|  |  |  |
| --- | --- | --- |
|  | Изображение выглядит как зарисовка, рисунок, символ, корона  Автоматически созданное описание |  |
| 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**

по дисциплине «Технологические основы Интернета вещей»

|  |  |
| --- | --- |
| **Выполнили:**  Студент группыИМБО-02-22 | Ким Кирилл Сергеевич |
| **Проверил:** | Синицын Иван Васильевич |

2024 г.

**Цель**:

Подключить и настроить устройство с поддержкой MQTT на платформе Rightech IoT Cloud, используя внешний MQTT-брокер.

**Выполнение практической работы:**

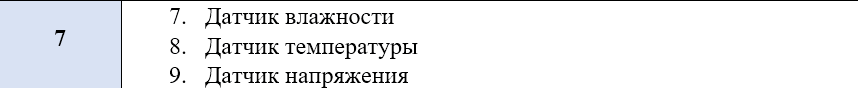
**Часть 1. Регистрация на платформе Rightech IoT Cloud**

Rightech IoT Cloud — это бескодовая (no-code) IoT-платформа для быстрого создания прикладных проектов интернета вещей. Для регистрации на платформе необходимо перейти по данной ссылке <https://dev.rightech.io/>

Программный продукт Rightech IoT Cloud (RIC, рус. Райтек ИоТ Клауд) от компании-разработчика КОМНЭТ является фреймворк-инструментом и предназначен для быстрого создания разработчиками приложений интернета вещей) Платформа RIC реализована на принципах универсализации и имеет низкую зависимость от конкретного оборудования и протоколов, что позволяет легко объединять разнородные устройства в едином разрабатываемом на платформе решении.

**Часть 2. Создание виртуальных устройств в облаке**

Согласно варианту по номеру бригады создайте в облаке виртуальные устройства для получения данных. Каждое устройство должно иметь свой профиль, соответствующий передаваемым на устройство данным. В качестве протокола для профилей устройств используйте MQTT.



**Рисунок 1 — Датчики для модели согласно варианту**

Модель объекта контроля — это формализованное представление устройства, подключаемого к платформе. В модели описываются параметры, которые отправляет устройство, и команды управления, которые оно может отработать. Единожды 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.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

**Рисунок 2 —Модель согласно варианту**

Для аргумента заполняются следующие поля:

1. тип данных: выбирается один из возможных типов данных: числовой, логический, строковый, объект или массив. Тип данных служит для корректного отображения значений параметров в интерфейсе. Также для определенных типов есть дополнительные возможности:

• числовой: указание множителя, единицы измерения, изображения и уровней перехода (разделы При разборе и Отображение);

• логический: указание изображения и уровней перехода (разделы При разборе и Отображение);

• объект: разбор данных в формате JSON (пример).

1. источник: по нему происходит поиск значения параметра в структуре данных, полученных от устройства; строго зависит от протокола. Конфигурирование источников для конкретных протоколов подробно рассмотрено в разделе примеров;
2. линейный: данный переключатель определяет, можно ли построить по параметру линейный график (плавно возрастающий или плавно убывающий): например, по температуре можно, а по широте или долготе - нет, такой график по отдельной координате не имеет смысла. При активации этого флага становится доступным добавление изображения в качестве иконки данного параметра и указание уровней индикации текущего значения параметра.

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

**Рисунок 3 — Созданный аргумент для температуры модели согласно варианту**

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

**Рисунок 4 — Созданный аргумент для влажности модели согласно варианту**

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

**Рисунок 5 — Созданный аргумент для напряжения модели согласно варианту**

**Часть 3. Отправка данных в облако**

Выполните передачу тестовых данных в каждое из созданных устройств. Данные должны соответствовать типу устройства, то есть, к примеру, в термометр должна поступить температура. Данные можно передавать при помощи утилиты mosquito\_pub. Ссылка на документацию по передаче данных на устройства по MQTT.

*Настройка MQTT брокера*:

Если MQTT брокер ещё не установлен, скачайте и установите MQTT брокер Mosquitto с [официального сайта](https://mosquitto.org/download/). Разрешите доступ для устройства через публичную сеть или используйте локальный сервер для тестов. Проверьте соединение с брокером, чтобы убедиться в корректной работе.

*Добавление устройства в облако Rightech*:

- В разделе «Объекты» на платформе Rightech добавьте новое устройство.

- Выберите ранее созданную модель и укажите параметры подключения:

- Адрес MQTT брокера.

- Порт (1883 или другой, который вы настроили).

- Топики для подписки и публикации данных (например, `device/data`).

- Логин и пароль для безопасного подключения, если это требуется вашим брокером.

*Мониторинг данных на платформе*:

Перейдем в панель управления в Rightech IoT Cloud. Проверим, поступают ли данные от устройства в реальном времени. В случае отсутствия данных проверьте:

Соединение устройства с брокером.

Корректность топиков MQTT.

Логи на платформе и устройстве.

Убедитесь, что данные передаются корректно и отображаются в интерфейсе платформы.

Пример MQTT запросов:

Например, с mosquitto\_pub клиентом из проекта [Eclipse Mosquitto](https://mosquitto.org/download/).

$ mosquitto\_pub -d -h dev.rightech.io -i <ric-mqtt-client-id> -t base/state/temperature -m 23

Публикация данных с устройства:

Топик: device/data

mosquitto\_pub -h broker\_address -t device/data -m '{"temperature": 25}'

Подписка на команды:

Топик: device/commands

mosquitto\_sub -h broker\_address -t device/commands

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

**Рисунок 6 — Передача значений температуры на датчик созданного объекта**

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

**Рисунок 7 — Передача значений влажности на датчик созданного объекта**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Рисунок 8 — Передача значений напряжения на датчик созданного объекта**

**Вывод:**

Созданы устройства, процесс отправки данных с облако, а также отображение этих данных на виртуальных устройствах в облачной платформе.