

Разработчик IoT

Длительность курса: 124 академических часа

1 Устройства и сети

- | | | |
|---|---|--|
| 1 | Общие положения Интернета вещей | <ul style="list-style-type: none">✓ разобраться, что такое Интернет вещей✓ понять, какие предпосылки были для его появления✓ изучить историю Интернета вещей, его перспективы и влияние✓ рассмотреть особенности применения Интернета вещей в разных сферах |
| 2 | Архитектура Интернета вещей | <ul style="list-style-type: none">✓ узнать экосистему и архитектуру Интернета вещей✓ научиться отличать большие данные от других данных✓ выбрать наиболее подходящую технологию сбора данных для своих задач |
| 3 | Системы бесконтактной идентификации RFID/NFC | изучить технологии бесконтактной идентификации; сформировать представления об области использования и их возможностей. |
| 4 | Датчики | изучить принципы работы сенсоров и их виды |
| 5 | Актуаторы и системы питания | сформировать представление о системах генерации и хранения энергии |
| 6 | Проектирование IoT-устройства на базе Arduino и Raspberry Pi | изучить базовые принципы электроники; изучить различие аналоговых и цифровых систем; научиться проектировать IoT-устройства на базе Arduino и Raspberry Pi |

Домашние задания

- 1 Мигание светодиодом с помощью Arduino по командам из Интернета. Обеспечить взаимодействие двух устройств между собой

Цель: Результатом ДЗ станут программы для Raspberry и Arduino, обеспечивающие взаимодействие по MQTT. Благодаря этому вы сможете обеспечивать удаленное управление светодиодом, подключенным к Raspberry.

В этом ДЗ тренируем умения:

1. Подключение исполнительных элементов к Raspberry
2. Подключение устройства к брокеру по MQTT с использованием разных библиотек
3. Удаленное управление устройствами

Для этого сделайте, пожалуйста, следующее:

1. Напишите программу для Raspberry на языке Python, которая позволит мигать светодиодом согласно международной азбуке Морзе. Для этого нужно получить сообщение по MQTT, подписавшись на определенный topic (например, "led/morse"). Payload этого сообщения необходимо ретранслировать по Морзе на светодиод. Длины единиц указаны в материалах к занятию.
2. Напишите программы для Arduino и Raspberry, которые позволят
 - каждые 5 секунд отправлять с Arduino сообщение с topic "led/single",
 - на Raspberry при получении сообщения с таким topic включать светодиод на 1 секунду.
3. В "чат с преподавателем" (см раздел в ЛК - "мои лекции"→"занятия"→"название занятия"→"чат с преподавателем") отправьте файлы с кодом или ссылку на репозиторий, например, на GitHub.

7 Сети и стандарты передачи данных

изучить основы сетей и систем связи
выявить важность и место сетевых технологий в Интернете вещей
изучить основные стандарты передачи данных;

8 Протоколы передачи данных

изучить термин "протокол передачи данных" в IoT;
выявить базовые принципы построения протоколов и механизмов сериализации и десериализации.

9 Моделирование поведения IoT-устройства

написать программный эмулятор IoT устройства, взаимодействующий с внешними система по MQTT

Домашние задания

- 1 Написать эмулятор выбранного устройства

Цель: Результатом ДЗ станет программный эмулятор IoT-устройства, подключенный по MQTT к брокеру. Эмулятор будет генерировать некоторые данные согласно логике

работы и реагировать на команды. Благодаря навыку написания эмуляторов можно отработать логику работы устройства в некотором бизнес-процессе, если в наличии не имеется работающего прототипа или готового устройства.

В этом ДЗ тренируем умения:

1. Взаимодействие по протоколу MQTT
2. Удаленное управление устройством
3. Формирование программных эмуляторов

Для этого сделайте, пожалуйста, следующее:

1. Напишите программу (язык программирования на выбор студента), которая будет эмулировать поведение какого-либо устройства. Под поведением устройства понимается отправка осмысленного набора данных и изменение каких-то данных при поступлении команд.

