

Методические указания по выполнению практических работ № 13-14

«Визуализация и удаленное управление посредством облачных платформ интернета вещей»

по дисциплине «Технологические основы Интернета вещей»

Оглавление

Практическая работа №13 – Визуализация данных в облачных платформах	2
Визуализация	2
Дашборды ThingsBoard	2
Виджеты ThingsBoard	3
Привязки устройств ThingsBoard	6
Пример создания дашборда для мониторинга	6
Задание практической работы №13	10
Дополнительное задание практической работы №13	11
Практическая работа №14 – Управление устройствами при помощи панелей управления облачных платформ	12
Управляющие виджеты ThingsBoard	12
Пример создания дашборда для управления устройством	13
Задание практической работы №14	20
Дополнительное задание практической работы №14	21
Литература для изучения:	21

Практическая работа №13 – Визуализация данных в облачных платформах

Визуализация

В решениях Интернета вещей данные могут визуализироваться на различных уровнях реализуемого решения. Можно визуализировать данные на уровне локального сервера, развернутого в той же сети, к которой подключены микроконтроллеры. Можно визуализировать данные для конечного пользователя, предоставляя ему динамику изменения отслеживаемых параметров, например график изменения температуры в отслеживаемом помещении. Однако, получить полную визуализацию всех данных, обрабатываемых IoT-сервисом, можно в облачной платформе.

В облаке можно собирать как данные телеметрии, к примеру, температуру, влажность, давление и т.д., так и служебную информацию, например, состояние устройств, температура внутренних модулей устройства и т.д. Таким образом можно отслеживать как состояние окружающей среды, данные о которой собирают устройства, так и состояние самих устройств, отслеживая тем самым состояние работы всей системы.

Платформа ThingsBoard, как и любая облачная платформа, включает в себя функционал для визуализации данных.

Дашборды ThingsBoard

Визуализация в платформе ThingsBoard реализуется через специальный механизм, называемый дашбордами. Дашборд представляет из себя область для размещения специальных элементов, предназначенных либо для визуализации информации (графики, измерительные шкалы и другие средства отображения параметров устройства), а также для управления физическими устройствами (Кнопки, поля ввода, потенциометры и т. д.).

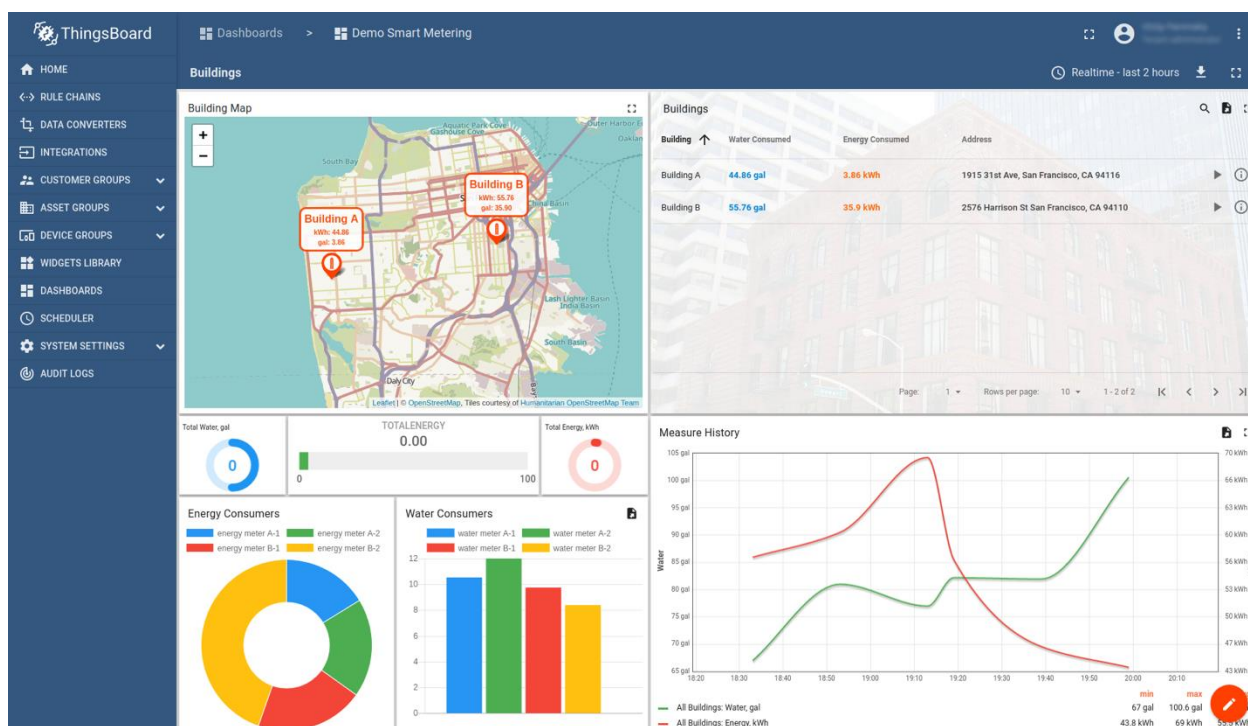


Рисунок 1 – Пример дашборда ThingsBoard

Виджеты ThingsBoard

Платформа ThingsBoard имеет обширную библиотеку виджетов, которая может расширяться дополнительными виджетами, написанными пользователями облачной платформы. Рассмотрим основные типы виджетов, представленных в платформе. Всего существует 5 типов виджетов:

- Последние значения;
- Временная последовательность;
- RPC (виджет управления);
- Виджет тревог;
- Статический виджет.

Последние значения

Виджет типа «Последние значения» отображает последние значения определенного атрибута сущности или точки данных временного ряда. Этот тип виджетов использует значения атрибута в качестве источника данных. Например, для датчика температуры можно отображать текущее значение температуры.

Временная последовательность

Виджеты типа «Временной последовательности» отображает исторические значения за выбранный период времени или последние значения в определенном временном промежутке. Временной интервал может быть как в реальном времени - динамически изменяемый таймфрейм для определенных последних значений, так и в хронологическом - фиксированный исторический таймфрейм.

RPC (виджет управления)

Виджет управления позволяет отправлять RPC-команды устройствам, обрабатывать и визуализировать ответ от устройства. Виджеты RPC настраиваются путем указания целевого устройства в качестве целевой конечной точки для команд RPC.

Виджет тревог

Виджет тревог отображает алармы, относящиеся к указанному объекту, в определенном временном интервале. Виджет сигналов тревоги настраивается путем указания объекта в качестве источника сигнала тревоги и соответствующих полей сигналов тревоги, которые будут отображаться. Можно задать временной интервал, за который необходимо вывести список тревог.

Статический

Статический виджет отображает статическое настраиваемое HTML-содержимое. Статические виджеты не используют никаких источников данных и обычно настраиваются путем указания статического содержимого HTML и, при необходимости, стилей CSS.

Библиотека виджетов (пакеты)

Виджеты сгруппированы в пакеты виджетов в соответствии с их назначением. Существуют пакеты виджетов на уровне системы и на уровне клиента.

