



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИИТ)
Кафедра математического обеспечения и стандартизации
информационных технологий**

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ № 13-14
по дисциплине «Технологические основы интернета вещей»

Выполнили студенты ИКБО-61-23
группы: Тагин Н.В.
 Филиппова Т.А.
 Астапович К.А

(подпись)

Принял ст. Образцов В.М.
преподаватель:

(подпись)

Москва 2025 г.

СОДЕРЖАНИЕ

РЕАЛИЗАЦИЯ ЗАДАНИЙ	3
1.1 Практическая работа №13	3
1.2 Практическая работа №14	9
ВЫВОДЫ.....	16
ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....	17

РЕАЛИЗАЦИЯ ЗАДАНИЙ

1.1 Практическая работа №13

Модель визуализации данных в облачной платформе позволяет наглядно отображать параметры устройств и управлять ими через интерактивный интерфейс. Данные можно визуализировать на разных уровнях: на локальном сервере, подключённом к сети микроконтроллеров, либо в облаке, где доступна полная информация о телеметрии и состоянии устройств. Телеметрические данные включают измерения температуры, влажности, давления и другие показатели окружающей среды, а служебная информация — состояние устройств, температуру внутренних модулей и прочие параметры работы системы. Облачная платформа обеспечивает централизованный контроль и позволяет отслеживать работу всей системы в режиме реального времени.

Для визуализации данных используется механизм **дашбордов**. Дашиборд представляет собой область, на которую можно размещать специальные элементы — **виджеты**. Виджеты делятся на две категории: для отображения информации и для управления устройствами. Виджеты отображения предназначены для представления динамических данных, например, с помощью графиков, линейных шкал, гистограмм или процентов. Виджеты управления позволяют пользователю взаимодействовать с устройствами через кнопки, ползунки, переключатели и поля ввода.

Создание дашиборда начинается во вкладке «Дашборды» раздела «Визуализация». Для нового дашиборда требуется указать имя, описание и тип — по объекту или по модели. После создания дашиборда появляется возможность добавлять виджеты. Каждый виджет настраивается через окно конфигурации, где указывается его имя, описание и внешний вид. Для отображения данных необходимо задать источник данных, выбрав объект и соответствующие атрибуты, определённые в модели устройства. При

необходимости можно подключить несколько объектов и задать для каждого отображаемые параметры, цвет и подписи.

После настройки виджет добавляется на дашборд. Добавленные элементы можно редактировать, перемещать и удалять. Для корректной работы интерфейса и сохранения изменений рекомендуется регулярно сохранять дашборд. Такой подход обеспечивает удобный и наглядный контроль состояния устройств и динамики измеряемых параметров, позволяя пользователю быстро получать актуальную информацию и управлять системой.

Перейдем к выполнению практической работы. Для работы были использованы ранее созданные объекты и модели из прошлых практических работ. Покажем итоговый дашборд на рисунке 1. На последующих рисунках покажем их настройку.