Например, можно смоделировать логику работы самоката следующим образом:

данные:

- статус самоката (включен или выключен),
- скорость,
- широта,
- долгота;

команды:

- включить и выключить самокат.

описание:

2. При поступлении команды включения самокат начинает движение, скорость становится больше нуля, координаты меняются (можно зациклить для движения по кругу). По команде выключения самокат останавливается.

Одновременно с моментом включения и выключения меняется и статус самоката.

Постарайтесь в своем эмуляторе реализовать как минимум три параметра, которые будут отправляться, при этом как минимум один из параметров должен реагировать на команду.

3. В "чат с преподавателем" (см раздел в ЛК - "мои лекции"→"занятия"→"название занятия"→"чат с преподавателем") отправьте

текстовое описание эмулированного устройства файлы с кодом или ссылку на репозиторий, например, на GitHub. Наиболее удобный и предпочтительный вариант - это docker-образ, в котором проект собирается и автоматически запускается, это позволит избежать казусов "на моей машине работает" и ускорит проверку.

10 Подробный разбор протокола

научиться писать сериализаторы протоколов

Домашние задания

- 1 Написать сериализатор данных выбранного устройства

Цель: Результатом ДЗ станет программный проект, который обеспечивает сериализацию данных по выбранному протоколу. Благодаря навыку сериализации можно программировать микроконтроллеры на желаемом устройстве для отправки данных на облачную платформу по одному из наиболее подходящих для задач устройства IoT протоколов.

В этом ДЗ тренируем умения:

1. Сериализация данных
2. Реализация протокола по документации

Для этого сделайте, пожалуйста, следующее:

1. Выберите протокол передачи данных, в котором описано формирование пакетов от устройства.

2. Реализуйте код, который позволит формировать пакет данных от устройства в виде бинарного пакета на основании данных, введенных в консоль.

Например, в консоль можно вывести сообщения "Введите температуру: ", ... "Введите влажность: " ... и т.д. После заполнения всех параметров выводится полученный бинарный пакет.

3. В "чат с преподавателем" (см раздел в ЛК - "мои лекции"→"занятия"→"название занятия"→"чат с преподавателем") отправьте файлы с кодом или ссылку на репозиторий, например, на GitHub. Наиболее удобный и предпочтительный вариант - это docker-образ, в котором проект собирается и автоматически запускается, это позволит избежать казусов "на моей машине работает" и ускорит проверку.

11 Получение данных и отправка команд

изучить механизмы десериализации данных и создать программу-десериализатор

Домашние задания

- 1 Написать десериализатор данных выбранного устройства

Цель: Результатом ДЗ станет программный проект, который обеспечивает десериализацию данных по выбранному протоколу. Благодаря навыку десериализации можно проектировать серверы, к которым подключаются IoT-устройства, и визуализировать полученные данные.

В этом ДЗ тренируем умения:

1. Десериализация данных
2. Реализация протокола по документации

Для этого сделайте, пожалуйста, следующее:

1. Для выбранного протокола реализуйте код, который позволит формировать пакет данных от устройства на основании полученного бинарного пакета.

Например, в консоль можно вывести сообщение "Введите пакет, полученный от устройства: "... После ввода пакета выводится пакет данных "Температура: ...", "Влажность: ..."

2. В "чат с преподавателем" (см раздел в ЛК - "мои лекции"→"занятия"→"название занятия"→"чат с преподавателем") отправьте файлы с кодом или ссылку на репозиторий, например, на GitHub. Наиболее удобный и предпочтительный вариант - это docker-образ, в котором проект собирается и автоматически запускается, это позволит избежать казусов "на моей машине работает" и ускорит проверку.

