

Разработчик IoT

Длительность курса: 132 академических часа

1 Общие положения "Интернета вещей"

Цели занятия:

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.

Краткое содержание:

что такое IoT;
история и возможное будущее "Интернета вещей",
планы и прогнозы внедрения;
сценарии и направления практического применения.

2 Архитектура "Интернета вещей"

Цели занятия:

узнать экосистему и архитектуру "Интернета вещей";
научиться отличать большие данные от других данных;
выбрать наиболее подходящую технологию сбора данных для своих задач.

Краткое содержание:

экосистема и архитектура "Интернета вещей";
что значит S в слове IoT;
драйверы и барьеры развития;
взаимодействие IoT с перспективными технологиями.

3 Системы бесконтактной идентификации RFID/NFC

Цели занятия:

изучить технологии бесконтактной идентификации; сформировать представление об их возможностях и областях использования.

Краткое содержание:

общие сведения о RFID;
метки и считывающие устройства RFID;
области применения RFID;
общие сведения о NFC;
преимущества NFC;
области применения NFC.

4 Датчики

Цели занятия:

изучить принципы работы датчиков и их виды;
изучить базовые принципы электроники;
изучить различие аналоговых и цифровых систем;
научиться строить электрические схемы в Tinkercad.

Краткое содержание:

разные типы датчиков (температуры, влажности, тока, освещенности);
основные термины и понятия электроники;
различия аналоговых и цифровых систем;
работа аналоговых входов Arduino;
принципы построения делителей напряжения;
моделирование в Tinkercad.

5 Датчики и актуаторы - часть 1

Цели занятия:

изучить принципы работы актуаторов и их виды; понять, что такое ШИМ и как он генерируется на Arduino.

Краткое содержание:

разные типы актуаторов (светодиод; пьезодинамик; мотор; сервопривод); работа аналоговых выходов Arduino; моделирование в Tinkercad.

6 Датчики и актуаторы - часть 2

Цели занятия:

научиться строить электрические схемы в Tinkercad с использованием разных актуаторов.

Краткое содержание:

доклады студентов и обсуждение; моделирование со светодиодом; моделирование с сервоприводом.

Домашние задания

1 Спроектировать систему в Tinkercad

Цель: Результатом ДЗ станет схема и код, благодаря которому вы сможете обеспечивать взаимодействие датчика и актуатора.

В этом ДЗ тренируем умения:

1. Подключение устройств в схеме на платформе TinkerCAD.
2. Написание кода для считывания и записи данных.

Для этого сделайте, пожалуйста, следующее:

1. Соберите схему, включающую в себя:

- * Arduino UNO
- * макетную плату
- * пьезоэлемент
- * датчик силы нажатия (Force Sensor)

* резистор/резисторы (необходимый номинал/номиналы подбираете самостоятельно)

2. Напишите код, который позволяет

* считать значение с датчика силы

* отобразить его в диапазон от 1000 до 2500

* записать полученное значение в качестве частоты звучания для пьезоэлемента, где 1000 - это 1 кГц, а 2500 - 2,5 кГц соответственно

Примечание: Длительность звучания пьезоэлемента - любая.

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

7 Контроллеры и системы питания

Цели занятия:

научиться проектировать IoT-устройства на базе Arduino и Raspberry Pi;
сформировать представление о системах питания IoT-систем.

Краткое содержание:

микроконтроллеры, их связь с датчиками и управляемыми элементами;
управление питанием, системы хранения энергии.

Домашние задания

1 Мигание светодиодом по азбуке Морзе с помощью Raspberry Pi

Цель: Результатом ДЗ станут программа для Raspberry на языке Python, которая осуществляет индикацию (светодиодом или соответствующим числом) согласно полученному тексту.

