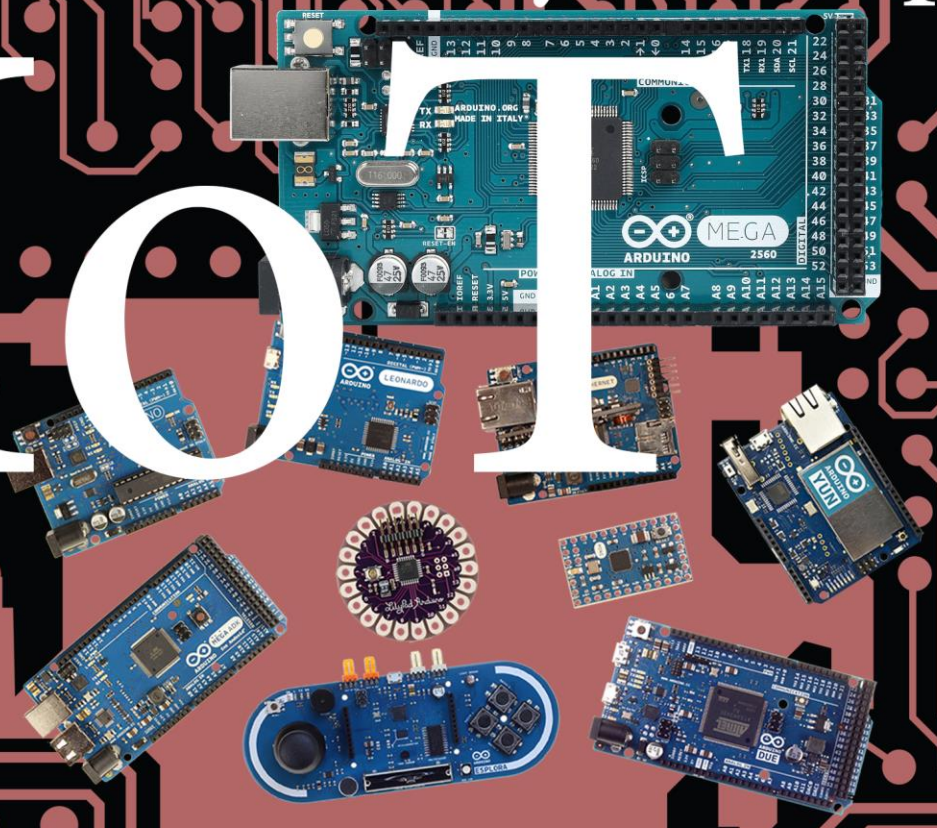
«Интернет вещей. Межмашинное взаимодействие.
Программирование в компьютерных системах и сетях.»

Вячеслав Мунистер

IoT



Министерство образования и науки
Донецкой народной Республики.

На правах рукописи

Действующая (вторая) редакция
на 1.11.2020

Мунистер Вячеслав Денисович

АВТОПРЕЗЕНТАЦИЯ
ПРОЕКТА ИНТЕГРИРОВАННОГО УЧЕБНОГО КУРСА

«Интернет вещей. Межмашинное взаимодействие. Программирование в компьютерных системах и сетях.»

ДЛЯ ПРОФЕССИОНАЛЬНЫХ МОДУЛЕЙ И ДИСЦИПЛИН В РАМКАХ МЕТОДОЛОГИИ ПРЕПОДАВАНИЯ ПРОФЕССИОНАЛЬНОГО ЦИКЛА ДЛЯ НАПРАВЛЕНИЙ ПОДГОТОВКИ УКРУПНЕННОЙ ГРУППЫ:
09.00.00 «Информатика и вычислительная техника» ПРОГРАММ СПО И ВПО

Донецк - 2020

ЦЕЛЕВАЯ КАРТА

Образовательный процесс нуждается в непрерывном процессе обновления, актуализации методологии и применения усвоенных знаний к новым фактам, как со стороны преподавателя, так и со стороны обучающегося. В частности, в условиях острой конкурентной борьбы, в сложных экономических реалиях, с которым сталкивается наша молодая республика.

Выпуск максимально готовых к жизни и профессиональной карьере молодых специалистов - извечная проблема всякой образовательной организации. Степень готовности – вовсе не эфемерное и непостижимое понятие, а вполне себе измеряемый показатель.

Текущее демографическое падение, приведет в скором времени к большой интенсивности столкновений между существующими образовательными учреждениями: в 2029 г. для учреждений СПО и в 2031 г. для ВПО. Именно в обозначенное время система подготовки профессиональных кадров столкнется с катастрофическими недоборами учащихся, так как мы войдем в период, когда вырастут те, кто родился в самый трагический в новой истории, период активных боевых сражений (2014-2015 гг,) с последующим оттоком населения, в том числе наиболее активной ее социальной группы – молодежи (на тот исторический фазис).

Приведу факты:

В 2012 году в Донецке родилось 8 тысяч 342 малыша, что на 441 ребенка больше чем в 2011 году. В 2013-ом – 8402 (прим. - Главстат).

В «В 2015 году на всей (!) территории Донецкой Народной Республики родилось примерно 9300 детей». В 2014 году в Донецке родилось 6 843 ребенка. в 2015 году – 3 050 детей; в 2016 году – 4 355 детей; в 2017 году – 4 532 ребенка.

Необходимо принимать постепенные меры, направленные на вывод образовательной деятельности на качественно новый уровень, демонстрируя яркие отличия для референтной группы будущих студентов, работодателей (источников возможных целевых заказов), поднимая престиж организации до значительных высот, прежде всего, чтоб максимально подготовиться к этому, стремительно приближающемуся периоду, ведь известно, что жизнь учителя без ученика представить довольно трудно.

Данная автопрезентация предназначена именно для демонстрации возможностей актуализации одного из компонентов деятельности – **учебно-методологической составляющей**: создания модульных интегрированных учебных курсов.

ГЕНЕЗИС ИНТЕГРАЦИИ В ПЕДАГОГИЧЕСКОМ ПРОЦЕССЕ

Под интеграцией в педагогическом процессе исследователи понимают одну из сторон процесса развития, связанную с объединением в целое ранее разрозненных частей. Этот процесс может проходить как в рамках уже сложившейся системы, так в рамках новой системы.

Сущность процесса интеграции — качественные преобразования внутри каждого элемента, входящего в систему.

Проблемы интеграции в педагогике рассматриваются в разных аспектах в трудах многих исследователей. В работах В. В. Краевского, А. В. Петровского, Н. Ф. Талызиной рассматриваются вопросы интеграции педагогики с другими науками. Г. Д. Глейзер и В. С. Леднёв раскрывают пути интеграции в содержании образования.

В работах Л. И. Новиковой и В. А. Караковского раскрыты проблемы интеграции воспитательных воздействий на ребёнка. Интеграция в организации обучения рассматривается в трудах С. М. Гапеенкова и Г. Ф. Федорец. Названными и другими учёными определены методологические основы интеграции в педагогике: философская концепция о ведущей роли деятельности в развитии учащегося; положение о системном и целостном подходе к педагогическим явлениям; психологические теории о взаимосвязи процессов образования и развития, **интегративный подход.**

Принцип интеграции предполагает взаимосвязь всех компонентов процесса обучения, всех элементов системы, связь между системами. Он является ведущим при разработке целеполагания, определения содержания обучения, его форм и методов.

Интегративный подход означает реализацию принципа интеграции в любом компоненте педагогического процесса, обеспечивает целостность и системность педагогического процесса.

Интегративные процессы являются процессами качественного преобразования отдельных элементов системы или всей системы. Многие исследования в отечественной дидактике и в теории воспитания опираются на выше перечисленные положения при разработке конкретных путей совершенствования образовательного процесса.

ДИДАКТИКА – ФУНДАМЕНТ И ТЕЗАУРУС РАЗРАБАТЫВАЕМОГО КУРСА

Дидактика (от греч. didakitos - поучающий, didasko - изучающий) - отрасль педагогической науки, раскрывающая теоретические основы образования и обучения в их наиболее общем виде. Дидактика выявляет закономерности, принципы обучения, задачи, содержание образования, формы и методы преподавания и учения, стимулирования и контроля в учебном процессе, характерные для всех учебных предметов, на всех возрастных этапах обучения.

Дидактика изучает закономерности и специфику образования и обучения в общеобразовательной, профессиональной, средней специальной, высшей школе и других системах обучения. Объект дидактики - процесс обучения.

Предмет - вскрытие закономерностей процесса обучения, изучение системы отношений: ученик - учебный материал, учитель-ученик, ученик-другие ученики.

Впервые это слово появилось в сочинениях немецкого педагога Вольфгана Ратке (1571-1635) для обозначения искусства обучения. Аналогичным образом, как «универсальное искусство обучения всех всему», трактовал дидактику и Я.А. Коменский. В начале XIX века немецкий педагог И.Ф. Гербарт придал дидактике статус целостной и непротиворечивой теории воспитывающего обучения. Неизменными со времен Ратке остаются и основные задачи дидактики - разработка проблем: чему учить, как учить; современная наука интенсивно исследует и такие проблемы: когда, где, кого и зачем учить, как учить эффективно.

Взаимосвязи между основными дидактическими категориями как структурными компонентами целостного дидактического процесса изображены на **рисунке 1**:

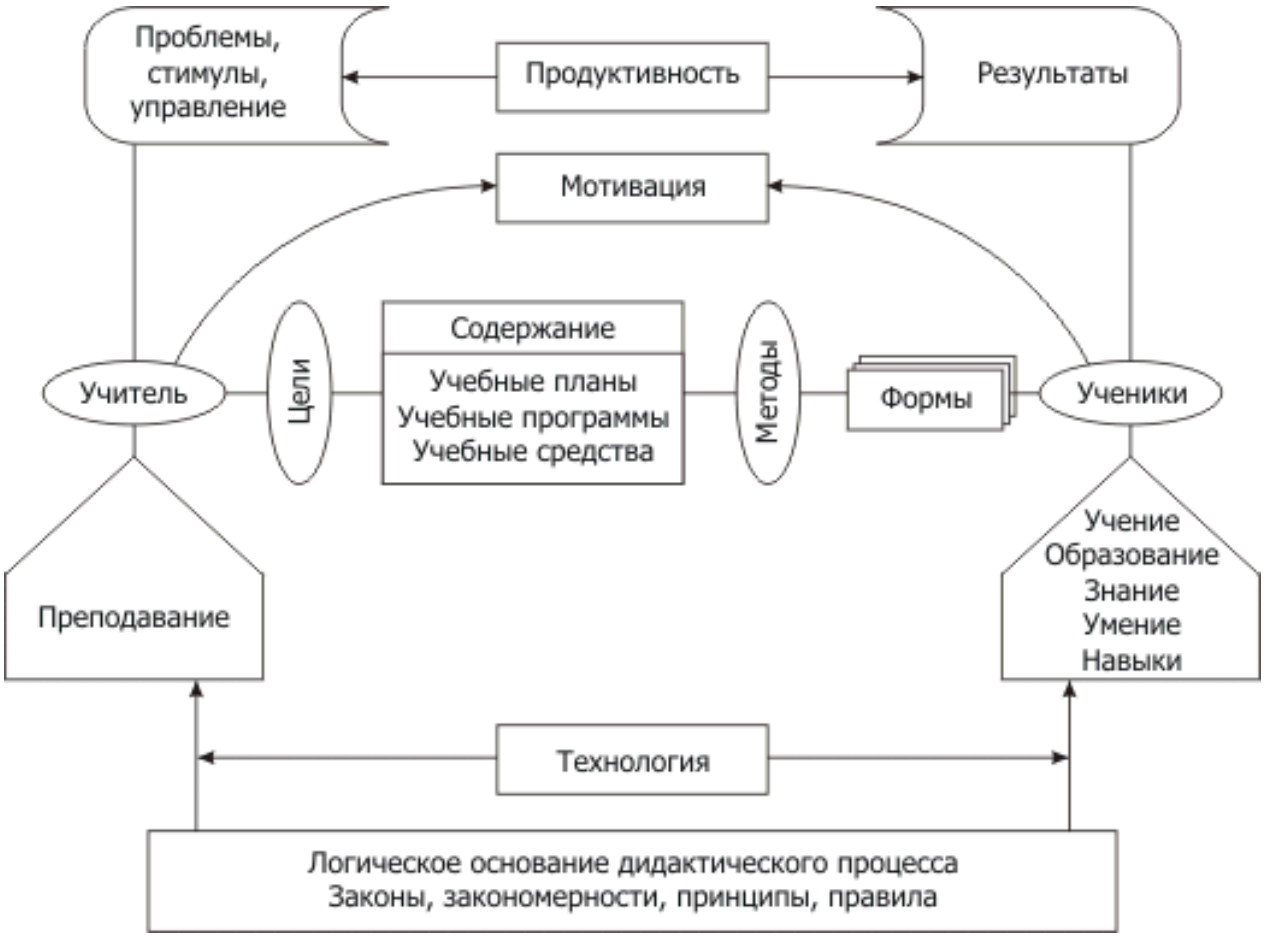


Рисунок 1 – Онтология основных дидактических категорий.

Основные категории дидактики: обучение, преподавание, учение, образование, знания, умения, навыки, а также цель, содержание, организация, виды, формы, методы, средства, результаты (продукты) обучения. В последнее время статус основных дидактических категорий предлагается присвоить понятиям «дидактическая система» и «технология обучения».

Обучение - целенаправленный процесс взаимодействия педагога с учащимися, их совместной деятельности, в ходе которой осуществляется образование, воспитание и развитие. Общение, в ходе которого происходит накопление знаний, умений, навыков, развитие, образование. Обучение делится на преподавание и учение.

Преподавание - упорядоченная деятельность педагога, направленная на реализацию цели обучения, обеспечение информирования, осознания и практического применения знаний.

Учение - процесс деятельности учащегося по освоению знаний, умений и навыков, опыта, творчества и эмоционально-ценностных отношений, в ходе которого возникают новые формы поведения и деятельности, применяются ранее приобретенные знания и навыки.

Образование - процесс становления культурного человека и результат обучения, система приобретенных знаний, умений, навыков, способов мышления, мировоззрения, нравственности и общей культуры.

Знания - информация, которая может преобразовываться и использоваться, совокупность информационных идей человека, в которых выражается теоретическое овладение этим предметом.

Навыки - умения, доведенные до автоматизма.

Целью в дидактике является образ конечного результата, то, к чему стремится процесс обучения.

Под содержанием обучения понимается система знаний, умений, навыков, способов деятельности и мышления, ценностных отношений, которыми учащиеся овладевают в процессе обучения.

Задачи дидактики:

Научное описание и объяснение процесса обучения и условий его реализации.

Усовершенствование процесса обучения и создание новых технологий обучения.

Как теория обучения и образования дидактика разрабатывает следующие проблемы:

она определяет педагогические основы содержания образования;

исследует сущность, закономерности и принципы обучения, а также пути повышения его развивающего влияния на учащихся;

изучает закономерности учебно-познавательной деятельности учащихся и **пути ее активизации в процессе обучения;**

разрабатывает систему общепедагогических методов обучения и условия их наиболее эффективного применения;

определяет и совершенствует организационные формы учебной работы в образовательно-воспитательных учреждениях.

Дидактика как наука изучает закономерности, действующие в сфере ее предмета, анализирует зависимости, обуславливающие ход и результаты процесса обучение, определяет методы, организационные средства, обеспечивающие осуществление запланированных целей и задач.

НАЗВАНИЕ ИНТЕГРИРОВАННОГО КУРСА:

«Интернет вещей. Межмашинное взаимодействие. Программирование в компьютерных системах и сетях.»

СОСТАВИТЕЛЬ: Мунистер В.Д.

СОДЕРЖАНИЕ КУРСА

Тип модуля	Название издания
1. *Учебно-теоретическое издание (Хрестоматия)*	«Компьютерные сети. IoT и межмашинное взаимодействие»
2. *Учебно-практическое издание (Практикум)*	«Дом, который построил сам себя. Сетевой практикум. IoT. »
3. *Учебно-практическое издание (Практикум)*	«Визуальное программирование (FBD) для микропроцессорных систем и IoT»

КЛЮЧЕВЫЕ СЛОВА

Тэги:	Модуль
Компьютерные сети, межмашинное взаимодействие, персональные компьютерные системы, корпоративные сети, беспроводные самоорганизующиеся сети, окружающий интеллект, средства идентификации.	«Компьютерные сети. IoT и межмашинное взаимодействие» **1 (100 стр.)**
Сетевые модели, основы сетевого планирования (экономика), циклы проектирования корпоративных компьютерных систем,	«Дом, который построил сам себя. Сетевой практикум. IoT. » **2 (154 стр)**

Моделирование информационных потоков данных в больших сетях, теория конечных автоматов для синтеза дискретных устройств (датчики, актуаторы). *Оценка рисков информационной безопасности, алгоритмы Дейкстра и Краскаля для Интернета Вещей, Концепция «Интернет всего», Графические конфигураторы (визуальное программирование)*	«Дом, который построил сам себя. Сетевой практикум. IoT. » (продолжение)
Визуальное программирование, Структурное программирование, Интегрированные среды разработки для микропроцессорных систем, Радиочастотная идентификация, Программные интерфейсы программирования, Программируемые логические контроллеры	«Визуальное программирование (FBD) для микропроцессорных систем и IoT» **3** **(104 стр.)**

СТРУКТУРА КУРСА: МОДУЛЬНАЯ

40%	**60%**
Практическая составляющая (практические работы) **13 ед.**	Теоретическая составляющая, Интер-отклик программа (интерактивное взаимодействие с источником знаний)

Данное соотношение применительно для реализации образовательных перспективных программ **прикладного бакалавриата** (как для **СПО**, так и для ВПО)

СТЕК ИЗУЧАЕМЫХ (ЗАТРАГИВАЕМЫХ) ТЕХНОЛОГИЙ И ПРОГРАММНЫХ СРЕДСТВ (СРЕД):

ПО/ (Программное обеспечение)	**Cisco Packet Tracer 7.2** – Симулятор сети передачи данных, выпускаемый фирмой Cisco Systems. Позволяет делать работоспособные модели сети, настраивать маршрутизаторы и коммутаторы, взаимодействовать между несколькими пользователями. (версия с поддержкой IoT) (бесплатно распространяемое ПО с регистрацией) **Arduino IDE 1.6** — интегрированная среда разработки для Windows, MacOS и Linux, разработанная на C и C++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей (ESP/STM) (бесплатно распространяемое ПО) **FLProg** — представляет собой IDE для визуального программирования микропроцессорных систем. (бесплатно распространяемое ПО)

	RemoteXY — это система разработки и использования мобильных графических интерфейсов для управления контроллерами со смартфона или планшета. В состав системы входят: Редактор мобильных графических интерфейсов для контроллеров, размещенный на сайте remotexy.com Мобильное приложение RemoteXY, позволяющее подключаться к контроллеру и отображать графические интерфейсы. (бесплатно распространяемое ПО)
ЯП (Языки программирования)	C/C++, FBD, JavaScript
(etc)	HTML5 + CSS3, Cisco IOS, NODE RED

КАРТА ПРИМЕНЕНИЯ (ПРИМЕР)

Профессиональный модуль	**Модуль курса**
МДК.01.01. Организация, принципы построения и функционирования компьютерных сетей	1,2
МДК.01.02. Математический аппарат для построения компьютерных сетей	1,2,3
МДК.02.01. Программное обеспечение компьютерных сетей	1, 2,3
МДК.02.02. Организация администрирования компьютерных систем	2
МДК.03.02. Безопасность функционирования информационных систем	2

НЕОБХОДИМЫЕ АППАРАТНЫЕ СРЕДСТВА
(ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ)

<table>
<tr><th>Средство</th><th>Название/описание или назначение</th><th>Имеется в наличии</th></tr>
<tr><td>Arduino UNO</td><td>Микропроцессорная система (опционально)</td><td>2</td></tr>
<tr><td>Arduino NANO</td><td>Микропроцессорная система (опционально)</td><td>1</td></tr>
<tr><td>ESP 8266 «Witty Cloud»</td><td>Микропроцессорная система с поддержкой Wi-Fi.
<table><tr><td>Поддерживаемые сети</td><td>802.11 b/g/n</td></tr><tr><td>Частота микроконтроллера</td><td>80 МГц</td></tr><tr><td>Скорость UART</td><td>115200</td></tr><tr><td>Рабочее напряжение</td><td>3.3 В</td></tr><tr><td>Напряжение питания</td><td>3.7-12 В</td></tr><tr><td>Максимальный ток потребления (при передаче)</td><td>240 мА</td></tr><tr><td>Размеры</td><td>30 x 31x 18 мм</td></tr></table>Многофункциональный контроллер, который имеет большие возможности по реализациям вне рамок данного практикума. Главная особенность – IEE 802.11 контроллер и удобство программирования и питания (данная плата состоит из двух модулей).</td><td>3</td></tr>
<tr><td>Any</td><td>Wi-Fi router
с поддержкой 802.11 b/g/n</td><td>0</td></tr>
</table>

Данное техническое оборудование не требует дополнительных требований к эксплуатации и охране труда, просто в эксплуатации, обладает низкой стоимостью и доступностью к приобретению.

На данных микропроцессорных системах нового поколения без ущерба к функциональности можно оттачивать практические и теоретические знания на междисциплинарных курсах и учебных практиках, лабораторных работах.

АННОТАЦИЯ: МОДУЛЬ ПЕРВЫЙ
РАЗРАБОТКА С 11.10.2020 ПО 17.01.2020

Учебно-теоретическое издание *(Хрестоматия)*	«Компьютерные сети. IoT и межмашинное взаимодействие»

Данное издание предназначено для восполнения недостающих теоретических знаний по дисциплинам, междисциплинарным курсам, связанных с принципами организации межсетевого взаимодействия, архитектуры информационных систем: («Организация, принципы построения и функционирования компьютерных систем», «Математический аппарат для построения компьютерных систем», «Дизайн архитектуры распределенных сетей», «Инфокоммуникационные системы и сети», «Информационные технологии», «Внедрение и поддержка программного обеспечения компьютерных систем», «Компьютерные и телекоммуникационные сети») студентов, осваивающих программы среднего и высшего профессионального обучения.

Получение недостающих знаний – серьезный инструмент общего процесса актуализации: поддержания практических и теоретических знаний индивидуума в актуальном состоянии, т.е. приведение их в соответствие с состоянием отображаемых объектов предметной области будущего специалиста в сфере информационных технологий и вычислительной техники. Я отождествляю вкладываемый смысловой контекст данной книги с понятиями необходимого и достаточного условий — известных вам по изучаемым математическим дисциплинам.

Учебное издание «Компьютерные сети. IoT и межмашинное взаимодействие» и выступает в роли достаточного условия процесса снятия информационной энтропии, касающегося профессионального ориентирования студентов вышеперечисленного профиля подготовки.

Издание содержит в себе ряд перспективных т.н. «Рабочих предложений» (RFC) от IETF, IEEE, и иных организаций, занимающихся сертификацией технологий в рассматриваемой области человеческой деятельности, а также статей с верифицированных иностранных и отечественных научных и публицистических изданий.

Часть информации подана в явном компрессированном виде, и неявном – полноценном.

Все это достигается за счет внедрения на страницы издания печатных QR-кодов с ссылками на те или иные интернет-ресурсы. Таким образом, книга получает куда более расширенное функционально-интерактивное предназначение.

АННОТАЦИЯ: МОДУЛЬ ВТОРОЙ
РАЗРАБОТКА С 21.02.2020 ПО 09.08.2020

Учебно-практическое издание (Практикум)	«Дом, который построил сам себя. Сетевой практикум. IoT. »

Это издание предназначено для получения практических навыков и умений посредством организации выполнения индивидуальных заданий в рамках проведения учебных практик как рекомендованного решения использования в контексте конкретных междисциплинарных курсов, связанных с принципами организации межсетевого взаимодействия, архитектуры информационных систем, в частности, в рамках платформы «Internet of things» и визуальным программированием, дидактикой преподавания математических дисциплин, вопросами информационной безопасности.

Подразумевается, что проведение практикума необходимо рассматривать неразрывно, в конкретно определенной последовательности. Однако, строгая итерационная и типизационная составляющая в выполнении элементов практикума не выражена рамочно. Элементарным преобразователем неделимой части (отраженной в виде главы) практикума может быть самостоятельное проведение лабораторной работы при условии вовлечения в рассматриваемый контекст учащегося (осваивающего данный элемент программы).

Также необходимо понимать правило достаточного условия для возможности допуска к заданиям, овладение базового уровня теоретических знаний по двум группам дисциплин: математического и общего естественно-научного учебного цикла (группа I) и общепрофессионального цикла (группа II):

Группа I: «Элементы высшей математики», «Дискретная математика».

К II группе относятся такие дисциплины как: «Теория информации и кодирования», «Технологии защиты информации», «Технические средства информатизации», «Основы программирования и баз данных», «Архитектура аппаратных средств», «Проектирование цифровых устройств», «Программное обеспечение компьютерных сетей», «Теория принятия решений», «Разработка мобильных и встроенных специализированных систем», «Компьютерная логика», «Разработка прикладных решений на базе современных платформ», «Системный анализ», «Технологии реинжиниринга и бизнес-инжиниринга».

Общая структура данного издания представлена таким образом, чтоб раскрыть наиболее значимые элементы содержания методологии этих двух групп. Целью разработки практикума было создание аддитивного эффекта от преподаваемых дисциплин в ключе расширения ассоциативного ряда у студентов (обучающихся) создания, придания интерактивности.

Но прежде всего – для получения четкого понимания, для чего нужны те или иные инструменты (компоненты) в осознанной профессиональной деятельности.

В основе итерационной последовательности неделимых элементов практикума (можно использовать обозначение – блок, контейнер) лежат базовые догмы класса прикладных методов управления проектами (в рамках общепринятого обозначения эти методы имеют общее название - «Сетевое планирование и управление», использующееся в бизнесе несколько десятков лет) в упрощенном формализованном виде, обеспечивающим приведение к планированию, и даже осуществить анализ сроков выполнения (ранних и поздних) нереализованных частей проектов; что позволяет увязать выполнение различных работ и процессов во времени, составить сетевой график, получив прогноз общей продолжительности реализации всего проекта.

Общая задумка концепции практикума с применением методов сетевого планирования и управления несет себе и одну далеко идущую цель (идеал), не совсем тривиальную задачу – создать интродукцию (первичное введение) в системный анализ – научный метод познания, представляющий собой последовательность действий по установлению структурных связей между переменными или постоянными элементами исследуемой системы, который, собственно и опирается на комплекс общенаучных, экспериментальных, естественнонаучных, статистических, математических методов и прямо востребован среди тех, кто планирует освоить магистерскую программу. А также аспирантскому сообществу — будущей научной интеллигенции.

Вернемся, однако, к структурному представлению, обращая внимание на предназначение вышеперечисленных неделимых блоков (контейнеров) практикума и описанию целевой карты по каждой из позиции:

I. «Актуатор как конечный автомат» — синтез «Теории алгоритмов», «Теории цифровых автоматов», «Дискретной математики» с усвоением важнейшего термина из учебного издания «Компьютерные сети. IoT & межмашинное взаимодействие» – актуатора (исполнительного устройства). Этот термин взят из теории автоматического управления.

Под исполнительным устройством понимают устройство, передающее воздействие с управляющего устройства на объект управления.

Иногда он рассматривается как составная часть объекта управления. Управляющим устройством может быть любая динамическая система (в учебно-теоретическом издании «Компьютерные сети. IoT и межмашинное взаимодействие» этот вопрос освещен в главах VII-IX.

Перед данным блоком содержатся теоретические сведения, оформленные в виде главы с названием «Дискретная математика в IoT: теория автоматов». В практическом боксе происходит последовательная пошаговая процедура алгоритмизации решения конкретной (прикладной) задачи – создание конечного автомата цифрового оконечного устройства с последующим исполнением в булевом базисе.

II. «Создание сетевой модели инфраструктуры IoT» – включает в себя работу по осуществлению сетевого планирования и управления: моделирование и визуализация сетевой модели разработки индивидуального технического задания – графического представление проекта. Данный метод планирования позволяет найти минимальные сроки завершения проекта на этапе отдельных работ в теоретизации решения задача, а также определить множество критических работ, увеличение продолжительности выполнения любой из которых приводит к увеличению времени выполнения всего проекта.

Теоретической основой для этого элемента практикума являются главы: "Основы сетевого планирования и управления" "Обзор цикла проектирования ККС""Разработка структуры IoT-платформы".

III. «Создание Smart Campus» — гайдлайн IoT на практике в рамках симулятора Cisco Packet Tracer. В этом модуле обучающиеся вплотную познакомятся с сетевой топологией «умного» кампуса, конфигурацией IoT-сети, типовой реализацией Smart-Industrial и Blockly custom software for IoT Simulations. Данный модуль практикума является теоретически-прикладным, основанным на мануале «IoT Simulations with Cisco Packet Tracer» магистра Университета прикладных наук Метрополия (г. Хельсинки, Финляндия) Andrea Finardi.

Индивидуальные задания как таковые отсутствуют, так как подразумевается вынесение их в виде отдельного модуля – закладывая, тем самым основы проведения хакатона, семинара, где сами участники будут формировать техническое здание и проводить тренинг в Cisco Packet Tracer.

IV. «Zero trust в IoT» — позволяет раскрыть специфику Internet of Things с кардинально новой стороны, доводя до студента общую проблематику информационной безопасности этой платформы. Уделено внимание понятию «нулевого доверия», IoT DDoS-атакам. Студенты научатся проводить оценку рисков и угроз информационной безопасности сетям.

Контейнер базируется на содержании глав «Оценка рисков информационной безопасности», «Определение топологии в самоподобных множествах», в которых, происходит ознакомление с политиками

информационной безопасности, началами фрактальной геометрии в упрощенном кратком виде (с целью ознакомления, назидательной дидактики преподавания смежных дисциплин).

V. «СЛАУ в IoT-инфраструктуре» — модуль, который дает возможность использовать потенциал линейной алгебры, в частности, решение систем линейных уравнений от трёх переменных, которые определяются как набор плоскостей, и, в свою очередь, в абстрактном виде могут выступать в виде геометрической реализации математического множества устройств одного из трех типовых элементов IoT-системы: координатора, маршрутизатора, конечного устройства (как это было описано в учебно-теоретическом пособии данного интегрированного курса).

Данный модуль позволит определить некоторый баланс между устройствами этих трех типов, дать важные ориентиры, касающиеся проблематики избыточности низкопотребляющих устройств, работающих в полудуплексном режиме, и сетей двух семейств: NB-IoT (Narrow Band Internet of Things) и ZigBee.

Содержание данного модуля предназначено не столь для практического применения в реальной жизни (так как на данный момент не существует инструментов мат. аппарата, позволяющих строго определить нужные количественные соотношения), а столь для фактического закрепления результатов освоения путем создания лучшего ассоциативного эффекта, учитывая огромное значение линейной алгебры в информационных системах и технологиях.

VI. «Алгоритмы Дейкстра и Краскала для IoT» завершает математический цикл практикума. Теория графов финализирует представление об создании инфраструктуры, подтверждая первично полученные умения по теории графов в контейнере «Создание сетевой модели инфраструктуры IoT».

VII. «Создание первого IoT-приложения» - наиболее значимый в практическом представлении модуль, являющийся частью одного из курсов IBM Developer. Состоит из двух частей: Первая - создание IoT-приложения, превращающего смартфон в IoT-датчик (актуатор) при помощи Bluemix - — публично-облачной платформы, разработанной IBM. Платформа поддерживает несколько языков программирования и сред разработки, а также инструментов в стиле DevOps для построения, выполнения, развёртывания и управления приложениями в облаке.

Вторая – «Создания IoT-приложения pingGo». В этой части описано, как настроить рабочее пространство IBM Bluemix, как создать демонстрационное приложение с помощью инструмента Node-RED и как успешно отослать SMS-сообщения из этого приложения на свой мобильный телефон с помощью сервиса Twilio.

Node-RED — это flow-based инструмент, созданный для визуального программирования, разработанный IBM для совмещения вместе: устройств, API, онлайн-сервисов и IoT, Он работает на Node.JS, и был разработан для работы на относительно малопроизводительных микропроцессорных системах, таких как: Raspberry Pi. BeagleBone Black. Arduino, которые давно используются в образовательной сфере.

С учётом озвученных факторов Node-RED удобно использовать на шлюзах между различными сетями устройств интернета вещей функционирующих на собственных, как правило, более простых протоколах и традиционным интернетом, построенных на TCP/IP, UDP. В этом случае он позволит более оптимально использовать свободные ресурсы шлюза, работающего, как правило, на Linux. Графический конфигуратор, базирующийся на принципе «вершина» и «ребро», значительно упрощает разработку и повышает наглядность функционирования IoT-системы.

Теоретической основой для завершающего элемента практикума является глава «Node Red - графический конфигуратор для Интернета Вещей» и содержимое самого контейнера в пошаговом формате.

Приложения, оформленные в конце издания, дополняют и уясняют вопросы курса в сфере IoT и несут на себе справочно-обзорную роль.

Таким образом, целями практикума являются: закрепление и систематизация полученных в ходе лекционных курсов знаний и развитие и практических умений и навыков студентов, по налаживанию аналитической и организационной работы технического содержания в сфере организации корпоративных компьютерных сетей нового поколения и IoT-решений в организациях всех форм собственности, то есть, в местах будущего применения полученных знаний после освоенных программ среднего и высшего профессионального образования

АННОТАЦИЯ: МОДУЛЬ ТРЕТИЙ. РАЗРАБОТКА С 09.08.2020 ПО 09.11.2020

Учебно-практическое издание (Практикум)	«Визуальное программирование (FBD) для микропроцессорных систем и IoT»

Как говорил еще Конфуций: задача учителя — открывать новую перспективу размышлениям ученика.

Раскрытие перспективы современных, многофункциональных и доступных к применению в практической деятельности микропроцессорных систем может достигаться за счет вовлечения в результат удивительного сплава мысли в области программной инженерии – визуального программирования.

Визуальное программирование — способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста. Визуальное программирование часто представляют, как следующий этап развития текстовых языков программирования.

В последнее время визуальному программированию стали уделять больше внимания — в связи с развитием мобильных сенсорных устройств и средств, обеспечивающих Human-machine interface на уровне взаимодействия оператора с органами управления системы.

Визуальное программирование в основном используется для создания программ с графическим интерфейсом для операционных систем с графическим интерфейсом пользователя.

Среда визуального программирования позволяет написать Web-приложение для браузера; позволяет создать консольное приложение (программа без графического интерфейса и без вывода сообщений в консоль) для программирования микроконтроллеров, программируемых микросхем.

В целях создания интегрированного курса, связывающего между собой основы алгоритмизации и программирования (а также реверс-инжиниринга), электронику и электротехнику, был создан данный лабораторный практикум.

Надеюсь, что разрабатываемый мной практикум (модуль 3), базирующийся на концепции обучения посредством визуального программирования и встроенным программам интродукции в архитектуру аппаратных средств, широко используемых во встроенных компьютерных системах, помогут вам в профессиональной деятельности.

ВОЗМОЖНОСТЬ МОДУЛЬНОГО ИСПОЛЬЗОВАНИЯ В РАМКАХ ДОПОЛНЕНИЯ (МАТЕРИАЛОВ СЕМИНАРСКИХ ЗАНЯТИЙ/ САМОСТОЯТЕЛЬНОЙ РАБОТЫ) СОДЕРЖИМОГО КУРСА В ОБЩЕПРОФЕССИОНАЛЬНЫХ ДИСЦИПЛИН (ОП):

(ПРИМЕР)

ОК	**Модуль курса**
Технологии физического уровня передачи данных	1, 2
Основы программирования и баз данных	2, 3
Технические средства информатизации	2
Информационные технологии	1,2,3

Теория алгоритмов	2,3
Дискретная математика	2
Основы алгоритмизации и программирования	2,3
Компьютерные сети и телекоммуникации	1,2
Интеллектуальные информационные системы	1,2,3
Вычислительная техника	1
Основы информационной безопасности	2
Архитектура встраиваемых систем	1,2,3

СФЕРА ПРИМЕНЕНИЯ:

1. Прикладной бакалавриат — специальная методика обучения, стандарт знаний специалиста, при котором студенты высших учебных заведений получают полный набор знаний и навыков, необходимых для того, чтобы сразу, без дополнительных стажировок профсориентированно работать по специальности.

Эксперимент по созданию прикладного бакалавриата в образовательных учреждениях Российской Федерации введен Постановлением Правительства Российской Федерации от 19 августа 2009 г. N 667 **«О проведении эксперимента по созданию прикладного бакалавриата в образовательных учреждениях среднего профессионального и высшего профессионального образования».**

2. Основные профессиональные образовательные программы среднего профессионального образования **базовой подготовки** и программы **среднего профессионального образования углублённой подготовки.**

3. Основные профессиональные образовательные программы высшего профессионального образования **ОКУ «Академический Бакалавр».**

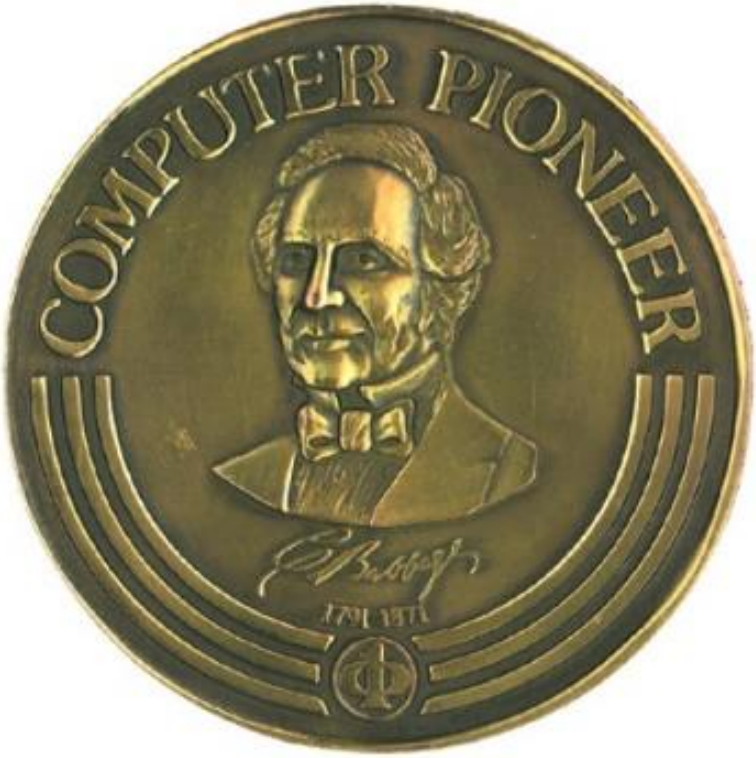
ПЕРСПЕКТИВЫ РАЗВИТИЯ КУРСА

Реализация	Использование
Создание методических указаний	к курсовым проектам по ранее не затронутым темам, но изучаемых в рамках: «Организация межсетевого взаимодействия для IoT» «Проектирование встраиваемых систем» «Проектирование информационных систем» «Проектирование безопасных экстранет сетей»
Создание лаборатории «Беспроводные технологии передачи информации» «Беспроводные самоорганизующиеся сети»	Реализация посредством привлечения аппаратных средств ESP8266/ESP32/LoraWan/Zigbee-совместимых устройств для демонстрации работы, изучения генерируемого траффика (и сниффиринга его посредством программных средств), обучение программированию устройств в рамках TCP/IP-стэка, создание предпосылок к введению новой тематики для дипломных проектов: «Разработка собственных прикладных протоколов обмена информацией» «Разработка технологии средств защиты информации беспроводных сетей»

Учебно-теоретическое издание

«Компьютерные сети. IoT и межмашинное взаимодействие»

Хрестоматия



2020 г.

GLÜCKSRITTER
MUNISTER

§ INSCRIPTUM

Вторая половина XX века подарила человечеству великое множество замечательных достижений в области цифровой электронной вычислительной техники - технической базы информационных технологий (ИТ). Благодаря появлению компьютеров информация, которой владеет человечество, стала своеобразным "сырьем" для производства множества "продуктов": новых знаний, управленческих решений, научных прогнозов, статистических сведений, всевозможных рекомендаций, заключений и т.д. Причем, в отличие от физического сырья (полезных ископаемых и др.), информация по мере использования не только не исчезает, а наоборот, пополняется новой, являя собой неисчерпаемую "сырьевую" базу интеллектуального труда.

Современными успехами компьютеризации и информатизации мировое сообщество обязано миллионам тружеников - ученым, инженерам, рабочим, создавшими новые поколения компьютеров, их программное обеспечение, мощные информационные сети.

Однако тех, кто закладывал фундамент компьютерной науки и техники, было не так много. На их долю выпало самое трудное - создать то, чего еще никогда не было. Среди них были ученые, инженеры и математики многих стран. Вторая мировая война и последовавшие за ней десятилетия "холодной" войны привели к разобщению ученых и засекречиванию работ, поскольку компьютеры (электронные вычислительные машины ЭВМ) - создавались прежде всего для военных целей.

Вследствие этих факторов имена создателей вычислительной техники и их творческий вклад были известны лишь узкому кругу специалистов.

За рубежом в странах Западной Европы и США этот пробел в литературе о становлении и развитии цифровой электронной вычислительной техники уже восполнен. Здесь появилось много книг, статей в периодических изданиях, созданы музеи с экспонатами ЭВМ первых поколений.

В странах бывшего Советского Союза этот процесс затянулся. Лишь в поздние девяностые и нулевые годы двадцать первого века, то есть, уже в наше время, началось рассекречивание многих ранее выполненных в данной отрасли работ и появилась возможность исследовать и оценить огромный творческий вклад ученых, инженеров, производственников в мировой процесс становления и развития вычислительной техники, информатики, ИТ.

И я хочу посвятить это учебно-теоретическое издание этим людям: родоначальникам компьютерных наук. Это необходимый шаг, так как многие из тех, кто будет упомянут, получили признание, выраженное в виде медали «Пионера компьютерной техники» (англ. Computer Pioneer Award) - самой престижной награды Компьютерного общества IEEE уже посмертно.

К сожалению, наш с вами Нобель или Пулитцер, стал вручаться относительно недавно – всего тридцать с небольшим лет назад. Да и к тому же, вручается он за выдающиеся достижения в компьютерных науках, с условием, что основной вклад должен был быть совершён более пятнадцати лет назад. Таким образом, медаль, выполненная из бронзы, на аверсе медали выполнен барельеф Чарльза Бэббиджа, является примерно тем, чем для художника признание – невероятной редкостью при жизни.

Я хочу начать с наших соотечественников, людей, говоривших на русском языке, сделавших так много, чтоб в конце концов стереть все границы и барьеры, стоящие на пути мгновенного и общедоступного общения между людьми, посредством информационных технологий. К сожалению, формат книги не позволит рассказать историю их жизни, трудовых и научных и даже боевых подвигов, голодных и бессонных дней, но не назвать их, я их не могу. Так как благодарен всем им своим основным призванием по жизни.

Эта книга посвящается: Лебедеву Сергею Алексеевичу, Ляпунову, Алексею Андреевичу, Глушкову Виктору Михайловичу, Лопато Георгию Павловичу, Столярову Геннадию Константиновичу, Никлаусу Вирту, Линусу Торвальду, Фридриху Бауэру, Питеру Науру, Джону Атаносову, Артуру Самуэлю, Маршиану Хоффу, Килби Джеку, Эриху Блоху, Кену Олсену, Рейнолду Джонсону, Дугласу Энгельбардту, Алану Перлису, Гради Бучу, Эдварду Фейгенбауму, Джину Бартику, Ирвину Джону Гуду, Кену Томпсону, Томасу Курцу, Джону Макарти, Айленду Сазерленду, Джеффри Чуан Чу, Барбаре Лисков, а также всем сотрудникам Института инженеров электротехники и электроники — IEEE и членам «Зала Славы Интернета».

Мунистер В.Д.

§ СОДЕРЖАНИЕ

Inscriptum стр. 22

I. Сетевой гайдлайн. стр. 25
I. Содержание курса. Система интер-отклика посредством QR. стр. 25
II. От ARPAnet до модели OSI/ISO. стр. 26

II. Сети: BAN, PAN, LAN, CAN, MAN. стр. 65
III. Информационные технологии и телекоммуникации. стр. 39

IV. Микроархитектура компьютерных сетей. стр. 46
I. Эталонный подход: Friend-to-friend и Peer-to-Peer обмен. стр. 46

V. Беспроводные сенсорные сети. стр. 70
I. Интеллектуальные системы на базе сенсорных сетей. стр. 70
II. Беспроводные самоорганизующиеся сети. стр. 76

VI. Архитектура Internet of things (IoT)
I. Средства и технологии передачи данных: IEEE 802.15, ZigBee. стр. 82
II. Средства идентификации, измерения, передачи данных LPWAN. стр. 89
III. Окружающий интеллект: платформа, технология, применение. стр. 93
IV. Актуаторы, айтрекеры – элементы сетей завтрашнего дня. стр. 99

VII. Cisco Packet Tracer. Добавление устройств IoT в сеть (л/р). стр.105
VIII. Модель межмашинного взаимодействия(M2M). стр. 114
IX. Организация межмашинного взаимодействия устройств сети с носимым айтрекером* стр.116

Список использованных источников. стр.117

Содержание курса. Система интер-отклика посредством QR.

Данное издание предназначено для восполнения недостающих теоретических знаний по дисциплинам, междисциплинарным курсам, связанных с принципами организации межсетевого взаимодействия, архитектуры информационных систем: («Организация, принципы построения и функционирования компьютерных систем», «Математический аппарат для построения компьютерных систем», «Дизайн архитектуры распределенных сетей», «Инфокоммуникационные системы и сети», «Информационные технологии», «Внедрение и поддержка программного обеспечения компьютерных систем», «Компьютерные и телекоммуникационные сети») студентов, осваивающих программы среднего и высшего профессионального обучения.

Получение недостающих знаний – серьезный инструмент общего процесса актуализации: поддержания практических и теоретических знаний индивидуума в актуальном состоянии, т.е. приведение их в соответствие с состоянием отображаемых объектов предметной области будущего специалиста в сфере информационных технологий и вычислительной техники. Я отождествляю вкладываемый смысловой контекст данной книги с понятиями необходимого и достаточного условий — известных вам по изучаемым математическим дисциплинам.

Учебное издание «Компьютерные сети. IoT и межмашинное взаимодействие» и выступает в роли достаточного условия процесса снятия информационной энтропии, касающегося профессионального ориентирования студентов вышеперечисленного профиля подготовки.

Издание содержит в себе ряд перспективных т.н. «Рабочих предложений» (RFC) от IETF, IEEE, и иных организаций, занимающихся сертификацией технологий в рассматриваемой области человеческой деятельности, а также статей с верифицированных иностранных и отечественных научных и публицистических изданий. Часть информации подана в явном компрессированном виде, и неявном – полноценном. Все это достигается за счет внедрения на страницы издания печатных QR-кодов с ссылками на те или иные интернет-ресурсы. Таким образом, книга получает куда более расширенное функционально-интерактивное предназначение.

Надеюсь, что тщательно подобранные, переработанное и адаптированные к чтению, материалы данного учебного курса (вместе с планируемым дополнением, выраженным в виде курса лабораторных работ в сетевом эмуляторе Cisco Packet Tracer) станут путеводной звездой для поколения новых инженеров – архитекторов Интернета завтрашнего дня.

От ARPANET до модели OSI/ISO.

Мы можем назвать имена создателей парового двигателя, самолёта или кинематографа. Однако в создании сети Интернет принимали участие множество блестящих учёных и коллективы целых университетов. Технология развивалась достаточно медленно, поэтому в разные годы вклад в становление «глобальной паутины» вносили самые разные люди. Как и большинство других, передовых для своего времени технологий, Интернет появился как военная разработка. Первые попытки создать беспроводное средство связи начались в самый разгар холодной войны. Руководство США было обеспокоено успехами СССР в освоении космоса. По мнению ряда американских военных специалистов, космические технологии сделали бы Советский Союз абсолютно неуязвимым в случае вооруженного столкновения. Поэтому сразу после успешного запуска советского «Спутника-1» в 1957 году, в Америке начались разработки новой системы для передачи данных. Все исследования велись под эгидой Министерства обороны США и держались в глубочайшем секрете. В создании новой технологии принимали участие технические кафедры лучших университетов страны.

В 1962 году сотрудник Массачусетского университета, по совместительству работавший в Управлении перспективными исследовательскими проектами при Министерстве обороны США (ARPA), — Джозеф Ликлайдер — предложил своё решение проблемы. Ликлайдер полагал, что осуществлять связь можно через компьютеры. Под его руководством в 1960-е годы началась работа над проектом, получившим название ARPANET. Планировалось, что сообщения в такой сети будут передаваться целиком, но подобная передача имела несколько серьёзных изъянов: невозможность взаимодействия большого количества пользователей, дороговизна, неэффективное использование пропускной способности сети, неспособность нормально функционировать при разрушении отдельных компонентов сети. Над устранением этих недостатков стал работать учёный из Калифорнийского университета — Пол Бэран. Итогом его работы стал новый способ передачи информации — коммутация пакетов. Фактически каждое сообщение разбивалось на несколько пакетов, каждый из которых шёл к адресату по своему каналу. Благодаря этому техническому решению, новая сеть передачи данных становилась практически неуязвимой.

В конце 1969 года состоялось историческое событие — по ARPANET было передано первое сообщение.

Сеанс связи осуществлялся между Калифорнийским и Стенфордским университетами и увенчался успехом только со второй попытки.

Для того чтобы передать на расстояние 640 км короткое слово «login», потребовалось полтора часа. На тот момент к сети было подключено всего 4 компьютера, расположенные в разных университетах Америки.

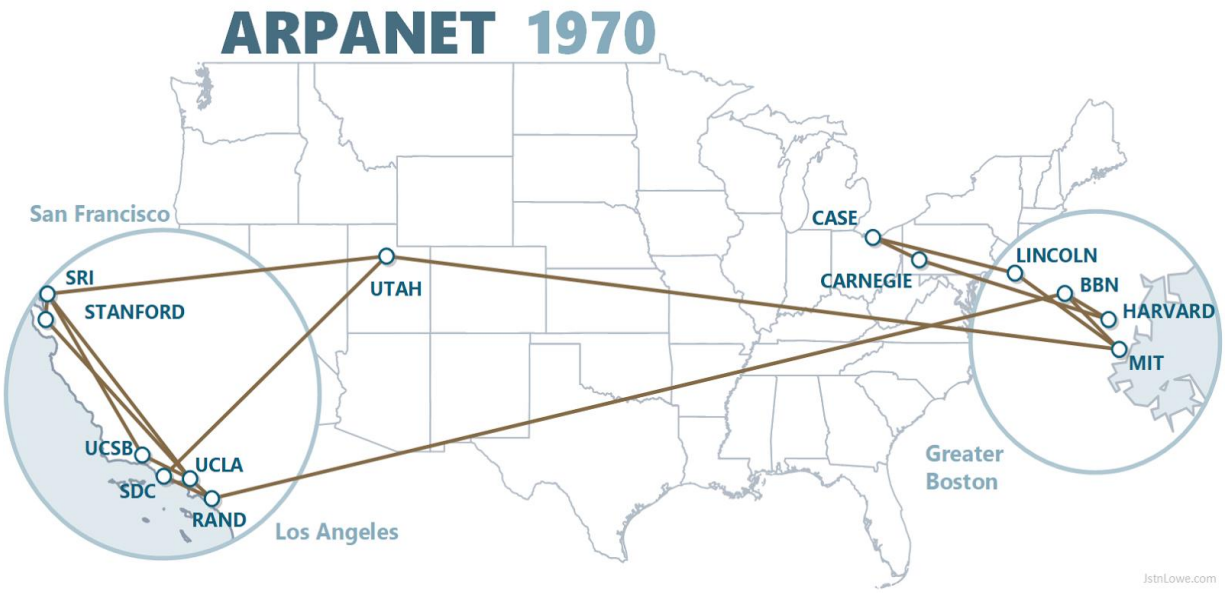


Рис. 1 – Сеть ARPANET в 1970м году.

К началу 1970-х была налажена электронная почта, позволяющая обмениваться сообщениями внутри сети. И в это же время интернет перестал быть исключительно американской системой. К сети подключились университеты Великобритании и Норвегии. По мере роста числа компьютеров в сети, их взаимодействие становилось всё более медленным и рассинхронизированным.

Налаживанием интеграции компьютеров в единую сеть занялся ещё один учёный, работавший в ARPA, — Винстон Сёрф. Сёрф разработал два протокола: протокол управления передачей (TCP); и дополнительный интернет-протокол (IP). Благодаря совместной работе двух протоколов, стало возможным наладить связи между множеством компьютеров, расположенных по всему миру.

В 1980-е годы ARPANET уже был достаточно удобным инструментом, с помощью которого между собой могли общаться университеты, научно-исследовательские лаборатории и институты. В 1984 году возникла система доменных имён. Каждому из компьютеров, включённых в сеть, было присвоено своё доменное имя. Со временем эта система изменилась: домен стал просто составной частью множества электронных адресов, а не именем конкретного устройства. Для удобства имена пользователя и домена стали отделять друг от друга символом — @.

Позднее появился и новый способ общения в сети: владельцы компьютеров могли не просто пересылать друг другу файлы, но и общаться в режиме реального времени в специальных чатах.

Для того чтобы упростить обмен электронной почтой в 1991 году появилась первая соответствующая программа. Однако всё это время Интернет оставался лишь набором каналов для передачи данных с одного компьютера на другой, и пользовались им только ведущие учёные Европы и США. Революционным решением, сделавшим Интернет достоянием всех владельцев компьютеров, стало появление и дальнейшее развитие системы WWW.

В начале 1990-х годов английский физик и программист Тим Бернерс-Ли начал работу над открытой системой, которая позволяла бы размещать в сети различные данные, таким образом, чтобы любой пользователь мог иметь к ним доступ. Изначально планировалось, что эта система позволит обмениваться нужной информацией учёным-физикам. Так появилась хорошо знакомая нам глобальная сеть — World Wide Web (WWW). Для размещения и поиска данных в цифровой сети потребовалось создание дополнительных инструментов: протокола передачи данных HTTP языка HTML. Ну а дальше…

И вот прошло еще два полных десятилетия. И количество подключенных устройств к глобальной сети – единой среде взаимодействия уже больше миллиарда. Значительно больше. На порядок. Десять миллиардов хостов – вопрос времени.

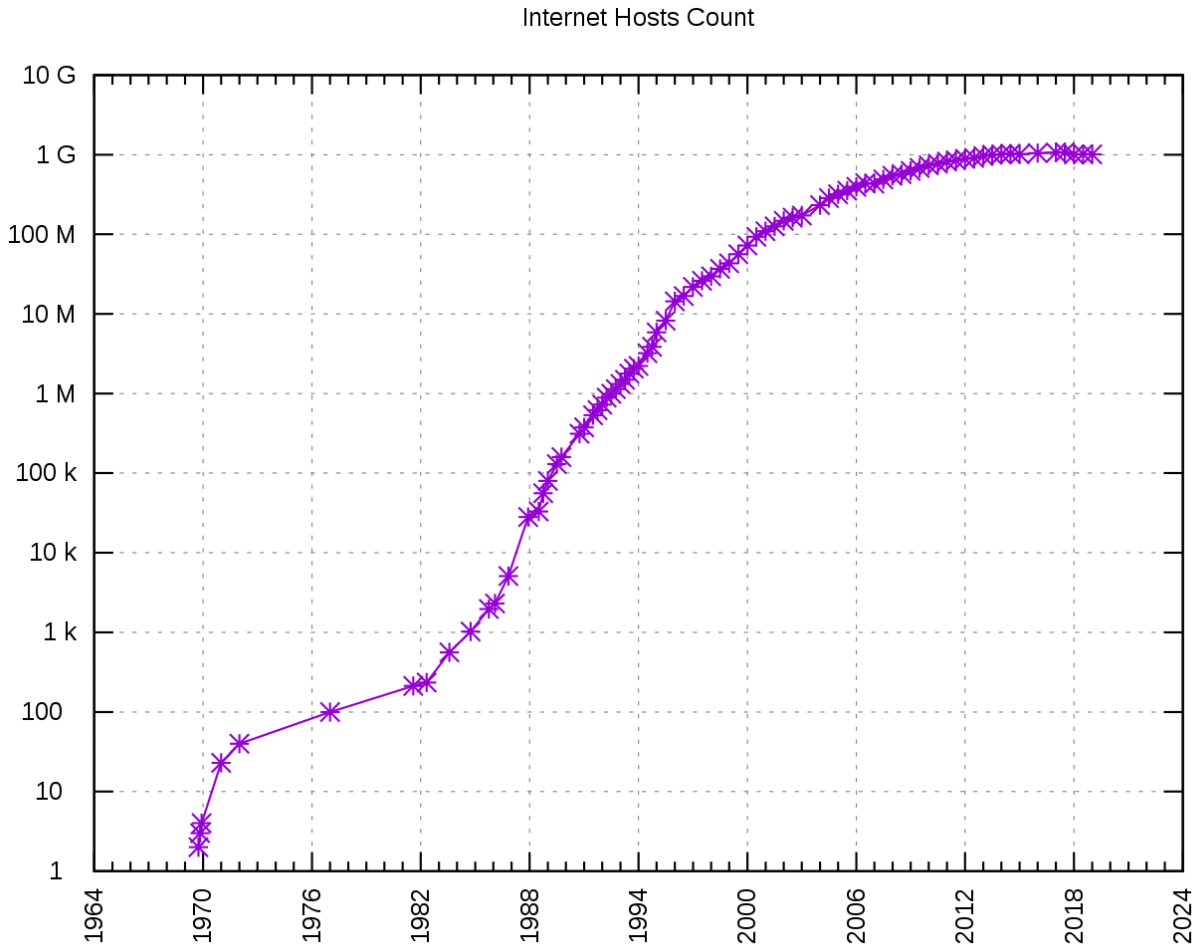


Рис. 2 - Количество Интернет-узлов по всему миру (логарифмическая шкала)

Если в 1970ом году, еще в пору становления ARPANET, было четыре узла, то сейчас их столько, что их количество можно подсчитать лишь по косвенным признакам. Для всей этой громады необходимо было предусмотреть программную, а как следствие, и аппаратную платформу взаимодействия, работающую по единому признаку. И не просто предусмотреть… А и внедрить. На базисе предустановленности.

Ведь, еще, в карикатуре Питера Штейнера, опубликованной 5 июля 1993 года на страницах литературно-публицистического еженедельника «The New Yorker» был подчеркнут самый интересный атрибут нового явления нашей жизни, заключающейся в фразе: «В Интернете никто не знает, что ты собака».

Данную фразу можно интерпретировать по-разному. Понятно, что речь идет не только об анонимности. Напрашиваются как вполне логичные и явные, так и не совсем, аллегории и суждения. Но мало кто вспоминает, что именно WWW десакрализировал роль человека в управлении – по его же и согласию. Вычислительные машины общаются между собой даже тогда, когда мы об этом и не просим. Благодарить за это надо общепринятую эталонную модель межсетевого взаимодействия OSI/ISO, и, термин, известный каждому, а именно «протокол».

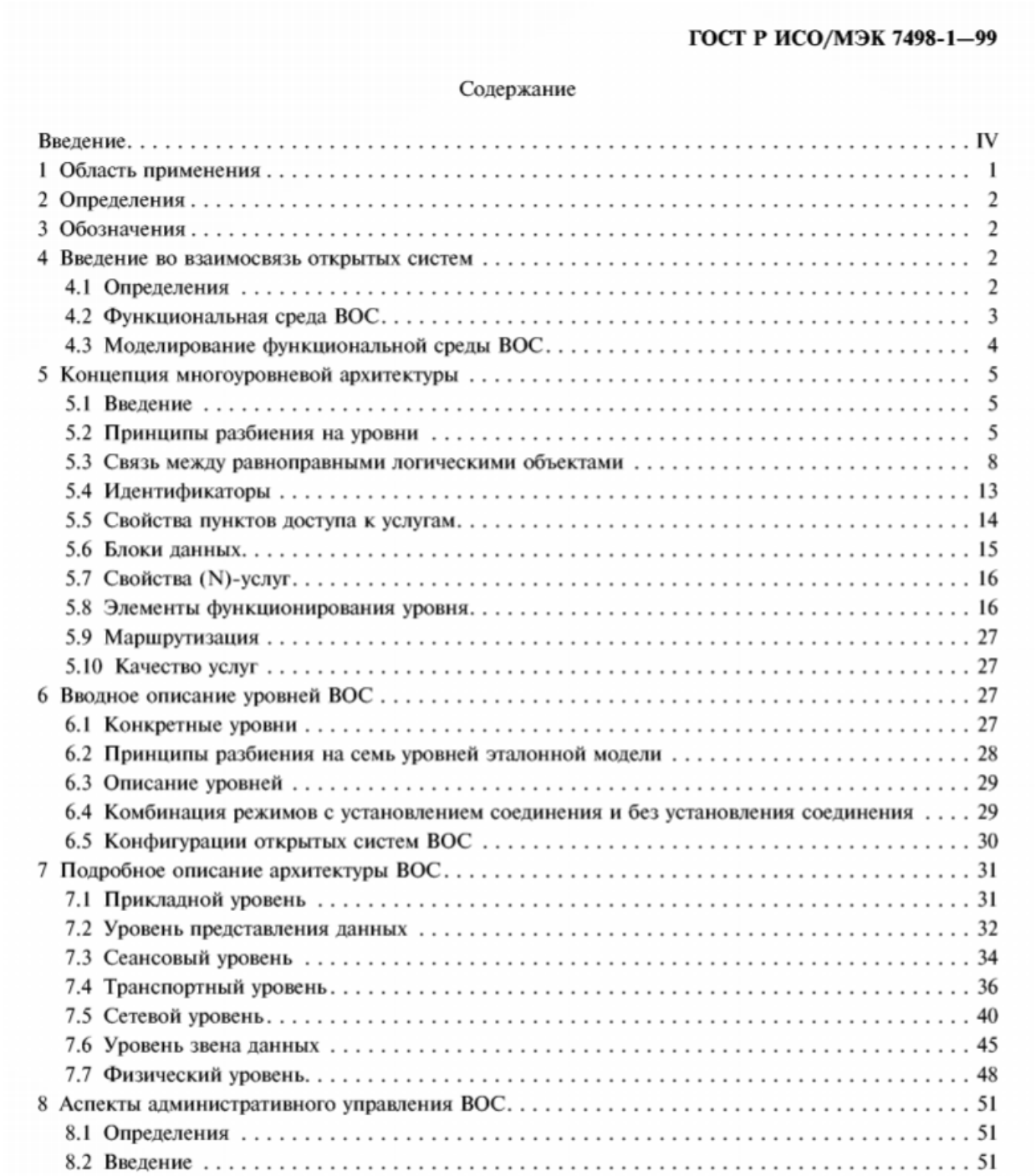
"On the Internet, nobody knows you're a dog."

Когда речь идет о данной модели – то в голове напрашивается поиск должной аналогии. Для создания хорошего ассоциативного эффекта у изучающего. Но так уж получилось, что человечество, создав довольно близкий к идеальному механизм инкапсуляции данных, предоставления их на нужном уровне разным ресурсам, не смогло реализовать такое на практике, отделавшись, давней, и совершенной близко не похожей, при детальном рассмотрении, системой государственной власти (практически любой из современных), и вовсе, казалось бы, наиболее схожей, системой документооборота.

Несмотря на большое значение данной системы, теоретическому описанию принципов работы набора сетевых протоколов, взаимодействующих друг с другом в рамках модели OSI/ISO было суждено предоставить не мне. Прежде чем ознакомится с содержимым данной книги, рекомендую считать QR-код, расположенный на этой страницы.

ГОСТ Р ИСО МЭК 7498

Считав этот код, и перейдя на портал Федерального Агентства по Техническому регулированию и метрологии, вы получите доступ к полному изложению следующего наименования: «Информационная технология. Взаимосвязь открытых систем. базовая эталонная модель. Базовая модель»:



ГОСТ Р ИСО/МЭК 7498-1—99

Содержание

Введение IV
1 Область применения 1
2 Определения 2
3 Обозначения 2
4 Введение во взаимосвязь открытых систем 2
4.1 Определения 2
4.2 Функциональная среда ВОС 3
4.3 Моделирование функциональной среды ВОС 4
5 Концепция многоуровневой архитектуры 5
5.1 Введение 5
5.2 Принципы разбиения на уровни 5
5.3 Связь между равноправными логическими объектами 8
5.4 Идентификаторы 13
5.5 Свойства пунктов доступа к услугам 14
5.6 Блоки данных 15
5.7 Свойства (N)-услуг 16
5.8 Элементы функционирования уровня 16
5.9 Маршрутизация 27
5.10 Качество услуг 27
6 Вводное описание уровней ВОС 27
6.1 Конкретные уровни 27
6.2 Принципы разбиения на семь уровней эталонной модели 28
6.3 Описание уровней 29
6.4 Комбинация режимов с установлением соединения и без установления соединения 29
6.5 Конфигурации открытых систем ВОС 30
7 Подробное описание архитектуры ВОС 31
7.1 Прикладной уровень 31
7.2 Уровень представления данных 32
7.3 Сеансовый уровень 34
7.4 Транспортный уровень 36
7.5 Сетевой уровень 40
7.6 Уровень звена данных 45
7.7 Физический уровень 48
8 Аспекты административного управления ВОС 51
8.1 Определения 51
8.2 Введение 51

Рисунок 3 – Содержимое QR-кода

У вас все получилось, и вы получили доступ к полному изданию стандарта? Тогда я вас поздравляю – только что был достигнут завершающей стадии обмена, процесс взаимодействия двух машин, т.е, прошлое базовое межмашинное взаимодействие: между вашим смартфоном и экраном монитора компьютера (если вы читаете электронный вариант издания).

Компьютерная сеть — это совокупность ПК и других устройств, объединяемых вместе с помощью сетевых кабелей таким образом, что они могут взаимодействовать друг с другом с целью совместного использования информации и ресурсов.

Принято иметь весьма стандартизованные, академические представления о типах компьютерных сетей. Мы считаем, что сети отличаются размерами и по топологическому признаку: некоторые размещаются внутри одного офиса, другие охватывают несколько зданий и даже весь земной шар.

Смею заметить, что данный подход к определению несколько устарел. Виновником это стал небезызвестный фактор, выраженный как закон Мура — эмпирическое наблюдение, изначально сделанное Гордоном Муром, согласно которому (в современной формулировке) количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждые 24 месяца.

Казалось бы – причем тут это. Да и закон, сформулированный еще почти полвека назад, уже давно оспаривается в научном мире. В частности, в последние годы. Такое мнение форсируется преимущественно публицистическими изданиями. И с этим совершенно не согласен Институт инженеров электротехники и электроники (IEEE).

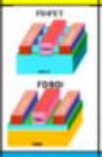
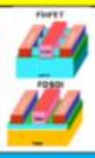
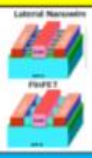
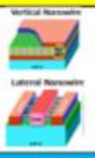
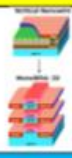
YEAR OF PRODUCTION	2015	2017	2019	2021	2024	2027	2030
Logic device technology naming	P70M56	P54M36	P42M24	P32M20	P24M12G1	P24M12G2	P24M12G3
Logic industry "Node Range" Labeling (nm)	"16/14"	"11/10"	"8/7"	"6/5"	"4/3"	"3/2.5"	"2/1.5"
Logic device structure options	finFET FDSOI	finFET FDSOI	finFET LGAA	finFET LGAA VGAA	VGAA, M3D	VGAA, M3D	VGAA, M3D
							
DEVICE ARCHITECTURE & MODULES							
Starting substrate	Si, SOI	Si, SOI	Si,SOI, SRB, QW	Si,SOI, SRB, QW	Si,SOI, SRB, QW	Si,SOI, SRB, QW	Si,SOI, SRB, QW
N-channel	Si	sSi	sSi, Ge	sSi, sGe, IIIV	sSi, sGe, IIIV	sSi, sGe, IIIV	sSi, sGe, IIIV
P-channel	Si	Si,SiGe	Si,SiGe	Si,SiGe	Ge	Ge	Ge
Channel formation	Etch	Etch, EPI	Etch, EPI	Etch, EPI	Etch, EPI	Etch, EPI	Etch, EPI
Contact material	Silicide	Low-SBH	Low-SBH	Low-SBH	Low-SBH	Low-SBH	Low-SBH
Contact integration	EPI	EPI	EPI WAC	WAC	WAC		
DEVICE PERFORMANCE BOOSTERS							
Main performance booster	SCE finHeight Vt	SCE finHeight Vt	Parasitics finHeight	Parasitics finHeight	Low Vdd 3D		
Scaling focus	Perf	Power	Power	Power	Function		
Channel strain	Yes	Yes	Yes	Yes	Yes		
S/D strain	Yes	Yes	Yes	Yes	Yes		
Transport scheme	DD	Quasi Ballistic	Quasi Ballistic	Ballistic	Ballistic TFET, JFET, NCMOS		

Рисунок 4 – Дорожная карта из брошюры IEEE по прогнозированию продолжения работы законы Мюра на 2015-2030 гг.

А ведь связь напрашивается очевидная – технологический процесс изготовления полупроводниковых (п/п) изделий и материалов прямо связан и с сетевыми технологиями. Правда, про это не желают упоминать. Надеясь на смекалку.

Если говорить очень упрощённо, возвращаясь к архитектуре компьютерных систем, то процессор, не нуждающийся в каком-либо представлении, — это миллиарды крошечных транзисторов и электрических затворов, которые включаются и выключаются при выполнении операций.

И, например, «7 нм тех.процесс» — это размер этих транзисторов в нанометрах. Для понимания масштабов стоит напомнить, что в одном миллиметре миллион нанометров, а человеческий волос толщиной 80000-110000 нанометров. Транзистором, напомню, называют радиоэлектронный компонент из полупроводника (материал, у которого удельная проводимость меняется от воздействия температуры, различных излучений и прочего), который от небольшого входного сигнала управляет значительным током в выходной цепи. Он используется для усиления, генерирования, коммутации и преобразования электрических сигналов. Сейчас транзистор является основой схемотехники подавляющего большинства электронных компонентов и интегральных микросхем. Размер транзистора полезно знать специалистам для оценки производительности конкретного процессора, ведь чем меньше транзистор, тем меньше требуется энергии для его работы.