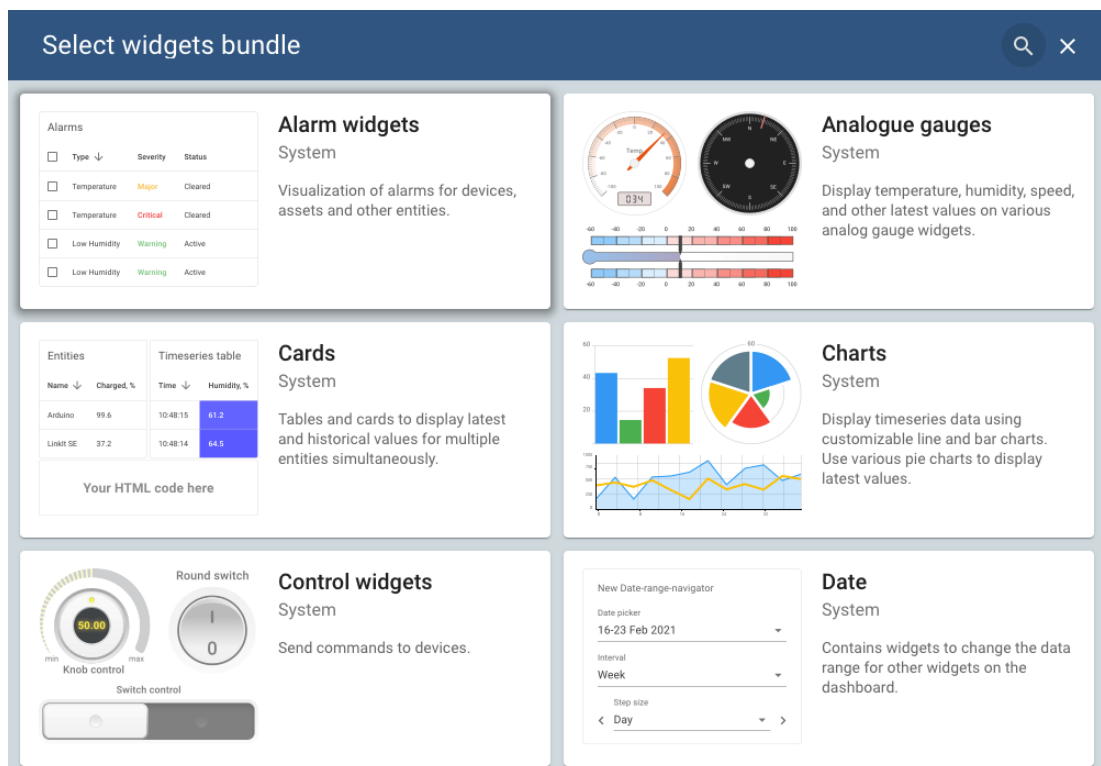


Рисунок 2 – Перечень пакетов виджетов

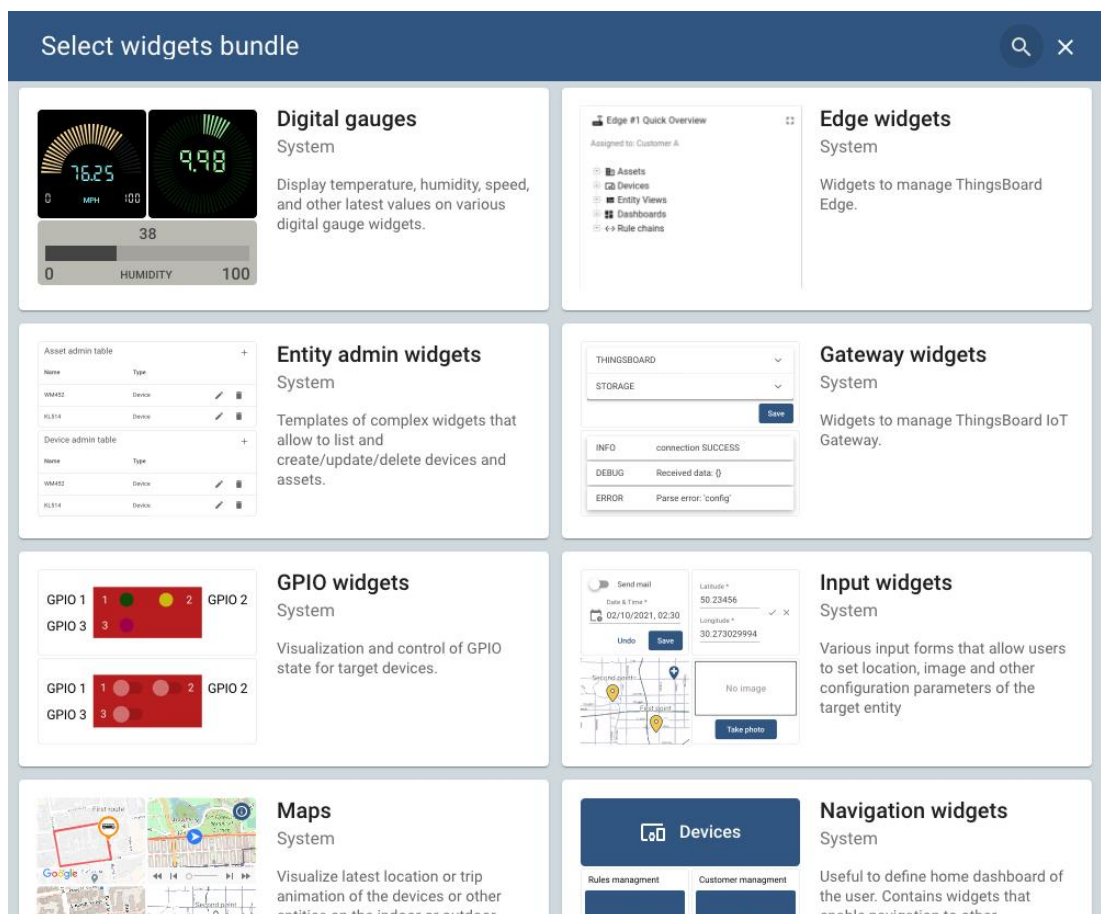


Рисунок 3 – Перечень пакетов виджетов (продолжение)

Виджеты тревог

Пакет виджетов сигналов тревоги полезен для визуализации сигналов тревоги для определенных объектов как в режиме реального времени, так и в режиме истории.

Аналоговые шкалы

Пакет аналоговых шкал полезен для визуализации температуры, влажности, скорости и других целочисленных или плавающих значений.

Карточки

Пакет карточки полезен для визуализации временных рядов данных или атрибутов в виджетах таблиц или карточек.

Графики

Пакет графиков полезен для визуализации данных в виде временных последовательностей наборов данных в реальном времени или же для просмотра накопленных данных.

Виджеты управления

Пакет управляющих виджетов полезен для визуализации текущего состояния и отправки команд RPC на целевые устройства.

Виджеты даты

Пакет виджетов даты полезен для изменения диапазона данных для других виджетов на панели управления.

Цифровые шкалы

Пакет полезен для визуализации температуры, влажности, скорости и других целочисленных или вещественных значений.

Виджеты администрирования

Шаблоны сложных виджетов, которые позволяют перечислять и создавать/обновлять/удалять устройства и активы.

Виджеты шлюза

Пакет виджетов шлюза полезен для управления расширениями, позволяющими отслеживать подключения устройств по различным протоколам, отслеживания происходящих событий на виртуальном устройстве и т.д.

Виджеты GPIO

Пакет виджетов GPIO полезен для визуализации и управления состоянием GPIO для физических устройств.

Виджеты ввода

Пакет входных виджетов полезен для изменения атрибутов объекта.

Виджеты карт

Пакет виджетов карты полезен для визуализации географического положения устройств и отслеживания маршрутов устройств как в режиме реального времени, так и в режиме истории.

Виджеты навигации

Пакет виджетов навигации полезен для определения домашней панели пользователя и создания многоуровневых панелей дашбордов, позволяющих создать иерархическую структуру мониторинга и управления сервисом Интернета вещей. Например, сервис может включать в себя несколько разных активов, в которых расположены собственные наборы датчиков. В основном дашборде может быть представлен перечень активов с их обобщенными показателями, а при необходимости рассмотреть более детальную информацию может присутствовать функционал разворачивания каждого из активов в собственный дашборд о всеми показателями.

