МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ** **ПОЛИТЕХНИЧЕСКИЙ** **УНИВЕРСИТЕТ»** **(МОСКОВСКИЙ** **ПОЛИТЕХ)**

Факультет информационных технологий

Кафедра «Прикладная информатика»

Форма обучения: очная

**ОТЧЕТ**

**о** **выполнении** **лабораторной** **работы** **№3**

**по** **дисциплине**

**«Платформы интернета вещей»**

Студент

(личная подпись)

(И.И. Вобленко)

(И.О. Фамилия)

**Москва** **2022**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc103428496)

[1 Платформа ThingsBoard 6](#_Toc103428497)

[2 Развертывание платформы ThingsBoard 10](#_Toc103428498)

[3 Виртуальные усторйства 12](#_Toc103428499)

[4 Механизмы правил 16](#_Toc103428500)

[5 Обратка сигналов тревоги 20](#_Toc103428501)

[6 Email оповещения 23](#_Toc103428502)

[7 Визуализация данных 26](#_Toc103428503)

[ЗАКЛЮЧЕНИЕ 29](#_Toc103428504)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30](#_Toc103428505)

# ВВЕДЕНИЕ

В данной лабораторной работе рассмотрено понятие «Платформы интернета вещей».

Актуальность данной работы заключается в том, что системы интернета вещей используются практически повсеместно от банального автоматизированного склада до умных городов. Чтобы поддерживать эти системы в рабочем состоянии и иметь возможность управлять ими крайне необходимо уметь пользоваться различными платформами интернета вещей, а также понимать их достоинства и недостатки [1].

Целью работы является изучение и освоение платформ интернета вещей.

Объектом исследования является платформа интернат вещей ThingsBoard.

Предметом исследования являются технологии платформ интернета вещей для сбора, обработки и визуализации данных, а также управления устройствами.

Платформа интернета вещей (далее IoT) – это комплекс программ, которые применяются для подключения различных интернет вещей к облачной инфраструктуре хранения информации и предоставления удаленного доступа к ним.

На данный момент наиболее востребованными программными платформами для интернета вещей считаются:

* облачная платформа Azure от компании Microsoft;
* коммерческое публичное облако Web Services от компании Amazon;
* Cloud Platform от компании Google;
* платформа для промышленных инноваций ThingWorx;
* суперкомпьютер Watson от компании IBM и др.

Программные IoT платформы отличаются по следующим техническим характеристикам:

* Масштабируемость – максимально возможное число интернет вещей, подключенных к платформе.
* Простота применения – гибкость программного интерфейса приложения и простота работы с исходником.
* Способы развертывания – частное или публичное облачное хранилище.
* Безопасность – обеспечение защиты информации посредством шифрования, мониторинга доступа и т.д.
* База данных – организационная структура хранения информации, которая передается с устройств.

В процессе использования IoT платформ применяется множество протоколов. Самыми востребованными вариантами считаются Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT), DirectDraw Surface (DDS), eXtensible Messaging and Presence Protocol (XMPP) и HyperText Transfer Protocol Secure (HTTP/HTTPS) [2].

Задачи, решаемые в процессе выполнения лабораторной работы:

1. Изучение платформы ThingsBoard;
2. Развертывание платформы ThingsBoard;
3. Изучение и создание виртуальных устройств;
4. Изучение и использование механизмов правил;
5. Изучение и использование обработки сигналов тревоги;
6. Изучение и использование системы email оповещений;
7. Изучение методов визуализации данных.

# Платформа ThingsBoard

ThingsBoard — это платформа Интернета вещей с открытым исходным кодом для сбора, обработки, визуализации данных и управления устройствами.

Он обеспечивает подключение устройств по стандартным отраслевым протоколам Интернета вещей - MQTT, CoAP и HTTP, а также поддерживает как облачные, так и локальные развертывания. Things Board сочетает в себе масштабируемость, отказоустойчивость и производительность, так что вы никогда не потеряете свои данные [3].

Платформа позволяет [4]:

* быстро разрабатывать, сопровождать и масштабировать IoT проекты
* интегрироваться с другими сервисами обработки IoT данных (AWS IoT, Apache Spark, SigFox, IBM Watson IoT)
* ретранслировать данные устройств в другие системы
* подключать устройства через стандартные протоколы IoT — MQTT, CoAP и HTTP
* создавать панели мониторинга для визуализации данных

Можно выделить следующие основные особенности платформы ThingsBoard.

Сбор телеметрических данных

Сбор и хранение телеметрических данных надежным способом, выдерживая сетевые и аппаратные сбои. Доступ к собранным данным осуществляется с помощью настраиваемых веб-панелей мониторинга или серверных API.

Многоквартирный дом

Поддержка многопользовательских установок "из коробки". У одного клиента может быть несколько администраторов клиентов и миллионы устройств и клиентов.

Визуализация данных

Предоставляет более 30 настраиваемых виджетов из коробки и возможность создавать свои собственные виджеты с помощью встроенного редактора. Встроенные линейные диаграммы, цифровые и аналоговые датчики, карты и многое другое.

Горизонтальная масштабируемость

Количество поддерживаемых серверных запросов и устройств линейно увеличивается по мере добавления новых серверов things board в режиме кластеризации. Никаких простоев, перезапусков сервера или ошибок приложений.

Механизм правил IoT

Обрабатывайте входящие данные устройства с помощью гибких цепочек правил, основанных на атрибутах объектов или содержимом сообщений. Пересылка данных во внешние системы или запуск аварийных сигналов с использованием пользовательской логики. Настройте сложные цепочки уведомлений о тревогах. Расширяйте функциональность на стороне сервера или управляйте своими устройствами с помощью настраиваемых правил. Определите логику вашего приложения с помощью конструктора цепочки правил drag-n-drop.

Отказоустойчивость

Все вещи на бортовых серверах идентичны. Никаких мастеров-рабочих или горячего ожидания. Сбой узла обнаруживается автоматически. Вышедшие из строя узлы могут быть заменены без простоев. Сохраненные данные реплицируются с использованием надежной базы данных NoSQL.

Управление устройствами

Предоставляет возможность регистрации устройств и управления ими. Позволяет отслеживать атрибуты устройств на стороне клиента и предоставлять их на стороне сервера. Предоставляет API для серверных приложений для отправки команд RPC на устройства и наоборот.

Безопасность

Поддерживает шифрование транспорта как для протоколов MQTT, так и для протоколов HTTP(s). Поддерживает проверку подлинности устройства и управление учетными данными устройства.

Управление активами

Предоставляет возможность регистрировать активы и управлять ими. Позволяет предоставлять атрибуты активов на стороне сервера и отслеживать связанные с ними аварийные сигналы. Возможность построения иерархии сущностей с использованием связей.

Настройка и интеграция

Расширяйте функциональность платформы по умолчанию с помощью настраиваемых цепочек правил, виджетов и транспортных реализаций. В дополнение к поддержке MQTT, CoAP и HTTP пользователи Things Board могут использовать свои собственные транспортные реализации или настраивать поведение существующих протоколов.

Управление аварийными сигналами

Предоставляет возможность создавать и управлять сигналами тревоги, связанными с вашими объектами: устройствами, активами, клиентами и т.д. Позволяет отслеживать сигналы тревоги в режиме реального времени и распространять сигналы тревоги по иерархии связанных объектов. Подавать сигналы тревоги при отключении устройства или событиях бездействия.

100% открытый исходный код

Things Board лицензирована под лицензией Apache License 2.0, поэтому вы можете бесплатно использовать любые ИТ-технологии в своих коммерческих продуктах. Вы даже можете разместить его как решение SaaS или PaaS.

Микросервисы или монолитность

Поддерживает монолитное развертывание для начала работы или небольших сред. Предоставляет возможность обновления до микросервисов для обеспечения высокой доступности и горизонтальной масштабируемости.

SQL, NoSQL и гибридная база данных

Поддерживает различные параметры базы данных и возможность выбирать, где хранить основные объекты и где хранить данные телеметрии [3].

# Развертывание платформы ThingsBoard

Так как системы интернета вещей имеют тенденцию сильно масштабироваться, а также должны быть гибкими в вопросе настройки и интеграции в системах пользователя, то рассмотрим наиболее универсальный способ развертывания платформы ThingsBoard, а именно развертывание посредством Docker-контейнера. Данный способ является системонезависимым (главное, чтобы под операционную систему используемой вычислительной системы была актуальная версия Docker), а также имеет быструю скорость развертывания, после одного раза настройки можно развернуть систему в пару команд.

В первую очередь необходимо установить Docker на вычислительную систему с официального сайта. Docker имеет версии под MacOS, Windows, Linux.

