



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практическим работам №9-12

по дисциплине «Технологические основы Интернета вещей»

Выполнили:

Студенты группы ИКБО-11-22

Голованев Никита Алексеевич
Гришин Андрей Валерьевич
Андрусенко Лада Дмитриевна
Егоров Илья Эдуардович

Проверил:

Кандидат технических наук, доцент

Жматов Дмитрий Владимирович

2024 г.

Оглавление

Практическая работа №9	3
Практическая работа №10.....	7
Практическая работа №11.....	10
Практическая работа №12.....	13
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	18

Практическая работа №9

Модель объекта контроля - это формализованное представление устройства, подключаемого к платформе. В модели описываются параметры, которые отправляет устройство, и команды управления, которые оно может обработать. Единоразово созданная модель может быть использована как для одного, так и для нескольких объектов, если они обладают одинаковым набором считываемых данных и выполняемых функций.

Если у вас есть три одинаковых датчика, передающих температуру и влажность, то нужно создать только **одну** модель, но **три объекта** (для каждого устройства)

Функции модели:

- поднимает уровень абстракции с уровня протоколов и аппаратной реализации до понятных человеку функций устройства, с которыми удобно работать без необходимости разбираться в протоколе. Преобразованные таким образом данные формируются в JSON, вид которого привычен для большинства разработчиков;



- автоматически рендерит интерфейс для визуализации данных от устройства;

- используется для формирования сценариев автоматизации и обработчиков данных.

Карточка модели

1. Имя модели
2. Описание модели
3. Шаблон (используемый протокол передачи данных)
4. Количество объектов с этой моделью. Нажмите на него, чтобы применить фильтр к списку объектов по данной модели

[Вход](#) [Регистрация](#)

— или используйте свой e-mail (или логин) и пароль

Язык

E-mail / логин

Пароль [\(забыли?\)](#)

[Войти](#)

Рисунок 1 – Регистрация на Rightech

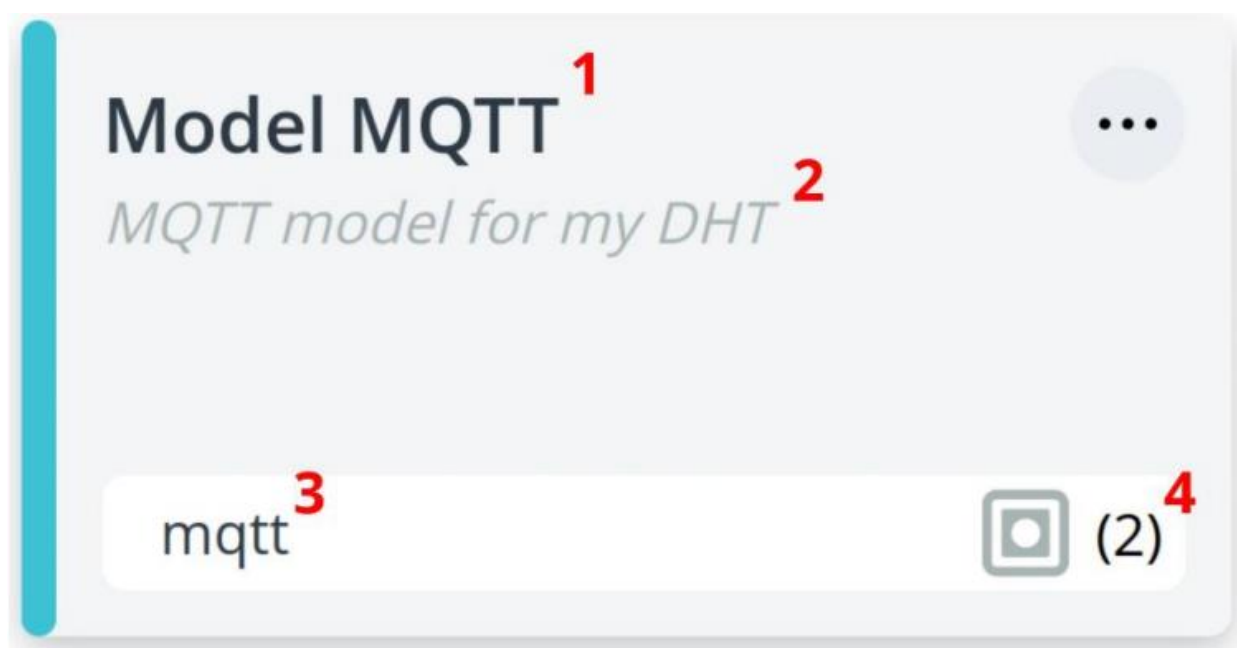


Рисунок 2 – Карточка модели

Параметры узлов модели

Сначала в каждом узле указывается его **тип**. В зависимости от выбранного типа формируются соответствующие поля формы для

последующего заполнения.

Типы:

– **Подсистема** - тип узла, который служит для организации структуры модели, позволяя объединять параметры в группы. Подсистема подразумевает, что данный узел содержит в себе несколько элементов в виде ответвлений древовидной структуры.

