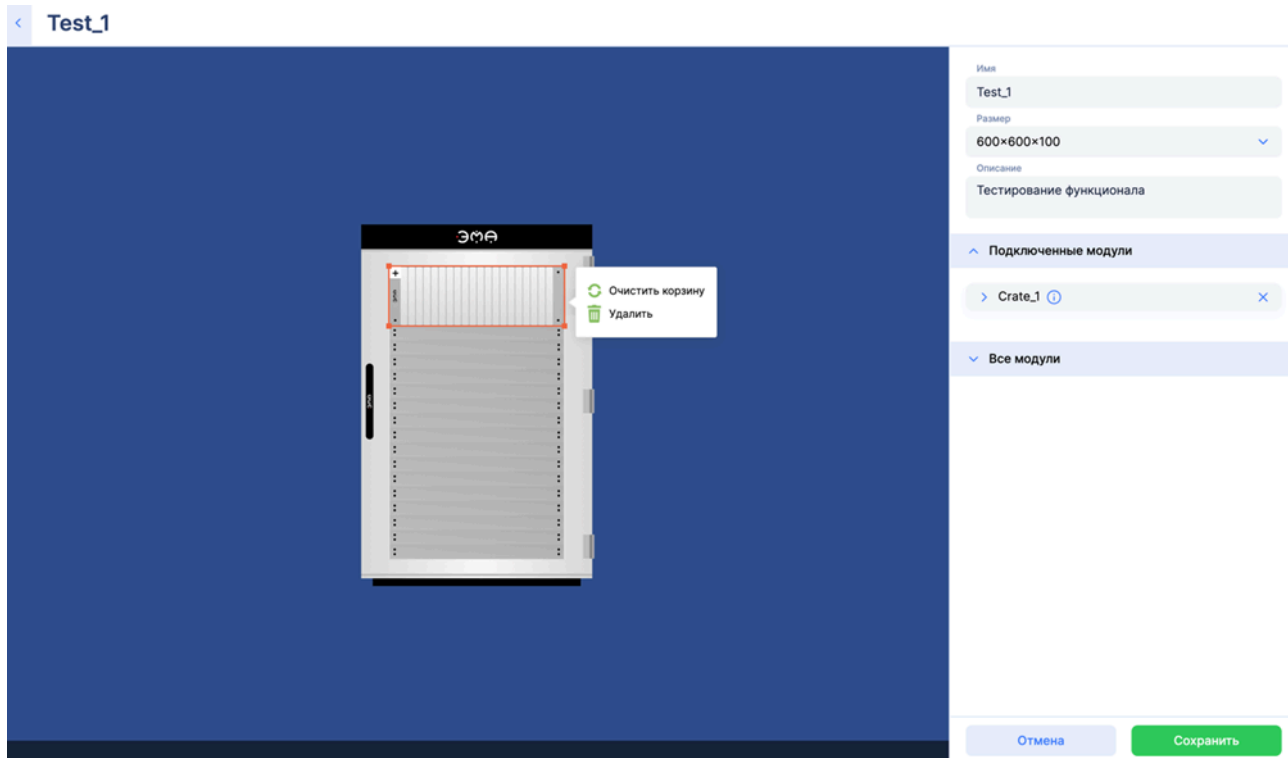


Рис. 23. Перемещение (изменение позиции) крейта через визуальный редактор

Меняется также и значение в поле «Позиция» (на примере, представленном на рисунках 17 и 21, с 4 на 0, рис. 24).



Рис. 24. Перемещение (изменение позиции) крейта через визуальный редактор

Для удаления крейта необходимо воспользоваться кнопкой «Удалить», повторно подтвердив удаление в модальном окне (рис. 25).

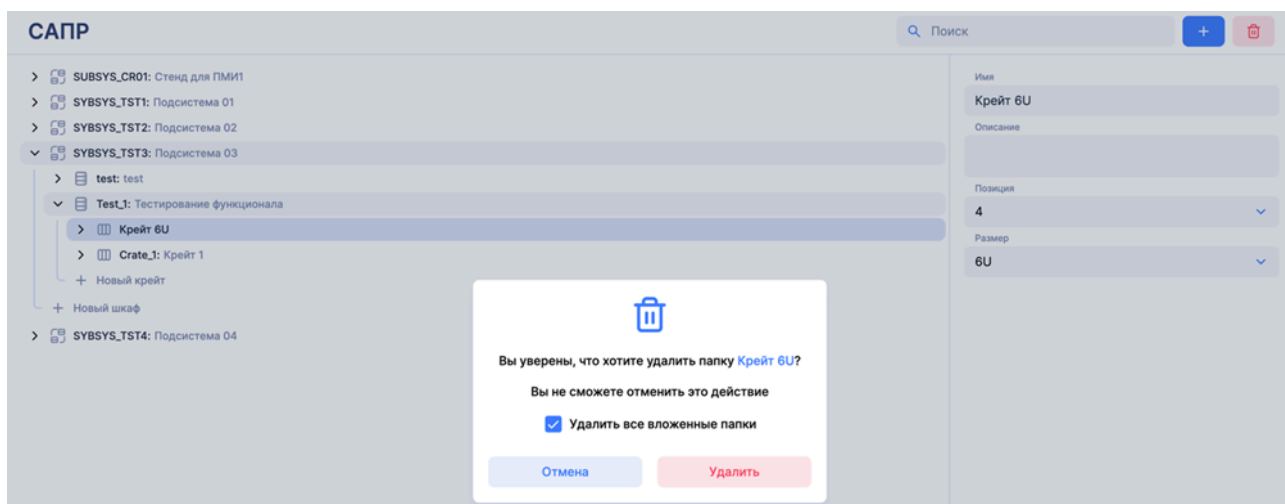


Рис. 25. Модальное окно удаление крейта

Крейт может быть удален и через визуальный редактор, для этого необходимо навести курсор на крейт и нажать кнопку «Удалить». Подтвердить удаление следует кнопкой «Сохранить» (рис. 26).

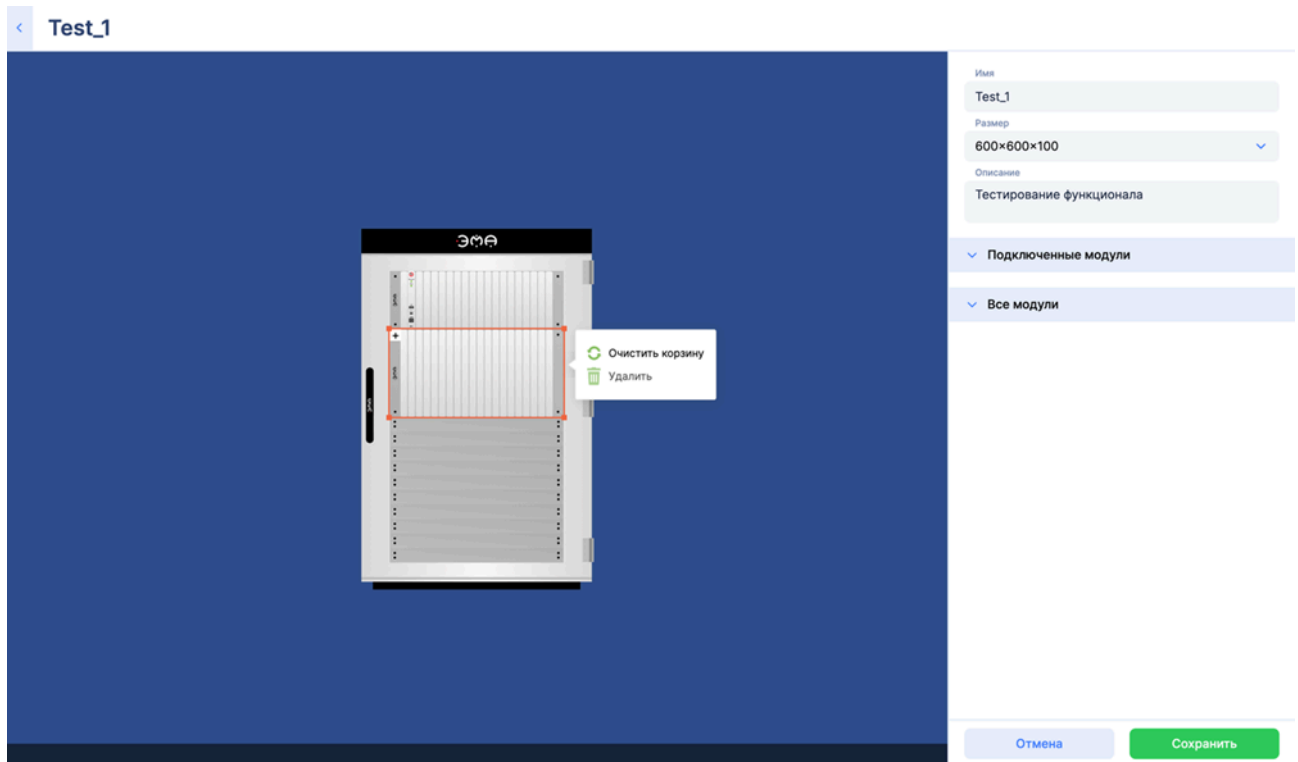


Рис. 26. Удаление крейта через визуальный редактор.

2.3.5 Добавление контроллера в крейт

Чтобы добавить контроллер в крейт необходимо нажать кнопку «Новый контроллер» (рис. 27).



Рис. 27. Интерфейс добавления нового контроллера в крейт

Далее необходимо заполнить поля «Имя», «Описание», «IP», «Порт», выбрать значение в поле «Позиция». В полях «IP» и «Порт» указываются IP-адрес и порт узла, на котором развернут экземпляр среды исполнения. После нажатия кнопки «Сохранить» контроллер отобразится под крейтом (рис. 28).