Собственно, только благодаря поступательному развитию в этом направлении, в усилиях по уменьшению технологического процесса, стал возможным полноценный беспроводной обмен в рамках технологий семейства GSM/3GPP/LTE. А однокристальные системы (SoC) - электронные схемы, выполняющие функции целого устройствами размещённые на одной интегральной схеме стали такими популярными (рис. 5).

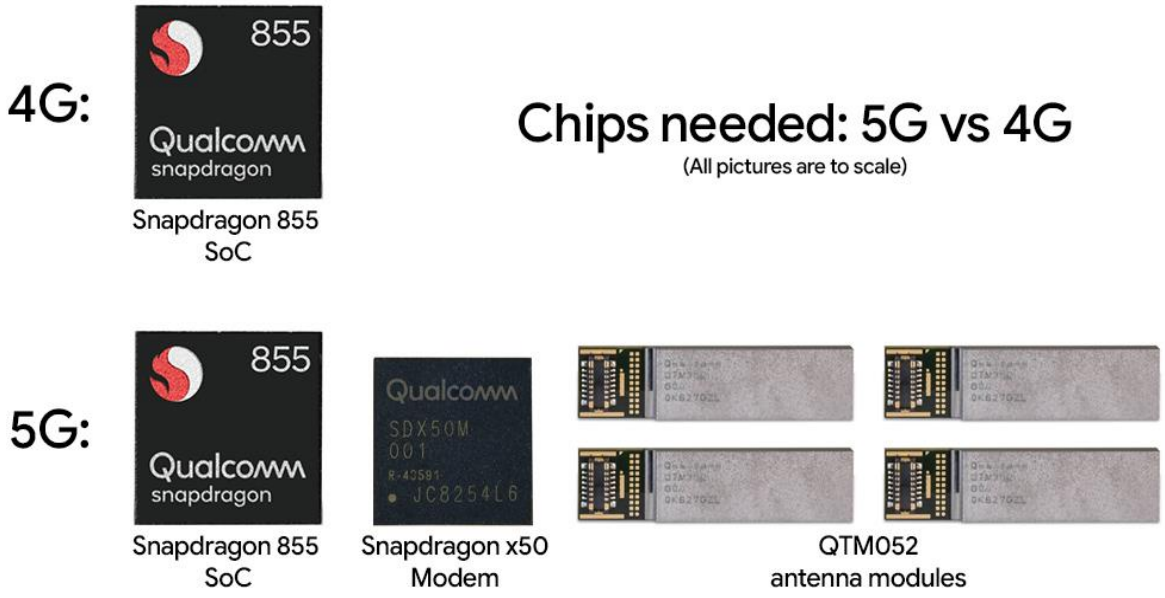


Рис.5 – SoC с поддержкой беспроводных сетевых технологий

Сегодня, в 2020 году на крохотной площадке в 49мм^2 (Qualcomm Snapdragon 855) помещается полноценная компьютерная система. И данный представитель SoC интегрирован в смартфон. Не в специализированное устройство, а в обычный смартфон. В нем есть всё, что есть в обычных компьютерах – CPU, GPU, RAM, ROM и многое другое: от полноценной реализации контроллера оперативной памяти, сигнального процессора (DSP) и до сопроцессора обработки изображений (рис.6).

Характеристики Qualcomm Snapdragon 855	
Техпроцесс	7 нм (TSMC)
Архитектура	64 бита
Центральный процессор	8 ядер Kryo 485 (1+3+4) 1 ядро Cortex A76 до 2,84 ГГц 3 ядра Cortex A76 до 2,42 ГГц 4 ядра Cortex A55 до 1,8 ГГц
Контроллер памяти	LPDDR4x 4-канальный (64 бита) 2133 МГц, до 34,13 ГБ/с До 16 ГБ
Графический процессор	Adreno 640 384 ядра DirectX 12, Vulkan, OpenGL 3.2, OpenCL 2.0 ≈ 1 TFLOPS (FP32)
Экран	4K внутренний, 4K внешний (до 2 шт.), HDR 10+, до 120 Hz
Сигнальный сопроцессор (DSP)	Hexagon 690
Сопроцессор обработки изображений (ISP)	Spectra 380, двойной, 14-битный сигнал Аппаратное ускорение машинного зрения
Фотокамера	Одиночная – до 48 Мп (с MFNR, ZSL) Двойная (одновременная работа) – до 22 или 16+16 Мп (с MFNR, ZSL) Максимальное разрешение – 192 Мп
Запись видео	До 4K 60 FPS, эффект боке, HDR10, HLG, Rec. 2020 Slow-Mo 720p 480 FPS
Воспроизведение видео	H.265 (HEVC), H.264 (AVC), HLG, HDR10, HDR10+, VP8, VP9
Мобильная связь	GSM, HSPA, CDMA, LTE Cat 20 до 2 Гбит/с (загрузка) / Cat. 13 до 384 Мбит/с (передача) Поддержка дискретного модема 5G Snapdragon X50
Wi-Fi	802.11a/b/g/n/ac/ad/ay/ax-ready, диапазоны 2,4, 5 и 60 ГГц, до 10 Гбит/с
Bluetooth	5.0
Навигатор	GPS, BeiDou, Galileo, QZSS, ГЛОНАСС, SBAS, двухчастотный
USB	3.1
Быстрая зарядка	4+

Рис.6 – Характеристики SoC Qualcomm Snapdragon 855

И что наиболее важно – данная SoC имеет на своем миниатюрном борту исправно работающий тандем из устройств, представляющих информационный обмен посредством технологий: Bluetooth (IEEE 802.15.1), Wi-Fi (IEEE 802.11), LTE (4G), и 5G. Кажется, что все это время мы недооценивали смартфоны, пренебрежительно относились к носимой электронике. Необходимо дать трезвый отчет тому, что с точки зрения детализированного подхода к определению роли практически любого гаджета появилась большая и стремительно увеличивающаяся информационная энтропия. И общая проблема заключается в том, что необходимо трактовать смартфоны, умные часы, устройства IoT как самостоятельные информационные системы, самоорганизующиеся сети[1], а не только как оконечные узлы.

[1] Самоорганизующаяся сеть – сеть, не имеющая определенной структуры, меняющаяся и распределяющая функции между узлами при подключении нового устройства, изменении характера трафика и т.д.

В иноязычной литературе данный пробел в познании рамок сетевого инжиниринга был решен за счет внедрения дополнительных типов сетей (по территориальному признаку).

На данный момент принято делить все сети по вышеперечисленному признаку на следующие типы (подвиды), которым и необходимо дать более детальную характеристику в актуальном формате:

BAN (Body Area Network — нательная компьютерная сеть) — сеть надеваемых или имплантированных компьютерных устройств.

PAN (Personal Area Network) — персональная сеть, предназначенная для взаимодействия различных устройств, принадлежащих одному владельцу.

LAN (**ЛВС,** Local Area Network) — локальные сети, имеющие замкнутую инфраструктуру до выхода на поставщиков услуг. Термин «LAN» может описывать и маленькую офисную сеть, и сеть уровня большого завода, занимающего несколько сотен гектаров. Локальные сети являются сетями закрытого типа, доступ к ним разрешён только ограниченному кругу пользователей, для которых работа в такой сети непосредственно связана с их профессиональной деятельностью.

CAN (Campus Area Network) — кампусная сеть, объединяет локальные сети близко расположенных зданий. Диапазон CAN составляет от 1 км до 5 км. Если два здания имеют один и тот же домен, и они связаны между собой сетью, то это будет рассматриваться только как CAN. Хоть и CAN в основном используется для корпоративных кампусов, канал передачи данных будет иметь высокую скорость.

MAN (Metropolitan Area Network) — городские сети между учреждениями в пределах одного или нескольких городов, связывающие много локальных вычислительных сетей.

WAN (Wide Area Network) — глобальная сеть, покрывающая большие географические регионы, включающие в себя как локальные сети, так и прочие телекоммуникационные сети и устройства. Пример WAN — сети с коммутацией пакетов (Frame relay), через которую могут «разговаривать» между собой различные компьютерные сети. Глобальные сети являются открытыми и ориентированы на обслуживание любых пользователей.

Большой прогресс в физиологических аппаратах, маломощных интегрированных схемах и беспроводных коммуникациях сделал возможным новое поколение т.н. беспроводных сенсорных сетей, ныне используемых для таких целей, как мониторинг пробок, урожая, инфраструктур и здоровья. Нательная компьютерная сеть позволяет провести недорогой и продолжительный мониторинг тела в реальном времени через Интернет. Несколько интеллектуальных физиологических аппаратов могут быть интегрированы в надеваемые устройства, которые могут использоваться для компьютерной реабилитации или заблаговременного обследования состояния здоровья. Эта область основывается на возможности имплантации очень маленьких датчиков внутрь человеческого тела, которые очень удобны и не нарушают нормальную деятельность человека. Имплантированные в тело аппараты будут отслеживать различные физиологические изменения, чтобы контролировать состояние здоровья пациента независимо от его местоположения. Эта информация будет передана по беспроводному каналу. Устройство будет мгновенно передавать всю информацию в режиме реального времени врачам во всем мире. Если обнаружена экстренная ситуация, врачи сразу же проинформируют пациента через компьютерную систему посредством отправки соответствующих сообщений или аварийных сигналов. Хотя технология все ещё находится в своей начальной стадии, она широко исследуются, и после её принятия ожидается прорыв в области здравоохранения.

BAN устройства могут быть встроены в тело, имплантированы, прикреплены к поверхности тела в фиксированном положении или совмещены с устройствами, которые люди носят в различных местах (в карманах, на руке или в сумках). Несмотря на уменьшение размера устройств, т.к. сети, состоящие из нескольких миниатюрных сенсорных блоков (BSU), объединяются с единым центральным блоком тела (BCU) устройства размером более дециметра (планшеты, КПК), по-прежнему играют большую роль, выступая, в таком случае, концентраторами информации, предоставляя пользовательский интерфейс для обзора и управления BAN приложениями «на месте».

Разработка технологии BAN началась еще в конце 90х годов прошлого века на основе идеи использования беспроводных персональных сетей для реализации связи «на», «рядом», и «вокруг» человеческого тела. Около десяти лет спустя термин BAN стал обозначать системы, где связь полностью «в пределах», «на» или «в непосредственной близости» от человеческого тела.

BAN может использовать беспроводные технологии в качестве шлюзов для достижения больших расстояний. Через шлюзы можно соединять надеваемые на человеческое тело устройства через Интернет. Таким образом, мед. работники могут получить доступ к данным о пациенте онлайн, используя Интернет вне зависимости от местоположения пациента.

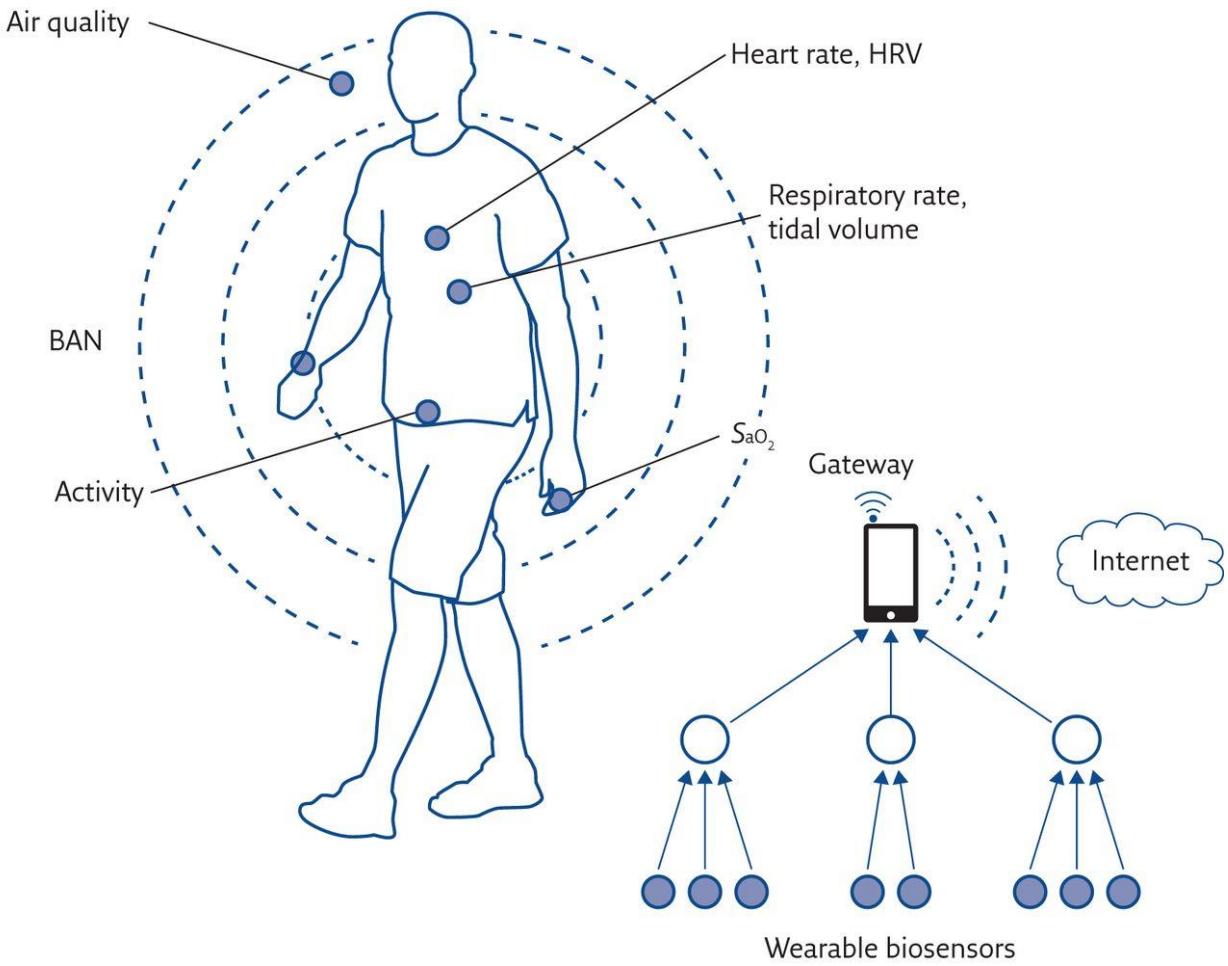


Рис.7 – BAN как носимая (Wearable) технология

Первоначальные применения нательной компьютерной сети тела прежде всего ожидаются в области здравоохранения, особенно для непрерывного мониторинга и записи важных данных о пациентах, страдающих от хронических заболеваний, таких как диабет, астма и сердечные приступы (см. рис.7).

Нательная компьютерная сеть может предупредить по сети больницу даже прежде, чем у пациента случится сердечный приступ, путём слежения за важными изменениями человека.

Также, технология позволяет автоматически вводить инсулин больным диабетом, как только уровень инсулина снижается.

Технология также применима в спорте, военном деле или охране безопасности. Продвижение технологии в новых областях поможет беспроводным обменом информацией и между людьми и машинами.

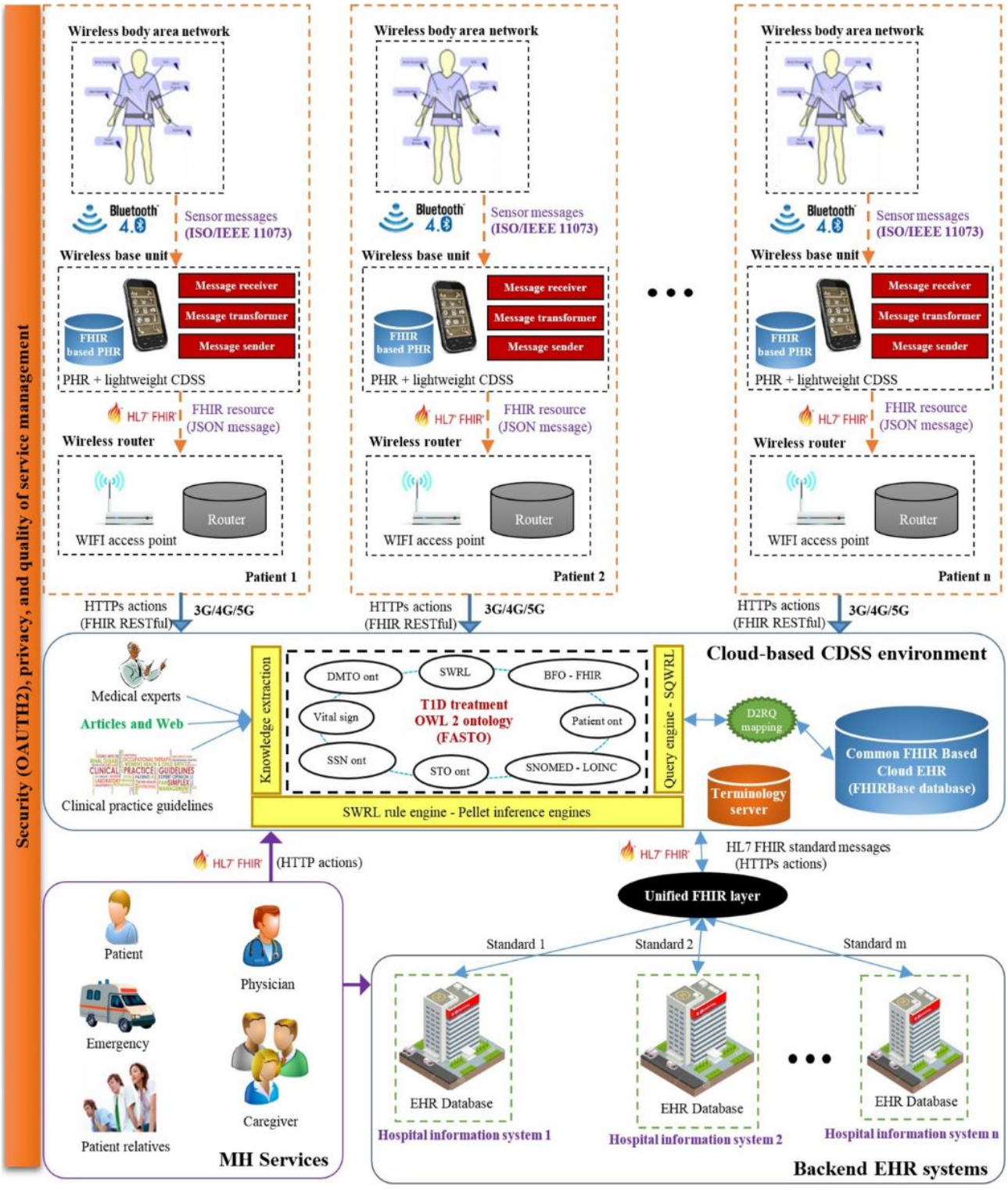


Рис.8 – Инфологическая карта применения BAN технологий в медицине

На рисунке 8, показана модель взаимодействия между тремя BAN/WBAN и неуказанной в явном виде, но явно более глобальной (по территориальному признаку) сети, т.е информационной системы, в которой, в упрощённом виде главным информационным субъектом является видовая составляющая трёх типов электронной медицинской карты, которые определены на рисунке по следующим аббревиатурам:

Electronic Health Record (EHR) — хранит информацию относительно всех медицинских заболеваний, хранителем является специально авторизованный центр (Health Authority). Медицинские записи являются официальными данными, могут быть доступны для других авторизованных центров и подобных представителей медицинских услуг, а также лабораторий, гос. учреждений и т.п. для улучшения качества здравоохранения.

Electronic Medical Record (EMR/MH Services) – хранит информацию относительно конкретной медицинской области (например, стоматология), хранителем является клиника или практикующий врач. Обычно это электронная версия истории болезни пациента в данном конкретном учреждении.

Personal Health Record (PHR) – хранит какую-то медицинскую информацию, хранителем, а точнее ответственным за полноту и качество информации, является сам пациент (или его представителя, например, член семьи).

Данная схема является примером практической реализации сети такого плана. Все те же, классические HTTP-запросы, клиент-серверная архитектура и топологическая организация.

Проблемы с использованием технологий и решений на базе BAN могут заключаться в:

Взаимодействии: BAN системы должны обеспечивать беспрепятственную передачу данных через стандарты такие, как Bluetooth, ZigBee и т. д., чтобы способствовать обмену информацией между взаимодействующими устройствами. Кроме того, эти системы должны быть масштабируемыми, обеспечивать эффективный переход между сетями и предлагать непрерывное соединение.

В системных устройствах: Датчики, используемые в Wireless BAN должны быть низкой сложности, небольшие по размеру, легкие в весе, мощные, легкие в использовании и перенастраиваемые. Кроме того, устройства хранения данных должны облегчить дистанционное хранение устройств и просмотр за пациентами, а также доступ к внешним средствам обработки и анализа через Интернет.

В системной и аппаратной безопасности: Требуются значительные усилия, чтобы сделать BAN безопасной и точной. Мы должны быть уверены, что данные о пациентах не могут перепутаться.

В вторжении в личную жизнь: Люди могут рассматривать технологию BAN как потенциальную угрозу для их свободы, если исследования выйдут за рамки безопасности здоровья. Общественное признание стало бы ключом к этой технологии, находящим более широкое применение.

В проверке (опросе) датчика: Распространенным устройствам зондирования свойственны аппаратные и сетевые ограничения. Это может привести к ошибочным данным, передаваемым обратно к конечному пользователю. Имеет первостепенное значение, особенно в области здравоохранения, проверка показаний датчиков. Благодаря этому можно сократить число ложных созданий сигналов тревоги и определить возможные слабые места в рамках аппаратного и программного обеспечения.

В согласованности данных: Данные, находящиеся на нескольких мобильных устройствах и беспроводных аппаратах пациентов, должны быть собраны и проанализированы. В BAN жизненно важные данные пациента могут проходить через множество узлов, сетей и компьютеров. Если мобильное устройство практикующего врача не содержит всю известную информацию, то качество медицинской помощи может снизиться.

В вмешательстве: беспроводная связь, используемая для датчиков тела, должна минимизировать помехи и повысить сосуществование узлов сенсорных устройств с другими сетевыми устройствами, доступными в окружающей среде. Это особенно важно для реализации крупномасштабных систем BAN.

В управлении данными: BAN генерирует данные в больших объёмах, поэтому управление информацией имеет первостепенное значение.

Помимо аппаратно-ориентированных задач, следующие вопросы, касающиеся человека, должны учитываться в развитии BAN:

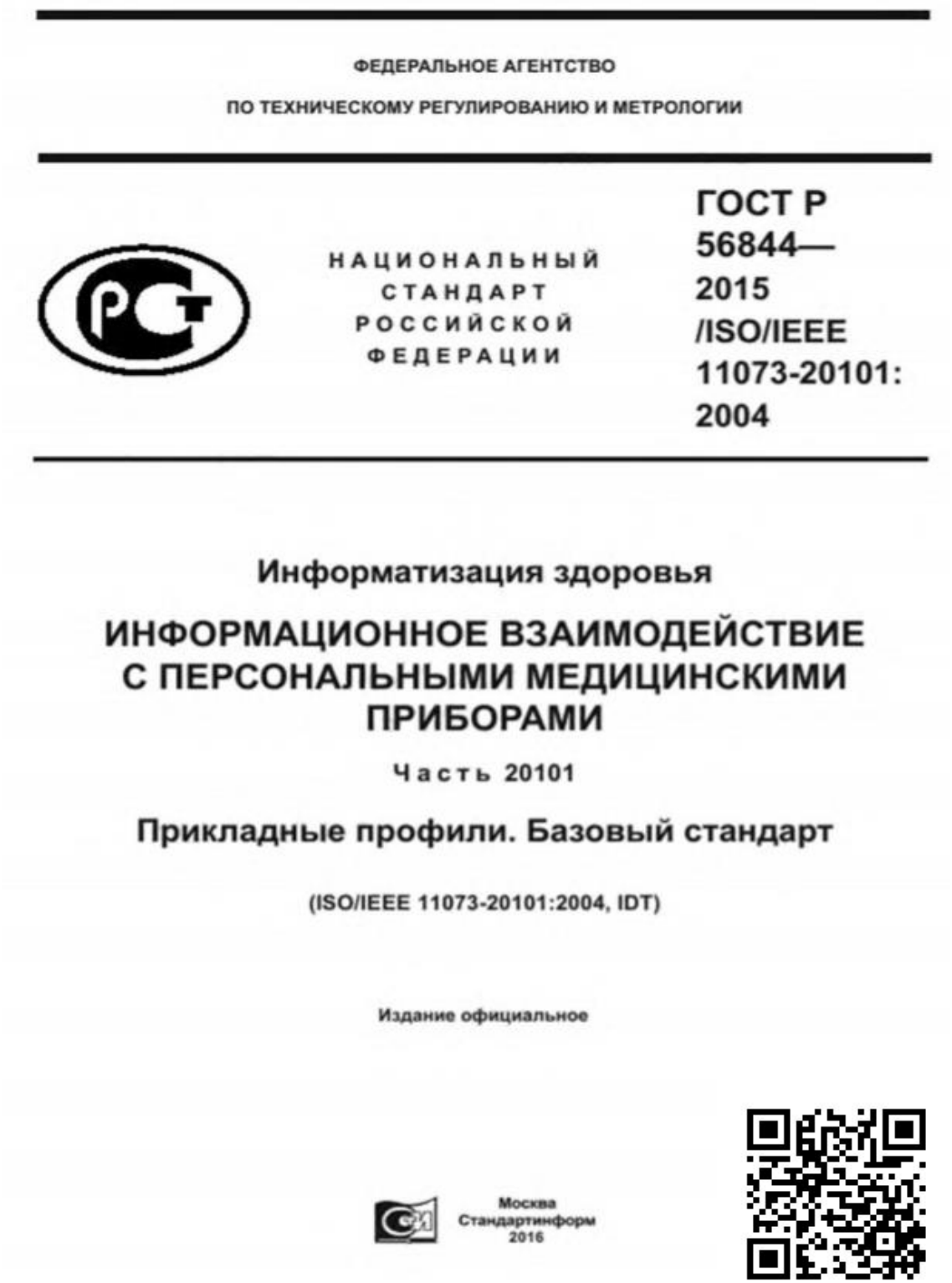
Стоимость: В наши дни потребители ожидают низких цен для мониторинга здоровья, которые должны сочетаться с высоким функционалом.

Постоянный мониторинг: Пользователям могут потребоваться различные уровни мониторинга, например, тем, кто подвержен риску ишемической болезни, необходимо, чтобы BAN постоянно работал, в то время как другим группы риска может потребоваться только контроль BAN'а, когда они ходят или двигаются. Уровень мониторинга влияет на количество требуемой энергии и определенный заряд энергии.

Размещение: BAN должен быть удобным для надевания, легким, ненавязчивым. Он не должен изменять повседневную деятельность пользователя, обременять его. Технология должна в конечном счете быть удобной для пользователя, выполнять свои задачи мониторинга без его ведома.

Производительность: Производительность BAN должна быть стабильной. Измерения датчиков должны быть точными, даже если BAN выключается и включается снова. Беспроводные каналы связи должны быть надежными и работать под различными средами пользователей.

Таким образом, нательные компьютерные сети – по-прежнему прерогативны для будущего, для технологий завтрашнего дня. Но техническое регламентирование информационного взаимодействия в отдельных отраслях уже вполне доступно не только для обозрения, а и для проектировочной и пуско-наладочной деятельности. Как например, семейство стандартов IEEE/ISO 11073:



ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56844— 2015 /ISO/IEEE 11073-20101: 2004

Информатизация здоровья

ИНФОРМАЦИОННОЕ ВЗАИМОДЕЙСТВИЕ С ПЕРСОНАЛЬНЫМИ МЕДИЦИНСКИМИ ПРИБОРАМИ

Часть 20101

Прикладные профили. Базовый стандарт

(ISO/IEEE 11073-20101:2004, IDT)

Издание официальное

Москва
Стандартинформ
2016

Рис. 9 - Стандарты комплекса ISO/IEEE 11073 определяют взаимосвязь между медицинскими приборами и внешними компьютерными системами

Общие положения сферы применения WBAN (BAN) отражены в стандарте IEEE 802.15.6.

Стандарт **IEEE 802.15.6** направлен на обеспечение конфиденциальности, аутентификации, целостности, защиты конфиденциальности и защиты воспроизведения. Все узлы и концентраторы должны выбрать три уровня безопасности: незащищенная связь (уровень 0, аутентификация, но без шифрования(уровень 1) и аутентификация и шифрование (уровень 2). В процессе сопоставления безопасности узел и концентратор должны совместно выбрать подходящий уровень безопасности.

Все узлы и концентраторы в WBAN должны пройти определенные этапы на уровне MAC перед обменом данными. А ассоциация по вопросам безопасности это процедура для идентификации узла и концентратора друг к другу, чтобы установить новый мастер-ключ (MK) совместно используемый между ними, или активировать существующий MK предварительно совместно используемый между ними**. Ассоциация безопасности**[2] в стандарте IEEE 802.15.6 основана на четырех ключевых протоколах соглашения, которые имеют проблемы безопасности. В опубликованной академической литературе есть несколько интересных предложений, которые соответствующим образом решают проблемы безопасности и конфиденциальности текущих процедур.

WBAN поддерживает разнообразие в реальном масштабе времени контроль здоровья и применения бытовой электроники. Последним международным стандартом для WBAN является стандарт IEEE 802.15.6 который направлен на обеспечение международного стандарта для низкой мощности, короткого диапазона и чрезвычайно надежной беспроводной связи в пределах окружающей области человеческого тела, поддерживая широкий диапазон скоростей передачи данных для различных приложений. В настоящем стандарте указываются краткосрочные беспроводные средства связи в непосредственной близости или внутри человеческого тела (но не только для людей).

Он использует существующие промышленные научные медицинские (ISM) полосы, а также частотные полосы, одобренные национальными медицинскими и / или регулирующими органами.

---

[2]Ассоциация безопасности - это симплексное соединение, которое позволяет предоставлять услуги безопасности для трафика, переносимого этим соединением. Услуги безопасности предоставляются ассоциации безопасности посредством использования протоколов идентификации (МСЭ-Т Y.1281).

Второй ступенью развития компьютерных сетей по территориальному признаку (по возрастанию) являются так называемые Personal Area Networks.

Персональная сеть (англ. Personal Area Network, PAN) — это сеть, построенная «вокруг» человека. PAN представляет собой компьютерную сеть, которая используется для передачи данных между устройствами, такими как компьютеры, телефоны, планшеты и персональные карманные компьютеры (КПК). Персональные сети могут использоваться как для информационного взаимодействия отдельных устройств между собой (интерперсональная коммуникация), так и для соединения их с сетями более высокого уровня, например, глобальной сети Интернет (восходящая линия связи), где одно "первичное" устройство берет на себя роль интернет-маршрутизатора.

Беспроводная персональная сеть (WPAN) является маломощной PAN, которая организуется на небольшом расстоянии с использованием беспроводных сетевых технологий, таких как: IrDA, Bluetooth, Z-Wave, ZigBee, Piconet.

Радиус действия WPAN составляет от нескольких десятков сантиметров до нескольких метров, так что все устройства находятся в одной рабочей области. PAN также может организовываться с использованием проводных компьютерных шин, таких как USB и FireWire.

Хотя, использование мобильного телефона в качестве точки доступа для других устройств через Wi-Fi соединение может быть использовано только одним пользователем, всё же такая сеть не считается PAN.

Беспроводная персональная сеть (WPAN) это та же самая персональная сеть, однако в ней, все соединения являются беспроводными. Беспроводной PAN основан на стандарте **IEEE 802.15**. В WPAN используются два вида беспроводных технологий. Это Bluetooth и Infrared Data Association.

Беспроводные персональные сети применяются для связи различных устройств (как портативных, так и настольных), включая компьютерную, бытовую и оргтехнику, средства связи и т. д. Такие сети могут иметь и более специализированное назначение, например, в медицине.

Ключевым понятием в технологии WPAN является "подключение". В идеальном случае, когда любые два WPAN-оборудованные устройства находятся в непосредственной близости (в пределах нескольких метров друг от друга) или на расстоянии нескольких километров от центрального сервера, они могут общаться, как будто соединены с помощью кабеля. Другой важной особенностью является способность каждого устройства выборочно блокировать связь с другими устройствами, с целью предотвращения несанкционированного доступа к информации.

Технология WPAN сейчас находится в стадии становления и переживает бурное развитие. На данный момент в цифровом режиме предлагаются рабочие частоты порядка 2,4 ГГц. Цель, которую хотят достичь в этой технологии - обеспечить стабильную и бесперебойную работу всех систем, использующих WPAN. Каждое устройство в WPAN будет иметь возможность подключиться к любому другому устройству в той же WPAN, при условии, что они находятся в зоне видимости друг друга. Кроме того, в будущем, во всем мире все беспроводные персональные сети будут взаимосвязаны. Так, например, археолог на сайте в Греции сможет использовать карманный компьютер для прямого доступа к базам данных в университете Миннесоты в Миннеаполисе, и передать результаты этой базе данных.

Беспроводные персональные сети **обычно охватывают диапазон** от нескольких сантиметров до 10 метров (33 фута). Эти сети можно рассматривать как особый тип (или подмножество) **локальных сетей**, которые поддерживают одного человека вместо группы.

Связь ведущего-ведомого устройства может иметь место в PAN, где несколько устройств подключаются к «главному» устройству, называемому ведущим. Ведомые реле передают данные через ведущее устройство. С Bluetooth такая настройка может достигать 100 метров (330 футов).

Хотя PAN, по определению, **личные**, они могут по-прежнему получать доступ к Интернету при определенных условиях. Например, устройство в пределах PAN может быть подключено к локальной сети, которая имеет доступ к Интернету, которая является глобальной сетью. Чтобы каждый тип сети был меньше следующего, но все они могут быть в конечном счете тесно связаны.

Личные сети могут быть беспроводными или сконструированы с помощью кабелей. USB и FireWire часто соединяют проводную PAN, в то время как WPAN обычно используют Bluetooth (и называются piconets) или иногда инфракрасные соединения.

Пример: клавиатура Bluetooth подключается к планшету для управления интерфейсом, способным достичь близкой «умной» лампочки.

Кроме того, принтер в небольшом офисе или доме, который подключается к ближайшему настольному компьютеру, ноутбуку или телефону, считается существующим в PAN. То же самое можно сказать о клавиатурах и других устройствах, использующих IrDA (Инфракрасная ассоциация данных).

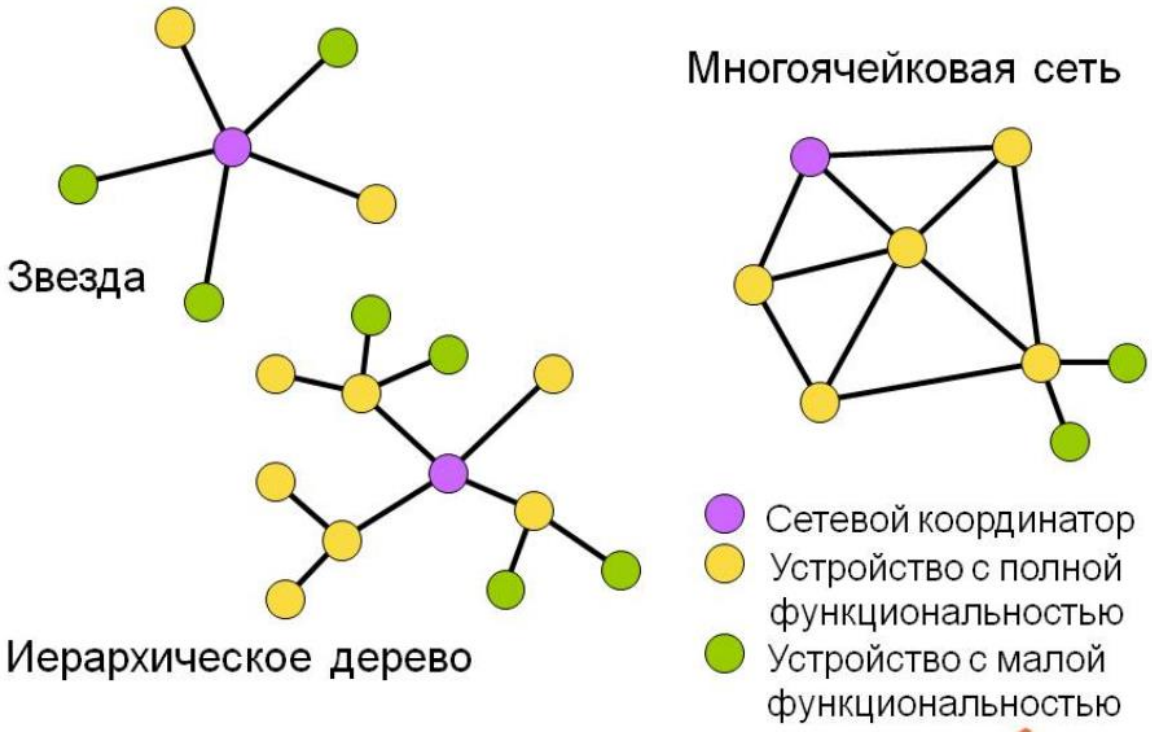
Преимущества персональной сети:

PAN для личного использования, поэтому преимущества могут быть более понятны, чем при разговоре о глобальных сетях, например, описывающих Интернет. Благодаря личным сетям ваши личные устройства могут соединяться для более удобного общения. Нет необходимости, чтобы все сообщения передавались через более крупную сеть только для того, чтобы их принимали люди в нескольких футах от них. PAN позаботится об этом.

Пример, упомянутый выше, — это беспроводная клавиатура или даже мышь. Устройствам такого типа не нужно управлять компьютерами в других зданиях или городах, поэтому они вместо этого строятся, чтобы просто общаться с ближайшим, как правило, устройством прямой видимости, таким как компьютер или планшет.

Поскольку большинство устройств, поддерживающих связь на коротком расстоянии, могут блокировать соединения, которые не являются предварительно авторизованными, WPAN считается защищенной сетью.

В топологическом плане PAN-сети обладают весьма большими возможностями (рис.10): возможно построение сетей на базе физических топологий «звезда», «дерево», «Mesh» (решетка).



Однако архитектура PAN, описанная в IEEE 802.15 подразумевает и особенности, которые необходимо учитывать при выборе той или иной топологии – зачастую, в PAN, из-за жестких рамок по электропитанию, часть носимых сетевых устройств не могут иметь полную функциональность в информационном обмене.

Поэтому, Сети PAN строятся из базовых станций трех основных типов: координаторов, маршрутизаторов и конечных устройств.

Координаторы запускает сеть и управляет ею. Он формирует сеть, выполняет функции центра управления сетью и доверительного центра (trust-центра) – устанавливает политику безопасности, задает настройки в процессе присоединения устройств к сети, ведает ключами безопасности.

Маршрутизатор PAN (устройство с полной функциональностью) транслирует пакеты, осуществляет динамическую маршрутизацию, восстанавливает маршруты при перегрузках в сети или отказе какого-либо устройства. При формировании сети маршрутизаторы присоединяются к координатору или другим маршрутизаторам, и могут присоединять дочерние устройства – маршрутизаторы и конечные устройства. Маршрутизаторы работают в непрерывном режиме, имеют стационарное питание и могут обслуживать «спящие» устройства. Маршрутизатор может обслуживать до 32 устройств (ZigBee).

Конечное устройство (устройство с малой функциональностью) может принимать и отправлять пакеты, но не занимается их трансляцией и маршрутизацией. Конечные устройства могут подключаться к координатору или маршрутизатору, но не могут иметь дочерних устройств.

Конечные устройства могут переводиться в спящий режим для экономии заряда аккумуляторов. Именно конечные устройства имеют дело с датчиками, локальными контроллерами и исполнительными механизмами.

Не зря, ранее в книге было упомянуто понятие **самоорганизующихся сетей.**

Сеть PAN – самоорганизующаяся, и ее работа начинается с формирования. Устройство, назначенное при проектировании координатором персональной сети (PAN координатор), определяет канал, свободный от помех, и ожидает запросов на подключение.

Устройства, пытающиеся присоединиться к сети, рассылают широковещательный запрос. Пока PAN координатор – единственное устройство в сети, отвечает на запрос и предоставляет присоединение к сети только он. В дальнейшем присоединение к сети могут предоставлять также присоединившиеся к сети маршрутизаторы.

Устройство, получившее ответ на широковещательный запрос, обменивается с присоединяющим устройством сообщениями, чтобы определить возможность присоединения. Возможность определяется способностью присоединяющего маршрутизатора обслужить новые устройства в дополнение к ранее подключенным.

Вступление в сеть (присоединение)

Существует два способа присоединения: MAC ассоциация и повторное сетевое присоединение (NWK rejoin).

MAC ассоциация

MAC ассоциация доступна любому устройству ZigBee и осуществляется на MAC уровне. Механизм MAC ассоциации следующий:

Устройство, позволяющее присоединиться к нему, выставляет на MAC уровне разрешение на присоединение.

Устройство, вступающее в сеть, выставляет на MAC уровне запрос на присоединение и передает широковещательный запрос маячка.

Получив «маячок» от устройств, готовых подключить присоединяемое устройство, последнее определяет, в какую сеть и к какому устройству оно желает присоединиться, и выставляет на MAC уровне требование о вступлении с флажком «повторное присоединение» в значении FALSE.Затем вступающее устройство направляет на выбранное для присоединения устройство запрос присоединения и получает ответ с присвоенным ему сетевым адресом.При MAC ассоциации данные передаются не зашифрованными, поэтому MAC ассоциация не является безопасной. Повторное сетевое присоединение вопреки названию может применяться и при первичном присоединении. Оно выполняется на сетевом уровне. При этом, если вступающее устройство знает текущий сетевой ключ, обмен пакетами может быть безопасным. Ключ может быть получен, например, при настройке.

При повторном подключении присоединяющееся устройство выставляет на сетевом уровне запрос присоединения и обменивается с подключающим устройством пакетами «запрос присоединения» – «ответ на запрос присоединения». Кроме случаев присоединения новых устройств структура сети меняется и в случаях, когда устройства покидают сеть и повторно присоединяться в других местах (это происходит, например, в случае перезагрузки устройства).

На рисунке 11 – пример переподключения. Устройство с адресом «0E3B» переподключается как «097D», а затем как «0260». Каждый раз оно присоединяется к другому маршрутизатору и получает адрес из имеющегося в распоряжении присоединяющего маршрутизатора диапазона адресов.

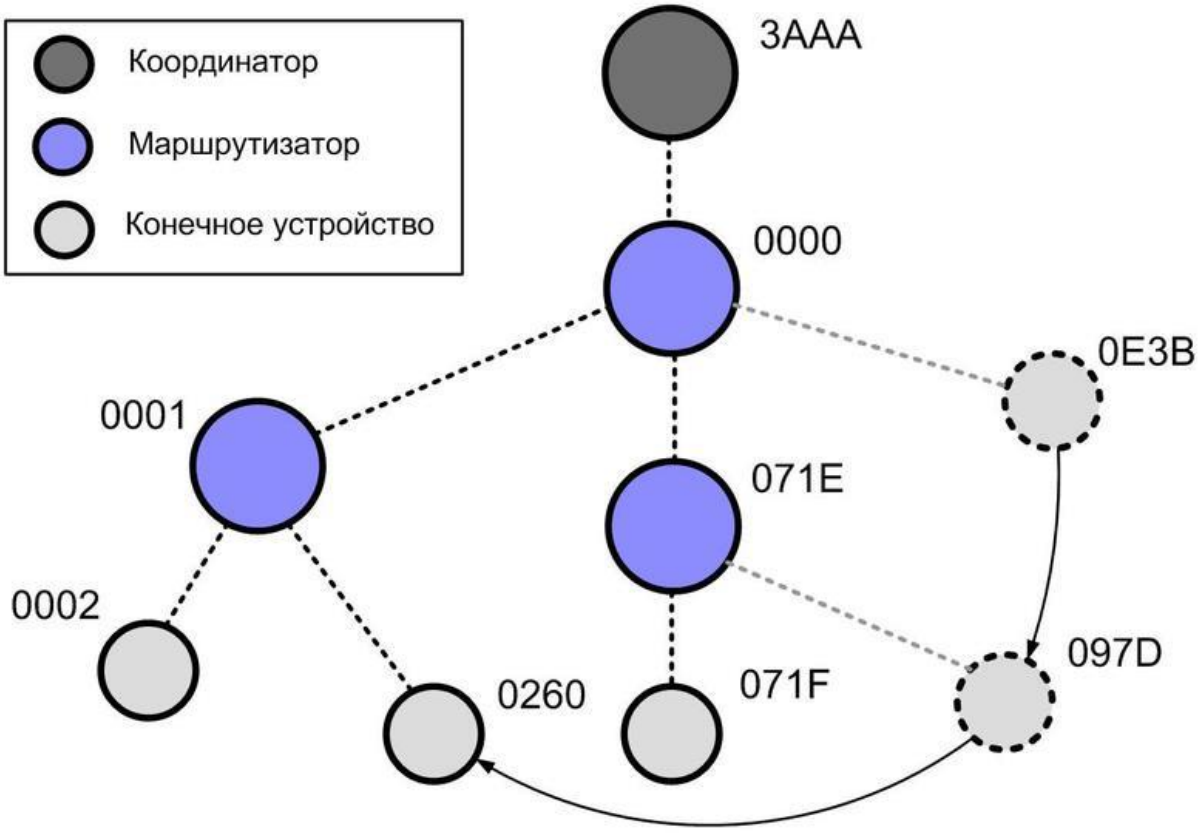


Рис.11 - Переподключение конечного устройства в древовидной сети

Таким образом, можно дать определение WPAN сетям в полном виде.

Беспроводная ad-hoc-сеть (беспроводная динамическая сеть, беспроводная самоорганизующаяся сеть) — децентрализованная беспроводная сеть, не имеющая постоянной структуры. Клиентские устройства соединяются «на лету», образуя собой сеть. Каждый узел сети пытается переслать данные, предназначенные другим узлам. При этом определение того, какому узлу пересылать данные, производится динамически, на основании связности сети. Это является отличием от проводных сетей и управляемых беспроводных сетей, в которых задачу управления потоками данных выполняют маршрутизаторы (в проводных сетях) или точки доступа (в управляемых беспроводных сетях).

Первыми беспроводными самоорганизующимися сетями были сети «packet radio» начиная с 1970-х годов, финансируемые DARPA после проекта ALOHAnet.

WLANs или WPANs?

WPAN определена в контексте личного рабочего пространства (POS - Personal Operating Space), которое обычно распространяется в радиусе до 10 метров и окружает человека или объект, находящийся в покое или в движении. WPAN также подразумевает низкую стоимость и **низкое энергопотребление**. Узел WPAN имеет небольшие размеры, что позволяет встраивать его в портативные устройства, такие как мобильные телефоны и КПК.

С другой стороны, WLAN представляет из себя систему более широкого радиуса действия и имеет более высокую сложность.

WLAN включает в себя центральный узел, называемый точкой доступа(AP - Access Point), который предоставляет доступ к каналу связи некоторому количеству конечных узлов. Типичный узел WLAN представляет из себя карту, устанавливаемую в персональные компьютеры и ноутбуки.

Локальная сеть представляет собой, по сути, сеть, используемую в одном здании или отдельном помещении, таком как квартира, для обеспечения взаимодействия используемых в них компьютеров и программ. Локальные сети, расположенные в разных зданиях, могут быть соединены между собой с помощью спутниковых каналов связи или волоконно-оптических сетей, что позволяет создать глобальную сеть, т.е. сеть, включающую в себя несколько локальных сетей.

Если используется одноранговая сеть, то все компьютеры, входящие в нее, имеют одинаковые права. Соответственно, любой компьютер может выступать в роли сервера, предоставляющего доступ к своим ресурсам, или клиента, использующего ресурсы других серверов.

В сети, построенной на архитектуре клиент/сервер, существует несколько основных компьютеров — серверов. Остальные компьютеры, которые входят в сеть, носят название клиентов, или рабочих станций.

ЛВС применяются для решения таких проблемы как:

Распределение данных. Данные в локальной сети хранятся на центральном ПК и могут быть доступны на рабочих станциях. В связи с этим не надо на каждом рабочем месте иметь накопители для хранения одной и той же информации.

Распределение ресурсов. Периферийные устройства могут быть доступны для всех пользователей ЛВС. Такими устройствами могут быть, например, сканер или лазерный принтер.

Распределение программ. Все пользователи ЛВС могут совместно иметь доступ к программам, которые были централизованно установлены на одном из компьютеров.

Существует ряд причин, для объединения отдельных персональных компьютеров в ЛВС:

Во-первых, совместное использование ресурсов позволяет нескольким ПК или другим устройствам осуществлять совместный доступ к отдельному диску (файл-серверу), принтерам, к сканерам и другому оборудованию, что снижает затраты на каждого отдельного пользователя.

Во-вторых, кроме совместного использования дорогостоящих периферийных устройств ЛВЛ позволяет аналогично использовать сетевые версии прикладного программного обеспечения.

В-третьих, ЛВС обеспечивает новые формы взаимодействия пользователей в одном коллективе, например работе над общим проектом.

В–четвертых, ЛВС дают возможность использовать общие средства связи между различными прикладными системами (коммуникационные услуги, передача данных и видеоданных, речи и т.д.).

Можно выделить **три принципа** LAN:

1) Открытость – возможность подключения дополнительных компьютеров и других устройств, а так же линий (каналов) связи без изменения технических и программных средств существующих компонентов сети.

2) Гибкость – сохранение работоспособности при изменении структуры в результате выхода из строя любого компьютера или линии связи.

3) Эффективность – обеспечение требуемого качества обслуживания пользователей при минимальных затратах.

У **локальной сети** есть следующие отличительные признаки:

- Высокая скорость передачи данных (до 10 Гб/с), большая пропускная способность;
- Низкий уровень ошибок передачи (высококачественные каналы передачи);
- Эффективный быстродействующий механизм управления обменом данных;
- Точно определенное число компьютеров, подключаемых к сети. В настоящее время трудно представить какую либо организацию без установленной в ней локальной сети, все организации стремятся модернизировать свою работу с помощью локальных сетей.

Так как учебно-теоретическое издание «Компьютерные сети. Интернет вещей и межмашинное взаимодействие» определено на интер-отклик систему взаимодействия с читателем с расстановкой приоритетов в подаче явного и неявного контента, то более детальную информацию по LAN вы сможете получить в учебном издании (QR-код представлен на текущей странице, как и содержимое) Уральского федерального университета им. Первого президента России Б.Н. Ельцина «Основы сетевых технологий» за авторством Руденкова Н.А. и Долинера Л.И.

Частный случай LAN: технологии семейства WLAN (IEEE 802.11) будут расмотрены более детально в Главе VI. Архитектура Internet of things (IoT).

1.2. Телекоммуникационные вычислительные сети....11
1.2.1. Общие понятия, терминология....11
1.2.2. Аппаратные и программные компоненты сети....11
1.2.3. Классификация информационно-вычислительных сетей....17

1.3. Топологии локальных вычислительных сетей....20
1.3.1. Физическая топология сети передачи данных....20
«Общая шина»....20
Топология «звезда»....21
Топология «кольцо»....22
Полносвязная топология....23
Ячеистая топология....23
Топология «дерево»....
1.3.2. Логическая топология сети передачи данных....
Разделение сети на логические сегменты....
Варианты создания VLAN....
Теги 802.1Q....
1.3.3. Сетевые устройства локальных сетей в топологии....
1.3.4. Пример построения простой информационно вычислительной сети....

CAN (Campus Area Network — кампусная сеть) — это группа локальных сетей, развернутых на компактной территории (кампусе) какого-либо учреждения и обслуживающие это учреждение - университет, промышленное предприятие, порт, оптовый склад и т.д. При этом сетевое оборудование (коммутаторы, маршрутизаторы) и среда передачи (оптическое волокно, медный завод, Cat5 кабели и др.) данных принадлежит арендатору или владельцу кампуса, предприятия, университета, правительства и так далее.

Кампусом называется группа компактно расположенных зданий или корпусов, например, промышленные предприятия, научные институты и вузы, студенческие городки, гостиничные комплексы, больницы. Для того чтобы создать единое информационное пространство организации, имеющей кампусную структуру, необходимо наличие сетевой интегрированной инфраструктуры, объединяющей отдельные здания.

На базе современных цифровых технологий пользователям кампусной сети предоставляются **следующие возможности**:

Поддержка мультимедийных приложений (голос, видео).

Поддержка широкополосных приложений (голосовые конференции, видео конференции, системы видеонаблюдения).

Увеличение емкости полосы пропускания для сетей, уже развернутых на базе традиционной технологии разделяемого Ethernet.

Поддержка любых новых приложений, которые могут внедряться в кампусе, независимо от специфики используемых сетевых протоколов.

Объединение пользователей с общими интересами (отдел, департамент), которые территориально расположены в различных частях кампуса, в единую виртуальную команду (виртуальную локальную сеть) и контроль доступа к информации, с которой работает группа.

Создание специализированных информационных центров, в которых происходит обработка и хранение данных, необходимых различным группам пользователей. Доступ к таким общим информационным ресурсам должен быть одинаково легким из любой точки кампуса.

Простой механизм подключения к единой сетевой инфраструктуре новых корпусов (зданий), не требующий перестройки существующей сетевой структуры. Высокая производительность и масштабируемость сетевой инфраструктуры, обеспечивающая растущие потребности пользователей в доступной полосе пропускания каналов связи.

Оперативное восстановление работоспособности сети при сбоях.

Сравнительно невысокая цена инсталляции и обслуживания сети при высокой надежности ее функционирования.

Правильно построенная кампусная сеть предприятия представляет собой **иерархическую структуру, состоящую их трех уровней**:

1. магистральный уровень (Core);
2. уровень распределения (Distribution);
3. уровень доступа (Access).

Уровень ядра – находится на самом верху иерархии и отвечает за надежную и быструю передачу больших объемов данных. Трафик, передаваемый через ядро, является общим для большинства пользователей. Сами пользовательские данные обрабатываются на уровне распределения, который, при необходимости, пересылает запросы к ядру.

Для уровня ядра большое значение имеет его отказоустойчивость, поскольку сбой на этом уровне может привести к потере связности между уровнями распределения сети.

Уровень распределения, который иногда называют уровнем рабочих групп, является связующим звеном между уровнями доступа и ядра. В зависимости от способа реализации, уровень распределения может выполнять следующие функции:

· обеспечение маршрутизации, качества обслуживания сети;

· агрегирование каналов;

· переход от одной технологии к другой (от 100Base-TX к 1000Base-T).

Уровень доступа занимается предоставлением доступа пользователей и рабочих групп к ресурсам объединенной сети. Основной задачей уровня доступа является создание точек входа/выхода пользователей в сеть. Уровень выполняет следующие функции:

· управление доступом пользователей и политиками сети;

· создание отдельных доменов коллизий (сегментация);

· подключение рабочих групп к уровню распределения;

· использование технологии коммутируемых локальных сетей.

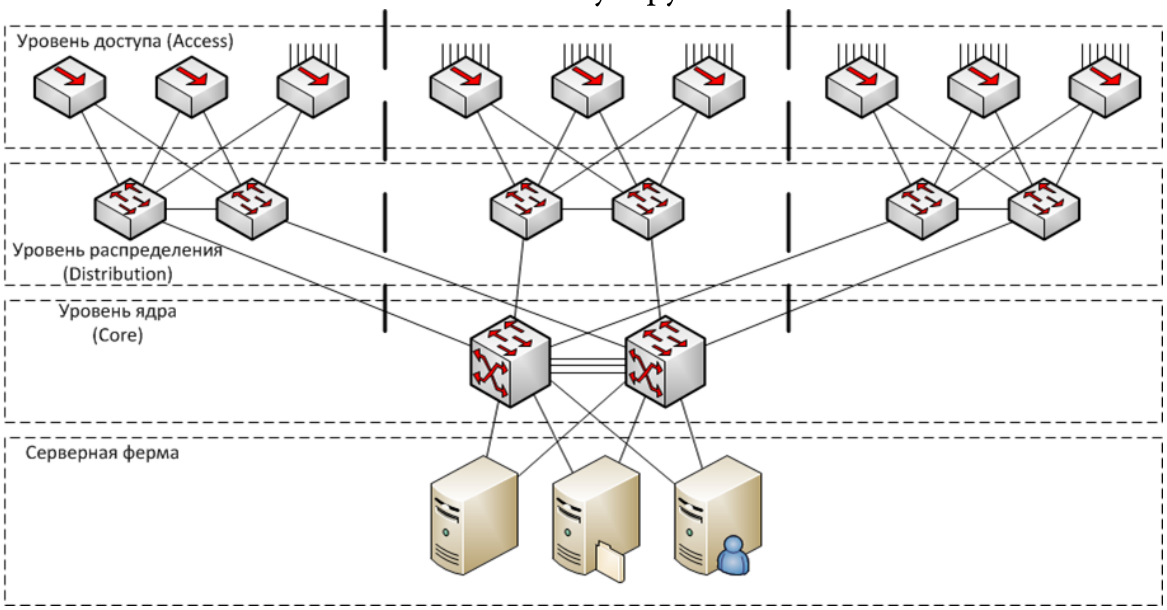


Рис. 12 – Иерархическая структура CAN

Такой подход к описанию CAN дает возможность выбрать оборудование, наиболее точно удовлетворяющее функциональным потребностям конкретной сетевой структуры.

Современная сетевая инфраструктура является разделяемым ресурсом, обеспечивающим работу широкого спектра информационных подсистем.

Эффективность и безопасность деятельности любого современного предприятия напрямую зависит от качества работы информационной инфраструктуры, отвечающей за обслуживание технологических и бизнес процессов.

При проектировании кампусной сети необходимо использовать модульный подход. Данный подход позволяет сформулировать требования по функциональности и защищенности для каждого модуля по отдельности. Построение сети связи с использованием модульного подхода обеспечивает большую гибкость решения и удобство при дальнейшем масштабировании.

Как правило, современная информационная инфраструктура предприятия состоит из следующих составных блоков (рис.13).

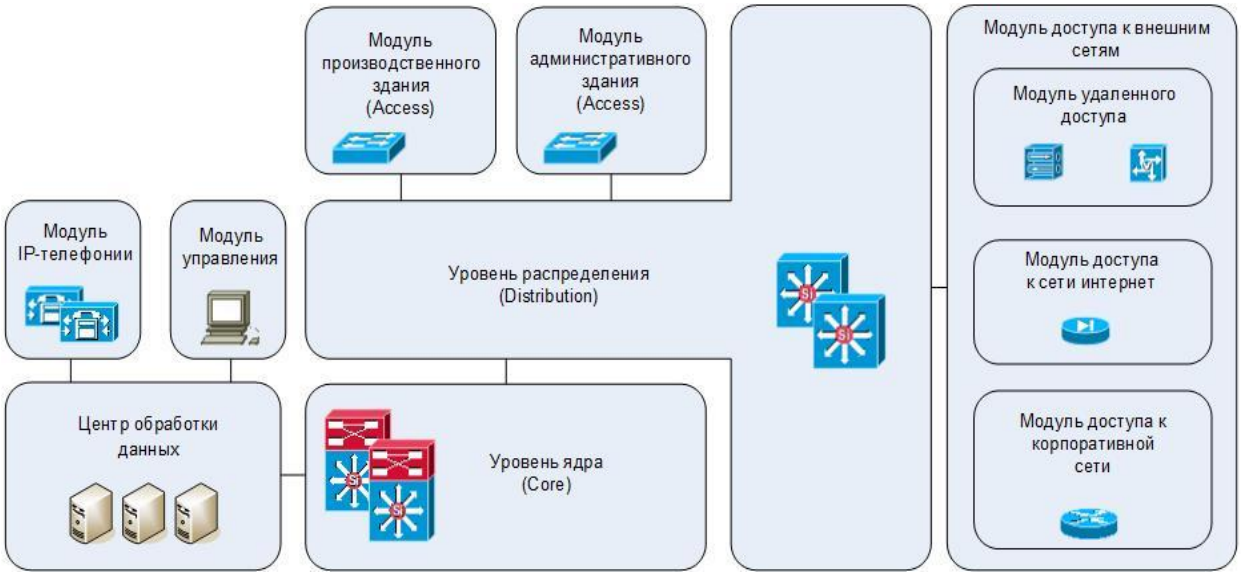


Рис. 13 – Логическая (модульная) схема типовой CAN-сети

1. Модуль управления сетью;
2. Модуль IP-телефонии;
3. Центр обработки данных;
4. Модуль производственного здания (включает в себя коммутатор доступа, рабочие станции и технологические контроллеры);
5. Модуль административного здания (включает в себя один или несколько коммутаторов и рабочие станции);
6. Модуль распределения;
7. Модуль ядра сети;

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ КАМПУСНОЙ СЕТЕВОЙ ИНФРАСТРУКТУРЫ.

! Магистральный уровень.

Здания или различные части кампуса объединяются высоконадежной коммутируемой магистралью. Высокая надежность ядра достигается за счет резервирования соединений между магистральными сетевыми устройствами и резервирования соединений, идущих от корпусов к ядру сети.

В составе ядра сети используются высокопроизводительные **коммутаторы третьего уровня сетевой модели OSI/ISO**, которые обеспечивают:

маршрутизацию на скорости среды передачи данных (десятки гигабит в секунду); быстрое восстановление при сбоях; балансировку нагрузки;

дополнительные сервисы, например безопасность, диагностику, управление, поддержку мультимедийных приложений в масштабе всей сети, что реализуется за счет встроенного интеллектуального программного обеспечения.

Для использования в качестве среды передачи данных наиболее эффективны две технологии, взаимно дополняющие друг друга и способные удовлетворить требованиям практически любого кампусного дизайна: оптоволоконные линии связи

Оптоволоконные линии связи могут связать кампусные городки, территориально разнесенные на десятки километров, и увеличить пропускную способность ядра сети до десятков гигабит в секунду. Экономически наиболее выгодно использовать оптоволоконные линии связи в компактных кампусах, где требуется обеспечить полную скорость доступа пользователей в любую точку кампуса.

Использование стандартных технологий **10-40GE** или Gigabit Ethernet позволяет гибко регулировать стоимость конечного решения в зависимости от эффективного радиуса кампуса: от сотен метров до 100 км.

Радиодоступ используется в случаях, когда требуется соединить корпуса, между которыми прокладывать физическую проводку неоправданно дорого или просто нецелесообразно, например, для подключения небольших корпусов, складских по мещений.

Идеальным решением в этом случае становится технология радио-Ethernet (все семейство беспроводных стандартов Ethernet IEEE 802.11), предоставляющая следующие возможности:

передавать данные на скорости до 1000 Мбит/с на расстояние до 50 км; быстро организовывать канал связи: для соединения двух корпусов достаточно установить два радиомоста, соединенных с коммутаторами в зданиях.

! Уровень распределения.

На уровне распределения обычно располагаются центральные коммутаторы здания – чаще всего высокопроизводительные коммутаторы. С ядром сети эти коммутаторы соединяются агрегированными каналами Gigabit Ethernet, 10G Ethernet. Каналы, ведущие к ядру сети, дублируются, осуществляется балансировка загрузки основных и дублирующих каналов. Коммутаторы рабочих групп соединяются с коммутаторами здания агрегированными каналами Gigabit Ethernet или Fast Ethernet.

Серверная группа (ферма) (рис.12) обычно располагается в одном из центральных корпусов, которые снабжены несколькими физическими каналами связи с другими корпусами кампуса. Коммутаторы, подключенные к серверам, поддерживают режим балансировки нагрузки на серверы. Для крупных серверных групп также используются дополнительные устройства кэширования информации, которые позволяют снизить загрузку серверов.

! Уровень доступа.

На уровне доступа используются коммутаторы второго уровня, предназначенные для рабочих групп. Они снабжаются высокоскоростными портами Fast Ethernet или Gigabit Ethernet (рис.14), служащими для подключения к центральным коммутаторам здания. Соединения с центральными коммутаторами здания резервируются. Возможно изменение баланса нагрузки, что позволяет повысить эффективность использования резервного канала связи.

Название	Скорость	Кабель	Стандарт
Ethernet	10 Мб/с	«Толстый», «тонкий» коаксиал, Витая пара	802.3
Fast Ethernet	100 Мб/с	Витая пара, оптика	802.3u
Gigabit Ethernet	1 Гб/с	Витая пара, оптика	802.3z, 802.3ab
10G Ethernet	10 Гб/с	Витая пара, оптика	802.3ae, 802.3an

Рис 14. – Типы стандартов Ethernet

Рабочие места пользователей подключаются к коммутируемым портам Fast Ethernet. Высокопроизводительные рабочие станции и серверы могут подключаться к коммутатору с помощью агрегированных каналов Fast Ethernet или по каналу Gigabit Ethernet. Агрегированные каналы позволяют плавно увеличивать производительность сетевого соединения путем объединения нескольких физических интерфейсов в один логический.

Кампусные сети, должны в обязательном порядке проектироваться с учетом требованиям к СКС (структурированным кабельным системам), описанным в ГОСТ Р 53245-2008 «Информационные технологии (ИТ). Системы кабельные структурированные» и ГОСТ Р 53246-2008 «Монтаж основных узлов системы. Методы испытания» и сопутствующих документах.

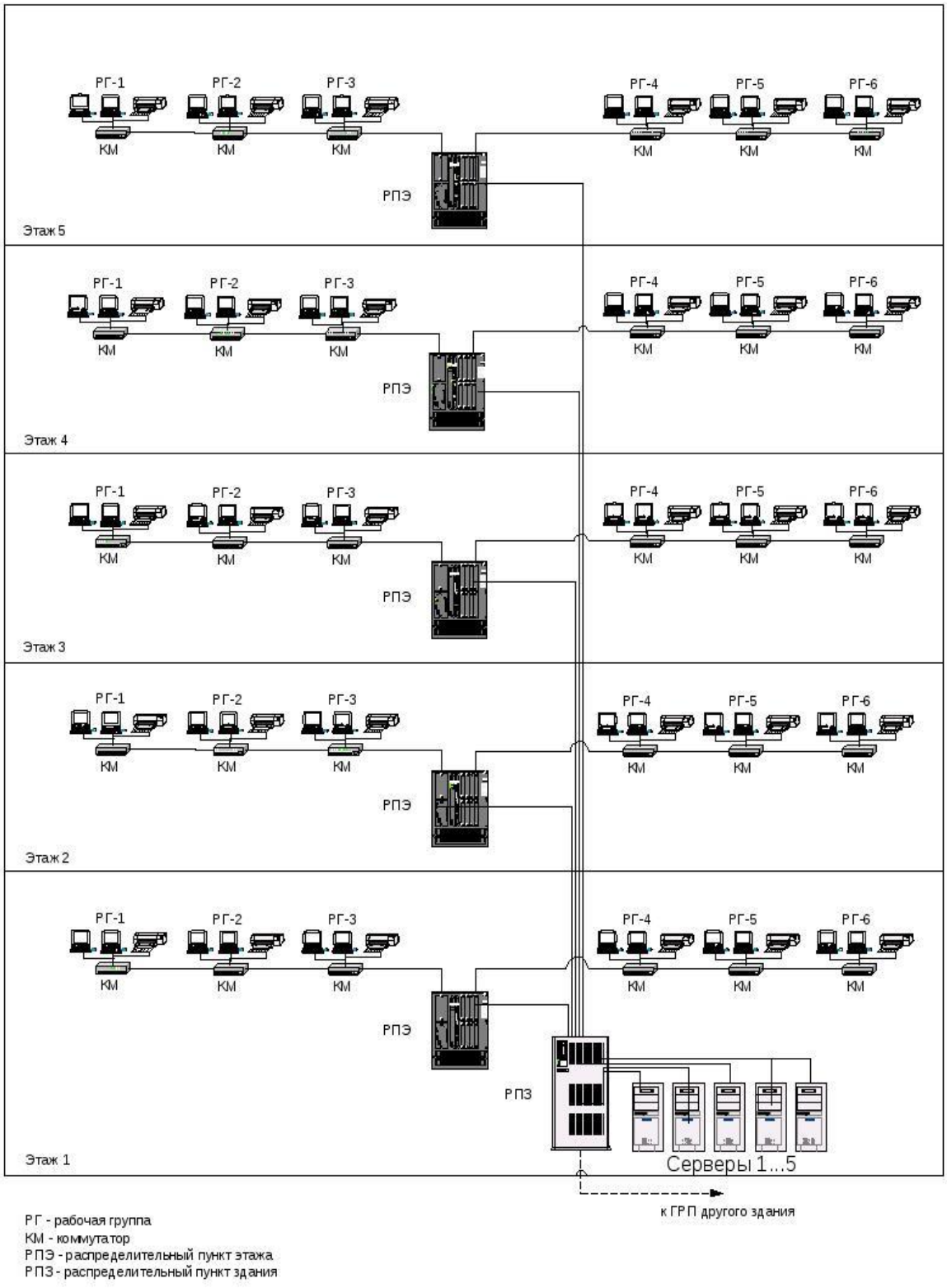


Рис. 15 – Принципиальная схема структурированной кабельной системы CAN

MAN: городские сети.

Городская вычислительная сеть (Metropolitan area network, MAN) (от англ. «сеть крупного города») объединяет компьютеры в пределах города, представляет собой сеть, по размерам меньшую, чем WAN, но большую, чем LAN. Является частным случаем совокупности CAN, объединенных между собой.

Самым простым примером городской сети является система кабельного телевидения. Она стала правопреемником обычных антенных сетей в тех местах, где по тем или иным причинам качество эфира было слишком низким. Общая антенна в этих системах устанавливалась на вершине какого-нибудь холма, и сигнал передавался в дома абонентов через кабельные сети.

Когда Интернет стал привлекать к себе массовую аудиторию, операторы кабельного телевидения поняли, что, внеся небольшие изменения в систему, можно сделать так, чтобы по тем же каналам в неиспользуемой части спектра передавались (причём в обе стороны) цифровые данные. С этого момента кабельное телевидение стало постепенно превращаться в MAN.

MAN — это не только кабельное телевидение. Недавние разработки, связанные с высокоскоростным беспроводным доступом в Интернет, привели к созданию других MAN, которые описаны в стандарте IEEE 802.16, описывающем широкополосные беспроводные ЛВС.

MAN (Metropolitan Area Network) — **опорная сеть провайдера**. То есть точки, связанные скоростными каналами. Расстояние — от 1 до 10 км. Это ещё не WAN, но точно MAN-решения.

MAN применяется для объединения в одну сеть группы сетей, расположенных в разных зданиях. В диаметре такая сеть может составлять от 5 до 50 километров.

