

Министерство науки и высшего образования РФ

Томский государственный университет
систем управления и радиоэлектроники

О.В. Пехов

**ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ИНТЕРНЕТА
ВЕЩЕЙ**

Учебно-методическое пособие
для студентов направлений подготовки
10.00.00 Информационная безопасность

Томск
2022

УДК 004.056
ББК 32.973.26-018.2
П 64

Рецензент:

Давыдова Е.М., доцент кафедры комплексной информационной безопасности электронно-вычислительных систем ТУСУР, канд. техн. наук

Пехов, Олег Валерьевич

П 64 Информационная безопасность интернета вещей: учебно-методическое пособие / О.В. Пехов. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2022. – 34 с.

Настоящее учебно-методическое пособие содержит описания самостоятельных работ и требования к курсовой работе по дисциплине «Информационная безопасность интернета вещей» для направлений подготовки, входящих в укрупненную группу специальностей и направлений 10.00.00 Информационная безопасность.

УДК 004.056
ББК 32.973.26-018.2

© Конев А.А., Якимук А.Ю. 2022
© Томск. гос. ун-т систем упр. и радиоэлектроники, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Самостоятельная работа №1	
Предварительная работа над Кейсом 1	5
Самостоятельная работа №2	
Языки разметки данных	13
Самостоятельная работа №3	
Программа с Моск-объектами	20
Самостоятельная работа №4	
Предварительная работа над Кейсом 3	24
Самостоятельная работа №5	
бLoWPAN	26
Курсовая работа	29
Литература	34

ВВЕДЕНИЕ

Целью преподавания дисциплины является формирование теоретических и практических навыков по разработке надежных, качественных систем на базе IoT устройств с применением современных технологий программирования.

Задачи изучения дисциплины:

- формирование и развитие теоретических знаний основных методов программирования;
- получение практической подготовки в области выбора и применения технологии программирования для задач автоматизации обработки информации.

Самостоятельная работа №1

Предварительная работа над Кейсом 1

1. Введение

Эта работа рекомендуется для самостоятельного выполнения (дома, перед занятием).

Задания содержат работу с требованиями заказчика, предполагается создание проекта устройства и изучение его компонентов. Для их выполнения не требуется какого-либо оборудования, за исключением компьютера с выходом в Интернет. От вас потребуется поиск информации, выполнение набросков, анализ, аргументация своей позиции.

Дайте ответы на предложенные задания в письменном/печатном виде. По итогам выполнения заданий будет проведено общее обсуждение для того, чтобы совместно определить оптимальное решение.

2. Изучение требований

Точные требования, которым должен удовлетворять склад, изложены в Приказе Минздравсоцразвития РФ от 23 августа 2010 г. N 706н "Об утверждении Правил хранения лекарственных средств". Прочитайте его текст. Найдите в этом тексте один пункт и один абзац, которые интересны вам как разработчику. Выпишите подробности ТЗ, которые вы узнали из этого текста:

1. Минимальная частота снятия показаний датчики
2. Срок хранения логов
3. Минимальная площадь склада

3. Структурная схема

Нарисуйте структурную схему системы. В схеме должны фигурировать следующие сущности: продукция на

складе, система мониторинга (в составе: датчики, сервер, интерфейс пользователя), сам пользователь.

Ответьте на вопросы:

1. В каком направлении двигаются данные? Нужно ли передавать что-то конечным устройствам, или они только отправляют данные? Нарисуйте на диаграмме направление данных.

2. Какие варианты соединения сущностей вы можете предложить? Надпишите сверху, какой канал связи (проводной/беспроводной) оптимален в каждом случае,

3. Как можно организовать питание устройств? Дорисуйте линии питания.

4. Сравнительный анализ датчиков

Для решения задачи понадобятся датчики, измеряющие температуру и влажность, Датчик – это, как правило, компактное устройство со стандартным интерфейсом взаимодействия: I2C, UART или, самый простой вариант, GPIO.

Поскольку интерфейсы у датчиков разные, обычно датчик используется в паре с микроконтроллером, который собирает информацию с датчика, обрабатывает её (например, нормирует) и отправляет уже в унифицированной форме далее. Наконец, чтобы получить информацию с датчика, необходима некоторая технология передачи данных. Это может быть проводной или беспроводной интерфейс.

Удобно, если датчик, микроконтроллер и приёмопередатчик совмещаются друг с другом по принципу конструктора. Тогда можно быстро тестировать различные конфигурации решения. Некоторые компании выпускают такого рода обучающие наборы.