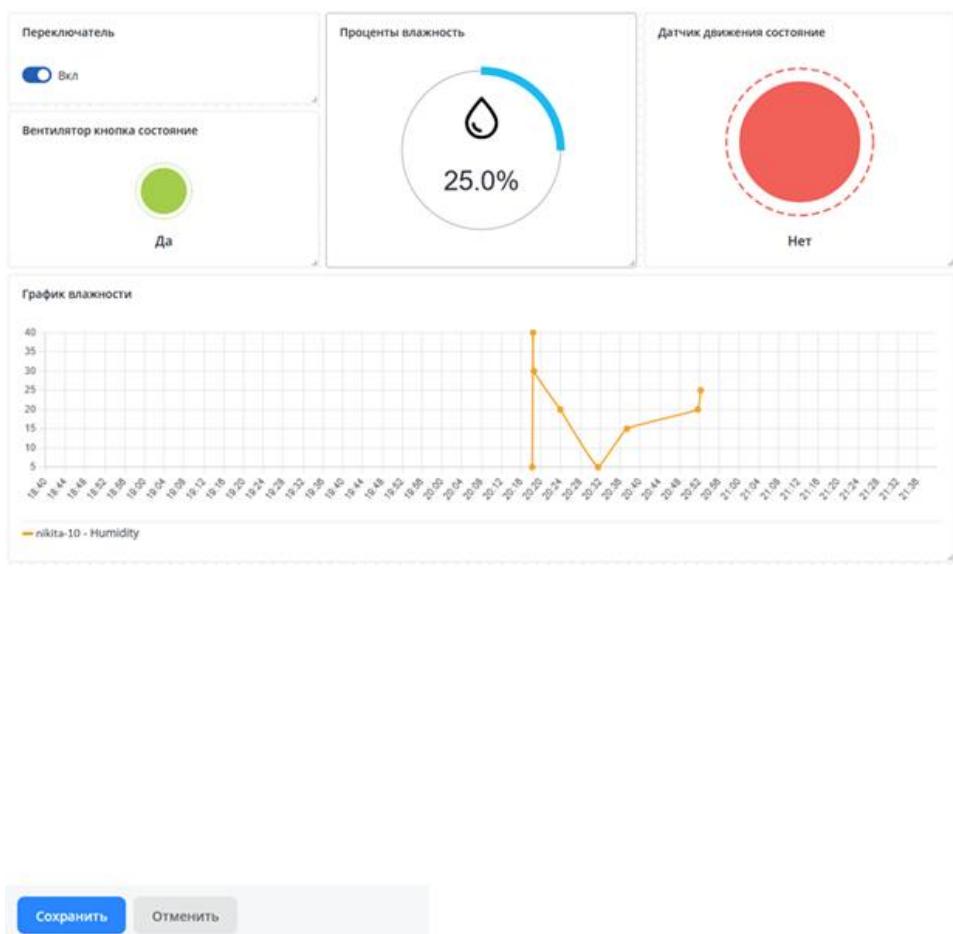


Рисунок 1 – итоговый дашборд

Далее покажем настройки каждого дашборда

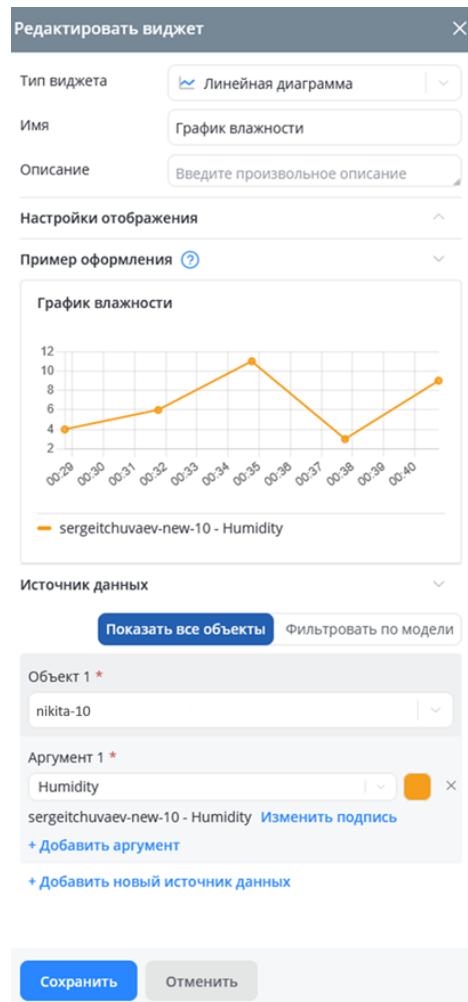


Рисунок 2 – параметры графика

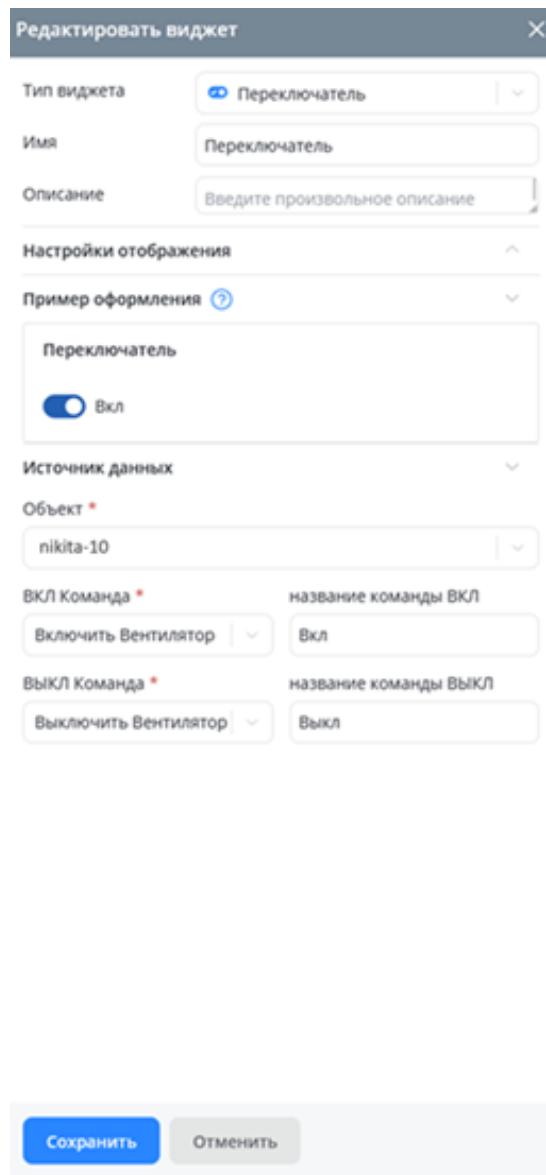


Рисунок 3 – параметры переключателя

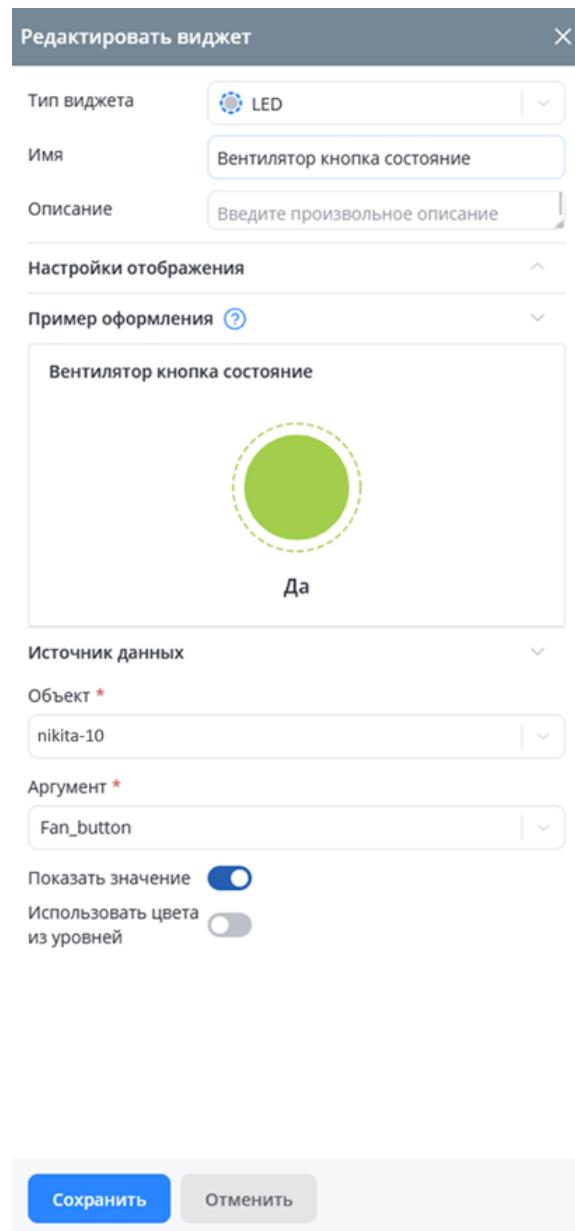


Рисунок 4 – параметры кнопки состояния вентилятора

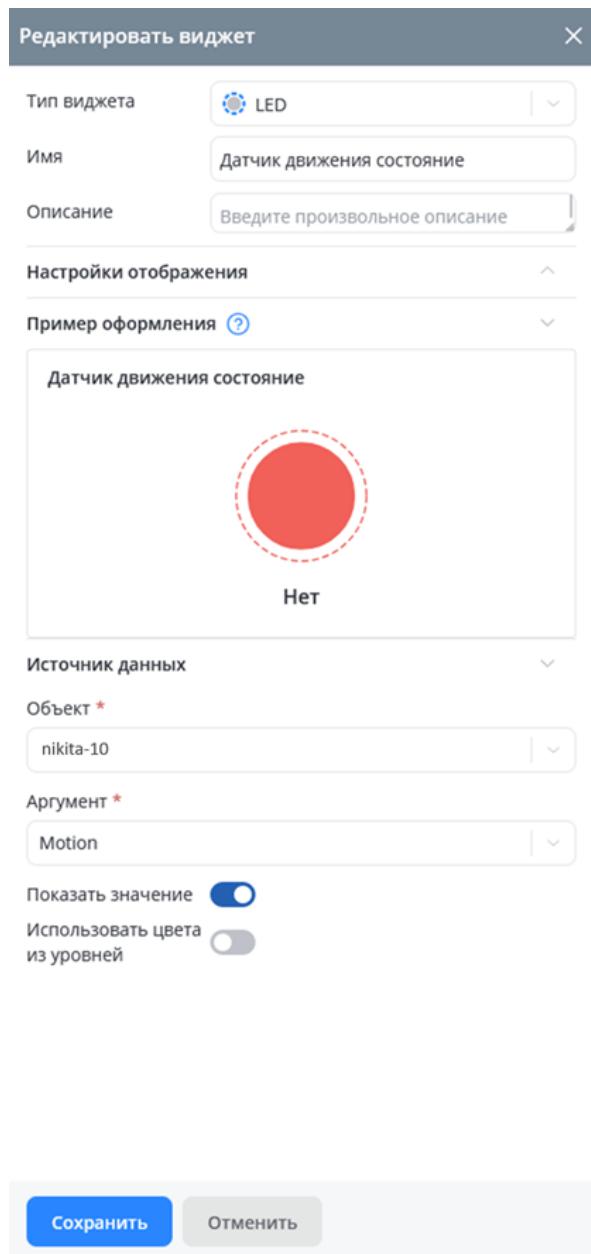


Рисунок 5 – параметры кнопки состояния датчика движения

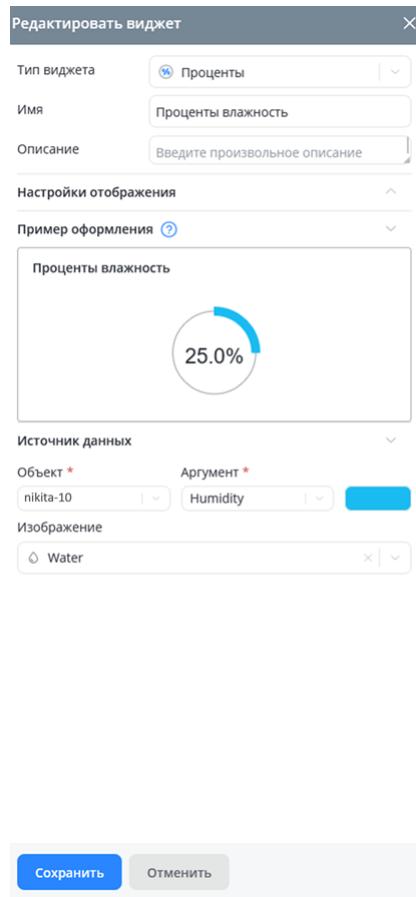


Рисунок 6 – параметры вывода процента влажности

1.2 Практическая работа №14

Практическая работа №14 посвящена использованию управляющих элементов интерфейса облачной платформы для удалённого взаимодействия с устройствами. В архитектуре IoT-решений панели управления позволяют не только наблюдать за состоянием системы, но и выполнять непосредственное управление, отправляя команды на устройство через облако. В платформе Rightech эта функциональность реализована через