Привязки устройств ThingsBoard

Привязки устройств предназначены для связывания виртуальных устройств с конкретным дашбордом для передачи данных в данный дашборд. Привязки позволяют работать как с одним устройством, так и с группой устройств. Взаимодействие не ограничивается только устройствами, в качестве источника может выступать любой объект облака, к примеру актив (здание/помещение/транспорт), к которому привязан датчик.

Пример создания дашборда для мониторинга

Рассмотрим пример создания дашборда для мониторинга параметров умного увлажнителя, рассматриваемого в прошлых практических работах.

В первую очередь необходимо создать дашборд. Сделать это можно на вкладке Dashboards. Создадим дашборд с наименованием Humidifier Visual. В итоге получим пустой дашборд с возможностью его редактирования (Рисунок 4).

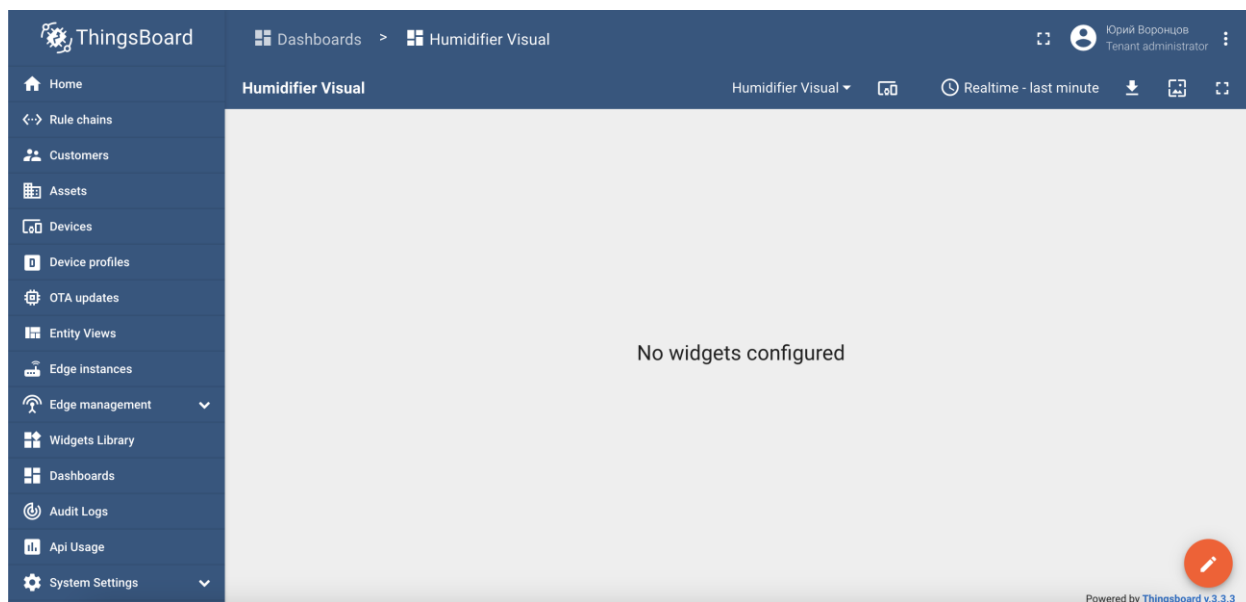


Рисунок 4 – Пустой дашборд

После этого необходимо добавить привязку устройства умного увлажнителя к данному дашборду. Для этого в дашборде присутствует отдельный управляющий элемент, открывающий меню работы с привязками, доступное в режиме редактирования (рисунок 5).

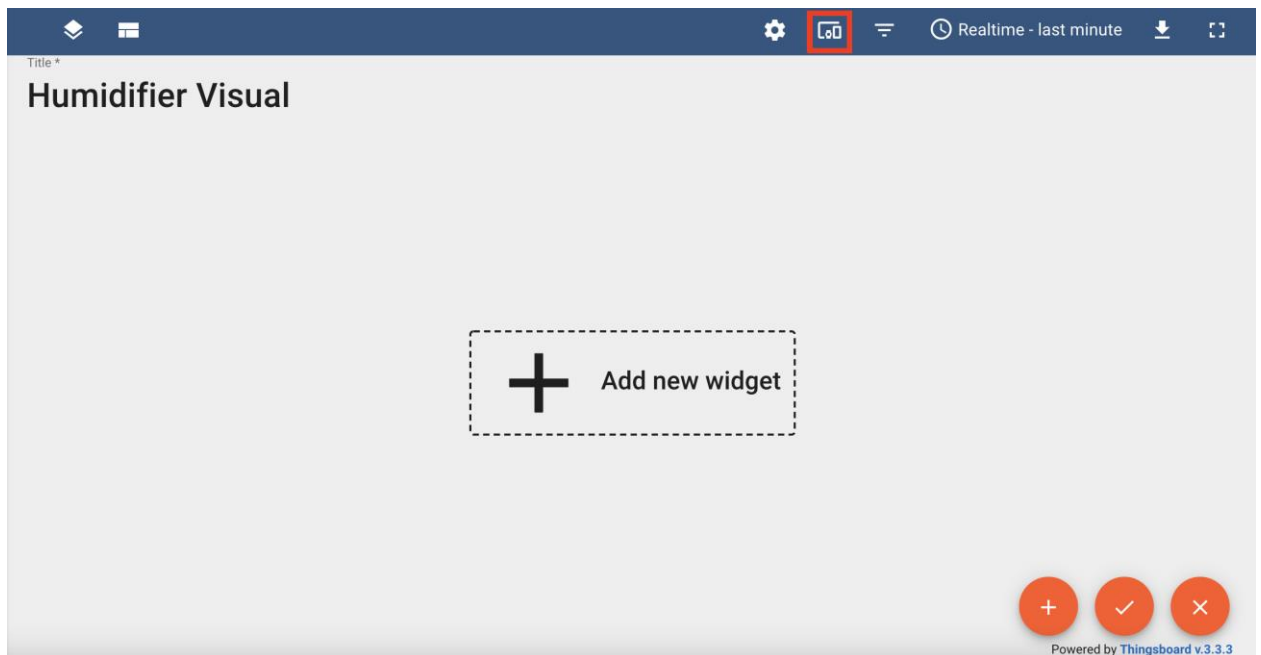


Рисунок 5 – Дашборд в режиме редактирования

Добавим привязку для одиночной сущности, в качестве сущности выберем устройство и интересующий нас увлажнитель (рисунок 6).

Add alias

Alias name *

Humidifier

Resolve as multiple entities

☐

Filter type *

Single entity

Type *

Device

Device *

Humidifier

Cancel

Add

Рисунок 6 – Привязка увлажнителя

После этого можно использовать выбранное устройство в виджетах дашборда.

Добавим несколько виджетов, позволяющих отслеживать состояние окружающей среды вокруг устройства, а также состояние самого увлажнителя. Для начала добавим график для отслеживания изменения уровня влажности с течением времени. Зададим в качестве привязки добавленный ранее увлажнитель, а также укажем в качестве параметра влажность. Для всех добавляемых виджетов будем задавать белый цвет фона. Остальные параметры оставим без изменений.