Предлагается использовать наборы российской компании-партнёра Unwired Devices, поскольку они просты

в освоении, в то же время предоставляют возможность для более глубоких экспериментов: исходные коды программы микроконтроллера есть в свободном доступе.

Ничто не мешает использовать и любые другие аппаратные средства для решения подобных задач.

- GPS
- Внешний датчик температуры (-55...125.13 °С) (на базе микросхемы LM75)
- Четыре внешних датчика температуры (-50...+150+0.7 °С) (на базе микросхемы LMT01)
- 6-осевой гироскоп (на базе микросхемы LSM6DS3)
- Датчик температуры и влажности воздуха (на базе микросхемы SHT21)
- Пироэлектрический датчик движения
- Датчик барометрического давления (на базе микросхемы LPS331)
- Датчик внешней освещённости (0...65535 лк) (на базе микросхемы OPT3001)
- Датчик температуры, влажности и давления воздуха (на базе микросхемы BME280)

Чтобы выбрать подходящий датчик для задачи, потребуется прочитать документацию на них. У разных датчиков – разные возможности и ограничения. Если есть цель заниматься IoT профессионально, то однократным изучением всего ассортимента обойтись не получится: инженер в области IoT должен пристально следить за новинками рынка.

Чтобы совершить обоснованный выбор датчика, выполните следующее задание: найдите в Интернете документацию на те четыре датчика из списка, которые могут быть полезны для решения задачи (измерение температуры и влажности). Подумайте о том, какие

характеристики датчиков важны в данной задаче. Вычлените их из технического описания датчиков. Самые существенные характеристики, как правило, написаны на первой же странице документации, Если какие-либо важные характеристики не отражены в документации, найдите их в Интернете на сайтах магазинов или производителей.

Составьте сравнительную таблицу характеристик датчиков. Слева должны быть названия микросхем датчиков, а сверху – выбранные характеристики (рис. 1).



Рис. 1. Вид сравнительной таблицы

Напишите 3 закономерности, которые вы заметили в характеристиках сходных датчиков. Например: «Более точные датчики, как правило...» или «Датчики с низким энергопотреблением чаще...»

Для выполнения задачи нам могут потребоваться данные об актуальных ценах. Важно, что нам не стоит рассматривать самые денежные варианты с отправкой компонентов с AliExpress и подобных сайтов, поскольку там возможна только поштучная покупка. Поэтому рассматривайте варианты российских магазинов:

- [http:// shogrunwds.com](http://shogrunwds.com) – магазин Unwired Devices – там можно посмотреть готовые конечные устройства, но учтите, что датчик отдельно стоит гораздо дешевле

- <http://www.compel.ru> – «Компэл», магазин электронных компонентов с оптовыми пенами
- <https://www.chipdipu> – «Чип и Дип», магазин с высокими ценами, но подходит в тех ситуациях, когда компонент нужен срочно
- <http://www.clectronstnik.ru> – «Электроник», крупный магазин, известный также как «Терразлектроника»

На основании таблицы выберите те датчики, которые войдут и ваше решение.

5. Выбор технологии передачи данных

Технология передачи данных между устройствами может быть как проводная, так и беспроводная: Wi-Fi, ZigBee, LoRa, и так далее. Забегая вперёд, использовать предлагается именно LoRa. Но необходимо оценить и другие варианты, чтобы понять логику выбора.

Изучите табличку о том, какую дальность связи могут обеспечить различные технологии беспроводной связи.

Таблица 1

Дальность связи в помещении

Технология	Дальность связи в помещении	Комментарий
<i>1</i>	<i>2</i>	<i>3</i>
Wi-Fi	20 метров	Источник - Википедия
ZigBee	15 метров	Источник - Википедия
LoRa	1500 метров	Источник – форум The Things Network

Основываясь на информации из открытых источников (например, Википедии), нарисуйте типовую топологию каждой из сетей с указанием расстояния между узлами.

Почему не подходит вариант с проводным соединением? Назовите одну, самую главную причину.

Назовите плюсы и минусы использования Wi-Fi в данной задаче. Если сложно, то шире - в сфере "Интернета вещей". Не менее трех пунктов.

6. Подсчет стоимости решения

Таблица о технологиях беспроводной связи дополняется информацией о стоимости конечных устройств без учета датчиков и прочей периферии, но с учётом микросхемы связи и программируемого микроконтроллера.

Таблица 2

Дальность связи в помещении и цена

Технология	Дальность связи в помещении и	Цена конечного устройства	Цена базовой станции	Комментарий
1	2	3	4	5
Wi-Fi	20 метров	240	1200	Взяты цена самого популярного WiFi-чипа ESP8266 и цена роутера потребительского класса
ZigBee	15 метров	1560	1560	В случае с ZigBee нет явного выделения базовой станции в силу топологии сети.
LoRa	1500 метров	1850	4650	Цены взяты из каталога Unwired Devices.