В этом ДЗ тренируем умения:

1. Подключение исполнительных элементов к Raspberry
2. Написание кода Python

Для этого сделайте, пожалуйста, следующее:
1. Напишите программу для Raspberry на языке Python, которая позволит мигать светодиодом согласно международной азбуке Морзе. Для этого нужно ввести в консоль некоторый текст, который необходимо ретранслировать по Морзе на светодиод, где 1 - включение светодиода на 1 секунду, 0 - выключение. Длины единиц указаны в материалах к занятию.

Примечание: Raspberry - это просто миниатюрный компьютер с некоторыми интерфейсами, в том числе с программируемыми портами ввода-вывода GPIO, к которым по ДЗ необходимо подключить светодиод. Поэтому при отсутствии Raspberry или возможности подключения светодиода замените Raspberry на свой личный компьютер, а вместо включения/выключения светодиода выведите в консоль результат в виде последовательности 1/0.

2. В "чат с преподавателем" (см раздел в ЛК - "мои лекции" → "занятия" → "название занятия" → "чат с преподавателем") отправьте файл с кодом или ссылку на репозиторий, например, на GitHub.

8 Протоколы передачи данных - часть 1

Цели занятия:

изучить термин "протокол передачи данных" в IoT;
изучить тенденции протоколов IoT на уровнях сетевой модели OSI;
разобраться в логике работы протокола MQTT;
запомнить принципы формирования основных пакетов MQTT;
научиться работать с брокером mosquitto.

Краткое содержание:

типы сетей;
протоколы и их разновидности;
обзор протокола MQTT;
схема взаимодействия MQTT;
структура пакетов MQTT;
QOS;
формирование топиков.

Цели занятия:

написать своего программного MQTT клиента;
научиться работать с клиентом MQTT X;
подключить устройство с Wi-Fi модулем к облачному брокеру.

Краткое содержание:

вариации реализации MQTT: терминал, код, GUI, Arduino.

Домашние задания

1 Взаимодействие 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 устройства к брокеру по MQTT с использованием разных библиотек
2. Удаленное управление устройствами

Для этого сделайте, пожалуйста, следующее:

1. Исправьте код программы для Raspberry таким образом, чтобы мигание светодиода по азбуке Морзе происходило по сообщению, полученному от брокера. Для этого нужно получить сообщение по MQTT, подписавшись на определенный topic (например, "led/morse"). Payload этого сообщения необходимо ретранслировать по Морзе на светодиод.
2. Напишите программы для Arduino и Raspberry, которые позволят
 - каждые 15 секунд отправлять с Arduino сообщение с topic "led/single",
 - на Raspberry при получении сообщения с таким topic включать светодиод на 1 секунду.

Подключение к брокеру как для Arduino, так и для Raspberry реализовать с логином и паролем!

Примечание: Под Arduino подразумевается либо модуль ESP8266/ESP32, либо - более

рекомендуемый вариант - плата на базе такого модуля (NodeMCUи др.).

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.

10 Моделирование поведения IoT-устройства

Цели занятия:

написать программный эмулятор IoT устройства; взаимодействующий с внешними система по MQTT.

Краткое содержание:

программное эмулирование устройства: отправка данных и реакция на команды.

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 и 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 Подробный разбор протокола

Цели занятия:

научиться писать сериализаторы протоколов.

Краткое содержание:

обзор протоколов на уровне побайтового формирования данных;
написание сериализатора (параметры-
>последовательность байт) для протокола.

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.

Примеры для протоколов(но для реализации можно выбрать другие)

modbus TCP - <https://ipc2u.ru/articles/prostye-resheniya/modbus-tcp/>

modbus RTU - <https://ipc2u.ru/articles/prostye-resheniya/modbus-rtu/>

MQTT - <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>

ВЕГА СИ-22 - https://iotvega.com/content/ru/si/si22/ВЕГА СИ-22 РП_rev 11.pdf (18-20 страницы)

ВЕГА GM-2 - https://iotvega.com/content/ru/si/gm2/01-ВЕГА GM-2 РП_rev 05.pdf (после 26 страницы)

Ссылка на списки продуктов веги - <https://iotvega.com/product>

Краткое содержание:

десериализация данных;
формирование клиент-серверной архитектуры;
сбор данных с возможностью управления устройством;
автоматизация процессов, преимущества использования.