После этого на Docker Hub найдем актуальный контейнер плтаформы ThingsBoard. Таким оказывается контейнер thingsboard/tb-postgres. Далее копируем с сайта Docker Hub команду установки данного контейнера и устанавливаем его на вычислительную машину:

docker pull thingsboard/tb-postgres

При корректной установке контейнера будет получено следующее сообщение:

Digest: sha256:b4b9d2094b80bb05e58e3ac157aeabc253b6aa6c401ba6d7fa338e13462bf19f

Status: Downloaded newer image for thingsboard/tb-postgres:latest

docker.io/thingsboard/tb-postgres:latest

Далее с помощью следующих команд создадим директории для хранения данных и логов, а также для выдачи соответствующих прав доступа к этим директориям:

mkdir -p ~/.mytb-data && sudo chown -R 799:799 ~/.mytb-data

mkdir -p ~/.mytb-logs && sudo chown -R 799:799 ~/.mytb-logs

Проверим, что все установилось запустив контейнер с помощью команды:

docker run -it -p 9090:9090 -p 1883:1883 -p 7070:7070 -p 5683-5688:5683-5688/udp -v ~/.mytb-data:/data -v ~/.mytb-logs:/var/log/thingsboard --name mytb --restart always thingsboard/tb-postgres

После чего перейдем по указанному порту и нашему ip:

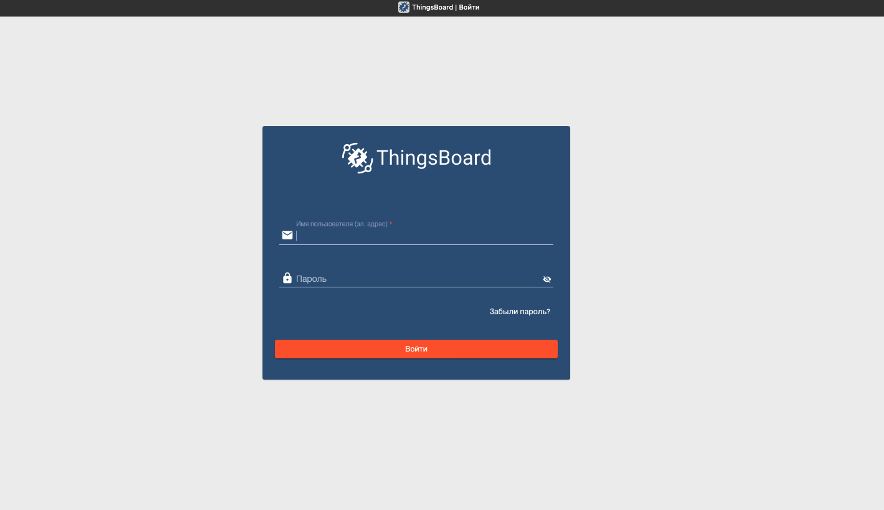


Рисунок 1– Запуск ThingsBoard

# Виртуальные усторйства

Виртуальное устройство — это описание устройства, применяемое для установления соединения между пользователем и физической рабочей станцией, подключенной к удаленной системе. Виртуальные устройства содержат информацию о физическом устройстве (дисплее или принтере), необходимую программам системы [5].

Создадим 3 виртуальных устройства в ThingsBoard. Для этого перейдем в раздел «Устройства» после чего нажмем значок плюса в правом верхнем углу, чтобы добавить новые устройства.

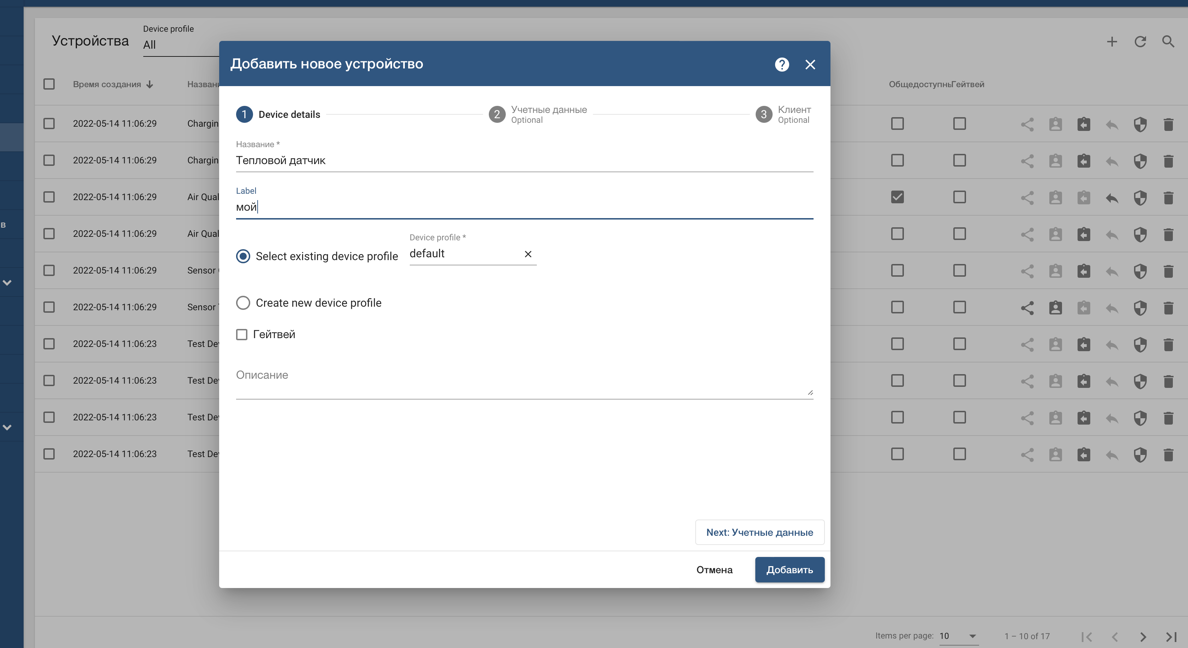
После этого заполним необходимые данные об устройстве.

Рисунок 2 – Добавление нового виртуального устройства

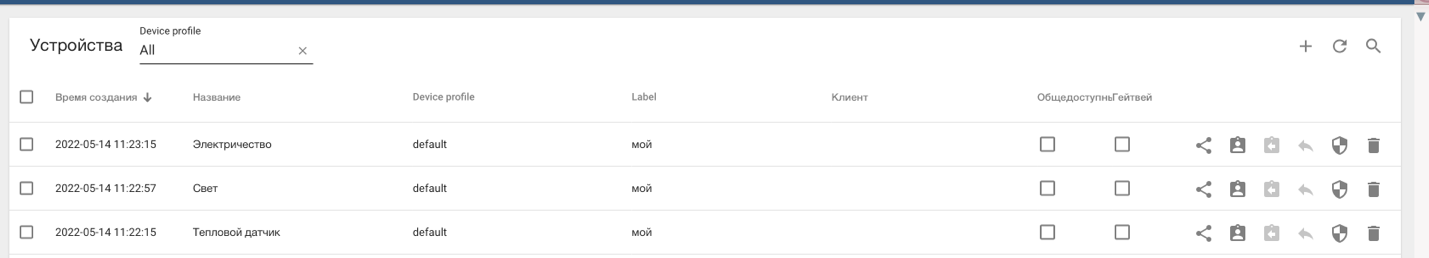


Рисунок 3 – Добавленные устройства

Протестируем возможность получения телеметрии устройств, получим напряжение устройства «Свет». Для этого скопируем уникальный токен (“ij5B5f0mw4m8QAN8EBLe”) данного устройства и воспользуемся следующей командой:

mqtt pub -q 1 -h "demo.thingsboard.io" -t "v1/devices/me/telemetry" -u "ij5B5f0mw4m8QAN8EBLe" -m "{'U':143}"

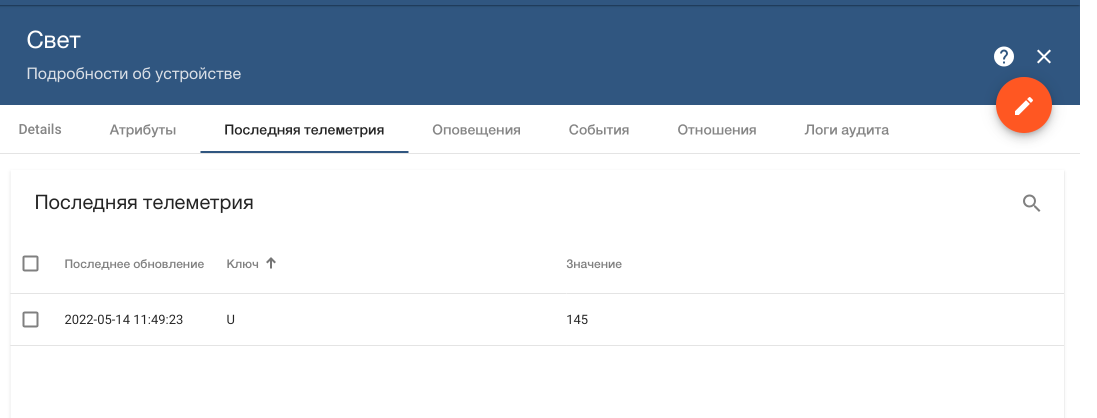
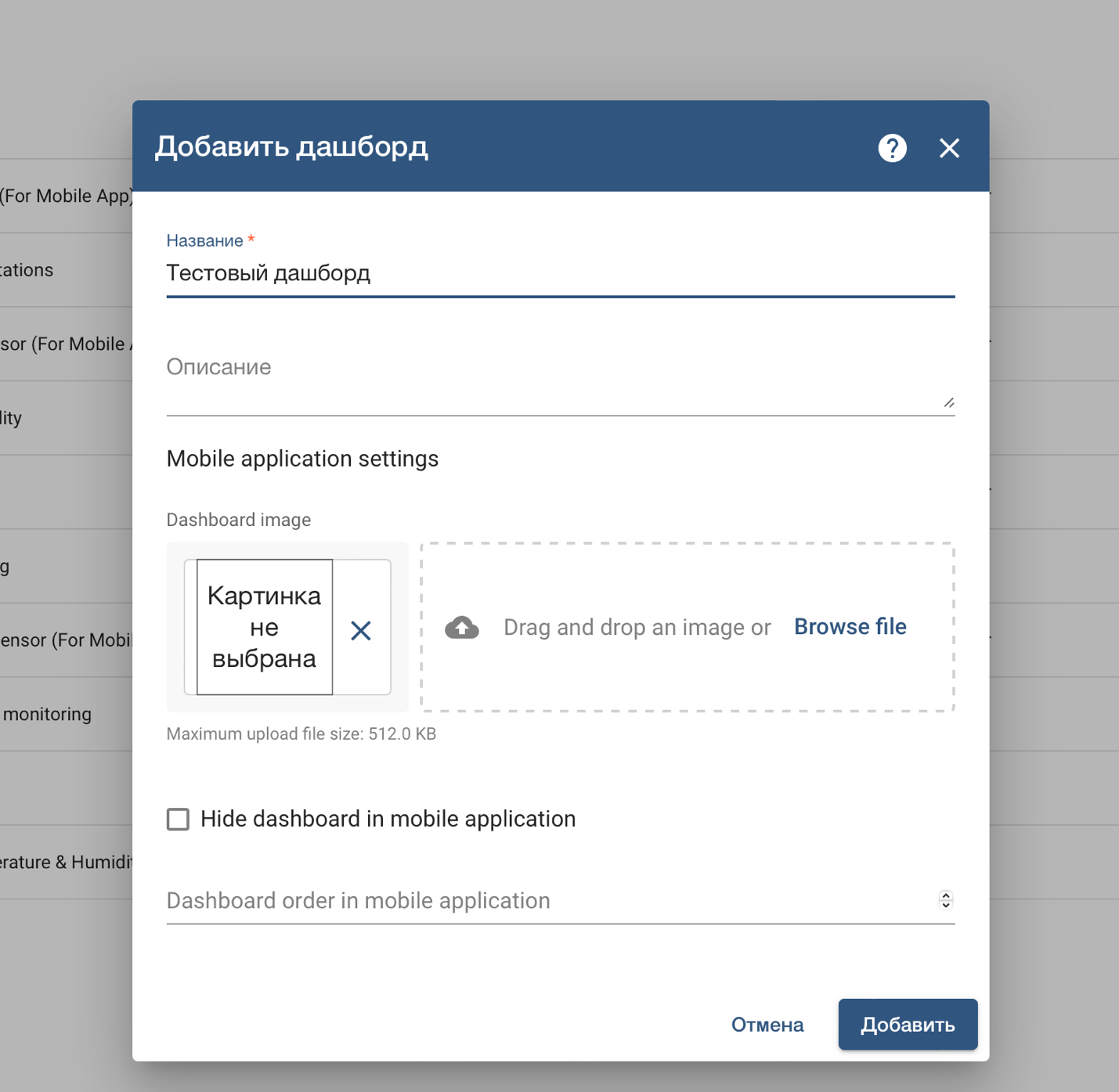
После чего в ThingsBoard посмотрим последнюю телеметрию устройства «Свет»:

Рисунок 4 – Отправка телеметрии

Теперь создадим новый виджет для отображения полученной телеметрии устройства «Свет». Сначала создадим новый дашборд.

Рисунок 5 – Создание нового дашборда

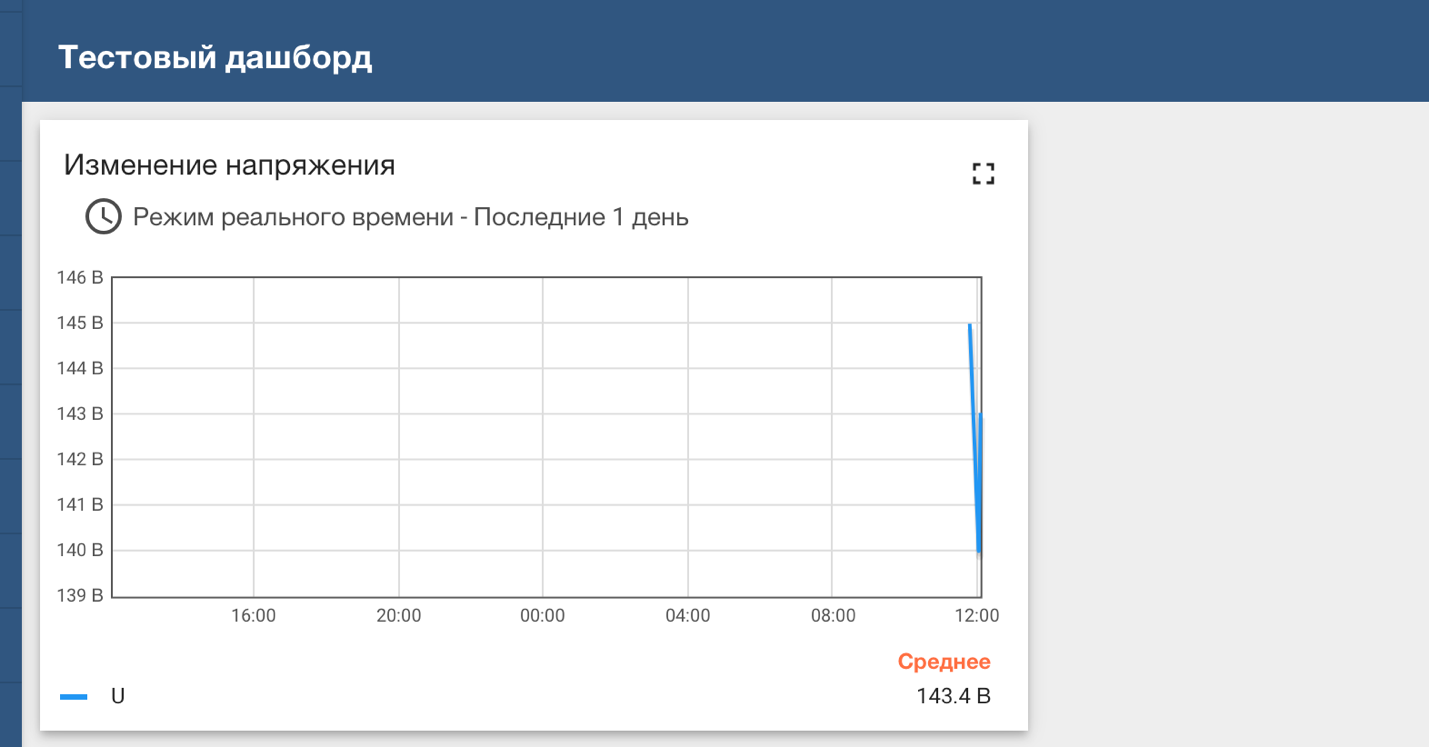
Далее добавим новый виджет на наш тестовый дашборд. Это виджет с отображение изменения напряжения в виде линейного графика. Также там отображается среднее значение напряжение за последние 5 минут. Временной интервал отображения изменения напряжения – 1 день.

Рисунок 6 – Добавление нового виджета

# Механизмы правил

Механизм правил - компонент, встроенный в приложение для достижения разделения бизнес-решений от кода приложения и использования предопределенных бизнес-решений для принятия семантических модулей, записанных данных. вводите данные, объясняйте бизнес-правила и принимайте бизнес-решения на основе бизнес-правил. Механизм обработки ядра платформы Things board, подобный двигателю автомобиля, предназначен для построения рабочего процесса на основе событий и представляет собой настраиваемую структуру со сложной обработкой событий.

Основные возможности [6]:

* сохраненные в данных устройства (данные в реальном времени / данные атрибутов) перед поддержкой базы данных для данных телеметрии или данных атрибутов, полученных для проверки и модификации перехвата;
* данные телеметрии устройства (данные в реальном времени) копия атрибута устройства или с устройства на связанные активы, так что может быть агрегирована телеметрия;
* пользовательские правила / функция оповещения, для срабатывания сигнала тревоги устройства, обновления, очистки;
* пусковое устройство, работающее в соответствии с событиями жизненного цикла; Устройство, такое как линия, устройство в автономном режиме, создание аварийного события
* дополнительные данные, необходимые для обработки
* загрузка. Температура устройства загрузки в определенном пороговом значении или свойства клиента устройства клиента;
* при передаче электронной почты происходит сложное событие, и с помощью атрибута "шаблон электронной почты" другие объекты;
* устройство дистанционного управления в соответствии с условиями, определенными передачей RPC.
* устройство передачи данных в реальном времени, команда RPC удаленного управления или сигнал тревоги или событие, инициируемое механизмом правил, для отправки в промежуточное программное обеспечение сообщений или во внешние сторонние системы (например, Kafka, MQTT, RabbitMQ, Spark, AWS services и т.д.)