Некоторые допущения:

- Допустим, что склад имеет площадь 30 000 квадратных метров (такие склады существуют); допустим, что размеры склада 300 на 100 метров.

- Количество датчиков на складе зависит от ассортимента склада, а точнее, от того, сколько на нем хранится уязвимых для перепадов температуры и влажности лекарств: инсулин может замерзнуть, лекарственные травы - отсыреть. Но для простоты будем считать, что необходимо установить 500 датчиков.

- Прокладка электропроводки для снабжения Wi-Fi роутеров потребует услуг электрика: для примера, электрификация дома с 50 розетками будет стоить около 150 000 рублей.

Исходя из таблицы и рисунка топологии, составленного вами ранее, посчитайте стоимость решений.

По итогам, оцените, насколько велика разница стоимости оснащения склада устройствами той или иной технологии. Как бы вы охарактеризовали решения по параметрам цена/качество?

7. Изучение существующих решений

Сравним получившееся решение с уже имеющимися на рынке. Вообще говоря, это следует делать на раннем этапе изучения задачи, но в данном случае целью было самостоятельно прийти к пониманию некоторых ключевых моментов.

- Решение от Sensor Tools
- Решение от Validation Center
- Решение от Testo

Перечислите три новых улучшения системы, которые подсказали вам эти статьи.

Перечислите три недостатка описанных в статьях решений.

8. Развитие пользовательского интерфейса

С учетом рассмотренных примеров, дополните структурную схему устройства (рис. 2): что бы вы в неё добавили? Подумайте о модулях ввода/вывода, которые бы сделали пользовательский интерфейс удобнее.



Рис. 2. Структурная схема устройства

Самостоятельная работа №2

Языки разметки данных

1. Введение

В прошлом кейсе мы уже поработали с данными, которые приходят с конечных устройств. Данные были в формате JSON, но это всего лишь один из возможных форматов. Предлагается взглянуть на самые популярные языки разметки данных, и оценить их с точки зрения наших задач.

2. XML

XML (eXtensible Markup Language) – это расширяемый язык разметки, который, в отличие от HTML, не фиксирует набор тегов и структуру документов. Оба языка связаны с SGML. SGML – это стандартизированный метаязык разметки, то есть язык, предназначенный для описания других языков разметки (рис. 1). XML является подмножеством SGML, определяя более узкий метаязык, на базе которого можно создавать свои форматы передачи и хранения данных. Таким образом, XML не специфицирует теги и формат документа. В то же время, HTML (до версии HTML 5.0) является конкретным применением языка SGML, жёстко определяя используемые в разметке теги и формат тела документа.

HTML в силу области своего применения и исторических причин является не таким строгим языком, как XML. XML делает синтаксис SGML более строгим: он не допускает незакрытых или пустых открывающих или закрывающих тегов. Благодаря этому XML нашёл применение как формат передачи и хранения данных. Например, если вы разархивируете обычным zip-архиватором документы формата .docx или .odt, то увидите структуру папок, и в них – XML-документы.

Данные, приходящие от устройства «Интернета вещей», в XML выглядели бы так, как на рис. 2.

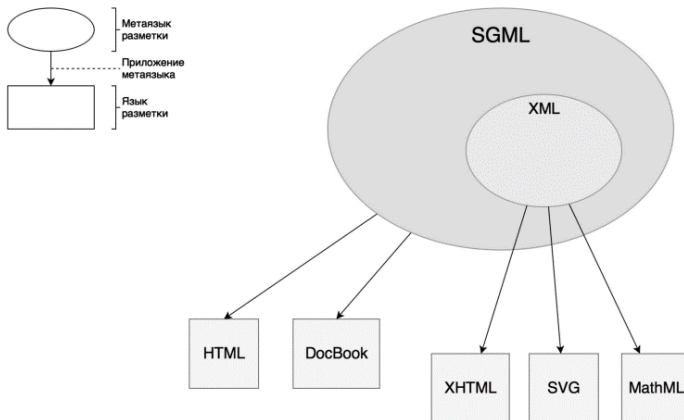


Рис. 1. SGML

```

<!--?xml version="1.0" encoding="UTF-8" ?-->
<data>
  <temperature>26.2</temperature>
  <humidity>22.2</humidity>
  <pressure>977</pressure>
</data>
<status>
  <deveui>807B85902000019A</deveui>
  <rsssi>-76</rsssi>
  <temperature>-10</temperature>
  <battery>3350</battery>
  <date>2017-03-06T14:11:57.228909Z</date>
</status>