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.

1 Облачные технологии - часть 1

Цели занятия:

изучить понятие “облака”;
сформировать представление об архитектуре облаков и их видах;
научиться выбирать модель облака для конкретной задачи.

Краткое содержание:

истоки облачных вычислений;
понятие “облака”;
виды облаков и их назначение;
архитектуры облачных систем;
виды облачных услуг: Paas. Iaas. Saas.

2 Облачные технологии - часть 2

Цели занятия:

проводить облачные и удаленные вычисления;
осуществлять миграции в облако на основе базовых фаз миграции.

Краткое содержание:

модели облачных вычислений;
частное, общедоступное, гибридное и общественное облако;
облачные и кластерные вычисления;
достоинства и недостатки облачных вычислений;
миграция, фазы миграции в облако.

3 Интернет вещей. Платформы Интернета вещей	Цели занятия: изучить платформы "Интернета вещей", их место и роль; научиться грамотно выбирать платформу в соответствии с ее типом и решаемыми задачами. Краткое содержание: виды и типы платформ, их назначение; место платформ в архитектуре "Интернета вещей"; основные компоненты платформ "Интернета вещей".
4 Обзор облачных платформ Интернета вещей - AWS	Цели занятия: научиться использовать инструменты AWS для реализации IoT решений. Краткое содержание: AWS Cloud Services; AWS IoT Core.
5 Взаимодействие MQTT-клиентов с облачным брокером	Цели занятия: подключать устройства в разном исполнении к брокеру MQTT; обеспечивать взаимодействие нескольких устройств на основе концепции "издание-подписка"; разбираться в командах Modbus TCP; понимать базовые принципы построения протоколов и механизмов сериализации и десериализации. Краткое содержание: повторение MQTT; работа с модулями ESP; протокол Modbus и его вариации; сбор данных и отправка команд.

Цели занятия:

создать модель своего устройства под конкретную задачу;
создать объект устройства;
подключить устройство к платформе "Интернета вещей";

Краткое содержание:

вход на платформу;
обзор основных модулей;
подробное рассмотрение сущностей модели и объекта;
визуализация данных;
подключение к платформе MQTT-клиента и модуля ESP.

Домашние задания

- 1 Подключить к платформе эмулятор устройства, спроектированного в ДЗ 4. Обеспечить отображение данных и отправку команд

Цель: Результатом ДЗ станет подключение эмулятора устройства, спроектированного в ДЗ 4, к платформе Rightech IoT Cloud. Благодаря этому вы сможете в дальнейшем взаимодействовать с эмулятором с помощью предоставленной платформой инструментов. Также можно будет осуществлять мониторинг параметров эмулятора и управление им с помощью команд.

При отсутствии выполненного ДЗ выполните задание для эмулятора, спроектированного на занятии 10.

В этом ДЗ тренируем умения:

1. Формирование цифровой модели устройства
2. Формирование объекта для устройства
3. Настройка устройства для подключения к облачной платформе
4. Визуализация данных в интерфейсе

Для этого сделайте, пожалуйста, следующее:

1. Спроектируйте 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 новый код эмулятора, id модели, id объекта и график по одному из параметров в виде картинки (скриншота).

7 **Обзор облачных платформ Интернета вещей - Google Cloud Platform**

Цели занятия:

научиться использовать инструменты Google Cloud для реализации IoT решений.

Краткое содержание:

GCP Services;
GCP IoT Core.

8 **Обзор облачных платформ Интернета вещей - Losant**

Цели занятия:

изучить основные возможности платформы; научиться создавать IoT решения на ее базе.

Краткое содержание:

платформа Losant и ее инструменты.