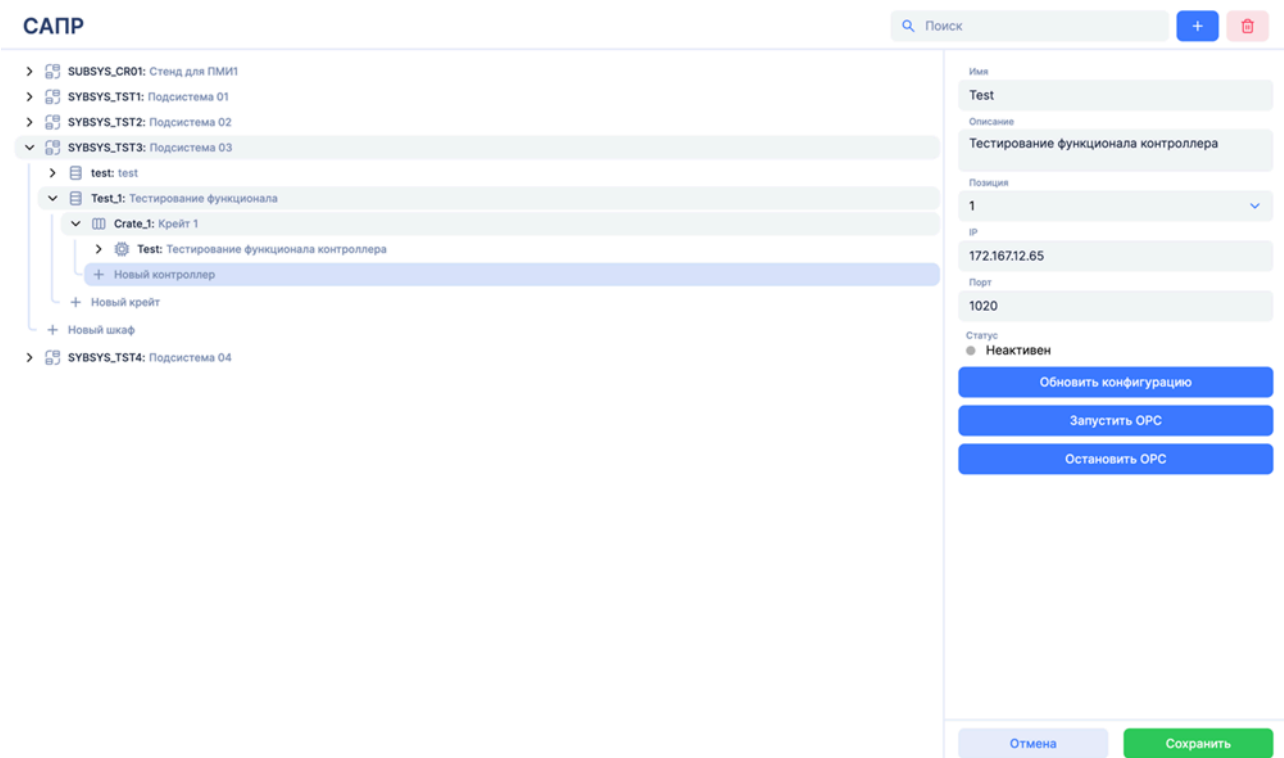


Рис. 28. Окно просмотра и редактирования контроллера

Если значения указаны верно и прошивка процессорного модуля доступна, то статус обновится на «Активен» (рис. 29).



Рис. 29. Пример контроллера со статусом «Активен»

Обновить конфигурацию контроллера можно кнопкой «Обновить конфигурацию». С помощью кнопок «Запустить OPC» и «Остановить OPC», можно запустить и остановить OPC-сервер (рис. 29).

Добавленный контроллер также отобразится и в визуальном редакторе с графическим примитивом под спойлером крейта (рис. 30).

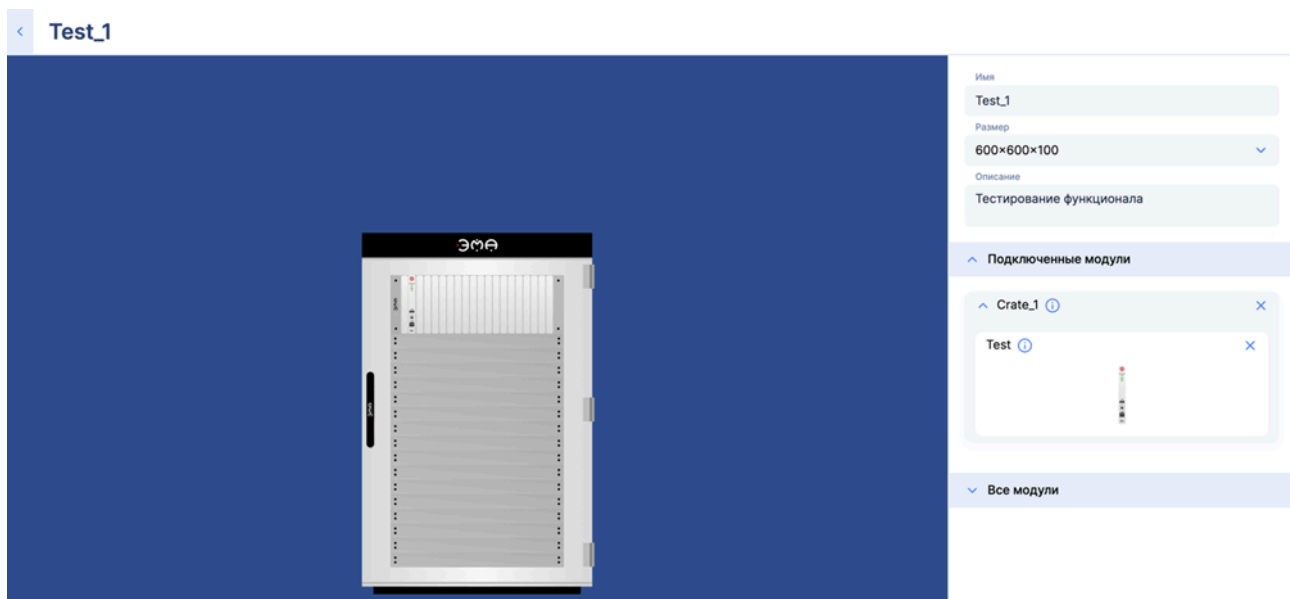


Рис. 30. Новый контроллер в визуальном редакторе

Контроллер также может быть добавлен через визуальный редактор. Для этого в списке «Модули процессорные» под спойлером «Все модули» необходимо выбрать модуль и нажать кнопку «Добавить» (рис. 31).

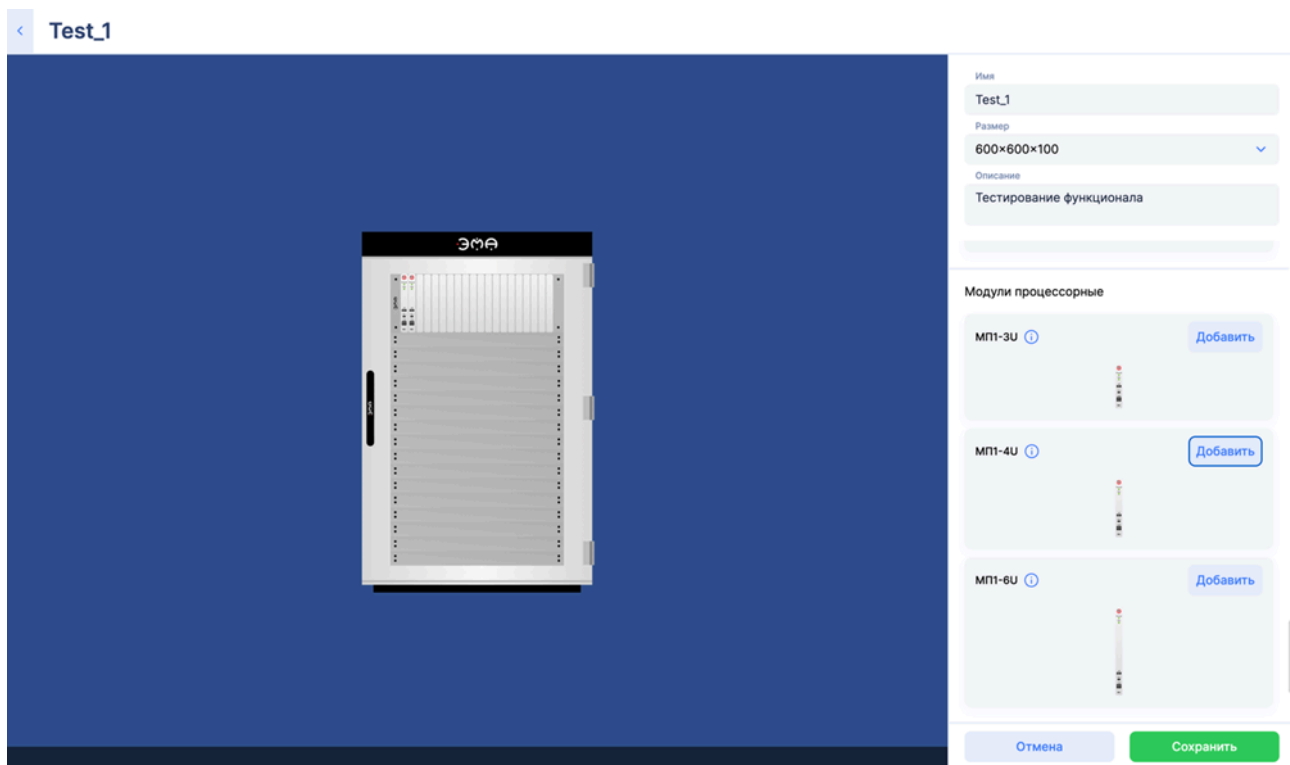


Рис. 31. Добавление контроллера в крейт через визуальный редактор

После чего новый контроллер отобразится в крейте (рис. 32). Для сохранения внесенных изменений необходимо нажать кнопку «Сохранить».