– **Аргумент** – это параметр, который передает устройство на платформу (например, текущее измерение сенсора). Аргументы могут быть числовыми, логическими, строковыми или представлять собой объект или массив. При этом для числовых аргументов можно указать единицу измерения, изображение и задать уровни.

– **Конфигурация** – это параметр, значение которого затем можно задать в интерфейсе объекта. Его можно использовать в работе автоматов и обработчиков. По своей сути, узлы с данным типом — это константы, которые необходимы для хранения дополнительной информации об объекте. Это может быть, к примеру, максимально допустимая температура, объем топливного бака, длительность работы и т.п.

– **Действие** – это операция, которая нужна для отправки команды на устройство или запуска автомата.

– **Событие** – это наступление определенных условий, которое либо было зафиксировано объектом, либо произошло во внешних по отношению к объекту контроля приложениях. События используются в автоматах при построении переходов между состояниями. Именно по произошедшим с объектом событиям эти переходы и происходят.

Задание

№ Варианта	Датчики
5	1. Датчик качества воздуха 2. Датчик освещенности 3. Датчик напряжения

```
C:\Program Files\mosquitto>mosquitto_pub -h dev.rightech.io -p 1883 -i mqtt-ilyegorow-w6sucv -t device/voltage -m "234" -u 123 -P 123

C:\Program Files\mosquitto>mosquitto_pub -h dev.rightech.io -p 1883 -i mqtt-ilyegorow-w6sucv -t 'device/qw' -m "234" -u 123 -P 123

C:\Program Files\mosquitto>mosquitto_pub -h dev.rightech.io -p 1883 -i mqtt-ilyegorow-w6sucv -t 'device/il' -m "234" -u 123 -P 123
```