Как правило, MAN не принадлежит какой-либо отдельной организации, в большинстве случаев её соединительные элементы и прочее оборудование принадлежит группе пользователей или же провайдеру, кто берёт плату за обслуживание. Об уровне обслуживания заранее договариваются и обсуждают некоторые гарантийные обязательства.

MAN часто действует как высокоскоростная сеть, чтобы позволить совместно использовать региональные ресурсы (подобно большой CAN). Это также часто используется, чтобы обеспечить общедоступное подключение к другим сетям, используя связь с WAN (глобальной сетью).

Информационные технологии — широкий класс дисциплин и областей деятельности, относящихся к технологиям создания, сохранения, управления и обработки данных, в том числе с применением вычислительной техники. В последнее время под информационными технологиями чаще всего понимают компьютерные технологии. В частности, ИТ имеют дело с использованием компьютеров и программного обеспечения для создания, хранения, обработки, ограничения к передаче и получению информации.

Согласно определению, принятому ЮНЕСКО, ИТ - это комплекс взаимосвязанных научных, технологических, инженерных дисциплин, изучающих методы эффективной организации труда людей, занятых обработкой и хранением информации; вычислительная техника и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы.

Сами ИТ требуют сложной подготовки, больших первоначальных затрат и наукоемкой техники. Их внедрение должно начинаться с создания математического обеспечения, моделирования, формирования информационных хранилищ для промежуточных данных и решений. **Телекоммуникации** - это любые формы связи, способы передачи информации на большие расстояния. Телекоммуникации - это также процессы передачи, получения и обработки информации на расстоянии с применением электронных, электромагнитных, сетевых, компьютерных и информационных технологий. Человечество занималось обработкой информации тысячи лет, а первые информационные технологии основывались на использовании счетов и книгопечатания. Ускорившееся за последние 40 лет развитие информационной технологии в первую очередь связано с появлением компьютеров. Успехи интегральной микроэлектроники обусловили ее проникновение почти во все стороны повседневной жизни, а также привели к многообразному переплетению различных ее отраслей.

Узкий смысл термина «информационная технология» определился к концу 1970-х годов, когда его стали употреблять в связи с использованием современной электронной техники для обработки информации. Информационная технология охватывает всю вычислительную технику и технику связи и отчасти — бытовую электронику, телевизионное и радиовещание. Она находит применение в промышленности, управлении, торговле, образовании, медицине, науке и военной сфере.

В последние 15 лет довольно популярным стало понятие новая информационная технология. Наблюдаются различные подходы к трактовке этого термина.

Под новыми информационными технологиями понимают совокупность внедряемых («встраиваемых») в системы организационного управления принципиально новых методов, способов и средств обработки данных, представляющих собой целостные технологические системы и обеспечивающих целенаправленное создание обработки, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с наименьшими затратами и в соответствии с закономерностями той социальной среды, где развивается эта информационная технология. Новейшие информационные технологии- это специальные термин, характеризующий использование новейших для данного этапа развития достижений науки и техники в области информатизации. Понятие «новая» является относительным и может использоваться на определенном отрезке времени. Так называемую «новизну» информационной технологии придает использование принципиально новых методов и средств преобразования информации.

Принципиальное значение современной информационной технологии состоит в замене машинно-бумажного процесса обработки данных на безбумажный, в котором не только не используются промежуточные носители данных, но и снижается объем фиксации данных на обычных документах. В подобной технологии впервые наблюдается феноменальное явление - процессы обработки информации отделены от процесса переноса массы. Только при обмене между человеком и машиной могут использоваться (но не обязательно) механические перемещения устройств.

Объектом исследования в информационной технологии являются не механические и программные средства, а деятельность человека, т.е. взаимодействие его в системе: **человек — ЭВМ — социальная среда**. Речь идет о создании и преобразовании моделей человеко-машинных систем. В этих моделях деятельность по созданию, использованию и совершенствованию сливается воедино и неразрывно взаимосвязана.

Предметом исследования выступают закономерности становления и развития методов информационной технологии, а также закономерности построения и функционирования средств ее реализации.

В настоящее время **информационная технология** обрела три наиболее характерные функции:

1)персонализация вычислений на основе компьютерных систем и систем интеллектуального интерфейса конечного пользователя с ПК;

2) использование баз данных и баз знаний;

3) применение вычислительных сетей.

3) применение встроенных компьютерных систем;

Эти функции реализуются посредством создания универсальных и специализированных информационных (информационно-технических, информационно-технологических) систем и комплексов.

История развития информационной технологии:

Принципиальное отличие информационной технологии от производственной состоит в следующем:

Информационная технология не может быть непрерывной , так как она соединяет работу рутинного типа (счетоводство, снятие копий, оперативный учет, и т.п.) и работу творческую, не поддающуюся пока формализации (принятие решений). Технология производства непрерывна и отражает строгую последовательность всех операций для выпуска продукции (конвейеризация процесса). Используемые в производственной сфере технологические понятия (норма, норматив, технологический процесс и т.п.) могут быть в настоящее время распространены только на рутинные операции над информацией.

Из всех видов технологий информационная технология сферы управления предъявляет самые высокие требования к «человеческому фактору», оказывая принципиальное влияние на квалификацию работника, содержание его труда, физическую и умственную нагрузку, профессиональные перспективы и уровень социальных отношений. Социальный подход ко всем новациям в информационной технологии особенно важен при внедрении человеко-машинных систем и переносе достижений компьютерной революции из одной социальной сферы в другую.

1. Информационная технология в своем развитии прошла несколько этапов. До второй половины XIXв. основу информационной технологии составляли перо, чернильница и бухгалтерская книга. Коммуникация (связь) осуществлялась путем направления пакетов (депеш). Продуктивность информационной обработки была крайне низкой: каждое письмо копировалось отдельно вручную; помимо счетов, суммируемых также вручную, не было другой информации для принятия решений.

2. На смену «ручной» информационной технологии в конце XIX в. пришла «механическая». Изобретение пишущей машинки, телефона, диктофона, модернизация системы общественной почты –все послужило базой для принципиальных изменений в технологии обработки информации, и, как следствие, в продуктивность работы. По существу , «механическая» технология проложила дорогу к формированию организационной структуры существующих учреждений.

3. 40-60-е годы XX в. характеризуются появлением «электрической» технологии, основанной на широком использовании электрических пишущих машинок со съемными элементами, копировальных машин на обычной бумаге (типа ксерокса), портативных диктофонов.

Они улучшили учрежденческую деятельность за счет повышения качества, количества и скорости обработки документов. Многие современные учреждения базируются на «электрической» технологии.

4. Появление во второй половине 60-х годов больших производительных ЭВМ на периферии учрежденческой деятельности (в вычислительных центрах) позволило сместить акцент в информационной технологии на обработку не формы, а содержания информации. Это было началом формирования «электронной», или «компьютерной» технологии. Как известно, информационная технология управления должна содержать как минимум три важнейших компонента обработки информации: учет, анализ и принятие решений. Эти компоненты реализуются в «вязкой» среде – бумажном «море» документов, которое становится с каждым годом все более необъятным.

5. Сложившиеся в 60-х годах концепции применения АСУ не всегда и не в полной мере отвечают задаче совершенствования управления о неограниченных возможностях «кнопочной» информационной технологии. Методологически эти концепции вычислительной мощности систем АСУ и применении наиболее общих имитационных моделей, которые в ряде случаев далеки от реального механизма оперативного управления.

Название «автоматизированная система управления» не совсем корректно отражает функции, которые такие системы выполняют: точнее было бы «автоматизированная система обеспечения управления (АСОУ), ибо в существующих АСУ понятие «система» не включает решающего

Звена управления-пользователя. Игнорирование этого принципиального обстоятельства, по-видимому, привело к тому, что расширение сет АСУ и повышение мощности их вычислительных средств обеспечили благодаря большим массивам первичных данных улучшение в основном учетных функций управления (справочных, статистических, следящих). Однако учетные функции отражают только прошлое состояние объекта управления и не позволяют оценить перспективы его развития, т.е. обладают низким динамизмом. В других компонентах технологии управления наращивание мощности АСУ не дало ощутимого эффекта. Отсутствие развитых коммуникационных связей рабочих мест пользователя с центральной ЭВМ, характерный для большинства АСУ пакетный режим обработки данных, низкий уровень диалоговой поддержки – все это фактически не обеспечивает высокого качества анализа пользователями данных статистической отчетности и всего интерактивного уровня аналитической работы. Тем самым эффективность АСУ на нижних ступенях управленческой лестницы, т.е. именно там, где формируются информационные потоки, существенно падает вследствие значительной избыточности поступающей информации при отсутствии средств агрегирования данных.

Именно по этой причине, несмотря на ввод дополнительных систем АСУ, с каждым годом возрастает количества работников, занятых учетными функциями: на сегодняшний день шестую часть всех работников аппарата управления составляет учетно-бухгалтерский персонал.

6. Начиная с 70-х годов сформировалась тенденция перенесения центра тяжести с развития АСУ на фундаментальные компоненты информационной технологии (особенно на аналитическую работу) с максимальным применением человеко-машинных процедур. Однако по-прежнему вся эта работа проводилась на мощных ЭВМ, размещаемых централизованно в вычислительных центрах. При этом в основу построения подобных АСУ была положена гипотеза, согласно которой задачи анализа и принятия решений относились к классу формализуемых, поддающихся математическому моделированию. Предполагалось, что такие АСУ должны были повысить качества, полноту, подлинность и своевременность информационного обеспечения лиц, принимающих решения, эффективность работы которых будет возрастать благодаря увеличению числа анализируемых задач.

Однако внедрение подобных систем дало весьма отрезвляющие результаты. Оказалось, что применяемые математические модели имеют ограниченные возможности практического использования: аналитическая работа и процесс принятия решений происходят в отрыве от реальной ситуации и не подкрепляются комммуникационным процессом формирования. Для каждой новой задачи требуется новая модель, а поскольку модель создавалась специалистами по математическим методам, а не пользователем, то процесс принятия решений происходит как бы не в реальном масштабе времени, и теряется творческий вклад самого пользователя, особенно при решении нетиповых управленческих задач. При этом вычислительный потенциал управления, сосредоточенный в вычислительных центрах, находится в отрыве от других средств и технологий обработки информации вследствие неэффективной работы нижних ступеней т необходимости непрерывных конверсий информации. Это также понижает эффективность информационной технологии при решении задач на верхних ступенях управленческой лестницы. К тому же для сложившейся в АСУ организационной структуры технических средств характерны низкий коэффициент их использования, значительные сроки (не всегда выполняемые) проектирования автоматизированных систем и невысокая их рентабельность из-за слабого воздействия результатов автоматизации на эффективность управления.

7. С появлением ПК на «гребне микропроцессорной революции» происходит принципиальная модернизация идеи АСУ: от вычислительных центров и централизации управления к распределенному вычислительному потенциалу, повышению однородности технологии обработки информации и деценрализации управления.

Такой подход нашел свое воплощение в **системах поддержки принятия решения** (СППР) и **экспертных системах** (ЭС), которые характеризуют новый этап компьютеризации технологии организационного управления, по существу, -этап **персонализации АСУ**. **Системность** - основной признак СППР и признание того, что самая совершенная ЭВМ не может заменить человека. В данном случае речь идет о структурной человеко-машинной единице управления, которая оптимизируется в процессе работы: возможности ЭВМ расширяются за счет структуризации пользователем решаемых задач и пополнения ее базы знаний, а возможности пользователя- за счет автоматизации тех задач, которые ранее было нецелесообразно переносить на ЭВМ по экономическим или техническим соображениям. Становится возможным анализировать последствия различных решений и получать ответы на вопросы типа « что будет, если…?», не тратя времени на трудоемкий процесс программирования.

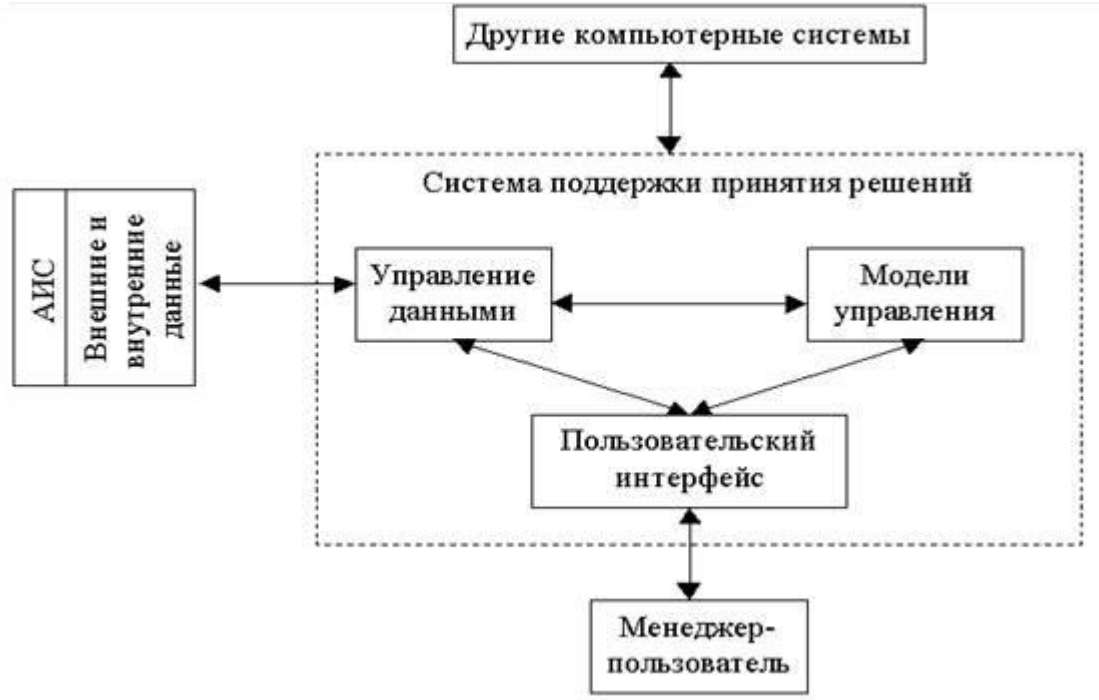


Рис. 16 - Система поддержки принятия решения

Важнейший аспект внедрения СППР (рис.16) и ЭС – рационализация повседневной деятельности работников управления. В результате их внедрения на нижних ступенях управления существенно укрепляется весь фундамент управления, уменьшается нагрузка на централизованные вычислительные системы и верхние ступени управления, что позволяет сосредоточить в них вопросы решения крупных долгосрочных стратегических задач. Естественно, что «компьютерная» технология. СППР должна использовать не только ЭВМ, но и другие современные средства обработки информации.

Концепция СППР требует пересмотра существующих подходов к управлению трудовыми процессами в учреждении. По существу на базе СППР формируемся новая человеко-машинная трудовая единица с квалификацией труда, его нормированием и оплатой. Она аккумулирует знания и умение конкретного человека (пользователя СППР) с интегрированными знаниями и умением, заложенным в ПЭВМ (экспертные системы, системы принятия решений, системы обеспечивающей технологии и др.).

В заключение раздела кратко остановимся на состоянии и тенденциях развития ИТ в США, странах Западной Европы, Японии можно охарактеризовать следующими тезисами:

1. Наличие большого количества промышленно функционирующих БД большого объема, содержащих информацию практически по всем видам деятельности общества.

2. Создание технологий, обеспечивающих интерактивный доступ массового пользователя к этим информационным ресурсам. Технической основой данной тенденции явились государственные и частные системы связи и передачи данных общего назначения и специализированные, объединенные в национальные, региональные и глобальные ИВС.

3. Расширение функциональных возможностей информационных систем, обеспечивающих параллельную одновременную обработку Баз Данных (БД) с разнообразной структурой данных, мультиобъектных документов, гиперсред, в том числе реализующих технологии создания и ведения гипертекстовых БД. Создание локальных, многофункциональных проблемно-ориентированных информационных систем различного назначения на основе мощных ПК и локальных сетей ПК.

4. Включение в информационные системы элементов интеллектуализации интерфейса пользователя с системами, экспертных систем, систем машинного перевода, индексирования информации и других технологических средств. Ведущие промышленно развитые страны имеют государственную политику в области развития ИТ и соответствующие программы, которые субсидируются правительством, государственными учреждениями, частными фирмами и ассоциациями.

§ IV. Микроархитектура компьютерных сетей.

I. Эталонный подход: Friend-to-friend и Peer-to-Peer обмен.

Архитектура сети – это набор параметров, правил, протоколов, алгоритмов, карт, которые позволяют изучать сеть.

Протокол – это набор семантических и синтаксических правил, определяющий поведение функциональных блоков сети или передачи данных. Другими словами, протокол – это совокупность соглашений относительно способа представления данных, обеспечивающего их передачу в нужных направлениях и правильную интерпретацию данных всеми участками процесса информационного обмена. Поскольку информационный обмен – это процесс многофункциональный, то протоколы делятся на уровни. За каждым уровнем закрепляется группа родственных функций. Для правильного взаимодействия узлов различных вычислительных сетей их архитектура должна быть открытой (известной). Существует несколько систем межсетевого взаимодействий, как например известная вам модель OSI/ISO, снискавшая массовое распространение во всем мире. Однако, модель OSI/ISO дает достаточно плоские, поверхностные представления, к самим сетям, к разности подхода обмена между конечными устройствами. Стек (реже именуемый моделью) протоколов TCP/IP еще более узконаправлен в своих функциях.

Сеть — базис, на который опирается всё, и производить изменения на ней довольно сложно — сервисы не терпят, когда сеть лежит, равно как и наоборот. Зачастую вывод из эксплуатации одного узла может сложить большую часть приложений и повлиять на много клиентов. Отчасти поэтому сетевой инжиниринг сопротивляться любым физическим изменениям — «потому что сейчас оно как-то работает (мы, возможно, даже не знаем как), а тут надо что-то новое настроить, и неизвестно как оно повлияет на сеть». А еще чаще необходимо расширить возможности передачи информации. И на помощь этому приходит **логическая архитектура передачи данных**:

Оверлейная сеть (от англ. Overlay Network) — общий случай **логической сети**, создаваемой поверх любой другой компьютерной сети. Узлы оверлейной сети могут быть связаны либо физическим соединением, либо логическим, для которого в основной сети существуют один или несколько соответствующих маршрутов из физических соединений.

Примерами оверлеев являются сети VPN и одноранговые сети, которые работают на основе интернета и представляют собой «надстройки» над классическими сетевыми протоколами, предоставляя широкие возможности, изначально не предусмотренные разработчиками основных протоколов. Коммутируемый доступ в интернет фактически осуществляется через оверлей (например, по протоколу PPP), который работает «поверх» обычной телефонной сети. Основное преимущество оверлейных сетей заключается в том, что они позволяют разрабатывать и эксплуатировать новые крупномасштабные распределённые сервисы без внесения каких-либо изменений в основные протоколы сети. Распространённым недостатком оверлеев являются повышенные затраты при передаче информации из-за дополнительного уровня обработки пакетов или неоптимальных маршрутов.

Одноранговая, децентрализованная, или пиринговая (англ. peer-to-peer, P2P — равный к равному) сеть — оверлейная компьютерная сеть, основанная на равноправии участников. Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера. В отличие от архитектуры клиент-сервера, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов. Участниками сети являются все пиры.

Впервые фраза «peer-to-peer» была использована в 1984 году при разработке архитектуры Advanced Peer to Peer Networking (APPN) фирмы IBM.

Устройство одноранговой сети

В сети присутствует некоторое количество машин, при этом каждая может связаться с любой из других. Каждая из этих машин может посылать запросы другим машинам на предоставление каких-либо ресурсов в пределах этой сети и, таким образом, выступать в роли клиента. Будучи сервером, каждая машина должна быть способной обрабатывать запросы от других машин в сети, отсылать то, что было запрошено. Каждая машина также должна выполнять некоторые вспомогательные и административные функции (например, хранить список других известных машин-«соседей» и поддерживать его актуальность).

Любой член данной сети не гарантирует своего присутствия на постоянной основе. Он может появляться и исчезать в любой момент времени. Но при достижении определённого критического размера сети наступает такой момент, что в сети одновременно существует множество серверов с одинаковыми функциями.

Идея классического peer-to-peer обмена заключается в том, что каждый peer знает и поддерживает информацию о других участниках. Когда новый клиент подключается к сети, он может узнать у любого пира информацию о том, где и какие файлы сейчас доступны.

Когда клиент начинает скачивает файл себе на компьютер, то скачанные части этот файла сразу становятся доступны для скачивания другим пользователям. Никто не даёт гарантию, что каждый сервер будет находиться длительное время в сети и давать скачивать информацию, напротив - ситуация, когда сервер пропадает в процессе загрузки, является естественной. В данном случае будет найден новый сервер, который может продолжить передачу данных.

Для поддержания списка активных peer-ов каждый сервер посылает другим серверам heartbeat. **Heartbeat (удар сердца)** - это сообщение, которое один сервер посылает другому, чтобы сказать ему, что он жив. Соответственно, если heartbeat долго не приходит, значит этот сервер нужно удалить из списка активных peer-ов. Постоянно обмениваться heartbeat-ом с большим количеством серверов трудоёмко. Поэтому у каждого сервера есть два параметра -- нижняя и верхняя граница на размер списка активных серверов. Когда это количество становится ниже нижней границы, запускается поиск новых участников. Сервер запрашивает у других серверов список активных peer-ов и добавляет некоторых из них в свой список, но при этом следит за тем, чтобы размер списка не превысил верхнюю границу.

В некоторых peer-to-peer сетях, кроме равноправных узлов, присутствуют сервера, которые выполняют административные функции, такие как поддержка базы онлайн пользователей. К частично децентрализованным сетям относятся, например, eDonkey, BitTorrent, Direct Connect, The Onion Router.

Одна из областей применения технологии одноранговых сетей — обмен файлами. Пользователи файлообменной сети выкладывают какие-либо файлы в папку общего доступа («расшаренную» от англ. share — делиться) на своём компьютере, содержимое которой доступно для скачивания другим пользователям. Какой-нибудь другой пользователь сети посылает запрос на поиск какого-либо файла. Программа ищет у клиентов сети файлы, соответствующие запросу, и показывает результат. После этого пользователь может скачать файлы у найденных источников. В современных файлообменных сетях информация загружается сразу из нескольких источников. Её целостность проверяется по контрольным суммам.

Friend-to-friend (друг-к-другу, F2F) — разновидность одноранговой сети (P2P), в которой пользователи устанавливают прямые соединения только с заранее выбранными пользователями (друзьями, англ. friend). Для аутентификации могут использоваться цифровые подписи или пароли.

В отличие от других типов приватных P2P-сетей, пользователи F2F-сети не знают, кто за пределами их круга друзей пользуется сетью. Этим обеспечивается анонимность пользователей. RetroShare, GNUnet и Freenet — примеры ПО, на основе которого можно создать F2F-сеть (GNUnet по умолчанию не настроен для работы в режиме F2F-сети). Термин «friend-to-friend-сеть» (F2F-сеть) был предложен Даниэлем Бриклином в 2000 году.

Преимущества F2F

1. Использование F2F-сетей позволяет избегать атак типа MITM, то есть пользователи могут без опасений обмениваться секретными данными (например, крипто-ключами) со своими друзьями. **Атака посредника**, или атака «человек посередине» (англ. Man in the middle (MITM)) — вид атаки в криптографии и компьютерной безопасности, когда злоумышленник тайно ретранслирует и при необходимости изменяет связь между двумя сторонами, которые считают, что они непосредственно общаются друг с другом. Является методом компрометации канала связи, при котором взломщик, подключившись к каналу между контрагентами, осуществляет вмешательство в протокол передачи, удаляя или искажая информацию. Одним из примеров атак типа «человек посередине» является активное прослушивание, при котором злоумышленник устанавливает независимые связи с жертвами и передаёт сообщения между ними. Тем самым он заставляет жертв поверить, что они разговаривают непосредственно друг с другом через частную связь, фактически же весь разговор управляется злоумышленником. Злоумышленник должен уметь перехватывать все передаваемые между двумя жертвами сообщения, а также вводить новые. В большинстве случаев это довольно просто, например, злоумышленник может вести себя как «человек посередине» в пределах диапазона приёма беспроводной точки доступа (Wi-Fi)

2. При использовании F2F-сетей пользователь может настроить фаерволл так, чтобы доступ к порту программы, обеспечивающей подключение к сети F2F, был разрешён только друзьям (так как IP-адреса друзей заведомо известны). Благодаря этому случайные люди не смогут доказать, что с IP-адреса пользователя можно было получить доступ к обсуждаемым файлам.

3. Поскольку программы, обеспечивающие подключение к сети F2F (как например Gnutella на рисунке 17), шифруют данные, передаваемые между соседними узлами сети, и используют неполное шифрование при передаче данных между оконечными точками, пользователи промежуточных узлов могут отслеживать, какого рода файлы передаются через них.

4. То, что соединения возможны только между доверенными узлами (между друзьями), защищает пользователей от взломщиков, которые могли бы использовать уязвимость.

Недостатки F2F-обмена:

В настройках программного обеспечения, обеспечивающей подключение к сети F2F, нужно вручную указывать список всех своих друзей. Ситуация усугубляется, если пользователь хочет опробовать несколько различных программ. Обычно не так много друзей (пиров (peer)) готовы участвовать в сети в режиме 24/7.

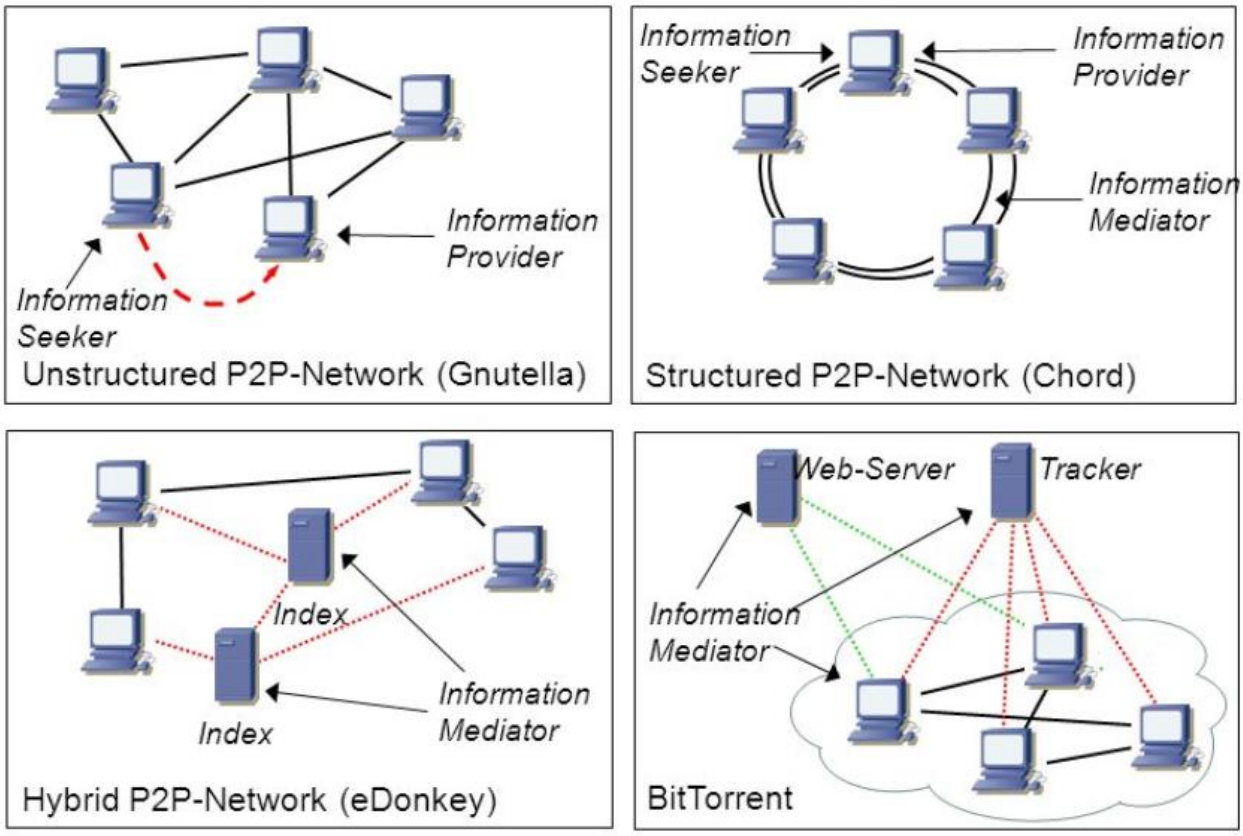


Рис. 17 – P2P/F2F решения

§ V. Беспроводные сенсорные сети.

I. **Интеллектуальные системы на базе сенсорных сетей.**

Беспроводная сенсорная сеть, или беспроводная датчиковая сеть, — **распределённая, самоорганизующаяся** сеть множества **датчиков** и **исполнительных** устройств, объединённых между собой посредством радиоканала. Область покрытия подобной сети может составлять от нескольких метров до нескольких километров за счёт способности ретрансляции сообщений от одного узла к другому.

Одним из первых прообразов сенсорной сети можно считать **SOSUS**, предназначенную для обнаружения и идентификации подводных лодок.

В середине 1990-х годов технологии беспроводных сенсорных сетей стали активно развиваться, в начале 2000-х годов развитие микроэлектроники позволило производить для таких устройств достаточно дешёвую элементную базу. Беспроводные сенсорные сети начала 2010-х годов в основном базируются на стандарте **ZigBee,** который будет рассматриваться в следующей главе. .

Многие отрасли и сферы деятельности (промышленность, транспорт, коммунальное хозяйство, охрана) заинтересованы во внедрении сенсорных сетей, и число потребителей непрерывно увеличивается. Тенденция обусловлена усложнением технологических процессов, развитием производства, расширяющимися потребностями частных лиц в сегментах безопасности, контроля ресурсов и использования товаро-материальных ценностей. С развитием микроэлектронных технологий появляются новые практические задачи и теоретические проблемы, связанные с применениями сенсорных сетей в промышленности, жилищно-коммунальном комплексе, домашних хозяйствах.

Использование недорогих **беспроводных сенсорных устройств** контроля параметров открывает новые области для применения систем телеметрии и контроля, такие как: Своевременное выявление возможных отказов исполнительных механизмов, по контролю таких параметров, как вибрация, температура, давление и т. п.; Контроль доступа к удалённым системам объекта мониторинга в режиме реального времени; обеспечение охраны музейных ценностей; обеспечение учёта экспонатов; автоматическая ревизия экспонатов; Автоматизация инспекции и технического обслуживания промышленных активов; Управление коммерческими активами; Применение как компоненты в энерго- и ресурсосберегающих технологий; Контроль экологических параметров окружающей среды.

Беспроводные сенсорные сети состоят из миниатюрных вычислительных устройств — **мотов**, снабжённых датчиками (например, температуры, давления, освещённости, уровня вибрации, местоположения и т. п.) и передатчиками, работающими в заданном радиодиапазоне. Гибкая архитектура, снижение затрат при монтаже выделяют беспроводные сети интеллектуальных сенсоров среди других беспроводных и проводных интерфейсов передачи данных, особенно когда речь идет о большом количестве соединенных между собой устройств, сенсорная сеть позволяет подключать до 65 000 устройств. Постоянное снижение стоимости беспроводных решений, повышение их эксплуатационных параметров позволяют постепенно перейти с проводных решений в системах сбора телеметрических данных, средств дистанционной диагностики, обмена информации на беспроводные.

Технология ретранслируемой ближней радиосвязи 802.15.4/ZigBee, известная как «Сенсорные сети», является одним из современных направлений развития самоорганизующихся отказоустойчивых распределенных систем наблюдения и управления ресурсами и процессами. Сегодня технология беспроводных сенсорных сетей, является единственной беспроводной технологией, с помощью которой можно решить задачи наблюдения и контроля, которые критичны к времени работы сенсоров. Объединённые в беспроводную сеть датчики образуют территориально-распределённую самоорганизующуюся систему сбора, обработки и передачи информации.

Основной областью применения является контроль и наблюдение измеряемых параметров физических сред и предметов. Принятый стандарт IEEE 802.15.4 описывает контроль доступа к беспроводному каналу и физический уровень для низкоскоростных беспроводных личных сетей, то есть два нижних уровня согласно сетевой модели OSI.

Сенсорные сети могут состоять из различных типов датчиков, например, сейсмических, датчиков определения магнитного поля, тепловых, инфракрасных, акустических, которые в состоянии осуществлять самые разнообразные измерения условий окружающей среды.

Военное применение

Беспроводные сенсорные сети могут быть неотъемлемой частью военного управления, связи, разведки, наблюдения и систем ориентирование (C4ISRT). Быстрое развертывание, самоорганизации и отказоустойчивость – это характеристики сенсорных сетей, которые делают их перспективным инструментом для решения поставленных задач.

Поскольку сенсорные сети могу быть основаны на плотном развертывании одноразовых и дешевых узлов, то уничтожение некоторых их них во время военных действий не повлияет на военную операцию так, как уничтожение традиционных датчик. Поэтому использование сенсорных сетей лучше подходит для сражений. Перечислим еще некоторые способы применение таких сетей: мониторинг вооружения и боеприпасов дружественных сил, наблюдение за боем; ориентация на местности; оценка ущерба от битв; обнаружение ядерных, биологических и химических атак. Мониторинг дружественных силы, вооружения и боеприпасов: лидеры и командиры могут постоянно контролировать состояние своих войск, состояние и наличие оборудования и боеприпасов на поле боя с помощью сенсорных сетей. К каждому транспортному средству, оборудованию и важным боеприпасам могут быть прикреплены датчики, которые сообщают их статус. Эти данные собирается вместе в ключевых узлах, и направляются руководителям.

Данные также могут быть переадресованы на верхние уровни иерархии командования для объединения с данными из других частей. Наблюдения боя: критические участки, пути, маршруты и проливы могут быть быстро покрыты сенсорными сетями для изучения деятельности сил противника. Во время операций или после разработки новых планов сенсорные сети могут быть развернуты в любое время для наблюдения за боем. Разведка сил противника и местности: Сенсорные сети могут быть развернуты на критических территориях, и могут быть собраны в течении нескольких минут ценные, подробные и своевременные данные о силах противника и местности, прежде чем враг сможет их перехватить. Ориентация: сенсорные сети могут быть использованы в системах наведения интеллектуальных боеприпасов. Оценка ущерба после боя: непосредственно перед или после нападения, сенсорные сети могут быть развернуты в целевой области для сбора данных об оценке ущерба. Обнаружение ядерных, биологических и химических атак: при применении химического или биологического оружия, использование которого близко к нулю, важное значение иметь своевременное и точное определение химических агентов.

Могут быть использованы сенсорные сети в качестве систем предупреждения химических или биологических атак и данные собранные в короткие сроки помогут резко уменьшить количество жертв. Также можно использовать сенсорные сети для подробной разведки, после обнаружения таких атак. Например, можно осуществлять разведку в случае радиационных заражений, не подвергая людей радиации.

Экологическое применение

Некоторые из направлений в экологии, где применяют сенсорные сетей: отслеживание движения птиц, мелких животных и насекомых; мониторинг состояния окружающей среды, с целью выявления ее влияния на сельскохозяйственные культуры и скота; орошения; широкомасштабный мониторинга земли и исследования планет; химическое / биологическое обнаружение; обнаружение лесных пожаров; метеорологические или геофизические исследования; обнаружение наводнений; и исследование загрязнения. Обнаружение лесных пожаров: поскольку моты могут быть стратегически и плотно развернуты в лесу, то они могут ретранслировать точное происхождение огня до того, как пожар станет неконтролируемым. Миллионы датчик могут быть развернуты на постоянной основе. Они могут быть оснащены солнечными батареи, т.к. узлы могут быть оставлены без присмотра на месяцы и даже годы.

Моты будут работать сообща для выполнения задач распределенного зондирования и преодоления препятствий, таких как деревья и скалы, которые блокируют работу проводных датчиков. Отображение биосостояния окружающей среды [требует сложных подходов к интеграции информации во временных и пространственных масштабах. Прогресс в области технологии дистанционного зондирования и автоматизированный сбор данных, позволили значительно снизить затраты на исследования. Преимущество данных сетей в том, что узлы могут быть соединены с Интернетом, который позволяет удаленным пользователям осуществлять контроль, мониторинг и наблюдения за окружающей средой.

Хотя спутниковые и бортовые датчики являются полезными в наблюдении за большим разнообразием, например, пространственной сложности видов доминирующих растений, они не позволяют наблюдать за мелкими элементами, которые составляет большую часть экосистемы. В результате возникает потребность в развертывании на местах узлов беспроводных сенсорных сетей.

Одним из примеров применения - это составление биологической карты окружающей среды в заповеднике в Южной Калифорнии. Три участка покрыты сетью, в каждой их которых по 25-100 узлов, которые используются для постоянного наблюдения за состоянием окружающей среды. Обнаружение наводнений: примером обнаружения наводнений является система оповещения в США.

Несколько типов датчиков, размещенных в системе оповещения, определяют уровень осадков, уровень воды и погоду.

Научно-исследовательские проекты, такие как COUGAR Device Database Project в Корнельском университете и проект DataSpace в Университете Rutgers, изучают различные подходы к взаимодействию с отдельными узлами в сети для получения снимков и долго собираемых данных. Сельское хозяйство: преимуществом сенсорных сетей также является возможность контролировать уровень пестицидов в воде, уровень эрозии почвы и уровень загрязнения воздуха в режиме реального времени.

Применение в медицине

Одним из применений в медицине является устройства для инвалидов; мониторинг пациентов; диагностика; мониторинг использования медикаментов в больницах; сбор физиологических данных человека; и мониторинга врачей и пациентов в больницах. Мониторинг физиологического состояния человека: физиологические данные, собранные сенсорными сетями могут храниться в течение длительного периода времени и могут использоваться для медицинского исследования. Установленные узлы сети могут также отслеживать движения пожилых людей и, например, предупреждать падения. Эти узлы невелики и обеспечивают пациенту большую свободу передвижения, в тоже время позволяют врачам выявить симптомы болезни заранее. Кроме того, они способствуют обеспечению более комфортной жизни для пациентов в сравнении с лечением в больнице. Для проверки возможности такой системы на факультете медицины Grenoble–France был создан “Здоровый умный дом ".

Мониторинг врачей и пациентов в больнице: каждый пациент имеет небольшой и легкий узел сети. Каждый узел имеет свою конкретную задачу. Например, один может следить за сердечным ритмом, в то время как другой снимает показания кровяного давления. Врачи могут также иметь такой узел, он позволит другим врачам найти их в больнице. Мониторинг медикаментов в больницах: Узлы могут быть присоединены к лекарствам, тогда шансы выдачи неправильного лекарства, могут быть сведены к минимуму. Так, пациенты будут иметь узлы, которые определяют их аллергию и необходимые лекарства. Компьютеризированные системы, как описано выше, показали, что они могут помочь свести к минимуму побочные эффекты от ошибочной выдачи препаратов.

Применение в доме

Автоматизация дома: смарт-узлы могут быть интегрированы в бытовые приборы, например, в пылесосы, микроволновые печи, холодильники и видеомагнитофоны. Они могут взаимодействовать друг с другом и с внешней сетью через Интернет или спутник.

Это позволит конечным пользователям легко управлять устройствами дома как локально, так и удаленно. Умная окружающая среда: дизайн смарт-среды может иметь два различных подхода, т.е., ориентированного на человека или на технологии. В случае первого подхода, смарт-среда должна адаптироваться к потребностям конечных пользователей с точки зрения взаимодействия с ними. Для технологически-центрированных систем должны быть разработаны новые аппаратные технологий, сетевые решений, и промежуточные приложения. Примеры того, как узлы могут быть использованы для создания смарт-среды описана в . Узлы могут быть встроены в мебель и технику, они могут общаться друг с другом и сервером комнаты. Сервер комнаты может также общаться с другими серверами комнат, чтобы узнать о услугах, которые они могут предложить, например, печать, сканирование и работа с факсом. Эти сервера и сенсорные узлы могут быть интегрированы в существующие встраиваемые устройства и составлять самоорганизующиеся, саморегулируемые и адаптивные системы, основанные на модели теории управления, как описано в работе.

Факторы, влияющие на разработку моделей сенсорных сетей.

Разработка сенсорных сетей зависит от многих факторов, которые включают в себя отказоустойчивость, масштабируемость, издержек производства, вид операционной среды, топологию сенсорной сети, аппаратные ограничения, модель передачи информации и потребление энергии. Эти факторы рассматриваются многими исследователями. Однако ни в одном из этих исследований полностью не учтены все факторы, которые влияют на разработку сетей. Они важны, поскольку служат в качестве ориентира для разработки протокола или алгоритмов работы сенсорных сетей. Кроме того, эти факторы могут быть использованы для сравнения различных моделей.

II. Беспроводные самоорганизующиеся сети.

С точки зрения инженера-электронщика, датчик или сенсор – это устройство, которое используется для сбора информации о физическом процессе или физическом явлении и преобразования его в электрические сигналы, которые можно обрабатывать, измерять и анализировать. Термин «физический процесс», используемый в приведенном определении датчика, может быть любой реальной информацией, такой как температура, давление, свет, звук, движение, положение, поток, влажность, излучение и т. д. Что же из себя представляет беспроводная сенсорная сеть в реальности?

Как упоминалось ранее, в предыдущей статье, типичная сенсорная сеть состоит из датчиков, контроллера и системы связи. Если система связи в сенсорной сети реализована с использованием беспроводного протокола, то эти сети называются беспроводными сенсорными сетями или просто **WSN** (Wireless Sensor Networks).

Типичная беспроводная сенсорная сеть может быть разделена на два элемента: **сенсорный узел** (рис.18) и **сетевая архитектура** (рис.19).

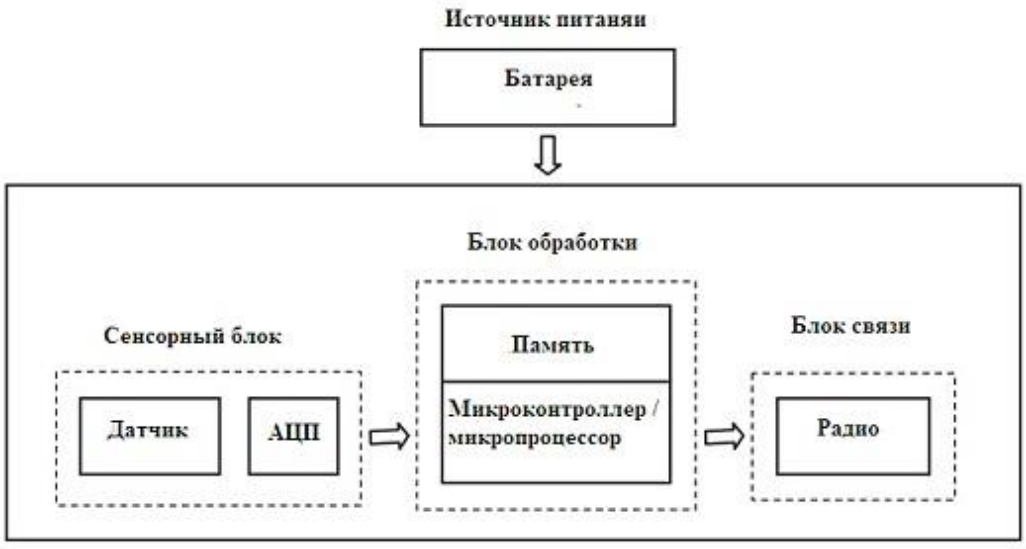


Рис. 18 – Сенсорный узел

Датчик собирает аналоговые данные из физического мира, и АЦП преобразует эти данные в цифровые данные. Основной процессор, который обычно является микропроцессором или микроконтроллером, выполняет интеллектуальную обработку данных и манипулирование ими.

Система связи состоит из системы радиосвязи, обычно радиостанции ближнего действия, для передачи и приема данных. Поскольку все компоненты являются устройствами с низким энергопотреблением, для питания всей системы используется небольшая батарея, такая как CR-2032 (такая используется в модуле часов реального времени (RTC) в вашем компьютере).

Несмотря на название, сенсорный узел состоит не только из сенсорного компонента, но и из других важных функций, таких как устройства обработки, связи и хранения. Благодаря всем этим функциям, компонентам и усовершенствованиям узел датчика отвечает за сбор данных физического мира, анализ сети, корреляцию данных и объединение данных другого датчика с собственными данными.

Архитектура беспроводной сенсорной сети

Когда большое количество сенсорных узлов развернуто в большой области для совместного мониторинга физической среды, объединение в сеть этих сенсорных узлов одинаково важно. **Сенсорный узел** в WSN не только связывается с другими сенсорными узлами, но также и с базовой станцией, используя беспроводную связь.

Рис.19 – Архитектура беспроводной сенсорной сети в общем виде

Базовая станция отправляет команды на сенсорные узлы, а сенсорные узлы выполняют задачу, взаимодействуя друг с другом. После сбора необходимых данных сенсорные узлы отправляют данные обратно на базовую станцию. Базовая станция также действует как шлюз для других сетей через Интернет. После приема данных от узлов датчиков базовая станция выполняет простую обработку данных и отправляет обновленную информацию пользователю через Интернет.

Если каждый узел датчика (сенсорного узла – см. рис 20) подключен к базовой станции, он известен как архитектура сети с одним переходом (или **<u>односкачковая архитектура – рис. 21</u>**). Хотя передача на большие расстояния возможна, потребление энергии для связи будет значительно выше, чем для сбора и вычисления данных.

Рис.20 – Практическая реализация сенсорного узла

Следовательно, **многоскачковая** сетевая архитектура обычно используется в серьезных приложениях. Вместо одной единственной линии связи между узлом датчика и базовой станцией данные передаются через один или несколько промежуточных узлов.

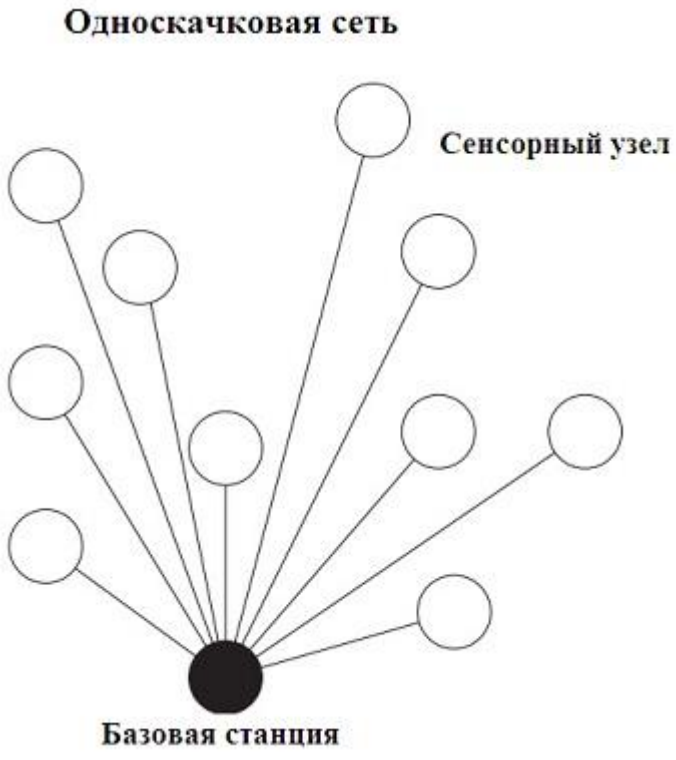


Рис.20 – Схема односкачковой сенсорной сети

Это может быть реализовано двумя способами: Архитектура плоской сети и архитектура иерархической сети.

В **плоской архитектур**е базовая станция отправляет команды всем сенсорным узлам, но сенсорный узел с совпадающим запросом ответит, используя свои равноправные узлы через **многоскачковый путь.** В иерархической архитектуре группа сенсорных узлов формируется в виде кластера, и сенсорные узлы передают данные в соответствующие головы кластера. Затем головы кластера могут передавать данные на базовую станцию (рис.21).

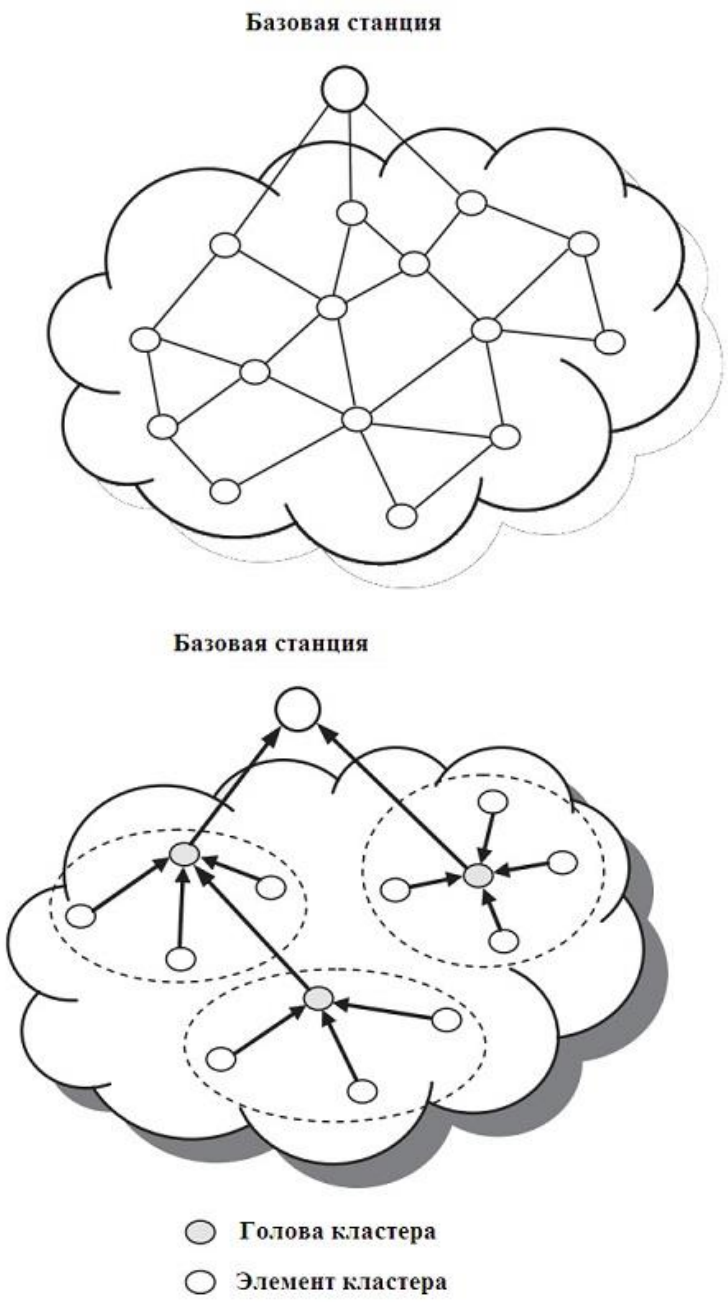


Рис. 21 – Процесс обмена

Топология сети – важная модель состояния сети, поскольку неявно она дает много информации об активных существующих узлах и связности сети. Так как **беспроводные сенсорные сети** обладают ограниченными энергетическими ресурсами, алгоритмы сбора информации о топологии должныпредполагать низкое энергопотребление.

В зависимости от требований предметной области формирование топологии сенсорной сети может происходить в двух режимах: топология типа "звезда" (иерархическая топология), либо топология типа "точка-точка" (однородная топология). В случае топологии типа "звезда" предполагается, что сеть состоит из объектов двух типов: полнофункциональные объекты и объекты с уменьшенной функциональностью (мы это рассматривали в главе **PAN**).

Объекты второго типа могут вести общение только с объектами первого типа. Ввиду повышенных нагрузок полнофункциональные устройства могут быть стационарными и иметь питание от внешних источников.

Такой способ организации сети может быть востребован для решения ограниченного спектра задач, например, в промышленности.

Второй вариант организации работы сети – "однородная" топология, когда не требуется разбиение сети сегменты (кластеры) и все объекты могут вести общение между собой в пределах области видимости, при этом вся сеть может разбиваться на сегменты, управляемые координаторами, а может и нет. Данный подход к формированию сети позволяет организовывать более сложные конфигурации сети, адаптировать такие сети к решению более сложных и нестандартных задач. Подобная гибкость достигается благодаря тому, что при таком подходе отдельные объекты могут самостоятельно организовываться сеть и адекватно реагировать на изменения в топологии сети со временем.

Кроме того, в рамках такой сети может быть реализована маршрутизация сообщений, когда объекты, не являющиеся непосредственными соседями, могут общаться между собой. Именно этот способ зачастую неявно подразумевается в большинстве печатных трудов, когда тематика работы непосредственно связана с понятием "сенсорная сеть".

Каждый координатор выбирает уникальный идентификатор подсети. Этот идентификатор обеспечивает связь между устройствами в сети с помощью коротких адресов и позволяет передавать данные между устройствами через независимые подсети. Все сети топологии «звезда» работают независимо от других сетей. После того, идентификатор выбран, координатор разрешает узлам подключаться к сети.

Количество статей, мнений, разработок, касающихся беспроводных сенсорных сетей уже достаточно велико, но всё это не дает представление о БСС как о единой структуре. Уже сегодня сенсорные сети нуждаются в первом шаге к направлению о представлении их как единой сложной системы.

Многообразие информации уже дает возможность собрать пазл эффективной, универсальной сенсорной сети, которая будет является отдельной, независимой технологией. Но простое (аддитивное) объединение отраслей (систем), которые содержит БСС не нерационально. С развитием вычислительной техники и средств связи наступила эра беспроводных сетей и распределенных вычислений. Пройдет еще несколько лет, и беспроводные технологии свяжут между собой огромные количество цифровых устройств, превратив Информационные Технологии во всепроницающую и вездесущую силу эпохи Информационного Общества. В свою очередь, беспроводные сенсорные сети, как элемент инфокоммуникационной структуры, позволяет расширить инфокоммуникационные возможности на периферию, давая возможность пользователю получить доступ к ранее недоступным услугам наблюдения состояния физических параметров контролируемого объекта или явления.

Изучение теоретических основ создания, развития, реализации БСС (беспроводных сенсорных сетей) на практике – залог карьерного успеха в будущем, когда все крупные компании, как полагают многие ученые, устремляться занять нишевые позиции на рынке информационных систем технологий. В следующей главе учебно-теоретического издания мы рассмотрим частный случай PAN/BAN сетей: Internet of things (IoT), содержащей в себе элементы концепции БСС (WSN) и, что немаловажно, уже активно внедряемый в нашу повседневную в жизнь уже сегодня.

I. Средства и технологии передачи данных: IEEE 802.15, Zigbee.

Семейство стандартов IEEE 802.15 образует беспроводную сеть WPAN (Wireless Personal Area Network) которая обеспечивает беспроводную связь между различного типа устройствами на небольших растояниях. Стандарты, которые входят с это семейство – это **Bluetooth** (IEEE 802.15.1), IEEE 802.15.3, **ZigBee** (IEEE 802.15.4) и UWB (Ultra Wideband) (IEEE 802.15.4a/b).

Беспроводная технология **Bluetooth**, основана на стандарте **IEEE 802.15.1**, является стандартом, определяющим функционирование компактных систем связи на небольших расстояниях между мобильными персональными компьютерами, мобильными телефонами и иными портативными устройствами. Bluetooth представляет собой недорогой радиоинтерфейс с низким энергопотреблением (мощность передатчика всего порядка 1 мВт) для организации персональных сетей, обеспечивающий передачу в режиме реального времени как цифровых данных, так и звуковых сигналов.

Изначально дальность действия радиоинтерфейса закладывалась равной 10 метрам, однако сейчас спецификациями Bluetooth уже определена и вторая зона около 100 м. Для работы радиоинтерфейса Bluetooth используется так называемый нижний (2,45 ГГц) диапазон ISM (industrial, scientific, medical), предназначенный для работы промышленных, научных и медицинских приборов. Радиоканал обладает полной пропускной способностью в 1 Мбит/с, что обеспечивает создание асимметричного канала передачи данных на скоростях 723,3/57,6 Кбит/с или полнодуплексного канала на скорости 433,9 Кбит/с. Если данные не передаются, то через Bluetooth-соединение можно передавать до 3-х дуплексных аудиоканалов по 64 Кбит/с в каждом направлении.

Возможна также и комбинированная передача данных и звука. В части организации обмена данными Bluetooth соответствует спецификации стандарта локальных сетей IEEE 802 и использует сигналы с расширением спектра путем скачкообразной перестройки частоты (FHSS) по псевдослучайному закону со скоростью 1600 переключений в секунду в полосе 2400-2483,5 МГц.

Bluetooth работает как многоточечный радиоканал, управляемый, аналогично сотовой связи GSM, многоуровневым протоколом с поддержкой обратной зависимостью. На данный момент актуальными версия этой технологии являются 4.0, 4.1, 4.2, 5.0 (рис.21).

	4.1	4.0	3.0	2.x	1.x
Базовая скорость	1 Мбит/с	1 Мбит/с	1 Мбит/с	1 Мбит/с	1 Мбит/с
Повышенная скорость передачи (EDR)	3 Мбит/с	3 Мбит/с	3 Мбит/с	3 Мбит/с	нет
High Speed	54 Мбит/с	54 Мбит/с	54 Мбит/с	нет	нет
Дальность (макс./мин. мощность)	100 м/ 10 м	100 м/ 10 м	100 м/ нет	100 м/ нет	100 м/ 10 нет
Режим низкого потребления	да	да	нет	нет	нет
Двойной профиль (одновременно Master и Slave)	да	нет	нет	нет	нет
Поддержка IPv6	готовится	нет	нет	нет	нет
Сопряжение NFC	да	да	да	да	нет
128-битное шифрование AES	да	да	нет	нет	нет

Рис. 21 - Сравнительная таблица версий (поколений) Bluetooth.

Первый чип с поддержкой Bluetooth 3.0 был выпущен компанией Sony в конце 2009 года. В настоящее время выпускается большое количество мобильных устройств с поддержкой этого стандарта.

Bluetooth 4.1

В конце 2013 года Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 4.1. Одно из улучшений, реализованных в спецификации Bluetooth 4.1, касается совместной работы Bluetooth и мобильной связи четвёртого поколения LTE Стандарт предусматривает защиту от взаимных помех путём автоматического координирования передачи пакетов данных.

Bluetooth 4.2

3 декабря 2014 Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 4.2. Основные улучшения — повышение конфиденциальности и увеличение скорости передачи данных.

Bluetooth 5.0

16-17 июня 2016 года Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 5.0. Изменения коснулись в основном режима с низким потреблением и высокоскоростного режима. Радиус действия увеличен в 4 раза, скорость увеличена в 2 раза.

Посредством Bluetooth можно объединить как два, так и сразу несколько устройств. В первом случае подключение осуществляется по схеме «точкаточка», во втором — по схеме «точка-многоточка». Независимо от применяемой схемы одно из устройств является ведущим (master), остальные — ведомыми (slave). Ведущее устройство задает шаблон, который будут использовать все ведомые устройства, а также синхронизирует их работу. Соединенные таким образом устройства образуют пикосеть (piconet). В рамках одной пикосети могут быть объединены одно ведущее и до семи ведомых устройств. Кроме того, допускается наличие в пикосети дополнительных ведомых устройств (сверх семи), которые имеют статус заблокированных (parked): они не участвуют в обмене данными, но при этом находятся в синхронизации с ведущим устройством.

Несколько пикосетей можно объединить в распределенную сеть (scatternet). Для этого устройство, работающее в качестве ведомого в одной пикосети, должно выполнять функции ведущего в другой (см. вторую схему). При этом пикосети, входящие в состав одной распределенной сети, не синхронизированы друг с другом и используют разные шаблоны.

Относительная универсальность является как преимуществом, так и недостатком Bluetooth. Во-первых, не все адаптеры поддерживают все профили (именно по этой причине универсальность Bluetooth является относительной). Во-вторых, в некоторых ситуациях эта самая универсальность может оказаться излишней (например, могут возникнуть трудности при нахождении устройства в сети с большим числом подключений).

Одним из главных недостатков сетей Bluetooth является обеспечиваемый уровень безопасности. Слабости защиты bluetooth, в частности, вызваны тем, что эта технология делает сильный упор на опознание устройств для безопасного обслуживания, а также на контроль, которым обладает пользователь над устройствами bluetooth и их конфигурацией. Современная bluetooth-технология не предлагает никакого средства опознания пользователя, что делает bluetooth-устройства особенно уязвимыми к так называемым spoofing-нападениям (радиодезинформации) и неправильному применению опознавательных устройств. Особенно слабым аспектом bluetooth является процесс «спаривания» (pairing) устройств, при котором происходит обмен ключами в незакодированных каналах. Если нападающий перехватит передачу процесса спаривания, то он сможет получить ключ инициализации путем калькуляции этих ключей для любого возможного варианта пароля и сравнения результатов с перехваченной передачей. Ключ инициализации используется для расчета ключа связи. Рассчитанный хакером ключ связи сравнивается с перехваченной передачей с целью узнать, верен он или нет.

Также причиной уязвимости является возможность использования коротких, а также заурядных/распространенных паролей (ситуация аналогична использованию простых паролей системными администраторами компьютерных сетей). Такие пароли значительно упрощают инициализацию. Именно это делает ключи связи очень простыми для извлечения из перехваченных передач.

Во многих приложениях требуются беспроводные сети связи (БСС), не обладающие высокой скоростью передачи, но **надежные**, живучие (способные к самовосстановлению), простые в развертывании и эксплуатации. Сети типа Bluetooth все же не, являются **надежными из-за вышеперечисленных причин**. Важно также, чтобы оборудование таких сетей допускало длительную работу от автономных источников питания, имело низкую стоимость, и было компактным. Пример такого приложения – «умный дом».

Такому сочетанию требований еще 10 лет назад не отвечал ни один из сетевых стандартов, что и привело к созданию стандартов IEEE 802.15.4 и **ZigBee**, описывающих устойчивые масштабируемые многошаговые беспроводные сети, простые в развертывании и поддерживающие самые разные приложения.

Стандарт IEEE 802.15.4 (ZigBee) ориентирован, главным образом, на использование в качестве средства связи между автономными приборами и оборудованием. В корпоративном секторе это могут быть, например, складские системы, системы автоматизации производства, различные датчики, сенсоры, сервоприводы, электронные метки, а в домашних условиях – персональные компьютеры, игровые приставки, системы безопасности, освещения, кондиционирования, радиофицированные игрушки.

Стандарт IEEE 802.15.4 определяет спецификации **физического слоя** (**PHY**) и **протокол управления доступом** (MAC), предлагая поддержку различных топологий сетей. Схемы сетевой маршрутизации призваны обеспечить сохранение энергии и кратчайшие задержки, укладывающееся в гарантированный временной интервал, а за счет наличия нескольких маршрутов к каждому узлу в сетях ZigBee предполагается предотвратить возможность "сбоя в одной точке".

Ключевые функции PHY включают в себя контроль за энергией и качеством звеньев, а также оценку каналов для более успешного сосуществования с сетями других беспроводных операторов.

MAC определяет автоматическое подтверждение получения пакетов, обеспечивает возможность передачи данных в определенные временные интервалы и поддерживает 128-битные функции-безопасности AES. Если в пределах досягаемости ZigBee-устройств окажется оборудование Wi-Fi или Bluetooth, их каналы могут быть использованы как туннель для трафика ZigBee.

Стандарт IEEE 802.15.4 предусматривает **радиус покрытия** от 10 до 75 м и пропускную способность канала - до 250 кбит/с. Передача на этой скорости ведется в диапазоне 2,4 ГГц. Небольшая мощность и скорость обусловлены малыми энергоресурсами связываемых устройств. Доступны также диапазоны 858 МГц (20 кбит/с) и 902-928 МГц(40 кбит/с). То есть 3 частотных диапазона.

Возможности: до 255 подчиненных устройств в сети и до 100 параллельно работающих сетей. Данный стандарт, активно продвигаемый организацией Альянсом ZigBee, заполнит вакуум в спектре беспроводных сетевых технологий, поскольку он предлагает разработчикам возможность создавать недорогие продукты с очень низким потреблением мощности и чрезвычайно гибкими функциями поддержки беспроводных сетей (рис.22).

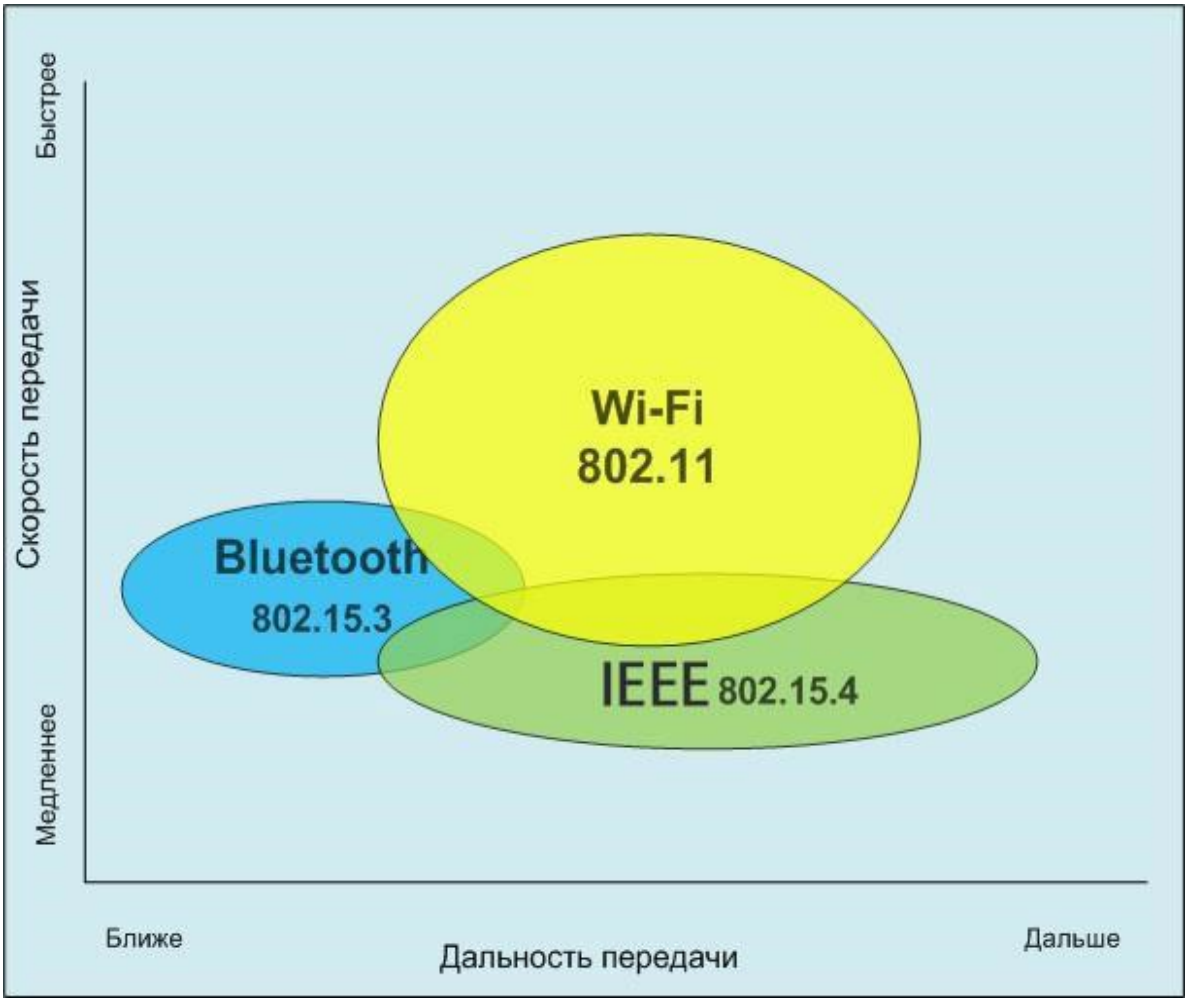


Рис. 22 - Сравнительная характеристика IEEE 802.15, 802.11, 802.15.3

Главные отличия и преимущества ZigBee от других беспроводных технологий: **ZigBee** работает по **ячеистому типу**, в то время как Wifi и Bluetooth присоединяются к центральному роутеру (топология «звезда») (рис.23). При отсутствии связи с роутером узел не может подключиться к другим членами сети. Например, если телевизор выходит в интернет через модем, то он не сможет воспроизвести фильм или получить его с планшета, если роутер не подключен к сети. В ячеистой структуре узлы связаны напрямую. Благодаря этому обрыв связи не является помехой для передачи данных. Этот механизм намного надежнее и применяется в самой сети интернет. Для реализации технологии «Умный дом» этот фактор очень важен. При аварии сигнал по Wifi может не передаться на большую дальность через бетонные стены.

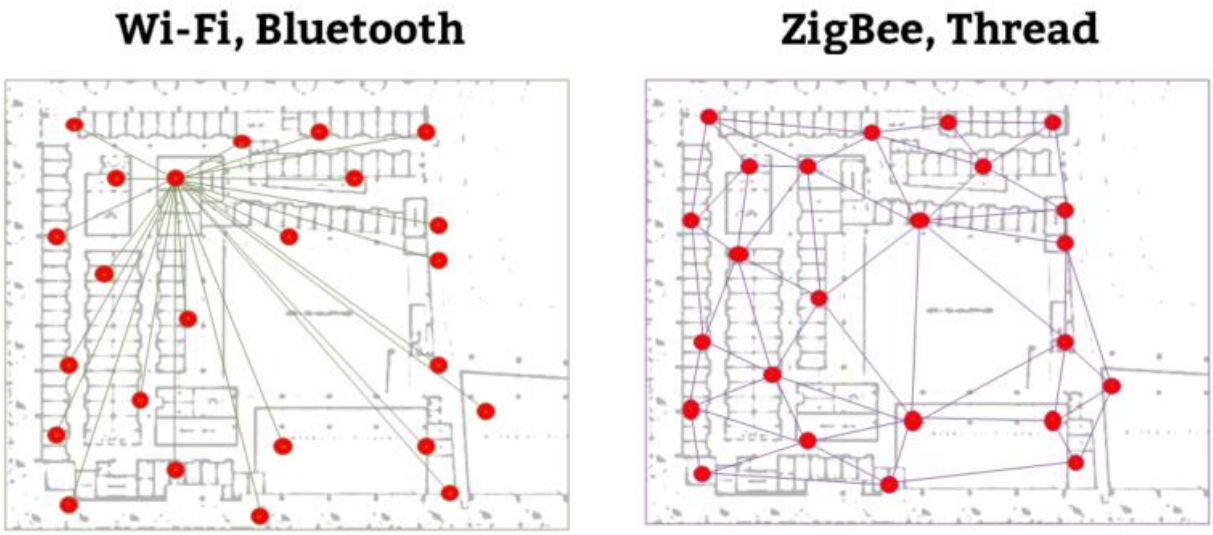


Рис. 23 – Ячеистый тип технологии передачи данных

ZigBee устройства потребляют минимальный объем энергии, поскольку функционируют в спящем режиме. Пропускная возможность Зигби (250 Кбит/секунду) намного ниже, чем у Wifi (300–1000 Мбит/секунду). Это связано с тем, что Зигби доставляет маленькие пакеты данных, а Wifi позволяет получить крупные файлы (видео и т.д.).