Основные части механизма правил [6]:

* Сообщение о правилах (Сообщение)
* Узел правил (Узел правил)

Настроим механизм правил ThingsBoard для обработки данных с устройства «Свет». Для этого сначала перейдем в раздел цепочки правил и добавим узел проверки напряжения. После перетащим узел правил Script Filter и введем туда следующий скрипт для проверки напряжения:

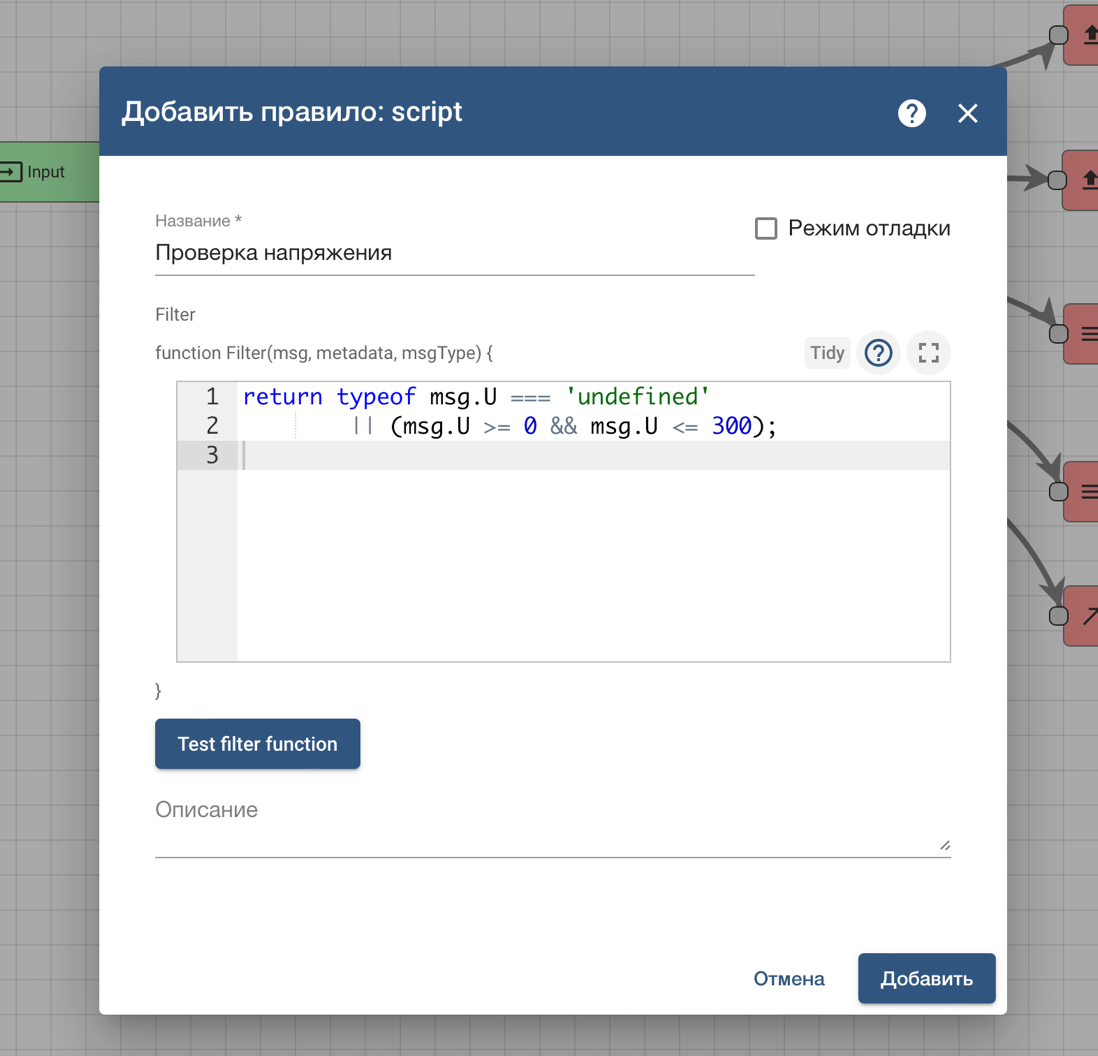
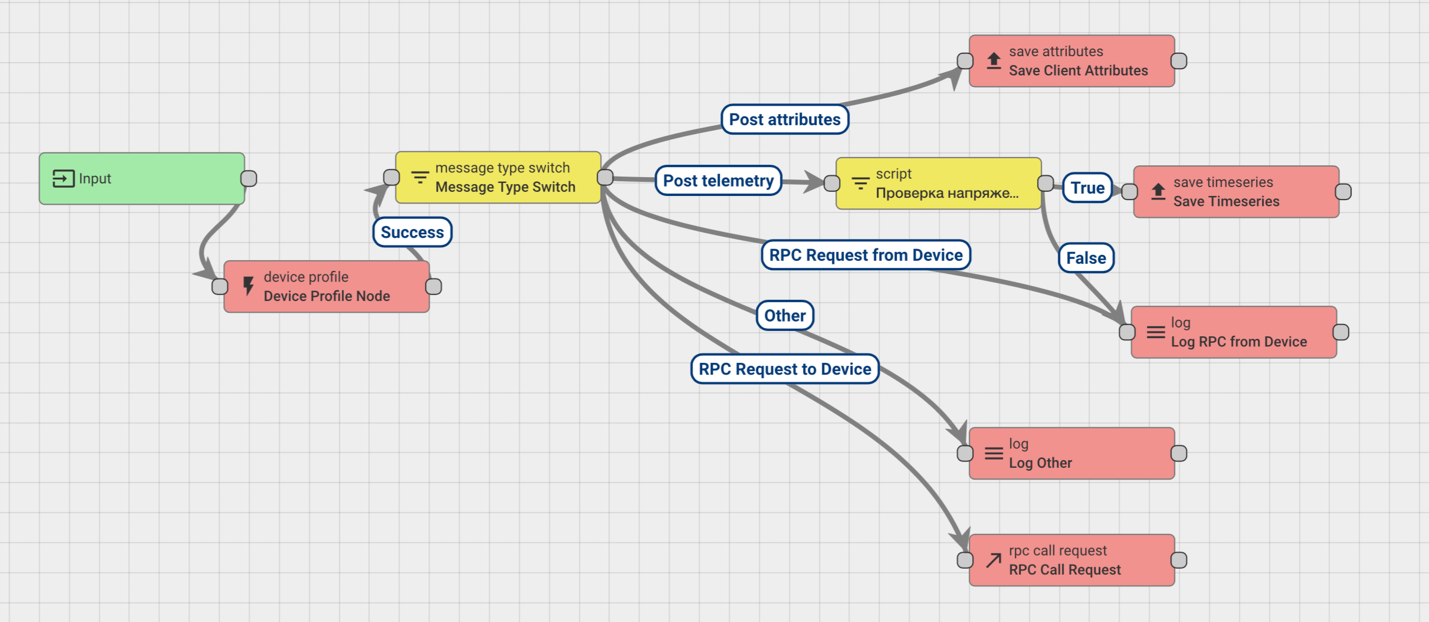
return typeof msg.U === 'undefined' || (msg.U >= 0 && msg.U <= 300);

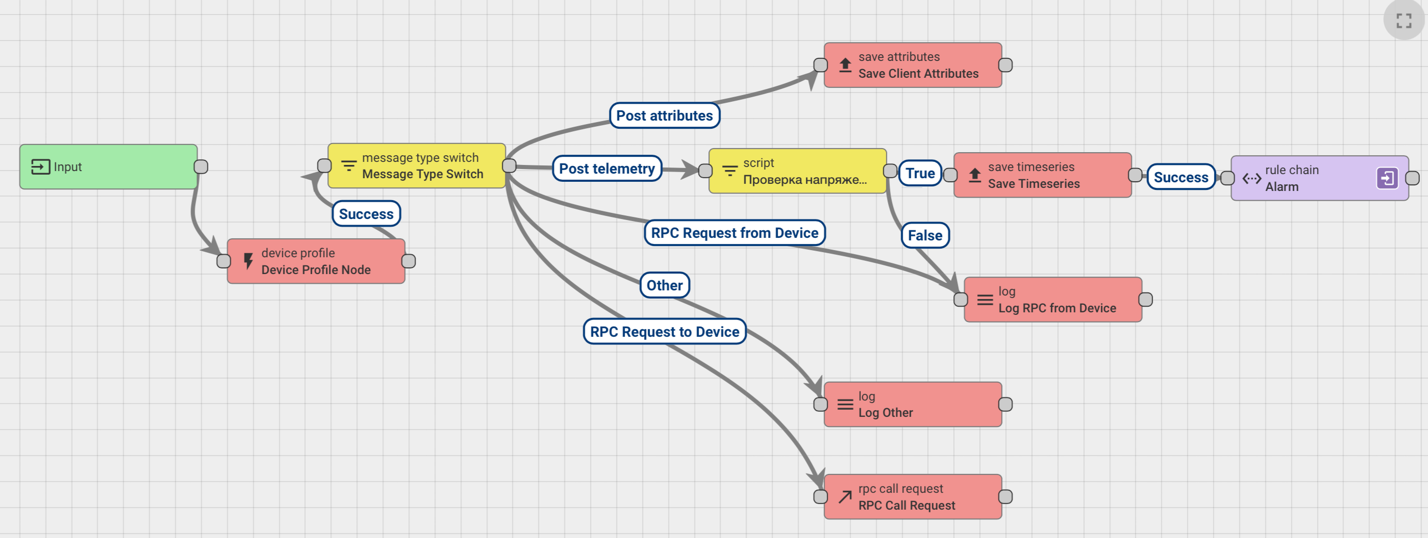
Рисунок 7 – Создание нового правила

Теперь установим новое правило в схеме правил для проверки отправляемой телеметрии, если все хорошо и возвращается значение True, то отправляем данные на сохранение, в противном случае выводим логи.

Рисунок 8 – Установка правила для проверки данных

# Обратка сигналов тревоги

Сигналы тревоги необходимы для сообщения о выходе значений нашей телеметрии за допустимые пределы. Например, при увеличении значений напряжения в 250 В, что может привести к поломки системы. Создадим новое правило для сигнала тревоги при поступлении напряжения больше 250В. Добавим переход к новой цепочке правил для сигналов тревоги.