2 Платформы и приложения

1	Облачные технологии. Часть 1	изучить понятие “облака”; сформировать представление об архитектуре облаков и их видах.
2	Облачные технологии. Часть 2	проводить облачные и удаленные вычисления; осуществлять миграции в облако на основе базовых фаз миграции.
3	Интернет вещей. Платформы Интернета вещей	изучить платформы Интернета вещей, их место и роль
4	Знакомство с платформой Rightech IoT Cloud	<p>подключить устройства к платформе Интернета вещей</p> <p>Домашние задания</p> <p>1 Подключить к платформе эмулятор устройства, спроектированного на занятии 5 первого модуля. Обеспечить отображение данных и отправку команд</p> <p>Цель: Результатом ДЗ станет подключение эмулятора устройства, спроектированного на занятии 5 1-ого модуля, к платформе Rightech IoT Cloud. Благодаря этому вы сможете в дальнейшем взаимодействовать с эмулятором с помощью предоставленных платформой инструментов. Также можно будет осуществлять мониторинг параметров эмулятора и управление им с помощью команд.</p> <p>В этом ДЗ тренируем умения:</p> <ol style="list-style-type: none">1. Формирование цифровой модели устройства2. Формирование объекта для устройства3. Настройка устройства для подключения к облачной платформе4. Визуализация данных в интерфейсе <p>Для этого сделайте, пожалуйста, следующее:</p> <ol style="list-style-type: none">1. Спроектируйте цифровую модель устройства на платформе Rightech IoT Cloud для своего эмулятора согласно тем параметрам, которые он отправляет, и командам, на которые он реагирует. За базовую модель возьмите модель MQTT.2. Создайте объект, в котором добавьте модель, созданную в предыдущем пункте. Назначьте уникальный идентификатор, логин и пароль.3. Измените программный код эмулятора таким образом, чтобы обеспечить подключение к платформе. Адрес для подключения mqtt://dev.rightech.io:1883.4. Убедитесь в успешности подключения, дождитесь получения значений по всем параметрам и отправьте каждую из спроектированных команд на устройство.5. В “чат с преподавателем” (см раздел в ЛК - “мои лекции”→“занятия”→“название занятия”→“чат с преподавателем”) отправьте id модели, id объекта и

график по одному из параметров в виде картинки (скриншота).

5	Обзор облачных платформ. Часть 1	изучить функциональные возможности крупных PAAS/IAAS/SAAS платформ
6	Обзор облачных платформ. Часть 2	изучить возможности application enablement платформ; выявить границы использования платформ подобного типа.
7	Польза облачных технологий	выяснить основные преимущества облачных технологий; изучение механизмов быстрого прототипирования.
8	Обработка, хранение и визуализация данных в платформах	изучить основы обработки данных изучить механизмы обработки данных и принципы визуализации
9	Обработка данных	реализовать обработчик данных с заданной логикой Домашние задания 1 Написать обработчик параметров для эмулятора устройства Цель: Результатом ДЗ станет обработчик данных для эмулятора. Благодаря этому вы сможете редактировать пакеты, являющиеся невалидными по одному из полученных в пакете значений. В этом ДЗ тренируем умения: 1. Написание программного кода обработчика 2. Назначение обработчика на необходимый объект Для этого сделайте, пожалуйста, следующее: 1. Сформируйте критерии валидности по одному или более параметрам (например, скорость не может принимать отрицательное значение, или количество топлива в баке не может быть больше объема бака). 2. Напишите код обработчика, который при получении в пакете невалидного значения параметра, заменит это значение на последнее валидное значение, которое приходило. Например, если скорость приходила 25, 26, 24, а затем пришло значение -25, то необходимо заменить его на последнее значение 24. 3. Назначьте обработчик на необходимый объект и заполните в обработчике, какие параметры являются входными, какие-выходными. 4. Запустите обработчик. 5. Отправив невалидное значение, убедитесь, что оно заменилось на последнее валидное. 6. В "чат с преподавателем" (см раздел в ЛК - "мои лекции"→"занятия"→"название занятия"→"чат с преподавателем") отправьте id обработчика.