Рисунок 3 – MQTT запросы

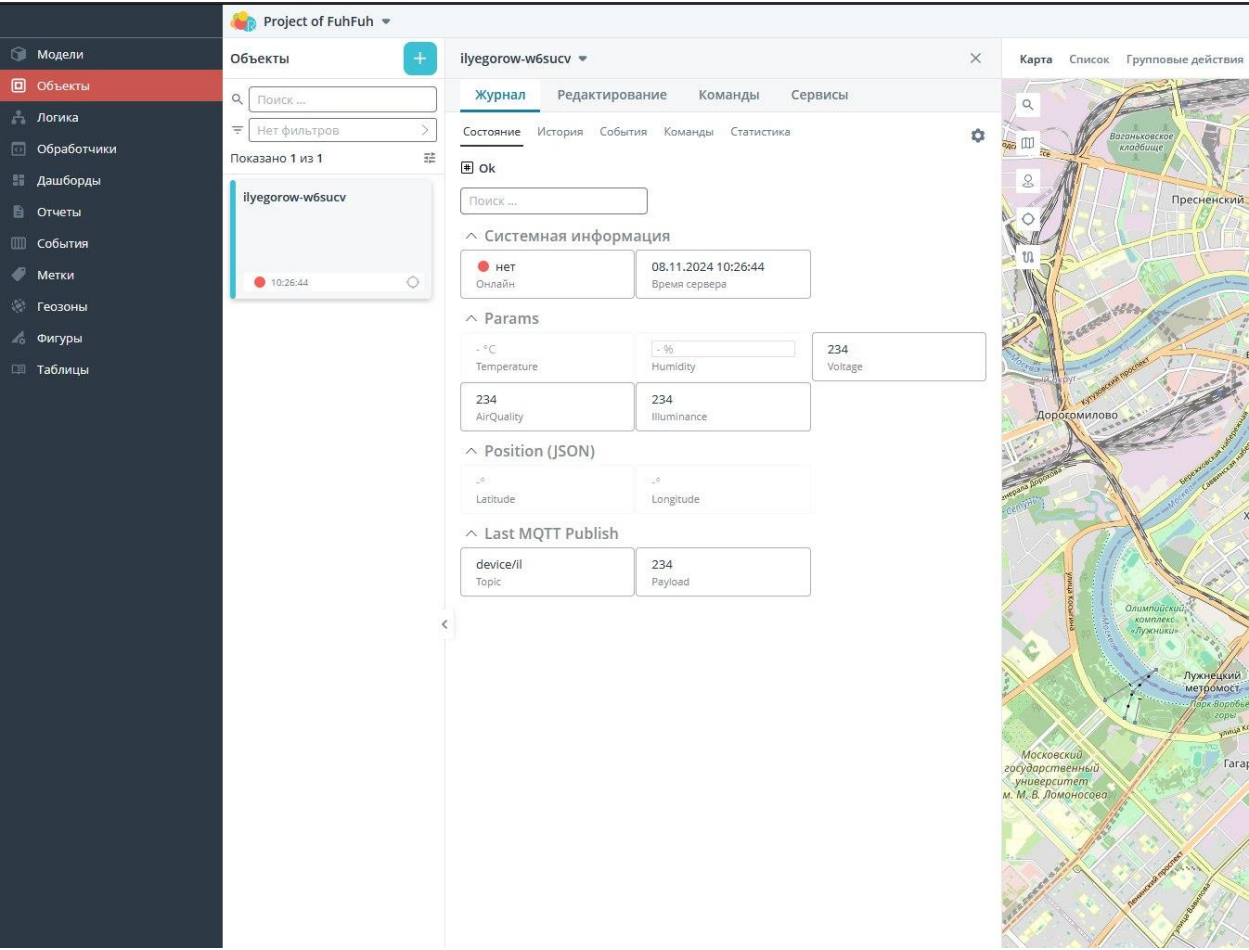


Рисунок 4 – Список устройств

Практическая работа №10

Логика

Цель раздела «Логика» заключается в повышении уровня абстракции: от работы с моделями и элементами устройства к созданию сценариев автоматического взаимодействия с ним. Каждый сценарий автоматизации — это алгоритм, который определяет логику поведения объекта. С помощью такого сценария можно отслеживать изменения в устройстве и, в зависимости от этих изменений, автоматически выполнять соответствующие действия без участия пользователя. Сценарий автоматизации основан на конечном автомате, состоящем из состояний и переходов между ними.

Конечный автомат (или детерминированный конечный автомат, ДКА) — это математическая модель, используемая для описания поведения систем с конечным числом состояний. Конечный автомат состоит из множества состояний, событий и правил, которые определяют, как система переходит из одного состояния в другое на основе событий или входных данных.

Основные элементы конечного автомата

1. Конечное множество состояний — набор возможных состояний, в которых может находиться система. Каждое состояние описывает определённое положение или характеристику системы. Например, для системы безопасности дома состояния могут быть "Система активна" и "Система отключена".
2. Начальное состояние — это состояние, в котором система находится в самом начале. Для каждого конечного автомата оно одно.