Рисунок 9 – Переход к новой цепочке правил

Создадим и добавим следующий скрипт проверки входных данных:

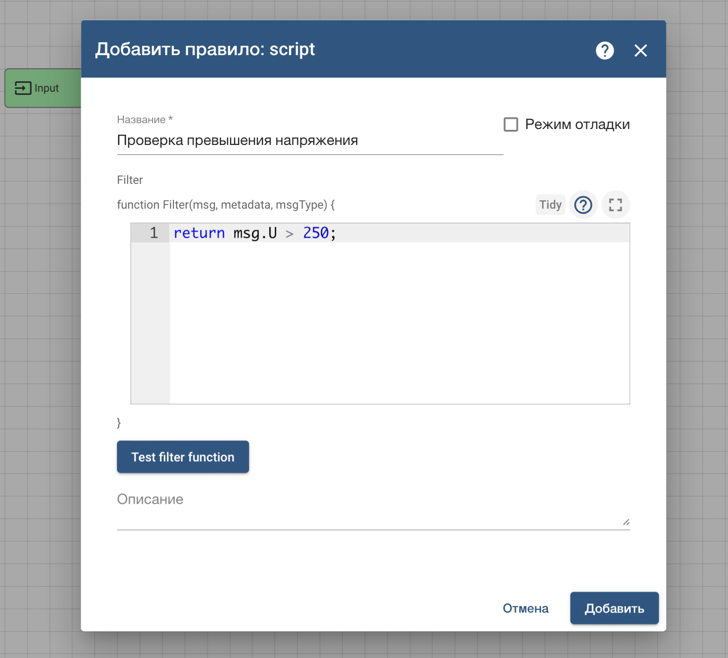
return msg.U > 250;

Рисунок 10 – Создание фильтра проверки превышения допустимых значений напряжения

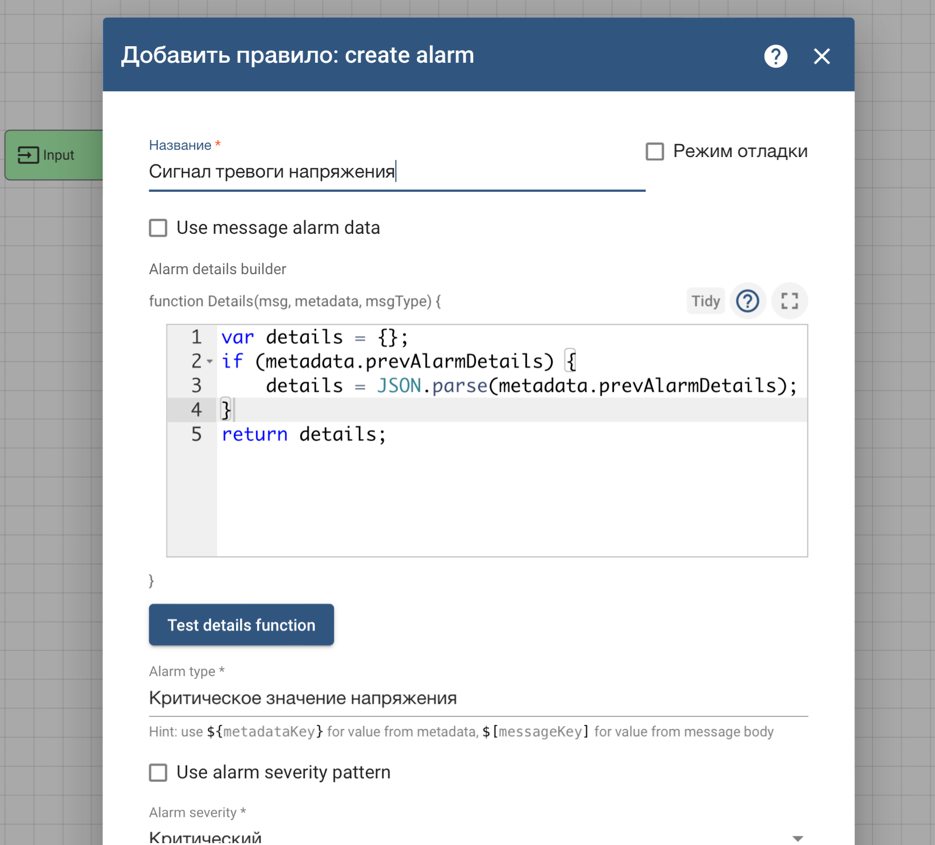
Далее создадим сигнал тревоги.

Рисунок 11 – Добавление сигнала тревоги

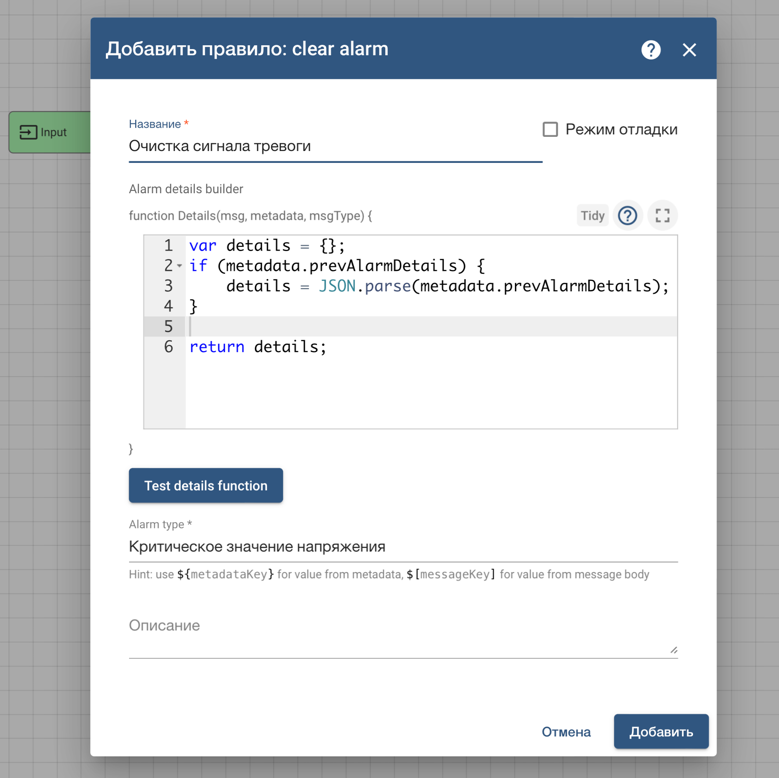
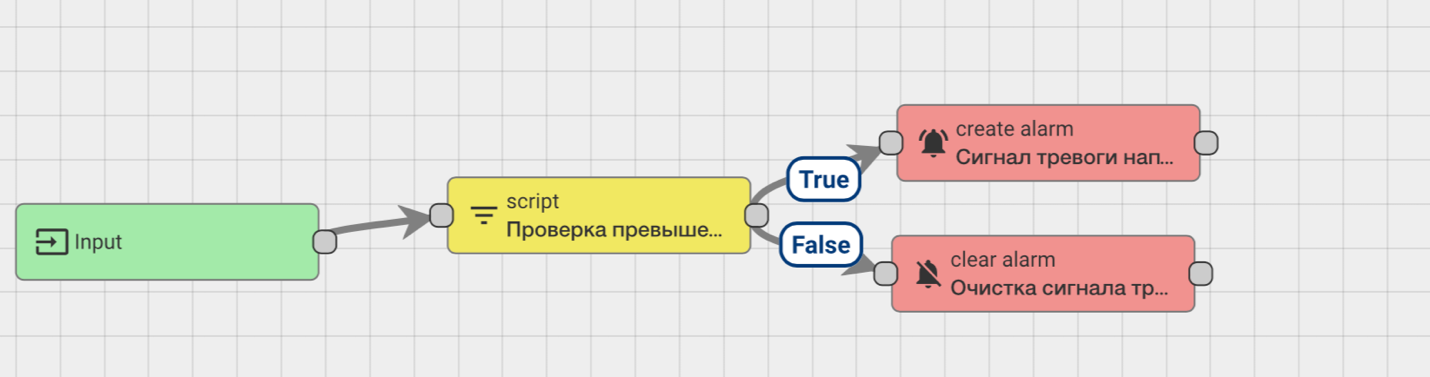
Добавим очистку сигнала тревоги.

Рисунок 12 ­– Очистка сигнала тревоги

Рисунок 13 – Итоговый вид цепочки правил

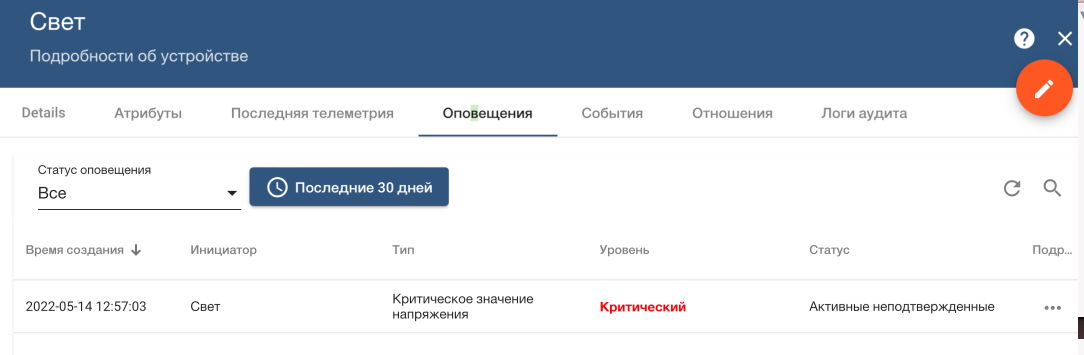
Теперь проверим работоспособность сигнала тревоги, отправив значение напряжения равное 268В и получим сигнал тревоги в оповещениях устройства.

Рисунок 14 – Сигнал тревоги в оповещениях

# Email оповещения

Помимо оповещений в UI самого ThingsBoard есть возможность настроить отправку email оповещений для сигналов тревоги. Для этого модифицируем нашу цепочку правил для обработки критических значений напряжения.