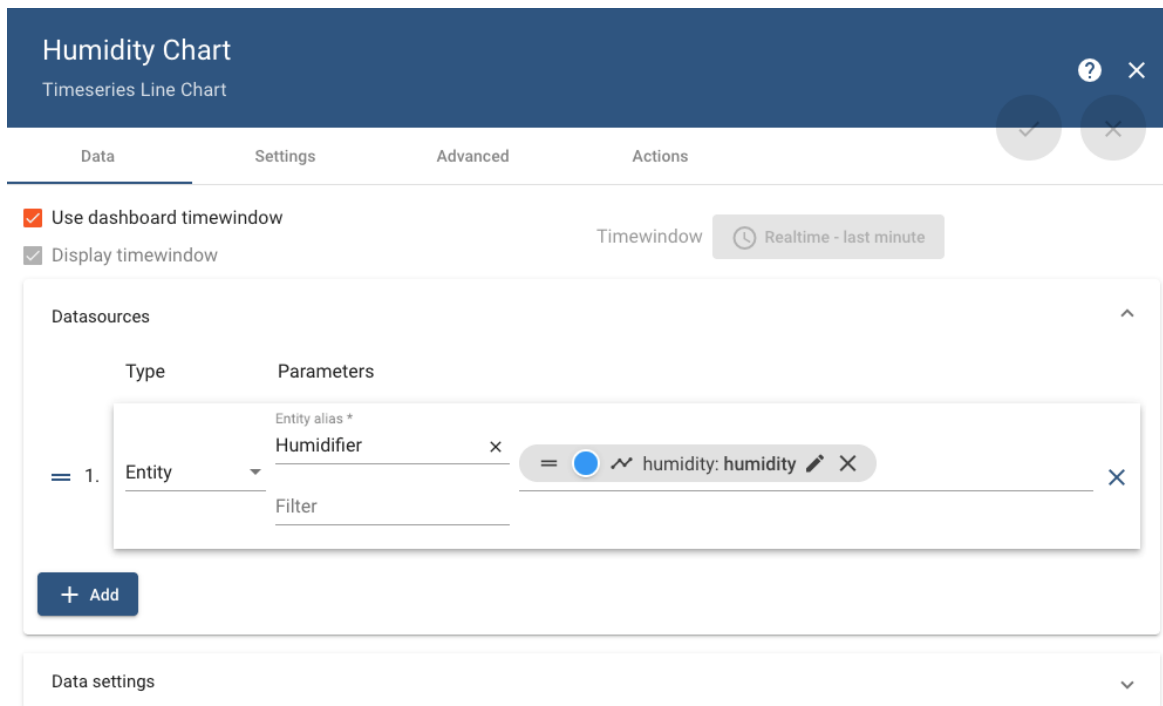


Рисунок 7 – Параметры графика

Добавим виджет с цифровым измерителем для отображения текущего значения влажности на устройстве. В целом параметры идентичны графику влажности и представлены на рисунке 8.

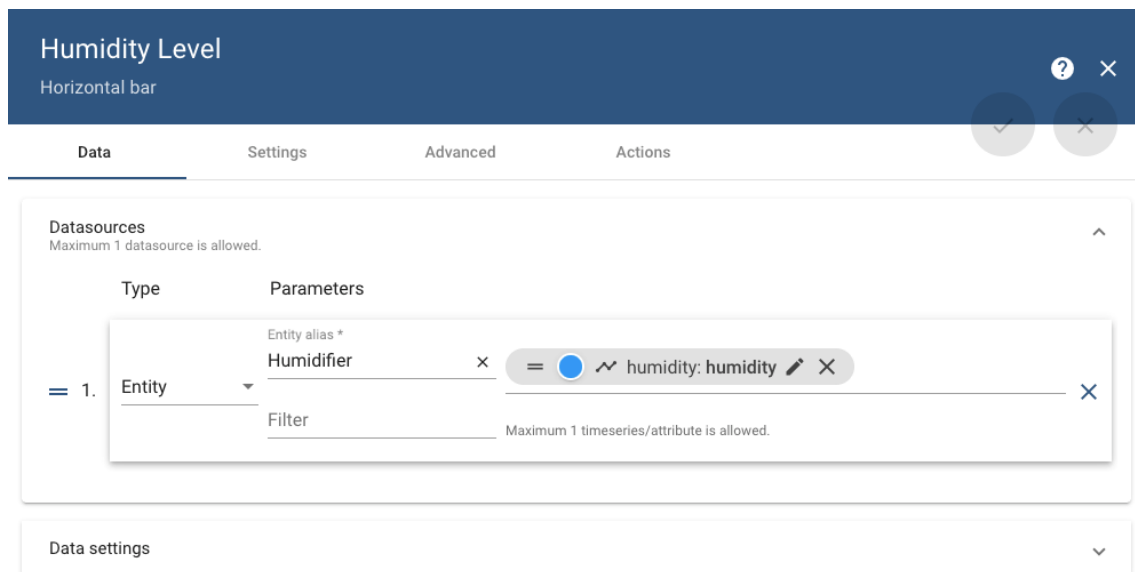


Рисунок 8 – Параметры цифрового измерителя

Также для контроля работы увлажнителя добавим виджет-индикатор, показывающий, активирован ли увлажнитель в текущий момент времени или нет. Данный индикатор может определять состояние при помощи RPC Call'ов, а также при помощи отслеживания состояния атрибутов. Для ответа на RPC существует возможность написать функцию парсинга этого самого ответа для проверки приходящих данных.

Привяжем состояние индикатора к атрибуту состояния устройства, изменяемого в цепочке правил, рассмотренной в предыдущих работах. Итоговая конфигурация индикатора представлена на рисунках 9-10.

Humidifier Status LED
Led indicator

Target device
Humidifier

Data settings

Рисунок 9 – Параметры виджета-индикатора

Humidifier Status LED
Led indicator

☐ Initial value
LED title
Humidifier Status

LED Color
rgb(76, 175, 80)

☐ Perform RPC device status check
RPC check device status method
checkStatus

Retrieve led status value using method
Subscribe for attribute

Device attribute/timeseries containing led status value *
status

Parse led status value function, f(data), returns boolean

```
1 return data ? true : false;
```

javascript Tidy ? Fullscreen

Рисунок 10 – Параметры виджета-индикатора для проверки состояния

Ещё одним виджетом, который поможет отслеживать состояние устройства, является виджет отображения тревог на устройстве. Он позволяет повысить уровень реагирования на возникающие сбои на устройстве, поскольку пропадает необходимость проверять тревоги на самом устройстве через его параметры.

Остановимся на базовом наборе параметров для отображения каждой тревоги: время создания, инициатор тревоги, тип тревоги, уровень тревоги, текущий статус тревоги. Итоговые параметры представлены на рисунке 11.

New Alarms table

Alarms table

?

×

Data

Settings

Advanced

Actions

☐ Use dashboard timewindow

Timewindow

Realtime - last day

☒ Display timewindow

Alarm status list

Any status

Alarm severity list

Any severity

Alarm type list

Any type

☐ Search propagated alarms

Alarm source

Entity alias *

Humidifier

×

Entity

Filter

Created time: createdTime

×

Originator: originator

×

Type: type

×

Severity: severity

×

Status: status

×

Data settings

▼

Рисунок 11 – Параметры виджета тревог

В результате получим удобный дашборд по мониторингу состояния увлажнителя.

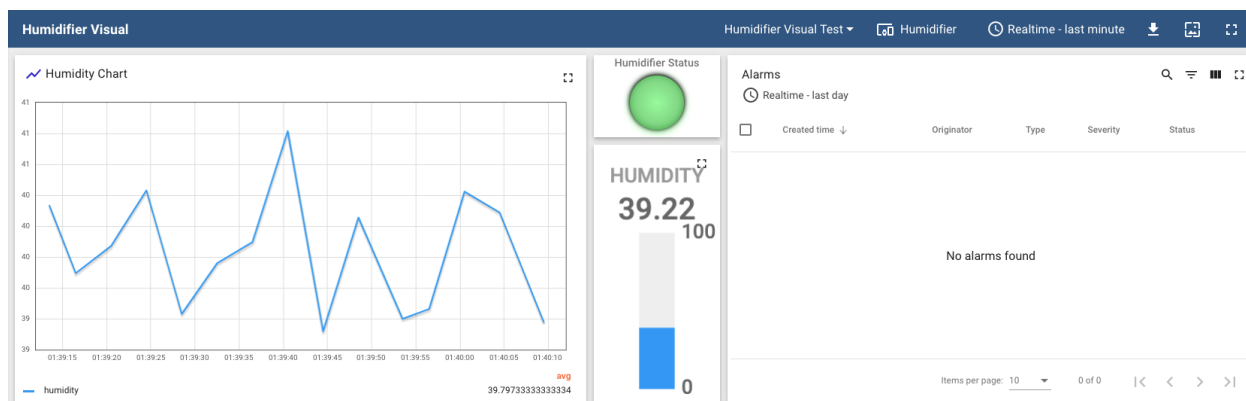


Рисунок 12 – Результирующий дашборд по отслеживанию увлажнителя

Рассмотрен лишь один вариант формирования дашборда, на самом дела существует гораздо более широкий набор виджетов для визуализации данных, позволяющих удобнее отслеживать состояние устройств.