3. Конечное множество событий или входных данных — события, которые приводят к изменению состояния системы. Эти события могут быть внешними (например, нажатие кнопки) или внутренними (например, изменение температуры).

4. Правила переходов или функция перехода — набор правил, которые определяют, как система переходит из одного состояния в другое в ответ на конкретное событие или входные данные. Переход всегда однозначно определён: при получении одного и того же события система всегда переходит в одно и то же состояние.
5. Конечное множество конечных состояний — определенные состояния, в которых система завершает выполнение своей логики. Это может быть состояние успешного завершения работы или сбоя.

Задание

№ Варианта	Датчики
5	Включение и выключение вентилятора по датчику движения Включение и выключения любого индикатора комбинированного датчика по кнопке

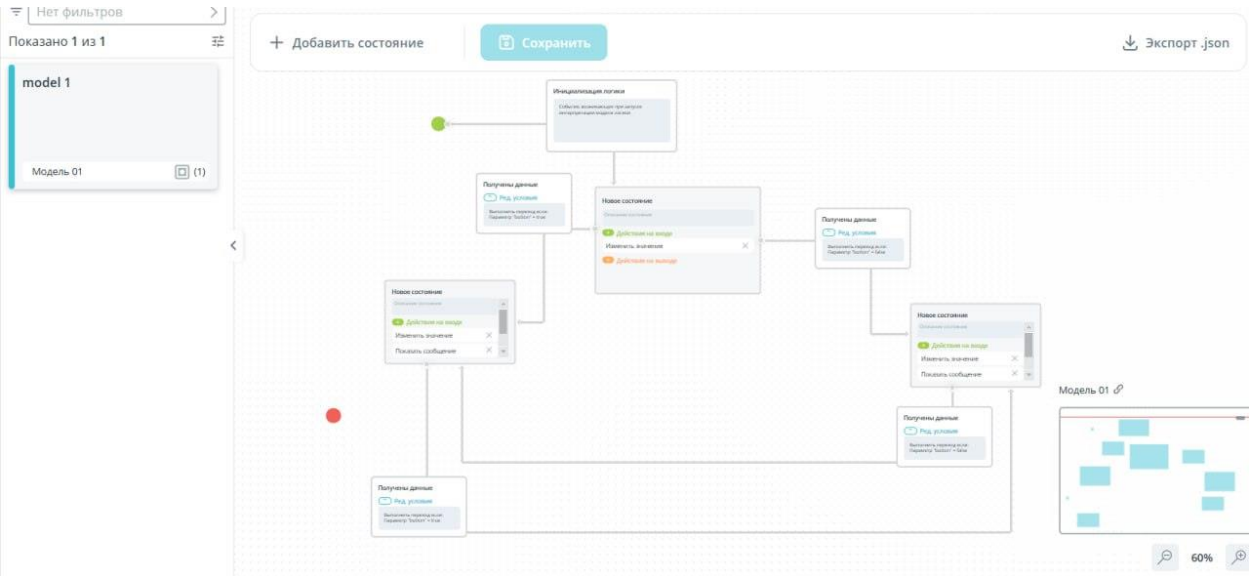


Рисунок 5 – Реализация автомата для второго задания



Рисунок 6 – Отправка запроса на изменение значения объекту



Рисунок 7 – Отправка запроса на изменение значения объекту

Практическая работа №11

Назначение реакций

Как уже было сказано ранее, платформы Интернета вещей используются для обработки данных и реализации логики работы IoT сервиса без привязки к конкретным физическим устройствам. Рассмотренный в предыдущей работе пример позволяет строить самую базовую логику по управлению устройствами, однако, никакой реакции от облачной платформы на ответы от физического устройства не предусмотрено, что не позволяет полноценно отслеживать правильность работы устройства.