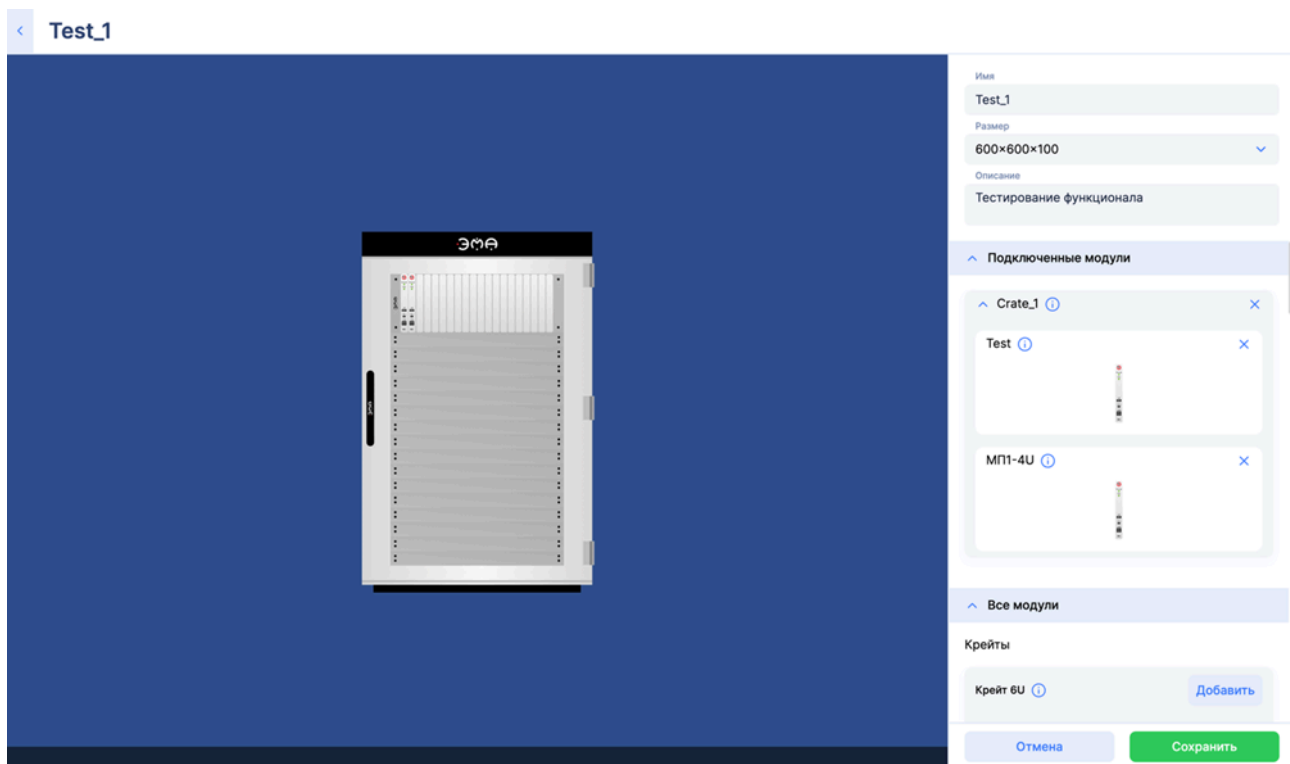


Рис. 32. Добавление контроллера в крейт через визуальный редактор

Перемещение (изменение позиции) контроллера также может осуществляться через визуальный редактор, используя технологию «drag-and-drop» (рис. 33).

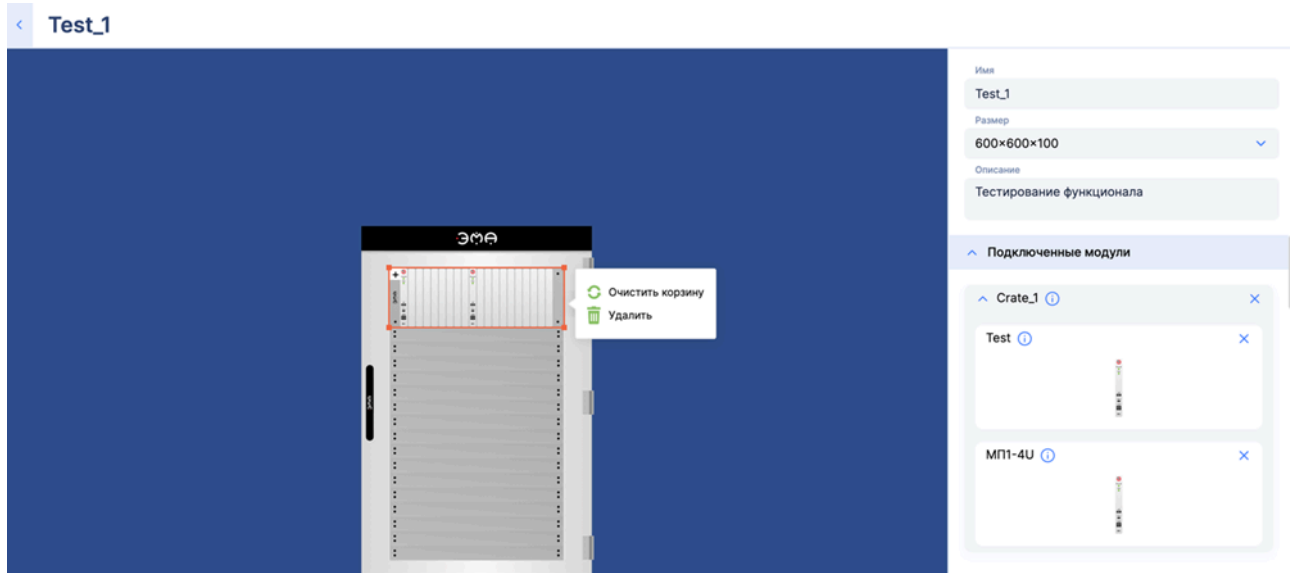


Рис. 33. Перемещение (изменение) позиции контроллера в визуальном редакторе

Аналогично работает и механизм удаления контроллера. Для этого следует нажать кнопку «Удалить» и подтвердить удаление в модальном окне, при необходимости отметив чекбокс «Удалить все вложенные папки» (рис. 34).

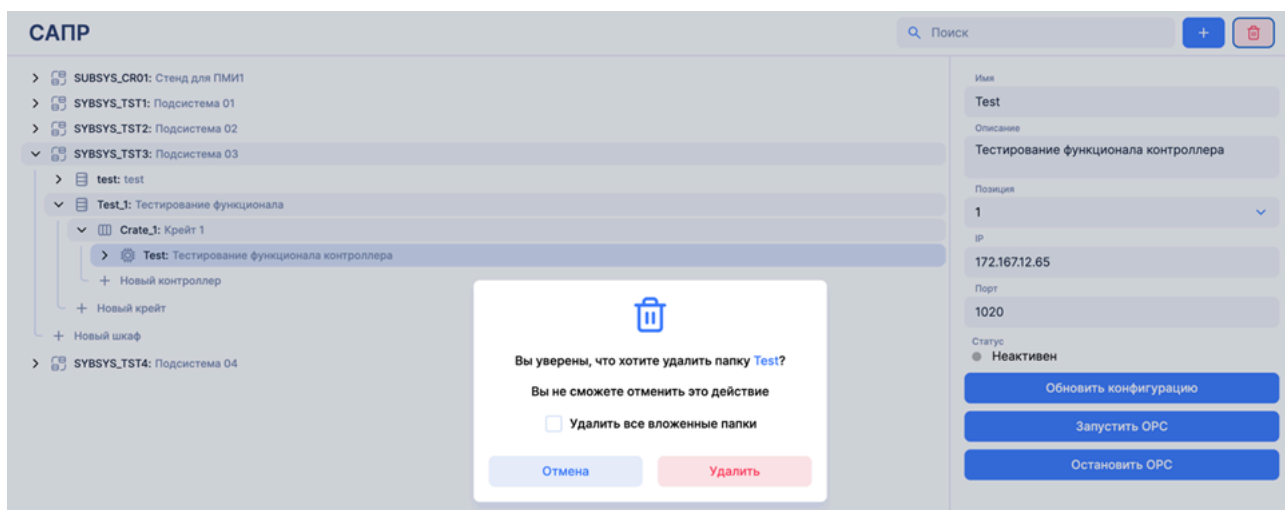


Рис. 34. Модальное окно удаления контроллера

Контроллер может быть удален и через аналогичную функцию в визуальном редакторе. Для этого в списке «Подключенные модули» необходимо нажать кнопку «Удалить» возле графического примитива контроллера, который следует удалить (рис. 35). После чего для сохранения изменений нажать на кнопку «Сохранить».