Задание практической работы №13

На основании созданных в прошлых работах устройств сформировать дашборды для отслеживания состояния данных устройств. При реализации дашбордов необходимо использовать следующие виджеты:

- Графики и виджеты отображения последнего значения — для визуализации количественных параметров (CO2, температуры, уровня активности и т.д.);

- Виджеты-индикаторы для отображения состояния устройств с состоянием активации (включено/выключено), например, шаровой кран, вентилятор и т.д.;
- Виджеты-переключатели или виджеты-индикаторы для отображения состояния кнопок;
- Виджеты отображения атрибутов устройства (цвета RGB ленты, уровня громкости/частоты пищалки и т.д.);
- Виджет отображения тревог устройства.

Виджеты необходимо использовать в зависимости от используемых в устройствах параметров. Для формирования данных для визуализации можно использовать утилиты mosquito.

В отчет необходимо включить итоговый дашборд, а также параметры каждого из созданных виджетов.

Дополнительное задание практической работы №13

Часть 1. Подготовить техническое описание проекта, а именно – результирующий набор датчиков, перечень требуемых для производства элементов, описание используемого корпуса, рассчитать энергопотребление устройства, определить климатическое исполнение устройства.

Часть 2. Провести экономическое описание проекта: затраты на материалы и производство, внедрение, эксплуатацию.

Практическая работа №14 – Управление устройствами при помощи панелей управления облачных платформ

Управляющие виджеты ThingsBoard

Платформа ThingsBoard имеет несколько типов виджетов для управления устройствами. Первый набор содержится в пакете Control Widgets, он включает в себя набор переключателей, потенциометров и узлов отправки RPC запросов.

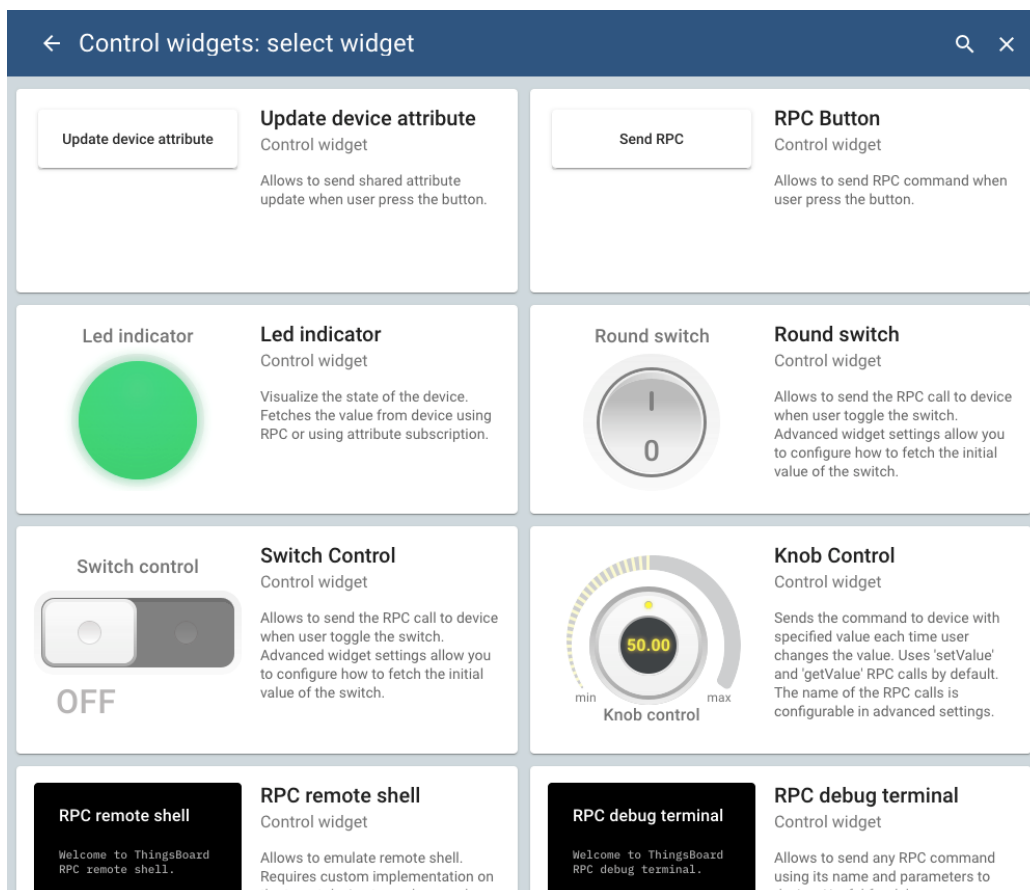


Рисунок 13 – Пакет виджетов Control Widgets

Некоторые из виджетов имеют возможность управляться через атрибуты виртуального устройства, вместо использования RPC запросов.

Второй тип виджетов – виджеты обновления атрибутов. Данный пакет носит название Input widgets. Управление происходит при помощи задания нового значения различным атрибутам устройства. В дальнейшем эти атрибуты могут использоваться для управления поведением виртуального устройства, например, использоваться при анализе поступающих данных в цепочках правил. Для каждого типа атрибутов присутствует свой отдельный виджет. Атрибуты подразделяются по типам: клиентские, серверные, разделяемые – а также по типу хранящегося в них значения: целочисленные, вещественные, логические, даты, местоположения, изображения. Помимо этого, существуют виджеты для выставления маркеров на картах для указания местоположения через визуальное отображение, а не через конкретные координаты. На рисунке 14 приведен частичный перечень возможных атрибутов.