```

Рис. 2. XML-документ

3. JSON

JSON расшифровывается как JavaScript Object Notation, но применимость этого формата обмена данными вышла далеко за пределы JavaScript, и в настоящий момент

существуют парсеры для практически всех основных языков программирования. Формат появился из необходимости взаимодействия сервера и браузера без использования Flash-плагинов или Java-апплетов. Изначально JSON позиционировался как строгое подмножество синтаксиса JavaScript, однако в строках он допускал использование невалидных в строках JavaScript символов. Впоследствии формат был закреплён стандартами RFC 7159 и ECMA-404, при этом стандарт ECMA описывает только допустимый синтаксис, в то время как RFC покрывает некоторые аспекты безопасности и совместимости.

О каких объектах идет речь в названии JSON? О тех же самых, о которых вы знаете из курсов ООП (в C++, к примеру). В языке JavaScript можно с лёгкостью конвертировать объект в строку и обратно, и это делается очень естественным образом. Благодаря такому механизму, очень легко перевести объект в строку JSON для отсылки, к примеру, по Интернету. Такой процесс ещё называется сериализация, а обратный процесс (превращение строки в полноценный объект с заполненными полями) – десериализация. Сериализация позволяет обмениваться структурами данных программам, которые написаны на разных языках и физически находятся в разных системах. Стоит отметить, что иногда может встречаться понятие маршаллинг. Очень часто эти два термина (сериализация и маршаллинг) синонимичны, однако следует учитывать, что в общем случае сериализация является частным случаем маршаллинга.

Пример того, как выглядят данные в формате JSON представлен на рис. 3.

```
{
  "data":{
    "temperature":26.2,
    "humidity":22.2,
    "pressure":977
  },
  "status":{
    "devEUI":"807B85902000019A",
    "rssi":-76,
    "temperature":-10,
    "battery":3350,
    "date":"2017-03-06T14:11:57.228909Z"
  }
}
```

Рис. 3. JSON-документ

4. YAML

YAML расшифровывается как Yet Another Markup Language. Это переводится как «Ещё один язык разметки» - язык создавался в 2001 году, уже после того, как был стандартизирован XML, и появился практически одновременно с JSON. В настоящий момент расшифровка поменялась на рекурсивный акроним «YAML Ain't Markup Language» («YAML - не язык разметки»), что подчеркивает его предназначение быть посредником в обмене данными между программами, а не языком разметки документов для

пользователя, как, например, Markdown или HTML. С течением времени стало явно прослеживаться основное применение данному формату: конфигурационные файлы.

Начиная с версии 1.2 YAML является надмножеством JSON. Однако, в отличие от последнего, он не стандартизирован официально: не существует RFC, касающиеся данного формата, или, к примеру, стандарта ECMA. В то же время, официальные спецификации, доступные на сайте YAML, занимают примерно 80 страниц машинописного текста, что, конечно, говорит в том числе и о сложности данного формата.

Обычно YAML не используется в качестве формата сериализации и передачи данных по сети.

Тот же файл будет выглядеть следующим образом, как на рис. 4.

```
---
data:
  temperature: 26.2
  humidity: 22.2
  pressure: 977
status:
  devEUI: 807B85902000019A
  rssi: -76
  temperature: -10
  battery: 3350
  date: '2017-03-06T14:11:57.228909Z'
```

Рис. 4. YAML-документ

5. Выбор формата разметки

1. Посчитайте объем каждого сообщения в символах (XML, JSON, YAML).

2. Оцените каждый из языков разметки с точки зрения удобства восприятия человеком:

- Наиболее удобочитаемый
- Наименее удобочитаемый

Подводя итог, какой язык разметки данных лучше всего подходит для задач «Интернета вещей»? Какой язык бы выбрали вы? Напишите, почему.

6. Разбор JSON

В прошлой задаче вы делали разбор JSON-выражения, пришедшего с конечного устройства. Задачей было выделить показатели датчиков из строки. Вы могли делать это разными способами: посимвольный разбор с множеством условий или циклов; поиском подстроки; используя регулярные выражения; средствами готового JSON-парсера.

Последний вариант самый надёжный, однако он требовал от вас дополнительного знания о том, что существуют готовые парсеры JSON. Все же остальные варианты плохи тем, что в случае изменения структуры сообщения, вам пришлось бы существенно изменять код.

Для дальнейшей работы убедитесь, что вы умеете пользоваться парсером JSON на привычном вам языке программирования.