встроенные управляющие виджеты, размещаемые на дашбордах. Такие элементы обеспечивают интуитивный доступ к выполнению команд модели устройства и позволяют пользователю быстро и наглядно инициировать нужные действия.

Управляющие виджеты представлены несколькими типами, отличающимися способом ввода и характером передаваемых данных. Виджет «Переключатель» используется для активации бинарных команд модели —

включения или выключения устройства. Кнопка позволяет выполнить любое действие, определённое в модели, например отправку сообщения или запуск таймера. Ползунок и стекпер предназначены для управления параметрами, имеющими диапазон значений: они позволяют задавать величины с плавной или дискретной регулировкой. Для устройств, использующих цветовое управление, предусмотрен отдельный виджет выбора RGB-цвета, который формирует соответствующее значение и передаёт его в команду.

Создание управляющего виджета требует предварительного определения соответствующей команды в модели устройства. Команда описывает, какое действие необходимо выполнить при взаимодействии с виджетом: публикацию сообщения в MQTT-топик, изменение значения параметра, выполнение сценария и т.п. Если виджет должен передавать динамические данные (например, выбранный цвет или текущее положение ползунка), в теле сообщения команды используется специальная конструкция подстановки, позволяющая передать текущее состояние виджета. Это обеспечивает гибкость управления и избавляет от необходимости вручную задавать значения при каждом вызове команды.

После создания команды управляющий виджет добавляется на дашборд и связывается с соответствующим объектом и действием. В зависимости от типа виджета пользователю могут быть доступны различные варианты конфигурации — от простого вызова команды до передачи сложной структуры данных. После размещения на дашборде виджеты можно свободно перемещать, настраивать или удалять.