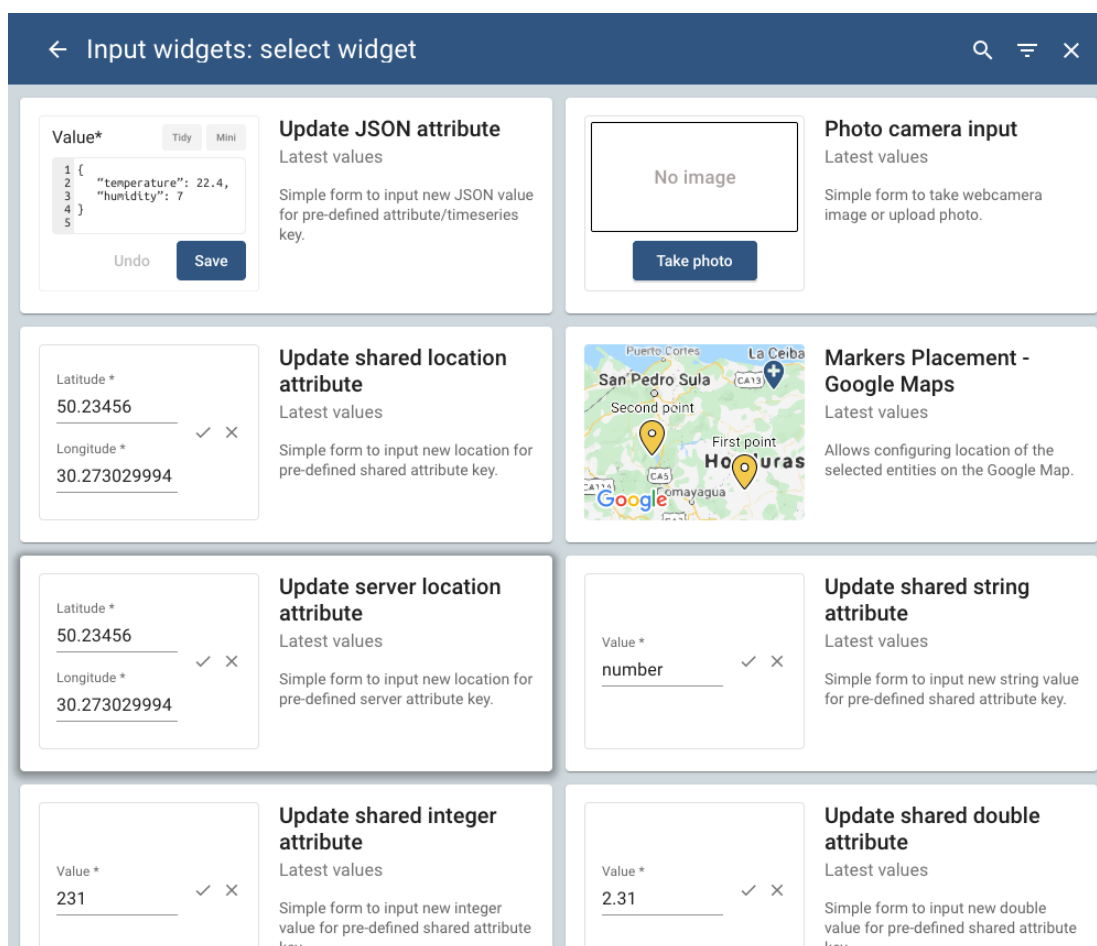


Рисунок 14 – Пакет input виджетов

Пример создания дашборда для управления устройством

Дополним дашбор из практической работы №13 управляющими элементами. К примеру, сделаем возможной настройку граничного значения автоматического срабатывания увлажнителя. Также добавим возможность ручного управления увлажнителем без использования полной логики цепочки правил.

Добавим в атрибуты виртуального устройства параметр `humidityLimit` для возможности контроля граничного значения.

Server attributes

Entity attributes scope
Server attributes

+ ↺ 🔍



<input type="checkbox"/>	Last update time	Key ↑	Value	
<input type="checkbox"/>	2021-12-02 23:09:25	active	false	
<input type="checkbox"/>	2021-12-02 23:15:32	humidityLimit	30	

Рисунок 15 – Добавление атрибута `humidityLimit`

Контролировать данный атрибут можно при помощи виджета обновления атрибутов. Добавим данный виджет (рисунок 16) для серверного атрибута `humidityLimit`.

Value *

2.31

✓ ✕

Update server double attribute

Latest values

Simple form to input new double value for pre-defined server-side attribute key.

Рисунок 16 – Виджет обновления серверного атрибута

Для отображения атрибута настроим виджет как показано на рисунках 17-18.

Update humidity limit attribute

Update shared double attribute

?

✕

Data

Settings

Advanced

Actions

Datasources

Maximum 1 datasource is allowed.

Type

Parameters

= 1.

Entity

Entity alias *

Humidifier

✕

Filter

=

ⓘ

humidityLimit: humidityLimit

✎

✕

Maximum 1 timeseries/attribute is allowed.

Data settings

Рисунок 17 – Параметры виджета обновления атрибута

Update humidity limit attribute

Update shared double attribute

?

✕

Data

Settings

Advanced

Actions

Widget title

Humidity level limit

☒

Show result message

☒

Show label

☒

Required

Label

'Required' error message

Max value

100

Min value

1

Рисунок 18 – Дополнительные параметры виджета обновления атрибута

Помимо этого, необходимо доработать цепочку правил, которую используем устройство увлажнителя, изменив скрипт узла проверки значения телеметрии, а также добавив перед ним узел получения атрибутов для получения атрибута `humidityLimit`. Во-первых, добавим узел получения атрибута (рисунок 19)

Рисунок 19 – Параметры узла получения атрибута `humidityLimit`

Перепишем скрипт для обработки приходящей телеметрии.

Листинг 1 – Скрипт обработки телеметрии

```
function getNewHumidifierStatus(humidity) {
    return humidity < metadata.shared_humidityLimit;
}

let newMsg = {};
let newMsgType = {};

newMsg = {
    "method": "setValveState",
    "params": {
        "state": getNewHumidifierStatus(msg.humidity)
    }
};

newMsgType = "POST_ATTRIBUTES_REQUEST";

return {msg: newMsg, metadata: metadata, msgType: newMsgType};
```

В итоге участок цепочки выглядит следующим образом.

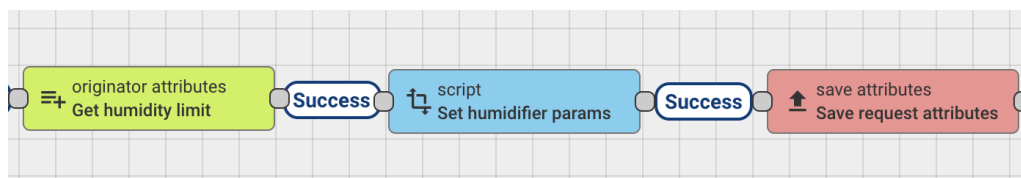


Рисунок 20 - дополненный участок цепочки

Также добавим переключатель, позволяющий переводить устройство в режим ручного управления. В качестве переключателя будем использовать виджет Switch Control (рисунок 21), которому зададим управление вариант управления через отправку RPC запросов. В качестве метода получения состояния будем использовать наименование `getConstantHumidityState` и в качестве метода установки состояния `setConstantHumidityState`.

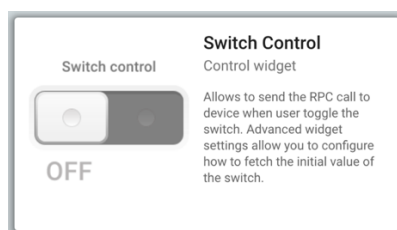


Рисунок 21 – Виджет переключатель

Параметры представлены на рисунках 22-24.