Задачей будет:

1. Сделать задачу из Кейса 1 с использованием JSON-парсера, если вы ещё этого не сделали.

2. Обработать те случаи, когда в данных нет необходимого ключа. В этих случаях программа должна не завершать работу, а корректно печатать сообщение об ошибке. Такая ситуация может возникнуть, к примеру, если поменялся формат данных или если возник «баг» на стороне устройства и оно передало неверные данные.

Для единообразия, возьмем за основу такую модельную ситуацию: производитель оборудования выпустил обновление, и теперь ключевые слова `humidity` и `temperature`

переименованы в `humid` и `temp`. Ваша программа должна выдавать ошибку «Нет необходимого ключа в данных».

Решите задачу и проверьте своё решение при помощи генератора данных из Кейса 1; самостоятельно модифицируйте его код.

Самостоятельная работа №3

Программа с Моск-объектами

1. Введение

В домашнем задании вам предлагается написать и отладить черновую версию программы СКУД, которую на занятии вы будете отлаживать с реальным оборудованием.

Для выполнения этого задания и для успешной работы на практикуме вам нужно знать о следующих аспектах Python:

- Строки
- Дата/время
- Массивы
- Словари
- Запись в файл
- Использование библиотек

Если язык Python для вас новый и у вас возникнут трудности, изучите материалы Недели 1 и 2 онлайн-курса "Программирование на Python".

2. Блок-схема

Прежде, чем писать программу, нарисуйте блок-схему. Перечитайте формулировку задания Кейса.

У вас имеется: считыватель электронного ключа, реле для управления замком, RGB-лампочка. Подумайте о том, как задействовать цветовую индикацию для удобства пользователя.

3. Альфа-версия программы

Вы уже умеете общаться с MQTT-сервером при помощи программы на Python, используя библиотеку Paho. Само собой разумеется, что так можно взаимодействовать и с локальным сервером Mosquitto, который изучался в практикуме по MQTT. Начните с того, что, используя стандартный пример для Paho, подключитесь к локальному

серверу (localhost). В окне с сервером вы увидите, что появился новый клиент (рис. 1).

```
volkova_ta@volkova-ubuntu:~$ mosquitto -p 1884
1505899485: mosquitto version 1.4.8 (build date Mon, 26 Jun 2017 09:31:02 +0100)
starting
1505899485: Using default config.
1505899485: Opening ipv4 listen socket on port 1884.
1505899485: Opening ipv6 listen socket on port 1884.
1505899494: New connection from ::1 on port 1884.
1505899494: New client connected from ::1 as 73c3718f-460e-4a72-9f9e-4398fc2921a
0 (c1, k60).
```

Рис. 1. Подключение нового клиента

Затем попробуйте переслать сообщение своей программе командой *mosquitto pub*. Начните с того, что сами имитируйте объект управления - то есть отправляйте ключи с клавиатуры через *mosquitto pub* и читайте о действиях объекта на экране монитора. В первой версии программы можно просто обозначить печатью (print) все действия устройства (открыть замок, зажечь лампочку). Например, программа может выводить вам: "green", "red", "lock is closed" и другие сообщения. Главное - воспроизвести намеченную вами блок-схему.

Схема взаимодействия сервера, программы и вас (в роли объекта управления) будет как на рис. 2.

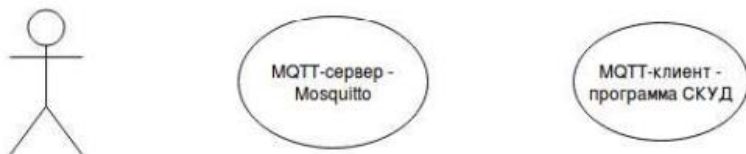


Рис. 2. Схема взаимодействия

Дополните эту схему стрелками, показывающими, кто с кем взаимодействует и каким образом (подпишите).

Начните писать программу с простых функций, буквально по одной строчке на каждую:

1. Открыть замок
2. Закрыть замок
3. Включить лампочку заданного цвета

4. Выделить ключ из пришедшего сообщения

5. Проверить наличие ключа в базе

Когда они готовы, вы можете скомбинировать эти функции в программу согласно вашей блок-схеме. Помните, что согласно философии языка Python простое – лучше, чем сложное! Первая версия программы должна быть максимально проста и абсолютно читаема.

4. Моск-объект

Особенность программирования систем, взаимодействующих с внешним миром, такова: устройства внешнего мира бывают недоступны, либо тестирование занимает очень много времени. Поэтому полезно бывает сразу вводить тоск-объекты (имитации реальных объектов) и тестировать программу с ними.