Скорость: Период задержки передачи сигнала Зигби **намного меньше**, чем у Bluetooth (несколько секунд) и составляет 30 миллисекунд. Показатель Зигби примерно равен времени от нажатия выключателя и возникновения света в люстре. В связи с этим, в последнее время Bluetooth стал меньше использоваться при установке системы «Умный дом».

Количество узлов: у Zigbee намного больше. Теоретически, к сети Wifi может присоединиться от 300 до 1000 участников. Но пользователи отмечают, что при работе уже с несколькими устройствами происходят задержки в работе и вряд-ли удастся проверить показатель. ZigBee система функционирует при любом количестве участников, что необходимо при установке «умного дома» на больших площадях.

Цена. Стоимость модуля ZigBee на порядок дешевле, чем цена Wifi модема.

На современный рынок, выпускается много ZigBee устройств: к ним относится розетка, диммер, лампочка, датчики движения, сенсоры контроля воды, температуры, и другие. На данный момент (январь 2020 года) по большому количеству причин среди производителей устройств на протоколе ZigBee лидирует китайская компания Xiaomi.

II. Средства идентификации, измерения, передачи данных LPWAN

Мы познакомились с стандартами беспроводной передачи информации, работающих на относительно небольших дистанциях (Bluetooth до 20 метров при условии прямой видимости, Wi-Fi и ZigBee на расстояния, не превышающие нескольких сотен метров при благоприятных условиях, с потерей эффективности передачи для конечных устройств). В завершении экскурса необходимо дать представление о новых беспроводных сетях – LPWAN.

LPWAN (англ. Low-power Wide-area Network — «энергоэффективная сеть дальнего радиуса действия») — беспроводная технология передачи небольших по объёму данных на дальние расстояния, разработанная для распределённых сетей телеметрии, межмашинного взаимодействия и интернета вещей. LPWAN является одной из беспроводных технологий, обеспечивающих среду сбора данных с различного оборудования: датчиков, счётчиков и сенсоров.

В основе принципа передачи данных по технологии LPWAN на физическом уровне PHY лежит свойство радиосистем — увеличение энергетики, а значит и дальности связи при уменьшении скорости передачи. Чем ниже битовая скорость передачи, тем больше энергии вкладывается в каждый бит и тем легче выделить его на фоне шумов в приёмной части системы. Таким образом, низкая скорость передачи данных позволяет добиться большей дальности их приёма.

Подход, используемый для построения LPWAN-сети, схож с принципом работы сетей мобильной связи. LPWAN-сеть использует топологию «звезда», где каждое устройство взаимодействует с базовой станцией напрямую. Сети городского или регионального масштаба строятся с использованием конфигурации «звезда из звезд».

Устройство или модем с LPWAN-модулем передает данные по радиоканалу на базовую станцию. Станция принимает сигналы от всех устройств в радиусе своего действия, оцифровывает и передаёт на удалённый сервер, используя доступный канал связи: Ethernet, сотовая связь, VSAT (спутниковая связь).

Полученные на сервере данные используются для отображения, анализа, построения отчетов и принятия решений.

Управление устройствами, обновление программного обеспечения происходит с использованием обратного канала связи.

Для передачи данных по радиоканалу, как правило, применяется нелицензируемый спектр частот, разрешенных к свободному использованию в регионе построения сети: 5,0 ГГц, 2,4 ГГц, 868/915 МГц, 433 МГц, 169 МГц.

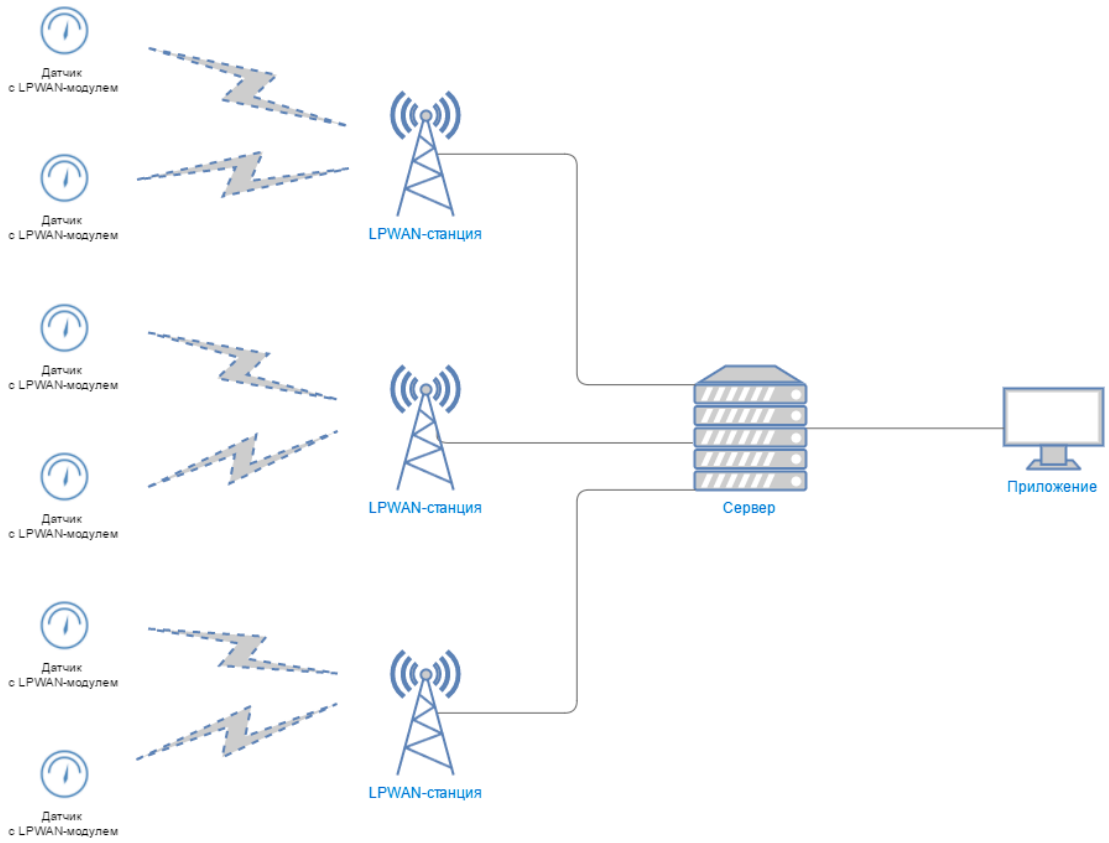


Рис. 24 – Топология LPWAN-сетей

Преимущества LPWAN

Большая дальность передачи радиосигнала по сравнению с другими беспроводными технологиями используемыми для телеметрии GPRS или ZigBee, достигает 10—15 км.

Низкое энергопотребление у конечных устройств, благодаря минимальным затратам энергии на передачу небольшого пакета данных.

Высокая проникающая способность радиосигнала в городской застройке при использовании частот суб-гигагерцового диапазона. Высокая масштабируемость сети на больших территориях. Отсутствие необходимости получения частотного разрешения и платы за радиочастотный спектр, вследствие использования нелицензируемых частот (ISM band).

Недостатки LPWAN

Относительно низкая пропускная способность, вследствие использования низкой частоты радио канала. Варьируется в зависимости от используемой технологии передачи данных на физическом уровне, составляет от нескольких сотен бит/с до нескольких десятков кбит/с.

Задержка передачи данных от датчика до конечного приложения, связанная с временем передачи радиосигнала, может достигать от нескольких секунд до нескольких десятков секунд.

Отсутствие единого стандарта, который определяет физический слой и управление доступом к среде для беспроводных LPWAN-сетей.

Технология LPWAN также, как и описанные ранее технологии, ориентирована на приложения, требующие гарантированной передачи небольшого объёма данных, возможности длительной работы сетевых устройств от автономных источников питания, большого территориального охвата беспроводной сетью. Основными областями применения технологии LPWAN являются беспроводные сенсорные сети, автоматизация сбора показаний приборов учета, системы промышленного мониторинга и управления.

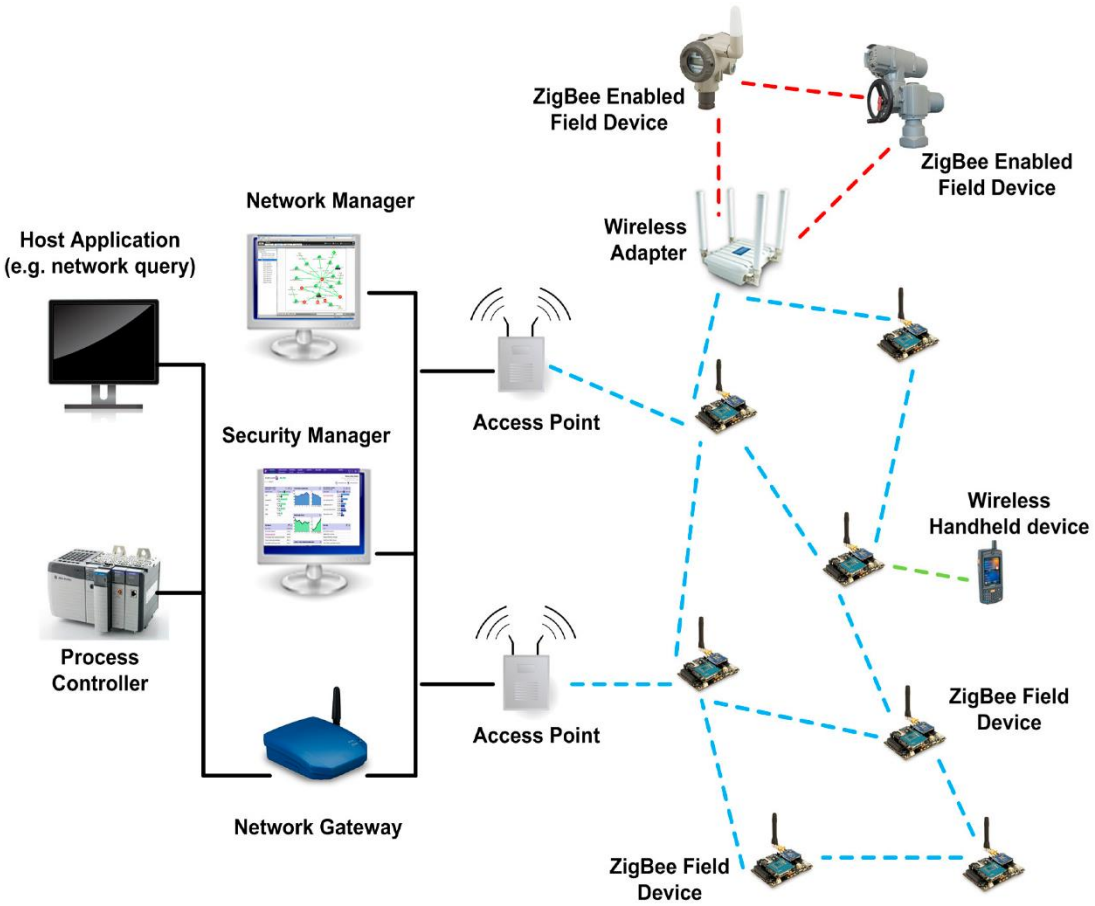


Рис. 24 – LPWAN на базе инкапсуляции в себе сети ZigBee

Типовые средства идентификации, передачи (равно как измерения данных) в LPWAN лучше всего описывают LoRa-модули (Родственная к LPWAN технология)

Каждый комплект может служить строительным блоком для развития сети, где разработчики могут рассчитывать на расстояние до 10 км и 10 лет автономной работы от двух батарей AAA. Технология LoRa использует модуль с расширенным спектром, который обеспечивает отличную устойчивость данных в шумовой среде и работает через физические препятствия.

Рис. 25 – Базовый комплект LoRa модулей

На рисунке 25 представлена базовая конфигурация для LPWAN сети: не хватает лишь оконечных устройств (сенсоров, датчиков и т.д). Здесь два модема (полнофункциональные устройства) и одно управляющее устройство, выполняющее роль координатора, в некоторой степени и программатора и средства ввода/вывода информации (ЖК-модуль, тактовые кнопки, интерфейсы). Вот так выглядят сети будущего. Никакой избыточности. Аскетизм предельной степени.

III. Окружающий интеллект: платформа, технология, применение.

Окружающий интеллект (англ. Ambient intelligence, AmI) — термин для обозначения окружающей среды, насыщенной электронными устройствами, которые реагируют на присутствие людей. В русскоязычных источниках термин «окружающий интеллект» упоминается, но пока не является устоявшимся эквивалентом английского Ambient intelligence. В англоязычных источниках парадигма окружающего интеллекта основывается на технологиях распределённых вычислений, построении персональных профилей контекстной ориентированности, клиенто-ориентированного дизайна человеко-компьютерного взаимодействия и характеризуется наличием следующих особенностей:

встроенность: многие сетевые устройства интегрированы в окружающую среду;

контекстная ориентированность: эти устройства могут распознавать пользователя и связанный с ним ситуационный контекст;

кастомизация: они могут быть приспособлены к потребностям конкретного пользователя;

адаптивность: они могут изменяться в ответ на реакцию пользователя;

упреждение: они могут предвидеть желания пользователя без каких-либо особых действий со стороны последнего.

По оценкам Консультативной группы Еврокомиссии по вопросам информационного общества и технологий (ISTAG), окружающий интеллект получает общественное признание благодаря созданию им следующих возможностей:

- облегчение контактов между людьми;
- ориентация на сотрудничество и культурное развитие;
- распространение знаний и навыков, повышение качества работы и выбора потребителей;
- формирование доверия и уверенности в себе;
- содействие устойчивому развитию личности, общества и окружающей среды в долгосрочной перспективе;
- простота и лёгкость контроля со стороны рядовых пользователей.

Ярким представлением окружающего интеллекта является платформа **Интернета вещей.**

Internet of Things, IoT — концепция вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключающее из части действий и операций необходимость участия человека.

Концепция сформулирована в 1999 году как осмысление перспектив широкого применения средств радиочастотной идентификации для взаимодействия физических предметов между собой и с внешним окружением. Наполнение концепции многообразным технологическим содержанием и внедрение практических решений для её реализации начиная с 2010-х годов считается устойчивой тенденцией в информационных технологиях, прежде всего, благодаря повсеместному распространению беспроводных сетей, появлению облачных вычислений, развитию технологий межмашинного взаимодействия, началу активного перехода на IPv6 и освоению программно-определяемых сетей. Современное решение из класса "Интернета вещей" содержит, в том или ином виде, порознь или комбинированно, следующие компоненты:

Исполнительные элементы (иногда называемые "актуаторами"). С вышестоящими контроллерами они связываются, как правило, по некоторому узкоспециализированному протоколу. Это могут быть унаследованные проводные протоколы (RS–232/485, 1–Wire, USB, CAN), беспроводные протоколы малой дальности (Bluetooth, ZigBee и т.п.) или современные протоколы сетей LP-WAN (Low-power Wide-area Network) с низким энергопотреблением и большой дальностью. Именно технологии LPWAN стали одной из важнейших компонент, определяющих облик современного IoT.

Использование IP на этом уровне также не исключается (поверх Ethernet, сотовых сетей или Wi-Fi), но датчики на базе специализированных локальных протоколов, как правило, получаются проще, функциональнее и дешевле, чем полноценные IP-хосты — а цена имеет в данном случае первостепенное значение. С другой стороны, по мере расширения выпуска готовых встраиваемых компьютеров в формате System-on-Chip (SoC) и System-on-Module (SoM) и снижения их стоимости доля "чистых" IP-решений может увеличиться. (В частности, NSG предлагает в данном классе вычислительное ядро NSG UltraLite для построения разнообразных систем автоматизации, в т.ч. IoT.)

Контроллеры исполнительных механизмов. В общем случае, они терминируют соединения с датчиками на физическом уровне, а протокол канального уровня либо также терминируют и преобразовывают данные в какой-либо из стандартных протоколов IP-стека (как правило, UDP, но теоретически не исключён и, например, TCP).

Сервер IoT непосредственно работает с датчиками: регистрирует их в системе, аутентифицирует (при необходимости), опрашивает, принимает показания, отсылает команды исполнительным элементам. Дальнейший обмен с прикладным сервером также идёт по сети IP. При этом могут использоваться разнообразные прикладные протоколы поверх TCP или UDP — например, SNMP или Zabbix. Современная тенденция состоит в использовании для этой цели механизма **MQTT (Message Queue Telemetry Transport**), как наиболее подходящего для поставленной задачи. Именно он строит общую крышу, под которую ныне становится возможным подвести самые разнородные решения и их компоненты. Это вторая ключевая компонента, отличающая IoT от разрозненных систем предыдущих поколений.

Прикладной сервер работает уже не с датчиками, а исключительно с данными, полученными от них: накапливает, хранит, обсчитывает какую-то статистику и аналитику, генерирует отчёты в разных формах… Наконец, сервер замыкает контур управления между датчиками и исполнительными элементами, если на нём задан какой-либо детерминированный алгоритм, по принципу "щёлкни кобылу по носу — она махнёт хвостом". Если же такой алгоритм не задан, то контур управления остаётся открытым и замыкается уже на пользовательском устройстве или вручную самим пользователем: получил данные — подумал головой — отправил команду.

Клиентские устройства и приложения позволяют пользователю видеть информацию от сервера и отдавать команды серверу (а через него — исполнительным элементам). Как частный случай, алгоритм для автоматического управления может быть задан на клиентах, а не на сервере.
Клиенты IoT могут быть наиболее разнообразными: стационарные компьютеры, мобильные устройства, специализированные пульты, со стандартными или специализированными приложениями. С прикладным сервером они могут взаимодействовать посредством HTTP, MQTT, электронной почты, сервисов мгновенного обмена сообщениями, консольных команд и многого другого, что можно придумать для этой цели сейчас или в будущем. Плюс средства, выходящие за рамки стека IP: SMS, USSD, голосовой телефонный интерфейс.

Ещё раз подчеркнём, что описанная выше архитектура IoT — пока ещё очень предварительная и не устоявшаяся. И вероятней всего она будет гибкой, атипичной к уже классическим системам. В частности, любые два или несколько смежных элементов могут быть объединены в одном устройстве. Или же, наоборот, они могут быть рассредоточены в территориально-распределённую инфраструктуру с несколькими серверами IoT, многими контроллерами IoT при каждом сервере, и множеством датчиков на каждом контроллере.

Например, как уже сказано выше, датчик IoT может объединяться с контроллером в "интеллектуальный датчик" с встроенной поддержкой IP-стека; контроллер в этом случае если и сохраняется, то вырождается в обычный коммутатор Ethernet или маршрутизатор IP, безо всякой специфики IoT. Контроллер может объёдиняться с сервером IoT, а сервер — с прикладным сервером. С другой стороны, функции прикладного сервера могут быть частично или полностью переданы клиентам. Серверы MQTT могут располагаться и на сервере IoT, и на прикладном сервере, и на отдельном хосте. И так далее. Именно такая гибкость позволяет, с одной стороны, модифицировать архитектуру в соответствии с практическими требованиями, а с другой стороны — рассматривать её как универсальный шаблон, пригодный для самых разных задач.

Как крайний случай, интеллектуальный датчик может содержать в себе все вышестоящие звенья, вплоть до прикладного сервера (**рис.26**); вне его остаётся только клиентское устройство. Противоположный вариант — все промежуточные сервисы располагаются где-то в облаке поставщика услуг, а у клиента остаются только датчики (например, с протоколом NB-IoT) на площадке и мобильное устройство в руках.

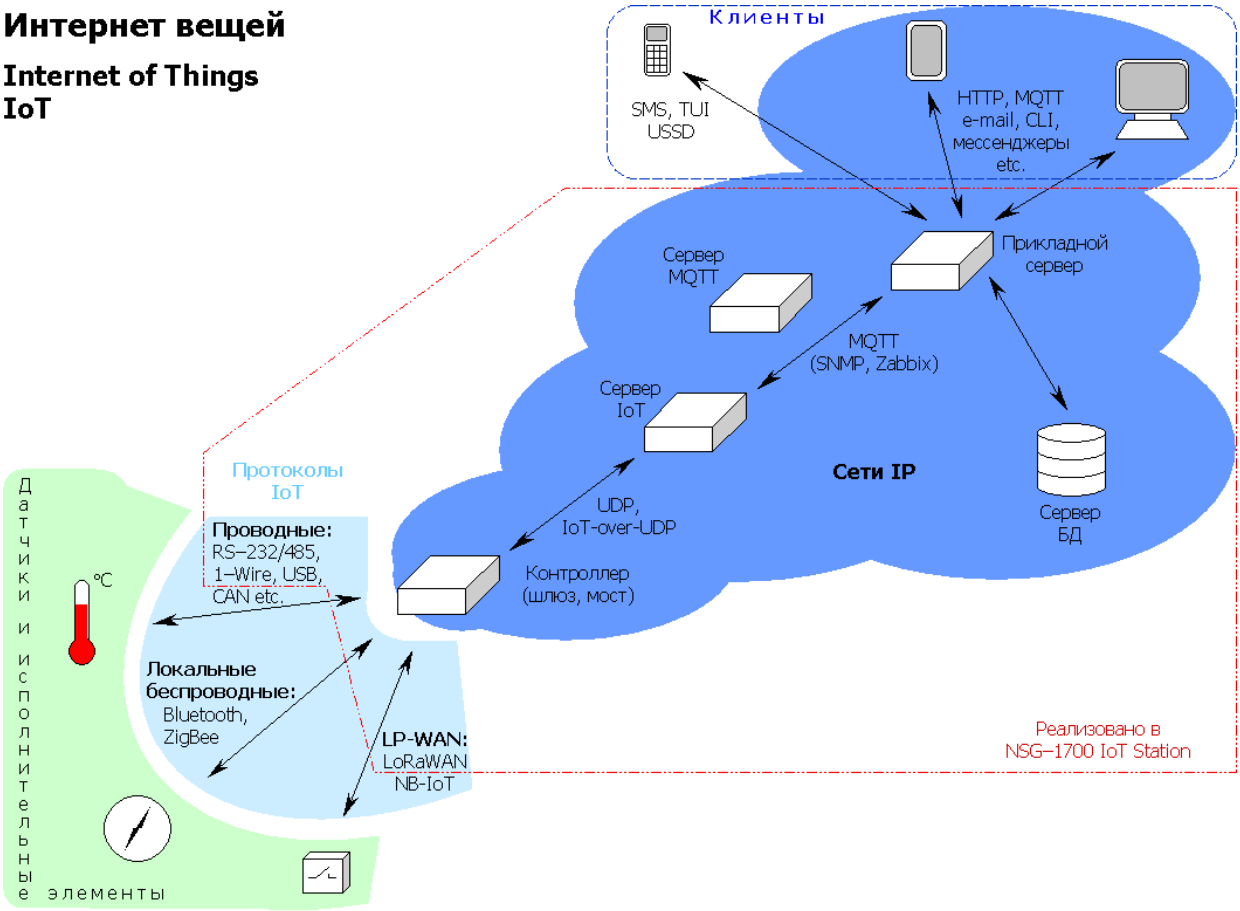


Рисунок 26 – IoT-платформа

Таким образом, результируя, концепция IoT позволяет выделить три **набора технологий**, описывающих средства измерения, передачи данных и идентификации – три столба данной системы. Опишем их:

Средства идентификации

Задействование в «интернете вещей» предметов физического мира, не обязательно оснащённых средствами подключения к сетям передачи данных, требует применения технологий идентификации этих предметов («вещей»). Хотя толчком для появления концепции стала технология RFID, но в качестве таких технологий могут использоваться все средства, применяемые для автоматической идентификации: оптически распознаваемые идентификаторы (штрихкоды, Data Matrix, QR-коды), средства определения местонахождения в режиме реального времени. При всеобъемлющем распространении «интернета вещей» принципиально обеспечить уникальность идентификаторов объектов, что, в свою очередь, требует стандартизации.

Для объектов, непосредственно подключённых к интернет-сетям, традиционный идентификатор — MAC-адрес сетевого адаптера, позволяющий идентифицировать устройство на канальном уровне, при этом диапазон доступных адресов практически неисчерпаем (248 адресов в пространстве MAC-48), а использование идентификатора канального уровня не слишком удобно для приложений. Более широкие возможности по идентификации для таких устройств даёт протокол IPv6, обеспечивающий уникальными адресами сетевого уровня не менее 300 млн устройств на одного жителя Земли.

Рис. 27 – Принципы обеспечения безопасности в IoT

Средства измерения

Особую роль в интернете вещей играют средства измерения, обеспечивающие преобразование сведений о внешней среде в машиночитаемые данные, и тем самым наполняющие вычислительную среду значимой информацией. Используется широкий класс средств измерения, от элементарных датчиков (например, температуры, давления, освещённости), приборов учёта потребления (таких, как интеллектуальные счётчики) до сложных интегрированных измерительных систем. В рамках концепции «интернета вещей» принципиально объединение средств измерения в сети (такие, как беспроводные датчиковые сети, измерительные комплексы), за счёт чего возможно построение систем межмашинного взаимодействия.

Как особая практическая проблема внедрения «интернета вещей» отмечается необходимость обеспечения максимальной автономности средств измерения, прежде всего, проблема энергоснабжения датчиков. Нахождение эффективных решений, обеспечивающих автономное питание сенсоров (использование фотоэлементов, преобразование энергии вибрации, воздушных потоков, использование беспроводной передачи электричества), позволяет масштабировать сенсорные сети без повышения затрат на обслуживание (в виде смены батареек или подзарядки аккумуляторов датчиков).

Средства передачи данных

Спектр возможных технологий передачи данных охватывает все возможные средства беспроводных и проводных сетей.

Для беспроводной передачи данных особо важную роль в построении «интернета вещей» играют такие качества, как эффективность в условиях низких скоростей, отказоустойчивость, адаптивность, возможность самоорганизации. Основной интерес в этом качестве представляет стандарт IEEE 802.15.4, определяющий физический слой и управление доступом для организации энергоэффективных персональных сетей, и являющийся основой для таких протоколов, как ZigBee, WirelessHart, MiWi, 6LoWPAN, LPWAN.

Среди проводных технологий важную роль в проникновении «интернета вещей» играют решения **PLC** — технологии построения сетей передачи данных по линиям электропередачи, так как во многих приложениях присутствует доступ к электросетям (например, торговые автоматы, банкоматы, интеллектуальные счётчики, контроллеры освещения изначально подключены к сети электроснабжения). 6LoWPAN, реализующий слой IPv6 как над IEEE 802.15.4, так и над PLC, будучи открытым протоколом, стандартизуемым IETF, отмечается как особо важный для развития «интернета вещей

IV. Актуаторы, айтрекеры – элементы сетей завтрашнего дня.

Данная глава предназначена для приобретения на вооружение двух лексических единиц, активно встречающихся в литературе, на практике, на производстве. Эти термины в русскоязычном сегменте в самое ближайшее время станут такими же привычными, как и понятия коммутации, маршрутизации. Более того, вы скорей всего знаете эти термины. **Актуаторы** уже упоминались чуть ранее в этом пособии, а айтрекеры уже давно на слуху в интернет-издания. Безусловно, я бы мог поместить «расшифровку» и этимологию этих понятий в самый конец книги, но… Это будет не совсем правильно. А вот на вопрос: почему так? Вы сможете получить ответ сейчас.

Перейдем к развернутому определению термина «актуатор» и «айтрекер», используя и смежные понятия:

Актуатор (исполнительный элемент, **актуатор**) — функциональный элемент системы автоматического управления, который воздействует на объект, изменяя поток энергии или материалов, которые поступают на объект. Большинство исполнительных устройств имеет механический или электрический выход.

Состоит из двух функциональных блоков: исполнительного устройства (если исполнительное устройство механическое, то его часто называют исполнительный механизм) и регулирующего органа, например, регулирующего клапана, и может оснащаться дополнительными блоками.

В теории автоматического управления под исполнительным устройством понимают устройство, передающее воздействие с управляющего устройства на объект управления. Иногда рассматривается как составная часть объекта управления. Управляющим устройством может быть любая динамическая система. Входные и выходные сигналы исполнительных устройств, а также их методы воздействия на объект управления могут иметь различную физическую природу.

Виртуальные приборы (англ. Virtual Instrumentation) — концепция, в соответствии с которой организуются программно-управляемые системы сбора данных и управления техническими объектами и технологическими процессами, при которой система организуется в виде программной модели некоторого реально существующего или гипотетического прибора.

Причём программно реализуются не только средства управления (рукоятки, кнопки, лампочки и т. п.), но и логика работы прибора.

Связь программы с техническими объектами осуществляется через интерфейсные узлы, представляющие собой драйверы внешних устройств — АЦП, ЦАП, контроллеров промышленных интерфейсов и т. п. Предшественницей концепции виртуальных приборов служила концепция слепых приборов, предусматривающая организацию системы в виде физического устройства («ящика», реализующего логику работы прибора, но не имеющего пользовательского интерфейса), и программно-реализуемых средств управления (представляющих собой HMI в чистом виде). Виртуальные приборы (англ. Virtual Instrumentation) — концепция, в соответствии с которой организуются программно-управляемые системы сбора данных и управления техническими объектами и технологическими процессами, при которой система организуется в виде программной модели некоторого реально существующего или гипотетического прибора, причём программно реализуются не только средства управления (рукоятки, кнопки, лампочки и т. п.), но и логика работы прибора. Связь программы с техническими объектами осуществляется через интерфейсные узлы, представляющие собой драйверы внешних устройств — АЦП, ЦАП, контроллеров промышленных интерфейсов и т. п.

Предшественницей концепции виртуальных приборов служила концепция слепых приборов, предусматривающая организацию системы в виде физического устройства («ящика», реализующего логику работы прибора, но не имеющего пользовательского интерфейса), и программно-реализуемых средств управления (представляющих собой HMI в чистом виде).

Концепция виртуальных приборов применяется в качестве базовой в таких продуктах, как:

1. LabVIEW фирмы National Instruments (США) (http://www.natinst.com),
2. реализуется на программной архитектуре VISA;
3. DASYLab фирмы DATALOG GmbH (Германия) (http://www.dasylab.com);
4. DIAdem фирмы GfS mbH (Германия);
5. ZETLab фирмы "ЭТМС" (Россия);

В вычислительной технике актуаторы представляют собой преобразователи, превращающие входной сигнал (электрический, оптический, механический, пневматический и др.) в выходной сигнал (обычно в движение, но не всегда), воздействующий на объект управления.

Устройства такого типа включают: электрические двигатели, электрические, пневматические или гидравлические приводы, релейные устройства, электростатические двигатели (англ. Comb drive), DMD-зеркала и электроактивные полимеры, хватающие механизмы роботов, приводы их движущихся частей, включая соленоидные приводы и приводы типа «звуковая катушка» (англ. Voice coil), а также многие другие.

Виртуальные (программные) приборы используют исполнительные устройства и датчики для взаимодействия с объектами реального мира. С помощью датчиков сигнал передаётся в виртуальный прибор, обрабатывается и выдаётся в реальный мир с помощью различного вида исполнительных устройств (рис.28).

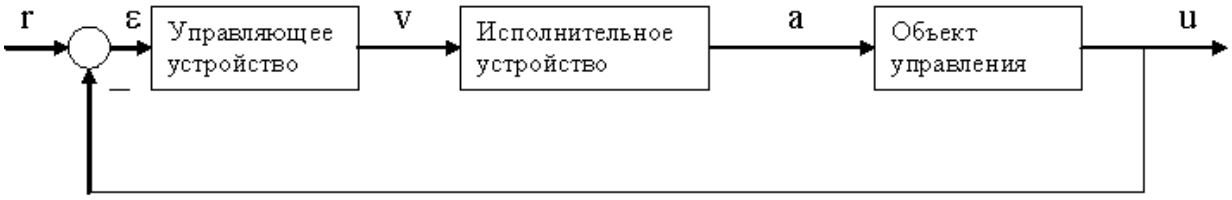


Рис. 28 – Компоненты простого актуаторы

Окулография (отслеживание глаз, трекинг глаз; айтрекинг) — определение координат взора («точки пересечения оптической оси глазного яблока и плоскости наблюдаемого объекта или экрана, на котором предъявляется некоторый зрительный раздражитель»).

Отслеживатель глаз (**айтрекер**) — устройство (рис.29), используемое для определения ориентации оптической оси глазного яблока в пространстве (то есть для отслеживания глаз).

Отслеживатели глаз используются в исследованиях зрительной системы, психологии, когнитивной лингвистике, промышленном управлении. Для отслеживания глаз используется несколько методов. Самый популярный — покадровый анализ видеосъёмки глаза, также используются контактные методы, такие как электроокулография.

Рис. 29 – Айтрекер

Применения систем айтрекинга включают в себя веб-юзабилити (удобство чтения веб-страниц конечным потребителем), рекламу, оптимизацию внешнего дизайна продукции и автоматизацию разработки. В общем коммерческое использование отслеживания глаз в большинстве сводится к тому, что группе потребителей предъявляется один и тот же визуальный стимул, в то время как отслеживаются движения глаз. Примерами конечных стимулов могут быть веб-сайты, телевизионные программы, трансляции спортивных состязаний, фильмы, рекламные ролики, страницы журналов, страницы газет, упаковки некоторых продуктов и прилавки магазинов, также банкоматы и пользовательские интерфейсы программного обеспечения.

Результирующие данные могут быть статистически анализированы и графически отражены для того, чтобы показать справедливость сделанных выводов. Путём исследования фиксаций, изменения размера зрачка, морганий и ряда других параметров исследователи в значительной степени могут определить эффективность созданного информационного ресурса или продукта. Пока некоторые компании пытаются решить подобные задачи внутренними ресурсами, другие привлекают фирмы, предлагающие услуги отслеживания глаз. Наиболее многообещающее поле использования коммерческого отслеживания глаз это веб-юзабилити. Несмотря на то, что традиционные техники юзабилити дают достаточно адекватные данные путём анализа кликов мышкой и прокручивания, айтрекинг даёт возможность анализировать связь между поведением пользователя и кликами мышкой.

Это даёт значительное улучшение оценки того, какие фрагменты веб-сайта являются наиболее привлекательными для пользователя, какие фрагменты веб-сайта вызывают трудности у конечного пользователя и какие пользователем не замечаются. Айтрекинг также может быть использован для оценки эффективности поиска, правильности концепции бренда, онлайн-исследования, юзабилити перехода между страницами, эффективности общего дизайна и многих других аспектов веб-дизайна. В процессе исследования может быть проведено сравнение двух сайтов-конкурентов.

Отслеживание глаз традиционно используется для оценки эффективности рекламы на различных медиаресурсах. Телевизионные видеоролики, рекламные буклеты, реклама на интернет-сайтах, показ эмблемы спонсоров в телепрограммах, все это открывает обширное поле деятельности для коммерческого отслеживания глаз. Анализируются заметность упаковки с продуктом или некоторого логотипа на витрине магазина, газеты, веб-сайта и телепрограммы. Это позволяет исследователям с высокой детализацией оценивать то, как потребители замечают или не замечают логотип конечного продукта, упаковку, POS. Таким образом, специалист по рекламе может оценить эффективность рекламной компании благодаря реальному визуальному восприятию.

Отслеживание глаз позволяет разработчикам упаковки продукта оценить её эффективность. Таким образом могут быть оценены заметность, привлекательность и соответствие современным трендам исследуемой упаковки с целью оптимального выбора. Отслеживание глаз часто используется, пока коммерческий продукт ещё находится на стадии прототипа. Прототипы часто тестируются парами для выявления наиболее эффективного своего дизайна, а также сравнение с решениями конкурентов.

Одно из наиболее многообещающих применений отслеживания глаз это оптимизация дизайна уличных терминалов. В настоящее время исследователи дошли до того, что предлагают интегрировать айтрекеры в серийно производимые уличные терминалы/банкоматы. Основной задачей этого является уменьшения времени взаимодействия между человеком и устройством.

Отслеживатели глаз могут также использоваться для оптимизации системы автофокуса цифровой фотокамеры (резкость наводится туда, куда смотрит пользователь).

The National Highway Traffic Safety Administration (NHTSA) утверждает, что интеграция отслеживателей глаз в автомобиле может сократить количество ДТП на 100 тысяч в год. В соответствие с их исследованиями до 80 % ДТП происходят в результате неправильных действий водителя в течение 3-х секунд перед аварией. Экипировка автомобилей айтрекерами позволит значительно увеличить класс безопасности этих автомобилей. «Лексус» обещает оснастить модель LS460 встроенным отслеживателем глаз, подающим предупреждающий сигнал в случае, если водитель отвлекается от дороги.

С 2006 года система айтрекинга используется в коммуникационном оборудовании для полностью парализованных людей. Они позволяют набирать текстовые сообщения, отправлять электронную почту, работать в интернете, используя исключительно их глаза. Отслеживание глаз позволяет достичь положительных результатов даже в случае церебрального паралича, при котором пациент совершает непроизвольные движения.

Понятно, что современный айтрекинг **не может обойтись** без интеграции **в мир сетевого окружения**. И работы в этом направлении уже ведутся. Рекомендую ознакомится с статьей на портале Хабрахабр «Айтрекинг в UX-исследованиях (QR-код с ссылкой указан на текущей страницей). Рекомендовал бы также найти небольшую статью «Управление компьютером при помощи глаз — практическая реализация» на том же портале.

Айтрекинг в UX-исследованиях

Конечным этапом или сущностной ценностью данного издания является закрепление понимания функционирования сетей в концепции Internet of Things (IoT), построенных в рамках LPWAN/BAN/PAN. Для это необходимо воспользоваться инструментом моделирования сети, содержащей в себе актуаторы (датчики), средства контроля (управляющие компоненты, полнофункциональные устройства).

Таким образом, необходимо было решить три **задачи** практического характера:

1. Исследование реально существующей интеллектуальной домашней сети.
2. Добавление проводных устройств ввода-вывода в интеллектуальную домашнюю сеть
3. Добавление беспроводных устройств ввода-вывода в интеллектуальную домашнюю сеть.

Для этих целей была разработана комплексная лабораторная работа. Для выполнения данной работы необходимо **следующее оборудование и программное обеспечение**: Персональный компьютер с 64х битной операционной системой Windows 7, 8, 8.1, 10 (рекомендуемо) или Ubuntu Linux 16.04 LTS (и выше) с установленным программным обеспечением: симулятором сети передачи данных Cisco Packet Tracer версии не ниже 7.1 (рекомендуется 7.2 и новее).

Напомним, что умный дом — единая система управления в доме, офисе, квартире или здании, включающая в себя датчики, управляющие элементы и исполнительные устройства.

Исследуем конечные устройства IoT в программе Cisco Packet Tracer 7.2:

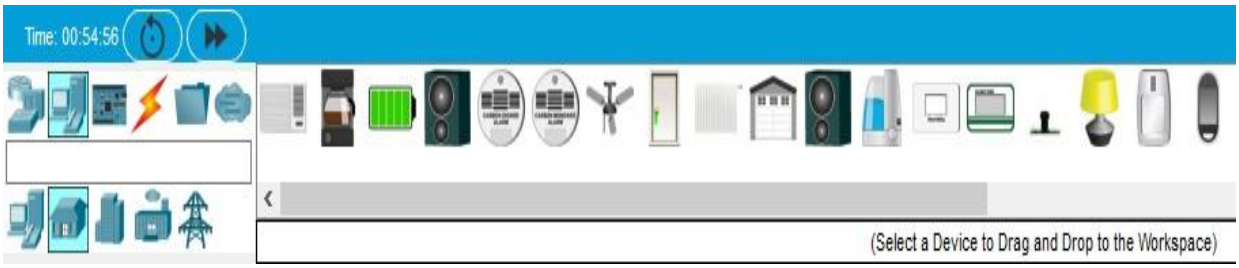


Рис.30 - Конечные устройства IoT в программе Cisco Packet Tracer.

В нижнем меню программы Cisco Packet Tracer во вкладке End Devices (рис.30) –> Home находятся различные элементы Smart Home IoT для умного дома.

Наведя мышкой на устройство, откроется окно основного перечня свойств данного элемента:

Рис.31 - Информационное окно с основной сетевой информацией об устройстве.

С интерфейсом на первоначальном этапе мы разобрались. Теперь мы должны открыть (рис.32) готовый проект сети (файл: **Packet Tracer - Adding IoT Devices to a Smart Home.pkt)**

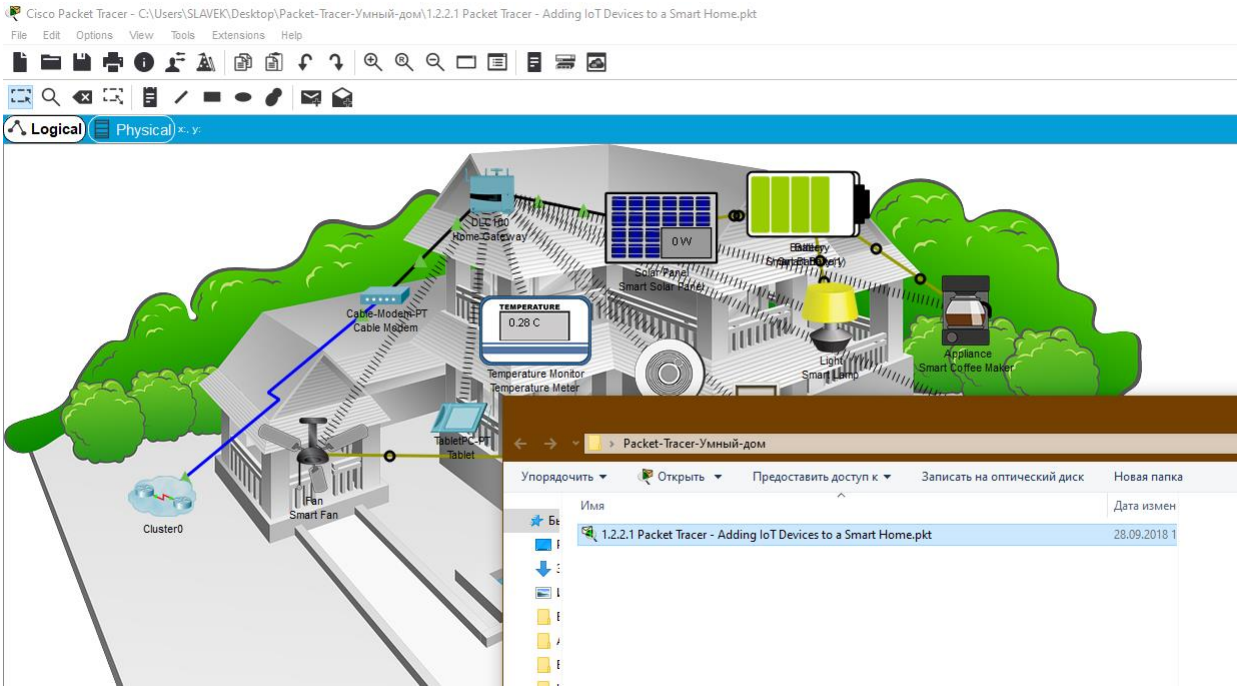


Рис. 32 – Проект сети

Давайте рассмотрим ее более детально. Интеллектуальная домашняя сеть, представленная на рисунке 32, состоит из инфраструктурных устройств, таких как «домашний» шлюз – управляющее устройство (устройство координатор по классификации PAN/LPWAN сетей).

Щелкнем значок Home Gateway, чтобы открыть окно устройства Home Gateway.

Перейдем на вкладку Config для просмотра настроек (рис.32):

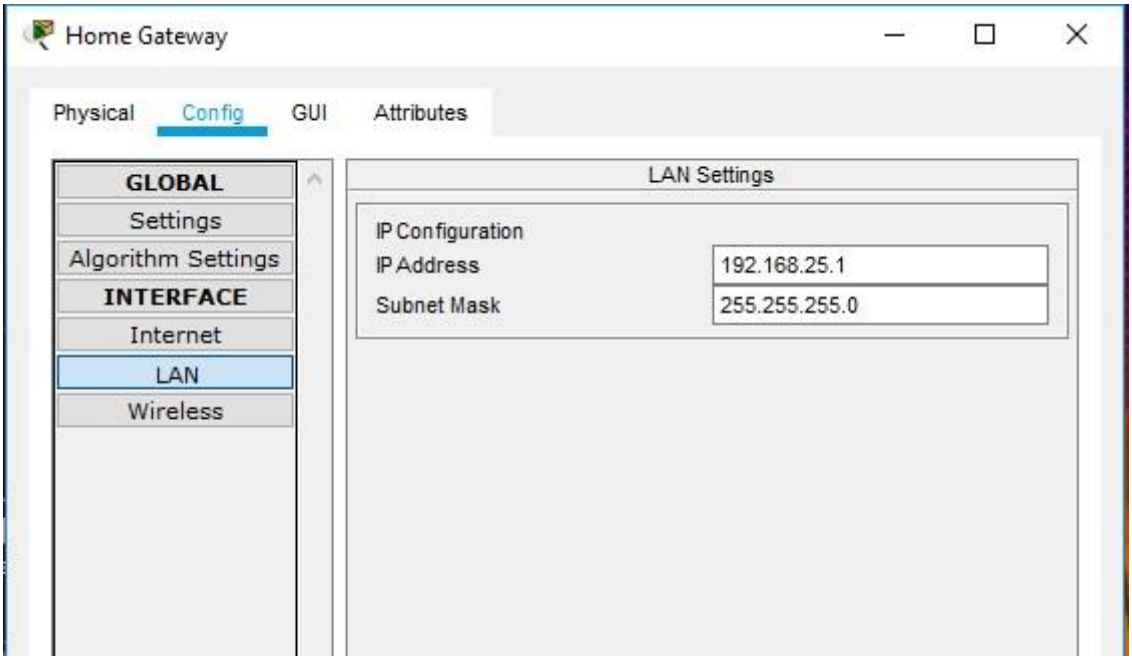


Рис. 32 – Окно настроек LAN главного шлюза.

Тут мы видим IP адрес сети. Нажмём на Wireless и увидим настройки беспроводной сети. Затем перейдите на вкладку «Конфигурация», а затем в левой панели щелкните «ЛВС», чтобы просмотреть настройки локальной сети главного шлюза. Запишите IP-адрес домашней сети для дальнейшего использования. Нажмите «Беспроводная связь» (Wireless) в левой панели, чтобы просмотреть настройки беспроводной сети домашнего шлюза.

Запишите SSID домашней сети и WPA2-PSK пароль для дальнейшего использования. Закройте окно Home Gateway.

Затем щелкните значок устройства планшета (**Tablet PC**), чтобы открыть окно планшета. В окне «Планшет» (**рис.33**) выберите вкладку «Рабочий стол», а затем щелкните значок «Веб-браузер». В окне веб-браузера введите IP-адрес Home Gateway **192.168.25.1** в поле URL и нажмите «Перейти». На экране входа в Home Gateway введите admin для имени пользователя и пароля и нажмите «Отправить».

В окне веб-браузера (**рис. 34**) введите IP-адрес Home Gateway 192.168.25.1 (адрес панели администрирования Home Gateway по умолчанию) в поле URL и нажмите «Перейти». На экране входа в Home Gateway введите admin для имени пользователя и пароля и нажмите «Отправить».

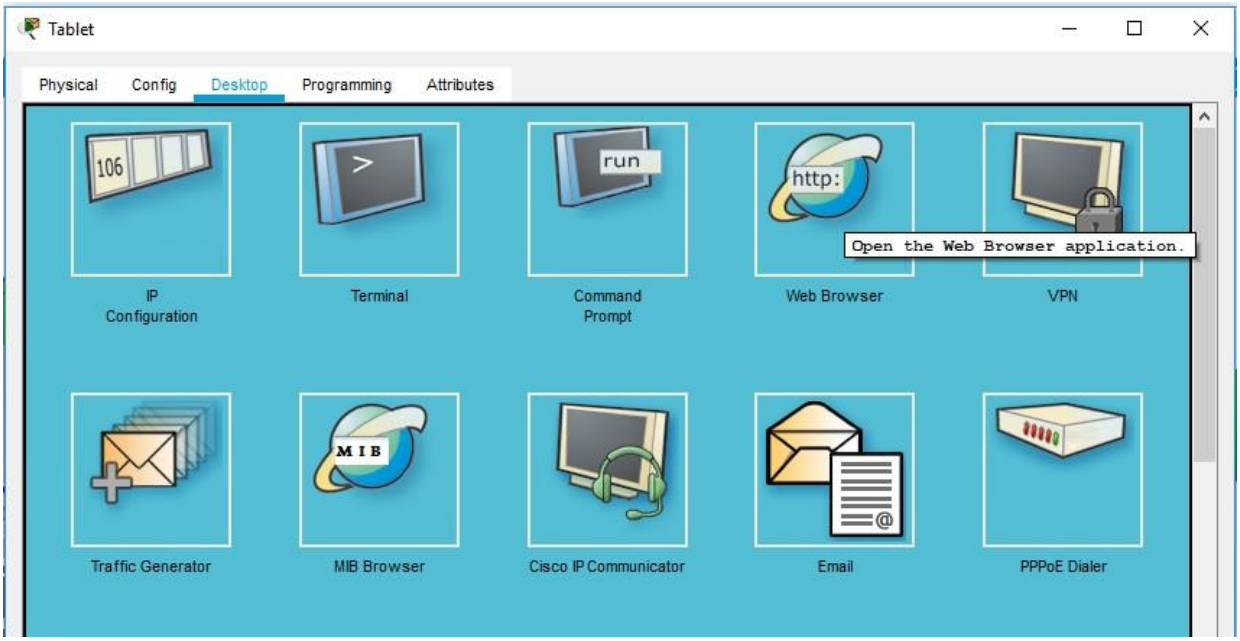


Рис. 33 – вкладка Desktop

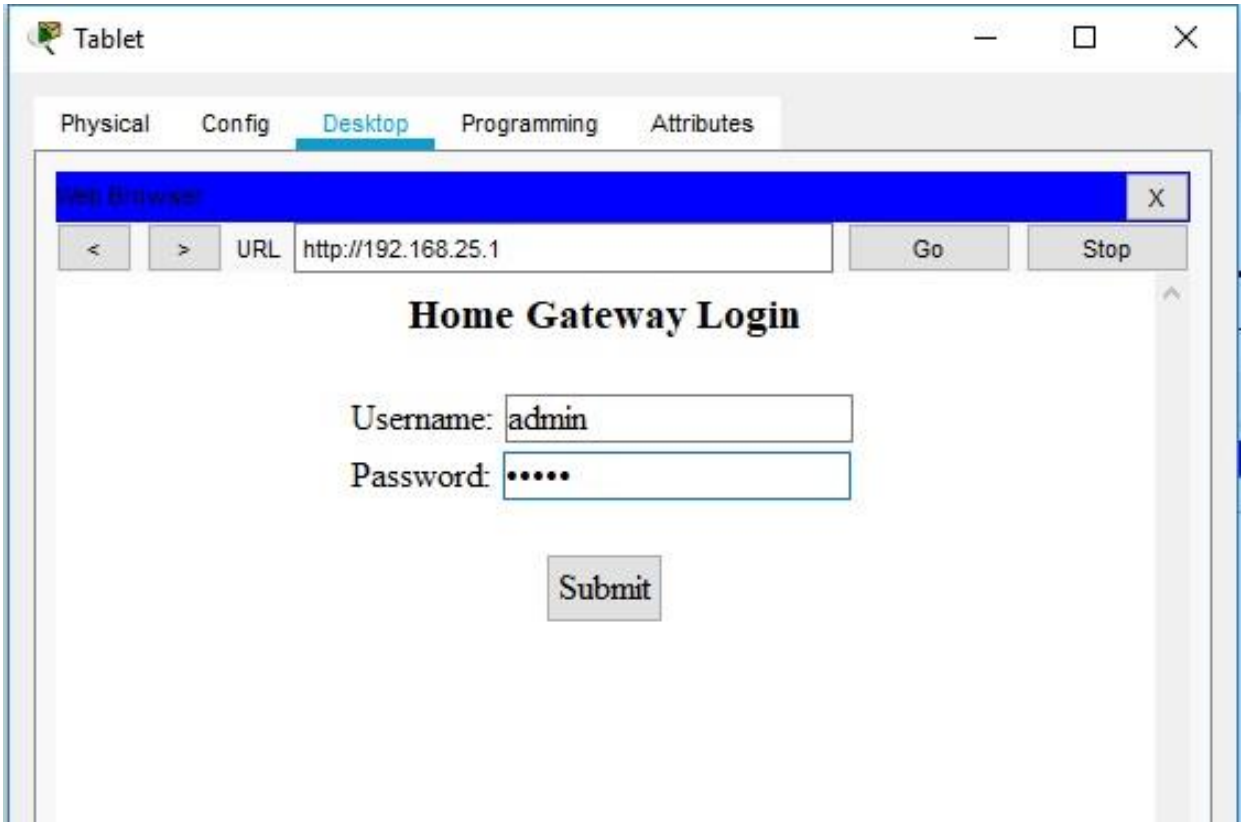


Рис. 34 – Панель сетевого администрирования Home Gateway

После того, как вы подключились к веб-интерфейсу Home Gateway, появится список всех подключенных устройств IoT. Когда вы нажимаете на устройство в списке, отображается состояние и настройки этого устройства.

В своем отчете отразите все подключенные устройства, отразив их тип, PTT (серийный номер).

Добавление проводных устройств ввода-вывода в интеллектуальную домашнюю сеть.

Подключение устройства в сеть с помощью кабеля.

а. В поле «Выбор устройства» выберите значок «Газонный разбрызгиватель»(Lawn Sprinkler), а затем щелкните в рабочей области, где вы хотите разместить разбрызгиватель.

б. Присоединение газонного разбрызгивателя к домашнему шлюзу.

В поле «Выбор типа устройства» щелкните значок [Подключения] (это выглядит как молния). Щелкните значок типа соединителя Copper Straight Through в поле «Выбор устройства». Затем нажмите значок «Разбрызгиватель» и подключите один конец кабеля к интерфейсу FastEthernet0 Sprinkler. Затем щелкните значок Home Gateway и подключите другой конец кабеля к доступному интерфейсу Ethernet.

Настройка разбрызгивателя для сетевого подключения

а. Нажмите значок устройства разбрызгивателя в рабочей области, чтобы открыть окно устройства. Обратите внимание: прямо сейчас имя разбрызгиватель для газона является общим IoT0.

Окно устройства откроется на вкладке «Спецификация» (рис.35), которая дает информацию об устройстве, которое может быть отредактировано.

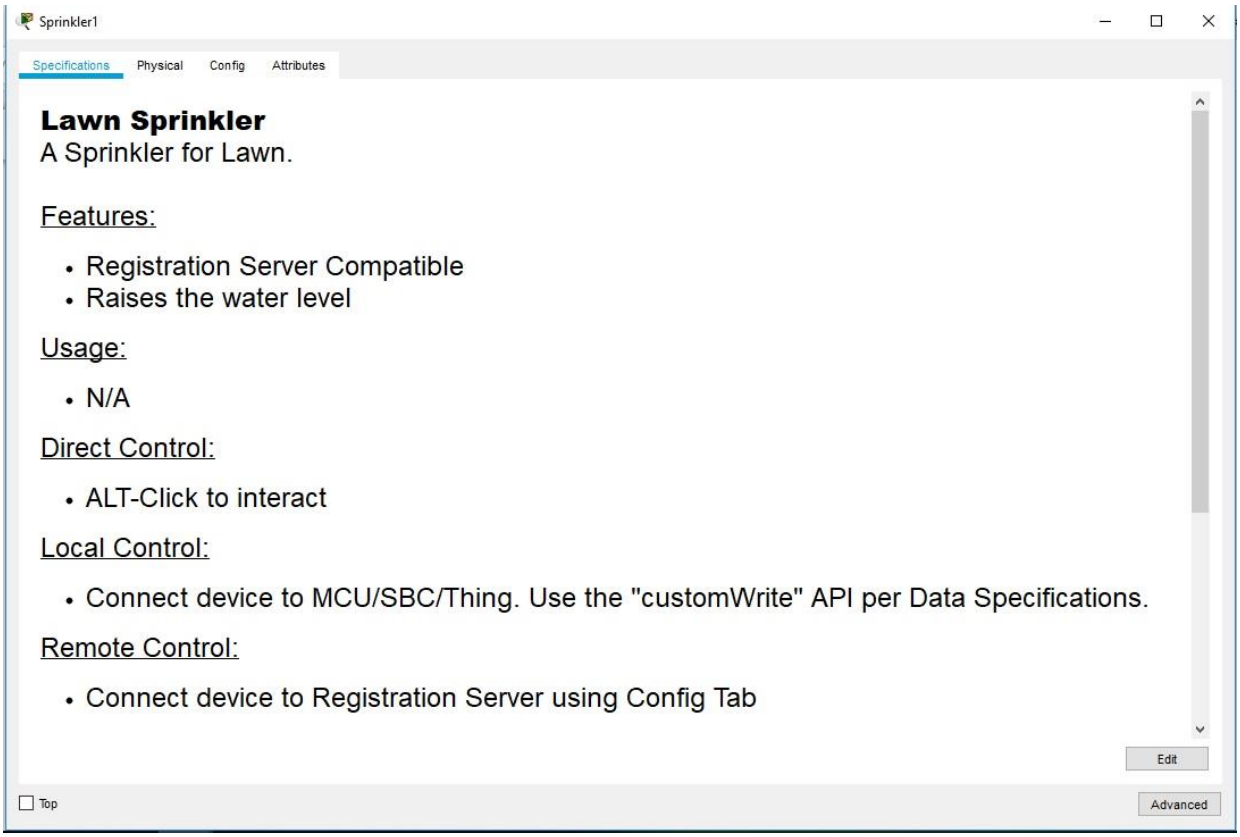


Рис. 35 – Окно спецификаций

6. Перейдите на вкладку «Конфигурация», чтобы изменить настройки конфигурации устройства. На вкладке «Конфигурация» (рис. 36) внесите следующие изменения в «Настройки»:

— Установите отображаемое имя в Sprinkler1 (обратите внимание, что имя окна изменяется на Sprinkler1).

— Установите сервер IoT на домашний шлюз.

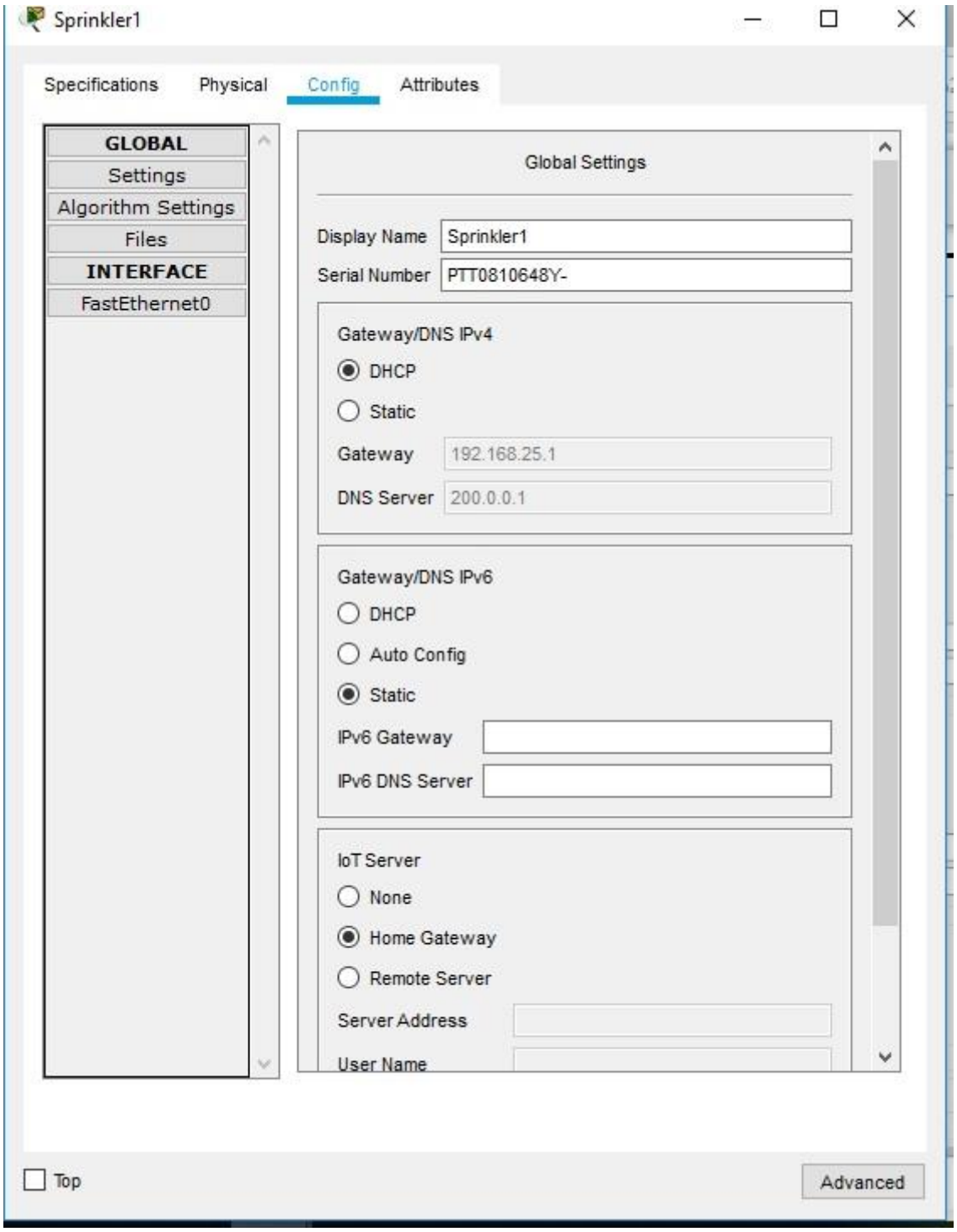


Рис. 36 – Вкладка конфигурации IoT-устройства

Нажмите FastEthernet0 и измените IP-конфигурацию на DHCP. Закройте окно разбрызгивателя.

c. Убедитесь, что разбрызгиватель находится в сети. Войдите в Home Gateway из планшета. Устройство Sprinkler 1 теперь должно появиться в списке IoT Server — Devices.

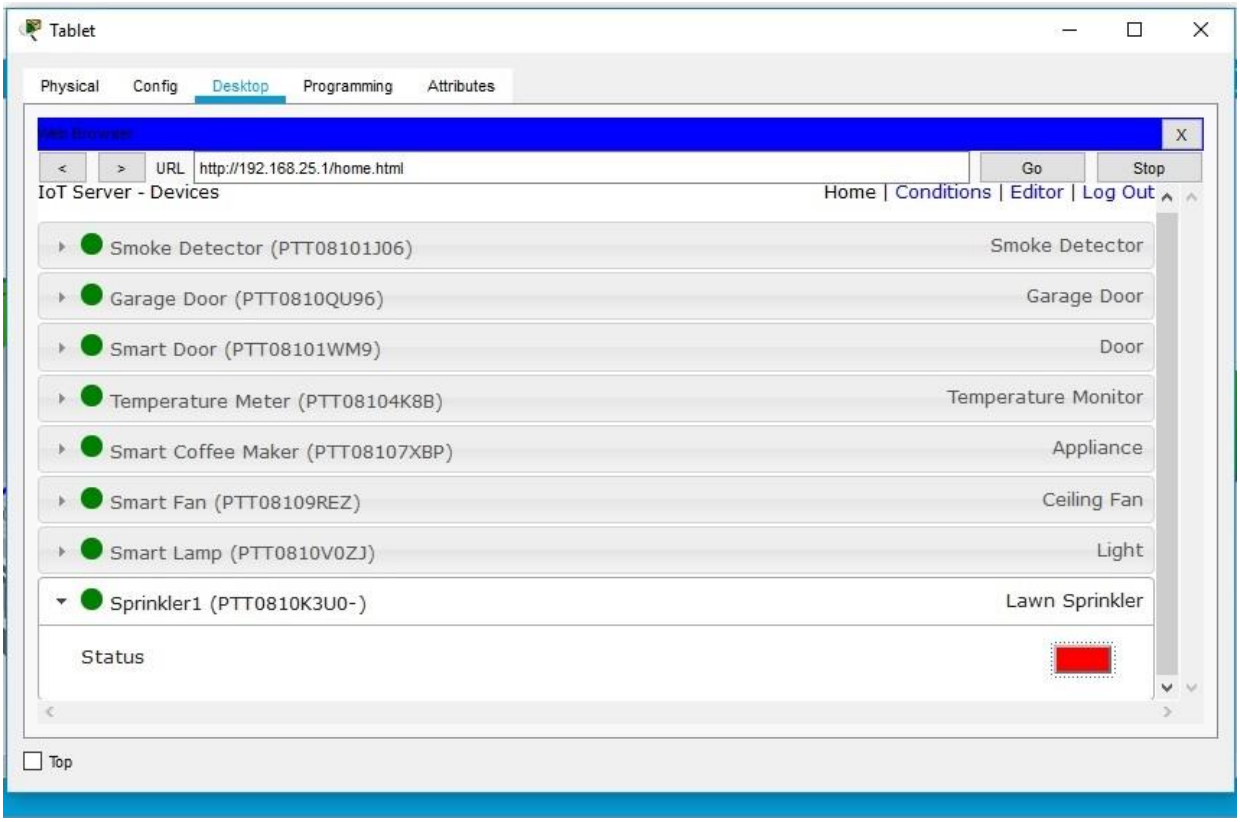


Рис. 37 – Status Sprinkler

Измените статус разбрызгивателя на включен. Для этого щёлкните по красной кнопке.

У кнопки должен поменяться цвет на зелёный. Закройте окно планшета. Экспериментируйте, добавив другие типы IoT-устройств в интеллектуальную домашнюю сеть.

Рис. 38 – Включенный IoT-разбрызгиватель воды

Добавление беспроводных устройств ввода-вывода в интеллектуальную домашнюю сеть

Добавление беспроводного устройства в сеть.

В поле «Выбор конкретного устройства» щелкните значок «Детектор ветра», а затем щелкните в рабочей области, где вы хотите разместить детектор ветра (Wind Detector).

Добавьте беспроводной модуль в детектор ветра.

Нажмите значок Wind Detector в рабочей области, чтобы открыть окно устройства IoT. В правом нижнем углу окна устройства IoT нажмите кнопку «Дополнительно». Обратите внимание, что в верхней части окна видны больше вкладок. Перейдите на вкладку «Конфигурация ввода-вывода» (рис.39).

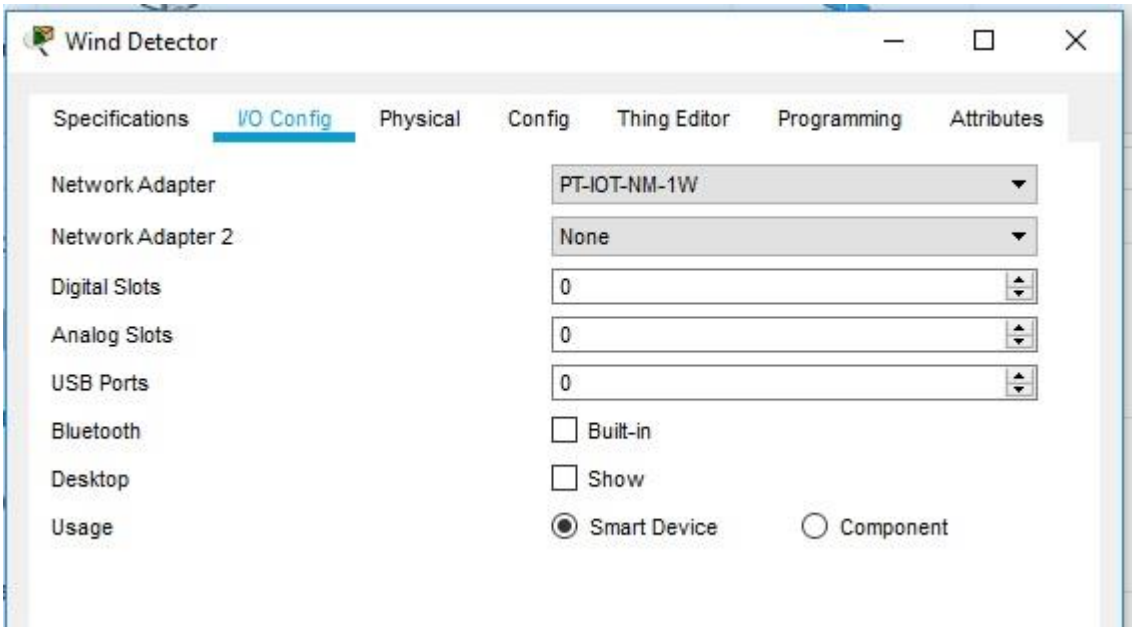


Рис. 38 – Вкладка «Конфигурация ввода-вывода»

Измените выпадающий список Network Adapter на PT-IOT-NM-1W, который является беспроводным адаптером.

Настройте детектор ветра для беспроводной сети.