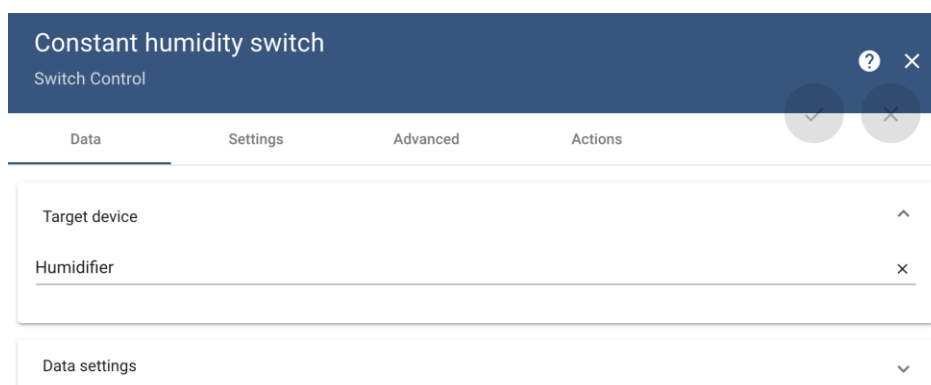


Рисунок 22 – Параметры указания устройства для переключателя

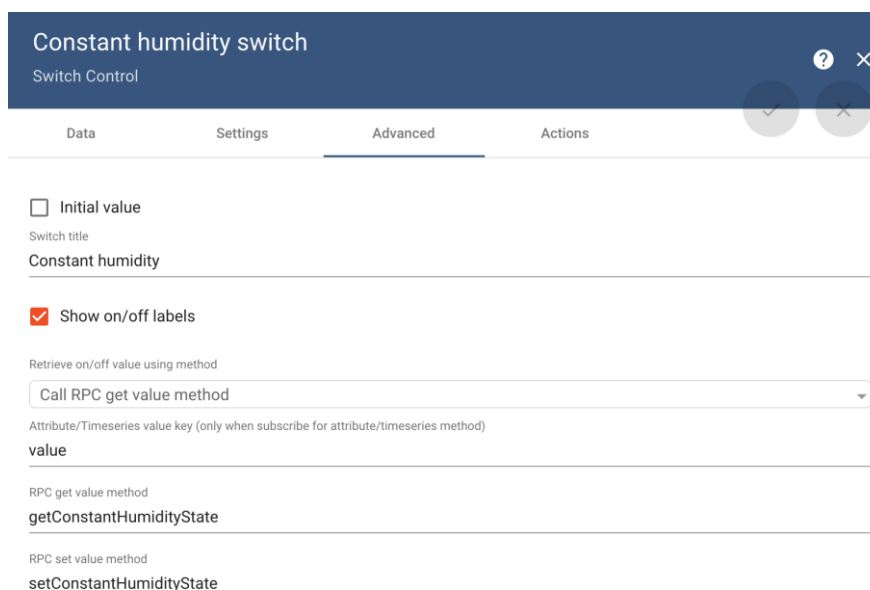


Рисунок 23 – Расширенные параметры переключателя

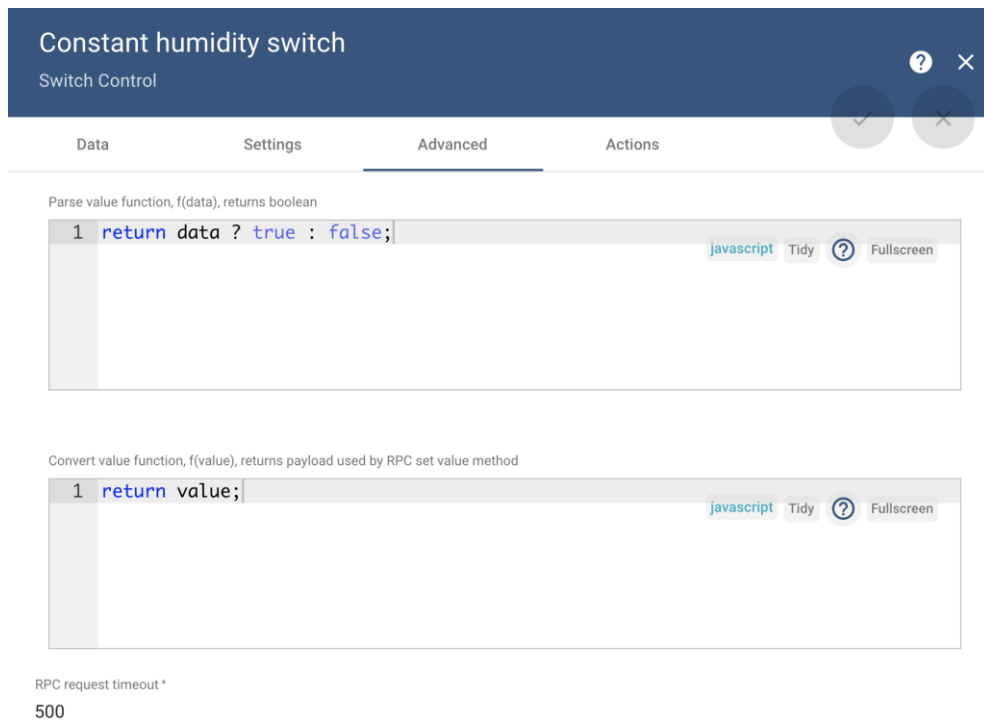


Рисунок 24 – Расширенные параметры переключателя (продолжение)

Чтобы управлять уровнем влажности добавим поворотный переключатель (Knob Control) для установки текущего уровня влажности (рисунок 25).

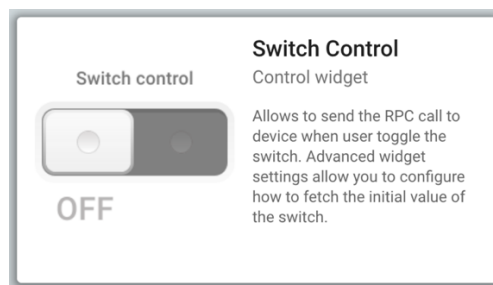


Рисунок 25 – Виджет Knob Control

Также настроим данный виджет на получение данных и отправку команд через RPC запросы. В качестве метода получения текущего значения параметра от устройства будем использовать метод `getConstantHumidityLevel`, а в качестве метода установки параметра будем использовать `setConstantHumidityLevel`. Итоговые параметры виджета представлены на рисунках 26-27.

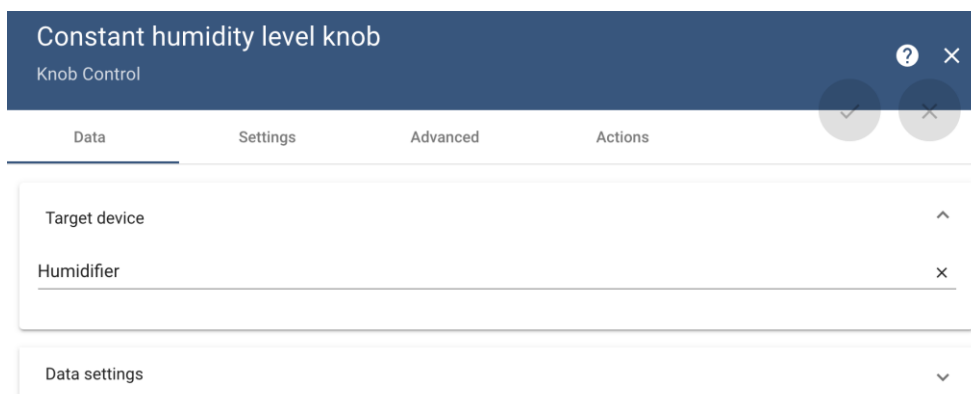


Рисунок 26 – Параметры виджета Knob Control

Constant humidity level knob
Knob Control
?
X

Data
Settings
Advanced
Actions

Minimum value *
0

Maximum value *
100

Initial value
50

Knob title
Constant humidity level

Get value method *
getConstantHumidityLevel

Set value method *
setConstantHumidityLevel

RPC request timeout *
500

☐ RPC request persistent

Рисунок 27 – Расширенные параметры виджета Knob Control

Помимо этого, для того, чтобы при активации ручного управления платформа не проводила проверку приходящей телеметрии через цепочку правил, добавим в неё проверку на активацию ручного режима управления. Информацию об активации ручного режима будем получать от устройства. То есть, при отправке RPC запроса на выставление ручного режима от виджета Switch Control будем получать от реального устройства состояние ручного управления – manualControl, в топик для сохранения атрибутов (v1/devices/me/attributes). Дополним цепочку для анализа данного параметра.