научиться создавать модель устройств под конкретную задачу;
создать цифровой объект в платформе;
реализовать сценарий облачной автоматизации и контроля

Домашние задания

- 1 Создать модель, объект и автомат по двум задачам (распределение преподавателем)

Цель: Результатом ДЗ станет модель, объект и автомат по двум задачам, то есть полноценное работающее решение по двум кейсам разной тематики. Благодаря этому вы сможете создавать необходимую автоматизацию для работы устройств в зависимости от поставленной задачи.

В этом ДЗ тренируем умения:

1. Создание модели под конкретную задачу
2. Создание объекта
3. Разработка автомата
4. Назначение автомата на необходимый объект

Список задач:

1. Мониторинг влажности и температуры воздуха на фармацевтическом складе. В случае если температура ниже 2 градусов Цельсия, необходимо отправить PUSH-уведомление с уровнем важности "Критическое" с текстом "Температура ниже допустимого предела". В случае если температура выше 8 градусов Цельсия, необходимо отправить PUSH-уведомление с уровнем важности "Критическое" с текстом "Температура выше допустимого предела". Аналогично в этом же автомате необходимо реализовать отправку уведомлений по отклонениям влажности, приняв за диапазон допустимой влажности 30..40 %.

Система контроля доступа с электронными картами при условии, что каждая карта имеет свой срок действия. Когда человек прикладывает свою карту к СКУД, на объект приходят значения по параметрам: номер карты и срок окончания действия (например, номер карты "62233", срок действия 09.09.2020 23:42:00). Если карта валидна, необходимо зажечь зеленую лампочку над дверью и отправить команду на открытие двери. Если карта невалидна, необходимо зажечь красную лампочку над дверью.

2. Адаптивное освещение офиса: автоматическая регулировка яркости лампочек в зависимости от того, насколько солнечно за окнами. В случае если освещение меньше 500 люкс, отправляем команду на включение большого дополнительного освещения. В случае, если освещение 500..700 люкс, отправляем команду на включение небольшого дополнительного освещения. Контроль освещения проводить только в случае, если в параметре "Количество людей в офисе" значение больше 0. Если людей в офисе нет, свет необходимо выключить полностью и контроль не проводить, пока кто-то не зайдет.

3. Умный мусорный контейнер. При помощи ультразвукового датчика определить уровень заполненности. Критерий заполненности определить

самостоятельно. При фиксации этого критерия отправить команду на изменение параметра “заполнен” с false на true. При этом индикация этого параметра должна поменяться с зеленого на красный цвет. При освобождении контейнера отправить false на параметр заполненности и изменить цвет иконки на зеленый. Реализовать эту логику для пяти мусорных контейнеров (контейнер == объект).
4. Умная теплица: требуется настроить полив цветов по расписанию. С помощью планировщика организовать расписание включения полива. Выбрать три времени в течение дня, в которое будет включаться полив. Завершать полив предлагается по таймеру на 10 секунд.

Каждому студенту на занятии будет выдано две темы, по каждой из которых предлагается выполнить следующее:
1. Сформируйте необходимую модель/модели и объект/объекты согласно поставленной задаче.
2. Разработайте автомат, корректно выполняющий поставленную задачу.
3. Назначьте автомат на необходимый объект/объекты.
4. Запустите автомат.
5. Протестируйте автомат, используя встроенный в платформу механизм ботов.
6. В “чат с преподавателем” (см раздел в ЛК - “мои лекции”→“занятия”→“название занятия”→“чат с преподавателем”) отправьте id моделей, объектов и автоматов.

11 **Платформы
Интернета вещей
как посредники
между “миром
устройств” и
“миром
приложений”**

изучить основы и перспективы межмашинного взаимодействия

12 **RESTful API**

научиться работать с HTTP API через специализированное ПО

Домашние задания

- 1 Сформировать запросы для получения списка устройств, данных с устройства (истории пакетов и текущего состояния) и отправки команд

Цель: Результатом ДЗ станет коллекция запросов, позволяющих работать с устройствами по посредством API платформы. Благодаря этому вы сможете выполнить задания по приложениям и чат-ботам в рамках третьего модуля.