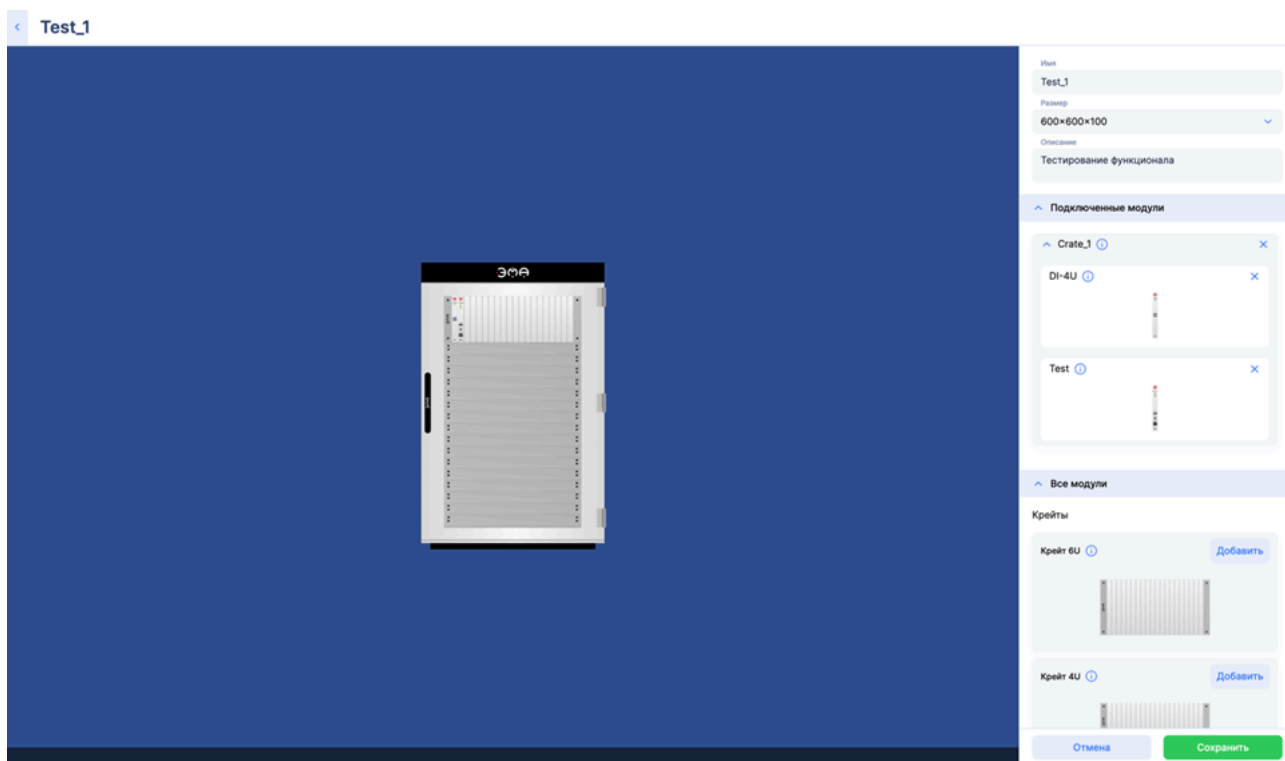


Рис. 35. Удаление контроллера через визуальный редактор

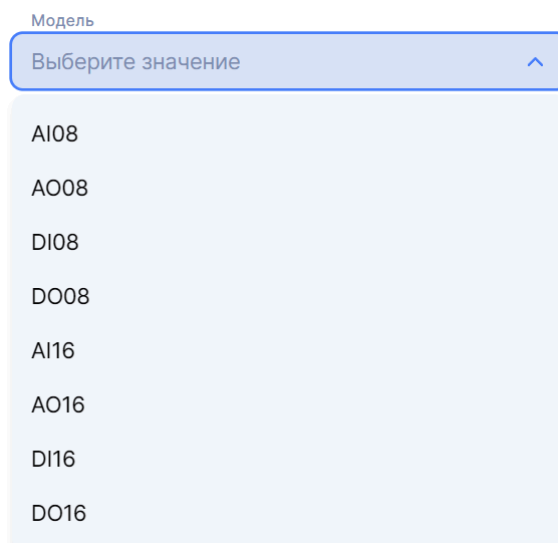
2.3.6 Конфигурация контроллера РСУ

Добавление нового модуля осуществляется с помощью кнопки «+ Новый модуль». К атрибутам модуля относятся: «Имя», «Описание», Вид модуля (Analog Input/Output; Digital Input/Output), «Позиция», «Источник». После заполнения всех полей и нажатия кнопки «Сохранить» модуль отобразится под контроллером (рис. 36).

The screenshot displays the SAПP configuration interface. On the left, a tree view shows the hierarchy: SUBSYS_CR01: Стенд для ПМИ1, SYBSYS_TST1: Подсистема 01, SYBSYS_TST2: Подсистема 02, and SYBSYS_TST3: Подсистема 03. Under SYBSYS_TST3, there are two racks (Шкаф 00 and Шкаф 01). Under Шкаф 01, there is a controller (Test_1: Тестирование функционала контроллера) and a crate (Crate_1: Крейт 1). Under the controller, there is a test (Test: Тестирование функционала модулей) and a test script (Test_script: Тестовый сценарий). The test script is expanded, showing a list of actions: Test_script, + Новый модуль, + Новый скрипт, + Новый контроллер, and + Новый крейт. The + Новый модуль button is highlighted. On the right, a form for adding a new module is visible. The fields are: Имя (Test_Input), Описание (Тестирование функционала модулей), Модель (AI08), Позиция (0), and Источник (Отсутствует). At the bottom, there are two buttons: Отмена and Сохранить.

Рис. 36. Добавление нового модуля

При добавление нового модуля в параметрах доступен следующий список моделей (Рис. 37).



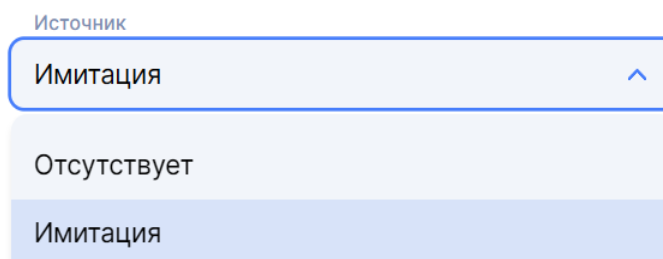
Модель

Выберите значение ^

- AI08
- AO08
- DI08
- DO08
- AI16
- AO16
- DI16
- DO16

Рис. 37. Список доступных моделей

В параметре «Источник» пользователю доступен имитатор (рис. 38).



Источник

Имитация ^

- Отсутствует
- Имитация

Рис. 38. Параметр «Источник»

После выбора имитатора в «Источнике» появляется параметр «Тип имитации» в котором необходимо выбрать значение (рис.39).

Имитация

Тип имитации

Выберите значение

Синусоидальный сигнал

Постоянное значение

Треугольный сигнал

Пилообразный сигнал

Прямоугольный сигнал

Случайное значение

Рис. 39. Параметр «Тип имитации»

Затем пользователь должен задать нужные значения в поля конфигурации имитатора (Амплитуда, Период функции, Смещение по вертикали, горизонтали, рис. 40).

Имитация

Тип имитации

Пилообразный сигнал

Амплитуда

0

Период функции (сек)

0

Смещение по вертикали

0

Смещение по горизонтали

0

Рис. 40. Конфигурация имитатора

Для сохранения внесенных изменений необходимо нажать кнопку "Обновить конфигурацию" в контроллере (рис. 41).

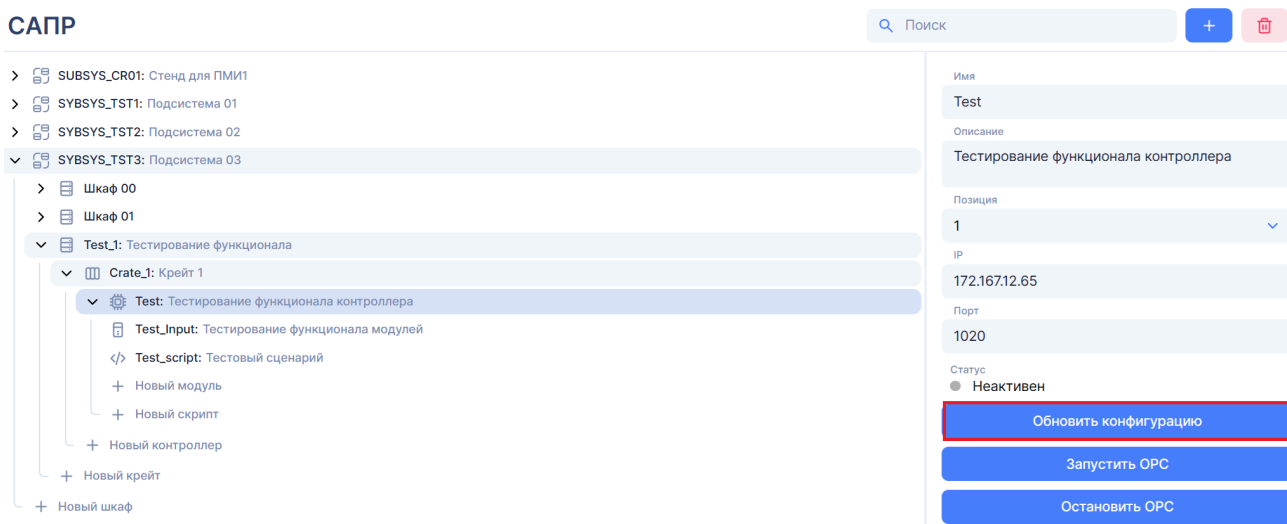


Рис. 41. Сохранение внесённых изменений

Аналогично (разд. [2.3.2-2.3.4](#)) реализована работа с модулями в визуальном редакторе (добавление, удаление, перемещение) (рис. 42).