IoT платформы имеют возможность отправлять оповещения (тревоги) при возникновении ошибок в ходе выполнения скриптов обработки приходящих данных. Данный механизм позволяет решить проблему отслеживания состояния физического устройства, которое передается в качестве ответа на отправляемый запрос о смене состояния. Помимо этого, можно использовать данный механизм для оповещений в случае поступления на устройство данных, выходящих за рамки допустимых значений.

Облачная платформа Rightech.io также поддерживает подобный механизм тревог-оповещений, следовательно, рассмотрим его на примере данной платформы.

Показать сообщение

Отправка сообщений позволяет следить за текущими изменениями, происходящими при исполнении логики. Если добавить данную команду на вход в какое-либо состояние, то при переходе объекта в данное состояние система отправит необходимое уведомление. Оно проинформирует о том, что совершилось определенное событие, вследствие которого объект теперь находится в данном состоянии. В итоге при регулярной отправке уведомлений вы всегда будете знать о ходе статусе процесса управления.

Формирование сообщений

Для формирования сообщения добавьте в состояние команду Показать сообщение. Справа появится боковая панель, на которой выберите степень важности сообщения:

- Информационное;
- Важное;
- Критическое.

Задание

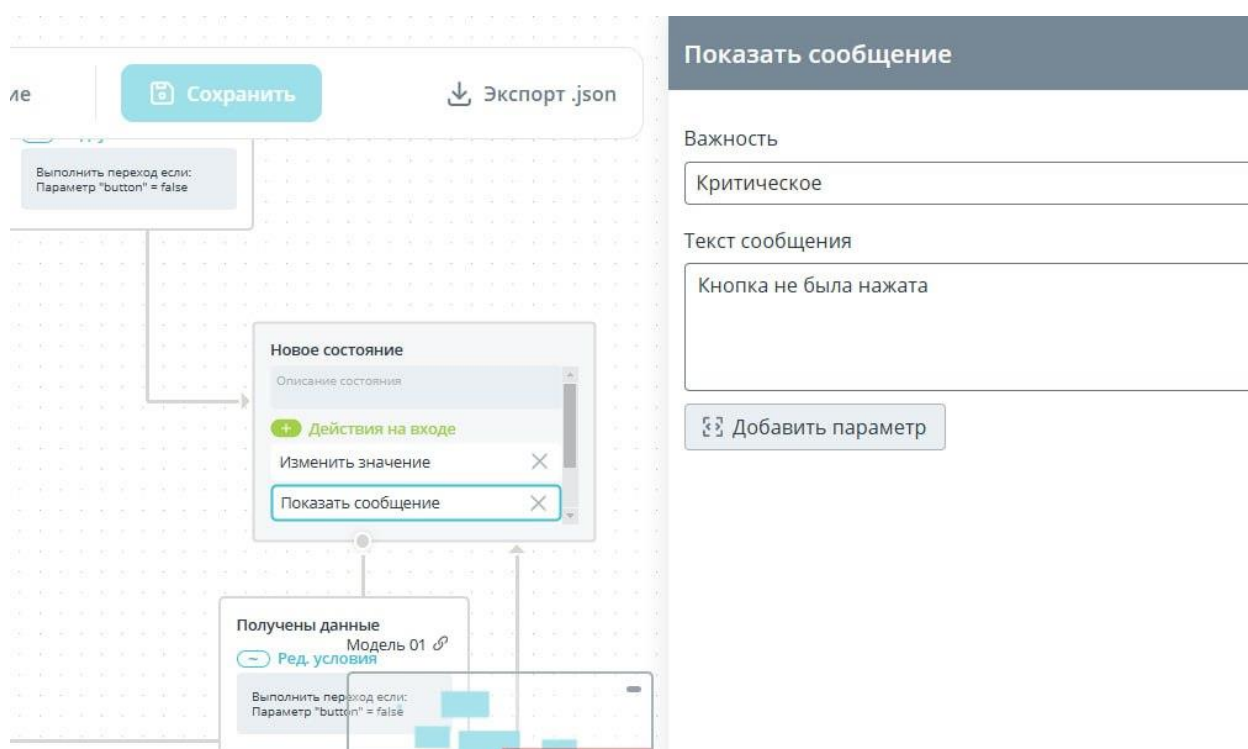


Рисунок 8 – Сообщение тревоги

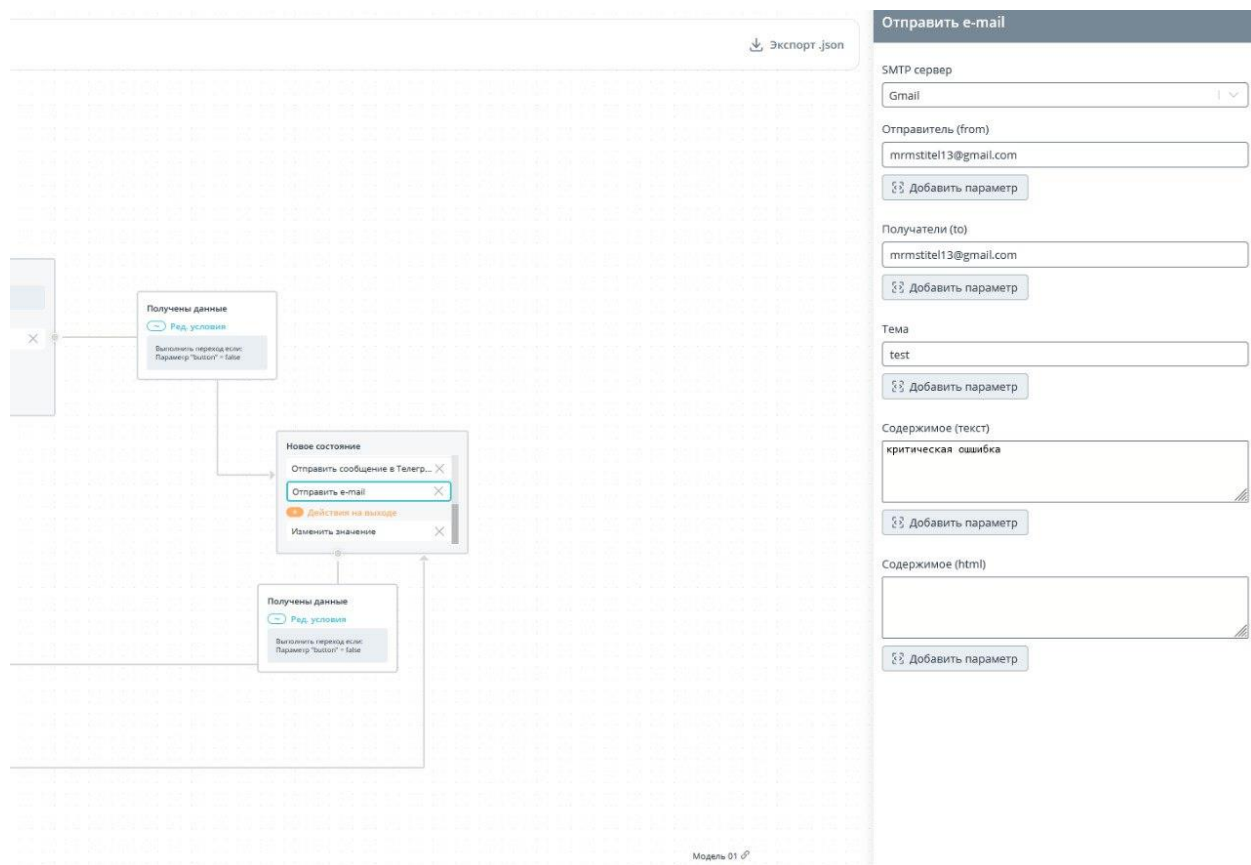


Рисунок 9 – Отправка на почту сообщения

Практическая работа №12

Протокол SMTP

SMTP — это простой протокол передачи почты. С английского языка переводится, как Simple Mail Transfer Protocol. SMTP сервер отвечает за отправку почтовых рассылок. Его задача, как правило, состоит из двух основных функций:

- проверка правильности настроек и выдача разрешения компьютеру, который пытается отправить электронное сообщение;
- отправка исходящего сообщения на указанный адрес и подтверждение успешной отправки сообщения. Если доставка невозможна, сервер возвращает отправителю ответ с ошибкой отправки.

Отправляя Email сообщения, SMTP-сервер отправителя устанавливает связь с тем сервером, который будет получать это сообщение. Такое "общение" происходит путем отправки и получения команд, формируя SMTP-сессию с неограниченным количеством SMTP-операций. Обязательными командами для каждой операции являются три:

- определение обратного адреса (MAILFROM);
- определение получателя e-mail сообщения (RCPT TO);
- отправка текста сообщения (DATA).

Определение адреса отправителя, получателя и наличие содержимого письма – это обязательные условия, без которых письмо не будет отправлено.