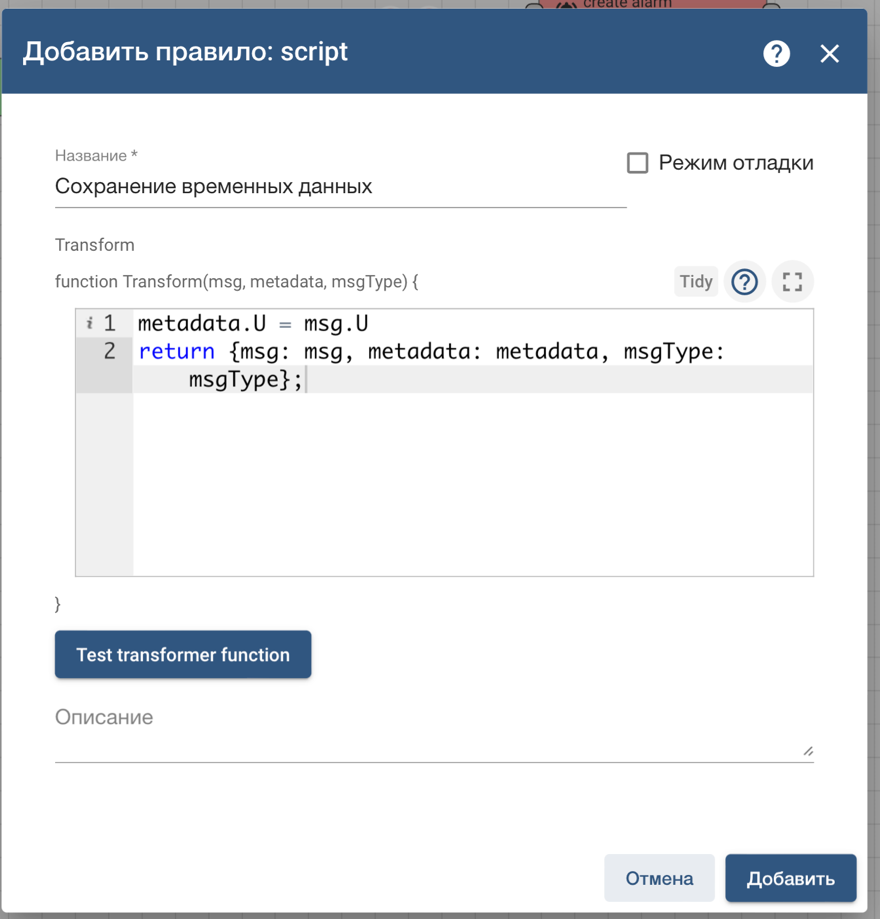
Сначала создадим скрипт сценария преобразования.

Рисунок 15 – Скрипт сценария преобразований

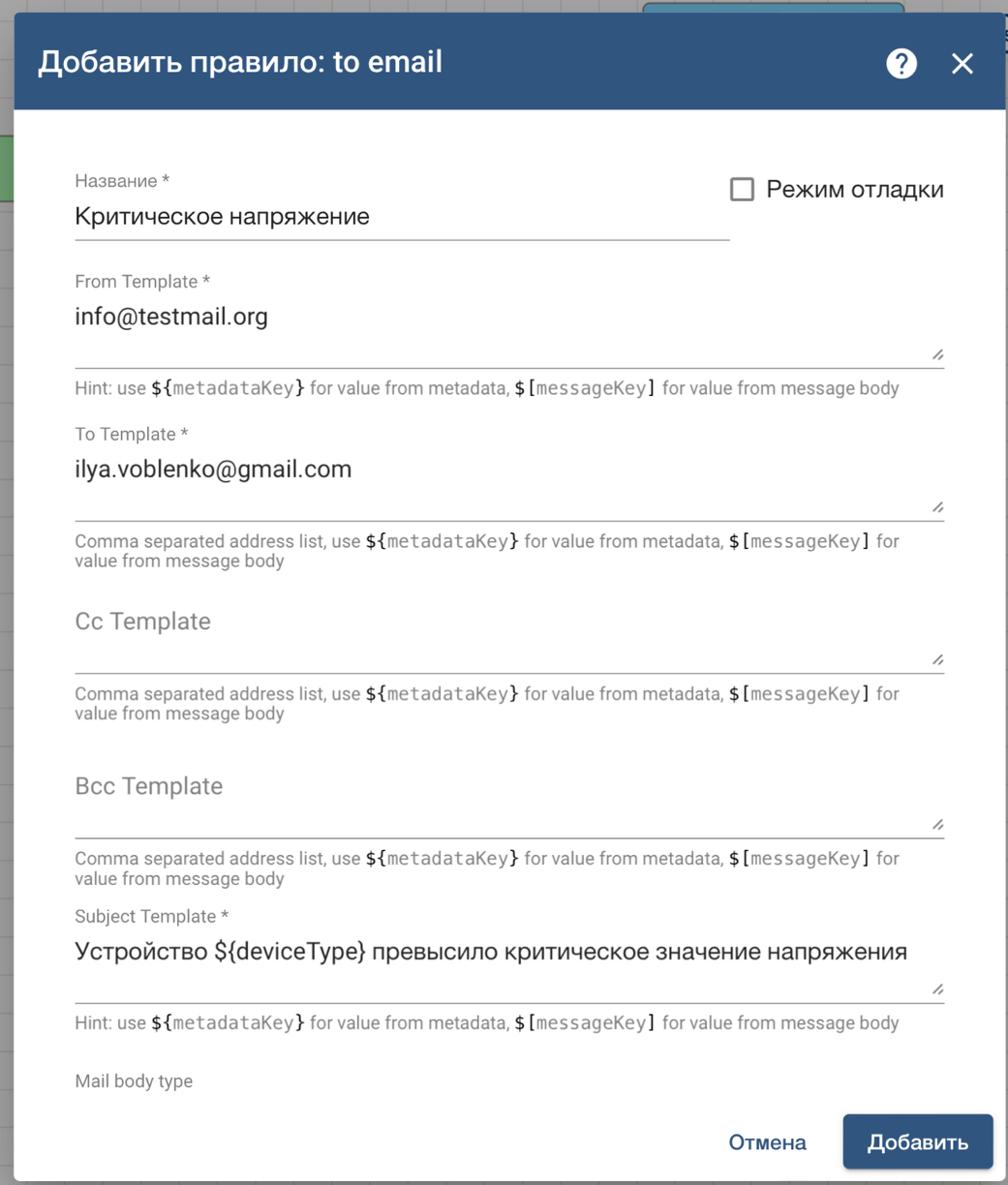
Далее создадим метод отправки сигналов тревоги по email.