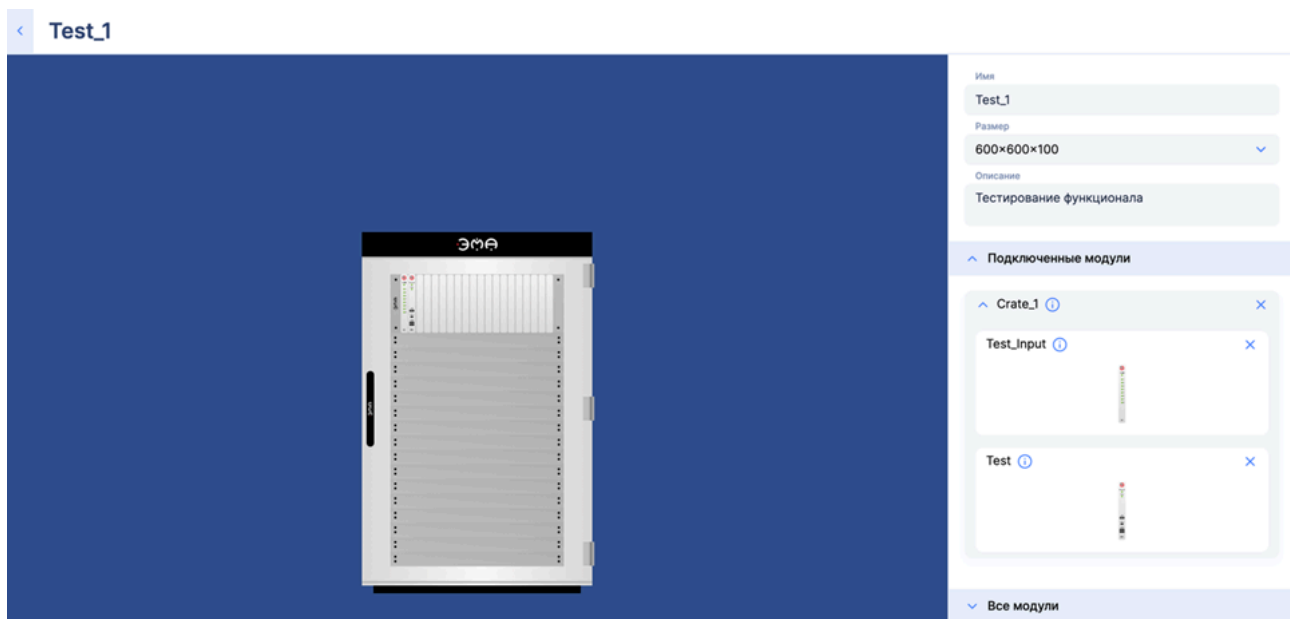


Рис. 42. Новый модуль в визуальном редакторе

Сценарий работы модуля задается с помощью кода на языке Lua. Для добавления сценария необходимо нажать кнопку «+ Новый скрипт». Для активации сценария по расписанию в поле «Тип» нужно выбрать соответствующее значение «По расписанию» (рис 43). В случае, если сценарий должен работать реактивно на изменение значений переменных, следует выбрать тип «По изменению значений» (рис. 38). Изменения необходимо сохранить кнопкой «Сохранить».

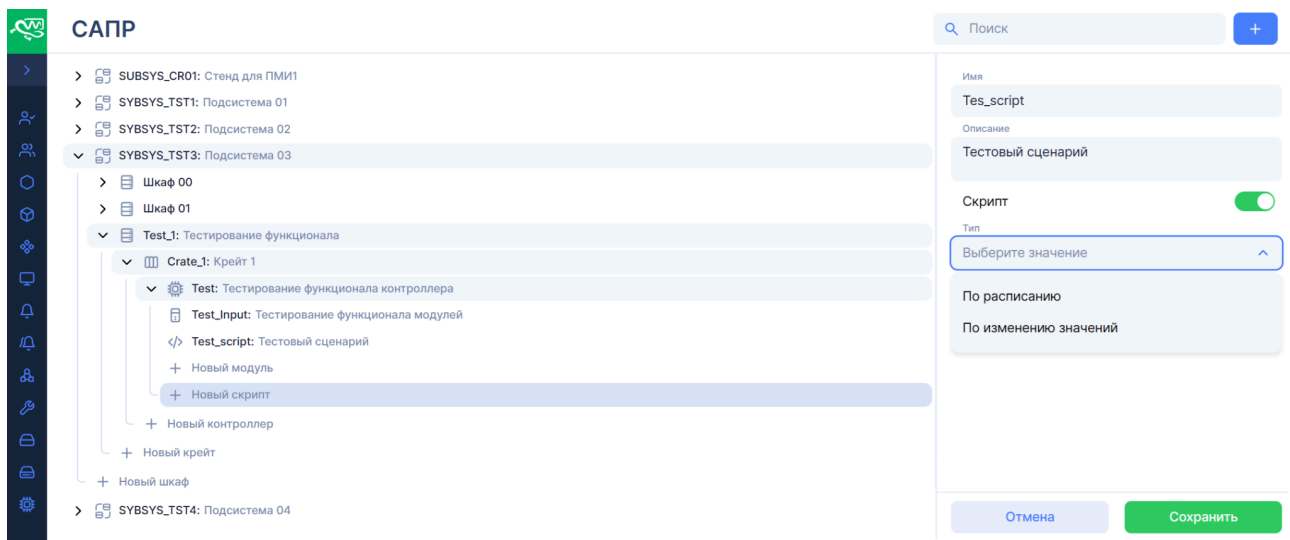


Рис. 43. Добавление сценария в конфигурацию контроллера РСУ

В типе «По расписанию» можно задавать период запуска сценария. Для этого необходимо в поле «Срок» нажать на кнопку «Редактировать» (рис. 44).

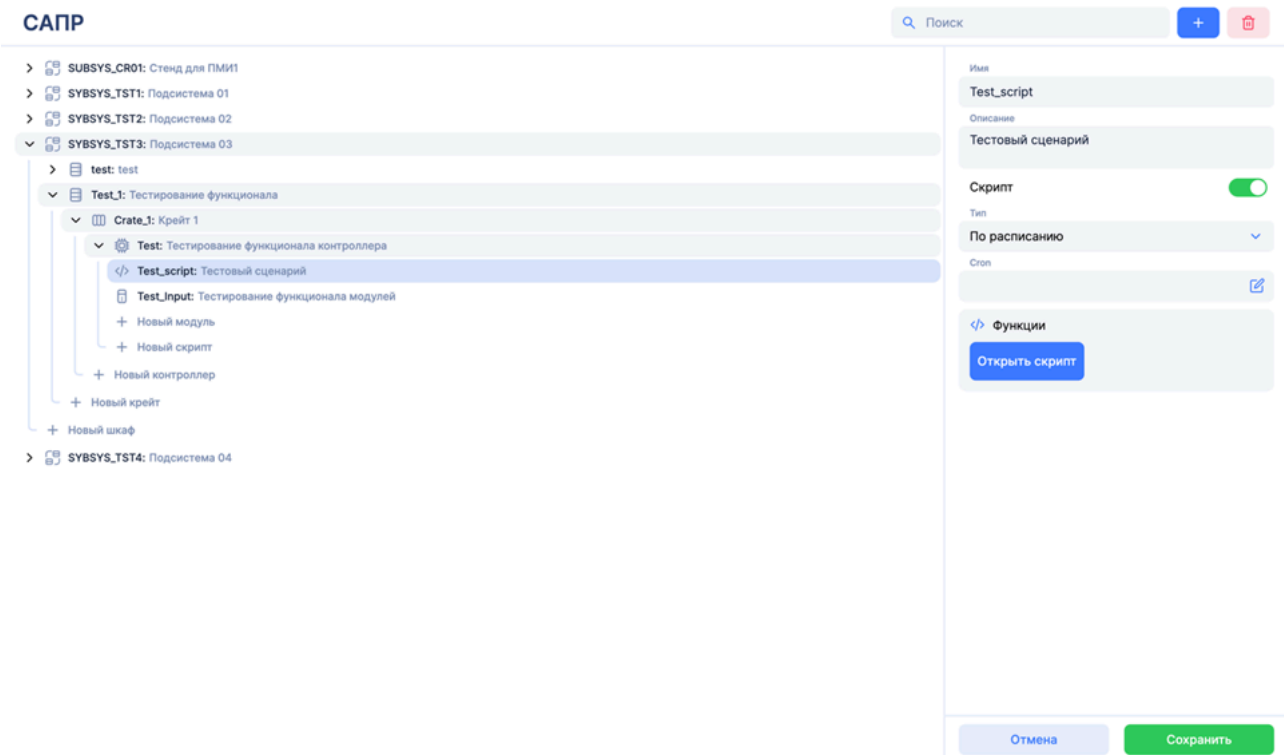


Рис. 44. Окно просмотра/редактирования сценария контроллера РСУ с типом «По расписанию»

В открывшемся модальном окне следует выбрать необходимое значение и нажать кнопку «Применить» (рис 45).