Задание

Время события	Событие	Сущность	Источник	Детализация события
28.11.2024 15:11:17	push-message	ilyegorow-wb5ucv	Система	Важность: Информационное; Текст: false
28.11.2024 15:11:17	Офлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:11:17	Онлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:11:10	Офлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:11:10	Онлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:08:12	push-message	ilyegorow-wb5ucv	Система	Важность: Критическое; Текст: Кнопка не была нажата
28.11.2024 15:08:12	Офлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:08:12	Онлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:08:01	Офлайн	ilyegorow-wb5ucv	Система	
28.11.2024 15:08:01	push-message	ilyegorow-wb5ucv	Система	Важность: Информационное; Текст: false
28.11.2024 15:08:00	Онлайн	ilyegorow-wb5ucv	Система	

Рисунок 10 – Список событий

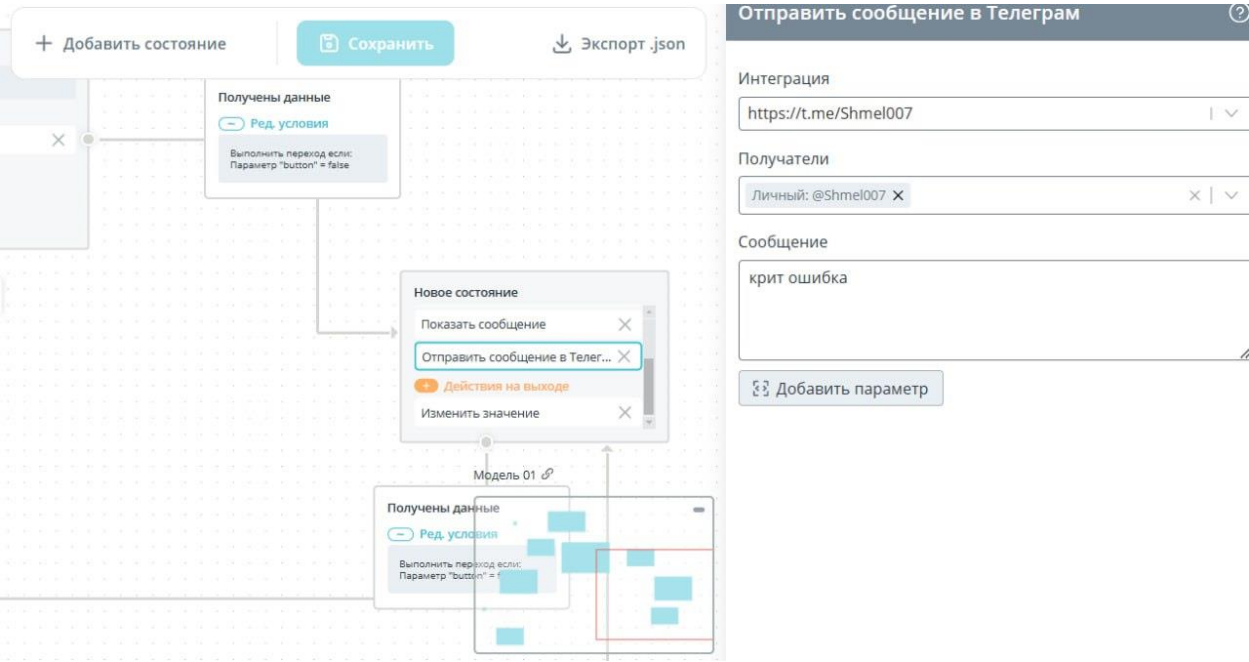


Рисунок 11 – Демонстрация отправки сообщения

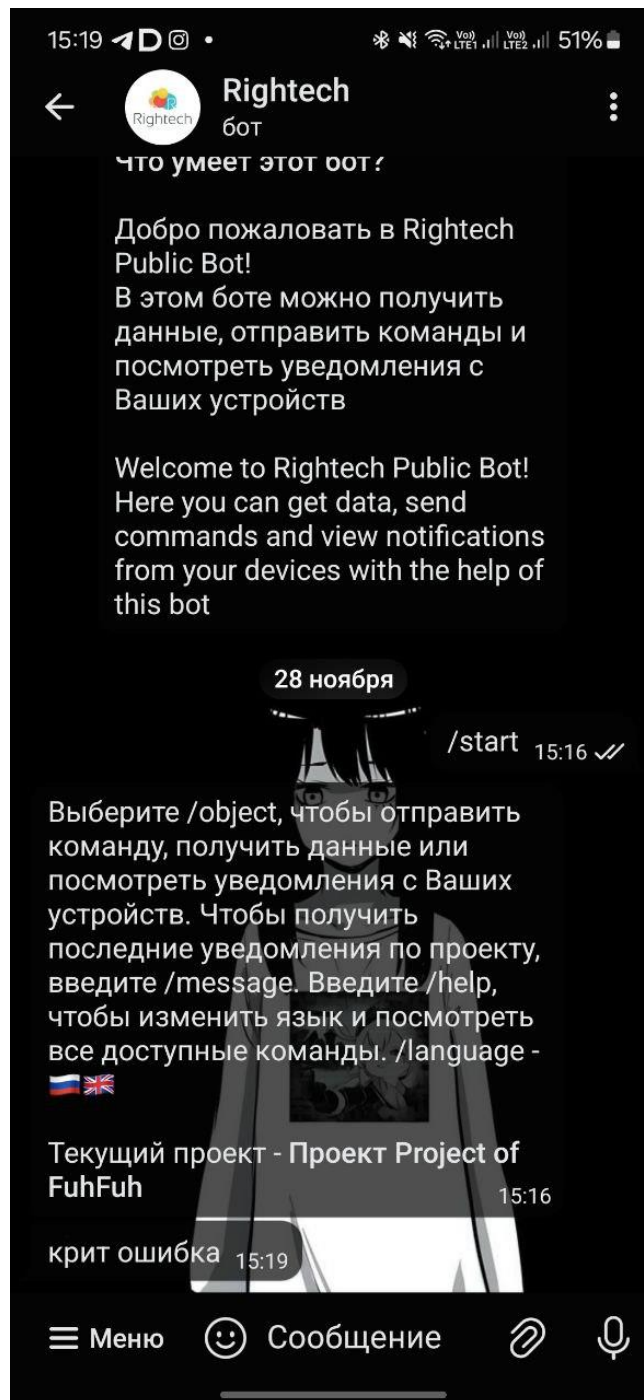


Рисунок 12 – Получение сообщения в Telegram

Project of FuhFuh

Проект

Команда

Роли

API Токены

Интеграции

https://t.me/Shmel007

интеграция

Активно

Pers SMTP

Gmail

Создать

SMTP - это широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP. Добавьте SMTP-сервер, чтобы получить возможность отправлять email из Автоматов.

Название канала сообщений *

Gmail

Описание

Опишите интеграцию

Хост *

smtp.gmail.com

Порт *

465

Логин *

mrmsttel13@gmail.com

Пароль *

.....

Отправитель

MIREA

Настройки безопасности

SSL

Сохранить

Удалить

Тестирование

Обратите внимание, что тестирование выполняется для сохраненной версии SMTP.