Рисунок 16 – Метод отправки email

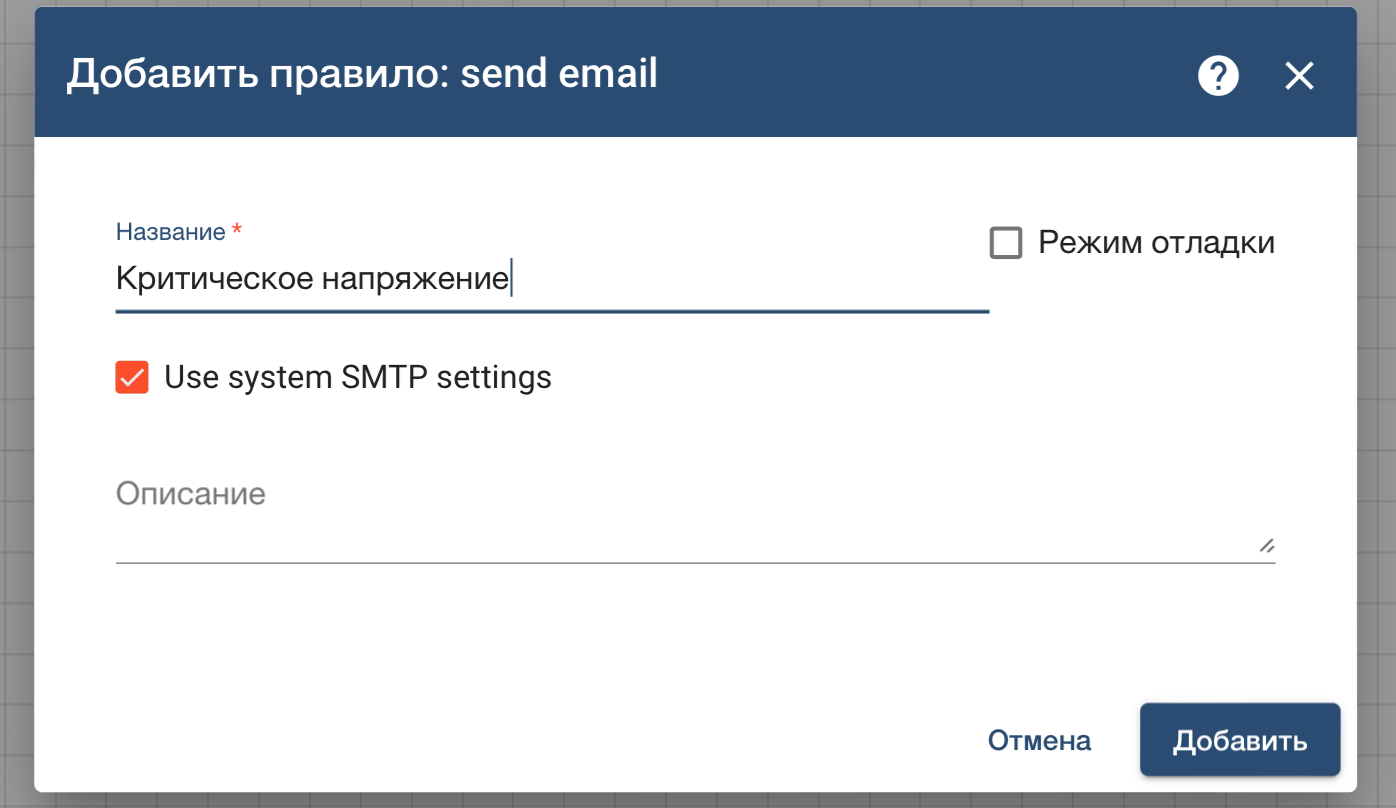
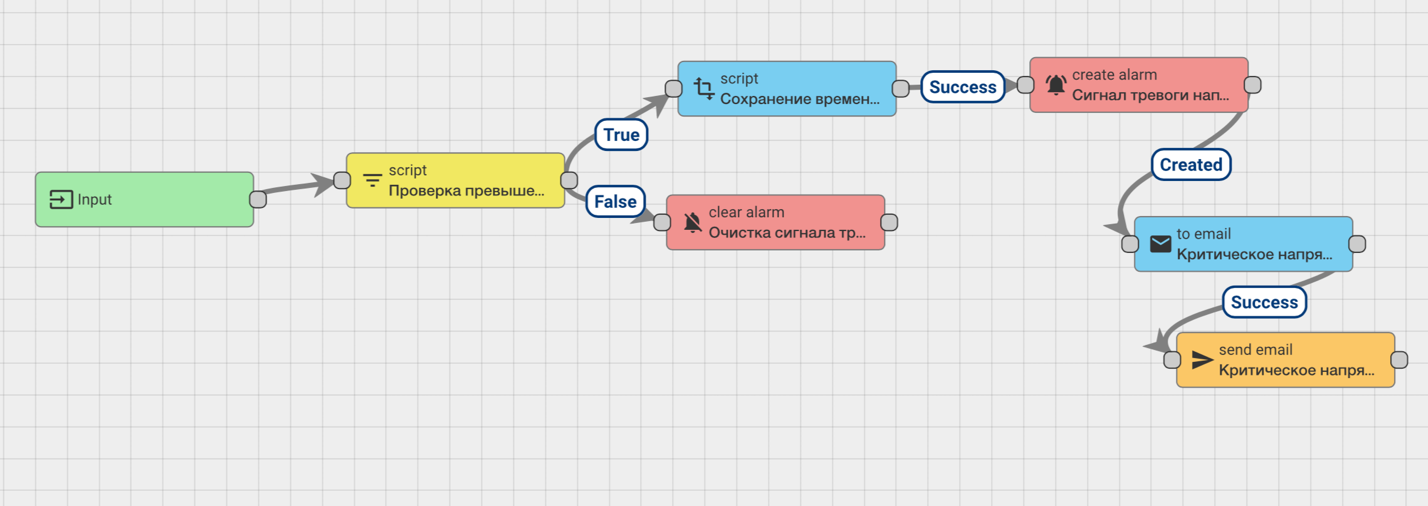
В конце добавим метод для отправки электронного письма.

Рисунок 17 – Метод отправки электронного письма

Итоговая схема правил имеет следующий вид:

Рисунок 18 – Итоговая цепочка правил для отправки email

# Визуализация данных

Платформа ThingsBoard предоставляетс очень гибкую и простую в использовании систему виджетов, которые группируются в единый дашборд. Это позволяет выводит на один экран всю необходимую информацию по системе интернета вещей и отслеживать ее работу в реальном времени в простом для восприятия формате.

Платформа ThingsBoard предоставляет более 30 настраиваемых виджетов из коробки и возможность создавать свои собственные виджеты с помощью встроенного редактора. Встроенные линейные диаграммы, цифровые и аналоговые датчики, карты и многое другое [3].

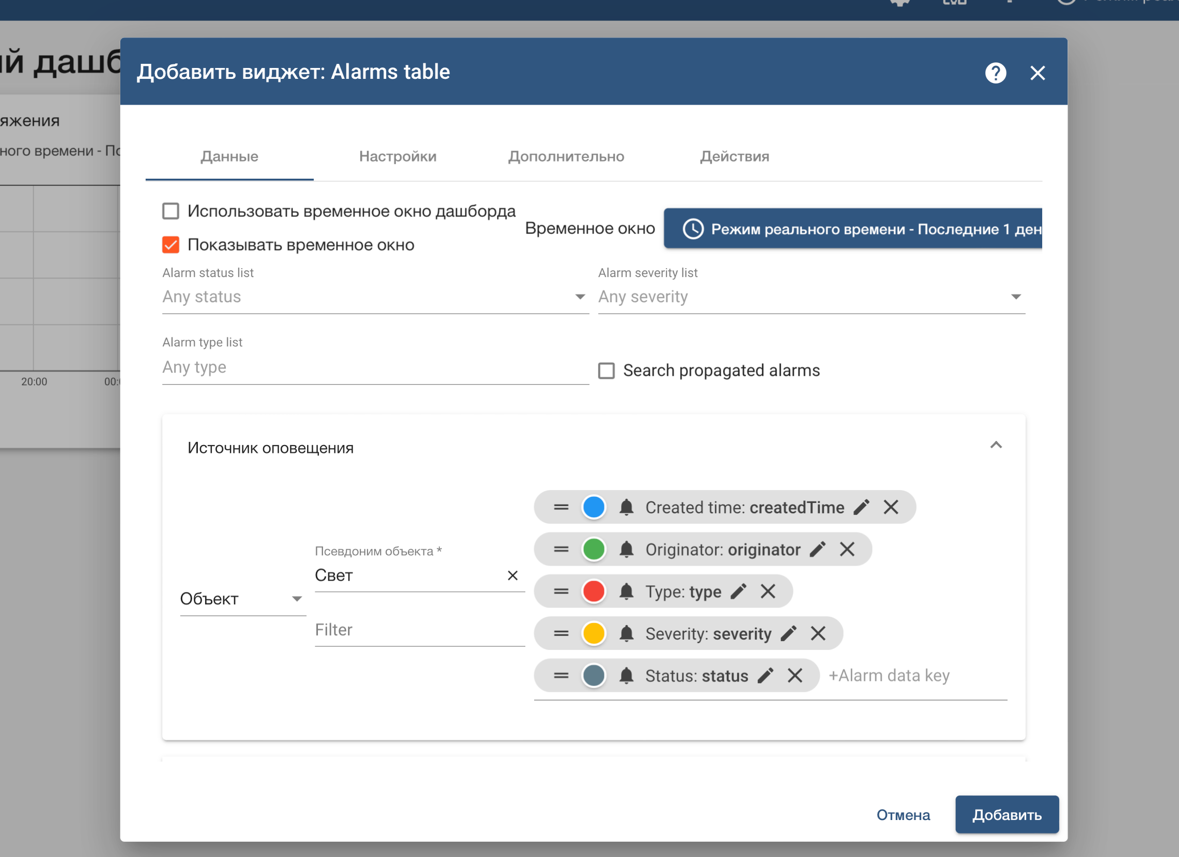
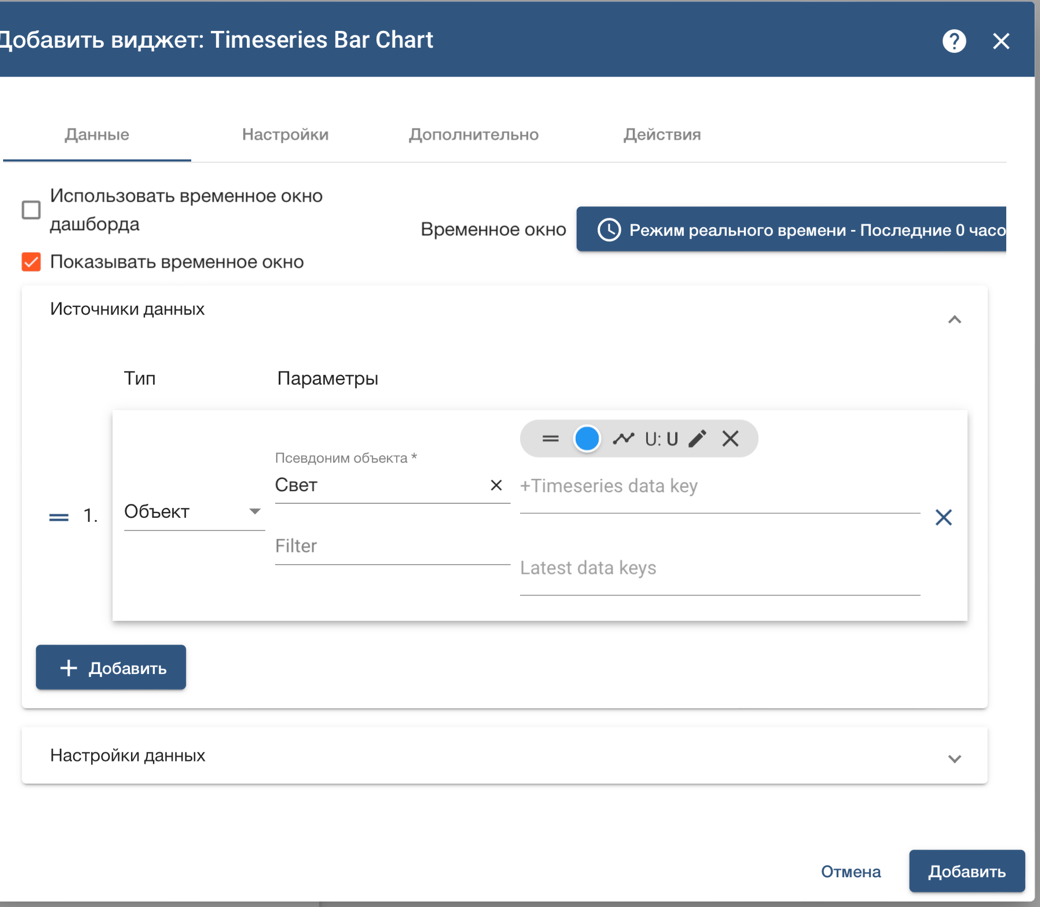
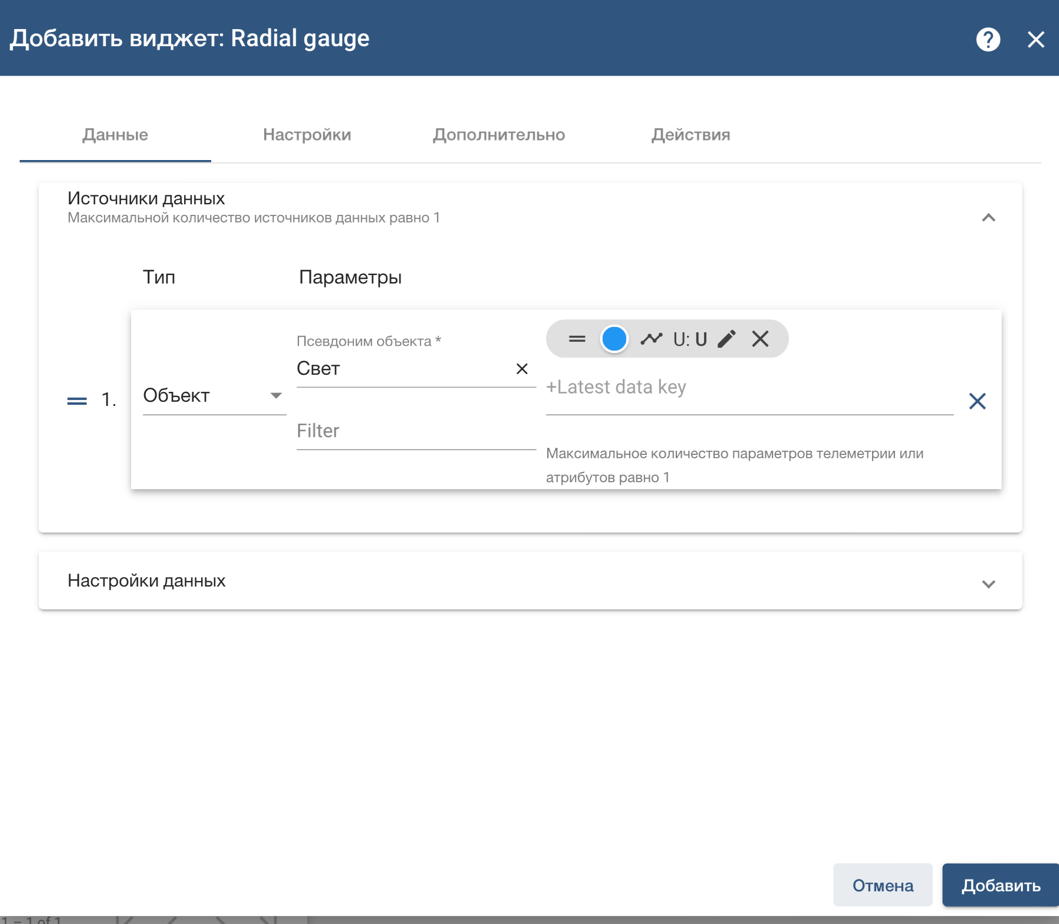
 Дополним тестовый дашборд. Добавим виджет с отображение сигналов тревоги в виде таблицы со следующими настройками:

Рисунок 19 – Виджет таблицы сигналов тревоги

Также добавим виджед для отслеживания динамики напряжения на устройстве «Свет» в рамках 1 часа.

Рисунок 20 – Виджет динамики напряжения устройства «Свет»

А также добавим аналоговый виджет для отслеживания текущих значений напряжения устройства «Свет» со следующими настройками:

Рисунок 21 – Аналоговый виджет

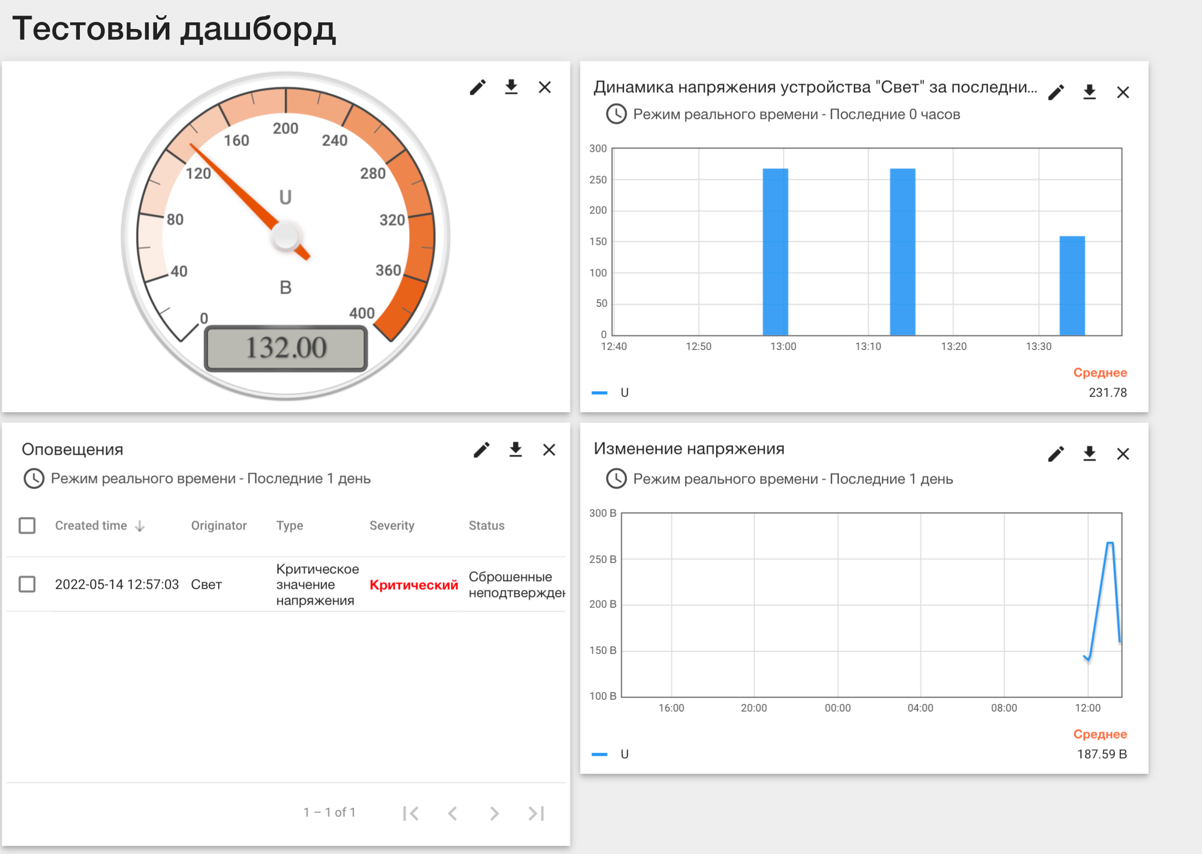
В итоге дашборд будет состоять из 4 виджетов и примет следующий вид:

Рисунок 22 – Итоговый вид тестового дашборда

# ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы мы познакомились с платформами интернета вещей, в частности с платформой ThingsBoard. Изучили основные возможности и особенности данной платформы интернета вещей. Научились методу развертывания платформы ThingsBoard на вычислительных машинах с помощью Docker-контейнеров. Познакомились с понятием виртуального устройства, создали 3 тестовых виртуальных устройства, проверили поступления телеметрических данных на эти устройства. Изучили механизм правил платформы ThingsBoard, научились изменять базовую Цепочку правил и добавлять в нее собственные скрипты. Освоили обработку сигналов тревоги для отслеживания критических показателей телеметрии, получаемых с устройств, создания и настройку новой цепочки правил для обработки сигналов тревоги. Создали правила для отправки email оповещений при срабатывании сигналов тревоги. А также изучили различные методы визуализации данных на платформе ThingsBoard и создали свой дашборд для контроля работы устройства «Свет».

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Центр2М «Воздействие Интернета вещей на различные отрасли» [Электронный ресурс] URL: https://center2m.ru/iot-business-model-zakupok#rec77621887 (дата обращения: 12.05.2022).
2. Intelvision «IoT платформа» [Электронный ресурс] URL: https://www.intelvision.ru/blog/internet-of-things (дата обращения: 12.05.2022)
3. ThingsBoard.io официальный сайт [Электронный ресурс] URL: https://thingsboard.io (дата обращения: 12.05.2022)
4. Habr «Демо доступ к IoT платформам» [Электронный ресурс] URL: https://habr.com/ru/company/euler2012/blog/329614/ (дата обращения: 12.05.2022)
5. IBM «Описание виртуальных устройств» [Электронный ресурс] URL: https://www.ibm.com/docs/ru/i/7.3?topic=server-virtual-device-descriptions (дата обращения 14.05.2022)
6. CodeTD «ThingsBoard Detailed rules engine, rules engine how to use TB» [Электронный ресурс] URL: https://www.codetd.com/en/article/10206096 (дата обращения 14.05.2022)