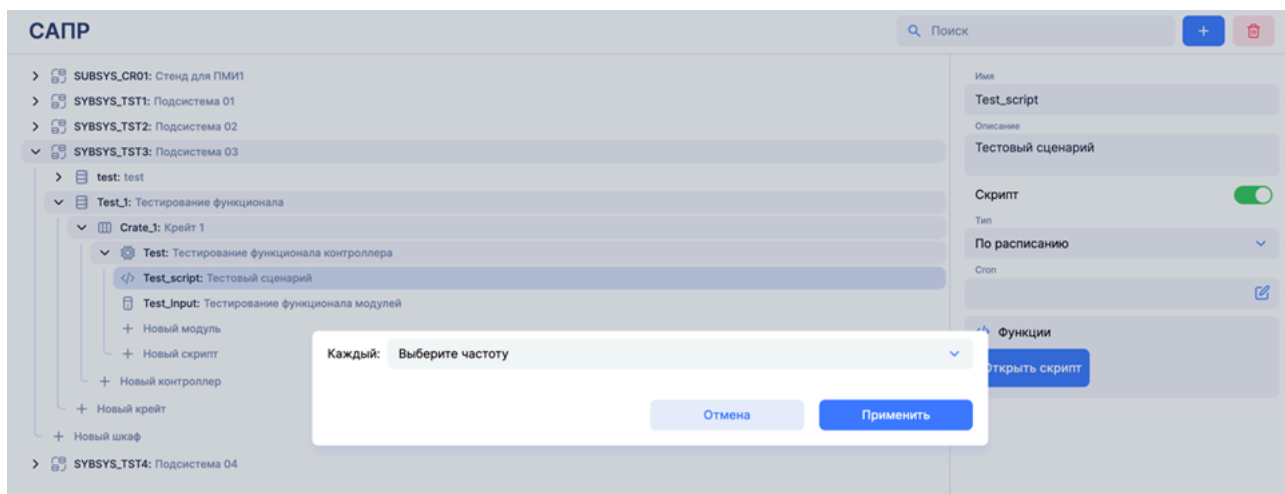


Рис. 45. Модальное окно редактирования периода запуска сценария контроллера PCY

Для открытия редактора сценариев необходимо нажать кнопку «Открыть скрипт» (рис. 44) в окне редактирования сценария, после чего пользователь окажется в редакторе (рис. 46).

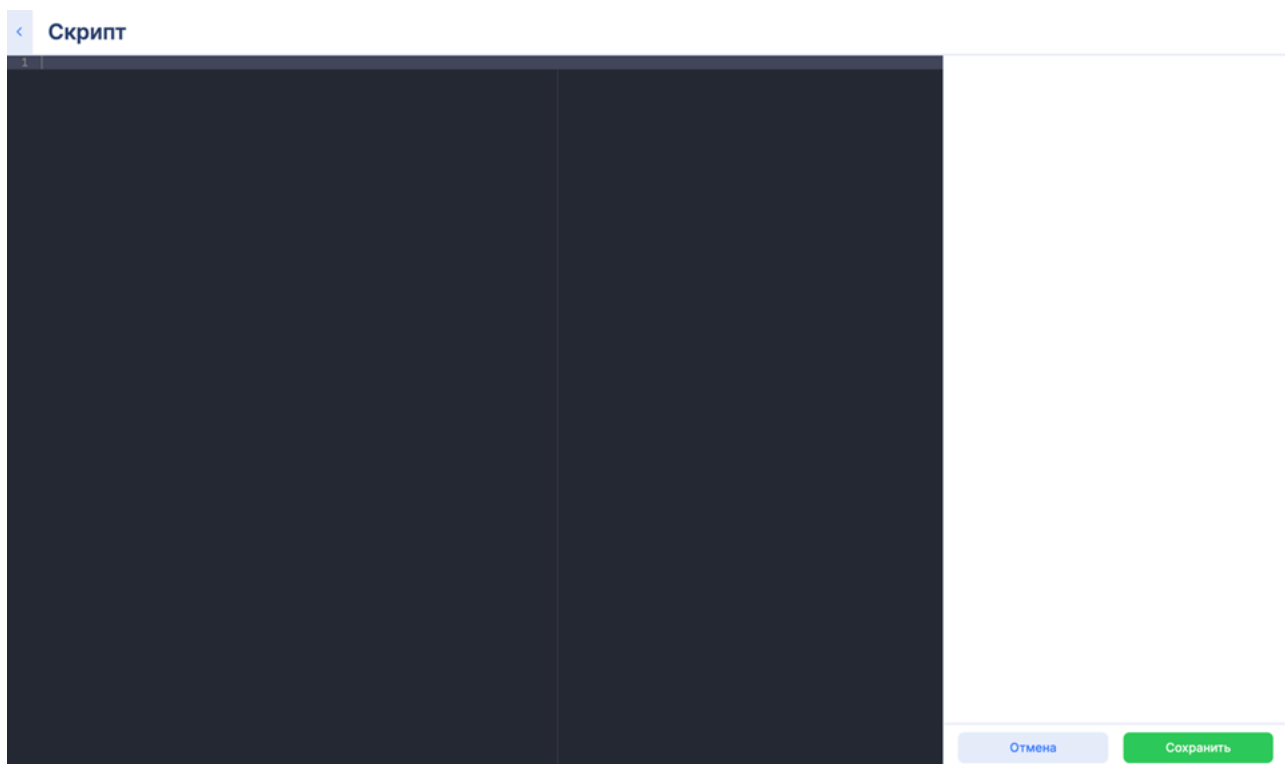


Рис. 46. Редактор сценариев контроллера PCY

Для привязки входных переменных необходимо добавить функцию, используя оператор `function` и указать ее аргументы. Входные переменные будут отображаться под спойлером «Входные переменные» (рис. 47).

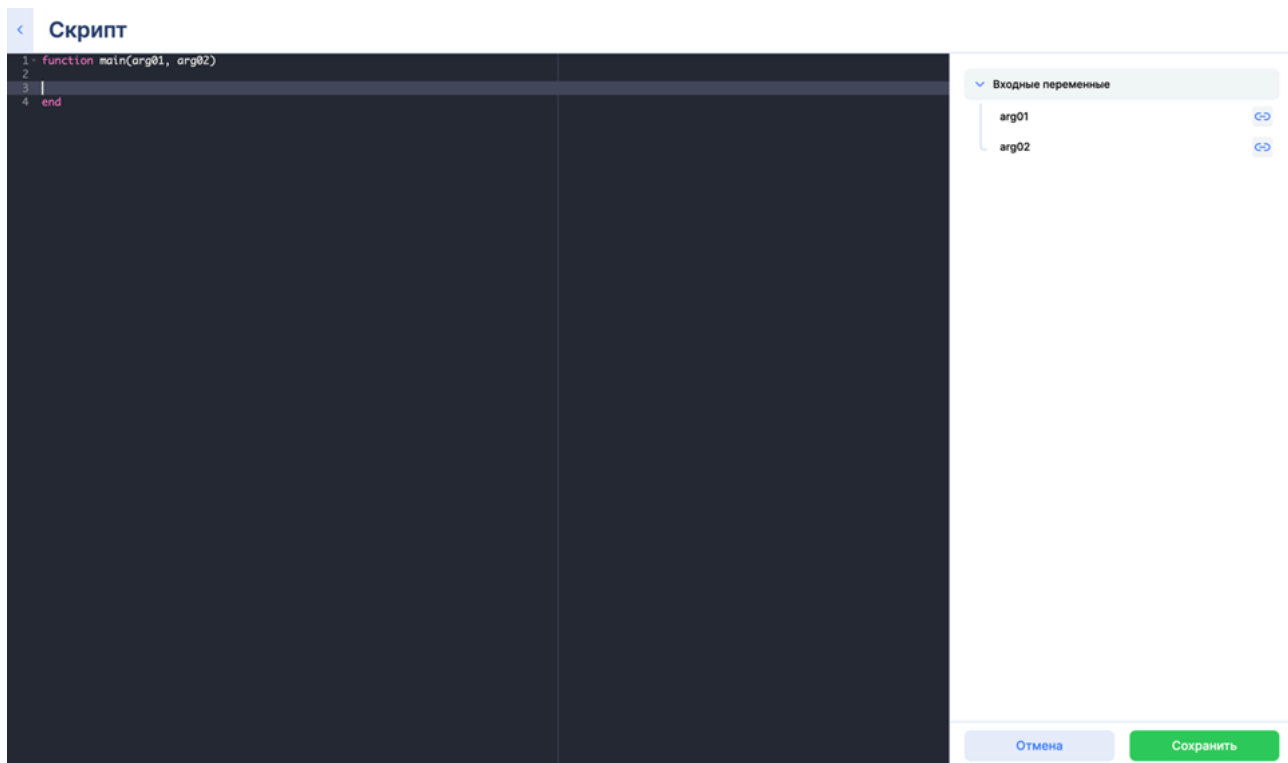


Рис. 47. Добавление функции в редактор сценариев контроллера РСУ

Чтобы привязать переменную к тегу I/O необходимо нажать на иконку привязки напротив переменной и выбрать необходимое значение в модальном окне, завершив привязку кнопкой «Привязать» (рис. 48).