9 **Обзор облачных платформ Интернета вещей - Rightech IoT Cloud**

Цели занятия:

изучить основные инструменты платформы;
сформировать представление о последовательности процессов и этапов передачи и обработки данных для построения комплексного решения.

Краткое содержание:

подключение устройств;
нормализация данных;
обработка и фильтрация данных;
автоматизация бизнес процесса;
хранение и визуализация данных;
интеграция с внешними системами.

10 **Обработка данных - часть 1**

Цели занятия:

создать обработчики данных для решения задач различного типа;
познакомиться с сервисом ботов.

Краткое содержание:

что такое обработчики;
общие примеры;
начинаем реализацию кейса для самоката и стойки.

Цели занятия:

закрепить навыки по работе с обработчиками.

Краткое содержание:

продолжаем практиковаться с обработчиками на примере кейса с самокатом.

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.
-

Цели занятия:

разобраться в понятии автоматизации.

Краткое содержание:

что такое автоматизация и где она применяется; автоматы и как ими пользоваться.

Домашние задания

1 Найти ошибки в автомате

Цель: Результатом ДЗ станет доработка реализованного автомата для выполнения описанной логики процесса. Благодаря этому вы сможете анализировать автоматы и проектировать их таким образом, чтобы не происходило некорректных зависаний в одном из состояний автомата.

В этом ДЗ тренируем умения:

1. Изучение состояний и переходов в автомате
2. Проектирование автомата под описанную логику

В файле материалов занятия представлен автомат “Garland manager”, созданный для моделей гирлянды и Zigbee-кнопки на платформе Rightech IoT Cloud.

Необходимая логика работы автомата:

Автомат предназначен для управления гирляндой по Zigbee-кнопке.

- В первом состоянии после старта автомата (“Ожидание нажатия кнопки”) запускается планировщик, который выключает гирлянду по расписанию. Автомат выйдет из этого состояния, если произойдет нажатие кнопки (переход в состояние “Получен пакет”) или сработает планировщик, выключающий гирлянду в определенное время (переход в состояние “Выключение”). Каждый раз после отработки определенного режима гирлянды происходит возврат в это состояние.
- Из состояния “Получен пакет” по таймеру и типу клика осуществляется переход в соответствующие

режимы гирлянды (состояния “Плавное свечение”, “Быстрое мигание” и “Включение/выключение”) или, в случае если пришли некорректные данные, происходит возврат в состояние “Ожидание нажатия кнопки”.

- Из состояния “Включение/выключение”, в зависимости от текущего режима гирлянды, происходит переход в состояние “Выключение” или “Включение”.

Ошибки реализации автомата:

- Ошибка № 1: В автомате есть состояние, после входа в которое выйти из него уже не получится. Как называется это состояние? Какое действие нужно добавить в это состояние, чтобы автомат работал корректно?

- Ошибка № 2: В автомате есть переход между состояниями, который никогда не произойдет. Как называется этот переход? Какое действие и в какое состояние нужно добавить, чтобы автомат работал корректно?

В “чат с преподавателем” (см раздел в ЛК - “мои лекции” → “занятия” → “название занятия” → “чат с преподавателем”) отправьте описание найденных ошибок.

13 Автоматизация процессов - часть 2

Цели занятия:

реализовать сценарии облачной автоматизации и контроля.

Краткое содержание:

завершаем реализацию кейса для самоката и стойки.

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. Диапазон допустимой температуры 2..8 градусов Цельсия. Диапазон допустимой влажности 30..40 %. В случае если происходит отклонение по любому из параметров, необходимо отправить PUSH-уведомление с уровнем важности “Критическое” с текстом “Отклонение параметров от допустимых значений”. В случае если все параметры в норме, необходимо отправить PUSH-уведомление с уровнем важности “Информационное” с текстом “Параметры имеют допустимые значения”.
2. 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 срок действия карты не истек (текущее время меньше срока действия карты), необходимо зажечь зеленую лампочку над дверью и отправить команду на открытие двери. Если карта невалидна, необходимо зажечь красную лампочку над дверью.
3. 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.
4. Умный мусорный контейнер. При помощи ультразвукового дальномера 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).