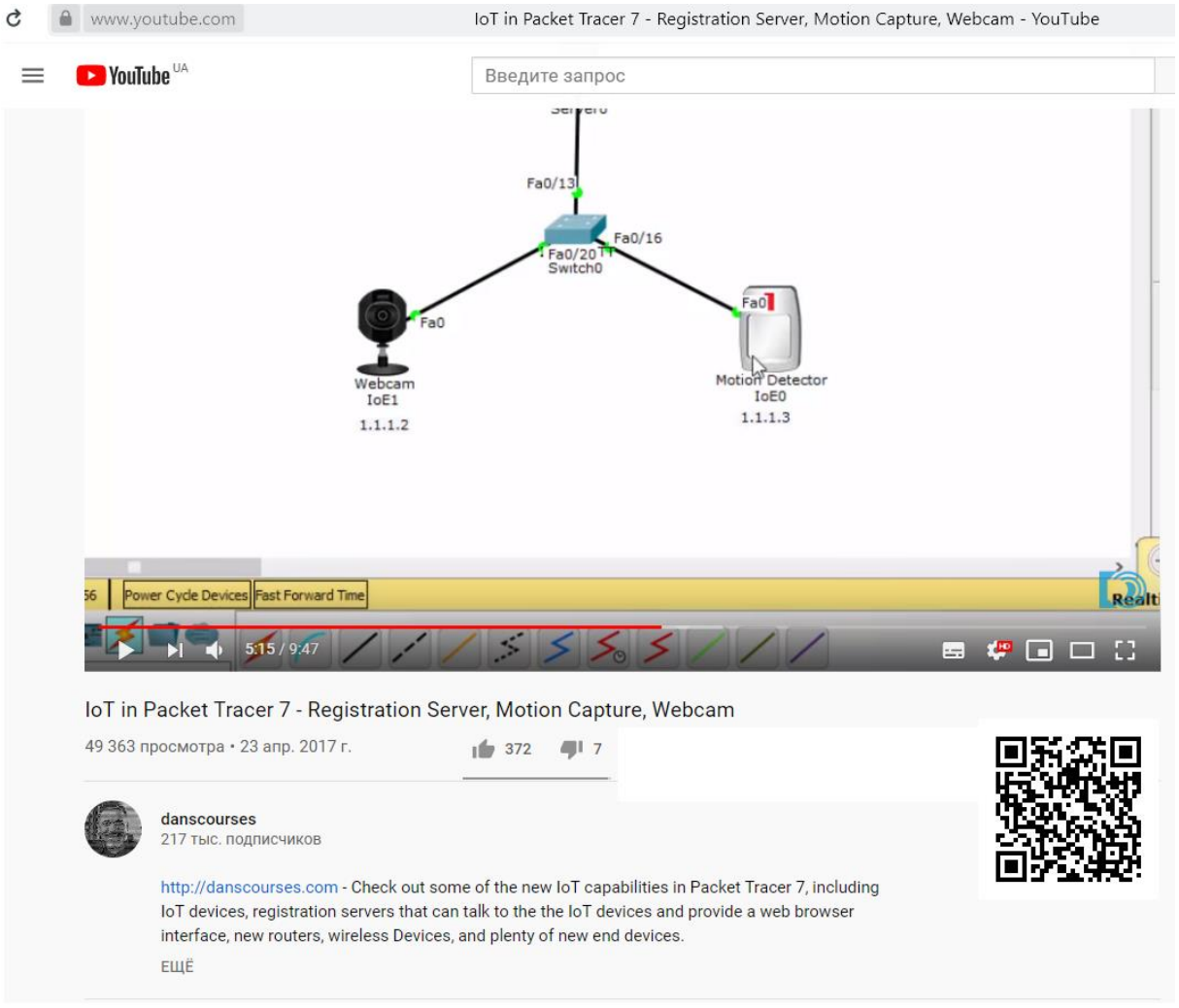
Перейдите на вкладку Конфигурация. Измените отображаемое имя на Wind_Detector и измените IoT-сервер на Home Gateway. Затем щелкните Wireless0 в левой панели. Измените тип аутентификации на WPA2-PSK и в поле PSC Pass Phrase введите mySecretKey. Это настройки беспроводной сети домашнего шлюза (Home Gateway), которые вы записали ранее.

Между детектором ветра и Домашним шлюзом должно быть установлено беспроводное соединение (полосы-волны между устройствами в общей схеме). Убедитесь, что детектор ветра находится в сети (проверьте через админ.панель)

Войдите в Home Gateway с помощью планшета. Устройство Wind Detector теперь должно появиться в списке IoT Server — Devices.

Экспериментируйте, добавив другие типы устройств IoT в интеллектуальную домашнюю беспроводную сеть. Можно так же согласовать работу двух устрой. Посмотрите в **видео** (**QR-на текущей странице - IoT in Packet Tracer 7 - Registration Server, Motion Capture, Webcam**), как можно согласовать работу датчика движения и вебкамеры.

В отчете вы должны приложить скриншот таблицы маршрутизации Home Gateway, описать его технические характеристики (количество портов LAN/WAN), приложить сетевые настройки для всех подключенных устройств. Также вам необходимо будет подключить Home Gateway проводным образом к любой локальной проводной (не IoT сети), **сделанной ранее вами**.



§ VIII. Модель межмашинного взаимодействия(M2M).

Представьте себе обширное предприятие, оснащенное «умным» оборудованием и технологиями, основанными на радиометках, — все машины соединены друг с другом и общаются в рамках производственного процесса с помощью датчиков и исполнительных механизмов. Операторы пользуются планшетами, связываясь с производственными системами для диагностики и управления. Данные о загруженности и работоспособности оборудования, а также диагностика накапливаются в корпоративных системах планирования ресурсов и оптимизации производства. Взамен оборудование получает команды подстройки производственного цикла, оптимизирующие соотношение затрат и качества. Машины также «общаются» со своими производителями, по мере необходимости заказывая ремонт и запчасти, чтобы избежать простоев. Системы, основанные на агентских модулях, распределяют нагрузку между производственными линиями, действующими в разных регионах мира, помогая оптимизировать затраты на цепочку поставок. Все это уже реальность — так называемые умные фабрики, а фабрики будущего смогут адаптироваться к новым требованиям гораздо быстрее, чем нынешние решения для гибкого производства. Умная фабрика соединяет машины, логистику и людей, обеспечивая оперативную повсеместную координацию.

Межмашинное взаимодействие (машинно-машинное взаимодействие, англ. Machine-to-Machine, M2M) — это как раз и есть общее название технологий, которые позволяют машинам обмениваться информацией друг с другом, или же передавать её в одностороннем порядке. Это могут быть проводные и беспроводные системы мониторинга датчиков или каких-либо параметров устройств (температура, уровень запасов, местоположение и т. д.). К примеру, банкоматы или платёжные терминалы могут автоматически передавать информацию по GSM-сетям, а также если у них закончилась чековая бумага или наличность, или же наоборот, что наличности слишком много и требуется приезд инкассаторов.

M2M также активно используется в системах безопасности и охраны, вендинге, системах здравоохранения, промышленных телеметрических системах (производство, энергетика, ЖКХ и др.) и системах позиционирования подвижных объектов на основе систем ГЛОНАСС/GPS. Одним из подклассов M2M является межмашинное взаимодействие с использованием мобильных решений, для него также может использоваться аббревиатура M2M (англ. Mobile-to-Mobile). Принципы, по которым различные полевые устройства соединяются с ИТ-системами предприятия, применимы не только к автоматизации и производственной отрасли.

Основное различие между IoT и M2M заключается в том, что IoT (Интернет вещей) использует беспроводную связь, в то время как M2M (межмашинное взаимодействие) может использовать как проводную, так и беспроводную связь. IoT подключает интеллектуальные устройства к сети для сбора данных, анализа и принятия разумных решений, а M2M позволяет устройствам связываться и выполнять необходимые действия без участия человека.

IoT — это Интернет вещей, а M2M — межмашинное взаимодействие. IoT подключает интеллектуальные устройства к сети для сбора данных, анализа и принятия разумных решений. С другой стороны, M2M позволяет устройствам связываться и выполнять необходимые действия без участия человека. IoT использует беспроводную связь, тогда как M2M может использовать проводную или беспроводную связь.

IoT требует активного подключения к Интернету, в то время как у M2M оно может отсутствовать. Требование к интернет-соединению у M2M зависит от работающего приложения, IoT сильно зависит от интернет-соединения и наличия «облака», тогда как M2M в часто полагается на проводную сеть.

Сегодня мир более связан, чем когда-либо. Развитие технологий объединяет не только людей, но и устройства и машины по всему миру. IoT и M2M — это две технологии, которые помогают повысить производительность, эффективность, точность и улучшить общее качество жизни. M2M и IoT очень являются очень близкими технологиями, но IoT является более новой из них. M2M является основой для IoT. Системы на базе IoT и M2M автоматически контролируют себя, реагируют на изменения и выполняют задачи без вмешательства человека. В частности, межмашинное взаимодействие (Machine-to-Machine, M2M) уже осуществляется в транспортной индустрии для диагностики автомобилей и их соединения с информационными системами. Средства M2M применяются в имплантируемых медицинских устройствах и глобальных логистических компаниях.

§ IX. Организация межмашинного взаимодействия устройств сети с носимым айтрекером*

В статье-обзоре, которую я предлагаю к прочтению, предлагается метод организации сетевой коммуникации устройств, присутствующих в повседневной жизни человека. Для связи устройств используется описываемый протокол CoAP, предназначенный для обмена сообщениями между устройствами с ограниченными ресурсами в целях экономии потребляемой электроэнергии.

По заявлению авторов: подобная сеть призвана эффективно и экономично способствовать повышению качества жизни людей. Механизм работы предлагаемого метода рассматривается на примере носимого дисплея дополненной реальности, который устанавливает соединение с компьютером по выводимому на монитор изображению идентификационного QR-кода. В результате дисплей получает возможность передать управление курсором мыши на мониторе компьютера пользователя встроенному айтрекеру.

Организуемая сеть демонстрирует высокую производительность, адаптивность к изменениям и модификациям, а также поддерживает автоматическое обновление программного обеспечения для всех элементов системы.

Для цитирования: Ершова О.А., Гусев А.П., Андреев А.М. Организация межмашинного взаимодействия устройств сети с носимым айтрекером // Вопросы радиоэлектроники. 2018. № 2. С. 151–158.
УДК 004.75

О.А. Ершова[1], А.П. Гусев[1, 2], А.М. Андреев[2]

[1] ПАО «ИНЭУМ им. И.С. Брука», [2] МГТУ им. Н.Э. Баумана

ОРГАНИЗАЦИЯ МЕЖМАШИННОГО ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ СЕТИ С НОСИМЫМ АЙТРЕКЕРОМ

Предлагается метод организации сетевой коммуникации устройств, присутс ловека. Для связи устройств используется протокол CoAP, предназначенны устройствами с ограниченными ресурсами в целях экономии потребляемой звана эффективно и экономично способствовать повышению качества жизн гаемого метода рассматривается на примере носимого дисплея дополненно ет соединение с компьютером по выводимому на монитор изображени В результате дисплей получает возможность передать управление курсор пользователя встроенному айтрекеру. Организуемая сеть демонстрирует в тивность к изменениям и модификациям, а также поддерживает автоматичес спечения для всех элементов системы.

Ключевые слова: *межмашинное взаимодействие, сетевые коммуникации, полненной реальности.*

Список использованных источников.

1. Одом У. Компьютерные сети. Первый шаг. Computer Networking: First-step / Пер. В. Гусев. — СПб.: «Вильямс», 2006. — 432 с. — (Первый шаг). — 3 000 экз. — ISBN 5-8459-0881-7. Таненбаум Э, Уэзеролл Д. Компьютерные сети. — Питер, 2012. — 960 с.
2. Малиновский Б.Н. История вычислительной техники в лицах. - К.: фирма "КИТ", ПТОО "А.С.К.", 1995. - 384 с.
3. Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: Учебное пособие для вузов / Сергей Александрович Орлов. - СПб.: Питер, 2002. - 464 с.: ил. - (Учебник для вузов).
4. Кирсанов, Э.А. Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы [Электронный ресурс]: учебное пособие / Э.А. Кирсанов, А.А. Сирота. — Электрон. дан. — Москва : Физматлит, 2012. — 344 с.
5. Сборник методических указаний для выполнения практических заданий и лабораторных работ курса "IoT Академия Samsung" [Электронный ресурс]: — Режим доступа: http://timp.keva.su/samsungIoT.7z (дата обращения: 19.11.2019).
6. Леонид Черняк. Платформа Интернета вещей. Открытые системы. СУБД, №7, 2012. Открытые системы (26 сентября 2012).
7. Росляков А.В., Ваняшин С.В., Гребешков А.Ю. Интернет вещей. Учебное пособие. — Самара: ПГУТИ, 2015. — 200 с.
8. Кенин, Александр Практическое руководство системного администратора / Александр Кенин. - М.: БХВ-Петербург, 2013. - 766 с.
9. Тихвинский В., Коваль В., Бочечка Г. Перспективы стандартизации интернета вещей в международных организациях связи//Первая миля. 2017. № 2 (63). С. 26 -32.
10. Алексеев В. Технологии «Интернета вещей» для сетей ISM не лицензируемого диапазона частот//Беспроводные технологии. 2017. Т. 1. № 46. С. 44 -50.
11. Шешалевич В.В. LPWAN - низкопотребляющие сети большого радиуса действия. Связь для интернета вещей//Безопасность информационных технологий. 2017. № 3. С. 6 -16.

12. Motlagh N. H., Taleb T., Arouk O. Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives//IEEE Internet of Things Journal. New Jersey, US: IEEE, 2016, pp. 899 -922.
13. Терентьев М.Н. Обзор публикаций, посвящённых самоорганизации беспроводных сенсорных сетей//Труды МАИ. 2017. № 94. URL: http://trudy.mai.ru/published.php?ID=81149
14. Трифонова С.В., Холодов Я.А. Исследование и оптимизация работы беспроводной сенсорной сети на основе протокола ZigBee//Компьютерные исследования и моделирование. 2012. Т. 4. № 4. С. 855 -869.
15. Бородин В.В., Петраков А.М., Шевцов В.А. Анализ алгоритмов маршрутизации в сети связи группировки беспилотных летательных аппаратов//Труды МАИ. 2016. № 87. URL: http://trudymai.ru/published.php?ID=69735
16. Шешалевич В.В. LPWAN -низкопотребляющие сети большого радиуса действия. Связь для интернета вещей//Безопасность информационных технологий. 2017. № 3. С. 6 -16.
17. Беспроводные сенсорные сети: обзор. Акулдиз И.Ф. - Перевод с английского: Левжинский А.С. [Электронный ресурс]: — Режим доступа: http://masters.donntu.org/2011/fknt/levzhinsky/library/translate.htm (дата обращения: 9.01.2020).

Учебно-практическое издание

«Дом, который построил сам себя. Сетевой практикум. IoT. »

Практикум

2020 г.

GLÜCKSRITTER
MUNISTER

§ INSCRIPTUM

Лабораторный практикум - существенный элемент учебного процесса в профессиональном учебном заведении, в ходе которого обучающиеся фактически впервые сталкиваются с самостоятельной практической деятельностью в конкретной области. Лабораторные занятия, как и другие виды практических занятий, являются средним звеном между углубленной теоретической работой обучающихся на лекциях, семинарах и применением знаний на практике. Эти занятия удачно сочетают элементы теоретического исследования и практической работы. Выполняя лабораторные работы, студенты лучше усваивают программный материал, так как многие определения и формулы, казавшиеся отвлеченными, становятся вполне конкретными, происходит соприкосновение теории с практикой, что в целом содействует уяснению сложных вопросов науки и становлению обучающихся как будущих специалистов.

Само значение слов «лаборатория», «лабораторный» (от латинского labor труд, работа, a laboro - трудиться, стараться, хлопотать, преодолевать затруднения) указывает на сложившиеся понятия, связанные с применением умственных и физических усилий к изысканию ранее неизвестных путей и средств для разрешения научных и жизненных задач.

Неслучайно слово «практикум», применяемое для обозначения определенной системы практических (преимущественно лабораторных) учебных работ, и выражает ту же основную мысль (греческое - praktikos), означает «деятельный», это значит, что предполагаются такие виды учебных занятий, которые требуют от обучающихся усиленной деятельности.

В целях создания интегрированного курса (выраженного теорией в учебно-теоретическом издании «Компьютерные сети. IoT и Межмашинное взаимодействие» с поддержкой интер-отклика) был создан лабораторный практикум «Дом, который построил сам себя. Сетевой практикум. IoT».

Надеюсь, что разработанный мной практикум, базирующийся на концепции обучения (сертификации) сетевых инженеров Cisco (CCNA) и программам интродукции (введения) в архитектуру как IoT от IBM, так и компьютерных и информационных сетей, систем и инструментов дискретной математики, поможет вам в профессиональной деятельности.

Мунистер В.Д.

§ СОДЕРЖАНИЕ

«Дом, который построил сам себя. Сетевой практикум. IoT »

INSCRIPTUM 120

СОДЕРЖАНИЕ КУРСА 121

ДИСКРЕТНАЯ МАТЕМАТИКА В IoT: ТЕОРИЯ АВТОМАТОВ 127

ПРАКТИКУМ: АКТУАТОР КАК КОНЕЧНЫЙ АВТОМАТ 137

ОСНОВЫ СЕТЕВОГО ПЛАНИРОВАНИЯ И УПРАВЛЕНИЯ 143

ОБЗОР ЦИКЛА ПРОЕКТИРОВАНИЯ ККС 148

РАЗРАБОТКА СТРУКТУРЫ IoT-ПЛАТФОРМЫ 155

ПРАКТИКУМ: СОЗДАНИЕ СЕТЕВОЙ МОДЕЛИ ИНФРАСТРУКТУРЫ IoT 162

ПРАКТИКУМ: СОЗДАНИЕ SMART CAMPUS 164

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПОТОКОВ ДАННЫХ В БОЛЬШИХ СЕТЯХ 188

ОЦЕНКА РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ… 196

ПРАКТИКУМ: ZERO TRUST в IoT 230

ПРАКТИКУМ: СЛАУ в IoT-ИНФРАСТРУКТУРЕ 246

ПРАКТИКУМ: АЛГОРИТМЫ ДЕЙКСТРА И КРАСКАЛЯ для IoT 249

ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 250

NODE-RED - ГРАФИЧ. КОНФИГУРАТОР ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ … 255

ПРАКТИКУМ: СОЗДАНИЕ ПЕРВОГО IoT-приложения… 256

Li-Fi: БУДУЩЕЕ ИНТЕРНЕТА… 259

ПРИЛОЖЕНИЕ. Обзор открытых IoT-платформ 262

§ СОДЕРЖАНИЕ КУРСА

Данное издание предназначено для получения практических навыков и умений посредством организации выполнения индивидуальных заданий в рамках проведения учебных практик как рекомендованного решения использования в контексте конкретных междисциплинарных курсов, связанных с принципами организации межсетевого взаимодействия, архитектуры информационных систем, в частности, в рамках платформы «Internet of things» и визуальным программированием, дидактикой преподавания математических дисциплин, вопросами информационной безопасности:

«Организация, принципы построения и функционирования компьютерных систем», «Математический аппарат для построения компьютерных систем», «Дизайн архитектуры распределенных сетей», «Инфокоммуникационные системы и сети», «Информационные технологии», «Внедрение и поддержка программного обеспечения компьютерных систем», «Компьютерные и телекоммуникационные сети», «Встроенные компьютерные системы».

Подразумевается, что проведение практикума необходимо рассматривать неразрывно, в конкретно определенной последовательности. Однако, строгая итерационная и типизационная составляющая в выполнении элементов практикума не выражена рамочно. Элементарным преобразователем неделимой части (отраженной в виде главы) практикума может быть самостоятельное проведение лабораторной работы при условии вовлечения в рассматриваемый контекст учащегося (осваивающего данный элемент программы).

Также необходимо понимать правило достаточного условия для возможности допуска к заданиям, овладение базового уровня теоретических знаний по двум группам дисциплин: математического и общего естественно-научного учебного цикла (**группа I**) и общепрофессионального цикла (**группа II**):

Группа I: «Элементы высшей математики», «Дискретная математика».

К II группе относятся такие дисциплины как: «Теория информации и кодирования», «Технологии защиты информации», «Технические средства информатизации», «Основы программирования и баз данных», «Архитектура аппаратных средств», «Проектирование цифровых устройств», «Программное обеспечение компьютерных сетей», «Теория принятия решений», «Разработка мобильных и встроенных специализированных систем», «Компьютерная логика», «Разработка прикладных решений на базе современных платформ», «Системный анализ», «Технологии реинжиниринга и бизнес-инжиниринга».

Общая структура данного издания представлена таким образом, чтоб раскрыть наиболее значимые элементы содержания методологии этих двух групп. Целью разработки практикума было создание аддитивного эффекта от преподаваемых дисциплин в ключе расширения ассоциативного ряда у студентов (обучающихся) создания, придания интерактивности. Но прежде всего – для получения четкого понимания, для чего нужны те или иные инструменты (компоненты) в осознанной профессиональной деятельности.

В основе итерационной последовательности неделимых элементов практикума (можно использовать обозначение – блок, контейнер) лежат базовые догмы класса прикладных методов управления проектами (в рамках общепринятого обозначения эти методы имеют общее название - «Сетевое планирование и управление», использующееся в бизнесе несколько десятков лет) в упрощенном формализованном виде, обеспечивающим приведение к планированию, и даже осуществить анализ сроков выполнения (ранних и поздних) нереализованных частей проектов; что позволяет увязать выполнение различных работ и процессов во времени, составить сетевой график, получив прогноз общей продолжительности реализации всего проекта.

Общая задумка концепции практикума с применением методов сетевого планирования и управления несет себе и одну далеко идущую цель (идеал), не совсем тривиальную задачу – создать интродукцию (первичное введение) в системный анализ – научный метод познания, представляющий собой последовательность действий по установлению структурных связей между переменными или постоянными элементами исследуемой системы, который, собственно и опирается на комплекс общенаучных, экспериментальных, естественнонаучных, статистических, математических методов и прямо востребован среди тех, кто планирует освоить магистерскую программу. А также аспирантскому сообществу — будущей научной интеллигенции.

Вернемся, однако, к структурному представлению, обращая внимание на предназначение вышеперечисленных неделимых блоков (контейнеров) практикума и описанию **целевой карты** по каждой из позиции:

I. «**Актуатор как конечный автомат**» — синтез «Теории алгоритмов», «Теории цифровых автоматов», «Дискретной математики» с усвоением важнейшего термина из учебного издания «Компьютерные сети. IoT & межмашинное взаимодействие» – **актуатора** (исполнительного устройства). Этот термин взят из теории автоматического управления.

Под исполнительным устройством понимают устройство, передающее воздействие с управляющего устройства на объект управления.

Иногда он рассматривается как составная часть объекта управления. Управляющим устройством может быть любая динамическая система (в учебно-теоретическом издании «Компьютерные сети. IoT и межмашинное взаимодействие» этот вопрос освещен в главах VII-IX.

Перед данным блоком содержатся теоретические сведения, оформленные в виде главы с названием «Дискретная математика в IoT: теория автоматов». В практическом боксе происходит последовательная пошаговая процедура алгоритмизации решения конкретной (прикладной) задачи – создание конечного автомата цифрового оконечного устройства с последующим исполнением в булевом базисе.

II. **«Создание сетевой модели инфраструктуры IoT»** – включает в себя работу по осуществлению сетевого планирования и управления: моделирование и визуализация сетевой модели разработки индивидуального технического задания – графического представление проекта. Данный метод планирования позволяет найти минимальные сроки завершения проекта на этапе отдельных работ в теоретизации решения задача, а также определить множество критических работ, увеличение продолжительности выполнения любой из которых приводит к увеличению времени выполнения всего проекта.

Теоретической основой для этого элемента практикума являются главы: "Основы сетевого планирования и управления" "Обзор цикла проектирования ККС" "Разработка структуры IoT-платформы".

III. **«Создание Smart Campus»** — гайдлайн IoT на практике в рамках симулятора Cisco Packet Tracer. В этом модуле обучающиеся вплотную познакомятся с сетевой топологией «умного» кампуса, конфигурацией IoT-сети, типовой реализацией Smart-Industrial и Blockly custom software for IoT Simulations. Данный модуль практикума является теоретически-прикладным, основанным на мануале «IoT Simulations with Cisco Packet Tracer» магистра Университета прикладных наук Метрополия (г. Хельсинки, Финляндия) Andrea Finardi.

Индивидуальные задания как таковые отсутствуют, так как подразумевается вынесение их в виде отдельного модуля – закладывая, тем самым основы проведения хакатона, семинара, где сами участники будут формировать техническое здание и проводить тренинг в Cisco Packet Tracer.

IV. **«Zero trust в IoT»** — позволяет раскрыть специфику Internet of Things с кардинально новой стороны, доводя до студента общую проблематику информационной безопасности этой платформы. Уделено внимание понятию «нулевого доверия», IoT DDoS-атакам. Студенты научатся проводить оценку рисков и угроз информационной безопасности сетям.

Контейнер базируется на содержании глав «Оценка рисков информационной безопасности», «Определение топологии в самоподобных множествах», в которых, происходит ознакомление с политиками

информационной безопасности, началами фрактальной геометрии в упрощенном кратком виде (с целью ознакомления, назидательной дидактики преподавания смежных дисциплин).

V. **«СЛАУ в IoT-инфраструктуре»** — модуль, который дает возможность использовать потенциал линейной алгебры, в частности, решение систем линейных уравнений от трёх переменных, которые определяются как набор плоскостей, и, в свою очередь, в абстрактном виде могут выступать в виде геометрической реализации математического множества устройств одного из трех типовых элементов IoT-системы: координатора, маршрутизатора, конечного устройства (как это было описано в учебно-теоретическом пособии данного интегрированного курса). Данный модуль позволит определить некоторый баланс между устройствами этих трех типов, дать важные ориентиры, касающиеся проблематики избыточности низкопотребляющих устройств, работающих в полудуплексном режиме, и сетей двух семейств: NB-IoT (Narrow Band Internet of Things) и ZigBee.

Содержание данного модуля предназначено не столь для практического применения в реальной жизни (так как на данный момент не существует инструментов мат. аппарата, позволяющих строго определить нужные количественные соотношения), а столь для фактического закрепления результатов освоения путем создания лучшего ассоциативного эффекта, учитывая огромное значение линейной алгебры в информационных системах и технологиях.

VI. **«Алгоритмы Дейкстра и Краскала для IoT»** завершает математический цикл практикума. Теория графов финализирует представление об создании инфраструктуры, подтверждая первично полученные умения по теории графов в контейнере **«Создание сетевой модели инфраструктуры IoT».**

VII. **«Создание первого IoT-приложения»** - наиболее значимый в практическом представлении модуль, являющийся частью одного из курсов IBM Developer. Состоит из двух частей: **Первая** - создание IoT-приложения, превращающего смартфон в IoT-датчик (актуатор) при помощи Bluemix -— публично-облачной платформы, разработанной IBM. Платформа поддерживает несколько языков программирования и сред разработки, а также инструментов в стиле DevOps для построения, выполнения, развёртывания и управления приложениями в облаке.

Вторая – «Создания IoT-приложения pingGo». В этой части описано, как настроить рабочее пространство IBM Bluemix, как создать демонстрационное приложение с помощью инструмента Node-RED и как успешно отослать SMS-сообщения из этого приложения на свой мобильный телефон с помощью сервиса Twilio.

Node-RED — это flow-based инструмент, созданный для визуального программирования, разработанный IBM для совмещения вместе: устройств, API, онлайн-сервисов и IoT, Он работает на **Node.JS**, и был разработан для работы на относительно малопроизводительных микропроцессорных системах, таких как: Raspberry Pi. BeagleBone Black. Arduino, которые давно используются в образовательной сфере.

С учётом озвученных факторов Node-RED удобно использовать на шлюзах между различными сетями устройств интернета вещей функционирующих на собственных, как правило, более простых протоколах и традиционным интернетом, построенных на TCP/IP, UDP. В этом случае он позволит более оптимально использовать свободные ресурсы шлюза, работающего, как правило, на Linux. Графический конфигуратор, базирующийся на принципе «вершина» и «ребро», значительно упрощает разработку и повышает наглядность функционирования IoT-системы.

Теоретической основой для завершающего элемента практикума является глава «Node Red - графический конфигуратор для Интернета Вещей» и содержимое самого контейнера в пошаговом формате.

Приложения, оформленные в конце издания, дополняют и уясняют вопросы курса в сфере IoT и несут на себе справочно-обзорную роль.

Таким образом, целями практикума являются: закрепление и систематизация полученных в ходе лекционных курсов знаний и развитие и практических умений и навыков студентов, по налаживанию аналитической и организационной работы технического содержания в сфере организации корпоративных компьютерных сетей нового поколения и IoT-решений в организациях всех форм собственности, то есть, в местах будущего применения полученных знаний после освоенных программ среднего и высшего профессионального образования.

§ ДИСКРЕТНАЯ МАТЕМАТИКА В IoT: ТЕОРИЯ АВТОМАТОВ.

Термин «дискретная математика» начал входить в научный обиход на рубеже 50-х и 60-х гг. XX в. для обозначения ряда разделов математики, таких, как теория булевых функций, теория конечных автоматов, теория графов, теория кодирования и др., которые стали интенсивно развиваться в связи с необходимостью создания сложных управляющих систем и бурным прогрессом вычислительной техники.

Иногда в него вкладывают и более широкий смысл, определяя дискретную математику как область математики, занимающуюся изучением дискретных структур, которые возникают как в пределах самой математики, так и в её приложениях, т.е. включая в дискретную математику все математические дисциплины, имеющие дело с дискретными множествами. В этом смысле к дискретной математике относят и теорию чисел, и всю конечную алгебру, и некоторые другие классические разделы математики.

Исторически дискретная математика значительно старше своей сестры – математики непрерывных величин, т.к. с момента своего зарождения математика являлась в основном дискретной, ее знания представляли собой набор конкретных правил и служили для решения практических задач.

Таковой она в значительной степени и оставалась, пока в XVII веке запросы естествознания и техники не привели к созданию методов, позволяющих математически изучать движение, процессы изменения величин, преобразование геометрических фигур. С употребления переменных величин в аналитической геометрии и создания дифференциального и интегрального исчисления начинается период математики переменных величин. Великим открытиям XVII века является введенное Ньютоном и Лейбницем понятие бесконечно малой величины, создание основ анализа бесконечно малых (математического анализа), разработка понятий предела, производной, дифференциала, интеграла.

От Ньютона математика пошла в основном по непрерывному пути, так как обслуживала нужды физики, которая изучала непрерывные процессы (движение планет, процессы в жидкостях и газах и т. д.).

Возрождение дискретной математики в форме работ по теории множеств, математической логике, теории графов, комбинаторике относится к середине XIX в. и было вызвано исследованиями в области электрических сетей, моделей кристаллов и структур молекул, хотя отдельные работы появлялись и ранее.

Например, известное рассуждение Эйлера о Кенигсбергских мостах, считающееся началом теории графов, было опубликовано в 1736 г.

Однако наиболее интенсивное развитие всех разделов дискретной математики связано с возникновением кибернетики как науки об общих процессах управления в природе, технике и обществе, а также с появлением мощной вычислительной техники, способной эти процессы исследовать.

Знание теории множеств, алгебры, математической логики и теории графов совершенно необходимо для четкой формулировки понятий и постановок различных прикладных задач, их формализации и компьютеризации, а также для усвоения и разработки современных информационных технологий. Понятия и методы теории алгоритмов и алгебры логики лежат в основе современной теории и практики программирования.

В отличие от традиционной математики (математического анализа, линейной алгебры и др.), методы и конструкции которой имеют в основном числовую интерпретацию, дискретная математика имеет дело с объектами нечисловой природы: множествами, логическими высказываниями, алгоритмами, графами.

Благодаря этому обстоятельству дискретная математика впервые позволила распространить математические методы на сферы и задачи, которые ранее были далеки от математики. Примером могут служить методы моделирования различных социальных и экономических процессов.

Одной из особенностей дискретной математики является ее «разбросанность», в ней нет такого ядра, какое представляют в математике непрерывных величин разделы дифференциального и интегрального исчисления. Поэтому содержание конкретных курсов в сильной степени зависит от того, для студентов каких специальностей предназначается курс.

Из множества разделов дискретной математики в нашем учебно-практическом пособии включены лишь наиболее употребительные в теории управления: математическая логика, теория графов и некоторые вопросы теории конечных автоматов и теории кодирования, и теория множеств.

С более полным содержанием вы можете ознакомиться посредством интер-отклика, перейдя по QR-коду, расположенному в нижней части страницы. А мы пока рассмотрим базис **Теории Автоматов** — раздела дискретной математики, изучающий абстрактные автоматы — вычислительные машины, представленные в виде математических моделей — и задачи, которые они могут решать.

Дискретная математика: учебное пособие
Томского Политехнического Университета. / А.В. Воронин (2009)

Автомат представляет собой кибернетическую систему, перерабатывающую дискретную информацию и меняющую свое внутреннее состояние лишь в допустимые моменты времени. В каждый момент времени автомат находится в некотором состоянии и может изменить это состояние под действием входного сигнала. Определить понятие состояния для общего случая весьма трудно, слишком оно фундаментально.

Неформально состояние системы – это ее характеристика, однозначно определяющая ее дальнейшее поведение, все последующие реакции системы на внешние воздействия. На один и тот же сигнал автомат может реагировать по-разному, в зависимости от того, в каком состоянии находится в данный момент. Таким образом, в своих состояниях автомат запоминает свою историю, свое «концентрированное» прошлое.

Число возможных историй бесконечно велико, даже если вариантов входных воздействий не много. Однако на множестве предысторий можно ввести отношение эквивалентности.

При этом две предыстории попадут в один класс эквивалентности, если они приводят автомат в одно и то же состояние. Очевидно, автомату не нужно запоминать конкретные входные истории. Достаточно, чтобы он запоминал классы эквивалентности, к которым данные истории принадлежат.

Наиболее интересен случай, когда число классов эквивалентности и соответственно состояний **конечно**.

Такой преобразователь называется **конечным автоматом**. Функциональный преобразователь (логическая схема, автомат без памяти) может рассматриваться как конечный автомат с одним состоянием.

Общее теоретико-множественное определение конечного автомата может быть сформулировано следующим образом.

Определение: Абстрактным конечным автоматом называется шестерка объектов:

$$A = \{S, S_0, X, Y, \varphi, \psi\},$$

где S – конечное непустое множество состояний;

$S_0 \in S$ – начальное состояние;

X – конечное непустое множество входных сигналов (он же алфавит, может обозначаться заглавной сигмой);

Y – множество выходных сигналов (или F);

$\varphi : S \times X \to S$ – функция переходов (или Qt);

$\psi : S \times X \to S$ – функция выходов

По способу формирования функций выходов среди синхронных автоматов выделяют **автоматы Мили** и автоматы **Мура**. В автомате Мили функция выходов ψ определяет значение выходного символа по классической схеме абстрактного автомата. Она является функцией с двумя аргументами и символ y (t) в выходном канале обнаруживается только при наличии символа во входном канале x (t). Функции перехода и выхода для автомата Мили можно записать в виде:

$$s(t+=h) \qquad \varphi(s(t), x(t));$$
$$y(t+=h) \qquad \psi(s(t)).$$

Считается, что реализация автоматов Мили, как правило, более проста, но в них возникают проблемы с синхронностью формирования выходных сигналов. Между моделями Мили и Мура существует соответствие, позволяющее преобразовать закон функционирования одного из них в другой или обратно.

Способы задания конечных автоматов:

Существует несколько способов представления конечных автоматов, каждый из которых имеет свои достоинства и недостатки. Функционирование несложного автомата удобно представлять графически, с помощью диаграммы состояний, являющейся ориентированным графом, у которого состояния представлены вершинами, а дуги соответствуют переходам из одного состояния в другое. Для сложных автоматов графическое представление становится слишком громоздким. Более целесообразным в этом случае представляются различные табличные способы задания в форме таблиц переходов и выходов либо в форме матриц смежности направленного графа.

Ниже (на рис.1) приведен **пример автомата**, описывающего поведение отца, отправившего сына в школу: Сын приносит двойки и пятерки. Но отец не наказывает сына за каждую двойку.

Он понимает, что она может быть и случайной, поэтому выбрана более гибкая тактика, которая описывается автоматом.

В практикуме будут рассматриваться более упрощенные конечные автоматы в качестве индивидуальных заданий, однако, этот пример необходимо понять в полной мере, как из условий, описанных словами, строятся конструкции данного вида.

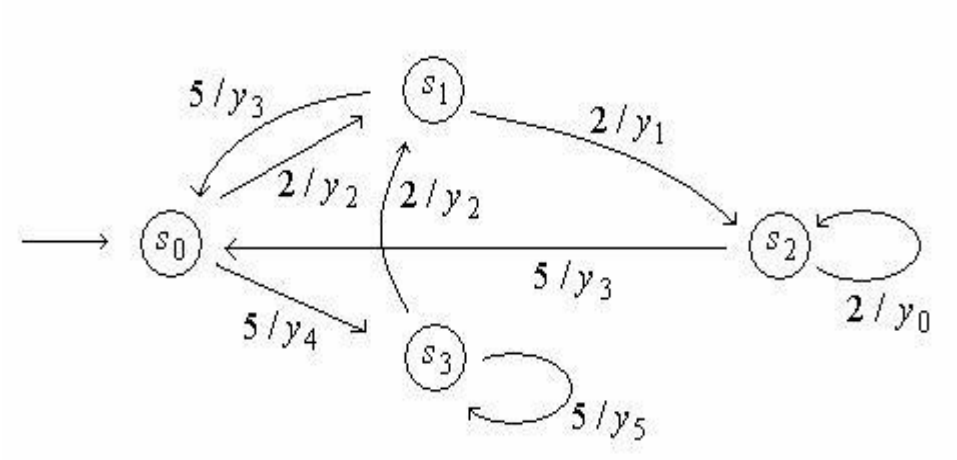


Рисунок 1 – Конечный автомат, описывающий поведение отца.

Автомат имеет четыре состояния {s0 , s1 , s2 , s3} и два входных сигнала Xi – оценки, получаемые сыном в школе: {2,5}. Начиная с начального состояния s0 (оно помечено входной стрелкой), автомат под действием входных сигналов переходит из одного состояния в другое и формирует выходные сигналы – реакции на входы.

Выходы автомата {y 0, y1, … , y5} будем интерпретировать как действия отца следующим образом: y0 – брать ремень; y1 – ругать сына; y2 – успокаивать сына; y3 – надеяться; y4 – радоваться; y5 – ликовать.

Сына, получившего двойку, ожидает дома совершенно разная реакция отца, в зависимости от предыстории его учебы. Отец помнит предысторию учебы сына и строит свое поведение с учетом этой предыстории. Например, после третьей двойки во входной истории 2,2,2 сына встретят ремнем, а в истории 2,2,5,2 – будут успокаивать.

Каждая предыстория определяет текущее состояние автомата (отца), при этом некоторые предыстории эквивалентны (те, что приводят автомат в одно и то же состояние). История 2,2,5 эквивалентна пустой истории. Текущее состояние автомата представляет все то, что автомат знает о прошлом с точки зрения его будущего поведения.

В приведенном примере автомат может рассматриваться как синхронный, если сын приносит по одной оценке, либо как асинхронный, если сын получает сразу несколько оценок.

Кроме того, рассмотренный автомат является, очевидно, автоматом Мили, т. к. реакция зависит как от состояния отца, так и от полученной оценки. Эквивалентом структурного представления автомата может являться, например, матрица смежности орграфа. В отличие от обычной матрицы смежности (см. разд. 3.1.2 в учебном пособии по Дискретной Математики) в качестве элемента δk l записываются входной и выходной сигналы, соответствующие переходу автомата из k-го состояния в l-е.

Если переход sk → sl происходит под воздействием нескольких сигналов, элемент δk l представляет собой множество пар «вход/выход» для этого перехода, соединенных знаком дизъюнкции.

Для рассмотренного примера матрица смежности задана табл. 1:

Таблица 1

	s_0	s_1	s_2	s_3
s_0		$2 / y_2$		$5 / y_4$
s_1	$5 / y_3$		$2 / y_1$	
s_2	$5 / y_5$		$2 / y_0$	
s_3	$5 / y_5$	$2 / y_2$		

Для автомата Мили таблица переходов в общем виде приведена ниже на табл.2:

Таблица 2

	s_0	s_1	…	s_k
x_1	$\varphi(s_0,x_1)$	$\varphi(s_1,x_1)$	…	$\varphi(s_k,x_1)$
x_2	$\varphi(s_0,x_2)$	$\varphi(s_1,x_2)$	…	$\varphi(s_k,x_2)$
…	…	…	…	…
x_k	$\varphi(s_0,x_k)$	$\varphi(s_1,x_k)$	…	$\varphi(s_k,x_k)$

Другим вариантом табличного представления является построение таблиц переходов и выходов, которые несколько отличаются для автоматов Мили и Мура.

Таблица выходов получается из таблицы переходов заменой функций $\varphi(s_i,x_k)$ на $\psi(s_i,x_k)$.

Таблица 3

	s_0	s_1	…	s_k
x_1	$\varphi(s_0,x_1)$	$\varphi(s_1,x_1)$	…	$\varphi(s_k,x_1)$
x_2	$\varphi(s_0,x_2)$	$\varphi(s_1,x_2)$	…	$\varphi(s_k,x_2)$
…	…	…	…	…
x_k	$\varphi(s_0,x_k)$	$\varphi(s_1,x_k)$	…	$\varphi(s_k,x_k)$

Табличное описание автомата Мура можно записать как табл. 3.

Для рассмотренного примера, который представляет собой автомат Мили, таблицы переходов и выходов имеют вид табл. 4 и табл. 5:

Таблица 4

$\varphi(s,x)$	s_0	s_1	s_2	s_3
x_1 (5)	s_3	s_0	s_0	s_3
x_2 (2)	s_1	s_2	s_2	s_1

Таблица 5

$\psi(s,x)$	s_0	s_1	s_2	s_3
x_1 (5)	y_4	y_3	y_3	y_5
x_2 (2)	y_2	y_1	y_0	y_2

Реализация конечных автоматов

Рассмотрим два вида реализации конечного автомата: программную и аппаратную. Программную реализацию конечного автомата можно выполнить на любом языке высокого уровня, причем топология блок-схемы программы будет повторять топологию графа переходов автомата.

Аппаратная реализация требует применения устройств памяти для запоминания текущего состояния автомата.

Обычно на практике, используют двоичные элементы памяти. Функциональный блок автомата реализуется как конечный функциональный преобразователь.

Таким образом, **общий подход к аппаратной реализации конечного автомата включает следующие шаги**:

входные и выходные сигналы и внутренние состояния автомата кодируются двоичными кодами;

по таблицам переходов и выходов составляются кодированные таблицы переходов и выходов – фактически табличное задание отображений **φ и Ψ**;

по кодированным таблицам переходов и выходов формируются аналитические выражения логических функций и проводится их минимизация;

полученные минимальные формы реализуются в заданном элементном базисе;

решаются схемотехнические вопросы синхронизации – привязки моментов выдачи выходного сигнала и изменения состояния внутренней памяти к моментам поступления входных сигналов на вход автомата.

Рассмотрим реализацию автомата из рассмотренного примера.

Входных сигналов два; мы их закодируем так: **2 → 0 , 5 → 1 .** **Выходных сигналов шесть.** Закодируем их: y0 → 000 , y1 → 001, …, y5 →101. **Внутренних состояний четыре**. Закодируем их: s0 → 00 , s1 → 01, s2 → 10 , s3 →11.
Таким образом, имеем: **X** ={0,1}, **Y** ={000,001,010,011,100}, **S** ={00,01,10,11}.

Структурная схема этого автомата после двоичного кодирования имеет вид, показанный на **рис. 2**, где F – функциональный преобразователь без памяти, реализующий отображения **φ и Ψ**, БП – блок памяти.

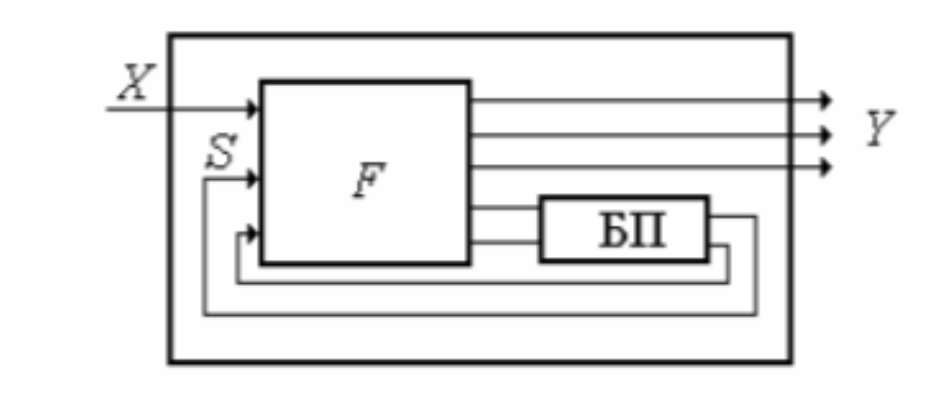


Рисунок 2 - Структурная схема автомата

В кодированной таблице переходов и выходов автомата (**табл. 6**) один двоичный разряд x кодирует входной сигнал Xi , пары двоичных разрядов q1, q2, p1, p2 кодируют соответственно текущее S_i и следующее S_{i+1} + состояния автомата , разряды z1 z2 z3, кодируют выходной сигнал y_i.

Таблица 6

x_i	s_i		s_{i+1}		y_i		
x	q_1	q_2	p_1	p_2	z_1	z_2	z_3
0	0	0	0	1	0	1	0
0	0	1	1	0	0	0	1
0	1	0	1	0	0	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	0	0
1	0	1	0	0	0	1	1
1	1	0	0	0	0	1	1
1	1	1	1	1	1	0	1

После записи логической формулы и минимизации в классе ДНФ (через Карты Карно), получим аналитические выражения для всех двоичных функций. Блоки T1 и T2 – триггеры, которые запоминают двоичный сигнал до прихода следующего. Вход t в триггере – синхронизационный вход, разрешающий переключение триггера.

Сигнал на этом входе должен появляться в момент получения автоматом очередного входного сигнала. Этот же синхросигнал обеспечивает появление на выходе импульсного значения выходного сигнала

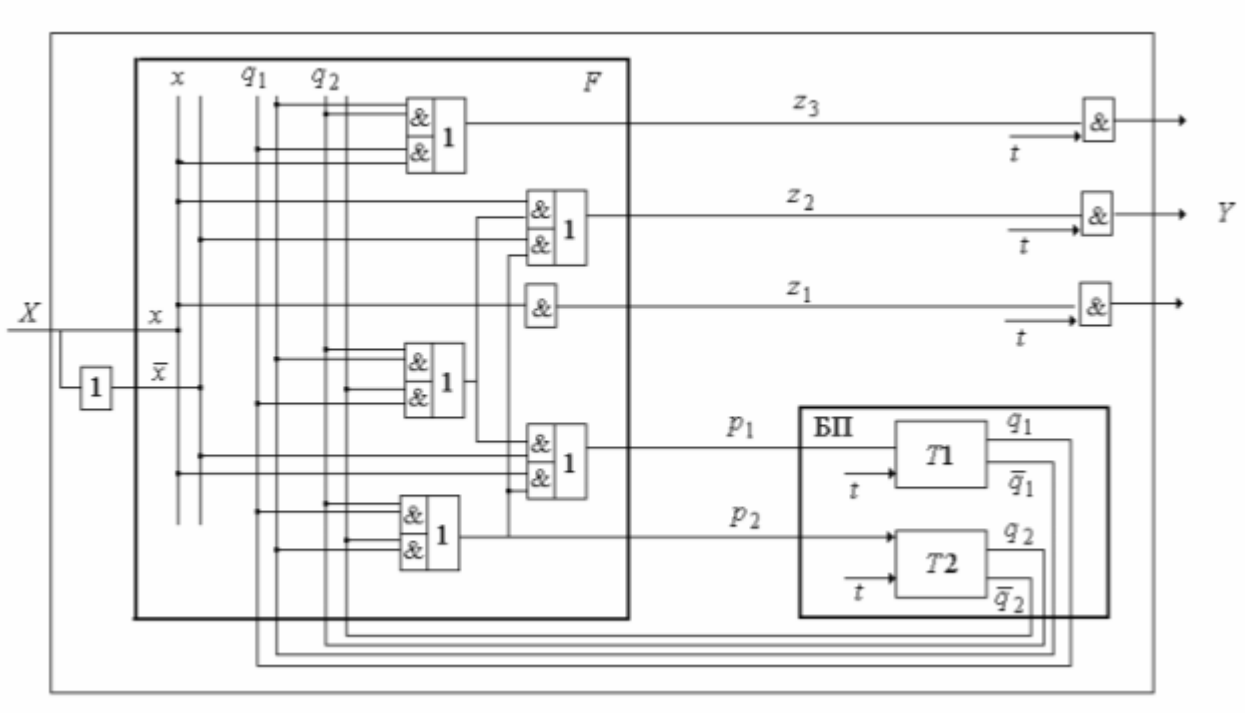


Рисунок 3 - Функциональная схема автомата

Электронные часы.

В качестве технического устройства, построенного на конечного автоматной модели, рассмотрим электронные часы. Электронные часы обычно показывают время, дату, дают возможность установки времени и даты, а также выполняют множество других функций. Управление всеми этими возможностями производится встроенным преобразователем, входами которого являются события нажатия внешних управляющих кнопок.

Рассмотрим простейший вариант, когда показываются и корректируются только дата (год, месяц, день) или время (минуты и секунды). Числа, соответствующие этим данным, хранятся в регистрах памяти и могут быть переданы на регистры отображения путем управления комбинационными схемами. Управление осуществляется двумя кнопками – «a» и «b».

Граф переходов устройства управления, организующего работу всех элементов часов, изображен на **рис. 4**. В начальном состоянии отображается время. Конечный автомат реагирует на нажатие кнопки «a» переходом в состояние «Установка минут», в котором событие нажатия кнопки «b» вызовет увеличение на единицу числа, хранящегося в регистрах, отведенных для минут. Событие нажатия кнопки «b» в состоянии «Установка месяца» вызовет увеличение числа, хранящегося в регистрах, отведенных для месяца и т. д.

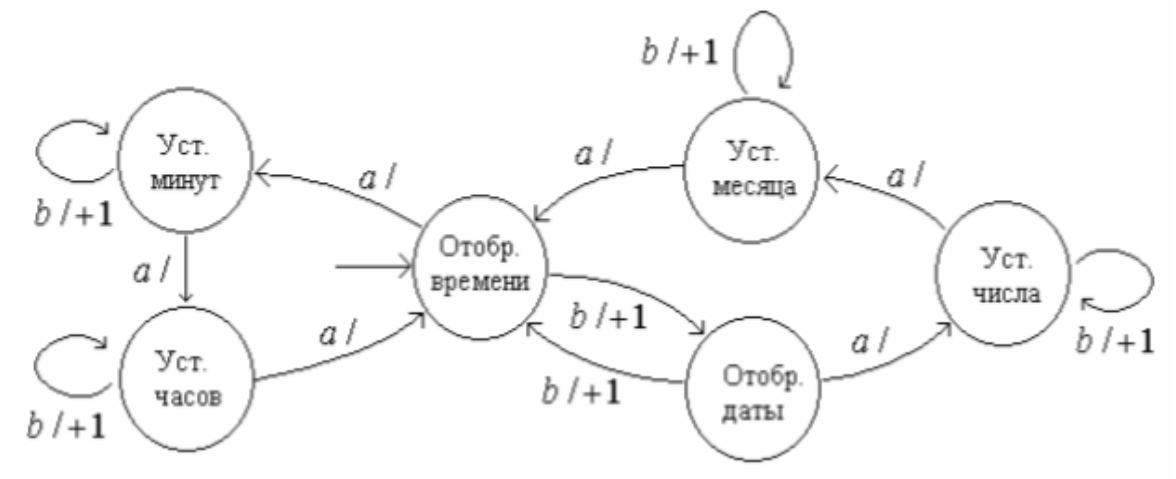


Рисунок 4 - Автомат устройства управления электронными часами

ПРАКТИКУМ: АКТУАТОР КАК КОНЕЧНЫЙ АВТОМАТ

Теоретическая часть из курса «Индустриальный интернет вещей»
N+1 – научно-популярного издания «о прогрессе во всех его проявлениях».
(nplus1.ru).

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ МОДУЛЯ

N+1

В этом модуле вы узнаете:
- что на самом деле называют «датчиком»;
- как устройства собирают, передают и получают информацию;
- что и как мы можем измерять и контролировать на ферме, в городе и на предприятии;
- какие факторы влияют на выбор датчика;

Модуль «Индустриальный интернет вещей» обязателен для изучения. Роль рассказчика может выполнять как тьютор/преподаватель, так и студент. Имеется возможность вынесения содержания интер-отклик курса на одно семинарское занятие. Необходимым условием является применение современных технологий репрезентации курса (смарт-устройства, проектора).

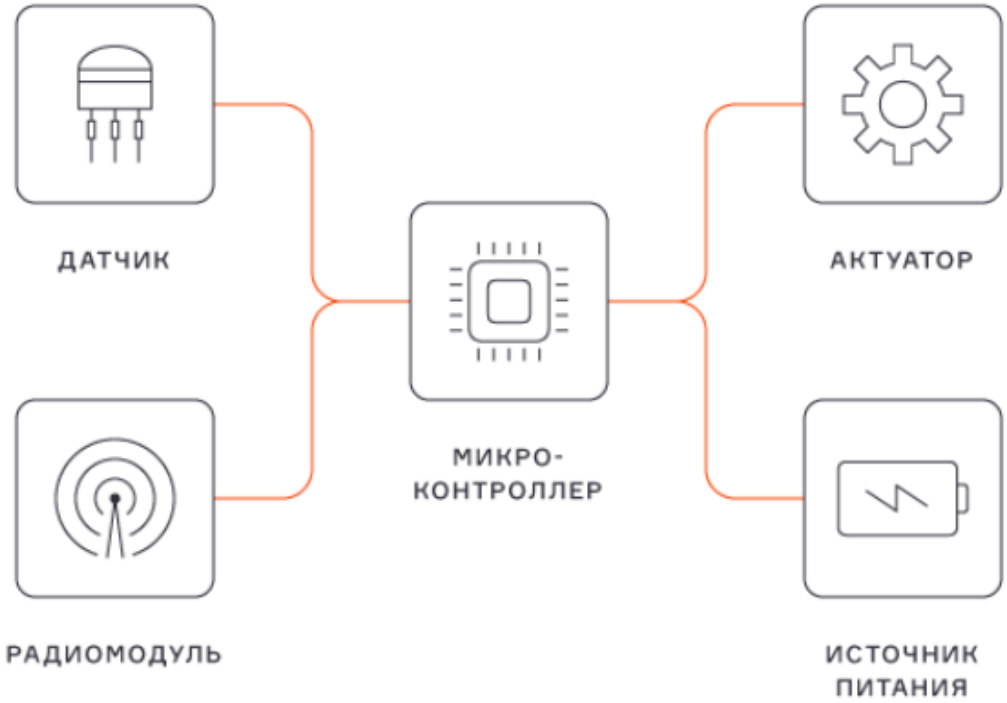


Рисунок 5 – «Готовое устройство»

ТЕЗАУРУС:

Разберем схему подробнее:

1. Датчик. Это чувствительный элемент, который контактирует с окружающей средой: измеряет показания, реагирует на объекты и создает сигнал об этом. Например, в датчике температуры металл или специальная термопленка реагируют на тепло вокруг.

2. Микроконтроллер. Это простой бортовой компьютер в базовом упрощенном понимании, который получает сигнал от датчика и реализует логику работы всего устройства. Например, датчик может отдавать показания раз в секунду, а микроконтроллер будет принимать решение, передавать ли данные человеку.

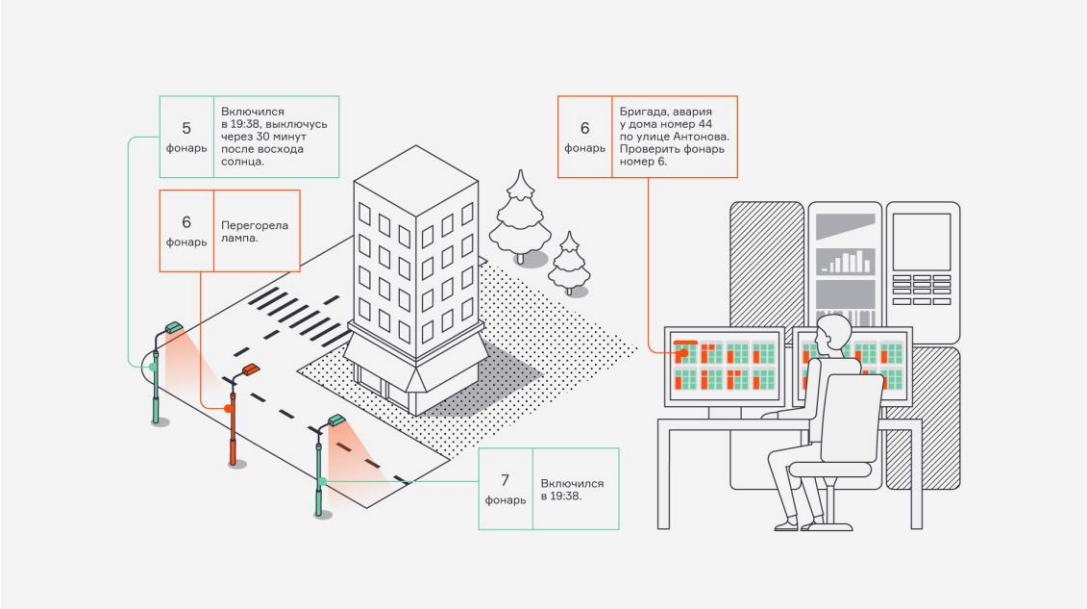
Если мы программируем его реагировать только на температуру выше +30 С, он отправит нам уведомление лишь когда она превысит эту норму.

3. Актуатор (исполнительное устройство) — это реле или транзистор, который мы добавляем к устройству, чтобы оно могло что-то переключать — по сигналу от микроконтроллера или по дистанционной команде, которую примет радиомодуль. Например, если температура выйдет за допустимый предел, актуатор может включить вентиляцию или кондиционер, а вы получите уведомление об этом.

4. Источник питания. Тут все просто — электронике нужно электричество. В зависимости от энергопотребления и задачи мы можем питать устройство от батарейки или от сети по проводу.

Встречаются устройства без датчика — например, умный замок.

Допустим следующее: Когда мы подходим к «умной» двери, на радио-модуль поступает входящий запрос от нашего телефона или пропуска. **Микроконтроллер** сверит номер пропуска или информацию о телефоне с хранящимися в его памяти данными о правах доступа. Если доступ разрешен, он отправит команду на **исполнительное устройство** — и актуатор запустит механизм, отпирающий замок.



После освоения теории синтеза конечных автоматов (в первой главе), а также теоретической составляющей, ориентированной на создание предоставления (образа восприятия) объекта практикума, перейдем к заданию. Но перед этим разберем еще один пример решения:

При помощи абстрактных автоматов можно описать практически что угодно. И как вы понимаете, например, любой актуатор (исполнительное устройство) Можно описать работу цифровой схемы, а можно – синтаксический или лексический анализатор. Попробуем описать известный вам еще с курса схемотехники RS-триггер, ведь – чем не он, не автомат?

Чтобы задать граф нужно словесное описание алгоритма работы триггера.

Читаем: Кодируем входной и выходной алфавиты:
A = {a0, a1}, где a0 – логическая 1 на входе R, a1 – логическая единица на входе S.
B = {b0, b1}, где b0 – логический 0 на выходе Q, b1 – логическая единица на выходе Q.
Строим граф автомата Мили:

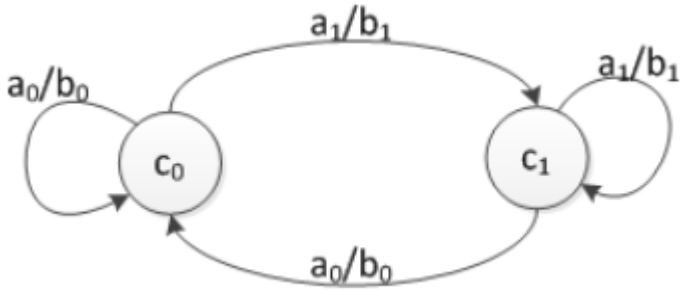


Рисунок 6 – «Принцип работы RS-триггера на графе»

Теперь можно построить таблицу переходов и выходов:

	a_0	a_1
C_0	C_0/b_0	C_1/b_1
C_1	C_0/b_0	C_1/b_1

R	S	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	-
1	1	1	-

Q_{t+1}

R	S	Q_t
0	0	0
0	1	1
1	0	0
1	1	-

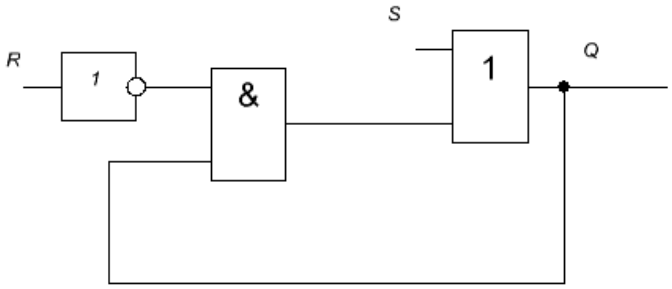
Рисунок 7 – «Таблица переходов и состояний RS-триггера»

Нанесём полученную функцию на карту Вейча (Карно) и минимизируем:

Q_t \ S				
R	0	1	1	1
	0	0	-	-

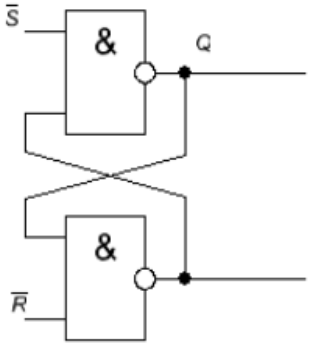
$$Q_{t+1} = S \vee Q_t \overline{R}$$

Строим по функции схему в булевом базисе:



И в базисе И-НЕ. Ведь, как вы знаете, чаще всего все на практике строится на базисах ИЛИ-НЕ или И-НЕ (ДНФ/КНФ).

$$Q_{t+1} = S \vee Q_t \overline{R} = \overline{\overline{S \vee Q_t \overline{R}}} = \overline{\overline{S} \,\&\, \overline{Q \,\&\, \overline{R}}} = \overline{S} \mid (Q_t \mid \overline{R})$$



Теперь если приложить немного старания, то можно самостоятельно синтезировать простую новогоднюю гирлянду.

ЛАБОРАТОРНАЯ РАБОТА №1: «Синтез КА IoT-актуатора»

Цель: Разработать конечный автомат предлагаемой триггерной системы; представить модель применения построенного устройства в рамках IoT-платформы (в рамках актуатора/датчика). Построить комбинационную логическую схему устройства в булевом базисе (или/и на базисе И-НЕ/ИЛИ-НЕ).

Варианты заданий

Рабочая группа I Синхронный RS-триггер
Рабочая группа II Однотактный D-триггер
Рабочая группа III Синхронный T-триггер
Рабочая группа IV DV-триггер
Рабочая группа V Асинхронный T-триггер

Требования к отчету:

1. Условное графическое обозначение объекта проектирования.
2. Предназначение выбранного типа триггера (по справочнику).
3. Граф-схема КА.
4. Таблица переходов и выходов.
5. Обозначение и расшифровка входов и выходов триггера.
6. Кодировка входного и выходного «алфавита».
7. Минимизация полученной через метод карт Вейча или Карно, или через метод Куайна-Маккласки (используя программные средства минимизации или вручную) (по ДНФ или КНФ)
8. Построение логической схемы в булевом базисе.
9. Описание умозаключения по поводу целевого предназначения конечного автомата.

§ ОСНОВЫ СЕТЕВОГО ПЛАНИРОВАНИЯ И УПРАВЛЕНИЯ

Мы научились синтезировать простейшие исполнительные устройства и при наличии. Безусловно, этого недостаточно. Ведь как понять, что нам нужно, (нужно заказчику, нам, от него к нам, предлагаемой к разработке системе, объекту) да и как организовать процесс проектирования даже таких простых устройств, если они выходят за рамки шаблонов? Ведь, рассматриваемых в практикуме вопросов гораздо больше, чем кажется. Это синергия больших теоретических познаний из разных направлений научной мысли, причем часто, очевидность взятого метода, идеи, далеко не всегда явна.

В этой главе кратко рассмотрим основные понятия сетевого планирования и управления, которые ответят на некоторые вопросы, однако, далеко не на все. Впрочем, если вам очевидно, что почти любую осознанную деятельность в практико-ориентированном ключе можно назвать проектом, то, в таком случае, приступим к изучению.

Сетевые модели планирования и управления проектами.

Проектом называют совокупность работ, направленных на достижение некоторой цели. Работы проекта, как правило, частично упорядочены. Выполнение работы не может быть начато до завершения всех предшествующих ей работ. Продолжительность выполнения каждой работы известна. Предполагается, что начатая работа продолжается без перерыва до ее завершения.

Проект считается выполненным, если выполнены все его работы. Сетевая модель – это графическое представление проекта. Она позволяет найти минимальные сроки завершения проекта и отдельных работ, а также определить множество *критических работ*, увеличение продолжительности выполнения любой из которых приводит к увеличению времени выполнения всего проекта.

Сетевая модель проекта представляет собой графическое описание плана работ, показывающее взаимосвязь между всеми работами, выполнение которых необходимо для завершения проекта. В терминах теории графов сетевая модель – это ориентированный граф без контуров и петель с неотрицательными весами вершин или дуг.

При анализе проектов используются два типа сетевых моделей, которые условно можно назвать:

1) «работы-вершины»;

2) «работы-дуги».

В сетевой модели первого типа вершины являются образами работ, а дуги служат для отображения отношения предшествования между работами.

В модели второго типа, наряду с работами-дугами, рассматриваются также новые понятия – события, которым соответствуют вершины сети.

Событие отражает результат – завершение одних работ и возможность начала других.

Направление дуги сети задает отношение предшествования работ проекта.

Указанные выше модели являются эквивалентными в том смысле, что для любого проекта можно построить как одну, так и другую модель.

В этой главе рассматриваются сетевые модели типа «работы-дуги». В таких моделях для задания соотношения предшествования работ-дуг часто приходится вводить **фиктивные** дуги нулевой длины (продолжительность выполнения соответствующих фиктивных работ равна нулю).

Нефиктивные дуги будем называть также **фактическими.**

Сети с одним и тем же множеством фактических дуг назовем **эквивалентными**, если они задают одно и то же отношение предшествования дуг-работ.

Вершину, не имеющую входящих в нее дуг, будем называть *входом*, а вершину, из которой не выходит ни одной дуги, – *выходом* сетевой модели. Без ограничения общности можно считать, что сетевая модель имеет один вход и один выход. Если исходная сеть не обладает этим свойством, то можно ввести фиктивный вход (выход) и связать с ним все входы (выходы) фиктивными дугами. Полученная сеть, очевидно, *эквивалентна* исходной сети.

Пример: Привести сетевую модель, изображенную на рис.8 к эквивалентной сети с одним входом и одним выходом.

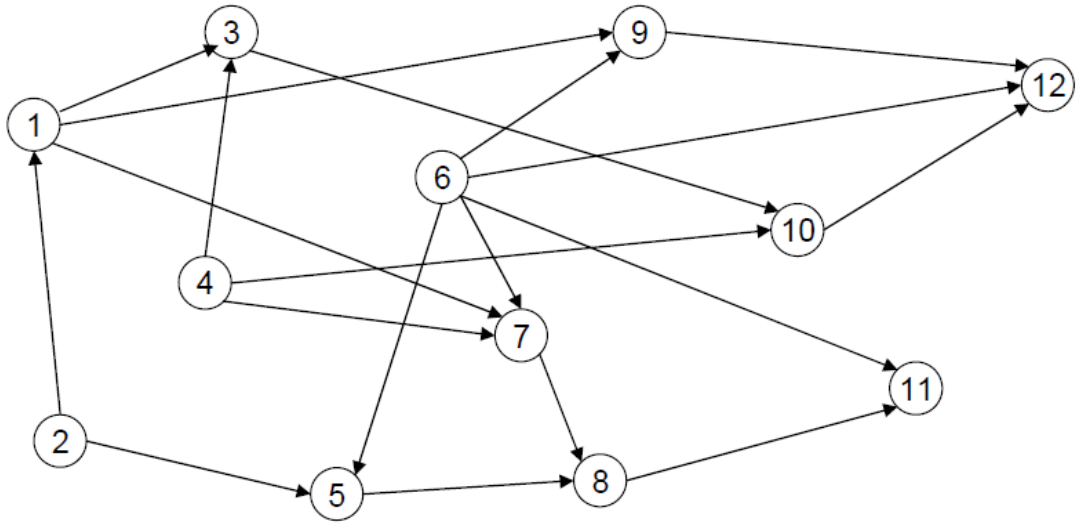


Рисунок 8 – «Граф-схема сетевой модели»

Решение. Входами сети являются вершины 2, 4 и 6. Введем дополнительную вершину 13 и свяжем ее фиктивными дугами (13, 2), (13, 4) и (13, 6) со всеми входами.

Вершины 11 и 12 являются выходами сетевой модели. Добавим вершину 14 и соединим с ней фиктивными дугами (11, 14), и (12, 14) все выходы сетевой модели (рис.9):

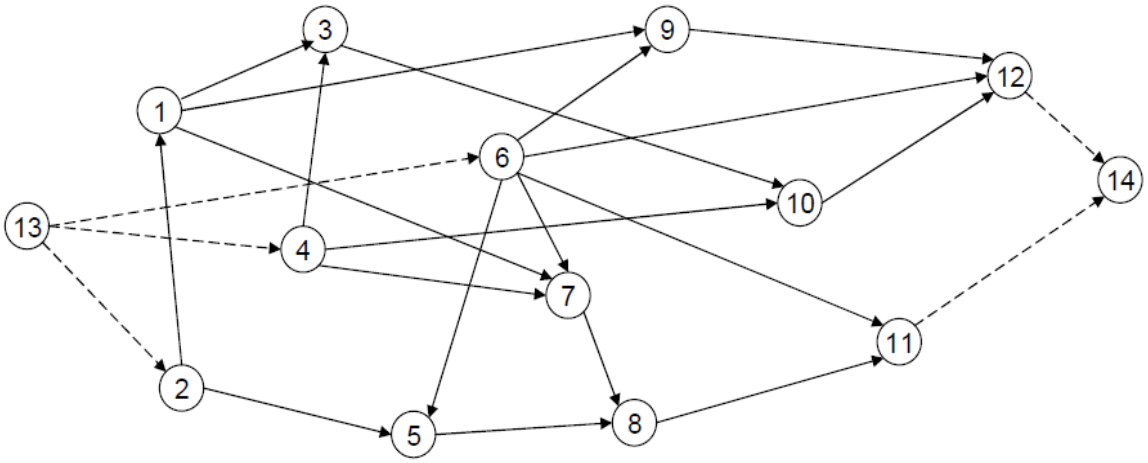


Рисунок 9 – «Граф-схема сетевой модели после соединения»

На рис. 9 представлена новая сеть с одним источником 13 и одним стоком 14. Фиктивные дуги изображены пунктирными линиями.