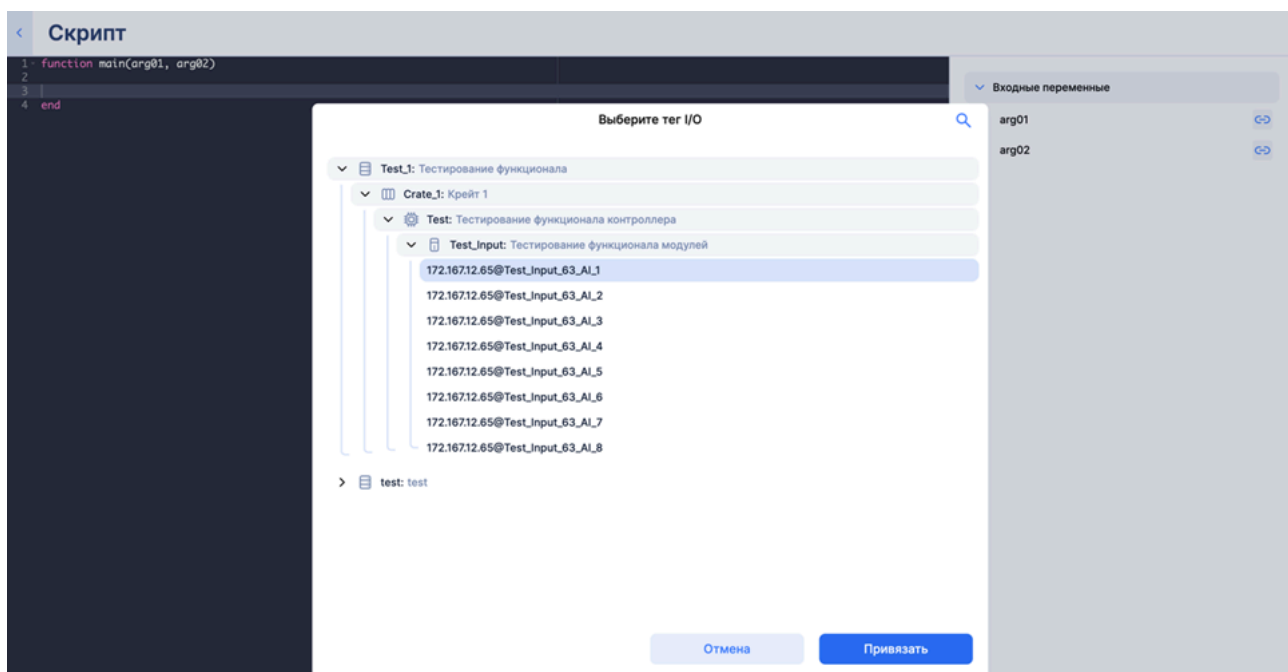
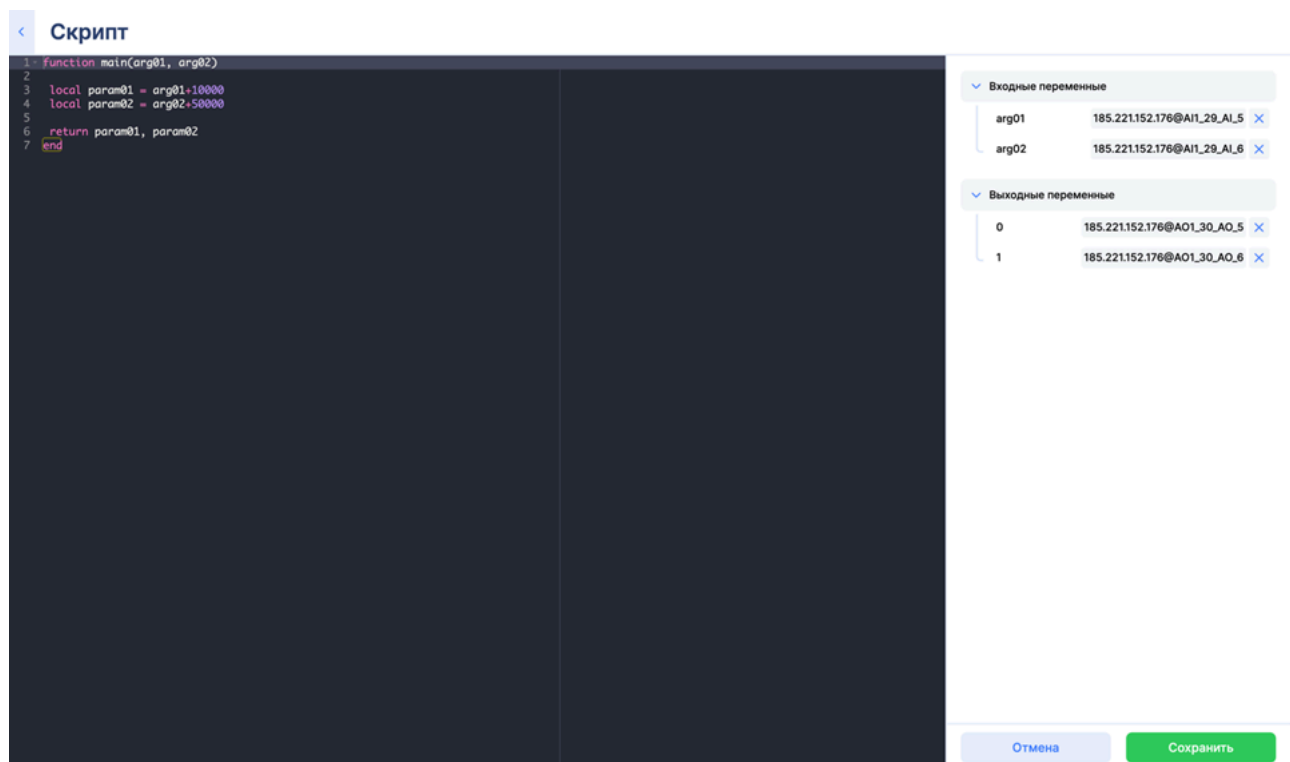


Рис. 48. Привязка входных переменных к тегу

Выходные переменные указываются после оператора return и отображаются под спойлером «Выходные переменные» (рис.49). Привязка выходных переменных к тегу осуществляется аналогично.



The screenshot shows a script editor with a dark background. The code is as follows:

```
1 function main(arg01, arg02)
2
3   local param01 = arg01-10000
4   local param02 = arg02+50000
5
6   return param01, param02
7 end
```

On the right side, there are two sections for variable bindings:

- Входные переменные (Input variables):**
 - arg01: 185.221.152.176@AI1_29_AI5
 - arg02: 185.221.152.176@AI1_29_AI6
- Выходные переменные (Output variables):**
 - 0: 185.221.152.176@AO1_30_AO5
 - 1: 185.221.152.176@AO1_30_AO6

At the bottom right, there are two buttons: "Отмена" (Cancel) and "Сохранить" (Save).

Рис. 49. Пример кода сценария с привязанными к тегам входными и выходными переменными

2.3.7 Визуальный редактор конфигуратора

Визуальный редактор представляет собой интерактивный инструмент для построения и редактирования конфигурации контроллера. Он обеспечивает наглядное отображение всех элементов системы и их взаимосвязей (рис. 50).

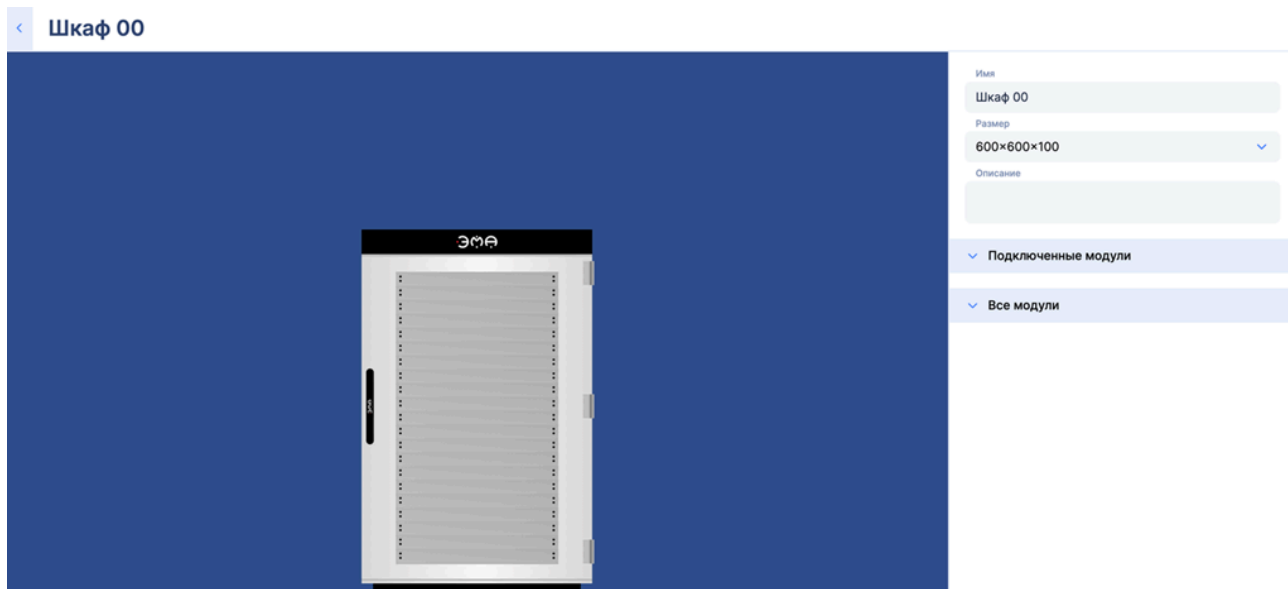


Рис. 50. Визуальный редактор конфигурации контроллера PCY

Визуальный редактор конфигурации является ключевым инструментом для проектирования системы. Визуальный редактор объединяет функционал CAD-системы и удобство графического интерфейса, что позволяет быстро и эффективно собирать любые конфигурации контроллера PCY. (рис. 51).