5. 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 по истечении таймера 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.
-

Цели занятия:

познакомиться с протоколом TLS;
изучить наиболее популярные механизмы
аутентификации и авторизации.

Краткое содержание:

основные концепции информационной безопасности;
механизмы аутентификации и авторизации;
двухфакторная аутентификация.

Домашние задания

- 1 Издание сертификата X.509 и подключение к IoT платформе по протоколу MQTTS

Цель: Результатом выполнения домашнего задания станет скрипт который осуществляет безопасный обмен данными по протоколу MQTTS.

В этом ДЗ тренируем умения:

1. Создания сертификатов.
2. Использования сертификатов в своих приложениях.

Для этого сделайте, пожалуйста, следующее:

1. Подключитесь к любой IoT платформе по протоколу MQTTS
 2. Если платформа поддерживает издание сертификатов для устройств - издайте его и попробуйте подключиться используя его
-

Цели занятия:

научиться работать с HTTP API через специализированное ПО.

Краткое содержание:

Postman;
работа с платформой посредством API.

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.
-

16 Визуализация данных с Grafana

Цели занятия:

изучить технологии визуализации данных при помощи ПО Grafana.

Краткое содержание:

изучение терминов и инструментов;
обзор встроенных плагинов и панелей Grafana;
изучение плагина Infinity Datasource;
изучение плагина Rightech IoT Datasource;

Домашние задания

1 Визуализация данных в приложении

Цель: Результатом выполнения домашнего задания станет панель grafana

В этом ДЗ тренируем умения:

1. Использовать внешние библиотеки визуализации
2. Выводить показания и строить графики

Результат ДЗ:

1. Файл с панелью grafana

Для этого сделайте, пожалуйста, следующее:

1. Разработать компонент "Показание устройства" для отображения текущих показаний
 2. Для выбранного устройства вывести показания и построить график
-

17 Вебхуки

Цели занятия:

изучение понятия "вебхук";
создания вебхуков в нескольких системах;
интеграция систем через вебхуки;

Краткое содержание:

входящие и исходящие вебхуки;
знакомство с сервисом IFTTT;
вебхуки в платформе Rightech;

18 Q&A по курсу

Цели занятия:

получить ответы на вопросы по курсу и текущим ДЗ;
разобраться с проектной работой.

Краткое содержание:

ответы на вопросы;
выдача проектной работы.

1 Консультация по проектам

Цели занятия:

получить ответы на конкретным проектным работам;
контрольный срез выполнения работ.

Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;
затруднения при выполнении ДЗ;
вопросы по программе.

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;
- визуализация данных с помощью инструмента Grafana;
- вызов вебхуков.

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 в процентах:
 - * аргон;
 - * азот;
 - * гелий;
 - * водород;
 - * углекислый газ;
 - * угарный газ;
 - * кислород;
- масса продукции.

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

****Устройство №2 - оснащение склада****

Склад также оснащен специальным оборудованием, позволяющим измерять различные параметры в процессе хранения фермерской продукции:

- температура;
- состояние кондиционера;
- состояние шлагбаума;
- влажность;
- выделение газов в процентах:
 - * аргон;
 - * азот;
 - * гелий;
 - * водород;
 - * углекислый газ;
 - * угарный газ;
 - * кислород;
- масса продукции;
- кратность воздухообмена.

На складе установлены исполнительные устройства, позволяющие включать или выключать кондиционер. 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”,
5. Создайте геозону склада
6. Реализуйте автоматы:
 - 6.1 Автомат, контролирующий температуру внутри грузовика/склада. Обеспечивает включение системы кондиционирования при повышении температуры до 10 градусов Цельсия и последующее 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 критического уровня о низком качестве вождения.