Проверка работоспособности управляющих элементов обычно выполняется через MQTT-подписку, что позволяет убедиться в корректности публикации сообщений и их соответствии заданной команде. Такой подход обеспечивает полный цикл взаимодействия: от формирования команды в модели до фактического выполнения действия устройством.

```
C:\Program Files\mosquitto>mosquitto_sub -d -h dev.rightech.io -i mqtt-nikita-10 -48yn12 -t base/state/fan_button
Client mqtt-nikita-10 -48yn12 sending CONNECT
Client mqtt-nikita-10 -48yn12 received CONNACK (0)
Client mqtt-nikita-10 -48yn12 sending SUBSCRIBE (Mid: 1, Topic: base/state/fan_button, QoS: 0, Options: 0x00)
Client mqtt-nikita-10 -48yn12 received SUBACK
Subscribed (mid: 1): 0
Client mqtt-nikita-10 -48yn12 received PUBLISH (d0, q1, r0, m1, 'base/state/fan_button', ... (1 bytes))
Client mqtt-nikita-10 -48yn12 sending PUBACK (m1, rc0)
0
Client mqtt-nikita-10 -48yn12 received PUBLISH (d0, q1, r0, m2, 'base/state/fan_button', ... (1 bytes))
Client mqtt-nikita-10 -48yn12 sending PUBACK (m2, rc0)
1
|
```