Последнее задание – сделать полноценный моск-объект, чтобы не вводить команды с клавиатуры, а сделать так, чтобы сообщения о ключах отправляла программа-имитатор с некоторой периодичностью. Или, например, чтобы ваша система СКУД уже не печатала на экране цвет лампочки, а вместо этого отправляла команду в топик grio для включения лампочки того или иного цвета, а дальше программа-имитатор будет печатать этот цвет на экран. Прделанная работа очень поможет вам дальше, когда вы будете дополнять основную программу СКУД новым функционалом.

Это будет ещё одна программа с использованием Python. И схема взаимодействия уже будет такая как на рис. 4.

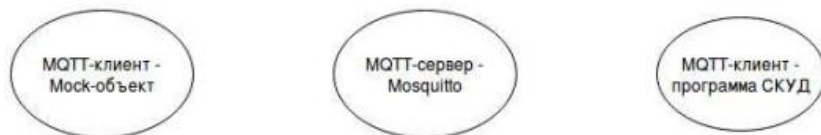


Рис. 4. Схема взаимодействия

Как и в предыдущем задании, нарисуйте стрелочки, показывающие взаимодействия, и подпишите их.

Самостоятельная работа №4

Предварительная работа над Кейсом 3

1. Изучение существующего решения

Прочитайте статью о том, как адаптивное освещение офиса используется в компании КРОК. Также прочитайте комментарии к статье. Используя сведения из статьи и свои собственные догадки, подумайте, какие проблемы могут появиться при создании и эксплуатации такого рода системы.

Перечислите как минимум три фактора риска.

2. Анализ экономической эффективности

Решает ли подобного рода система обозначенную заказчиком проблему? Необходимо заранее в этом убедиться, чтобы в дальнейшем избежать обвинений в свой адрес.

Посчитайте экономию электроэнергии, взяв за пример графики от КРОКа (рис. 1).

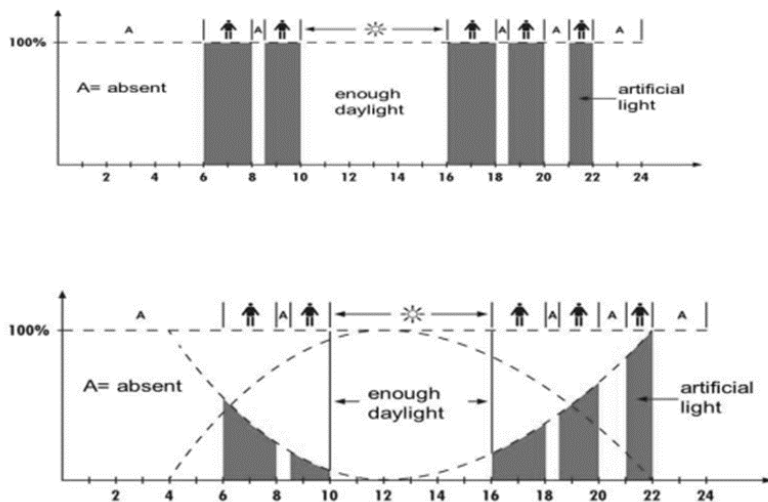


Рис. 1. Графики КРОКа

Параболу освещенности возьмите как на графике КРОКа: рассвет в 4.00, самый яркий солнечный свет в 12.00, полное наступление темноты в 22.00. Стоимость 1 кВт-часа в Москве: 4.34 рубля за квт/ч (дневной тариф). Будем считать, что один модуль связи стоит, как и в прошлой задаче, 155, также 10\$ - стоимость периферийных устройств (датчиков, кнопок, исполнительных устройств), и примерно 150 000 рублей стоит разработка и внедрение системы.

Посчитайте, исходя из этих данных, за сколько лет инвестиции окупят себя (срок окупаемости). Задача решается средствами школьной математики.

Подумайте, исходя из каких соображений компания может внедрять у себя такую систему.

Самостоятельная работа №5

6LoWPAN

1. Определение этапов работы и схема интерфейса

Этот проект – система адаптивного освещения – достаточно несложен, и понятно, что систем такого вида уже существует немало. Части задачи легко отделимы друг от друга. Прочитайте задание кейса и подумайте:

- Подумайте, на какие этапы можно было бы разбить эту задачу?

- В каком порядке было бы удобно делать систему?

- Что войдет в первый прототип, который вы покажете заказчику через 2 недели?

Количество этапов может быть любым, но вряд ли их будет больше 7.

Перечислите эти этапы.