В этом ДЗ тренируем умения:

1. Формирование API-запросов под конкретные задачи
2. Взаимодействие с платформой по API

Для этого сделайте, пожалуйста, следующее:

1. Сформируйте запрос для получения списка всех доступных объектов.
2. Сформируйте запрос для получения данных текущего состояния любого доступного объекта.

3. Сформируйте запрос для получения истории пакетов за определенный период времени (например, с 10:00 10 сентября 2020 года до 10:00 11 сентября 2020 года) любого доступного объекта.
 4. Сформируйте запрос для отправки одной из команд любого доступного объекта.
 5. В “чат с преподавателем” (см раздел в ЛК - “мои лекции”→“занятия”→“название занятия”→“чат с преподавателем”) отправьте POSTMAN коллекцию сформированных запросов и полученных ответов.
-

13 Разработка прототипа приложения

разработать прототип приложения;
научиться использовать внешний библиотеки;
настраивать переходы в приложении.

Домашние задания

- 1 Подключить библиотеку стилизованных компонентов reactstrap или аналогичную, заменить обычные компоненты на стилизованные, получить ключ API к личному аккаунту RIC. Изучить понятия Promise в JS, Redux, ActionCreator, Dispatch, Reducer, а также как делать

Цель: Результатом выполнения домашнего задания станет web-приложение, в котором вы сможете переходить по разделам. Благодаря этому вы получите представление о маршрутизации в приложении.

В этом ДЗ тренируем умения:

1. Использовать внешние библиотеки компонентов
2. Настраивать переходы в приложении

Результат ДЗ:

1. Ссылка на репозиторий с обновленным приложением

Для этого сделайте, пожалуйста, следующее:

1. Установите библиотеку стилизованных компонентов по своему усмотрению, например, можно использовать Bootstrap или Reactstrap
 2. Измените ссылки в Menu-компоненте на Link из React Router
 3. Создайте компонент Home, который отрисовывает текст “Выберите устройство”
 4. Создайте компонент DeviceStatistic, который в props получает идентификатор устройства и тут же его отрисовывает.
 5. Создайте компонент ContentRouter, который зависимости от URL будет отрисовывать либо Home, либо DeviceStatistic.
 6. Измените App.jsx так, чтобы в нем отрисовывались Меню и Контент
-

14 Приложения - безопасность

изучить наиболее популярные механизмы авторизации и аутентификации;
реализовать один из механизмов авторизации;
создать прототип приложение, использующее авторизацию к платформе Интернета вещей по REST API.

Домашние задания

- 1 Написать собственный Hello world на React.js, используя перечисленные на занятии технологии

Цель: Результатом выполнения домашнего задания станет приложение, которое послужит шаблоном с основными механизмами авторизации для выполнения следующих домашних заданий и использования в будущих проектах.

В этом ДЗ тренируем умения:

1. В реализации и понимании процессов стандартизированных механизмов авторизации и аутентификации.
2. Реализации наиболее популярные механизмы авторизации.
3. Реализации двухфакторной аутентификации.
4. Использовать библиотеки для авторизации.

Для этого сделайте, пожалуйста, следующее:

1. В реализованный ранее прототип приложения встройте логику авторизации с платформой по токену.
2. Вынесите token в env-переменную или в конфиг приложения.
3. Добавьте механизмы авторизации по jwt токену.
4. Добавьте механизмы авторизации по cookie.
5. Сформируйте имитацию двухфакторной аутентификации и осуществите авторизацию.

15 Управление глобальным состоянием приложения

научиться управлять глобальным состоянием приложения;

Домашние задания

- 1 Управление глобальным состоянием приложения

Цель: Добавить в приложение управление темой с помощью контекста. Тема может быть не только dark/light вместо этого можно опираться на выбранную библиотеку, например primary/secondary или info/danger/success. Тема должна применяться ко всем компонентам, которые поддерживают вариативное отображение.