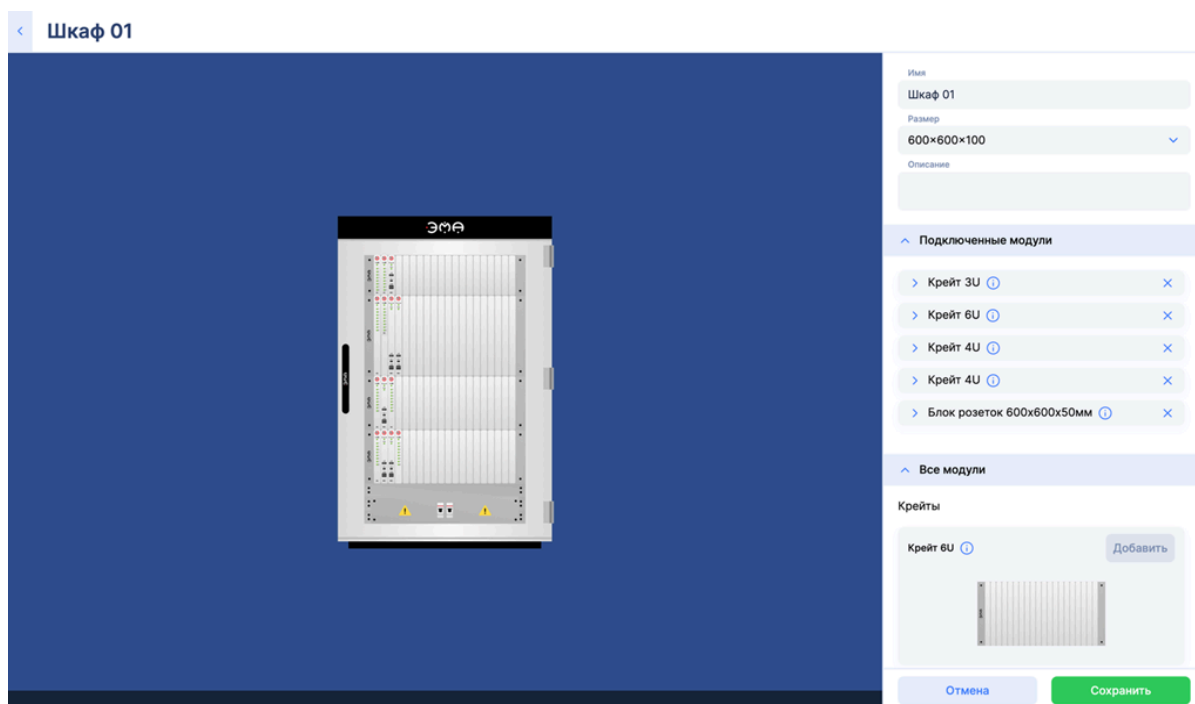


Рис. 51. Визуальный редактор конфигурации контроллера PCY

Редактор позволяет работать со следующими компонентами:

- Серверный шкаф – основной контейнер, в котором размещаются все устройства (рис. 52);



Рис. 52. Серверный шкаф

- Крейты – секции для установки модулей. Поддерживается несколько типоразмеров (3U, 4U, 6U) (рис. 53)

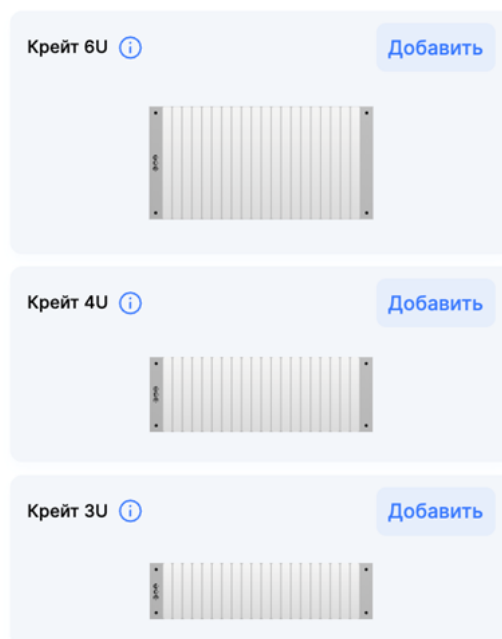


Рис. 53. Крейты

- Процессорные модули – центральные элементы системы, выполняющие функции управления и обработки данных (рис 54)

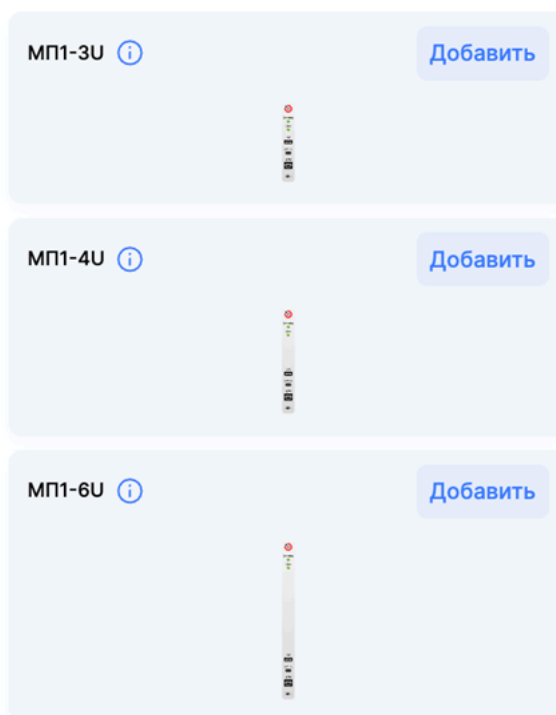


Рис. 54. Модули процессорные

- Модули ввода/вывода (I/O) – предназначены для подключения датчиков, исполнительных механизмов и других внешних устройств (рис. 55)

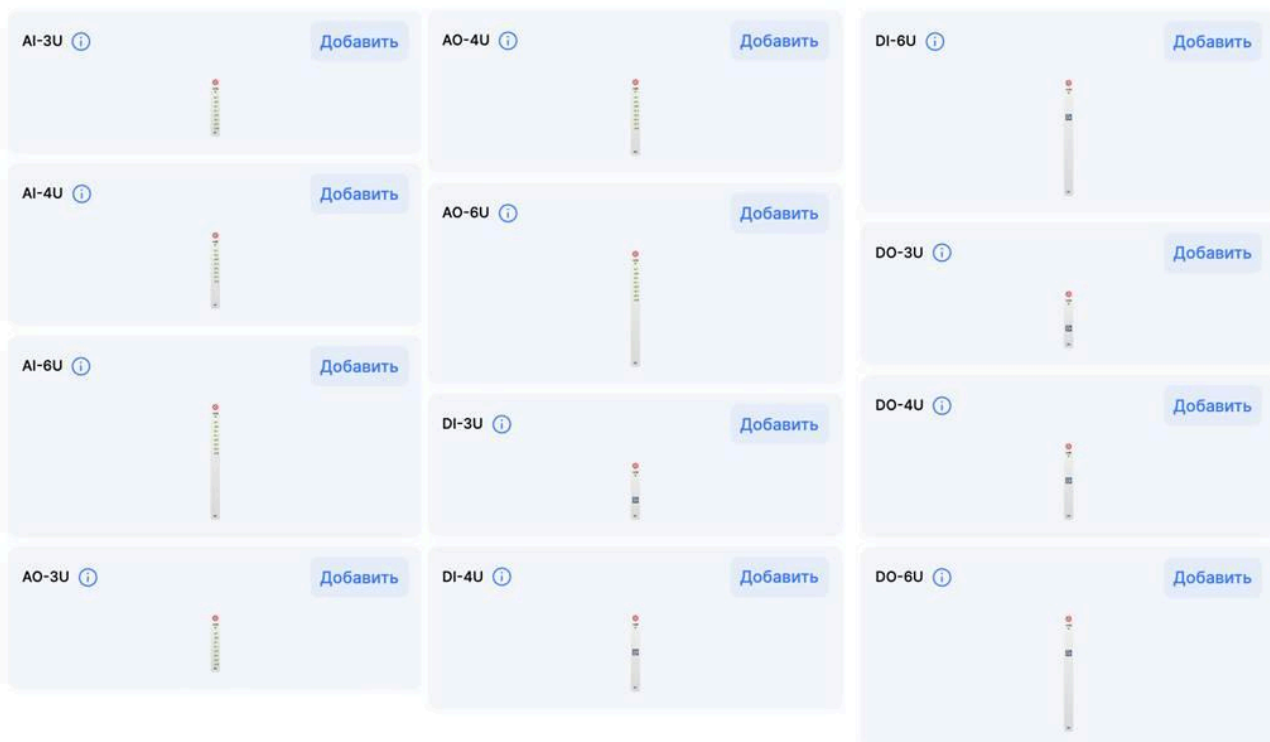


Рис. 55. Модули ввода/вывода

- Блоки питания – обеспечивают энергоснабжение всех установленных модулей (рис. 56)

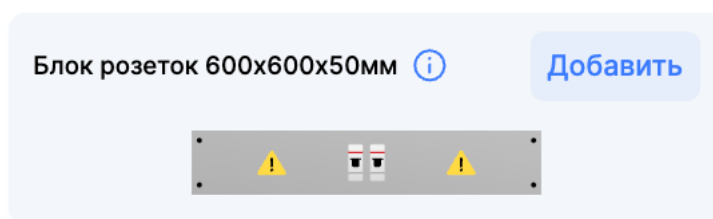


Рис. 56. Блок питания

Каждый элемент представлен в виде графического примитива и имеет собственные параметры (имя, описание, размер, позиция, для контроллеров – также IP-адрес и порт).

Редактор поддерживает все основные операции, необходимые для конфигурирования:

1. Добавление элементов:

- элементы выбираются из панели «Все модули»;
- добавление выполняется кнопкой «Добавить».

После этого объект появляется в шкафу и в списке подключенных модулей.

2. Перемещение элементов («drag-and-drop»):

- любой элемент можно переместить в другое место внутри шкафа;
- перетаскивание меняет его позицию как визуально, так и в параметрах (поле «Позиция»).

3. Редактирование параметров:

- в правой панели доступно редактирование свойств шкафа;
- изменения сохраняются кнопкой «Сохранить».

4. Удаление элементов:

- выполняется кнопкой «Удалить»;
- для предотвращения случайных действий используется подтверждение удаления кнопкой «Сохранить».

5. Иерархическое отображение структуры:

- все модули отображаются в виде списка с возможностью разворачивания и сворачивания разделов (крейты, подключённые модули и т.д.). Это облегчает навигацию при работе со сложными конфигурациями.

Более подробная работа с редактором описывается в разд. [2.3.3](#), [2.3.4](#), [2.3.5](#), [2.3.6](#).