Упрощение сетевой модели

В ряде случаев исходную сеть можно упростить, исключив часть вершин и фиктивных дуг. Введем следующие обозначения:

$S = (X, U)$ – сеть с множеством вершин X и множеством дуг U;

$i(j)$ – начальная вершина дуги j;

$k(j)$ – конечная вершина дуги j;

$\overline{U}$ – множество фактических дуг сети;

U_i – множество дуг из $\overline{U}$, принадлежащих объединению всех путей из входа сети в вершину i;

U^i – множество дуг из $\overline{U}$, принадлежащих объединению всех путей из вершины i в выход сети.

Зафиксируем пару вершин i и k, для которых

1) не существует дуги $j \in \overline{U}$, инцидентной одновременно i и k;

2) не существует дуг $p, q \in \overline{U}$, для которых

– $i(p) = i(q)$, где $k(p) = I$ и $k(q) = k$ или

– $k(p) = k(q)$, где $i(p) = i$ и $i(q) = k$.

Результатом операции ***склеивания*** вершин *i* и *k* будет сеть *S'*, полученная из исходной сети *S* удалением вершины *k* и замыканием **инцидентных ей дуг** на вершину *i*. Когда в результате склейки между какой-то парой вершин *i* и *l* в *S ′* будет более одной дуги.

– если среди них есть фактическая дуга, то удаляются все параллельные ей фиктивные дуги;

– если *i* и *l* в *S ′* связаны только фиктивными дугами, то все они кроме одной удаляются.

Таким образом, если в исходной сети *S* дуга имела вершину *k* начальной (конечной), то в полученной сети *S ′* эта дуга имеет начальной (конечной) вершину *i*.

Очевидно, в *S'* существует путь из одной вершины в другую тогда и только тогда, когда такой путь есть в *S*.

Эквивалентность сетей *S* и *S ′* проверяется с помощью следующего необходимого и достаточного условия. Пусть вершины *i* и *k* удовлетворяют условиям 1–2, описанным выше. Сеть *S ′*, полученная из *S* склеиванием *i* и *k* и удалением *k*, эквивалентна *S*, тогда и только тогда, когда выполняется хотя бы одно из условий: $U_i = U_k$ или $U^i = U^k$.

Пример. Упростить сетевую модель, представленную на **рис. 10.**

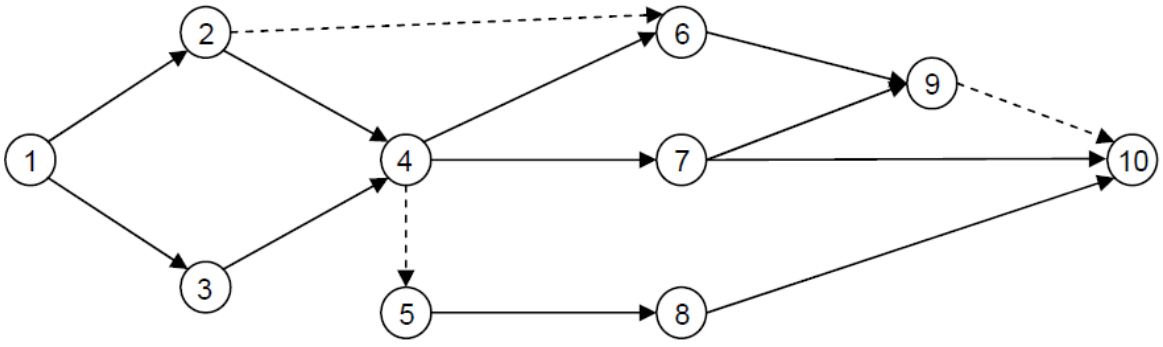


Рисунок 10 – «Граф-схема выбранной сетевой модели»

Решение. Очевидно, для выполнения приведенных выше условий вершины – кандидаты на склеивание должны быть соединены *фиктивной* дугой. Рассмотрим такие пары вершин в данной сети. Вершины 4 и 5 удовлетворяют свойствам 1–2. Очевидно, $U_4 = \{(1, 2),(2, 4),(1, 3),(3, 4)\} = U_5$. Следовательно, вершины 4 и 5 могут быть склеены.

Для вершин 9 и 10 свойство 2 не выполняется, так как вершина 7 связана с обеими вершинами 9 и 10 фактическими дугами. Следовательно, 9 и 10 нельзя склеить.

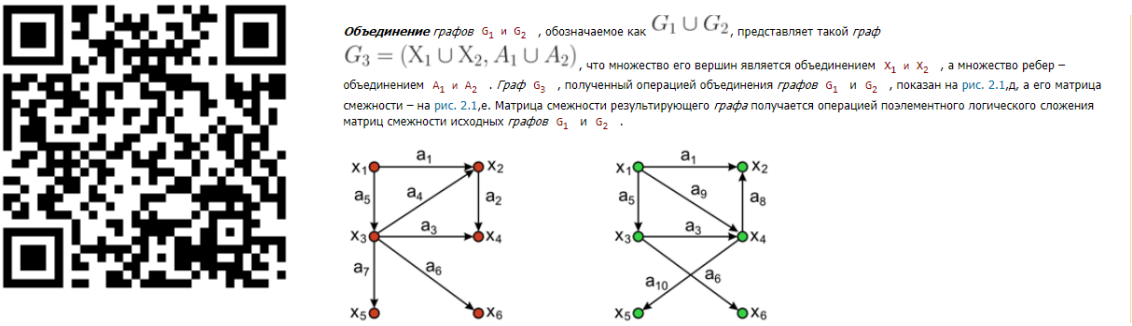
Для вершин 2 и 6 свойства 1–2 выполняются, но $U_2 = \{(1,2)\} \neq U_6 = \{(1, 2), (2, 4), (1, 3), (3, 4), (4, 6)\}$ и $U^2 = \{(2, 4), (4, 6), (4, 7), (5, 8), (6, 9), (7, 9), (7, 10), (8, 10)\} \neq U^6 = \{(6, 9)\}$. Следовательно, вершины 2 и 6 не могут быть склеены с сохранением эквивалентности сетей.

Теоретическая справка

Пусть **V** – некоторое непустое множество ($\mathbf{V} \neq \varnothing$).
$\mathbf{V}^{(2)}$ – множество всех его двухэлементных подмножеств, **{u,v}** – неупорядоченная пара элементов множества **V**. $\mathbf{V}^{(2)} = \{\{u,v\} \mid u,v \in V\}$.

Неориентированный граф G – пара множеств **(V,E)**, $\mathbf{E} \subseteq \mathbf{V}^{(2)}$, где
V – множество **вершин** графа **G**,
E – множество **рёбер** графа **G**.

Если **|V|=p**, а **|E|=q**, то обозначают граф **G**, как **(p,q)**-граф или **p**-граф.
Смежные вершины графа G – вершины, соединенные ребром.
Смежные ребра графа G – ребра, имеющие общую вершину.
Инцидентные ребро и вершина – вершина является одним из концов ребра.
Конечный граф – множество вершин графа конечно.



Аннотация: По ссылке (QR-код) НОУ ИНТУИТ | Операции над графами.

Приводятся основные операции над графами такие как объединение, пересечение, кольцевая сумма, удаление вершины, удаление ребра, замыкание и стягивание. Эти операции рассматриваются для представления графов матрицами смежности. Цель лекции: Дать представление об операциях над графами и возможных способах их представления в матричных структурах.

§ ОБЗОР ЦИКЛА ПРОЕКТИРОВАНИЯ ККС

Харьковский национальный университет радиоэлектроники, Кафедра Медиасистем и технологии,
Бабич Анна Витальевна

В настоящее время единого стандартизированного жизненного цикла для корпоративных компьютерных сетей (ККС) не существует. Более того, проектирование и развертывание малосегментных КС или КС, качество функционирования которых не является предметом коммерческой и/или иной выгоды, в формализации процесса проектирования в большинстве случаев и не нуждается. Тем не менее, при проектировании КС в достаточно больших масштабах, к которым относятся, прежде всего, корпоративные КС (ККС) и КС провайдеров услуг Интернет, и вообще, при наличии прямой зависимости прибыли предприятия (организации) от качества услуг, предоставляемых компьютерной сетью как системой, появляется необходимость в обеспечении гарантированного уровня качества работы КС. И это значит, что моменты неуправляемости, непрогнозируемости и непрозрачности, которые присущи проектированию в стиле «свободного плавания», из процесса проектирования должны быть исключены.

Общая схема этапов проектирования ККС

Исходя из опыта крупных сетевых интеграторов, как отечественных, так и зарубежных, разрабатывающих свои методологии проектирования ККС, можно выделить следующие типовые этапы выполнения сетевых проектов (**рис.11**).

Далее, указанные этапы будут рассмотрены подробнее. Следует заметить, что в реальности этапы, начиная от анализа требований и заканчивая разработкой физической модели, как правило, выполняются параллельно, а не последовательно, как это показано на **рис. 11**. И это естественно, так как, к примеру, в процессе формулировки требований к проекту сети, одновременно с этим прорисовывается каркас технической модели ККС; без построения функциональной модели невозможно с адекватной полнотой и достаточностью сформулировать цели и задачи проектируемой ККС и т.д.

Рисунок 11 - Этапы проектирования ККС

Обзор цикла проектирования ККС

1.1 Анализ требований

Под анализом требований понимается определение проблем и деловых целей предприятия, а также формулировка задач и целей проектирования в соответствии с ними. Анализ требований к сети поможет оценить деловую значимость информационно-технологических решений, определить главные цели и выбрать приоритеты для отдельных частей компьютерной системы, которую необходимо улучшить или расширить. Четкое определение требований к функциям сети поможет избежать реализации ненужных свойств сети, что сэкономит средства предприятия.

Иначе говоря, прежде чем проектировать сеть, нужно понять, какие выгоды должно получить предприятие от модернизации ККС (например, сокращение производственного цикла, более оперативный прием заказов или повышение производительности труда за счет более эффективного взаимодействия сотрудников), какие задачи будет решать сеть, какими будут основные потоки трафика, как физически будут расположены пользователи и ресурсы, нужно ли задание приоритетов видов трафика, как будут решаться вопросы защиты информации внутри сети, как сеть будет подключена к Интернет, как решить задачи биллинга, управления правами доступа пользователей.

Кроме того, на этапе анализа требований необходимо изучение состояния зданий и сооружений в месте развертывания сети, анализ существующей инфраструктуры. Эта информация жизненно необходима как для постановки задачи проектирования, так и для самого проектирования.

1.2 Разработка функциональной модели

Функциональная (или бизнес-модель) производства отображает последовательность работ и технологических процессов предприятия, а также каждого из подразделений в отдельности, определяет набор сетевых задач, выполняемых в каждом из подразделений, на основании которых формулируются требования к проектируемой сети, предъявляемые к ней спецификой бизнес-процессов каждого из подразделений в отдельности и предприятия в целом.

Одновременно с формулировкой требований к корпоративной сети, нужно получить общее представление о том, что происходит в каждом отделе. Именно это и описывает функциональная модель. В ней обычно не упоминается компьютерная система, она концентрируется на деловой практике и последовательности работ. Сначала строится модель, в которой отражается последовательность работ всего предприятия, а затем - модель для последовательности работ в каждом отделе. Здесь же необходимо указать, как выполняются работы, кто выполняет эти работы и каковы взаимосвязи между рабочими группами и отделами.

Для разработки функциональной модели ККС необходимо собрать бригаду, состоящую из руководителей отделов, ведущих специалистов и сотруднков отдела автоматизации, и выполнить следующее:

- опросить руководителей отделов и конечных пользователей корпоративной сети, чтобы определить их функции и выяснить, как их компьютерные системы помогают им в работе;
- выяснить, как работа переходит из одного отдела в другой, и каким образом информация и задачи передаются от одного сотрудника к другому;
- узнать, в чем заключаются зависимости - кто утверждает какой-либо этап работы и в какой последовательности должны завершаться этапы;
- понять, какие узкие места имеются у системы - слишком большое время ответа или же неэффективная обработка данных.

1.3 Разработка технической модели

После разработки функциональной модели и определения того, какие процедуры требуют изменения или улучшения, необходимо построить техническую модель ККС. Техническая модель описывает в достаточно общих терминах, какое компьютерное оборудование нужно использовать, чтобы достичь целей, определенных ранее.

Чтобы построить техническую модель, нужно проанализировать существующее оборудование, определить системные требования, оценить сегодняшнее и завтрашнее состояния техники.

Анализ существующего оборудования сводится к инвентаризации аппаратной и прикладной базы, эксплуатируемой в корпоративной сети, в результате которой будет принято решение об использовании части оборудования в новом проекте ККС. Данное решение должно опираться на соответствие оборудования требованиям, предъявляемым к проектируемой сети на этапе создания функциональной модели, а также на этапе определения системных требований к технической модели. Процесс инвентаризации можно и нужно автоматизировать. Существуют программы, которые могут автоматически исследовать состав аппаратного и программного обеспечения уже работающих в сети компьютеров.

В общем случае такие программы могут выяснить тип процессора, имеющуюся память, тип диска и свободное пространство на нем, имеющиеся дополнительные контроллеры - такие как сетевые адаптеры IoT и т.п. Для программного обеспечения можно узнать наименование и версию приложений, версии операционной системы, установленные сетевые драйверы.

Для выяснения системных требований необходимо ответить на следующие вопросы:

- Что нужно соединять?

Требуется ли сотрудникам какого-либо подразделения общаться с небольшим (большим) количеством людей в пределах небольшой территории или же им нужно общаться с небольшим (большим) количеством людей в пределах географически обширной области? Объем и распределение графика поможет определить требуемую мощность компьютеров, а также типы и скорости коммуникационного оборудования и сервисов.

- Что из существующего аппаратного и программного обеспечения будет использоваться в новой системе?

Какие системы нужно оставить в разрабатываемой корпоративной сети? Нужно ли эти системы соединять в сеть? Будут ли существующие системы нормально работать в новой сети? Существуют ли какие-либо стандарты предприятия, существуют ли преобладающие приложения? Какое оборудование и приложения нужно добавить, чтобы достигнуть поставленных производственных целей?

- Какие объемы информации будут передаваться по сети?

Объем передаваемой информации определяет требуемую пропускную способность сети. Определите это подсчетом количества пользователей сети, среднего количества выполняемых транзакций в день каждым из пользователей и среднего объема транзакции.

Такой подсчет поможет определить технологию доступа к среде передачи данных и требования к глобальным сервисам.

- Какое время реакции сети является приемлемым?

Будут ли пользователи ждать одну секунду, полсекунды или две секунды? Такие измерения помогут определить требования к скорости оборудования, приложений и коммуникационных связей. В течение какого времени сеть существенно необходима для работы предприятия?

Нужна ли сеть 24 часа в день и 7 дней в неделю или же только в течение 8 часов в день и 5 дней в неделю? Нужно ли увеличить сегодняшние параметры использования сети?

- Какие требования предъявляются к среднему времени устранения неисправностей? Как отражаются операции по обслуживанию и ремонту сети на эффективности ведения дел предприятием? Потеряет ли предприятие 5 миллионов долларов или же 100 тысяч долларов, если сеть будет неисправна в течение одного часа? Каков будет ущерб от простоя сети с течение двух часов?

- Каков планируемый рост системы? Каков текущий коэффициент использования сети и как он может измениться в течение ближайших 6 месяцев, одного года, двух лет? Даже если вы тщательно спланировали сеть, но не учли возможности ее роста и развития, то системные требования придется изменить и увеличить. Рост сети нужно планировать заранее, а не просто реагировать на фактический рост ее нагрузки.

1.4 Разработка физической модели

После того, как для сети выбрана техническая модель, необходимо оценить, насколько она удовлетворяет производственным требованиям. Нужно вернуться к функциональной модели и сопоставить ее требования с техническими решениями. Например, если на предприятии сотрудники часто перемещаются из отдела в отдел, то требованием функциональной модели является высокая мобильность. Техническая модель должна в таком случае обеспечивать быстрое присоединение и отсоединение рабочей станции.

После оценки соответствия технической модели производственным требованиям, необходимо построить физическую модель. Физическая модель конкретизирует специфику технической модели и является очень подробным описанием сети, с указанием технических характеристик пассивного, активного и оконечного оборудования, в то время как техническая модель использует для ее описания более общие термины.

На стадии физического моделирования проектировщик должен точно описать, какие компоненты нужны, в каком количестве, где они будут расположены, и как эти компоненты будут соединяться друг с другом в корпоративную сеть.

1.5 Установка и наладка системы

К данному этапу у вас, в результате работы над этапами предыдущими, уже должно иметься сформированное техническое задание (ТЗ), которое с этого момента будет претворяться в жизнь.

Схематически набор и последовательность работ на этапе установки и наладки системы можно представить так, как на **рис. 12:**

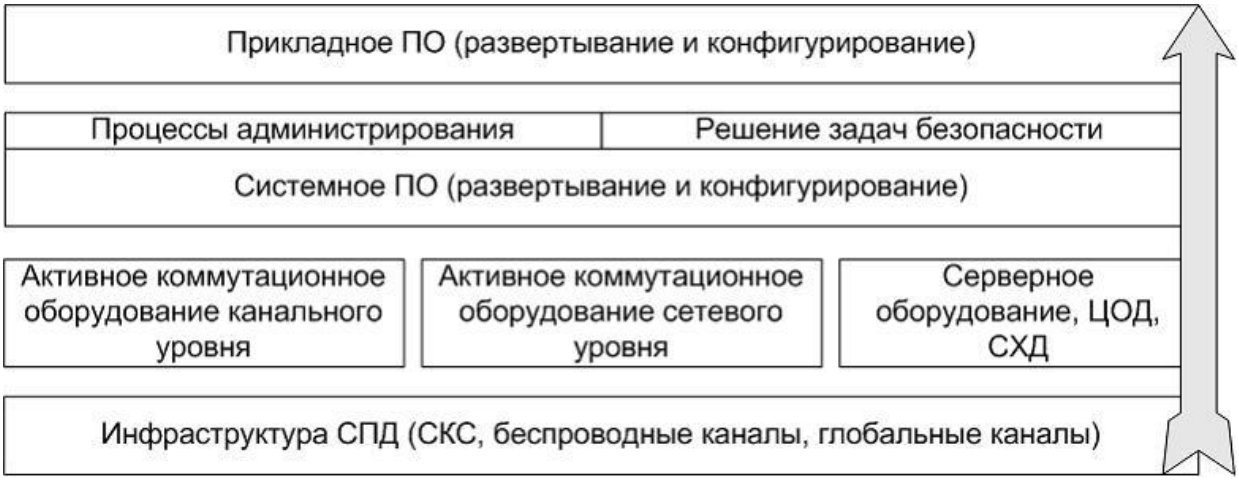


Рисунок 12 - Порядок работ на этапе установки и наладки

Инфраструктура среды передачи данных (СПД) - фундамент ККС, от качественного исполнения которого будет зависеть качество работы сети в целом.

Именно здесь находятся самые «грязные» и трудоемкие работы: выполняется монтаж структурированной кабельной системы (СКС), что само по себе является сложным многоэтапным процессом, развертываются беспроводные линии связи, реализуются внешние подключения и подключения к глобальным каналам. При развертывании СКС «с нуля», необходимо иметь лицензию на выполнение соответствующих монтажных работ.

Развертывание и конфигурирование активного оборудования включает в себя, помимо базовой настройки, еще и расширенную, которая может состоять из:

- контроля широковещательного трафика,
- управления пропускной способностью канала связи,
- приоритезации трафика и управления очередями,
- создания списков доступа и развертывания политики безопасности на уровне коммутирующего оборудования,
- развертывания VLAN,
- агрегирования каналов для отдельных участков магистрали,
- и пр.

Все функции, в которых возникнет необходимость в соответствии с требованиями, предъявляемыми к ККС в ТЗ, должны быть учтены при выборе и приобретении оборудования как канального уровня, так и сетевого.

Если в ТЗ обозначено создание центра обработки данных (ЦОД), то работа над ЦОД начинается параллельно с монтажом коммуникационного оборудования СКС. По сути, создание ЦОД - такой же сложный многоэтапный процесс, как и развертывание СКС, и должен иметь свою сопроводительную документацию (ТЗ), так же, как и процесс создания СКС.

Развертывание и конфигурирование системного и прикладного ПО включает в себя множество задач, решаемых администраторами ККС и поддерживаемых ими на протяжении всего срока эксплуатации проектируемой ККС. К таким задачам относятся:

- создание системы и процедур автоматизации администрирования и управления ККС,
- развертывание системы безопасности на уровне серверов и конечных станций,
- создание системы восстановления и обеспечения отказоусточивости средствами ОС:
- развертывание и настройка прикладного ПО в соответствии с требованиями.

1.6 Тестирование системы

Оценка эффективности работы сети (или тестирование сети) предполагает использование технических, организационных и программных решений и полностью согласуется со схемой администрирования системы. Оценка эффективности сети осуществляется в реальном режиме времени и может быть реализована с помощью встроенных инструментальных средств операционной системы и с помощью специальных программ типа анализаторов сети.

1.7 Сопровождение и эксплуатация системы

Этот этап не имеет четко определенных временных границ, а представляет собой непрерывный процесс. Для каждого из упомянутых этапов и даже для отдельных более мелких задач может быть разработано техническое задание.

§ РАЗРАБОТКА СТРУКТУРЫ IoT-ПЛАТФОРМЫ

IoT платформа – программное обеспечение, предназначенное для подключения интернет вещей (датчиков, контроллеров и других устройств) к облаку и удаленного доступа к ним. Она представляет собой промежуточный уровень между аппаратным уровнем (уровнем сенсоров) и прикладным.

С момента появления термина «Интернет вещей» сети, состоящие из большого количества устройств, общающихся между собой, стремительно развиваются. Вследствие этого, IoT (Internet of Things) становится одной из основных технологий в современном обществе. С точки зрения технологических и технических аспектов развития IoT в настоящее время существует четкое разделение между аппаратными и программными платформами для подключения устройств, причем большинство поставщиков предлагают именно программные IoT платформы.

Платформы IoT обеспечивают бесшовную интеграцию различных аппаратных средств, используя протоколы связи, применяя различные типы топологии (прямое подключение или шлюз) и используя SDK при необходимости и т.д.

Используя интерфейсы интеграции с северной границей, предоставляемые платформой, вы также можете передавать собранные данные IoT в определенные системы анализа и хранения данных, а также передавать данные на подключенные устройства (конфигурация, уведомления) или между ними (элементы управления, события), используя различные виды пользовательских приложений.

Самыми популярными программными IoT платформами являются: Microsoft Azure IoT**, Amazon Web Services (AWS**) IoT, Google Cloud, ThingWorx IoT, IBM Watson, Artik от Samsung Electronics, Cisco IoT Cloud Connect, Salesforce IoT Cloud и многие другие.

Критериями отличия программных IoT платформ друг от друга являются:

- масштабируемость – количество конечных устройств, которые могут подключаться к платформе, включая эффективную балансировку нагрузки серверов;
- простота использования – гибкость API интеграции и простота управления исходным кодом;
- варианты развертывания – публичное или частное облако;
- безопасность – защита данных путем шифрования, контроля доступа пользователей и т.д.
- база данных – вариант хранения данных, получаемых с устройств, наличие гибридных облачных баз данных и т.д.

Среди протоколов, используемых платформами IoT, наиболее популярными являются MQTT, CoAP, HTTP/HTTPS, AMQP, XMPP, DDS.

Большинство современных программных плат IoT поддерживают аналитику в реальном времени - агрегирование потоков, фильтрация и др. (например, Storm, Samza), пакетную – операции с накопленным набором данных (например, Hadoop, Spark) и интерактивную аналитику данных - многократный исследовательский анализ как потоковых, так и пакетных данных (Spark MLLIB и т. д). Также существует прогностический метод аналитики, основанный на различных способах статистического и машинного обучения.

Начать разбираться с IoT (Internet of Things) платформами останавливает отсутствие IoT устройства, которое было бы совместимо по протоколам и способам доступа.

(из статьи «Что нам стоит IoT построить? Свой IoT на Amazon за один день» Алексей Сушков, портал Хабрахабр):

Но когда я понял, что в качестве устройства можно использовать обыкновенный смартфон, то реализация работающей цепочки заняла один день.

Возьмем смартфон с ОС Android/IOS актуальной ревизии, который будет эмулировать IoT устройство с датчиками температуры, влажности и давления и отсылать показания на Amazon IoT платформу. На платформе заведем правило, которое при поступлении данных от нашего устройства будет вызывать сервис нотификаций, который в свою очередь будет отсылать e-mail с полученными данными.

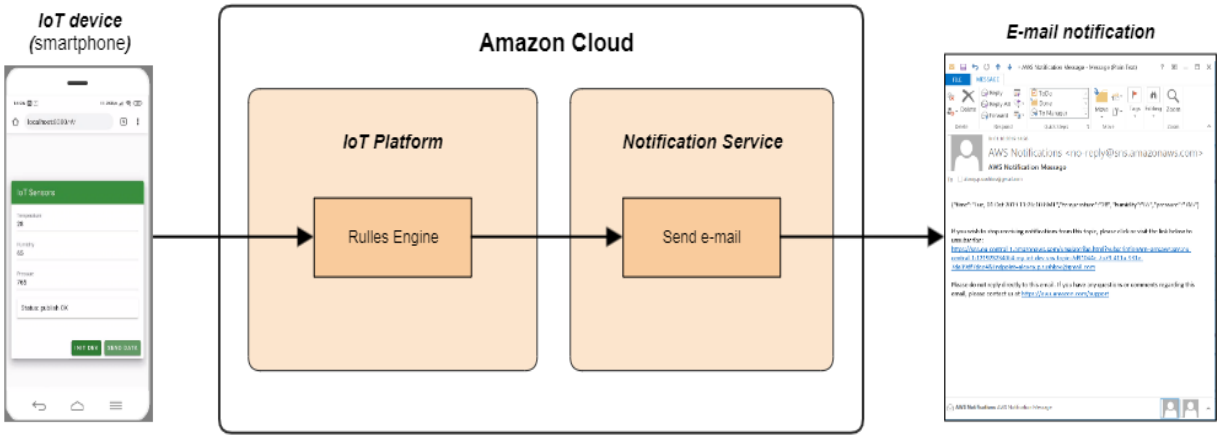


Рисунок 13 – «M2M – IoT — Internet of Everything (IoE)»

Почему именно IoT платформа от Amazon? И зачем вообще нужно понимать, как работают IoT платформы?В мире становится всё больше IoT устройств, об этом говорят, как аналитические агентства, так и мировая статистика.

Мы и сами прекрасно видим (и вторим об этом), что все больше систем подключаются к интернету и управляются автоматически или людьми: умные дома, автомобили, носимые устройства. И сейчас уже говорят не просто об IoT, а о IoE (Internet of Everything), т.к. устройства которые подключаются к платформам используются не только в промышленных системах, но и людьми.

Поэтому нужно самим разбираться в принципах работы, хотя бы для того, чтобы понимать, как можно эффективно использовать свои устройства или какие есть ограничения и нюансы с безопасностью.

Почему Amazon?

Amazon создает сервисы с учетом мировых трендов и в результате получаются “универсальные” системы, основные принципы которых используют все производители. У облачной платформы есть еще больший плюс – это возможность самостоятельно развернуть систему за пару часов, не привлекая корпоративную IT службу и безопасность)

Почему смартфон, а не какой-нибудь IoT Starter Kit?

При внимательном рассмотрении смартфон хорошо эмулирует IoT устройство:

- В нём есть Linux-ядро, на котором можно запускать приложения;
- Есть мобильная связь с Интернет;
- С помощью программных средств можно эмулировать показания датчиков.

Т.е. работа с настоящим IoT устройством ничем не будет отличаться от работы со смартфоном, кроме использования специфичного **SDK** для получения показаний датчиков. Всё остальные коммуникации будут аналогичны.

SDK (от англ. software development kit) — **набор средств разработки**, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

Программист, как правило, получает SDK непосредственно от разработчика целевой технологии или системы. Часто SDK распространяется через Интернет. Многие SDK распространяются бесплатно для того, чтобы побудить разработчиков использовать данную технологию или платформу. Поставщики SDK иногда подменяют слово «software» в словосочетании «**software development kit**» на более точное слово.

Например, Microsoft и Apple предоставляют Driver Development Kit (DDK) для разработки драйверов устройств, PalmSource называет свой инструментарий для разработки PalmOS Development Kit (PDK), а Oracle — Java Development Kit (JDK).

Amazon рисует достаточно наглядную схему своей платформы, на котором мы разберем структуру типичной IoT платформы:

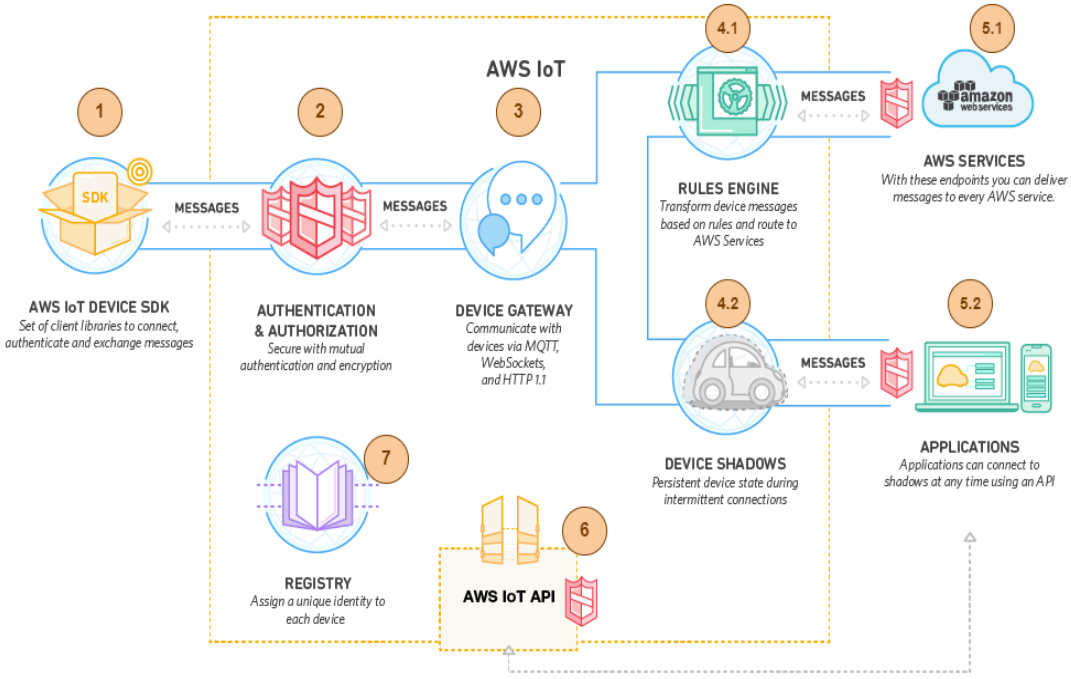


Рисунок 14 – AWS IoT платформа

- (1) Есть устройства, которые взаимодействуют с IoT платформой с помощью SDK.
- (2) Устройства посылают сообщения, которые проверяются службой аутентификации и авторизации.
- (3) Сообщения приходят на Device Gateway, используя разные протоколы и далее попадают в обработчик правил (4.1) и копируются (4.2) на тени устройств (Device Shadows).
- (4.2) Device Shadows – это такие цифровые двойники, которые хранят текущие состояния устройств, которые всегда доступны приложениям. С другой стороны, при отсутствии связи с устройством Device Shadow выполняет управляющие команды от приложений и при восстановлении связи синхронизирует актуальное состояние с устройством.
- (4.1) Обработчик правил в зависимости от поступивших данных выполняет заранее определенные действия (5.1), например, сохраняет данные в DB, посылает SMS или e-mail нотификацию, вызывает HTTP API, отправляет данные в систему аналитики и т.п.
- (5.2) Приложения используют эти данные для контроля и управления устройствами с помощью AWS API (6)
- Информация о всех устройствах хранится на AWS IoT платформе (7).

На следующей схеме (рис.14) рассмотрим CASE-ориентированную схему, которая выглядит несколько сложно. Разберем и ее основные элементы.

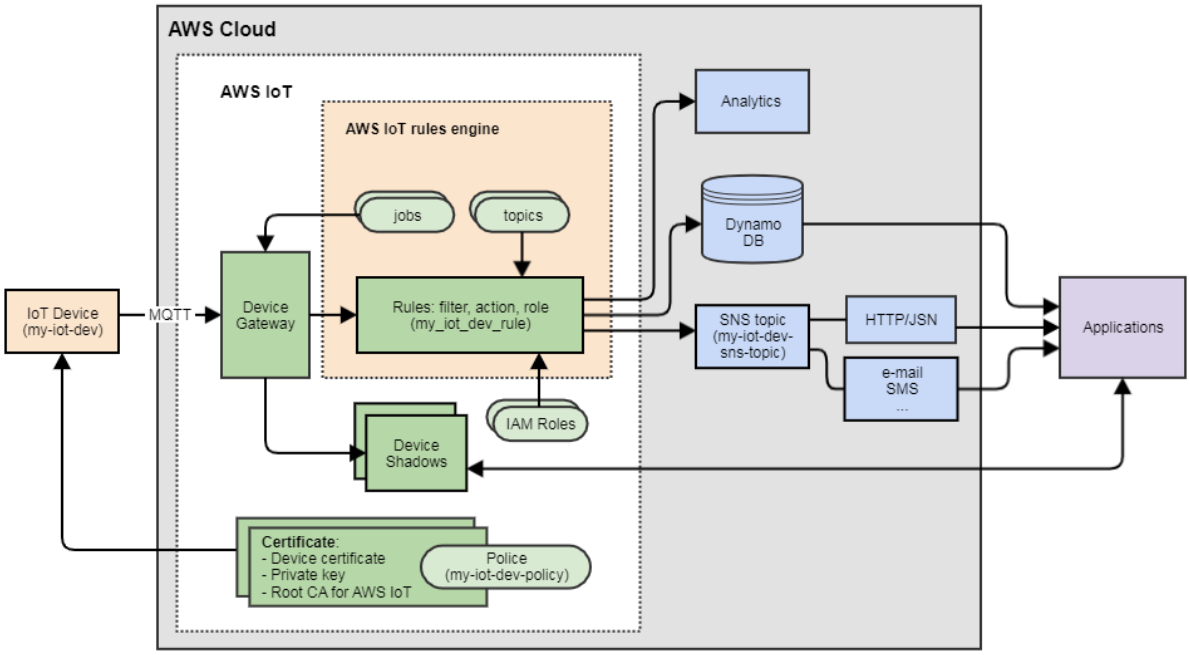


Рисунок 14 – Структура AWS Cloud

Появляются знакомые из курса «Безопасность функционирования информационных систем» понятия:

Jobs – выполняют стандартные действия над устройствами, например устанавливают приложения, обновляют прошивки, производят перезагрузку устройств и т.п.

Topics – сущности MQTT протокола. Сообщения от IoT устройств посылаются в определенные топики.

IAM Roles – AWS пользователи, от имени которых выполняются правила и которые имеют доступ к необходимым AWS ресурсам.

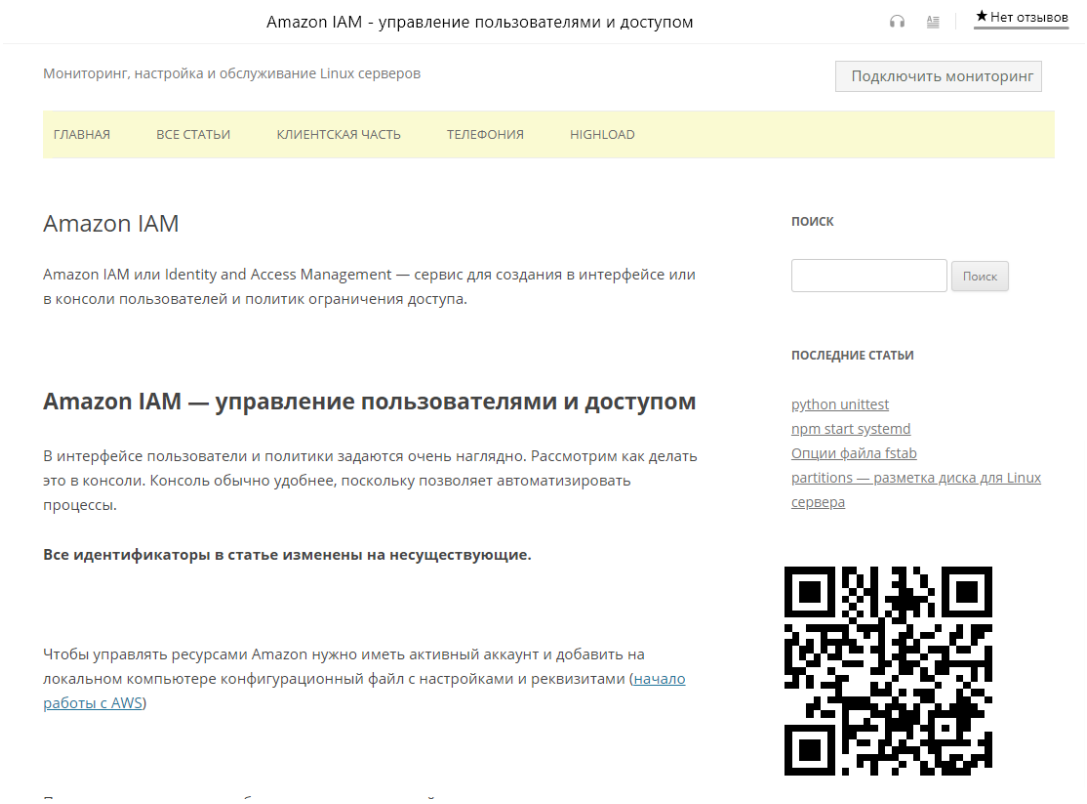
Правила состоят из:

- Filter — фильтр сообщений для обработки. Задается в виде SQL запроса.
- Action — действие, которое надо выполнить.
- Role — одна или несколько IAM ролей.
- Certificate – загружаются на IoT устройство, с их помощью происходит аутентификация устройств на AWS платформе. Которые, в свою очередь, состоят из: сертификата устройства, X.509 Private key, Корневой сертификат AWS платформы.

Policy – к сертификатам прикрепляются политики, которые определяет какие действия можно совершать устройству. С помощью политик происходит авторизация устройств.

Amazon IAM или Identity and Access Management — сервис для создания в интерфейсе или в консоли пользователей и политик ограничения доступа.

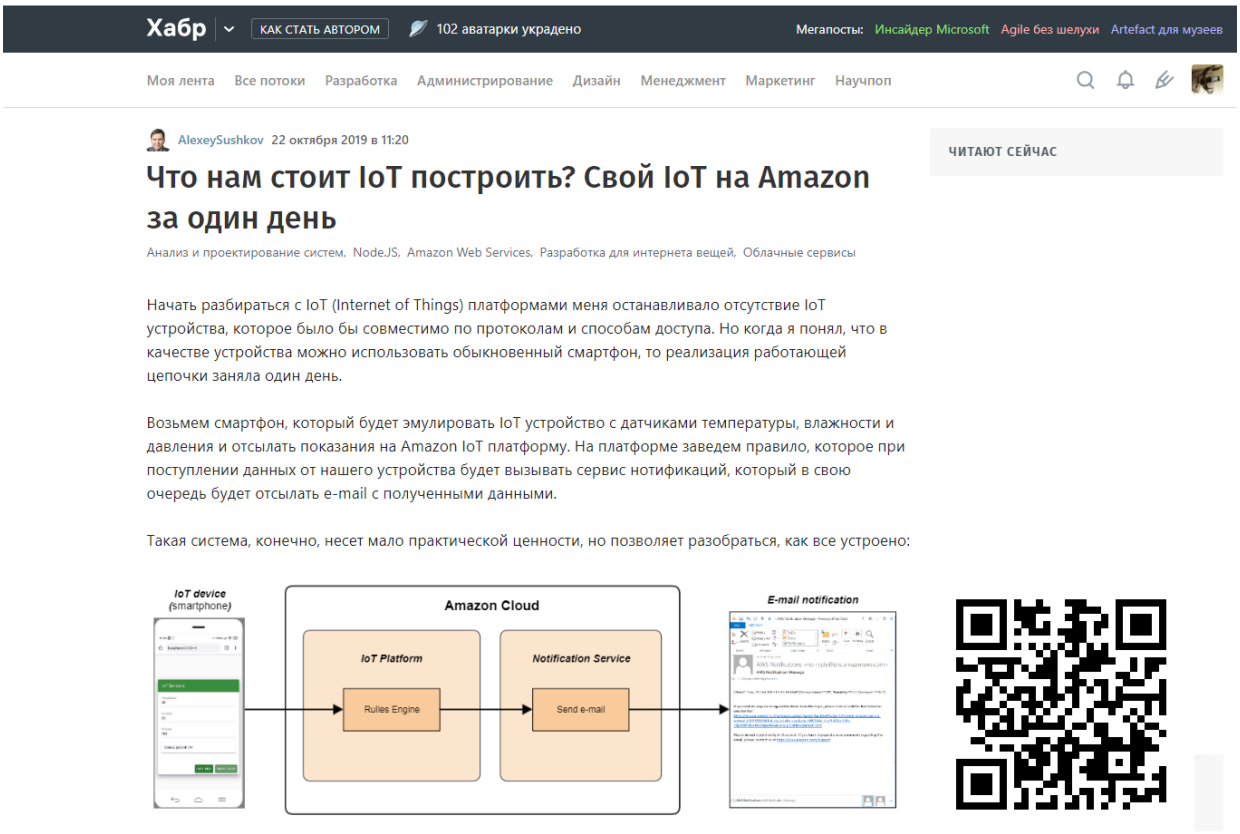
Роли IAM позволяют предоставить права доступа пользователям или сервисам, у которых обычно нет доступа к ресурсам AWS вашей организации. Пользователям IAM или сервисам AWS можно присвоить роли для получения временных данных для доступа, которые они могут использовать для вызовов API AWS. В результате не требуется предоставлять долгосрочные данные для доступа или назначать разрешения для каждого объекта, которому требуется доступ к определенному ресурсу (см. QR).



Полная инструкция по подключению IoT устройства к Amazon платформе есть в открытом доступе по следующему запросу: Getting Started with AWS IoT. Но для понимания объёма задачи перечислю шаги, которые нужно сделать, чтобы схема заработала:

- Создаем на платформе устройство my-iot-dev
- Получаем сертификат устройства X.509, private key, public key
- Получаем корневой сертификат AWS платформы (Root CA for AWS IoT)
- Создаем политику my-iot-dev-policy. Для нашей демо-версии разрешаем все действия: iot:*.

Пошаговое выполнение отражено в интер-отклик ресурсе:



Задача: Выполнить пошагово процедуру развертывания на IoT-платформе. Отчетность по выполнению шаблонного задания реализовать посредством демонстрации видео-захвата активного окна браузер (где будет продемонстрирована панель управления Amazon и т.д.)

Контрольные вопросы

Мы построили IoT, используя платформу от Amazon. Все подобные Amazon, платформы построены по одинаковым принципам, поэтому если встанет вопрос выбора IoT системы, то мы будем готовы задать следующие вопросы. И дальше, зная ответы от Amazon, сможем сделать выводы, насколько зрелая предлагается платформа:

Устройства

- Как устройства добавляются в системе?
- Как обеспечивается аутентификация и авторизация устройств?

Платформа

- Как защищены ключи и сертификаты на платформе?
- Как формируются правила?
- Какие действия могут выполнять правила?
- Как осуществляется мониторинг и управление устройствами?

Взаимодействие

- Как осуществляется взаимодействие приложений с устройством?

§ ПРАКТИКУМ: СОЗДАНИЕ СЕТЕВОЙ МОДЕЛИ ИНФРАСТРУКТУРЫ IoT

ЛАБОРАТОРНАЯ РАБОТА №2: «Решение задачи сетевого планирования»

Цель: разработать сетевую модель проекта разработки и модели его инфраструктуры.

Напомним, что Сетевая модель — графическое изображение плана выполнения комплекса работ, состоящего из нитей (работ) и узлов (событий), которые отражают логическую взаимосвязь всех операций. В основе сетевого моделирования лежит изображение планируемого комплекса работ в виде графа.

Задание:

Разработать **сетевую модель проекта разработки проекта** развертывания IoT-взаимодействия на базе платформы **Amazon Web Services,** определив итерационную последовательность наиболее значимых (по функциональному признаку) шагов, по созданию, на базе освоения навыков, приобретенных ранее. Подразумевается, что количество этапов порядка десяти.

Указание: в мануале, выраженным через интер-отклик, названия всех шагов присутствует. Допускает парафраз и укрупнение (объединение). При описании процессов (этапов) обязательно используйте табличную (матричную) форму подачи. Постройте граф, составьте таблицу описанного графа. Опишите критический путь, если он вообще уместен для вашего проекта (если решение проекта занимает строгую последовательность.

Отразите примерное время длительности процесса (этапа), т.е. отразите в скалярном виде количество времени: в минутах или в секундах, сколько занимает выполнение работы над конкретным этапом.

Разработайте сетевую модель инфраструктуры (достаточно построить только граф-схему), в которой опишите связи между программным и аппаратным блоком. Учитывайте, что **Amazon Web Services – это публичное облако**, т.е., это аппаратный элемент, как и оконечное устройство доступа к панели управления, так, как и IoT-датчик.

Отразите в письменной форме в качестве заключения свои предложения по упрощению модели проекта разработки проекта развертывания взаимодействия. Опишите, как можно упростить по времени и по количеству шагов (этапов) выполнение данной работы.

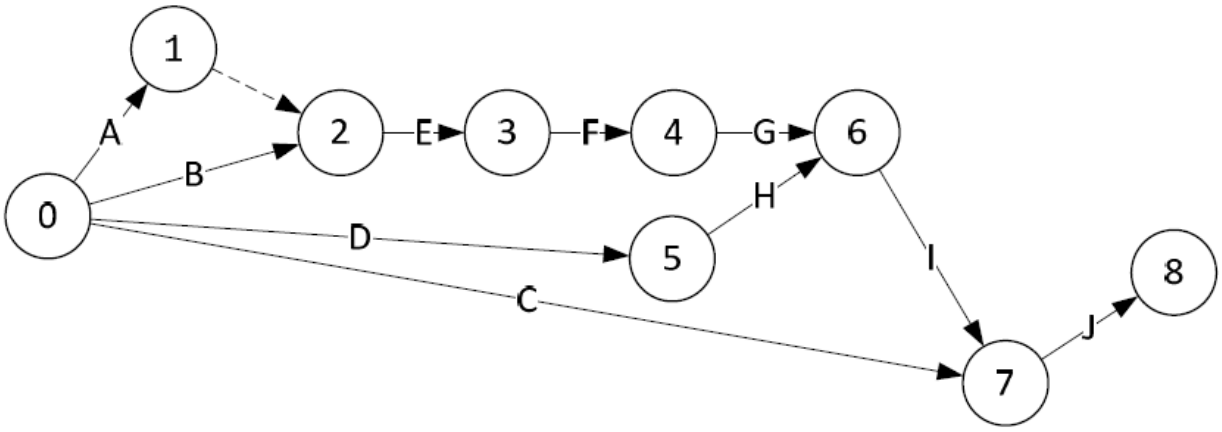
Пример создания сетевой модели на конкретном примере (проекте) представлен ниже:

Пример

Издатель имеет контракт с автором на издание его книги. Ниже представлена последовательность (упрощенная) процессов, приводящая к реализации проекта издания книги. **Необходимо разработать сетевую модель для этого проекта.**

	Процесс	Предшествующий процесс	Длительность (недели)
A:	Прочтение рукописи редактором	-	3
B:	Пробная верстка отдельных страниц книги	-	2
C:	Разработка обложки книги	-	4
D:	Подготовка иллюстраций	-	3
E:	Просмотр автором редакторских правок и сверстанных страниц	A,B	2
F:	Верстка книги (создание макета книги)	E	4
G:	Проверка автором макета книги	F	2
H:	Проверка автором иллюстраций	D	1
I:	Подготовка печатных форм	G,H	2
J:	Печать и брошюровка книги	C,I	4

На рисунке 15 показана сеть, представляющая взаимосвязь процессов данного проекта. **Фиктивный процесс** (2, 3) введен для того, чтобы "развести" **конкурирующие** процессы **A** и **B**. Номера узлов сети возрастают в направлении выполнения проектов.



Пути данного графа.

Путь	Длительность	Критичность
A-E-F-G-I-J	3+2+4+2+2+4=17	v
B-E-F-G-I-J	2+2+4+2+2+4=16	
D-H-I-J	3+1+2+4=10	
C-J	4+4=8	

Рисунок 15 – Модель сети проекта и таблица путей графа

Критический путь: A-E-F-G-I-J, длительность 17 недель.

§ ПРАКТИКУМ: СОЗДАНИЕ SMART CAMPUS

Andrea Finardi «IoT Simulations with Cisco Packet Tracer»

Cisco Packet Tracer - это многоплатформенный инструмент Cisco, позволяющий учащимся создавать модель структуры IoT без использования аппаратного обеспечения или уже существующей сети.

Это бесплатное средство, оно работает в основных операционных системах и доступно для загрузки со страницы Cisco NetAcad для всех учащихся и преподавателей, имеющих действующую учетную запись NetAcad.

Инструмент доступен в течение многих лет для всех студентов, участвующих в курсах Cisco, и первоначально был разработан для поддержки практических упражнений для студентов, посещающих сертифицированные курсы Cisco Network Associated (CCNA) Academy.

На момент написания этой книги последняя доступная версия была 7.2.1. Согласно отчету о корпоративной социальной ответственности за 2017 год Cisco Networking Academy, также называемый NetAcad, на протяжении многих лет обучил более 7,8 миллионов человек в более чем 170 странах. Он используется двадцатью двумя тысячами преподавателей по всему миру.

Инструмент был создан для того, чтобы студенты могли экспериментировать с сетью без необходимости в дорогостоящей сетевой инфраструктуре и длительных процедурах настройки оборудования. На самом деле этот инструмент предлагает обширный набор аппаратных средств и кабелей, которые позволяют учащимся настраивать сеть от базовой до очень сложной, что позволяет им научиться программировать приложения Cisco через интерфейс командной строки (CLI). В нем также рассказывается, как решать проблемы, связанные с сетью, так как инструмент также содержит реалистичные функции для отладки.

Начиная с версии 7.0 Cisco также представила функции **IoT**, что позволяет студентам практиковаться, настраивая устройства IoT и автоматизацию IoT. Также в том же выпуске была предложена возможность моделирования IoT на более низком уровне с использованием одноплатного компьютера (SBC) и цифровых датчиков. Эта вводная статья сосредоточена только на предоставлении моделирования IoT с использованием Cisco Packet Tracer со стороны автора.

Цель этой главы - описать методологию проведения работы следующего этапа практикума, объяснить процесс симуляции в программной среде, методологии и достижения практической части в IoT.

Как упоминалось ранее, первоначальная необходимость в этой работе возникла из-за необходимости создания практического раздела для курса «Интернет вещей», который читают в Хельсинкском университете прикладных наук Метрополия с января 2018 года

У лектора одного из курсов уже была структура теоретических занятий, однако практические занятия также были необходимы, чтобы дать студентам возможность ознакомиться с компонентами IoT. Из-за дополнительной сложности, связанной с наличием реального оборудования, такого как микроконтроллеры, датчики и исполнительные механизмы, было решено использовать симулятор IoT. Выбор был сделан, чтобы использовать инструмент моделирования Cisco Packet Tracer. После того, как потребности и инструмент были прояснены и согласованы, следующим шагом было решить, как структурировать практические занятия, которые из-за ограниченности времени в учебном курсе, было решено развести на две сессии.

Начиная с этого шага, она проводилась в соответствии с типичными методологиями ИТ-проектов. Фактически работа была разделена на пять основных подкатегорий, и были организованы периодические контрольные сессии, чтобы управлять содержанием результатов. Как видно из рисунка 15 ниже, под четырьмя основными категориями были упомянуты этапы: сбор требований, анализ инструмента, разработка среды моделирования, развертывание моделирования во время занятий и, наконец, сбор обратной связи.

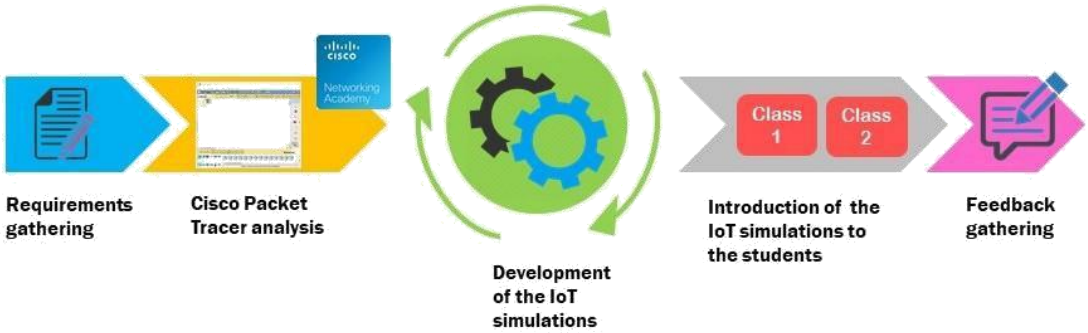


Рисунок 16 – Разработка IoT-практикума на базе Cisco Packet Tracer

Также замечу, что при доступе к порталу Cisco NetAcad было много отличных онлайн-курсов, которые объясняли функционирования IoT. Наряду с конкретными «классами» Cisco Packet Trace, этот инструмент также использовался во многих других сетевых курсах, помогая студентам получать знания поэтапно.

Практикум базируется на обучающих курсах NetAcad: Introduction of Cisco Packet Tracer (0118), Introduction of Cisco Packet Tracer (1217), Packet Tracer 1o1 (2016-11) и Intro to IoT – English – 2016.

Пример:

Цель: Разработать сеть Smart-Campus, который моделирует университетскую среду, где, наряду с традиционными сетями корпусов и аудиторий (в общем виде), существует сеть IoT, которая позволяет подключать различные устройства IoT, распределенные по территории кампуса.

Оборудование: ПК, ПО: Cisco Packet Tracer версии 7.1 и выше.

Реализация:

В качестве примера реализована IoT-сеть управления доступом RFID и интеллектуального решения для полива спортивных полей была включена в примере конфигурации сети ниже (рис.17).

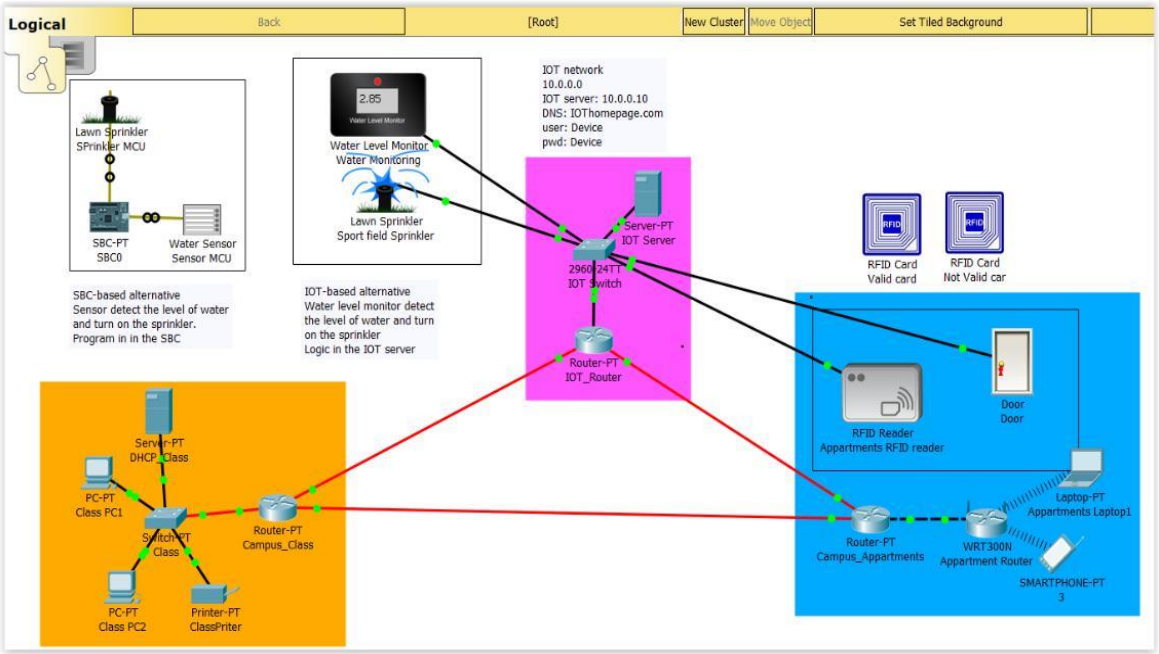


Рисунок 17 - Схема Cisco Packet Tracer для моделирования Smart-Campus

Компоновка сети в этом примере неоднородна и включает в себя: «магистральную сеть», беспроводную локальную сеть для жилых зданий (Appartment Network), локальную сеть учебного отдела (High School/School Network) и выделенную сеть IoT.

«Магистральная сеть» была создана с использованием трех взаимосвязанных маршрутизаторов. Каждый маршрутизатор имеет соединение с двумя другими для создания избыточной инфраструктуры (частично полносвязная топология, partial mesh), которая могла бы противостоять сбоям магистралей между маршрутизаторами (рис.18).

Чтобы представить реалистичную сеть, в которой маршрутизаторы физически размещались в разных зданиях кампуса (вспомним ограничения), вместо традиционных прямых медных кабелей использовалась оптическая проводка в рамках Fast-Ethernet.

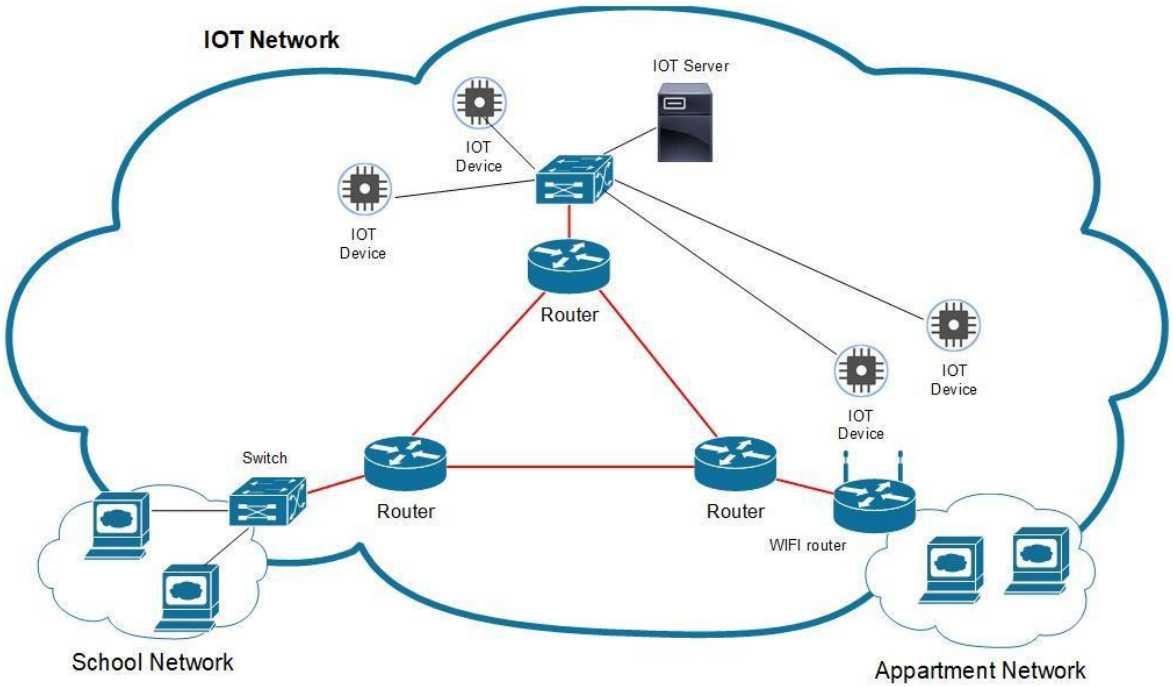


Рисунок 18 - Сетевая топология

Для обеспечения простой маршрутизации между магистральными устройствами, чтобы обеспечить полное соединение между тремя сетями, в конфигурации маршрутизатора использовался базовый протокол информации о маршрутизации (RIP).

RIP - это очень простой и старый протокол маршрутизации, который периодически разделяет таблицу маршрутизации между устройствами. В реальных сложных сценариях протокол обычно не используется из-за его ограничения масштабируемости, поскольку фактически протокол допускает не более пятнадцати сетевых переходов (прыжков) Однако из-за простоты настройки RIP был идеальным кандидатом на использование протокола маршрутизации в упражнении Cisco Packet Tracer.

В каждом маршрутизаторе настройка была выполнена с добавлением IP-адресов напрямую подключенных сетей в конфигурации RIP, как показано на рисунке 19, тогда обмен таблицей маршрутизации позаботится о распределении логики маршрутизации между устройством, как показано на рисунках 20 и 21.

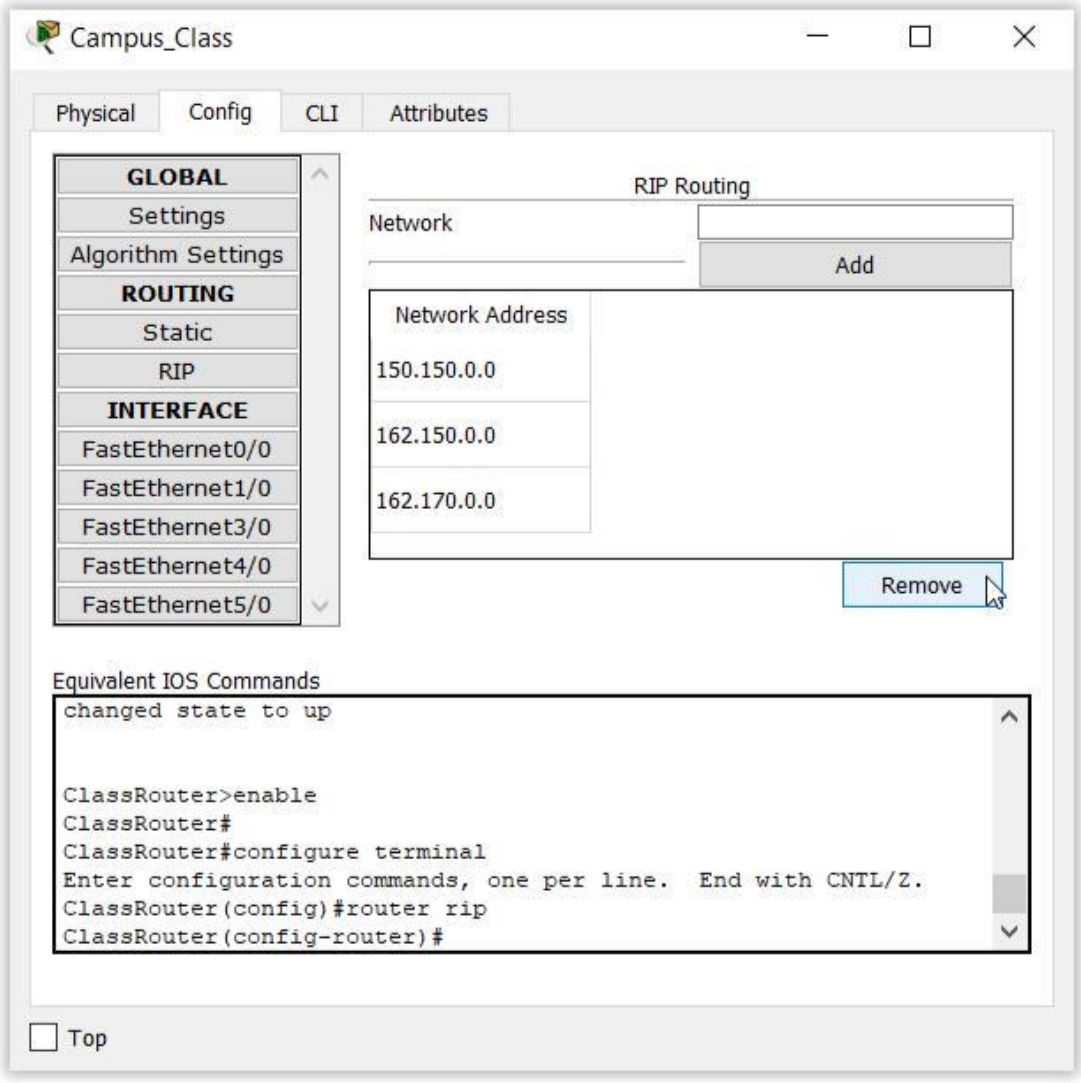


Рисунок 19 - Пример простой настройки RIP в классе маршрутизатора

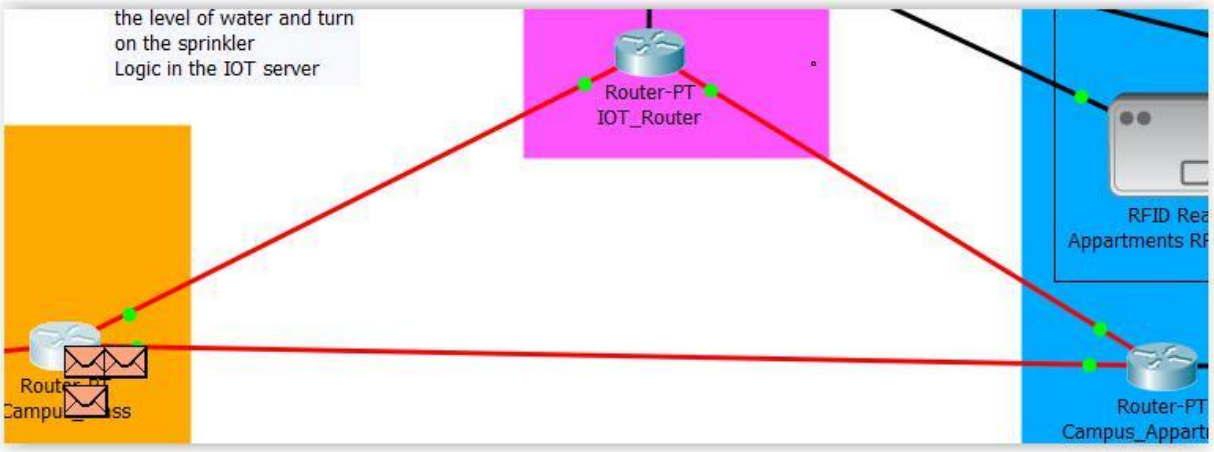


Рисунок 20. Пример передачи сообщений RIP в режиме симуляции

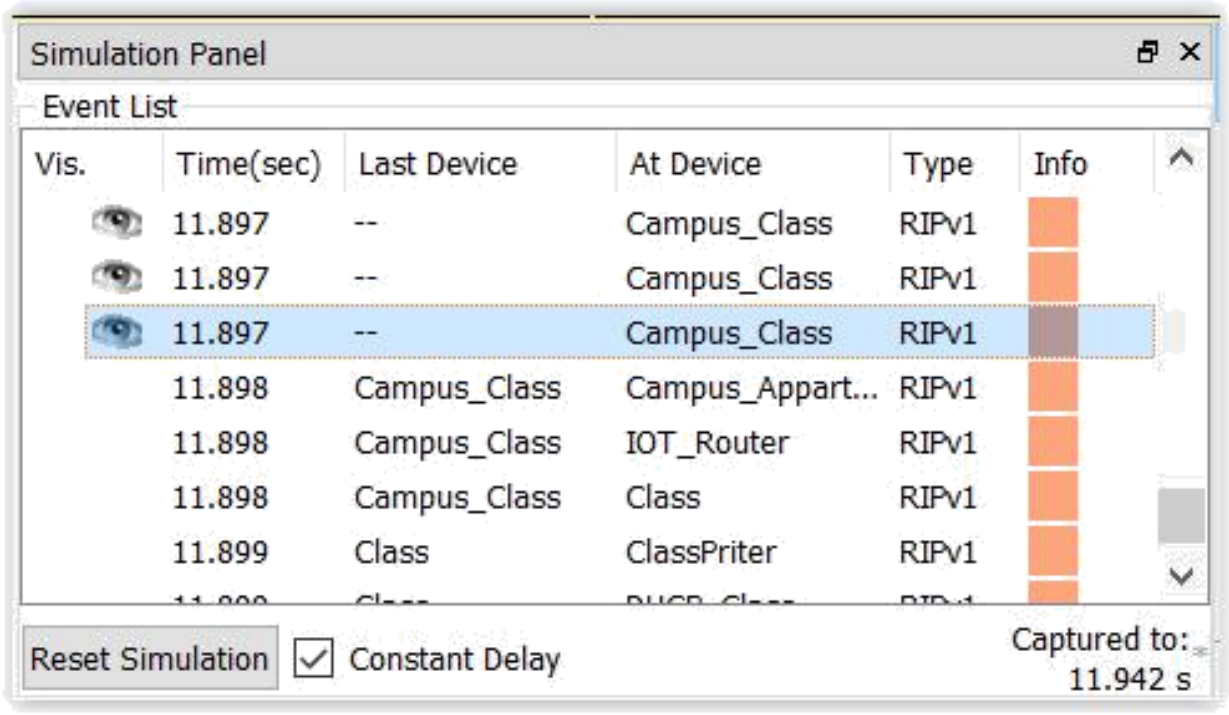


Рисунок 21 - пакет, захваченный во время широковещательной рассылки RIP- сообщений

Наряду с backbone-связью каждый маршрутизатор также был подключен к одной из трех подсетей: сеть класса здания, сеть многоквартирного дома и сеть IoT.

Все три сети были физически разделены путем помещения их в собственный выделенный физический контейнер.

Первая сеть представляла собой простую сеть для эмуляции класса ПК, где два компьютера, один сетевой принтер и сервер были подключены кабелями Ethernet к коммутатору класса.