Получатель *

mrmsttel13@gmail.com

Тема письма *

Rightech test mail

Тело письма (текст) *

Test

Отправить тестовое письмо

Тестовое письмо успешно отправлено

Рисунок 13 – Интеграция почты

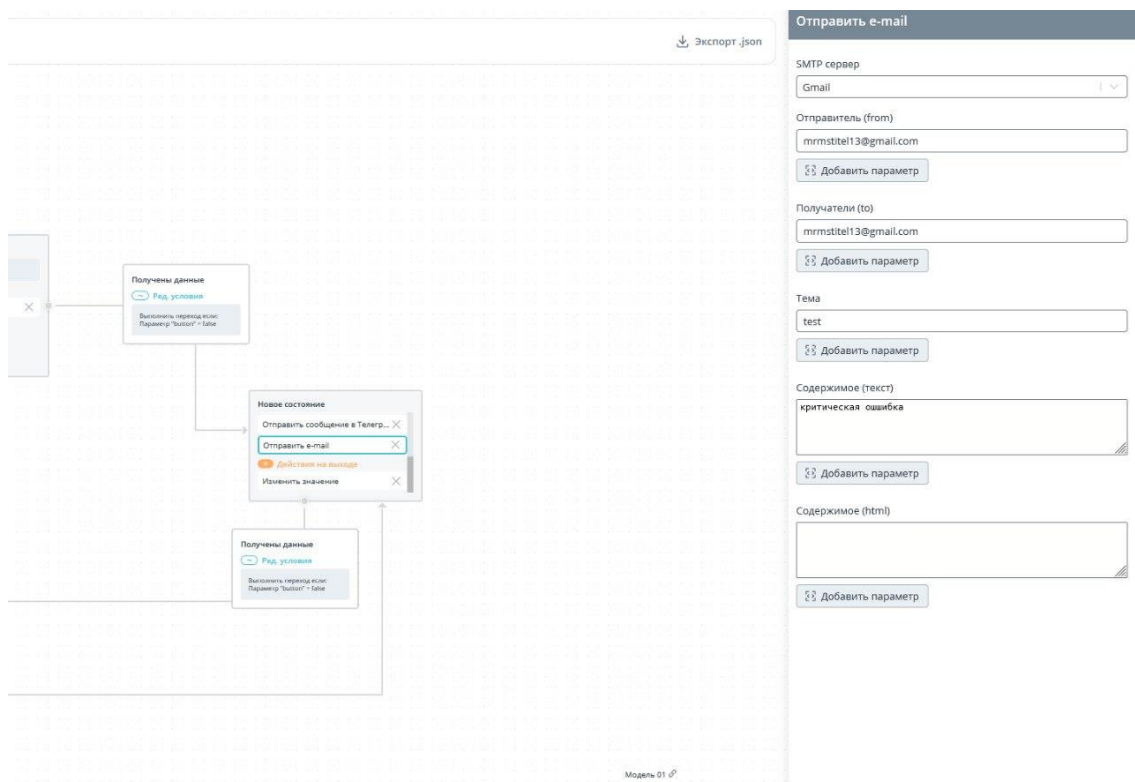


Рисунок 14 – Демонстрация отправки сообщения

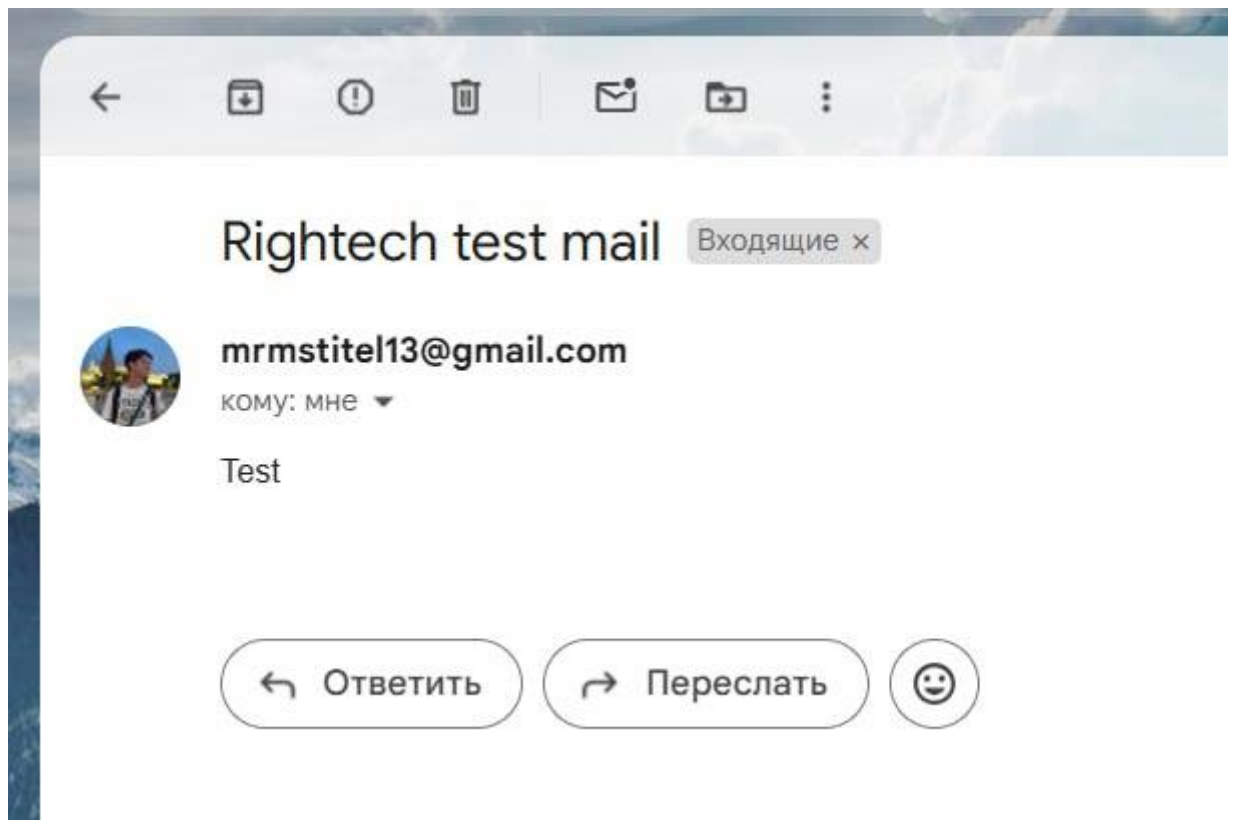


Рисунок 15 – Получение сообщение на почту

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Документация на чемодан: https://wirenboard.com/wiki/Wb-demo-kit_v.2
2. Веб-интерфейс WirenBoard:
https://wirenboard.com/wiki/Wiren_Board_Web_Interface
3. Утилита для извлечения исторических данных из внутренней базы данных: <https://wirenboard.com/wiki/Wb-mqtt-db-cli>
4. Протокол MQTT: <https://en.wikipedia.org/wiki/MQTT>
5. Описание улиты mosquito_sub: http://mosquitto.org/man/mosquitto_sub-1.html
6. Описание улиты mosquito_pub: https://mosquitto.org/man/mosquitto_pub-1.html
7. Описание протокола MQTT: <https://ipc2u.ru/articles/prostye-resheniya/chto-takoe-mqtt/>, <https://habr.com/ru/post/463669/>
8. Подключение к контроллеру по SSH: <https://wirenboard.com/wiki/SSH>
9. Визуализация: <https://tableau.pro/m11>
10. Графики: <https://tableau.pro/m16>
11. Гистограммы: <https://tableau.pro/m19>
12. Круговые диаграммы: <https://tableau.pro/m23>
13. Paho MQTT: <https://www.emqx.com/en/blog/how-to-use-mqtt-in-python>
14. Rightech IoT Cloud: [Документация](<https://dev.rightech.io>)
15. MQTT протокол: [Описание](<https://mqtt.org>)
16. Mosquitto MQTT: [Инструкция по установке](<https://mosquitto.org/man/mosquitto-8.html>)