Рисунок 7 – подписка на топик и изменение состояния кнопки вентилятора

```
C:\Program Files\mosquitto>mosquitto_sub -d -h dev.rightech.io -i mqtt-nikita-10 -48yn12 -t base/state/color
Client mqtt-nikita-10 -48yn12 sending CONNECT
Client mqtt-nikita-10 -48yn12 received CONNACK (0)
Client mqtt-nikita-10 -48yn12 sending SUBSCRIBE (Mid: 1, Topic: base/state/color, QoS: 0, Options: 0x00)
Client mqtt-nikita-10 -48yn12 received SUBACK
Subscribed (mid: 1): 0
Client mqtt-nikita-10 -48yn12 received PUBLISH (d0, q1, r0, m1, '/state/color', ... (19 bytes))
Client mqtt-nikita-10 -48yn12 sending PUBACK (m1, rc0)
{"color": "#EC5750"}
Client mqtt-nikita-10 -48yn12 received PUBLISH (d0, q1, r0, m2, '/state/color', ... (19 bytes))
Client mqtt-nikita-10 -48yn12 sending PUBACK (m2, rc0)
{"color": "#5D7CE5"}
```

Рисунок 8 – подписка на топик и изменение состояния цвета RGB ленты

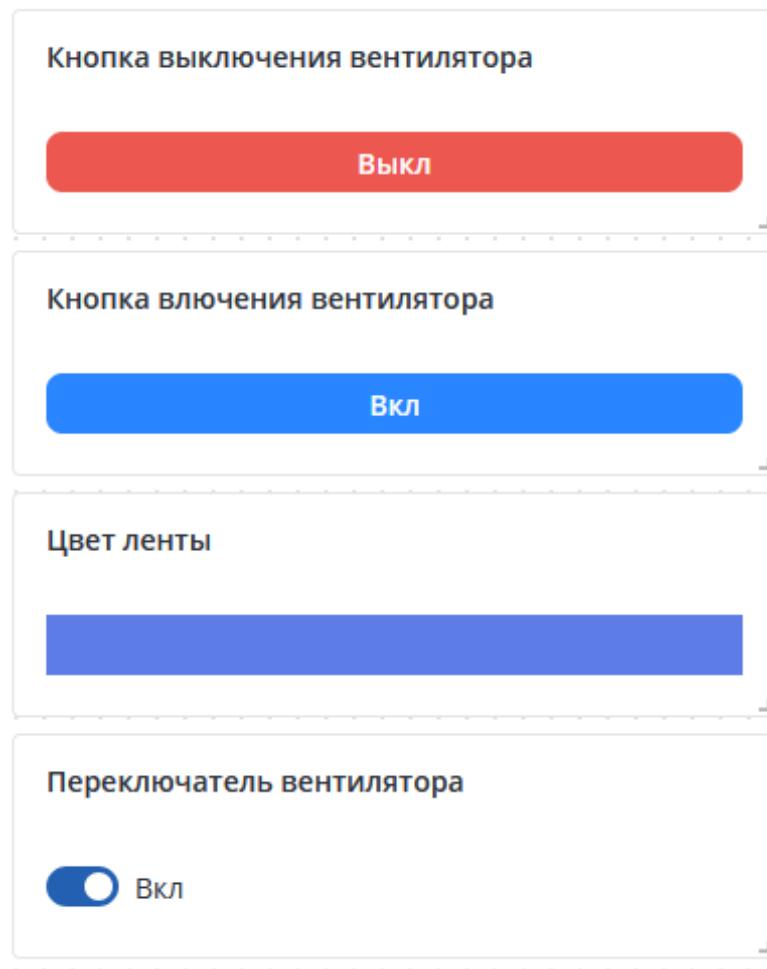


Рисунок 9 – созданные кнопки и переключатели

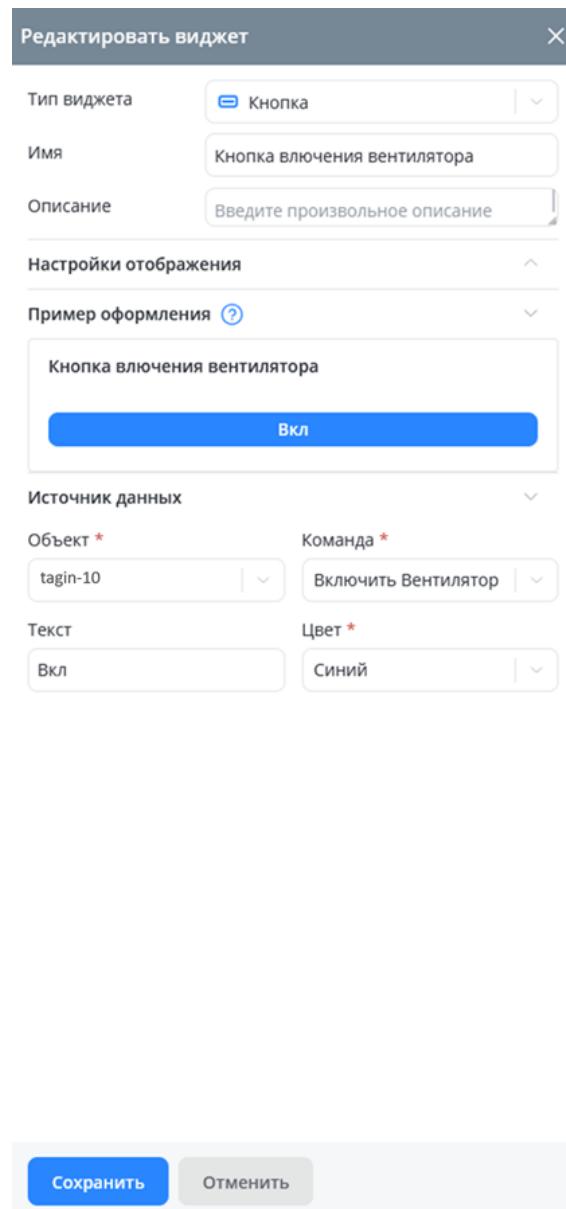


Рисунок 10 – конфигурация кнопок вентилятора

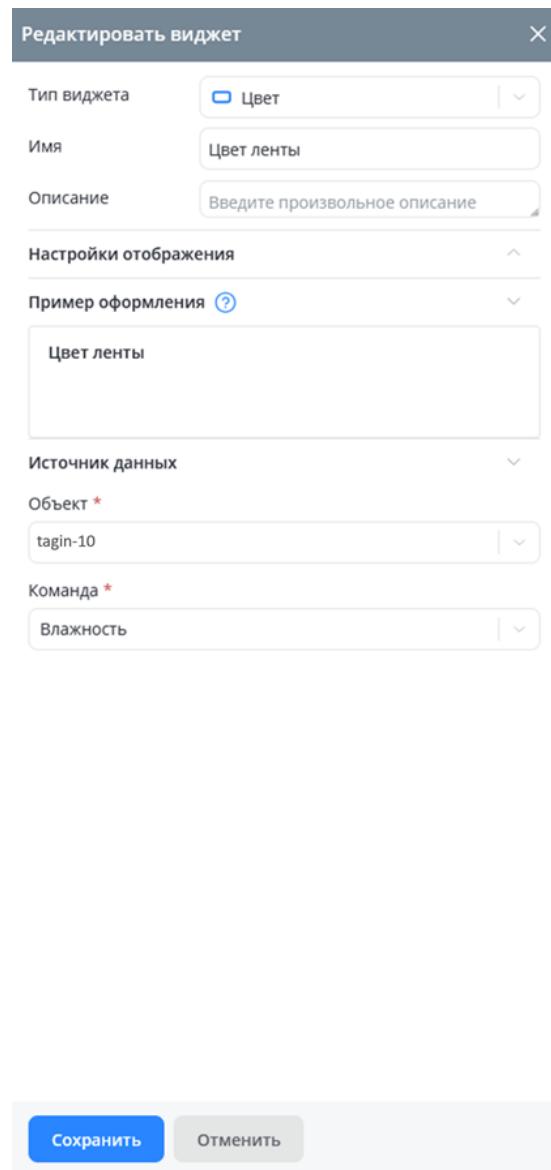


Рисунок 11 – конфигурация цвета ленты