Затем коммутатор был подключен к одному из портов маршрутизатора. Функции DHCP выполнялись локальными серверами (Server-PT). Кластер Appartment была также простой сетью WLAN, которая имитировала беспроводную связь в жилых домах студентов. В этом случае маршрутизатор WLAN был использован для создания локальной беспроводной сети, затем маршрутизатор был подключен к одному из магистральных маршрутизаторов. Функции DHCP в этой сети также выполнялись маршрутизатором WLAN.

Последней, но самой важной сетью была сеть IoT. Это была сеть на основе коммутатора, подключенная к третьему магистральному маршрутизатору. Все устройства IoT и сервер IoT были подключены к одному коммутатору. В первоначальной спецификации моделирования IoT предполагалось использовать маршрутизатор WLAN для подключения всех устройств IoT, что делает моделирование ближе к реальности. Однако из-за диапазона покрытия сигнала WLAN и отсутствия беспроводных повторителей в Cisco Packet Tracer у самых дальних устройств IoT будет очень плохое покрытие, и иногда подключение к серверу IoT прерывается.

IoT-сервер, в дополнение к IoT-бэкэнду, также использовался в качестве DNS-сервера и DHCP-сервера, выделяющего IP-адрес подключенным IoT-устройствам.

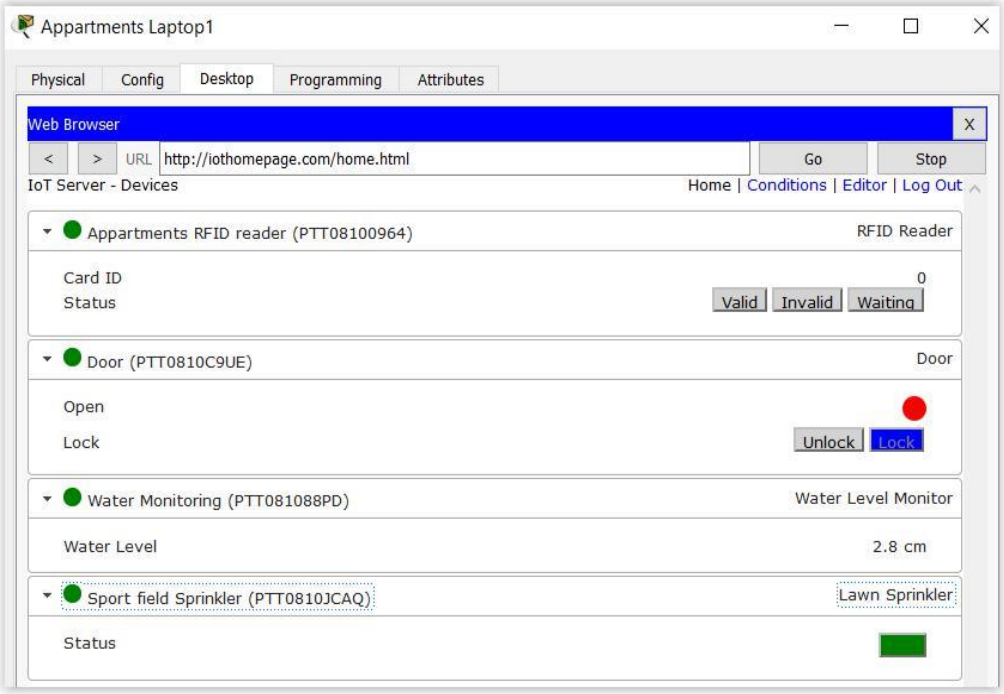
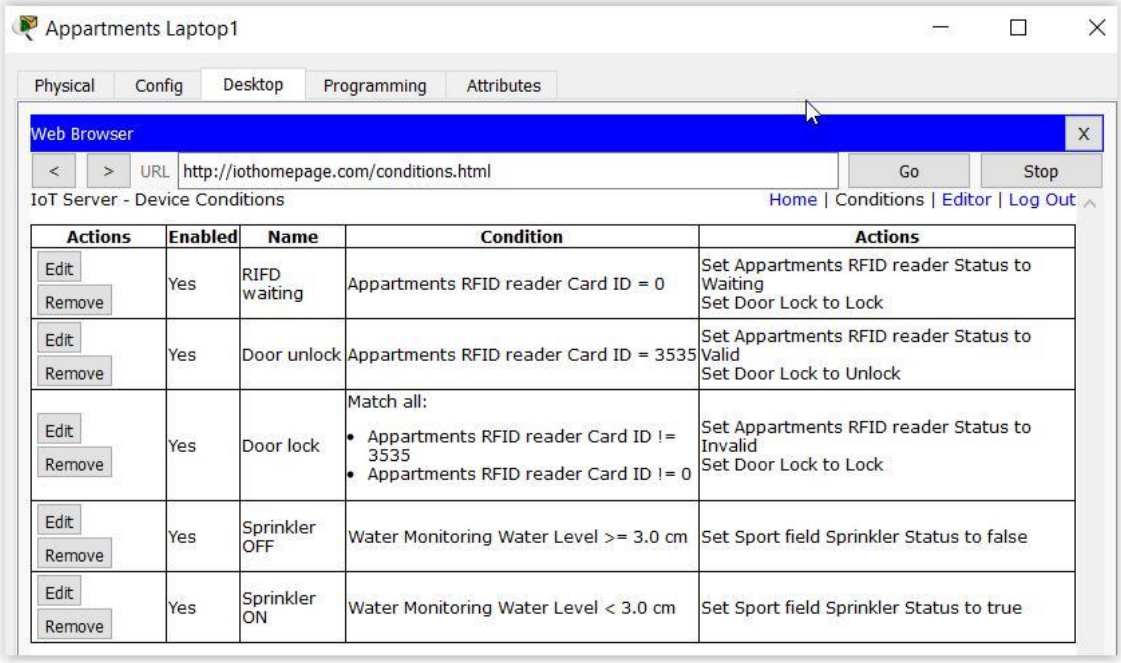


Рисунок 22 - IoT подключенные устройства в моделировании Smart-Campus

Как показано на рисунке 22 выше, в этом упражнении Cisco Packet Tracer были подключены четыре устройства IoT: считыватель RFID, дверь квартиры, датчик контроля воды и разбрызгиватель спортивного поля.

На рисунке 23 также видна логика, назначенная этим четырем устройствам.



Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	RIFD waiting	Appartments RFID reader Card ID = 0	Set Appartments RFID reader Status to Waiting Set Door Lock to Lock
Edit Remove	Yes	Door unlock	Appartments RFID reader Card ID = 3535	Set Appartments RFID reader Status to Valid Set Door Lock to Unlock
Edit Remove	Yes	Door lock	Match all: • Appartments RFID reader Card ID != 3535 • Appartments RFID reader Card ID != 0	Set Appartments RFID reader Status to Invalid Set Door Lock to Lock
Edit Remove	Yes	Sprinkler OFF	Water Monitoring Water Level >= 3.0 cm	Set Sport field Sprinkler Status to false
Edit Remove	Yes	Sprinkler ON	Water Monitoring Water Level < 3.0 cm	Set Sport field Sprinkler Status to true

Рисунок 23 - Предварительно установленные условия для IoT-устройств

Первая автоматизация была направлена на создание решения для контроля доступа IoT RFID для управления доступом в студенческих квартирах с использованием считывателя RFID, пары карт RFID и умной двери.

Концепция была очень проста: когда авторизованная RFID-карта соприкасается с считывателем RFID, дверь открывается, но если использовалась несанкционированная RFID-карта, дверь остается закрытой. Для достижения такого сценария параметры карты RFID были изменены путем редактирования вкладки атрибута карты, как показано на рисунке 24 ниже.

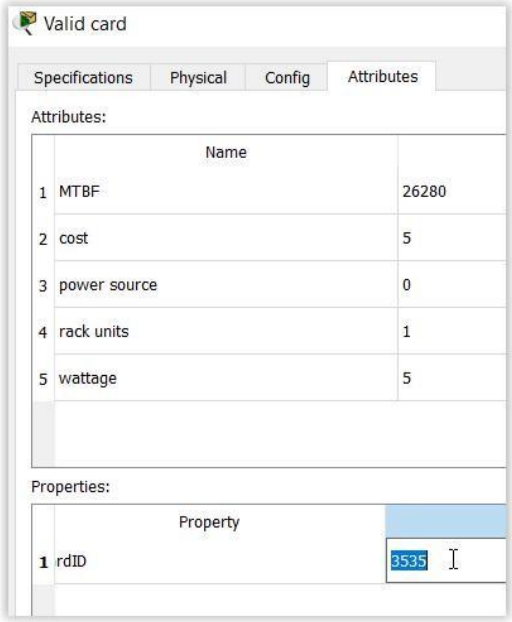


Рисунок 24 - модификация атрибута RFID-карты

В этом случае использовались две карты, одна со значением «id3535», а другая со значением «id1212». Как показано на рисунке 23, логика авторизации была выполнена на бэкэнд-сервере IoT, так как было установлено условие разблокировки двери при замене «id3535».

Смена карты был сделана простым перетаскиванием карты на считыватель RFID с помощью мыши. Зеленый значок, появившийся в считывателе, означал, что карта была авторизована, в то же время значок двери стал красным с зеленого. Если использовалась неправильная карта со значением 1212 в примере, значок считывателя оставался красным вместе со значком запертой двери, как видно на рисунке 26 (разница видна при цветной печати):

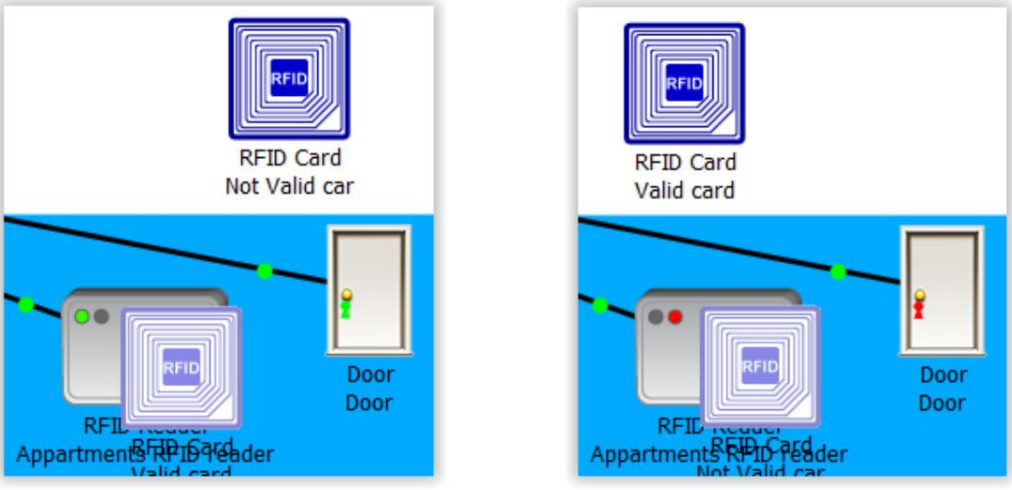


Рисунок 25 – Возможные варианты срабатывания

Для этого конкретного случая, и только из-за логики работы считывателя было установлено, третье условие ожидания. Считыватель фактически оставался в цикле ожидания, готовый принять новую карту, пока любая карта не была помещена на него. Пока считыватель был в этом цикле, умная дверь оставалась запертой.

Спецификации того, как работает считыватель, и все ожидаемые состояния были четко объяснены на вкладке технических характеристик устройства в симуляторе Cisco. В этом сценарии «умная дверь» использовалась как устройство, которое реагировало на условия, однако, благодаря своей универсальности, устройство могло использоваться в более сложных симуляциях. Прямое взаимодействие с дверью также было возможно, нажав ALT на клавиатуре и щелкнув значок двери.

Моделирование RFID было создано только с основной целью показать разные сценарии IoT. Однако ясно, что в более сложных приложениях использование внутренней логики IoT было не лучшим вариантом, поскольку при простых условиях было невозможно достичь различных комбинаций карт и разграничения уровней доступа.

Следует также упомянуть, что в Cisco Packet Tracer считыватель RFID не всегда работает правильно. При первом запуске симуляции читатель не принимает карты. Студентам рекомендуется всегда останавливать и перезапускать внутреннюю программу считывателя при первом запуске упражнения.

Это было сделано путем нажатия на устройство, затем на кнопку продвижения и, наконец, на вкладке «Программирование», как только там можно было остановить и снова запустить программу, как показано на рисунке 26 ниже.

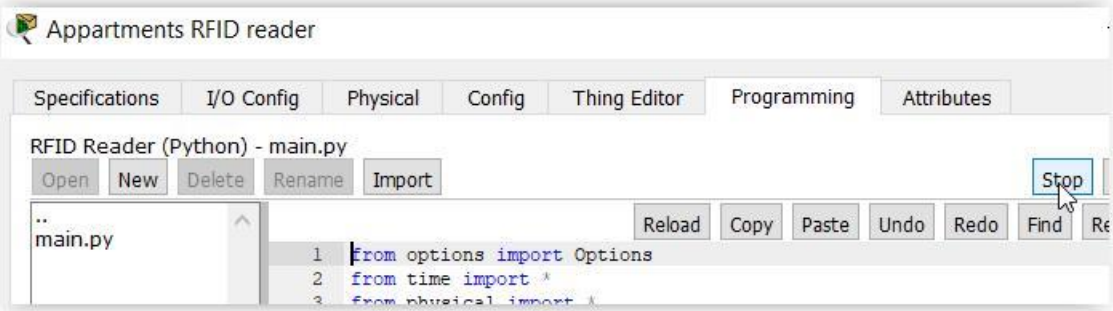


Рисунок 26 – Остановка программы RFID-reader.

Во втором случае, IoT в этом демонстрационном упражнении был смоделирован через реализацию интеллектуальной системы полива спортивных полей, где датчик «занимался» определением уровня воды и, в случае низкого уровня, запускал разбрызгиватель воды (логика видна на рисунке 23). Чтобы сделать автоматизацию более реалистичной, переменные среды контейнеров для спортивных площадок были изменены, чтобы имитировать дождь в течение нескольких часов в течение дня.

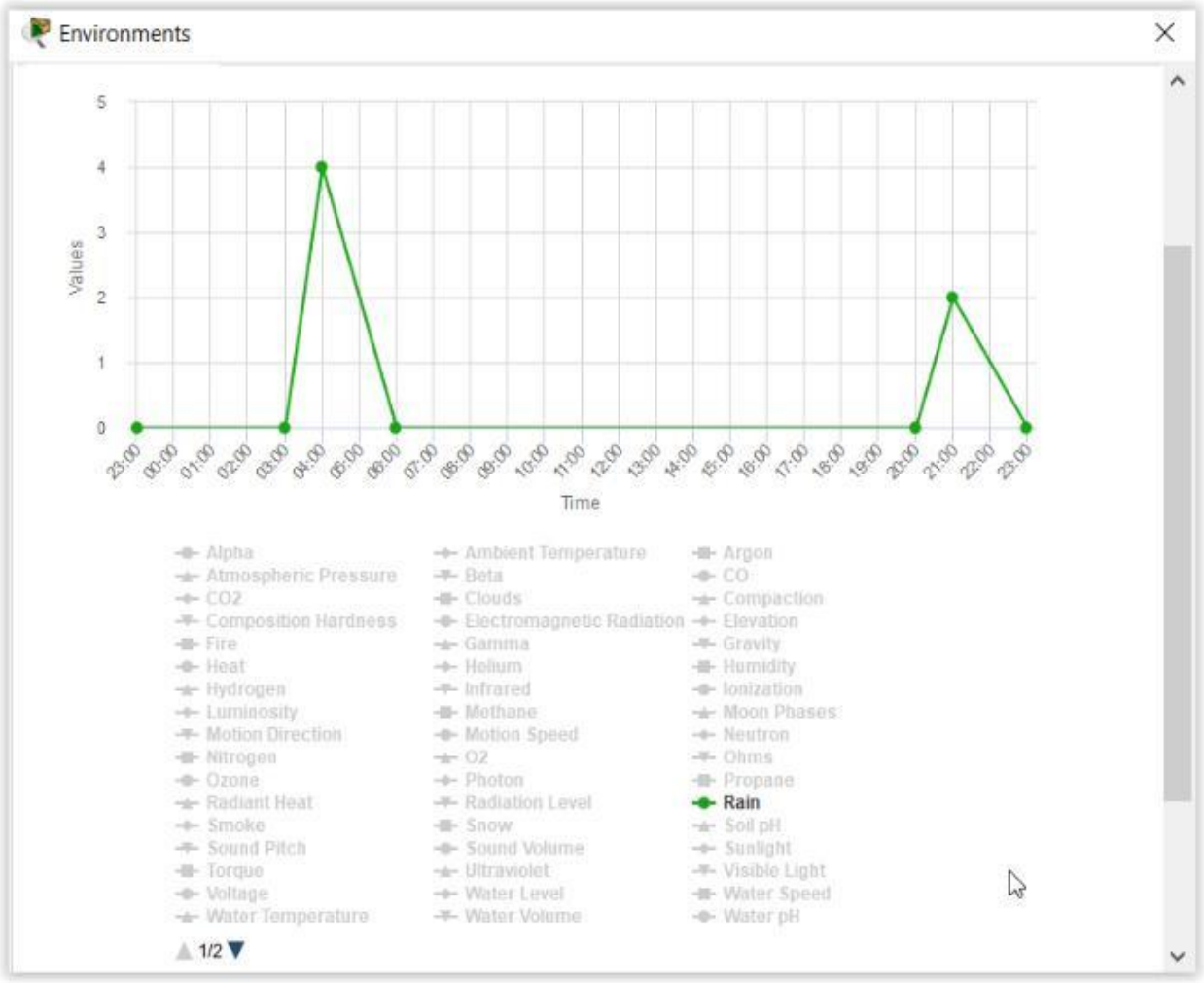


Рисунок 27 – «Переменные среды» для моделирования Smart-Campus

Как видно из рисунка 27, осадки выпадали с 03:00 до 06:00 и с 20:00 до 23:00. Дожди повысили уровень воды в контейнере для спортивной площадки, в результате чего датчик воды обнаружил большее количество воды и не запустил разбрызгиватели.

На рисунке 28 показано действие разбрызгивателя, когда уровень воды был ниже или выше трех сантиметров.

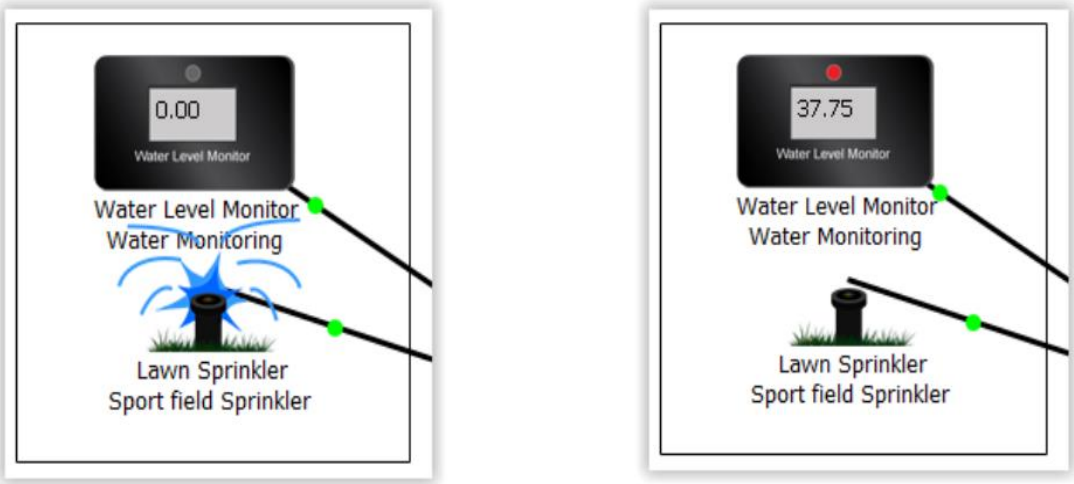


Рисунок 28 - Смарт-спринклер работает и не работает

Скорость поглощения воды в полевых условиях также может быть увеличена путем корректировки солнечного света и температуры в переменных среды.

Пример IoT микроконтроллера

Как и в предыдущих двух симуляциях умного дома, в упражнении Cisco Packet Tracer для интеллектуального кампуса был добавлен пример микроконтроллера. Моделирование воспроизводило тот же случай спринклера IoT при использовании не интеллектуальных устройств. Входные контакты модуля **SBC (микроконтроллера)** были подключены через специальный кабель IoT к «не умному» датчику воды. Выход SBC был затем подключен к разбрызгивателю газона, как показано на рисунке 29:

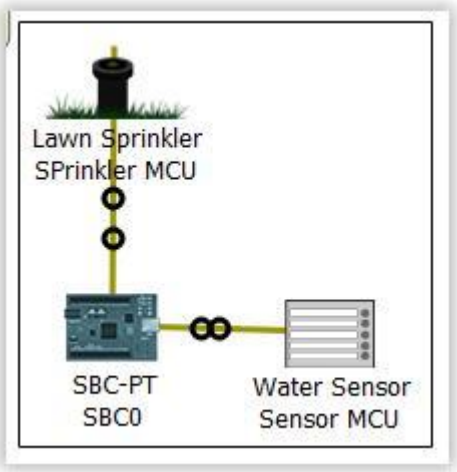


Рисунок 30 - Пример настройки микроконтроллера под IoT-среду.

Как и в IoT-версии моделирования, датчик воды обнаружил уровень воды и включил ороситель, если было обнаружено значение ниже порогового значения.

Разница по сравнению со случаем IoT заключалась в том, что логика была определена не на внутреннем сервере IoT, а в пользовательской программе **Blockly***, хранящейся в самом SBC.

Другое небольшое отличие заключалось в том, что используемый датчик воды возвращал число от 0 до 255, затем это значение было сопоставлено в программе SBC с диапазоном от нуля до двадцати сантиметров.

Также в этом случае переменные среды дождя были изменены, чтобы выпадать дождь в течение дня.

Перспективы развития сети:

В виду возможностей, открываемых сложностью упражнения, студентам было предложено широко модифицировать симуляцию, добавив новые функции сети и IoT.

На уровне сети все устройства имели доступ к удаленному серверу IoT. Ограничения доступа могут применяться, чтобы убедиться, что только авторизованные IP-адреса могут подключаться к нему.

Из-за ограничений беспроводного диапазона все устройства IoT были подключены через коммутатор, поскольку в реальных приложениях это, вероятно, было бы нереально, студенты могли бы перестроить сетевую настройку IoT, чтобы создать более широкую сеть WLAN или кластер проводной подсети IoT. В перспективе IoT в сценарии кампуса можно добавить еще много симуляций: электричество, генерируемое для питания уличных фонарей, сеть датчиков безопасности, планы эвакуации, чтобы открыть все двери и окна и т. д.

Кроме того, примеры микроконтроллеров могут быть интегрированы в моделирование, чтобы охватить случаи, когда покрытие WLAN сети IoT было недостаточным.

---

* Блокли (англ. Blockly) — библиотека для создания среды визуального программирования, которая может быть встроена в произвольное веб-приложение. Блокли включает в себя графический редактор, позволяющий составлять программы из блоков, и генераторы кода для подготовки исполнения программы в среде веб-приложения.

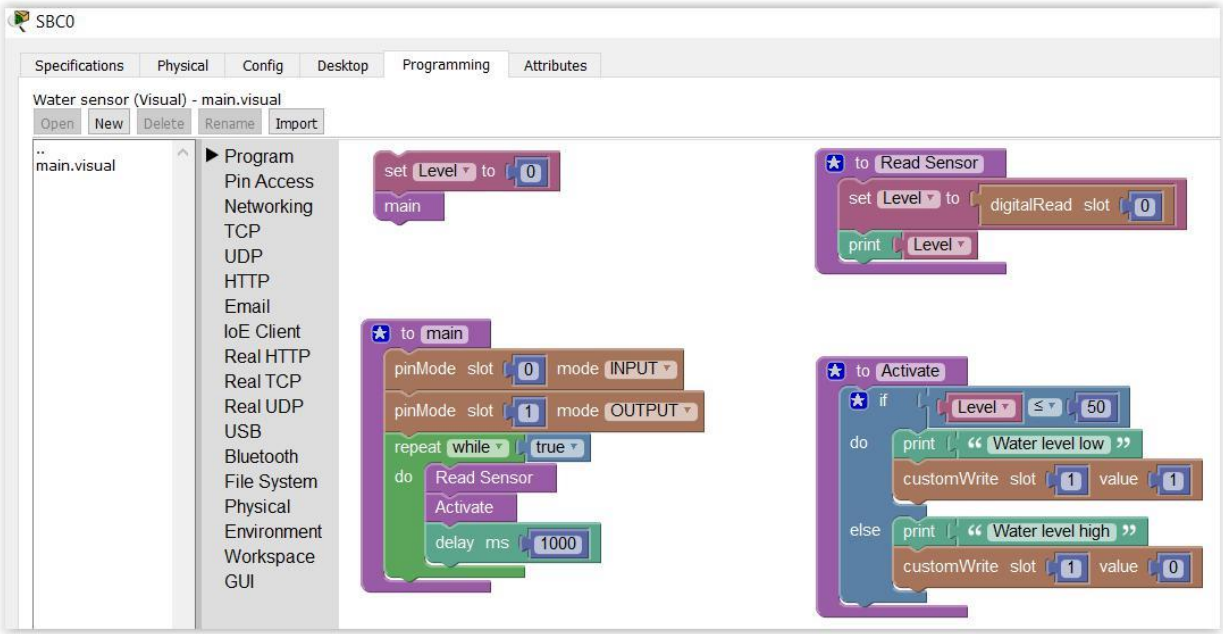


Рисунок 31 – Blockly для программирования логики работы датчиком

Сетевая карта сети:

Class building network IP: 150.150.0.0 Subnet: 255.255.0.0 DHCP server: 150.150.0.1

Apartments building network SSID: AppartmentWIFI Password: HomeWIFI IP: 210.140.0.0 Subnet: 255.255.0.0

IOT network IP: 10.0.0.0 Subnet: 255.0.0.0 DHCP server: 10.0.0.10

IOT Server (remote) IP: 10.0.0.10 DNS: IOThomepage.com User: Device Password: Device

Лабораторная работа №3: Создание «умной» сети кампуса.

Цель: Разработать сеть Smart-Campus, который моделирует университетскую среду, где, наряду с традиционными сетями корпусов и аудиторий (в общем виде), существует сеть IoT, которая позволяет подключать различные устройства IoT, распределенные по территории кампуса.

Оборудование: ПК, ПО: Cisco Packet Tracer версии 7.1 и выше.

Объект исследования: IoT-компоненты в Cisco Packet Tracer (рис. 32), IoT-инфраструктура, Smart City, Smart Campus.

Постановка задачи (общее задание):

Создать сеть учебного заведения, состоящего из 4 сооружений. Общее количество узлов всей сети не должно превышать 250 устройств. Реализовать подсистемы IoT для каждого из зданий, используя все перечисленные датчики. Распределить нагрузку в развертывании на рабочие группы по указаниям преподавателя. Отчетность в форм-факторе примера, обозначенного ранее.

Частное задание для каждой из рабочих групп представлено на следующих страницах.

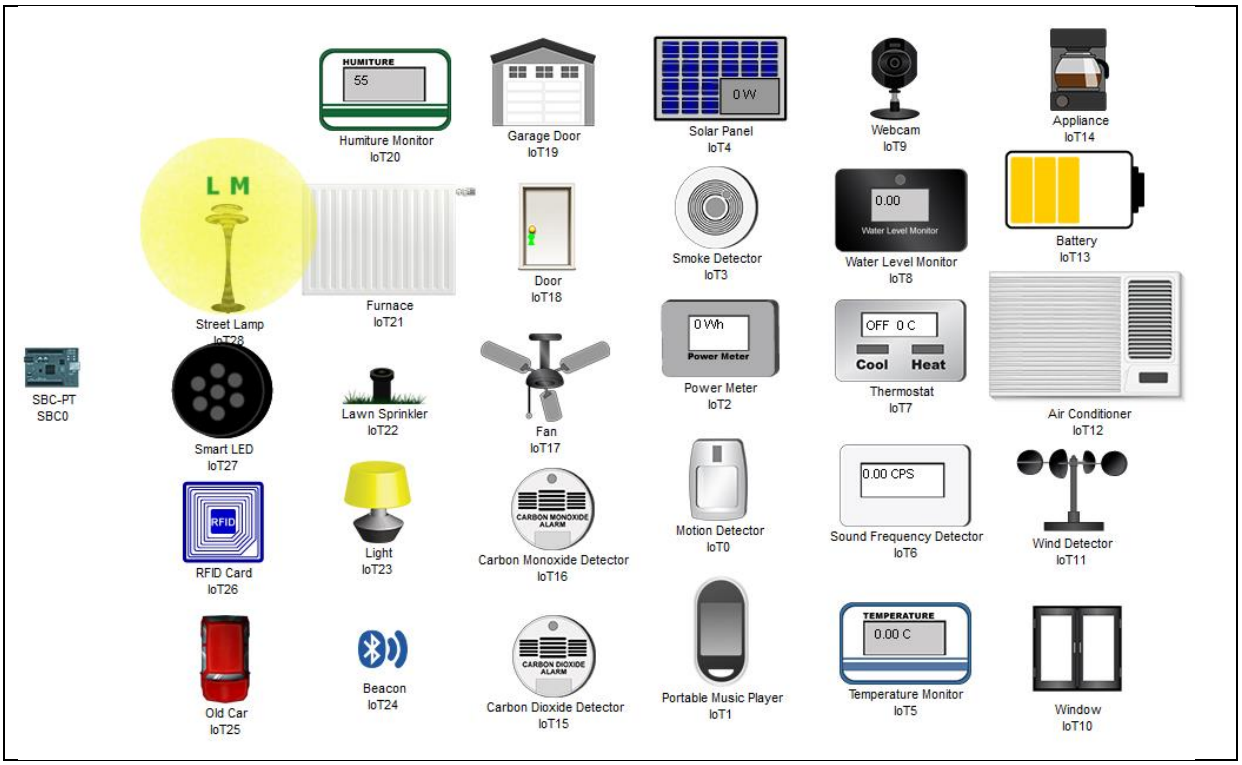


Рисунок 32 –IoT-датчики в Cisco Packet Tracer

Методические указания к работе.

Разработка сетевой инфраструктуры с чистого листа без прямых на то указаний со стороны заказчика - сложная, и порой, рискованная, работа. Поэтому, в первую очередь, необходимо создать «дорожную карту» этапов проектирования, развертывания системы.

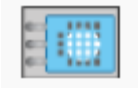
Безусловно, мы должны понимать, что разработка предлагаемой сети, подразумевает большие усилия, нежели проектирование обычной локально-вычислительной сети. Как было показано ранее, в примере, IoT-сегмент по сути представляет «сеть» внутри сети. И ее топологическая составляющая непосредственно связана с подключаемой «классической» сетью. Безусловно, формальный вид такой задачи нужно сформулировать при помощи методов сетевого моделирования, обозначив процессы (этапы), связи между ними (даже если они нелинейны/неочевидны).

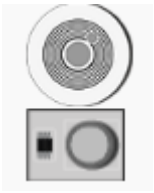
1. Разработать сетевую модель проекта.
   1.1. Определить процессы (например: выбор топологии сети (в целом), активного сетевого оборудования «магистрального уровня» и так далее вплоть до конфигурации микроконтроллера отдельно взятой IoT-подсистемы (определив их численность и функции, обосновав и по необходимости, объединив между собой на уровне конечных устройств (датчиков), если это не противоречит логике.
   1.2. Провести анализ, построить граф-схему, определить примерные сроки реализации проекта (процессы, которые трудно проанализировать с точки зрения временных интервалов не заполнять)
2. Разработать логическую схему системы.
3. Оформить финальное представление о топологии сети.
4. Определить целевое предназначение конкретных узлов (сегментов), обозначив их прямо в симуляции (имеется в виду «классический сегмент сети»).
5. Определить, нужна ли IoT-инфраструктура. Реализовать ее. Дать описание используемым датчикам, конфигурациям, подсистемам.

6. Посредством CASE-средств (UML) или иным графическим способом сформулировать и отразить графически сценарии использования всех IoT-компонентов.

Далее приведена общая сводная таблица по всем основным устройствам IoT в симуляторе и их влиянии на поведение «окружающей среды» или характеристики.

Название на английском	**Перевод**	
ATM Pressure Sensor	Датчик давления 	Диапазон обнаружения по умолчанию составляет от 0 до 110 кПа.
Carbon Dioxide Detector	Газосигнализатор (углекислый газ) 	
Carbon Monoxide Detector	Газосигнализатор (угарный газ) 	
Door	Дверь 	Влияет на уровень (концентрацию) аргона, угарного газа, двуокися углерода, водорода, гелия, метан, азот, кислород, озон, пропан и дыма. Когда дверь открыта, эти газы уменьшатся до 2% от максимума.
Fan	Вентилятор 	Влияет на скорость ветра, влажность и температуру окружающей среды. При установке низкой скорости скорость ветра устанавливается в рамках 0,4 км/ч. Скорость охлаждения температуры окружающей среды установлена на -1 ° C / час. Скорость снижения влажности установлена на -1% в час.

		При установке высокой скорости, скорость ветра устанавливается 0,8 км / ч. Скорость изменения температуры окружающей среды и влажности в два раза выше.
Fire Sprinkler	Пожарный спринклер (разбрызгиватель) 	Влияет на уровень воды со скоростью возрастания 0,1 см в секунду. Влияет на влажность со скоростью 5% в час.
Garage Door	Гаражная дверь 	Влияет на аргон, угарный газ, двуокись углерода, водород, гелий, метан, азот, кислород, озон, пропан и дым. Когда дверь открыта, общее количество этих газов уменьшится до 4% от максимума. Когда дверь открыта, коэффициент переноса (исчезновения) влажности и температуры увеличивается на 50%.
Home Speaker	«Умная колонка» 	Влияет на громкость звука, повышая ее на 65 дБ. Влияет на белый шум на +20%
Humidifier	Увлажнитель 	Влияет на влажность (абсолютную) со скоростью + 1% в час.
Humidity Sensor	Датчик влажности 	
Humiture Monitor,	Монитор влажности воздуха 	Определяет температуру и влажность окружающей среды и выводит значение в виде суммы температуры и влажности окружающей среды, деленной на 2.
Lawn Sprinkler, Floor Sprinkler	Разбрызгиватели Воды.	

LED	Светодиод 	
Light	Торшер (в контексте) 	
Car	Автомобиль 	Влияет на угарный газ со скоростью +1% в час. Влияет на диоксид углерода (углекислый газ) со скоростью +2% в час. Влияет на дым в размере +3% в час. Влияет на температуру окружающей среды со скоростью +1% в час.
Photo Sensor	Фотодатчик 	
Piezo Speaker	Пьезоэлектрический динамик 	
Smoke Detector, Smoke Sensor	Детектор дыма, Датчик дыма 	
Solar Panel	Солнечная панель 	
Temperature Monitor	Монитор температуры 	

Temperature Sensor	Датчик температуры 	
Window	Окно 	
Drain Actuator	Дренажный привод 	
Furnace, Heating Element	Нагревательный элемент 	Влияет на влажность со скоростью -2% в час. Влияет на температуру окружающей среды со скоростью +10 ° C в час.
AC, Air Cooler	Охладитель воздуха (в контексте) 	Влияет на влажность со скоростью -2% в час. Влияет на температуру окружающей среды со скоростью -10 ° C в час.

Некоторые из перечисленных выше устройств не имеют интерфейса Ethernet, следовательно, данные компоненты IoT не могут быть напрямую подключены к домашнему шлюзу. Такие датчики или актуаторы нужно подключать к плате микроконтроллера (MCU-PT). MCU-PT может быть шлюзом (однако, в таком случае сеть «видит» только плату микроконтроллера, но не компоненты IoT (рис.33).

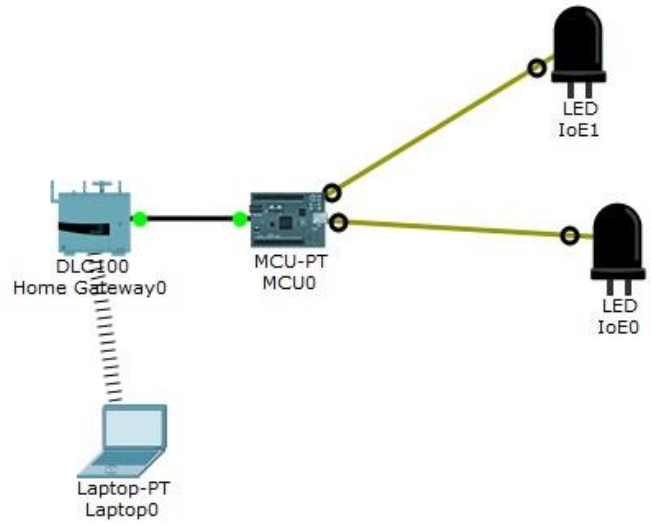


Рисунок 33 – Реализация сети с непрямым доступом к IoT-устройству

Домашний шлюз (Home-Gateway) использует API для удаленного управления, и, в конечном итоге, программную составляющую логики работы устройства на MCU-PT, чтоб получить статус датчика (рис.34)/изменение состояния актуатора.

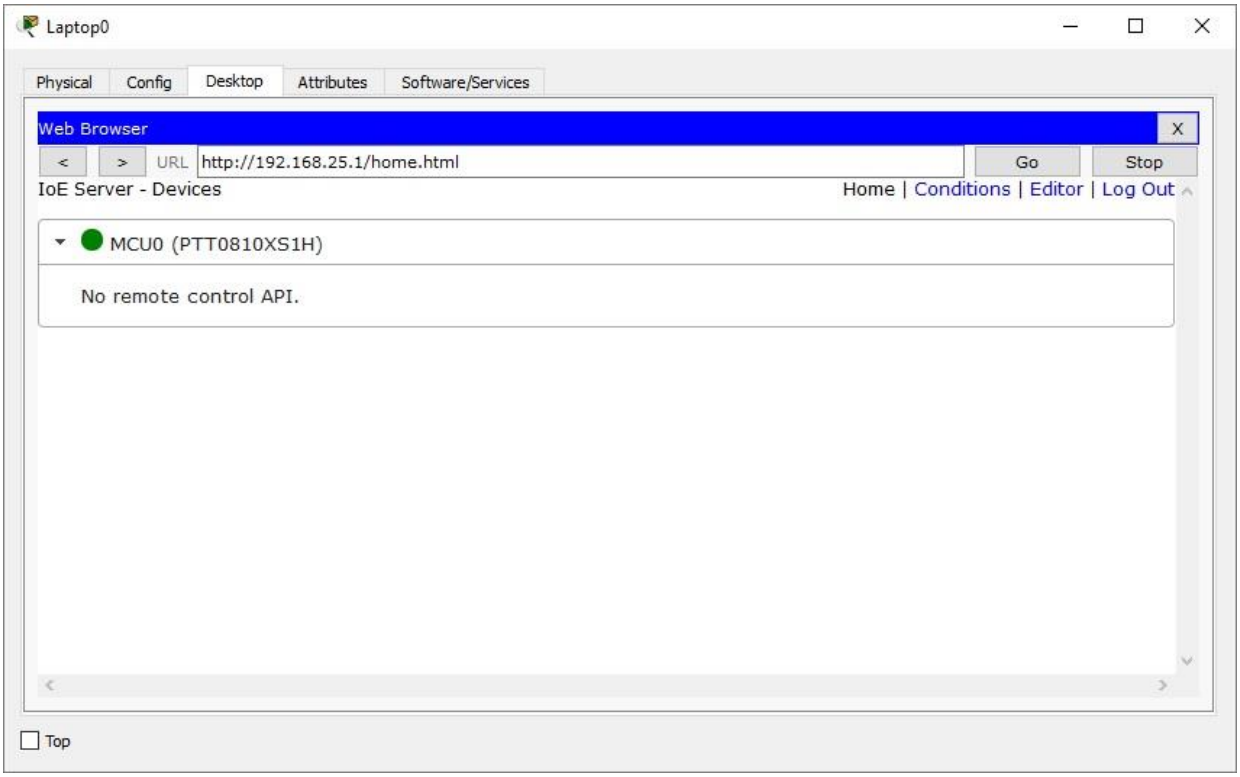


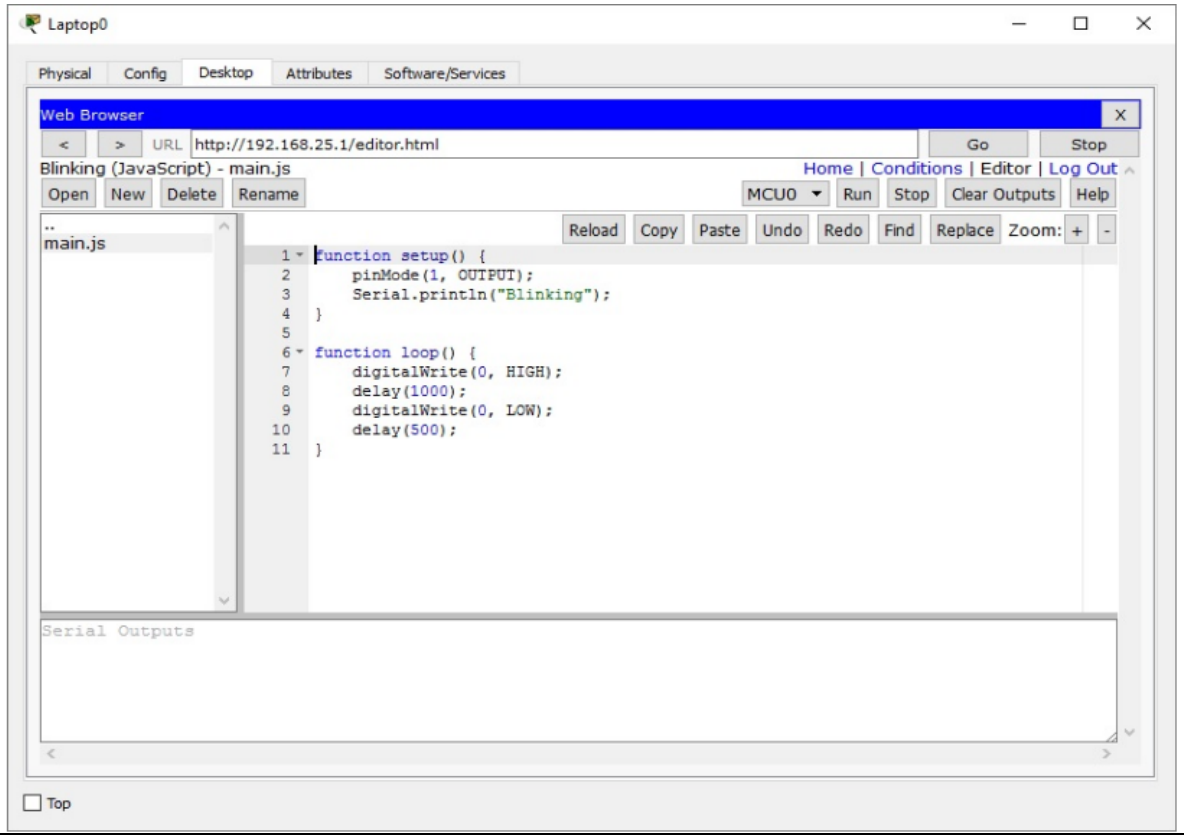
Рисунок 34 – Веб-интерфейс статуса работы MCU

Безусловно, для реализации механизма правильной работы, как и говорилось ранее, нам нужно провести программирование MCU, определив порты, которые будут связаны с IoT-устройствами, которыми нельзя управлять напрямую. В Cisco Packet Tracer редактор для программирования IoT-устройств включен в веб-интерфейс Home Gateway. Это позволяет программировать Javascript или Python микроконтроллера MCU-PT.

Следующий пример кода заставляет светодиод, подключенный к порту digital0, мигать.

```
function setup() // первоначальные настройки
 {
        pinMode (0, OUTPUT); // режим пина.
        Serial.println("Blinking");
}

function loop() {
        digitalWrite(0, HIGH);  //На pin Digital 0 подаем 5В
        delay(1000);
        digitalWrite(0, LOW);
        delay(500);
}
```



Как видно на рис.35 , Packet Tracer 7.1.1 эмулирует интегрированную среду разработки (IDE) Arduino для программирования объектов IoT.

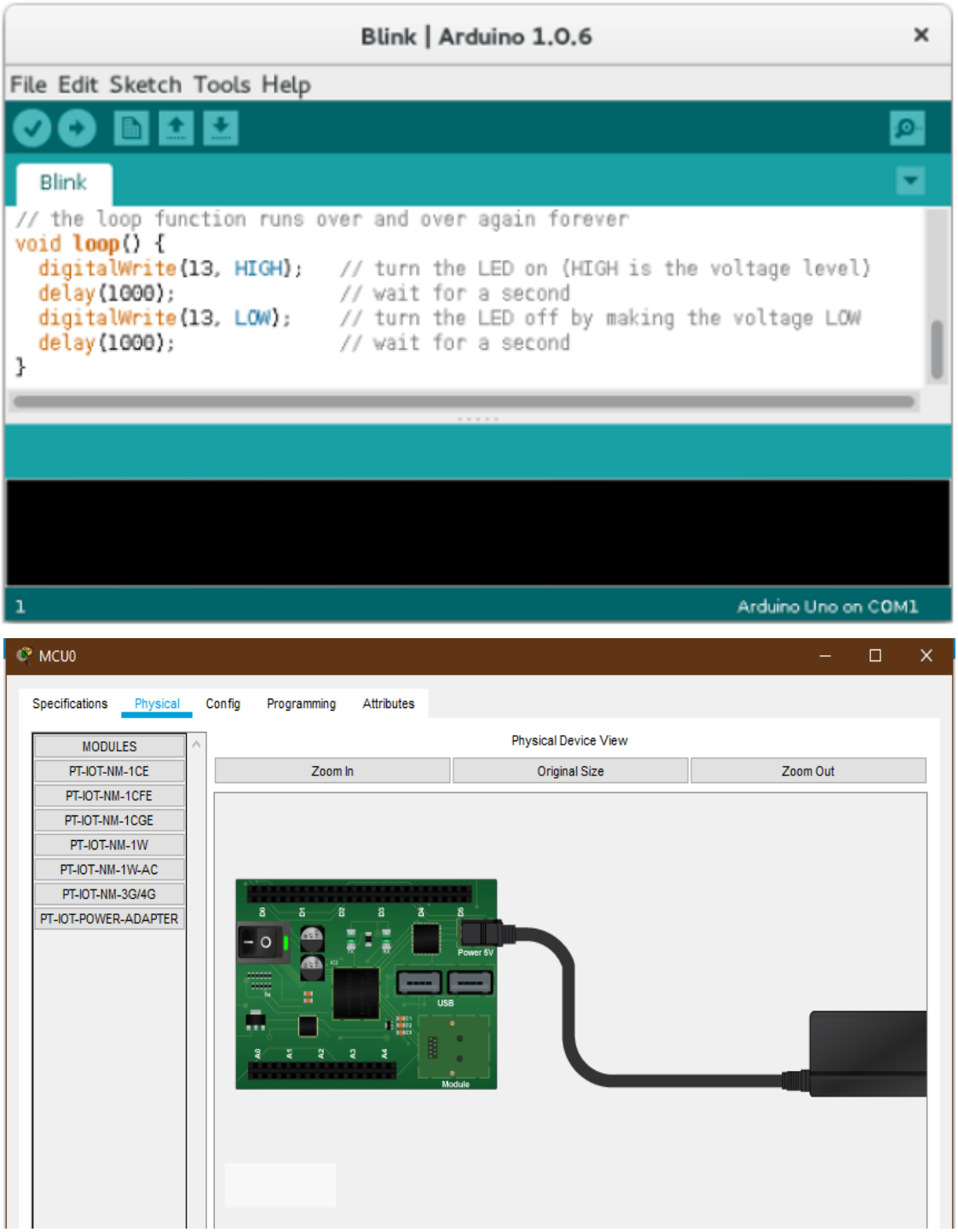


Рисунок 35 – IDE Arduino и MCU

Данный элемент практикума дает первичные практические навыки в симуляции IoT-сетей. Данная работа является интродукцией к методическим указаниям к лабораторным работам «Компьютерные сети. Internet of Everything», в котором будут освещены все аспекты IoT в симуляторе Cisco

В расположенной, посредством интер-отклика статье «Boosting Campus Network Design Using Cisco Packet Tracer» представлен полноценный вариант настройки IoT-окружения, который, в свою очередь, поможет вам в создании проекта по общим требованиям:

Volume 2, Issue 11, November– 2017 International Journal of Innovative Science and Research Technology
ISSN No:-2456 –2165

Boosting Campus Network Design Using Cisco Packet Tracer

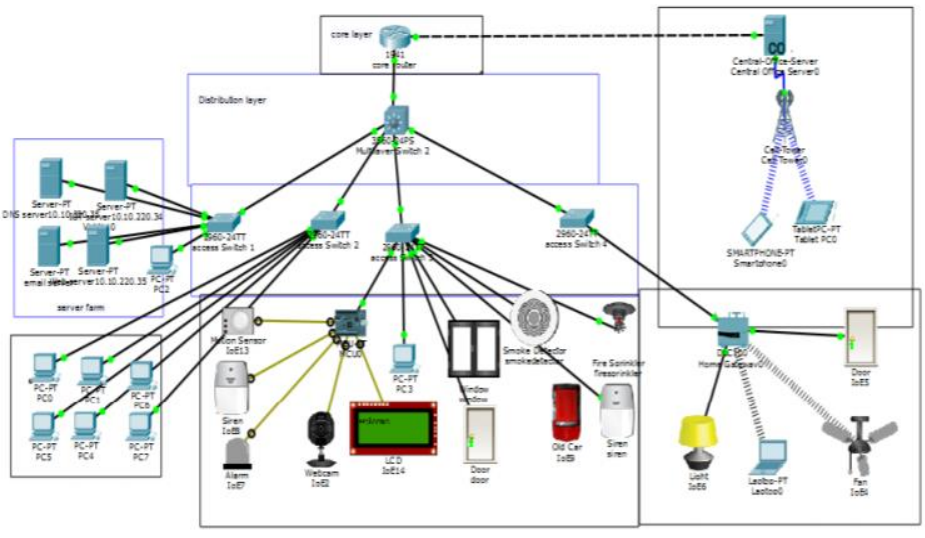
Isa Shemsi
CSE Department, SIT, Symbiosis International University, Pune, India
isa.beriso@sitpune.edu.in /isashemsi@gmial.com

Abstract:-Campus Network (CN) is a set of Virtual Local Area network (VLAN), which covers the entire university. It provide difference service such as connect user to internet, data sharing among user, accessing different web service for different functionalities. As Campus Network (CN) provides students, teachers, and different university member for different application, to sustain different activities in the university, so it need to design in advance. To sophisticate the campus network service, this paper proposed Smart Campus Network Design(SCND) by integrating internet of thing device with classically network device in campus network and each smart device for different application must be registered to IOE server and controlled by legitimate user. To design the proposed campus network design, I used cisco packet tracer simulator software.

what is considered as "Things". Till current years, groups of scholars and organizations tried to make clear the definition of IoT. Haller et al. [1] proposed a definition of IoT with "A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business process." To spread the coverage of IoT definition, Sarma et al. [2] defines the "Things" from physical objects to virtual objects which represents as the identities with Internet connectivity. Although IEEE IoT Initiative is proceeding to draft a formal definition of IoT agreements for the defi

In this paper, I define a of Things that indicat registered on IOT ser controlled through web by legitimate user [4].

Volume 2, Issue 11, November– 2017 International Journal of Innovative Science and Research Technology
ISSN No:-2456 –2165



§ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПОТОКОВ ДАННЫХ В БОЛЬШИХ СЕТЯХ (НАУЧНАЯ СТАТЬЯ)

Шахтурин Д. В. Казанский государственный технический университет им. А.Н.Туполева

Аннотация: Рассмотрены понятия фрактал и фрактальная размерность применительно к телекоммуникационным технологиям для моделирования топологии больших сетей. Предложен критерий оптимального согласования топологии большой сети с геометрией объекта, на котором она размещается. Рассмотрено приложение методов фрактальной геометрии для определения задержек сообщений в распределенных в топологическом смысле сетях.

Ключевые слова: фрактальная геометрия, сложная распределенная сеть, основная количественная характеристика сети, критерий согласования, моделирование, задержка сообщений

Существующие тенденции развития современных связанных и распределенных сетей свидетельствуют о необходимости разработки эффективных методов моделирования и оптимизации сетей огромных размеров. Возможности применение к сложным сетям, которым свойственны непрерывное расширение и динамические характеристики, обычных методов моделирования существенно ограничены.

Свойства распределенных в топологическом смысле сетей (компьютерных, телекоммуникационных, сотовых станций и т.д.) существенно зависят от геометрии, способа размещения узлов, расстояния между ними и т.д. Например, с топологических позиций глобальную информационную сеть Интернет можно рассматривать как совокупность большого числа распределенных точек (узлов), взаимодействующих между собой через каналы связи [1, 2].

Информационные и коммуникационные свойства большой совокупности (ансамбля) распределенных объектов качественно и количественно отличаются от аналогичных свойств отдельного объекта [1, 2].

Например, в сетевых структурах со множеством узловых станций появляются совершенно новые свойства, такие как живучесть, надежность, множественность маршрутов доставки сообщений до пользователя, неустойчивость, конфликтность и т.д. Здесь можно привести некоторую физическую аналогию.

Например, свойства газа в сосуде определяются как результат коллективного взаимодействия большого числа (ансамбля) молекул или частиц. При этом изолированная молекула или их малое количество сами по себе вовсе не обнаруживают общих макроскопических свойств, обусловленных только их большой совокупностью и геометрией сосуда (давление, температура, диффузия, объем, ламинарное или неустойчивое турбулентное течение и т.д.) [3-5].

Следовательно, можно утверждать, что свойства этого класса сетей главным образом зависят от геометрии или топологии. При этом одной из фундаментальных характеристик сетей является их топологическая размерность D_c. Поэтому множество свойств сетей зависят от размерности D_c, и эту зависимость можно выразить функциональным соотношением вида "свойство сети = $f(D_c)$". Подсчитав размерность топологии сети D_c, можно количественно выразить, например, системные свойства сети и найти общие информационные закономерности движения потоков данных как функцию $f(D_c)$.

Размерность топологии сложных систем и сетей нельзя вычислить, используя обычную евклидову размерность, имеющую только целочисленные значения. На практике рассматриваемые большие сети, несмотря на их внешне нерегулярную структуру, характеризуются некоторым основополагающим порядком, обусловленным внешними ограничениями и моделью их роста. Данное обстоятельство позволяет использовать метод определения размерности топологии этих сетей, основанный на приложении свойства самоподобия, присущего фракталам.

При этом топология глобальной информационной сети является примером случайного фрактала, поскольку ее малая часть подобна целой. Так, топология сети сотовой связи в масштабе отдельного городского района подобна топологии сети городского масштаба, а топология городской сети – топологии сети регионального масштаба (и т.д., по восходящей иерархии масштабов) [1].

Следовательно, узлы информационной сети можно рассматривать как множество точек, вложенных в пространство. Размерность этой совокупности точек имеет дробную размерность, или фрактальную размерность.

Рассмотрим вопрос о **фрактальности геометрии городских сетей** коммуникаций, имеющих структуру графов. Антропогенная деятельность человека, как показывают проведенные исследования, может иметь фрактальный характер. Это, в частности, относится к транспортной сети метро в Париже [6] и развитию некоторых мегаполисов [7].

Сложность описания больших сетей заключается в том, что интенсивность информационного потока данных изменяется в различных областях зоны обслуживания. Интенсивность информационного потока в отдельных областях может быть оценена при помощи географических и демографических характеристик зоны обслуживания. При этом распределение плотности населения в зоне обслуживания однозначно определяет интенсивность информационного потока. Следовательно, узлы сети сосредоточены в областях с высокой интенсивностью информационного потока (высокой плотностью пользователей) и редки в областях с низкой интенсивностью (низкой плотностью пользователей). Например, для достижения эффективной конфигурации сети мобильной связи основные объекты сети (базовых станций и центров коммуникации) должны быть расположены близко к предполагаемым источникам трафика.

Таким образом, сети коммуникаций (транспортные сети, сети сотовых станций и других телекоммуникаций, сети магазинов, больниц и систем обслуживания населения и т.д.) обуславливают развитие и рост городов, причем с сильной обратной связью. Следовательно, возможно представление коммуникационных сетей города имеющих тесную корреляционную связь с географической геометрией улиц и районов. Геометрия городских застроек непременно обуславливает топологию сетей коммуникаций.

В качестве примера проведено исследование фрактальности транспортной сети города Казани. Использовалась карта города масштабом 1:100000 издания 2006 г. Подсчитывалось число $N(R)$ узлов сети, расположенных внутри окружностей радиусом R с центром в Казанском Кремле, являющимся географическим центром города (рис. 1). При равномерном распределении вдоль прямых линий значение $N(R)$ пропорционально расстоянию R.

Рисунок 36 – Топология транспортной сети г. Казани

Если пространственное распределение узлов сети компактно, то есть если их плотность (число узлов на единицу площади) постоянна, значение N(R) пропорционально R^2.

При фрактальном распределении значение N(R) пропорционально R^{Dc}, где Dc – фрактальная размерность коммуникационной сети. Это означает, что плотность числа узлов уменьшается с возрастанием R.

Таким образом, для окружности радиуса R и площади, пропорциональной R2, плотность ρ(R) есть

$$c(R) \sim N(R) / R^2 \sim R^{Dc-2} .$$

Из (1) следует, что плотность узлов сети при увеличении радиального расстояния убывает по степенному закону. При этом если фрактальная размерность сети меньше евклидовой размерности пространства, вмещающего сеть ($D_0 = 2 < Dc$), то плотность узлов сети асимптотически стремится к нулю с ростом R. Однако плотность узлов сети будет в среднем постоянна, если фрактальная размерность Dc приближается к евклидовой размерности пространства.

Результаты топологических расчетов для г. Казани показаны в полулогарифмическом масштабе на рис. 2 (Rmax = 21 км). Для сравнения на этом же рисунке приведены аналогичные данные для линий железнодорожной транспортной системы Парижа [5].

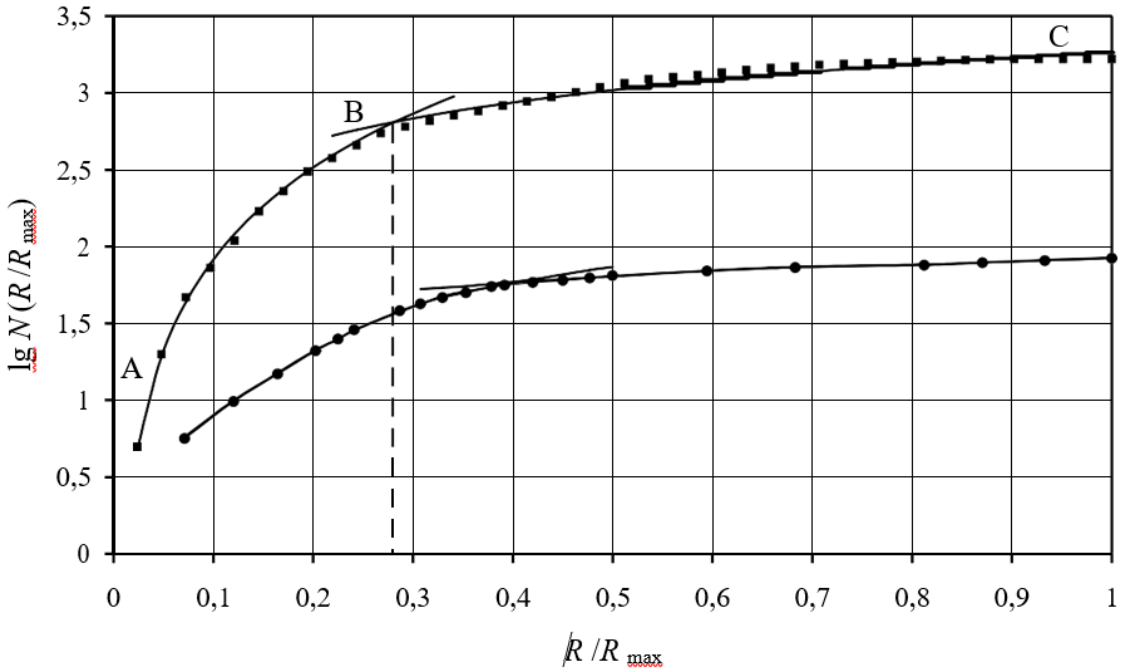


Рис. 37 - Графики зависимости числа узлов $N(R/R_{max})$ транспортной сети г. Казани (▪–▪–▪), числа станций железнодорожной сети г. Парижа (●–●–●) от относительного радиального расстояния R/R_{max} в полулогарифмическом масштабе

Вблизи центра (R ≤ 6 км) плотность узлов сети пропорциональна значению R2 (**рис. 2**, участок АВ). При удалении большем, чем R ≈ 6 км, наблюдается резкий переход к закономерности вида $N(R) \sim R^{0,8}$ (рис. 37, участок ВС). Значение R = 6 км (рис. 2, точка В) примерно соответствует радиусу исторического центра города (**рис. 3б**). Следовательно, при R ≤ 6 км число узлов городской транспортной сети изменяется как квадрат расстояния от центра. Для R > 6 км сеть образует фрактальную размерность около 0,8, плотность узлов при этом определяется уравнением

$$c(R) \sim R^{-1,2}$$

Сравнение графиков транспортных сетей г. Казани и г. Парижа (рис. 37) свидетельствует о существовании общих глобальных закономерностей развития топологии сетей коммуникаций крупных мегаполисов.

При этом одной из фундаментальных характеристик подобных сетей является их топологическая размерность D_c.

Применение функции плотности узлов дает нужную методику моделирования и синтеза больших сетей. Дополнительное преимущество этого подхода состоит в том, что число узлов N(R) монотонно спадает в радиальном направлении от некоторой центральной точки.

Кроме того, хорошо согласуется с эмпирическим законом Зипфа о распределении населения вокруг городских центров [8]. Можно предположить, что сети, узлы которых распределены случайным образом, согласно соответствующей функции плотности с одинаковой фрактальной размерностью, хотя и различны, но качественно подобны и, следовательно, обладают некоторыми общими глобальными свойствами.

Таким образом, проведенное исследование позволило выявить несколько важных характеристик фрактального моделирования больших сетей узлы, которых распределены случайным образом в соответствии функции плотности. Во-первых, по мере удаления от начальной точки плотность узлов монотонно убывает, поскольку топологическая размерность DE больше фрактальной размерности D_c.

Во-вторых, по мере приближения фрактальной размерности к топологической ($Dc \rightarrow D_E$) узлы сети будут равномерно покрывать круг соответствующим образом заданного радиуса, тогда как за пределами этого радиуса не окажется ни одного узла.

Поэтому для полной характеристики транспортной сети г. Казани были определены обобщенные фрактальные размерности Dq.

$$D_q = \lim_{e \to 0} \frac{1}{q-1} \cdot \frac{\ln \sum_{k}^{N} p_k^q}{\ln e}, \quad -\infty \le q \le \infty, \tag{4}$$

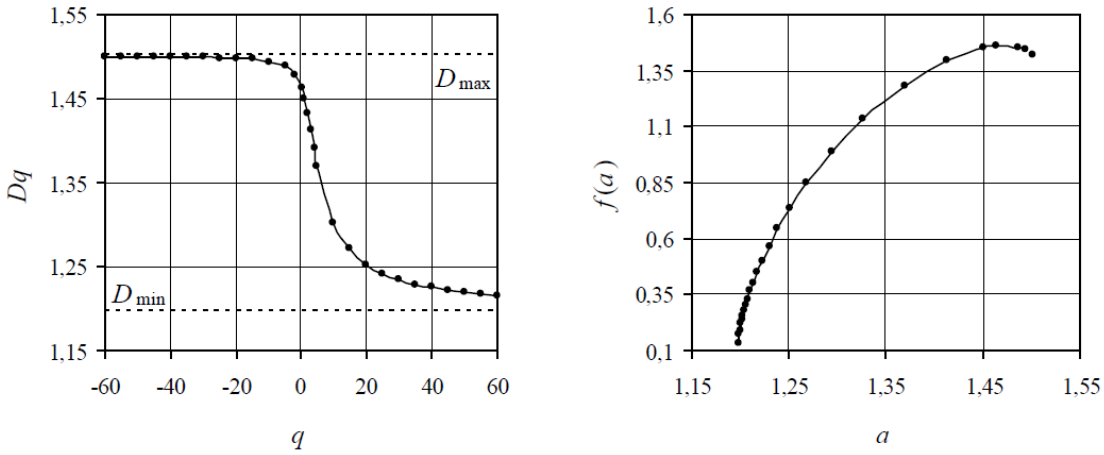
и мультифрактальный спектр $f(a)$

$$f(a(q)) = \tau(q) + q \cdot a(q), \tag{5}$$

где $\tau(q) = -\lim_{\varepsilon \to 0} \frac{\ln \sum_{k}^{N} p_k^q}{\ln \varepsilon}$ – последовательность показателей массы;

$a(q) = -\frac{d}{dq}\tau(q) = \lim_{\varepsilon \to 0} \frac{\sum_{k}^{N} p_k^q \ln p_k}{\ln \varepsilon \sum_{k}^{N} p_k^q}$ – показатель массы Липшица-Гельдера.

На рис. 38 приведены обобщенные размерности «транспортной» сети г. Казань:



а) *б)*

Рисунок 38 - Обобщенные размерности (а) и мультифрактальный спектр (б) транспортной сети г. Казани

ПРОДОЛЖЕНИЕ СТАТЬИ

ДОПОЛНИТЕЛЬНЫЙ МАТЕРИАЛ

Наукові праці ОНАЗ ім. О.С. Попова, 2010, № 1

УДК 621.39

Тихонов В.И.
Тіхонов В.І.
Tikhonov V.I.

ФРАКТАЛЬНАЯ ТОПОЛОГИЧЕСКАЯ МОДЕЛЬ ОТКРЫТОЙ ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ

Аннотация. *Обоснованы принципы построения топологических моделей открытых телекоммуникационных сетей на основе теоретико-множественной концепции иерархических классов. Сформулированы аксиомы топологического пространства для типового фрактального сегмента открытой сети.*

Таким образом, системные свойства (живучесть, надежность, время доставки) больших сетей существенным образом зависят от топологии. При этом основной характеристикой сетей является их фрактальная размерность, удобная для моделирования и описания больших распределенных в топологическом смысле сетей. Эти положения показаны на примере транспортной сети г. Казани, имеющей структуру фрактальных графов.

Используя, предложенный фрактальный критерий покрытия объекта сложной сетью определена характерная особенность такого вида сетей: большое или избыточное количество узлов сети не гарантирует оптимальную распределенность узлов на топологии сложного объекта без учета фрактальности его геометрии. Последнее позволило выявить основные свойства фрактального описания топологии сложных сетей.

Полученные результаты позволяют использовать фрактальную размерность как числовую характеристику для анализа информационных свойств сетей.

Применение методов фрактальной геометрии для моделирования задержек в больших сетях дает возможность изучать закономерности движения информационных потоков данных с целью предсказания развития и повышения эффективности использования этих сетей.

По результатам проведенных исследований можно сделать вывод о том, что предложенный подход, основанный на методах фрактальной геометрии, позволяет осуществлять количественное сравнение, анализ и синтез сложных сетей.

АНАЛИЗ СЕТЕВОЙ ИНФРАСТРУКТУРЫ ПРЕДПРИЯТИЯ

Щербаков С.В., Коннов А.Л., канд. техн. наук, Оренбургский государственный университет

Сетевая инфраструктура представляет собой совокупность различного оборудования, программного обеспечения, которая формирует особую среду для эффективного процесса обмена данными, а также для работы бизнес- приложений.

В настоящее время будущее каждой компании напрямую связано с возможностью её оперативного реагирования на тенденции развития рынка. Именно поэтому современная компания обязана функционировать как хорошо отлаженный механизм. **Она должна быть управляемой**.

Степень такой управляемости организации зависит от того, как хорошо в ней поставлен сбор, обработка и хранение информации, необходимой для принятия решения. Если информационная система (ИС) организована должным образом, то компания в состоянии решать поставленные задачи. В основе такой **ИС лежит сетевая инфраструктура**.

Когда в организации установлено более одного персонального компьютера (ПК), которые не объединены в одну общую локальную сеть, это приводит к возникновению многочисленных проблем. Все они связаны с поиском, восстановлением и передачей информации, отсутствием возможности пользоваться данными дома либо во время командировок, совместной работой над различной документацией, подключением к сети интернет при помощи периферийного оборудования.

Это и многое другое существенно уменьшает эффективность работы любой организации. Но правильная организация и эксплуатация объектов сетевой инфраструктуры легко решает данные проблемы. Именно поэтому любой руководитель компании обязан обращать на это своё внимание.

Сетевая инфраструктура предприятия представляет собой комплекс следующих устройств. В локальную сеть входит программное обеспечение аппаратных средств, которые объединены в одну общую платформу. К активному оборудованию относятся коммутаторы, маршрутизаторы и конверторы интерфейсов.

Пассивные устройства представляют собой различные монтажные шкафы, кабели, коммутационные панели, кабельные каналы. Периферийное оборудование и компьютеры включает рабочие станции, копиры, серверы, сканеры и принтеры. Самое основное место в СИ занимает локальная вычислительная сеть (ЛВС). С её помощью осуществляется объединение вычислительных и локальных ресурсов с возможностью организации раздельного доступа к ним. Благодаря локальной сети осуществляется связь всех компьютерных установок. Она можетбыть как проводной или беспроводной, так и комбинированной.

Такая сеть может располагаться в одном помещении на различных этажах, в разных помещениях, а также на большом расстоянии друг от друга. Для связи всех её пользователей используются специальные устройства – коммутаторы (свитчи) и маршрутизаторы. Все возможности локальной сети могут использоваться одновременно, независимо от того, где находятся рабочие места. С её помощью открывается моментальный доступ к нужной информации, возможность обмениваться данными и мультимедийными носителями, а также подсоединяться к существующей на предприятии сети интернет.

Именно поэтому внедрение сетевой инфраструктуры очень важно для любой компании. Надёжность и производительность локальной сети, независимо от того, будет она кабельной или беспроводной, зависит ещё и от того, какие в ней применяются технологии, активное оборудование и сетевое программное обеспечение. Если вы хотите правильно и эффективно спроектировать такую сеть, то обязательно **производить анализ информационных потоков** вашей организации, при этом учитывая перспективу развития самой инфраструктуры. Для каждой организации правильное построение СИ является залогом безопасного и эффективного использования информации. После того как была сформирована база такой системы, происходит внедрение сетевых сервисов.