1.1. Интерфейс пользователя

В наборе Unwired Kit есть плата UMBK-4BTN, которая позволяет считывать нажатия кнопок. Протестируйте её работу. Через консоль активируйте драйвер 4btn. Будут появляться уведомления о том, что кнопка с определенным номером нажата или отпущена.

Каким должно быть взаимодействие системы с пользователем? Впишите в таблицу функционал кнопок. В системе обязательно должен быть автоматический режим и возможность выхода из него. Имейте в виду, что это именно кнопки, а не тумблеры, без удержания.

Представьте ответ в виде таблицы (рис. 1)

Кнопка	Действие
1	
2	
3	
4	

Рис. 1. Вид таблицы для ответа

А теперь переделайте схему с учетом того, что у вас исчезли уведомления о том, что кнопка отпущена, но

появились уведомления о длительном нажатии кнопки. Как изменится интерфейс? Ответ представьте в виде таблицы.

2. Класс устройств LoWPAN на примере LoRa

Прежде, чем изучать протокол 6LoWPAN, стоит изучить общую концепцию под названием LoWPAN. Это сокращение расшифровывается как Low power Wireless Personal Area Networks. Таким образом, это не название конкретного протокола, а целый класс устройств со своими особенностями.

Прочитайте начало документа:

RFC4919: "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals".

В разделе Overview указаны 10 основных характеристик LoWPAN - подумайте, можно ли распространить эти характеристики на всю сферу "Интернета вещей", и если да, то какие. В левом столбце выпишите краткое название признака (его суть), а в правом - то, как этот признак реализуется в нашем устройстве на базе LoRa. Для примера, мы заполнили первую строку (табл. 1).

Таблица 1

Признаки

Номер признака	Признак	Устройство на базе LoRa
1	2	3
1	Маленький размер пакета (максимум 127 байт)	Максимум 126 байт
2		
3		

Используя получившуюся таблицу, ответьте на вопрос: относится ли наше устройство на базе LoRa к классу устройств LoWPAN. Если возникнут затруднения, то читайте официальную документацию.

3. Новый протокол связи – 6LoWPAN

Прочитайте две статьи о 6LoWPAN. Это полезно, чтобы узнать историю, контекст возникновения и некоторые ключевые особенности технологии:

1. Статья "6LoWPAN - взгляд на беспроводные IP-сети от Texas Instruments":

- Введение
- Первый раздел "6LoWPAN: краткая характеристика"

2. Перевод статьи инженера Texas Instruments "Раскрываем тайны 6LoWPAN" об устройстве 6LoWPAN.

- Введение
- Первый раздел "Архитектура 6LoWPAN-сети"
- Второй раздел "Краткое описание системного стека"

По итогам прочтения, заполните табличку сравнения с LoRa (рис. 2).

Технология	Топология	Дальность связи	Частота	Длина адреса в сети
LoRa				
6LoWPan				

Рис. 2. Вид таблицы «Сравнение 6LoWPAN и LoRa»

В чем смысл названия 6LoWPan? Какая ещё технология из изученных ранее обладает такой же топологией?

Курсовая работа

1. Введение

Курсовая работа завершает изучение дисциплины «Информационная безопасность Интернета вещей».

Цель курсовой работы реализовать собственную систему «Интернета вещей».

Перечень тем работы формируется студентами самостоятельно и утверждается преподавателем. Выполнение работы возможно группой студентов, с четким разграничением обязанностей между исполнителями проекта. Исполнитель работы планирует и определяет сценарий функционирования разрабатываемой системы.

2. Требования к содержанию курсовой работы

Разрабатываемая система Интернета вещей может состоять из списка следующих частей, в котором обязательные позиции выделены:

1. конечное устройство в составе:
 - микроконтроллер;
 - сенсорика;
 - исполнительные устройства;
 - питание (от батарей или от сети);
2. связь (скорее всего беспроводная, кроме отдельных случаев, где проводная наиболее оправдана). Протокол связи - LoRa, WiFi, ZigBee, Bluetooth, GSM - тот, который наиболее подходит для данной задачи;
3. серверная часть;
4. облачный сервис (по необходимости, в целях масштабирования)
5. уровень пользователя:

– пользовательский интерфейс
(веб/desktop/мобильное приложение);

– аналитика (в том числе - с применением методов машинного обучения)

В пояснительной записке к проекту, помимо системы, должны быть проработаны следующие вопросы:

1. стоимость;
2. энергопотребление;
3. польза/прибыль;
4. целевая аудитория;
5. обзор аналогов;
6. прототип;
7. результаты апробации;
8. научная новизна (опционально).