Результатом выполнения задания станет приложение, в котором можно переключать визуальную тему. В процессе выполнения задания вы сможете закрепить знания об управлении глобальным состоянием приложения.

В этом ДЗ тренируем умение управлять контекстами и глобальным состоянием

Для этого сделайте, пожалуйста, следующее:

1. Создайте новый контекст для управления темой.
2. Создайте компонент для переключения темы, результат переключения должен сохраняться в созданном контексте
3. Добавьте вариативное отображение для компонентов, которые это поддерживают.

16 **Чат-бот для быстрых команд устройству**

разработать чат-бота;
научиться реализовывать обработчики команд.

Домашние задания

- 1 Обработчик сообщения “Прибыл” для чат-бота. По нажатию на кнопку “Прибыл” бот отправляет запрос в платформу на запуск автомата и сообщает пользователю о результатах запроса

Цель: Результатом выполнения задания станет приложение, которое упростит доступ к контролю устройством.

В этом ДЗ тренируем умение писать собственные обработчики сообщений для чат-бота

Для этого сделайте, пожалуйста, следующее:

1. Добавьте обработчик команды /start, который возвращает клавиатуру с одной кнопкой “Начать работу” (текст может быть любым), а также описывает, что произойдет, если нажать на эту кнопку.
2. Добавьте обработчик нажатия на кнопку.

17 **Q&A по курсу**

получить ответы на вопросы по ДЗ;
получить ответы на вопросы по приложениям.

1 Консультация по проектам и домашним заданиям

получить ответы на вопросы по проекту, ДЗ и по курсу.

Домашние задания

- 1 Итоговый проект (описание Проекта в удобном формате приложено в материалах)

Цель: Проект состоит из трех частей в соответствии с пройденными модулями.

Проектное ДЗ не включает в себя ДЗ, выполненные в рамках курса, но базируется на тех же принципах выполнения и пройденном материале.

Ориентировочное время выполнения - 18 часов.

Предоставляется 2 варианта для выполнения:

1. базовый проект
2. творческий проект

Цель:

Предложенный базовый проект является реальной задачей для многих частных производителей сельхозпродукции и крупных компаний, занимающихся логистикой, хранением и сбытом товаров. Иными словами, вам предлагается решить реальный кейс из IoT сферы.

В рамках реализации данного решения проверяем получение следующих умений:

- прототипирование и разработка программных эмуляторов;
- проектирование сценариев взаимодействия умных устройств и эмуляторов с внешними сервисами и облачными платформами Интернета вещей;
- создание “цифровой копии” реальных устройств, групп устройств;
- организация взаимодействия устройств с помощью облачных технологий;
- потоковая обработка данных устройств;
- организация клиент-серверного взаимодействия с использованием REST-full API;
- реализация чат-ботов.

После выполнения задания вы сможете проектировать и разрабатывать кейсы в области Интернета вещей с использованием современных подходов. Также вы научитесь создавать гибкие, масштабируемые решения с использованием облачных технологий и организовывать взаимосвязи устройств различного рода.

****Описание базового проекта****

1. В базовом проекте предлагается реализовать комплексное решение “Умное фермерство”.

В рамках решения необходимо обеспечить полный автоматизированный контроль параметров перевозимой фермерской продукции от места выращивания до склада хранения.

****Устройство №1 - оснащение кузова грузовика****

Кузов грузовика для перевозки продукции оснащен специальным оборудованием, позволяющим измерять различные параметры:

- температура внутри кузова;
- температура снаружи кузова;
- состояние кондиционера;
- влажность внутри кузова;
- влажность снаружи кузова;
- скорость грузовика;
- широта;
- долгота;
- уровень тряски;
- фиксация резких торможений;
- выделение газов внутри кузова;
- масса продукции.

Для управления температурой внутри кузова имеется исполнительное устройство, отвечающее за включение и выключение кондиционера.