Этап 3. Работа с API

1. После того, как фермер передал свой груз водителю, основная потребность фермера - отслеживать состояние переданного груза. Для этого разработайте интерактивную визуализацию для фермера по отслеживанию параметров продукции в процессе перевозки и хранения. Отображение необходимых показателей реализуйте в Grafana на нескольких панелях одного дашборда. Панели могут быть скомбинированы на ваше усмотрение.

2. Описание основных панелей визуализации:

- Time series

Панель с графиками изменения температуры внутри кузова грузовика и склада (2 графика разного цвета на одной панели).

- Time series

Панель с графиками изменения влажности внутри кузова грузовика и склада (2 графика разного цвета на одной панели).

- Radial Gauge

Панель с круговой шкалой текущей скорости грузовика. Шкалу разделите на три смысловых диапазона: нормальная скорость, допустимая скорость, превышение скорости. Для каждого диапазона задайте цвета и численные критерии на свое усмотрение.

- Bar gauge

Панель с гистограммой текущих значений выделяющихся газов внутри кузова.

- Bar gauge

Панель с гистограммой текущих значений выделяющихся газов внутри склада.

Примечание: Цвета, используемые для того или иного газа на данной гистограмме должны соответствовать цветам на гистограмме для газов внутри кузова.

Описание дополнительных панелей визуализации:

- Stat

Панель с текстовым значением текущего статуса грузовика: "На склад", "На ферму", "Стоянка-полный", "Стоянка-пустой".

- State timeline

Панель с горизонтальной полосой дискретного изменения статуса грузовика с течением времени.

2. В автомате, спроектированном в пункте 6.2 этапа 2, при входе и выходе из геозоны добавьте действия для отправки запросов на вызов вебхука, отправляющего в Telegram соответствующие уведомления: “Вы попали в геозону склада, шлагбаум открыт” и “Вы покинули геозону склада, шлагбаум закрыт”.

****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:

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 данных на одном из объектов
5. Создайте геозону для использования в сценарии контроля геозоны
6. 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 критического уровня при наступлении конкретных событий

Этап 3. Работа с API

Разработайте интерактивную визуализацию по отслеживанию параметров устройств.

Отображение необходимых показателей реализуйте в Grafana на нескольких панелях одного дашборда. Панели могут быть скомбинированы на ваше усмотрение.

Основные панели визуализации:

- Time series

Панель с графиками изменения некоторого параметра подвижного устройства и идентичного параметра стационарного устройства (2 графика разного цвета на одной панели).

- Time series

Панель с графиками изменения другого параметра подвижного устройства и идентичного параметра стационарного устройства (2 графика разного цвета на одной панели).

- Radial Gauge

Панель с круговой шкалой параметра одного из устройств. Шкалу разделите на три смысловых диапазона, обозначающих нормальное значение, допустимое значение и превышение предела. Для каждого диапазона задайте цвета и численные критерии на свое усмотрение.

- Bar gauge

Панель с гистограммой некоторых текущих значений подвижного устройства.

- Bar gauge

Панель с гистограммой некоторых текущих значений подвижного устройства.

Дополнительные панели визуализации:

- Stat

Панель с текстовым значением некоторого строкового параметра одного из устройств.

- State timeline

Панель с горизонтальной полосой дискретного изменения некоторого строкового параметра одного из устройств с течением времени.

2. В автомате, спроектированном в пункте 6.2 этапа 2, при входе и выходе из геозоны добавьте действия для отправки запросов на вызов вебхука, отправляющего в Telegram соответствующие уведомления: “Вы попали в геозону, ...” и “Вы покинули геозону, ...”.

2 Защита проектных работ

Цели занятия:

защитить проект и получить рекомендации экспертов.

Краткое содержание:

презентация проектов перед комиссией;
вопросы и комментарии по проектам.