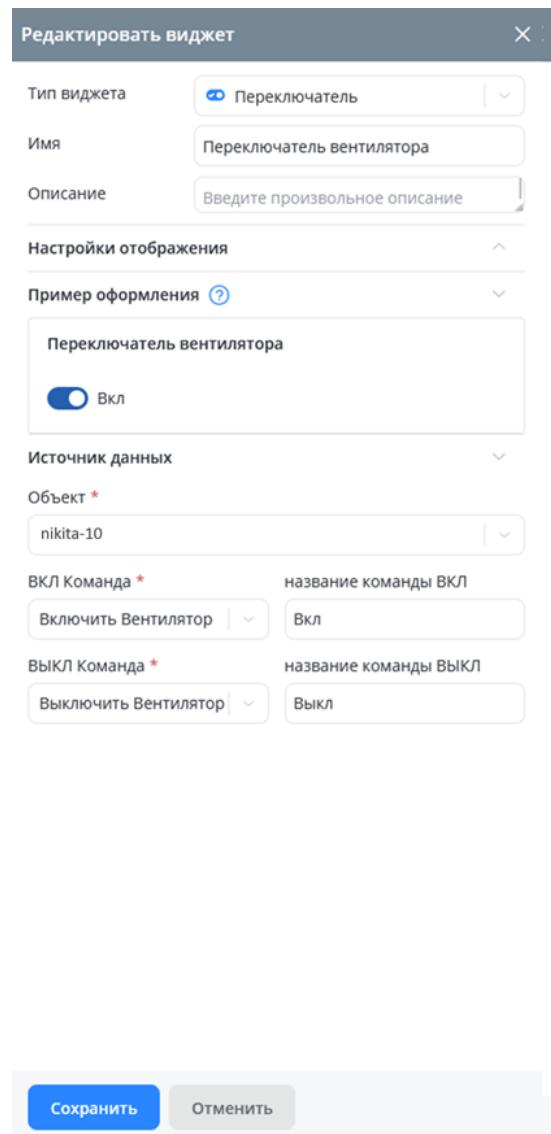


Рисунок 12 – конфигурация переключателя вентилятора

ВЫВОДЫ

В ходе выполнения практических работ были изучены ключевые механизмы визуализации и управления устройствами в облачной платформе Rightech.

В рамках работы №13 была рассмотрена система построения дашбордов и принципы отображения телеметрии IoT-устройств. Были исследованы способы представления данных с помощью различных типов виджетов, а также методика настройки источников данных, включающая привязку параметров объекта и конфигурацию визуальных элементов. Это позволило на практике освоить инструменты визуального мониторинга состояния устройства и оценить возможности облачной платформы по централизованному отображению телеметрии.

В работе №14 была изучена функциональность управляющих виджетов, предназначенных для инициирования команд модели устройства. Были рассмотрены различные типы элементов управления и особенности формирования динамических команд, включая передачу текущего состояния виджета через механизм подстановок. Практически реализовано создание команд модели, их привязка к виджетам и проверка корректности взаимодействия через MQTT-брюкер. Это позволило освоить принципы удалённого управления устройствами на основе облачной инфраструктуры.

Таким образом, обе работы продемонстрировали полный цикл взаимодействия IoT-устройства с облачной платформой — от получения и визуализации телеметрических данных до формирования и отправки управляющих команд. Полученные навыки являются основой для дальнейшей разработки функциональных и масштабируемых решений в области Интернета вещей.

ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. MQTT — Википедия. URL: <https://en.wikipedia.org/wiki/MQTT> (дата обращения: 09.12.2025).
2. Mosquitto_sub — руководство пользователя. URL: http://mosquitto.org/man/mosquitto_sub-1.html (дата обращения: 09.12.2025).
3. Mosquitto_pub — руководство пользователя. URL: https://mosquitto.org/man/mosquitto_pub-1.html (дата обращения: 09.12.2025).
4. MQTT: основа связи в IoT — Habr. URL: <https://habr.com/ru/post/463669/> (дата обращения: 09.12.2025).
5. Подключение к контроллеру по SSH — Wiren Board Wiki. URL: <https://wirenboard.com/wiki/SSH> (дата обращения: 09.12.2025).