****Устройство №2 - оснащение хранилища****

Хранилище также оснащено специальным оборудованием, позволяющим измерять различные параметры в процессе хранения фермерской продукции:

- температура;
- состояние кондиционера;
- состояние шлагбаума;
- влажность;
- выделение газов;
- масса продукции;
- кратность воздухообмена.

В хранилище установлены исполнительные устройства, позволяющие включать или выключать воздухообмен и/или систему охлаждения. Также предусмотрено управление шлагбаумом, расположенным перед складом.

В рамках задания, пожалуйста, выполните три основных этапа:

Этап 1. Эмулирование данных

1. Разработайте программный эмулятор двух описанных устройств, позволяющий

- подключаться к платформе Rightech IoT Cloud по протоколу MQTT,

- отправлять значения параметров,

- получать команды из платформы,

- обеспечивать реакцию на команды (например, переключать состояние кондиционера в зависимости от полученной команды включения или выключения).

2. Программный эмулятор может быть выполнен в любом из трех вариантов:

- циклическая отправка сгенерированных в коде параметров,

- ввод параметров для отправки в окне терминала,

- ввод параметров для отправки в графическом интерфейсе пользователя (GUI).

3. Выбор наиболее удобного варианта остается на ваше усмотрение.

Примечание 1 - В случае возникновения сложностей с данным этапом эмулирование данных можно произвести с помощью встроенного в платформу сервиса ботов. Это позволит перейти к выполнению последующих этапов, но этап 1 будет оценен в 0 баллов.

Примечание 2 - Вместо эмулятора можно спроектировать реальное устройство (стенд), позволяющее измерять параметры и исполнять команды.

Этап 2. Работа на платформе Rightech IoT Cloud

На платформе Rightech IoT Cloud сделайте следующее:

1. Создайте цифровые модели устройств, в которых будут отражены все полученные параметры и заготовлены команды для исполнения
2. Создайте объекты для подключения устройств
3. Обеспечьте подключение эмуляторов
4. Реализуйте обработчик, который в зависимости от полученных данных формирует актуальный статус грузовика (строковый параметр в модели):
 - Если скорость грузовика больше 0 и масса продуктов больше 0, отображать статус “На склад”,
 - Если скорость грузовика больше 0 и масса продуктов равна 0, отображать статус “На ферму”,
 - Если скорость грузовика равна 0 и масса продуктов больше 0, отображать статус “Стоянка-полный”,
 - Если скорость грузовика равна 0 и масса продуктов равна 0, отображать статус “Стоянка-пустой”,
 - Создайте геозону склада

Реализуйте автоматы:

6.1 Автомат, контролирующий температуру внутри грузовика/склада. Обеспечивает включение системы кондиционирования при повышении температуры до 10 градусов Цельсия и выключение при снижении до 5 градусов. Обратите внимание, как для грузовика, так и для склада необходима одна и та же логика выполнения сценария, поэтому нет необходимости реализовывать 2 автомата.

6.2 Автомат, обеспечивающий автоматическое открытие шлагбаума при попадании грузовика в геозону склада (взаимодействие двух объектов в одном автомате) и закрытие при выходе из геозоны.

6.3 Автомат, контролирующий качество вождения. При получении уровня тряски более 80 процентов или при фиксации резкого торможения, отправлять уведомление на почту фермеру о низком качестве вождения.

Этап 3. Работа с API

После того, как фермер передал свой груз водителю, основная потребность фермера - отслеживать состояние переданного груза. Для этого разработайте веб-приложение для фермера для отслеживания параметров продукции в процессе перевозки и хранения. Отображение значений реализуйте на специальных дашбордах, отражающих основные показатели. Графики могут быть скомбинированы на ваше усмотрение. Например, на одной вкладке могут находиться график температуры и влажности, на другой - скорости и массы, либо для каждого графика может быть предусмотрена отдельная вкладка.