Наиболее важный компонент в работе - это именно польза, которую приносит такая система. Рекомендуется разрабатывать применимые к нашей жизни системы, протестированные в реальных условиях. К примеру, если делается счетчик пешеходов для решения задач урбанистики - то хорошо бы выйти на реальный перекресток и провести натурные испытания. Если делается система фотофиниша для спортивных соревнований - то прийти на настоящие соревнования и испытать систему там. И так далее.

Для аттестации студента по курсу, на уровне вуза - научная новизна не принципиальна. Студент может добросовестно решить уже достаточно известную задачу, например, сделать умный магазин по образцу Amazon или станцию экомониторинга, и если он справится, то он заслуживает сертификат выпускника IoT Академии. Ситуация меняется, если мы говорим о межвузовском конкурсе: здесь жюри будет ожидать увидеть "изюминку", то есть свежую идею: особенно это касается номинации "Бизнес

и общество", где оценивается в первую очередь именно идея, и во вторую - техническая реализация.

Процесс проектированию согласуется с календарным графиком. Документы работы должны выполняться в соответствии с ограничениями и нормами стандартов ЕСКД (единой системы конструкторской документации) и вузовского стандарта оформления студенческих работ ОС ТУСУР 01-2013.

3. Требования к оформлению и исполнению курсовой работы

1. Курсовая работа выполняется в соответствии с требованиями ГОСТ 2.105-95 ЕСКД и ОС ТУСУР 01-2013. Допускается основные надписи для текстовых документов (формы 2 и 2а ГОСТ 2.104-2006 ЕСКД) выполнять на первых двух листах текста.

2. Титульный лист выполняется по прилагаемому образцу (Приложение А), техническое задание – по форме, приведенной в Приложении Б.

3. Пояснительная записка работа должна быть исполнена грамотно и аккуратно в соответствии с требованиями ОС ТУСУР 01-2013. по составу и форме.

4. В тексте обязательны ссылки на использованные источники.

5. Все исходные и расчетные данные должны сводиться в таблицы для удобства расчетов и проверки.

4. Календарный график выполнения работы

Таблица 1

График выполнения работы

Наименование этапов работы	Отчетность	Срок	Объем	Итого
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
Выдача и согласование ТЗ	Оформленное ТЗ	1.03	0	0
Анализ ТЗ. Поиск и выбор источников. Изучение принципа работы и функциональных возможностей	Лит. источники. Структура устройства (краткое описание).	15.04	25%	25%
Разработка структурной схемы. Выбор элементной базы. Составление эскизной эл. схемы. Выявление нестандартных узлов.	Структ. схема. Приближенная схема.	30.04	50%	75%
Составление графических документов и ПЗ. Корректировка по замечаниям.	Сдача работы на проверку. Защита работы	15.05-20.05	25%	100%

5. Рейтинг по этапам работы

Таблица 2

Рейтинг по этапам работы

Промежуточный контроль (отчетность студента по этапам работы)	Сроки этапов	Баллы нарастающи м итогом
<i>1</i>	<i>2</i>	<i>3</i>
Согласование функциональных возможностей и ТЗ на разработку с преподавателем. Подбор источников и прототипа. Изучение принципа работы устройства.	15.04	0-40
Разработка структурной схемы модифицированного устройства. Выбор элементной базы. Составление эскизной эл. схемы. Выявление необходимых нестандартных узлов.	30.04	40-80
Написание и оформление ПЗ, Сдача работы на проверку. Корректировка замечаний, сделанных преподавателем при проверке. Успешная защита курсовой работы. При оценке работы учитываются неординарные решения, качество оформления ПЗ, моделирование устройства в ППП.	15.05- 20.08	80-100

Литература

1. Петин, В. А. Проекты с использованием контроллера Arduino / В. А. Петин. – 2-е изд., перераб. и доп. – СПб.: БХВПетербург, 2015. – 448 с
2. Лафоре, Роберт. Объектно-ориентированное программирование в C++ / Роберт Лафоре. – 4-е изд. – Питер, 2004. – 928 с.
3. Интернет вещей: видео, аудио, коммутация. Научно-популярная литература / А. Суомалайнен ; ред. Д. А. Мовчан. - Электрон. текстовые дан. - М. : ДМК Пресс, 2019. - on-line : рис., схемы. - Библиогр.: с. 118-120.
4. Ли, П. Архитектура интернета вещей / П. Ли ; перевод с английского М. А. Райтман. — Москва : ДМК Пресс, 2019. — 454 с.
5. Бирюков, А. А. Умные устройства безопасности на микроконтроллерах Atmel / А. А. Бирюков. — Москва : ДМК Пресс, 2017. — 162 с.