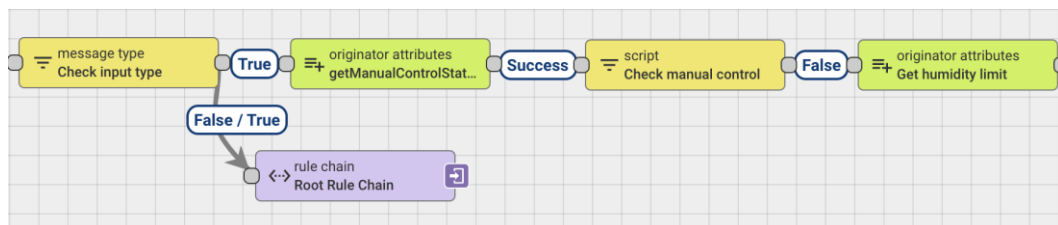


Рисунок 28 – Дополненная цепочка правил

Поскольку состояние ручного управления сохраняется в атрибуты, будем запрашивать данный параметр из атрибутов и анализировать его в узле проверки. На рисунках 29-30 приведены параметры добавленных узлов.

getManualControlStatus

?
X

Enrichment - originator attributes

Details
Events
Help

Name *

getManualControlStatus

☒ Debug mode

☒ Tell Failure

If at least one selected key doesn't exist the outbound message will report "Failure".

Client attributes

manualControl X Client attributes

Hint: use ``${metadataKey}`` for value from metadata, ``${messageKey}`` for value from message body

Shared attributes

Shared attributes

Hint: use ``${metadataKey}`` for value from metadata, ``${messageKey}`` for value from message body

Server attributes

Server attributes

Hint: use ``${metadataKey}`` for value from metadata, ``${messageKey}`` for value from message body

Latest timeseries

Latest timeseries

Hint: use ``${metadataKey}`` for value from metadata, ``${messageKey}`` for value from message body

☐ Fetch Latest telemetry with Timestamp

If selected, latest telemetry values will be added to the outbound message metadata with timestamp, e.g: "temp": {"ts":1574329385897, "value":42}

Рисунок 29 – Параметры узла получения статуса ручного управления

Check manual control

?
X

Filter - script

Details
Events
Help

Name *

Check manual control

☒ Debug mode

Filter

function Filter(msg, metadata, msgType) {

Tidy ?

1 return metadata.cs_manualControl === "true";
2

}

Рисунок 30 – Параметры узла проверки статуса ручного управления

В результате получим дашборд следующего вида.

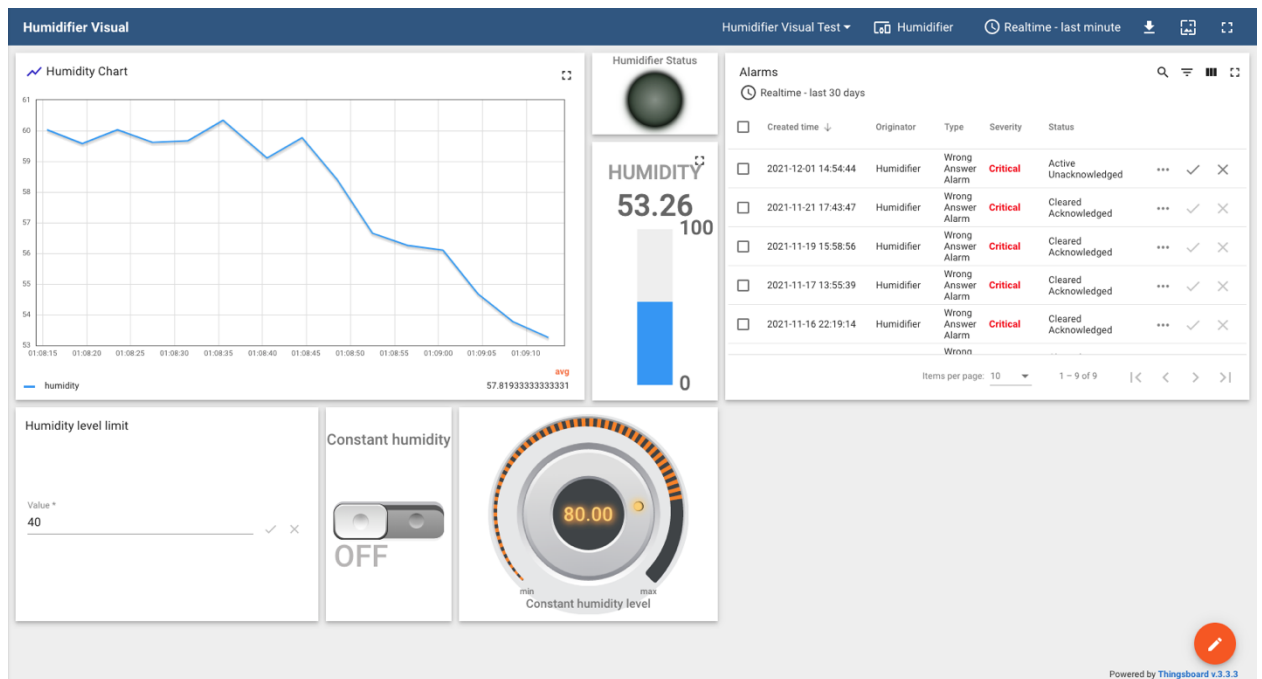


Рисунок 31 – Результирующий дашборд

Поскольку управляющие виджеты работают через RPC запросы, можно отследить данные запросы через подписку на топик RPC запросов `v1/devices/me/rpc/request/+`. Проверим поступающие запросы при помощи `mosquitto_sub`. При обновлении страницы дашборда можно увидеть запросы на получение параметров (рисунок 32).

```
% mosquitto_sub -v -h "demo.thingsboard.io" -t "v1/devices/me/rpc/request/+"
-u "HeuI2DOMWdRxm5xhAIMy" -v
v1/devices/me/rpc/request/3754 {"method":"getConstantHumidityState"}
v1/devices/me/rpc/request/3755 {"method":"getConstantHumidityLevel"}
```

Рисунок 32 – Запросы на получение параметров

Также изменим состояние переключателя и установим новое значение для потенциометра. Можно увидеть, что данные действия так же вызывают отправку сообщения с RPC запросом (рисунок 33).

```
v1/devices/me/rpc/request/3756 {"method":"setConstantHumidityLevel","params":60.74}
v1/devices/me/rpc/request/3757 {"method":"setConstantHumidityState","params":true}
```

Рисунок 33 – Запросы на установку параметров

Обработка входящих RPC запросов на конечном устройстве позволит выполнить соответствующее действие. Данный механизм позволяет реализовать удаленное управление физическими устройствами.

Задание практической работы №14

Реализуйте управляющие виджеты в созданном ранее дашборде:

- Виджеты установки порогового значения активации устройства (вентилятора, шарового крана и т.д.);
- Виджеты ручного включения/выключения устройства (вентилятора, шарового крана и т.д.) из облачной платформы.

В отчет необходимо включить итоговый дашборд, а также параметры каждого из созданных виджетов.

Дополнительное задание практической работы №14

Подготовить презентацию для защиты проекта, отражающую результаты разработки на протяжении всего семестра. В презентацию необходимо включить:

- техническое описание проекта со всеми используемыми модулями;
- схему подключения всех компонентов устройства;
- краткое описание использованных технологий;
- скриншоты реализованного интерфейса;
- экономическое обоснование;
- а также дополнительные слайды на своё усмотрение.

Литература для изучения:

1. Обзор возможностей дашбордов платформы ThingsBoard:
<https://thingsboard.io/docs/user-guide/dashboards/>;
2. Обзор привязок для дашбордов платформы ThingsBoard:
<https://thingsboard.io/docs/user-guide/ui/aliases/>;
3. Обзор действий виджетов: <https://thingsboard.io/docs/user-guide/ui/widget-actions/>;
4. Обзор библиотеки виджетов: <https://thingsboard.io/docs/user-guide/ui/widget-library/>