Автоматические средства срабатывают не всегда корректно, например, автомат, который включается по прибытии автомобиля в геозону, может не сработать, если GPS-трекер дал сбой. На этот случай водитель должен сам сообщить о своем прибытии и запустить дальнейшую цепочку событий. Для этого реализуйте чат-бота, в котором есть встроенная клавиатура с одной кнопкой:

“Прибыл”. По нажатию на кнопку отправляется запрос на открытие шлагбаума. Если запрос проходит успешно (получено значение, что шлагбаум открыт), водителю приходит сообщение с дальнейшими инструкциями, например, “Отлично! Вы можете проехать на разгрузку к складу №5”. В случае неудачи (за отведенное время шлагбаум не открылся), вежливо сообщить водителю, что что-то пошло не так и можно попробовать еще раз.

Чат-бот может быть реализован в любом мессенджере: Facebook, Telegram, Vk и т.д.

Описание творческого проекта

При выполнении творческого проекта необходимо составить описание функционирования эмулятора, всех спроектированных сущностей платформы и приложений.

Предлагается реализовать проект на одну из представленных тем:

#умный город

#умная промышленность

#умная медицина

#умное сельское хозяйство

#умный транспорт

#умное ЖКХ

#умная энергетика

#умный ритейл

#умные технологии и носимые устройства

В рамках задания, пожалуйста, выполните три основных этапа:

Этап 1. Эмулирование данных

Разработайте программные эмуляторы подвижного и стационарного устройства, позволяющие

- подключаться к платформе Rigtech IoT Cloud по любому из поддерживаемых протоколов,

- отправлять значения 10 и более параметров подвижного устройства и 5 и более параметров стационарного устройства,

- получать команды из платформы,

- обеспечивать реакцию на команды.

Программный эмулятор так же, как и в базовом проекте, может быть выполнен в любом из трех вариантов:

- циклическая отправка сгенерированных в коде параметров,

- ввод параметров для отправки в окне терминала,

- ввод параметров для отправки в графическом интерфейсе пользователя (GUI).

Выбор наиболее удобного варианта остается на ваше усмотрение.

Примечание 1 - В случае возникновения сложностей с данным этапом, эмулирование данных можно произвести с помощью встроенного в платформу сервиса ботов. Это позволит перейти к выполнению последующих этапов, но этап 1 будет оценен в 0 баллов.

Примечание 2 - Вместо эмулятора можно спроектировать реальное устройство (стенд), позволяющее измерять параметры и исполнять команды.

Этап 2. Работа на платформе Rigtech IoT Cloud

На платформе Rigtech IoT Cloud сделайте следующее:

- Создайте цифровые модели устройств, в которых будут отражены все полученные параметры и заготовлены

- команды для исполнения
- Создайте объекты для подключения устройств
- Обеспечьте подключение эмуляторов
- Реализуйте обработчик данных на одном из объектов
- Создайте геозону для использования в сценарии контроля геозоны

Реализуйте автоматы:

6.1 Автомат, контролирующий некоторые параметры и отправляющий команды при наступлении конкретных событий

6.2 Автомат, обрабатывающий сценарий по реакции на вход/выход из геозоны

6.3 Автомат, отправляющий уведомления на почту при наступлении конкретных событий

Этап 3. Работа с API

1. Разработайте веб-приложение для отслеживания параметров объекта/ов. Отображение значений реализуйте на специальных дашбордах, отражающих основные показатели. Графики могут быть скомбинированы на ваше усмотрение.

2. Реализуйте чат-бота, в котором есть встроенная клавиатура с одной или несколькими кнопками. По нажатию на кнопку должна быть отправлена одна или несколько команд на исполнение или произведен запуск автомата. На любое нажатие кнопки должна быть реакция в виде ответа чат-бота.

3. Чат-бот может быть реализован в любом мессенджере: Facebook, Telegram, Vk и т.д.

2

Защита проектных работ

защитить проект и получить рекомендации экспертов.