С их помощью обеспечивается надёжность и доступность всех ресурсов компании. Эффективность ведения бизнеса определяется доступностью пользовательских сервисов, систем видео и голосовой связи, а также систем унифицированных коммуникаций. Немаловажным значением обладает и защита сетевой инфраструктуры.

Ведь при возникновении чрезвычайной ситуации велика вероятность потери не только самого оборудования, но и информации. Многие такие аварии и катастрофы угрожают целостности самого бизнеса. Они являются предсказуемыми с определённым процентом вероятности. Такие события могут быть как природные (наводнение, землетрясение и т.п.), так и механические (разрыв водопроводных коммуникаций, выход из строя жёсткого диска и другие).

Отсутствие специальной программы, с помощью которой осуществляется восстановление работоспособности сетевой инфраструктуры, ставит под большую угрозу дальнейшую деятельность организации. Особенно это касается предпринимателей малого и среднего бизнеса. Во многих компаниях такое планирование чрезвычайных мероприятий фокусируется уже на стадии ИТ. Такая программа сохранения непрерывности любых бизнес-операций, а также послеаварийного восстановления способна стать, пожалуй, самым ценным и эффективным вкладом отдела ИТ в успешное процветание самой компании. Существует так называемое «горячее» запасное оборудование, которое готово моментально включиться в работу при аварийной ситуации.

К примеру, вы можете дублировать самые важные данные занесением их на специально предназначенную базу. А вот «холодное» – представляет собой устройства, которые можно оперативно подготовить к выполнению тех либо иных задач. Это может быть набор не подключённых серверов, на которых установлено всё необходимое для работы программное обеспечение сетевой инфраструктуры. Таким образом, вы сможете без проблем и очень быстро переключиться с неисправного оборудования и продолжить работу.

К примеру, если в вашей фирме имеется совсем небольшой отдел IT, а именно: несколько серверов, немного больше ПК, интернет, сетевое оборудование и т.п. Все эти устройства довольно часто обслуживает один администратор. Его рабочий день должен проходить следующим образом:

Вначале необходимо проверить работу серверов, убедиться в исправной работе интернета, почты и прочих приложений, произвести пробное подключение к каждому из серверов, проверить свободное пространство, ОЗУ и другие мощности, убедиться в работе сетевой оргтехники и проверить задание резервного копирования. Многие могут спросить: затем такая постоянная ежедневная проверка? Тут всё просто.

Если вы вовремя не обнаружите ошибку либо проблему в работе оборудования, то это может привести к необратимому процессу, который способен стать настоящей катастрофой для вашей фирмы. К примеру, если внезапно закончится свободное место для резервного копирования данных, то и восстанавливать вам нечего будет потом. Но не каждый администратор добросовестно относится к своей работе и может просто забыть произвести плановую проверку.

В таком случае руководителям компании помогут специальные системы мониторинга, которые будут выполнять всю работу автоматически. Вам достаточно только указать, что и когда нужно проверять. Что может мониторить такая система. Это рабочие станции, сервера на базе различных операционных систем, доступность сайтов, серверные и клиентские приложения, а также службы, принтеры, сканеры и прочее сетевое оборудование, отправка уведомления и отчёты на e-mail адрес либо в sms, построение графиков и многое другое.

Очень важным вопросом также является конфигурирование и поддержка сетевой инфраструктуры. Ведь с каждым годом мощность серверов и скоростей увеличивается. Это влечёт за собой необходимость своевременного и профессионального обслуживания, а также потребность в оперативном решении текущих и перспективных задач. Чем выше требования предъявляются к СИ, тем больше нужно использовать различного эффективного и функционального оборудования. Кроме этого требуются более глубокие познания и опыт построения сетей на таких устройствах. Это следует помнить, знать и своевременно решать.

Также на предприятии периодически следует проводить и аудит сетевой инфраструктуры. Данный комплекс мероприятий направлен на определение состояния, в котором в данный момент находится СИ организации, проводится поиск самых уязвимых мест. По результатам такого аудита составляется специальный отчёт, в котором будет отображено текущее состояние сетевой инфраструктуры и предложена организация эксплуатации сетевой инфраструктуры.

Данная проверка нужна в следующих случаях. Перед модернизацией и после нее, для определения истинной причины возникшей проблемы, для оценки качества и эффективности сервиса, во время передачи функций администрирования.

Обслуживание и создание эффективной ИТ инфраструктуры организации – это необходимое условие успешного ведения бизнеса. Сетевая инфраструктура становится с каждым днём сложнее. В неё входит огромное количество оборудования и программного обеспечения, обслуживать которое должен довольно большой штат сотрудников. Именно поэтому многие фирмы начинают задумываться: производить им поддержку самостоятельно либо же воспользоваться внешним обслуживанием аутсорсингом.

Аутсорсинг сетевой инфраструктуры – это превосходный способ сохранения и страхования ваших инвестиций в IT. Данная процедура позволяет снизить расходы, которые связаны с развитием и функционированием корпоративной сети, а также существенно повысить качество IT услуги. Преимущества перехода на аутсорсинг заключаются в следующем. Наблюдается сокращение издержек на сопровождение СИ. В данном случае поставщик услуг IT аутсорсинга берёт на себя второстепенные обязательства. Сам же управляющий персонал может сконцентрировать своё внимание на решении других задач, которые более важны.

Таким образом, компания добивается конкурентного преимущества. Также имеет место снижение рисков в проектах. Расширение и модернизация СИ, внедрение бизнес-приложений и IT сервисов – всё это выполняют узкоспециализированные профессионалы. Таким образом удаётся минимизировать вероятность ошибок, которые довольно часто могут возникать у специалистов более широкого профиля. Увеличение качества самого обслуживания.

Передача всех обязанностей, связанных с технической поддержкой и сопровождением, одному поставщику даёт возможность стандартизировать обслуживание и гарантировать высокое качество работы.

Особенности проектирования и анализа IoT-систем

Мунистер Вячеслав Денисович

Проведенный анализ сетевой инфраструктуры применителен к абсолютно «классическому» представлению сетевой инфраструктуры – в виде локально-вычислительной сети предприятия. И практически не отражает объект исследовательско-практической работы нашего интегрированного курса несмотря на абсолютно тождественные представления к структуре анализа. Как было замечено ранее, сетевая инфраструктура ежегодно усложняется как с точки зрения аппаратного обеспечения, так и программной составляющей. Безусловно, интеграция концепции IoT вносит и свою проблематику в общий анализ с точки зрения организации.

В отчёте Национального разведывательного совета США (англ. National Intelligence Council) 2008 года «Интернет вещей» фигурирует как одна из шести **подрывных технологий**, указывается, что повсеместное и незаметное для потребителей превращение в интернет-узлы таких распространённых вещей, как товарная упаковка, мебель, бумажные документы, может заметно **повысить риски** в сфере национальной **информационной безопасности.**

Согласно указанному выше документу, к 2025 г. узлами IoT, т. е. потенциальными целями хакеров, смогут стать все окружающие нас предметы.

Безусловно, Internet of Things (Internet of Everything) необходимо рассматривать как автохтонный, двунаправленно-интегрированный в общую структуру, и, при этом, в какой-то степени, автономный элемент в составе классической сетевой инфраструктуры. Предлагаемое мной рассуждение ориентировано на обозначение аспектов проблематики в области информационной безопасности интернета вещей в прикладных вопросах, на примерах из элементов практикума. Наиболее общие вопросы информационной безопасности будут отражены чуть позже, в более строго документальном и отчасти, регламентированном виде.

Проблематику подсистемы IoT необходимо рассматривать с наиболее общей категории, которой, несомненно, является определение «части и целого». И это совершенно несложно, определяя неделимым и наименьшим компонентом IoT датчик или актуатор.

Таким образом, первой **дилеммой**, связанной с проблематикой выбора, как следствие, и безопасности системы, является **уровень релевантности выбираемого устройства.**

Релевантность в информационной науке и информационном поиске означает степень соответствия найденного документа или набора документов информационным нуждам пользователя.

В общем виде, в нашем случае, это вся та же степень соответствия желаемого к действительному. И это - **качественная шкала**.

Да, безусловно, однозначно есть дискретно обособленные категории – «подходит» или «не подходит» заданный компонент для решения конкретной задачи. Но этого недостаточно. Термин релевантности должен захватывать больший объем предъявляемых критериев к датчику или актуатору. А также, в целом, позже, на конечном этапе анализа, вносить свою роль в оценку всей сетевой инфраструктуры.

И данная оценка не может быть плановой, односложной. Мы должны понимать, что конечное устройство (на данном этапе размышления мы говорим именно про него) – это не просто точка сбора информации, а также и концентратор воздействия окружающей среды (и это влияние может также быть двунаправленным: например – исполнительный механизм открывания двери своим рабочим состоянием/бездействием, может влиять на множество разнообразных параметрических показателей, но также есть определенные факторы, которые могут повлиять на поведение (степень возможности реализации того или иного перехода в обозначенное на изначальном уровне состояние этого же устройства). Нужно не просто дать ответ на вопрос – возможна ли эксплуатация? А и ответить на факт полного (или достаточного) соответствия этого актуатора к выбранной для него задачи; и степени конгруэнтности.

В таком случае правильно предложить выборку и произвести оценку методом ранговой оценки или ранжированием (еще один уместный термин из теории информационного поиска). И чаще всего, на этот, первом уровне нашего анализа, все сводится к изучению электронных компонентов. Для этого обязательно потребуются дополнительные знания, вводные условия обслуживания (эксплуатации).

Например: нам нужно создать «умное» устройство, основная функция которого: реакция на обнаружение (перекрытие какого-нибудь «логического вентиля», назовем эту неназванную абстрактную конструкцию так.

Сразу оговорим – в требованиях прямо не указываются какая-то дополнительная функциональная нагрузка на предмет, как и желательное здесь, уточнение семантики фразы «реакция на обнаружение». Такое и в реальных условиях, увы, редко оговаривается в понятном для всех виде. Такж

Тогда, на помощь приходят специализированные технические издания, интернет-ресурсы, позволяющие определить класс устройства. В нашем случае это детектор.

Детектор (лат. detector — открыватель, обнаружитель) — техническое средство или вещество, которое указывает на наличие определенного свойства объекта измерения при превышении порогового значения соответствующей величиной.

Детектор — то же, что датчик, **первичный преобразователь**, элемент измерительного, сигнального, регулирующего или управляющего устройства системы, преобразующий контролируемую величину в удобный для использования сигнал.

Но правильно мы размышляем, определяя сходу тип устройства? Есть сомнения. **Ведь реакцию на обнаружение** могут реализовать и такие устройства как:

Датчик движения (англ. motion sensor, сенсор движения) — **сигнализатор**, фиксирующий перемещение объектов и используемый для контроля за окружающей обстановкой или автоматического запуска требуемых действий в ответ на перемещение объектов.

Более чувствительные датчики движения называют также датчиком присутствия (англ. presence sensor или occupancy sensor).

Также, **большую степень релевантности** в подходящих условиях эксплуатации может обеспечить PIR-sensor. PIR-sensor переводится с английского как Pyroelectric (Passive) InfraRed sensor - пироэлектрический (пассивный) инфракрасный сенсор.

Пироэлектричество — это свойство генерировать определенное электрическое поле при облучении материала инфракрасными (тепловыми) лучами. Да и обычный фоторезистор вполне можно приспособить для той же цели.

Тогда, как определиться с выбором? Все зависит от вкладываемой «прочности» будущего проекта. Часть устройств, представленных выше (на выбор) имеют определенное времени наработки на отказ, который, однако, на практике, трудно отследить (он может и не наступит за весь «жизненный цикл» системы), часть – и вовсе не имеют (тот же фоторезистор).

Второй фактор – условия эксплуатации (закрытая обитаемая среда, или неблагоприятные условия, требующие помещения в корпус и т.д., точность, целевое предназначение (сфера услуг, производственная сфера).

Третий фактор – степень интеграции. Существуют как готовые встроенные решения, так и простые полупроводниковые приборы.

Четвертый фактор – определение объекта «отслеживания». Что является спусковым крючком для перехода разрабатываемого устройства, который позднее станет частью IoT-системы из одного состояния в другое? Это тень – образованная движением механизма (который поддается логике с точки зрения изменения), или это интенсивность света («логический вентиль» открывается только при конкретном показателе светового потока?

То есть все сводится к инженерному анализу. К которому прибавляется и технический (привожу пример рассуждений по схожей тематике):

«Сначала необходимо выбрать эксплуатационное напряжение и степень защиты. Если датчик будет монтироваться снаружи помещения, то его класс защиты должен быть не менее, чем IP 44. Это означает защиту датчика от попадания посторонних предметов внутрь размером больше 1 мм, защиту от влаги. Далее следует обратить внимание на режим эксплуатации по температуре. Нужно выбирать модели, которые способны работать при температуре в вашем регионе.

Мощность устройства также играет большую роль. Лучше выбрать датчики с запасом по мощности. Некоторые модели оснащены регулятором порога срабатывания. То есть, настраивается чувствительность датчика. Например, при выпадении снега лучше снизить чувствительность, так как снег отражает свет, который может повлиять на срабатывание датчика. Пределы настройки чувствительности также бывают разными. Время задержки включения датчика также может регулироваться. Такая регулировка необходима для защиты от ложных срабатываний. Например, в темное время на чувствительный элемент может на короткое время попасть свет от случайного источника (фар автомобиля)»

Вторым уровнем данного анализа является аспект избыточности оконечных устройств, а также проблематика беспрерывного дуплексного соединения и выделения несущей частоты. Данные вопросы относят нас к **физике процесса** – к топологическим особенностям расположения датчиков, IoT-концентраторов, коммутаторов, маршрутизаторов и иного оборудования, к технологиям физического уровня передачи данных.

Здесь необходимо произвести факт выявления т.н. «узких мест», связанных с ограничениями (задержками деинкапсуляции пакетов данных, совместимость протоколов/физических ограничений (или неоднородность среды). Ведь может оказаться, что приобретенный, или созданный вами датчик, который должен предоставлять элемент структуры, приближенной к системе реального времени (многие сложные технологические процессы, да и просто компьютеризированные или автоматизированные системы «высокого уровня» могут быть этого класса) подведут по ряду причин – и вовсе не антропогенного происхождения.

Третий уровень анализа сводится к выявлению проблем умышленного/произвольного/случайного воздействия как на сеть, так и на ее сегменты и так далее. Скорей всего, ваша сеть (система) будет представлена в смешанном виде – часть обмена будет осуществляться проводным образом, часть беспроводным. Необходимо принимать за внимание не только информационную безопасность каналов связи, но и места «конвертации» передающих средств. Выход из строя беспроводного модуля связи может произойти по ряду неочевидных причин. Для умышленного негативного воздействия – это будут наиболее уязвимые элементы. Безусловно, нужно отработать сценарии переключения «контекста». Возможно, на программно-аппаратном уровне, еще на этапе проектирования, Вам понадобится внедрение полносвязной топологии.

Затем, наконец, вам необходимо выявить и «зону неопределенности»— множество вариантов развития реакции всей системы, каждый из которых оптимален при некотором реально возможном сочетании внешних условий. В таком случае вы можете действительно говорить о готовности рассматривать разработанную вами IoT-систему с основных догм информационной безопасности, т.е, как минимум, определять риски, угрозы, внедрять модели безопасностей. В противном случае вашей системе придется более долгий процесс доработки.

Но и этого описанных выше требований будет также недостаточно для полноценного анализа разработанной системы. Информационная безопасность сводится не только к средствам информатизации, но и к оценке объекта внедрения (предприятия).

В связи с этим предлагается изучить посредством интер-отклика содержание учебного пособия ИрГУПС «Оценка информационных рисков предприятия» Н. И. Глухова.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4

ГЛАВА 1. Место и роль системы оценки рискозащищенности информационных активов в системе управления деятельностью предприятия 6
Тема № 1. Актуальность рискозащищенности предприятия 6
Тема № 2. Исследование сущности информационных активов хозяйствующих субъектов 9
Тема № 3. Место системы защиты информационных активов в системе управления деятельностью предприятия 21
Тема № 4. Анализ основных угроз информационным активам хозяйствующих субъектов 29
Контрольные вопросы 39

ГЛАВА 2. Методические подходы к оценке информационных рисков хозяйствующих субъектов 40
Тема № 5. Анализ современных методических подходов к обеспечению защищенности информационных активов хозяйствующих субъектов 40
Тема № 6. Исследование особенностей организационного направления в сфере обеспечения рискозащищенности информационных активов предприятия 44
Тема № 7. Сравнительный анализ методических подходов и инструментария для оценки информационных рисков хозяйствующих субъектов 66
Контрольные вопросы 80

ГЛАВА 3. Разработка методики оценки информационных рисков хозяйствующего субъекта 82
Тема № 8. Разработка методики оценки информационных рисков хозяйствующих субъектов 82
Тема № 9. Методические подходы к оценке затрат на обеспечение защищенности информационных активов хозяйствующего субъекта 105
Контрольные вопросы 114

ЗАКЛЮЧЕНИЕ .....

Библиографический список .....

ПРИЛОЖЕНИЯ .....

ПРОБЛЕМЫ БЕЗОПАСНОСТИ ИНТЕРНЕТ ВЕЩЕЙ: ОБЗОР

Соколов М. Н., Смолянинова К. А., МГТУ им. Н.Э. Баумана, Москва.
Якушева Н. А., University of Udine, Udine (Italy).

Одной из сложных задач в развитии концепции Интернет вещей (IoT) во многих приложениях являются сложные проблемы обеспечения информационной безопасности в широком спектре защиты от угроз злоумышленника. Эти проблемы являются особенно актуальными, поскольку прогнозируется рост потребности пользователей в IoT.

Для анализа поставленной в работе задачи принята одна из наиболее распространенных архитектур IoT, состоящая из трех уровней – уровень восприятии (perception), сетевой уровень и прикладной уровень. Для каждого из этих уровней приведены основные проблемы обеспечения ИБ. Отмечаются основные причины сложности обеспечения ИБ на сетевом уровне – гетерогенный характер структуры (многообразие вещей, разные технологии сетей) и большое число объектов. IoT принимает информацию от большого числа устройств, собирает большие данные различных форматов от множества источников с неоднородными характеристиками.

Результатом являются отказы DoS из-за перегрузок в сети, программные ошибки из-за сложности отладки в реальном масштабе времени с помощью имитатора внешней нагрузки.

Введение.

IoT является новым шагом в технологическом прогрессе. Интернет вещей позволяет людям и «вещам» соединиться в любое время и в любом месте, используя различные сети связи. В документах вместо термина «вещь» («things») применяют такие термины – объект («objects»), узел («node»), прибор («device») и др. Основными компонентами IoT являются всепроникающие сенсорные сети USN (Ubiquitious Sensor Networks) и радио-частотная идентификация RFID (Radio Frequency IDentification). Вещью в RFID является RFID- метка (RFID-тег), а в USN - сенсорный датчик или группа датчиков. Сетевые структуры сетей USN построены на базе протокола IPv.6 – 6LoWPAN (Low energy IPv6 based Wireless Personal Area Networks protocol).

Уже сегодня можно наблюдать, как через Интернет между собой связаны различные устройства, работающие без участия человека – системы управления освещением, системы управления, автоматические системы полива, датчики пожарной и охраной сигнализации, светофоры и др. Одной из главных проблем IoT является обеспечение информационной безопасности (ИБ).

В настоящей работе рассматриваются проблемы безопасности Интернет вещей только для одной из ее компонент – **сенсорных сетей**. Ежегодный прирост рынка сенсорных сетей порядка 7,8%.

Покажем различие сенсорных сетей и IoT. **Сенсорные сети** используются для конкретных приложений, а IoT должен поддерживать различные виды приложений и может рассматриваться как сенсорная сеть общего назначения.

Примерами приложений сенсорных сетей в РФ могут быть выполненные работы институтом точной механики и вычислительной техники им. С.А. Лебедева РАН – система аварийной связи для горноспасателей, система для решения комплекса задач по обеспечению безопасности промышленных объектов и сооружений г. Москвы и др.

Все многочисленные приложения IoT можно объединить в три группы – индустриальный или промышленный (industry), окружающей среды (environment), общественный (society). Настоящая работа посвящена анализу проблем обеспечения информационной безопасности (ИБ) IoT.

Для решения поставленной задачи анализу подлежат.

1. **Многоуровневая структура IoT.**
2. **Проблемы обеспечения безопасности** на каждом из уровней принятой структуры IoT.
3. **Некоторые исследования обеспечения** информационной безопасности IoT.

Многоуровневая структура IoT. Для IoT определены три основные характеристики - комплексные знания (в результате получения информации об объекте, в любом месте и в любое время), надежная передача (с помощью протоколов связи, маршрутизации, шифрования, сетевой безопасности, с высокой точностью и реального времени), интеллектуальная обработка (с учетом множества вычислений, нечеткого опознания и других технологий для анализа и обработки Big Data и получения необходимых данных.

В соответствии с этими характеристиками структура IoT может быть разделена на три уровня – уровень восприятии (perception), сетевой уровень и прикладной уровень. Задача уровня восприятия получить надежное считывание с сенсоров, RFID-меток. Сетевой уровень обеспечивает повсеместный доступ, передачу информации, обработку, хранение. Он состоит из уровня доступа (мобильные сети связи), и основного уровня обмена (Интернет, сети следующего поколения NGN, виртуальные частные сети).

Большинство сенсорных сетей используют беспроводные сети связи: беспроводные персональные сети (WPAN) (например, Bluetooth), беспроводные локальные сети (WLAN) (например, Wi-Fi), беспроводные городские сети (WMAN) (например, WiMAX), беспроводные глобальные сети (WWAN) (например, 2G, 3G и 4G сети), спутниковую сеть (например, GPS). Сенсорные сети в IoT используют протоколы связи на основе IP (например, IPv6).

Прикладной уровень анализирует и обрабатывает принятую информацию для принятия правильного решения и контроля за управлением, приложениями и услугами. На прикладном уровне выполняются функции по сбору и хранению данных, по обеспечению эффективности энергообеспечения и логистики и др.

Проблемы информационной безопасности на уровнях структуры IoT. Следует отметить, что в некоторых работах рассматривается более, чем трехуровневая архитектуре IoT. Также принята пятиуровневая архитектура IoT включающая, например, промежуточный уровень (Middleware) между сетевым и прикладным уровнем. Этот уровень выполняет функцию обработки сообщений информации взаимодействующих однотипных сенсорных датчиков.

В настоящей работе ограничимся анализом проблем информационной IoT на каждом из трех уровней – уровне восприятия, сетевом и прикладном уровнях.

Проблемы ИБ на уровне восприятия. Основная проблема безопасности на уровне восприятия состоит в физической безопасности приборов восприятия и безопасность сбора информации.

Большинство узлов восприятия, для которых характерно развертывание в необслуживаемой людьми среде при отсутствии стандартов, разнообразие, простота, ограничение энергообеспечения и слабая способность к защите безопасности.

Поэтому IoT **не может обеспечить унифицированную систему** защиты безопасности и является уязвимой к угрозам злоумышленника. Так как беспроводная сенсорная сеть на уровне восприятия является источником информации, то ИБ на этом уровне важна.

Проблемы безопасности на этом уровне включает физической захват сенсорных узлов, захват узла шлюза, утечка информации сенсора, угрозы целостности данных, истощение энергообеспечения, угрозы перегрузки, атаки типа DoS (отказ в обслуживании), угрозы маршрутизации установлением в сеть нелегитимных сенсоров, и угрозы копирования узла.

Проблемы ИБ на сетевом уровне. Угрозы ИБ существующих сетей связи распространяются и на IoT, который построен на них. Это относится к несанкционированному доступу, перехвату данных, конфиденциальности, целостности, атаках типа человек посредине, Dos-атакам (отказ в обслуживании), вирусам, сетевым червям, руткитам и др. Кроме того, существуют межсетевые проблемы аутентификации, которые могут быть причиной атак DoS.

В IoT стоят более сложные проблемы обеспечения безопасности по сравнению с теми, с которыми сталкивались ранее. Это вызвано двумя причинами – **гетерогенный характер структуры** (многообразие вещей, разные технологии сетей в соединении) и большим числом объектов.

IoT принимает информацию от большого числа устройств, собирает большой массив данных различных форматов от множества источников с неоднородными характеристиками. В результате этого на сетевом уровне имеют место более **сложные проблемы безопасности**. К ним относятся возможные проблемы масштабируемости сети, вызванные малопредсказуемым объемом передачи данных от большого числа узлов, и приводящие к возможности осуществления атак DoS, DDoS.

Отдельное внимание уделяется уязвимостям программного обеспечения (software vulnerabilities), приводящим к нарушению ИБ после внедрения.

Причинами программной уязвимости могут являться неизбежные ошибки разработчиков сложного многослойного программного обеспечения (ПО), ошибки ядра программы, неполнота обработки исключений, применение незащищенного кода. необработанных массивов с возможностью их переполнения злоумышленником, ошибки в обработке Big Data, ошибки БД, отсутствие должной индексации или закрепления запросов БД, web-уязвимости, недостаточная производительность или масштабируемость ПО, ошибки распределенной работы приложений, а также виртуальных платформ и облаков. Следует отметить сложность ПО в IoT, вызванную большим разнообразием используемых аппаратных платформ и операционных систем.

Для **проектирования ПО** необходимо эмулировать поведение приборов IoT, т.е. **создать имитатор внешней среды для серверов**. По причине ограничений в приборах (энергообеспечение, производительность процессора, память) в IoT стоит сложная задача избежать сильного расхождения между эмулятором и датчиком. Также для «отгрузки» отлаженного рабочего релиза IoT приложения, необходимо провести полноценное тестирование, включая нагрузочное тестирование, тестирование производительности, комплексное тестирование взаимодействия модулей.

Другой причиной программной уязвимости могут являться бэкдоры (backdoor, back door - черный ход) - это участки кода, внесенные разработчиком, для последующей возможности использования для просмотра данных, а в случае ОС удаленного управления компьютером.

Бэкдором могут быть как бы случайные ошибки в коде, которые при определенном подборе констант или сочетании клавиш, или действиях в приложении могут давать доступ к каким-либо данным. Бэкдоры также устанавливаются и на оборудование производителями с целью управления или тестирования. Однако этот «черный ход» может быть обнаружен злоумышленником и использован им.

Проблемы ИБ на прикладном уровне. Широкое применение IoT является результатом интеграции компьютерной технологии, технологии связи и различных областей промышленной отрасли.

Кроме нарушение информационной безопасности традиционных сетей связи (в результате угроз повтора, подслушивания, искажения информации, раскрытия информации и др.) приложения IoT сталкиваются с дополнительными проблемами безопасности на прикладном уровне - при использовании облачных вычислениях, обработке информации, обеспечении прав на интеллектуальную собственность, защите приватности и др.

Зарубежные специалисты уделяют большое внимание научным и экспериментальным исследованиям в обеспечении информационной безопасности IoT. Например, в работе [Массачутесского технологического института] показано, что **наибольший риск безопасности возможен на нижнем уровне архитектуры - на уровне восприятия**. При этом отмечается, что некоторым угрозам безопасности на других уровнях архитектуры IoT так же характерен высокий уровень риска

Выводы.

1. Стремительное развитие за последние 2-3 года в практическом плане концепции Интернета вещей вызвано широким распространением беспроводных технологий и межмашинного обмена, развитием технологии облачных вычислений и началом перехода на IPv6. **Однако использование IoT во многих областях ограничено сложными проблемами в части обеспечения ИБ.**

2. Необходимо продолжить эти работы в плане анализа предложений специалистов по решению проблем безопасности в IoT для использования в РФ.

МЕТОДИКА ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Аннотация

Целью данной работы является определение и оценка рисков информационной безопасности для типовой распределенной информационной системы телекоммуникационного предприятия, расположенной в пределах трех контролируемых зон.

Основной акцент при обеспечении информационной безопасности в рассматриваемой информационной системе, делается на минимизацию ущерба от угроз безопасности, направленных на целостность и доступность программно-аппаратного комплекса информационной системы, а не на конфиденциальность информационных ресурсов, обрабатываемых с их помощью. В рамках исследования были рассмотрены международные и национальные стандарты в сфере защиты информации, регламентирующие вопросы менеджмента рисков информационной безопасности. В частности, были установлены основные требования к оценке и обработке рисков информационной безопасности, исходя из международного стандарта «ISO 27001:2013 Информационные технологии. Методы защиты. Системы менеджмента информационной безопасности», а также проведено сравнение данного стандарта с его версией от 2005 года.

В качестве ведущего метода оценки и обработки рисков был выбран качественный метод, как наиболее экономичный, в условиях отсутствия готовых данных о количестве реализованных атак в рассматриваемой информационной системе за отдельный промежуток времени.

В процессе были рассмотрены ценные активы организации, и, основываясь на бизнес-процессах телекоммуникационного предприятия были выделены основные и второстепенные активы, а также соответствующие им угрозы информационной безопасности, в соответствии с банком данных угроз безопасности информации Федеральной службы по техническому и экспортному контролю.

Результатом проделанной работы стал расчет рисков информационной безопасности, основанный на выделении ценных активов организации, степени потенциального ущерба при реализации угроз на такие активы и вероятности реализации угроз для рассматриваемой информационной системы телекоммуникационного предприятия. Кроме этого, были выделены приемлемые риски, обработка которых не требуется в связи с тем, что фактическая стоимость их минимизации выше убытков от реализации соответствующих им угроз.

В заключении были предложены возможные меры по минимизации рисков информационной безопасности, включающие в себя систему резервного копирования, систему защиты от несанкционированного доступа, систему антивирусной защиты, межсетевое экранирование, а также организационные меры и меры физической защиты. Предложенный метод позволяет однозначно и обоснованно оценить риски информационной безопасности организации в условиях недостаточности исходных данных, а также отсутствии дополнительных программно-аппаратных средств для оценки рисков информационной безопасности, что позволяет применять его для типовых организаций, основываясь лишь на масштабировании рассматриваемой системы, при условии отсутствия в обрабатываемых сведениях информации, составляющей государственную тайну.

Процедура обработки рисков помогает не только выявить и устранить существующие уязвимости и минимизировать вероятность реализации существующих угроз информационной безопасности, но и повысить уровень грамотности сотрудников предприятия, участвующих в процессе оценки и обработки рисков.

На сегодняшний день перед каждым предприятием, обеспокоенного вопросами безопасности своих информационных ресурсов, встает вопрос об организации системы защиты информации, которая бы позволила в полной мере обеспечить безопасность функционирования телекоммуникационного оборудования и циркулирующей информации в информационной системе предприятия. Эффективность защиты информации зависит от подхода к ее организации и правильного выбора методов расчета рисков информационной безопасности.

Существует множество методик оценки и обработки рисков, которые применимы к любой информационной системе, вне зависимости от уровня конфиденциальности обрабатываемой в ней информации, однако, как правило, для грамотного построения системы защиты информации с использованием таких методик требуется большой объем информации о реализованных атаках, а также о попытках их реализации, подлежащий программному анализу с целью выявления наиболее актуальных угроз информационной безопасности (далее – ИБ), то есть необходима своеобразная отправная точка, с которой и следует начинать создание системы защиты, об этом говорят стандарты **BS 7799-3 и NIST 800-30**, что не всегда возможно реализовать практически, ввиду ограниченности временных и финансовых ресурсов – это особенно актуально для теле-коммуникационных организаций, так как объемы данных в таких предприятиях огромны, а анализ каждого пакета слишком дорогостоящая и трудоем- кая процедура.

В данной работе предлагается **метод расчета рисков для системы, которую можно охарактеризовать большими объемами данных, и неопределенным числом пользователей.** [1–3]

Необходимо отметить, что существует ряд методик оценки рисков информационной безопасности, позволяющих однозначно и с высокой степенью обоснованности выделить актуальные риски, международные и национальные стандарты предлагают достаточно исчерпывающий выбор методов по данному вопросу, однако их применение возможно только в условиях небольшого объема данных, и малого числа пользователей, а сами методики весьма обобщенные. Примерами конкретизированных методик, применение которых возможно на практике, являются работы [4–6], однако их использование целесообразно при наличии ограниченного числа конечных точек.

Отличительной чертой любого телекоммуникационного предприятия является чувствительность к безопасности и надежной работе всего аппаратно-программного комплекса для обеспечения непрерывности функционирования ключевых бизнес-процессов организации, что просто обязывает руководителей организации создать и поддерживать эффективную систему информационной безопасности. [7]

В рамках данной работы предложен качественный метод оценки рисков ИБ, основанный на разбиении информационной системы телекоммуникационного предприятия на типовые сегменты (включающие не более трех контролируемых зон), обладающие одинаковыми характеристиками с точки зрения информационной безопасности. А сама методика расчета рисков основывается на совокупности способов и методов определения и оценки рисков, предложенных рядом международных и российских стандартов в сфере информационной безопасности, применение которых возможно к рассматриваемой информационной системе:

1. Определение ценности активов

Одним из ключевых документов, описывающих требования к методу обработки и оценки рисков является международный стандарт «**ISO 27001: Информационные технологии. Методы защиты. Системы менеджмента информационной безопасности»** (далее – стандарт ISO 27001). Процесс расчета рисков информационной безопасности актуален на всех этапах работы системы защиты информации и является интересным для владельца информации в первую очередь с точки зрения потерь в экономической сфере.

Несмотря на то, что в рамках требований ISO 27001:2013 не рассматриваются явные формулы для расчета рисков, исходя из данных документа можно выделить следующее:

– в процессе оценки рисков должны быть установлены критерии приемлемости риска и критерии для оценки рисков ИБ;

– должны быть даны гарантии того, что оценка рисков ИБ даст обоснованные и непротиворечивые массивы актуальных, для рассматриваемой системы, рисков ИБ;

– должна быть произведена идентификация рисков ИБ, направленных на такие свойства информационных ресурсов, как конфиденциальность, целостность и доступность;

– а также, должна производиться идентификация владельца риска, где под владельцем понимается физическое, юридическое лицо или подразделение, отвечающее за управление риском и обладающее необходимыми для этого полномочиями, в данном случае, речь может идти о руководителях, специалистах по информационной защите, отделах по ИБ и пр.; [8]

– в процессе анализа рисков ИБ должна быть произведена оценка потенциальных потерь в случае реализации риска;

– должна быть оценена вероятность реализации рисков и определена величина рисков;

– в процессе оценки рисков ИБ должно быть произведено сопоставление рисков с установленными критериями, а также определен вектор приоритетных направлений по их обработке. [9]

Стандарт **ISO 27001:2013** существенно урезан, в отличие от стандарта ISO 27001:2005, где процесс оценки рисков был достаточно подробно рассмотрен, и включал в себя такие этапы, как идентификация уязвимостей и идентификация активов и их владельцев. [10–11]

Исходя из **ГОСТ Р ИСО 31000-2010**, существует множество методов по оценке рисков ИБ: «идентификация риска, анализ последствий реализации рисков ИБ, оценка эффективности существующих средств управления, количественная оценка уровня рисков ИБ, сравнительная оценка рисков ИБ, качественная, количественная или смешанная оценка вероятностных характеристик риска».

Выбор метода оценки рисков ИБ должен основываться на следующих факторах:

– временные, финансовые, информационные ресурсы;

– степень неопределенности оценки рисков ИБ;

– наличие либо отсутствие возможности получение количественных оценок выходных денных, где выходными данными могут являться мнения, решения, перечни, а также рекомендации, в зависимости от метода и этапа оценки рисков ИБ. [12]

На практике, расчет рисков необходимо начинать с документа «Методология оценки и обработки рисков», который разрабатывается до анализа и обработки рисков.

Итогом мероприятий, проведенных в соответствии с методикой должен стать отчет с суммарными результатами всех мероприятий по оценке степени рисков и их обработке.

В данном случае рассматривается **корпоративная распределенная многопользовательская информационная система** (далее – ИС), имеющая подключение к сетям общего пользования, обрабатывающая информацию разного уровня конфиденциальности, не содержащую сведения, составляющие государственную тайну (рис.39)

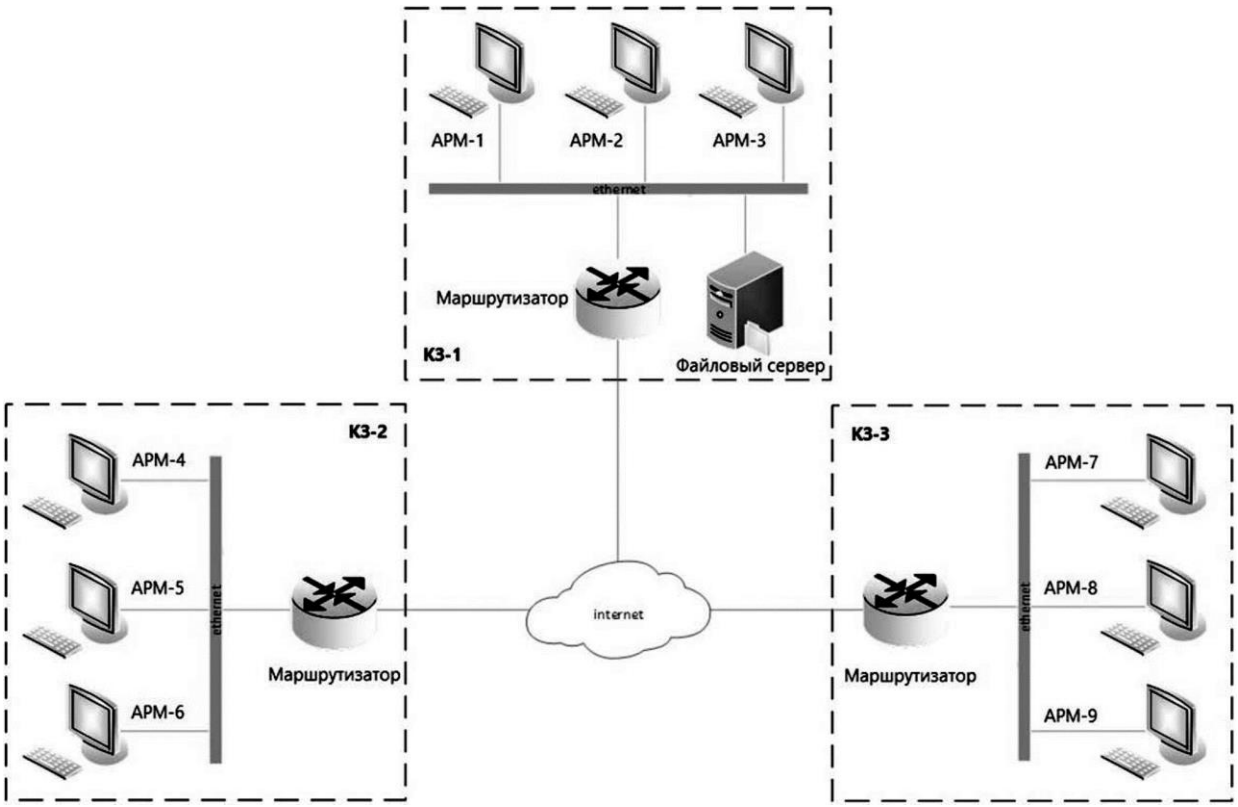


Рисунок 39 - Схематичное расположение распределенной информационной системы

В соответствии с ГОСТ Р ИСО/МЭК 27005-2010 «Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности» ценные активы организации можно разделить на **основные** и **вспомогательные**.

Основные активы:

1. Бизнес-процессы – совокупность различных видов деятельности, в результате которой создается продукт или услуга, представляющие интерес для потребителя.

2. Информация – сведения, являющиеся предметом собственности, подлежащие защите от нарушения конфиденциальности, целостности и доступности, в соответствии с требованиями правовых документов и требованиями владельца информации, вне зависимости от формы представления. Сведения, компрометация которых никаким образом не повлияет на деятельность организации, не рассматриваются как ценный актив.

Идентификатор актива	Актив организации		Конфиденциальность	Целостность	Доступность	Ценность актива
A.	Основные активы Информация	Информация, необходимую для реализации назначения или бизнеса организации	2	4	4	**4**
B.		Информация личного характера, которая определена особым образом, соответствующим национальным законам о неприкосновенности частной жизни	3	1	1	**3**
C.		Стратегическая информация, необходимая для достижения целей организации	2	2	1	**2**
D.		Информацию, обработка которой требуют продолжительного времени и/или связаны с большими затратами на ее приобретение	3	2	2	**3**
E.	Аппаратно-программный комплекс		–	3	4	**4**
F.	Носители информации		–	1	2	**2**
G.	Сеть		–	3	4	**4**
H.	Сотрудники		–	1	1	**1**
I.	Место функционирования организации		–	1	1	**1**

Рисунок 40 - Шкала ценности активов

Вспомогательные активы:

1. Аппаратно-программный комплекс – совокупность технических и программных средств, предназначенных для выполнения взаимосвязанных эксплуатационных функций по обработке информации ограниченного распространения, включающая в себя активную аппаратуру обработки данных, стационарную аппаратуру, периферийные обрабатывающие устройства, операционные системы и прикладное программное обеспечение.

2. **Носители данных** – носитель для хранения данных, включая электронный носитель и аналоговый.

3. **Сеть** – совокупность телекоммуникационных устройств, используемых для соединения нескольких физически удаленных сегментов информационной системы.

4. **Персонал** – в широком смысле, все субъекты, имеющие легитимный доступ в пределы контролируемой зоны и являющиеся потенциальными внутренними нарушителями.

5. **Место функционирования организации** – пределы контролируемой зоны, в которой функционирует информационная система.

ГОСТ Р ИСО/МЭК 27005-2010 условно разделяет ин- формацию на: «информацию, необходимую для реализации назначения или бизнеса организации, информацию личного характера, которая определена особым образом, соответствующую национальным законам о неприкосновенности частной жизни, стратегическую информацию, необходимую для достижения целей организации, информацию, обработка которой требу- ют продолжительного времени и/или связаны с большими за- тратами на ее приобретение».

Первоначально необходимо определить **ценность активов** (далее – ЦН) организации, в данном случае будет рассмотрена **четырех-балльная система оценки ценности активов:**

1 – реализация риска, направленного на конфиденциальность, целостность и/или доступность актива не будет иметь последствий, как для организации в целом, так и бизнес-процессов, в частности.

2 – реализация риска, направленного на конфиденциальность, целостность и/или доступность актива приведет к **незначительным потерям** для организации, в условиях, когда восстановление прежнего состояния системы возможно без остановки бизнес-процессов.

3 – реализация риска, направленного на конфиденциальность, целостность и/или доступность актива приведет к значительным финансовым потерям и/или окажет **существенное негативное влияние** на престиж организации, в условиях, когда восстановление прежнего состояния системы возможно, но требует больших временных и/или финансовых ресурсов.

4 – реализация риска, направленного на конфиденциальность, целостность и/ или доступность актива может **привести к полной остановке** бизнес-процессов, большим финансовым потерям и/или окажет значительное негативное влияние на престиж организации.

Так как **бизнес-процессом** является совокупность различных видов деятельности, в результате которой создается продукт или услуга, то в перечне актуальных угроз и существующих уязвимостей остальных ценных активов будут содержаться угрозы и уязвимости актуальные и для бизнес-процессов.

Особенностью рассматриваемой категории предприятий является то, что основной ущерб бизнес-процессам организации способны нанести угрозы доступности сетевого оборудования и программно- аппаратного комплекса, а не угрозы, направленные на нарушение конфиденциальности информационных ресурсов предприятия.

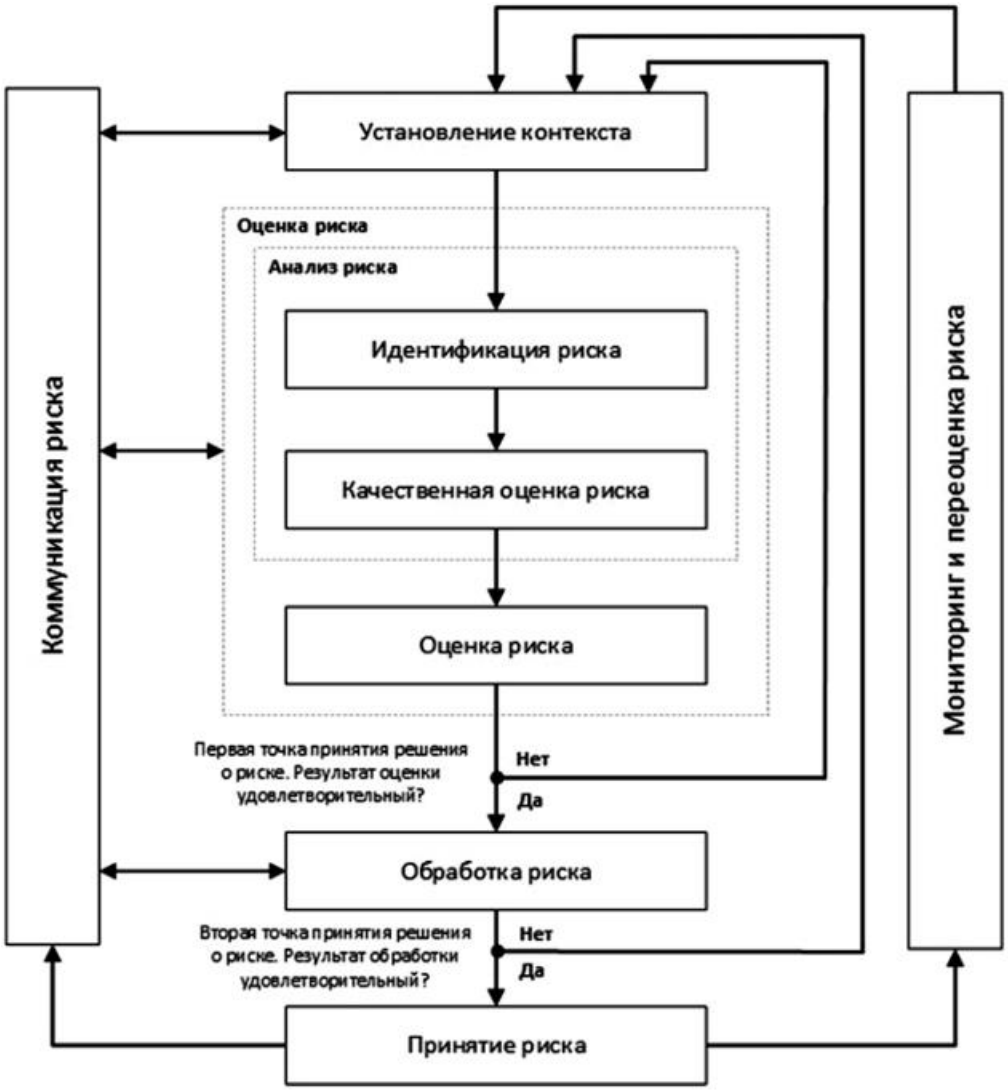


Рисунок 41 - Процесс оценки и обработки рисков ИБ

2. Оценка рисков информационной безопасности

Целесообразно обработку рисков ИБ рассматривать, как итеративный процесс, это позволит повысить уровень детализации оценки рисков при каждой последующей итерации.

Пример **итеративного процесса оценки и обработки рисков** ИБ подробно описан в **ГОСТ Р ИСО/МЭК 27005-2010** и показан на **рис. 41**, где под контекстом риска понимается установление критериев для обработки рисков ИБ, а также назначаются ответственные сотрудники или подразделение, занимающиеся вопросом менеджмента рисков ИБ.

Под идентификацией риска понимается процесс нахождения и определения рисков ИБ, под оценкой риска понимается присвоение числовых значений последствиям реализации риска, а также вероятности его реализации. Приятие риска означает, что ущерб от реализации риска является приемлемым, а вероятность его реализации мала настолько, что позволяет не проводить процедур обработки риска ИБ. Коммуникация риска позволяет осуществлять обмен сведениями об актуальных рисках между причастными сторонами.

Под обработкой риска понимается процесс минимизации последствий от реализации риска и/или процесс минимизации вероятности реализации риска ИБ. [13]

Пример деятельности по обработке рисков ИБ представлен на **рис. 42** в соответствии с **ГОСТ Р ИСО/МЭК 27005-2010**.

Следующим шагом является определение степени уязвимости каждого из ценных активов организации (далее – СУ).

В рамках данной работы будет рассмотрен выборочный ряд угроз ИБ, с ID в соответствии с **банком данных угроз ФСТЭК**:

– «угроза длительного удержания вычислительных ресурсов пользователями» (**014**);

– «угроза загрузки нештатной операционной системы» (**018**);

– «угроза приведения системы в состояние «отказ в обслуживании» (**140**);

– «угроза программного выведения из строя средств хранения, обработки и (или) ввода/вывода/передачи информации» (**143**);

– «угроза утраты вычислительных ресурсов» (**155**);

– «угроза утраты носителей информации» (**156**);

– «угроза физического выведения из строя средств хранения, обработки и (или) ввода/вывода/передачи информации» (**157**);

– «угроза форматирования носителей информации» (**158**);

– «угроза хищения средств хранения, обработки и (или) ввода/вывода/передачи информации» (**160**);

– «угроза неправомерного шифрования информации» (**170**);

– «угроза распространения «почтовых червей» (**172**);

– «угроза физического устаревания аппаратных компонентов» (**182**);

– «угроза внедрения вредоносного кода через рекламу, сервисы и контент» (**186**);

– «угроза маскирования действий вредоносного кода» (**189**).

Угрозы ИБ	Ценные активы организации								
	A.	B.	C.	D.	E.	F.	G.	H.	I.
014	—	—	—	—	2	—	—	—	—
018	1	1	1	1	3	—	—	—	—
022	—	—	—	—	2	—	2	—	—
023	—	—	—	—	3	—	—	—	—
030	2	2	2	2	1	—	—	—	—
034	—	—	—	—	—	—	1	—	—
036	1	1	1	1	1	—	1	—	—
091	3	3	3	3	—	—	—	—	—
113	—	—	—	—	2	—	—	—	—
121	2	2	2	2	2	—	—	—	—
122	—	—	—	—	2	—	—	—	—
139	1	1	1	1	3	3	—	—	—
140	—	—	—	—	3	—	3	—	—
143	3	3	3	3	2	2	—	—	—
155	1	1	1	1	3	3	3	—	—
156	3	3	3	3	—	2	—	—	—
157	—	—	—	—	1	1	—	—	—
158	1	1	1	1	—	1	—	—	—
160	1	1	1	1	2	2	—	—	—
170	2	2	2	2	—	—	—	—	—
172	—	—	—	—	—	—	2	—	—
182	—	—	—	—	1	—	1	—	—
186	2	2	2	2	—	—	2	—	—
189	—	—	—	—	1	—	1	—	—

Рисунок 42 – Ценные активы организации

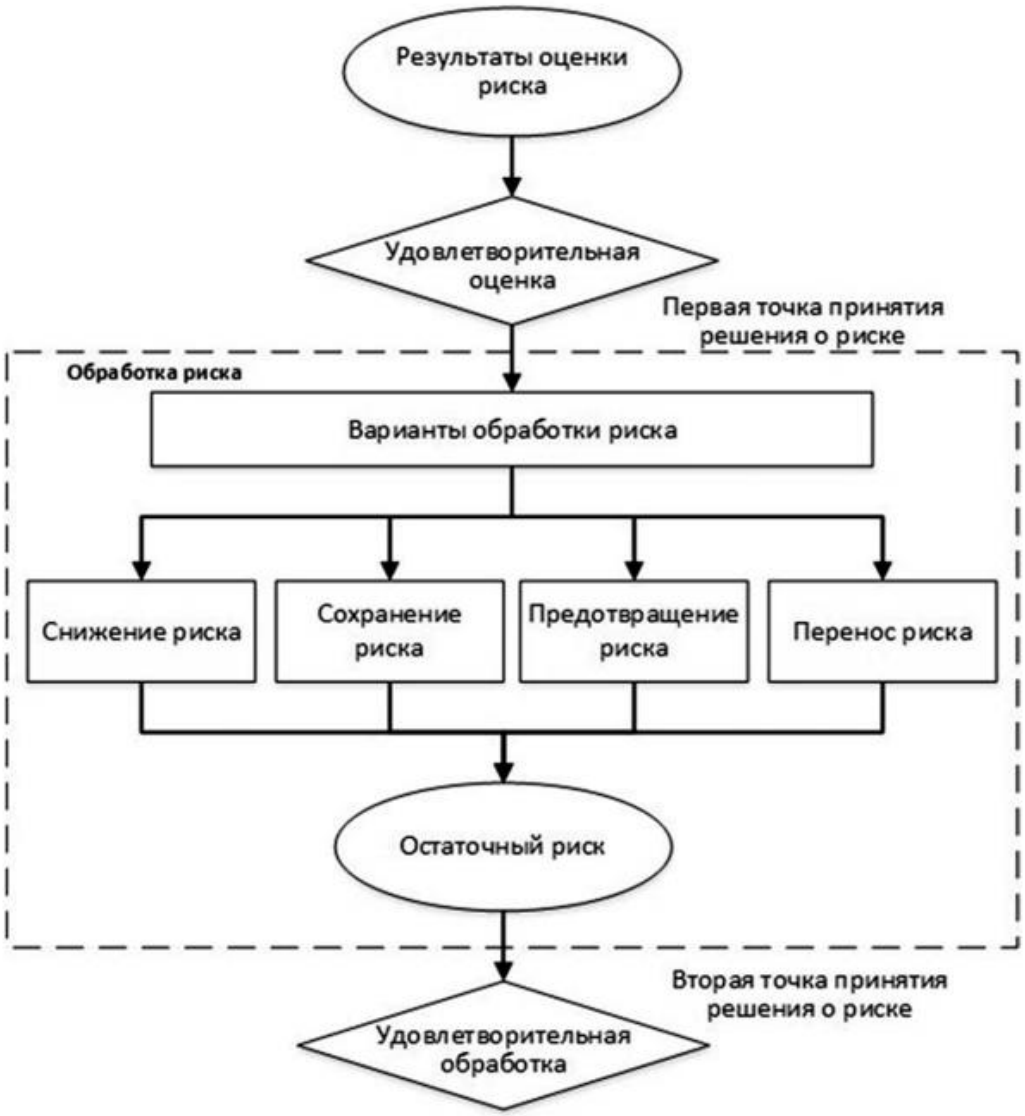


Рисунок 43 – Деятельность, направленная на обработку рисков ИБ

На рис. 42 представлен **результат оценки уязвимости** актива для перечня угроз, где **1** – низкая уязвимость по отношению конфиденциальности, целостности и/или доступности ценного актива организации, **2** – средняя степень уязвимости, а **3** – высокая степень уязвимости.

Последним этапом перед расчетом рисков ИБ является оценка вероятности реализации угроз ИБ (**далее – В**), представленных на **рис. 42.**

Оценка вероятности представлена на **рис. 44**, где **1** – угроза существует, но не встречалась в рассматриваемой сфере, 2 – угроза возникает в рассматриваемой сфере 2–3 раза в год, **3** – угроза была реализована в рассматриваемой системе, **4** – угроза возникает 2–3 раза в год в рассматриваемой системе.

ID угрозы	014	018	022	023	030	034	036	091	113	121	122	139	140	143	155	156	157	158	160	170	172	182	186	189
Вероятность	2	1	2	3	1	2	2	4	2	2	2	3	2	2	2	4	3	3	3	2	2	3	3	2

Рисунок 44 – Вероятность реализации угрозы

3. Отчет об оценке рисков ИБ

Общий уровень риска ИБ для каждого из ценных активов организации рассчитывается по формуле 1, а на **рисунке 45** в табличном виде представлен результат для активов A, E, G.

$$Р = ЦН \times СУ \times В \quad (1)$$

Приемлемым риском считается риск, чье числовое значение находится в промежутке от 1 до 10, такой риск считается незначительным, и обработка такого риска не требуется.

Средний риск, чье числовое значение находится в диапазоне от 11 до 21 рекомендован к обработке с целью его минимизации. [15–16] Высокий риск, чье числовое значение находится в диапазоне от 22 до 64, данный риск считается существенным, и его обработка обязательна.

4. Возможные контрмеры

Допустим, что руководитель предприятия принимает решение, что риски с числовым значением выше 20 подлежат обработке с целью их минимизации.

Возможные контрмеры представлены на **рис. 46**. [17–20].

После обработки рисков ИБ, остаточный рискстал приемлемым для каждой из актуальных угроз информационной безопасности.

Ценный актив организации	Угрозы	ЦН	СУ	В	Р	Числовое значение оценки риска
Информация, необходимую для реализации назначения или бизнеса организации	018	4	1	1	4	Низкий
	030	4	2	1	8	Низкий
	036	4	1	2	8	Низкий
	091	4	3	4	48	Высокий
	121	4	2	2	16	Средний
	139	4	1	3	12	Средний
	143	4	3	2	24	Высокий
	155	4	1	2	8	Низкий
	156	4	3	4	48	Высокий
	158	4	1	3	12	Низкий
	160	4	1	3	12	Низкий
	170	4	2	2	16	Низкий
	186	4	2	3	24	Высокий
Аппаратно-программный комплекс	014	4	2	2	16	Средний
	018	4	3	1	12	Средний
	022	4	2	2	16	Средний
	023	4	3	3	36	Высокий
	030	4	1	1	4	Низкий
	036	4	1	2	8	Низкий
	113	4	2	2	16	Средний
	121	4	2	2	16	Средний
	122	4	2	2	16	Средний
	139	4	3	3	36	Высокий
	140	4	3	2	24	Высокий
	143	4	2	2	16	Средний
	155	4	3	2	24	Высокий
	157	4	1	3	12	Средний
	160	4	2	3	24	Высокий
	182	4	1	3	12	Средний
	189	4	1	2	8	Низкий
Сеть	022	4	2	2	16	Средний
	034	4	1	2	8	Низкий
	036	4	1	2	8	Низкий
	140	4	3	2	24	Высокий
	155	4	3	2	24	Высокий
	172	4	2	2	16	Средний
	182	4	1	3	12	Средний
	186	4	2	3	24	Высокий
	189	4	1	2	8	Низкий

Рисунок 45 – Оценка рисков ИБ

Ценный актив организации	Угрозы	Риск	Приемлемый риск	Планируемые меры	Остаточный риск
Информация, необходимую для реализации назначения или бизнеса организации	091	48	От 1 до 19	Система резервного копирования, система защиты от НСД	12
	143	24		Система антивирусной защиты, межсетевое экранирование	12
	156	48		Учет носителей информации	12
	186	24		Система антивирусной защиты, межсетевое экранирование; Организационные меры	8
Аппаратно-программный комплекс	023	36		Межсетевое экранирование, система доверенной загрузки, система антивирусной защиты; Организационные меры	12
	139	36		Системы видеонаблюдения, адекватные средства физической защиты; Организационные меры.	12
	140	24		Система межсетевого экранирования	12
	155	24		Система межсетевого экранирования	12
	160	24		Системы видеонаблюдения, адекватные средства физической защиты; Организационные меры.	8
Сеть	140	24		Система межсетевого экранирования	12
	155	24		Система межсетевого экранирования	12
	186	24		Система антивирусной защиты, межсетевое экранирование; Организационные меры	8

Рисунок 46 – Рекомендованные контрмеры

Заключение

Предложенная методика позволила однозначно и обоснованно оценить риски информационной безопасности организации в условиях большого объема обрабатываемой информации и неограниченного числа пользователей и потребовала минимальных финансовых вливаний. Применение рассмотренного метода на практике способствовало выявлению основных угроз защиты безопасности, основываясь на банке данных угроз безопасности информации ФСТЭК России. Исходя из результатов оценки рисков информационной безопасности, в последствие, была создана модель угроз рассматриваемого телекоммуникационного предприятия.

Стоит отметить, что предложенная методика одинаково применима, как к автоматизированной информационной системе, так и к системам обработки информации без использования средств автоматизации. Однако, применение специализированных программных продуктов, позволяющих осуществлять оценку рисков ИБ, все же является приоритетным, так как может позволить функционировать системе управления рисками в режиме реального времени, при условии достаточности временных и финансовых ресурсов, в отличие от рассмотренного метода, практическая реализация которого возможно в качестве разового или периодически проводимого мероприятия.

[УКАЗАТЕЛИ]

1. Некрылова Н.В. Предпосылки реализации элементов управления рисками бизнес-процессов в стандартах на системы менеджмента промышленного предприятия // Известия высших учебных заведений. Поволжский регион. Общественные науки. 2015. № 2 (34). С. 204–215.

2. Андреева Н.В. Функциональная модель системы управления информационной безопасностью как средство внедрения стандартов линейки ISO/IEC 2700x (BS 7799) // Научно- технический вестник информационных технологий, механики и оптики. 2007. № 39. С. 40–44.

3. Пугин В.В., Губарева О.Ю. Обзор методик анализа рисков информационной безопасности информационной системы предприятия // T-Comm – Телекоммуникации и Транспорт. 2012. № 6. С. 54–57.

4. Плетнев П.В., Белов В.М. Методика оценки рисков информационной безопасности на предприятиях малого и среднего бизнеса // Доклады Томского государственного университета систем управления и радиоэлектроники. 2012.

№ 1–2 (25). С. 83–86.

5. Одинцова М.А. Методика управления рисками для малого и среднего бизнеса. // Экономический журнал. 2014. № 3 (35). URL: https://cyberleninka.ru/article/n/metodika-upravleniya-riskami-dlya-malogo-i-srednego-biznesa (дата обращения: 01.02.2018).

6. Глушенко С.А. Применение системы Matlab для оценки рисков информационной безопасности организации // Бизнес-информатика. 2013. № 4 (26). С. 35–42.

7. Губарева О.Ю. Оценка рисков информационной безопасности в телекоммуникационных сетях. // Вестник Волжского университета им. В.Н. Татищева. 2013. № 2 (21). С. 76–81.

8. Дорофеев А.В. Менеджмент информационной безопасности: переход на ISO 27001:2013 // Вопросы кибербезопасности. 2014.№ 3 (4). С. 69–73.

9. ISO/IEC 27001:2013. Information technology. Security techniques. Information security management systems. Requirements. Berlin: ISO/IEC JTC 1/SC 27. 2013. 23 p.

10. Дорофеев А.В. Подготовка к CISSP: телекоммуникации и сетевая безопасность // Вопросы кибербезопасности. 2014. № 4 (7). С. 69–74.

11. Ильченко Л.М. Анализ системы менеджмента информационной безопасности на базе стандарта ISO 27001:2013. // Материалы 5 научно-практической конференции студентов, аспирантов и курсантов «IT вчера, сегодня, завтра». 2017. С. 51–61.

12. ГОСТ Р ИСО 31000-2010. Менеджмент риска. Принципы и руководство.; Введен с 01.09.2011. Москва: Изд-во Стандартинформ, 2012.

13. ГОСТ Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности. Взамен ГОСТ Р ИСО/МЭК ТО 13335-3-2007 и ГОСТ Р ИСО/ МЭК ТО 13335-4-2007; Введ. с 30.11.2010. Москва: Изд-во Стандартинформ, 2011.

14. Банк данных угроз безопасности информации // Федеральная служба по техническому и экспортному контролю URL: https://bdu.fstec. ru (дата обращения: 01.02.2018).

15. Шаго Ф.Н., Зикратов И.А. Методика оптимизации планирования аудита системы менеджмента информационной безопасности // Научно-технический вестник информационных технологий, механики и оптики. 2014. № 2 (90). С. 111–117.

16. Выборнова О.Н., Давидюк Н.В., Кравченко К.Л. Оценка информационных рисков на основе экспертной информации (на примере ГБУЗ АО «Центр медицинской профилактики») // Инженерный вестник Дона. 2016. № 4 (43). С. 86.

17. Пащенко И.Н., Васильев В.И. Разработка требований к системе защиты информации в интеллектуальной сети Smart Grid на основе стандартов ISO/IEC 27001 и 27005 // Известия ЮФУ. Технические науки. 2013. № 12 (149). С. 117–126.

18. ГОСТ Р ИСО/МЭК 31010-2011. Менеджмент риска. Методы оценки риска; Введ. с 01.12.2012. Москва: Изд-во Стандартинформ; 2012.

19. Эмануэль А.В., Иванов Г.А., Гейне М.Д. Применение менеджмента рисков на основе стандарта ИСО 14971: методические подходы // Вестник Росздравнадзора. 2013. № 3. С. 45–60.

20. Лютова И.И. Моделирование уровня приемлемого риска информационной безопасности // Вестник Адыгейского государственного университета. Серия 5: Экономика. 2014.№ 2 (141). С. 175–180.

§ ПРАКТИКУМ: ZERO TRUST в IoT

Введение

Одна из самых распространённых проблем современности – это утечка данных из корпоративной сети. Новости о том, что злоумышленники украли очередную базу данных или получили конфиденциальную финансовую информацию, появляются регулярно. Ещё больше таких событий никогда в новости не попадают. Как защитить себя и свою организацию от подобных проблем? Возможно, с помощью принципа «нулевого доверия», Zero Trust.

В начале 2020 года Национальный институт стандартов и технологий США (NIST) опубликовал черновик второй редакции документа, в котором рассматриваются основные логические компоненты архитектуры с нулевым доверием (**Zero Trust Architecture, ZTA**).

Нулевое доверие (Zero Trust) относится к развивающемуся набору парадигм сетевой безопасности, в основе которых лежит принцип «никому ничего не доверяй». **В отличии от классических подходов, уделяющих больше внимания защите периметра, модель Zero Trust акцентируется на безопасности ресурсов, а не сегментов сети предприятия**.

Сейчас мы изучим модель усиления кибербезопасности, построенную на принципах архитектуры с нулевым доверием, оценим риски ее использования и познакомимся с некоторыми популярными сценариями развертывания.

Первый проект ZTA от NIST появился в сентябре 2019 года, хотя концепция нулевого доверия существовала в кибербезопасности задолго до появления самого термина «нулевое доверие».

Агентство оборонных информационных систем (DISA) и Министерство обороны США в 2007 года опубликовали работу, посвященную безопасной стратегии предприятия. Данная стратегия, получившая название «Черное ядро», предусматривала переход от модели безопасности на основе периметра к модели, ориентированной на безопасность отдельных транзакций.

В 2010 году главный аналитик Forrester Research Джон Киндерваг для описания различных решений, меняющих фокус восприятия угроз (от безопасности, построенной на стратегии защиты периметра, к контролю над всеми имеющимися данными), сформулировал термин «нулевое доверие».

Модель Zero Trust стала попыткой решения классической проблемы, когда проникнувший в сеть злоумышленник получает доступ ко всем ее компонентам. Достаточно сказать, что, по данным Microsoft Vulnerabilities Report, последствия 88 % критических уязвимостей можно было бы устранить или, как минимум, смягчить, лишив пользователей административных прав.

Защищенные на уровне периметра корпоративные сети предоставляют аутентифицированным пользователям авторизованный доступ к широкому набору ресурсов. В результате несанкционированное боковое перемещение внутри сети стало одной из самых серьезных проблем кибербезопасности.

Модель нулевого доверия

Для развертывания модели Zero Trust необходимо распределить минимальные привилегии доступа и максимально детализировать пакеты с данными. В модели с нулевым доверием вы определяете «защищаемое пространство», состоящее из наиболее важных и ценных данных и ресурсов, и фиксируете маршруты трафика по организации с точки зрения их отношения к защищаемым ресурсам.

Как только появляется понимание связей между ресурсами, инфраструктурой и сервисами, можно создавать **микро-периметры** — межсетевые экраны на уровне сегментов корпоративных сетей.

При этом пользователи, которые могут удаленно проходить **микро-периметры**, находятся в любой точке мира, используют различные устройства и данные.

Отличительной особенностью архитектуры Zero Trust является большое внимание к аутентификации и авторизации до предоставления доступа к каждому ресурсу компании. При этом требуется минимизация временных задержек в механизмах аутентификации.

На рис. 47 представлена абстрактная модель предоставления доступа:

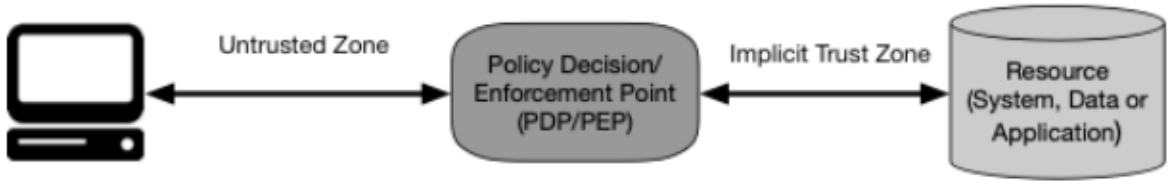


Рисунок 47 – Модель доступа предоставления в **Zero Trust Architecture**

В модели пользователю (или устройству) необходимо получить доступ к корпоративному ресурсу через «контрольно-пропускной пункт». Пользователь проходит проверку через точку принятия решения о доступе на основе политики безопасности (**Policy Decision Point, PDP**) и через точку реализации политики (**Policy Enforcement Point, PEP**), отвечающую за вызов PDP и правильную обработку ответа.

Идея в том, чтобы переместить точку применения политики как можно ближе к приложению. PDP/PEP не может применять дополнительные политики за пределами своего местоположения в потоке трафика.

Принципы нулевого доверия

Приведем семь основных принципов ZT и ZTA (в сокращенном виде), которые должны учитываться при построении безопасной системы. Данные принципы являются «идеальной целью», однако не все из них могут быть полностью реализованы в каждом конкретном случае.

Все источники данных и услуг считаются ресурсами. Сеть может состоять из нескольких устройств разного класса. Компания вправе классифицировать личные устройства в качестве ресурсов, если они могут получить доступ к данным и услугам, принадлежащим предприятию.

Все коммуникации защищены независимо от их местоположения в сети. Доверие не может быть связано с местоположением. Запросы на доступ от пользователей, расположенных в сетевой инфраструктуре предприятия (например, внутри традиционного периметра), должны отвечать тем же требованиям безопасности, что и запросы, поступающие из любой другой сети. Коммуникации должны осуществляться максимально безопасным способом, обеспечивать конфиденциальность и аутентификацию источника.

Доступ к отдельным корпоративным ресурсам предоставляется для каждой сессии. Аутентификация и авторизация для одного ресурса не дают доступ к другому.

Доступ к ресурсам определяется динамической политикой, включая наблюдаемое состояние идентификации клиента, приложения и других атрибутов (например, измеряемых отклонений в наблюдаемой модели использования). Политика — это набор правил доступа, основанных на атрибутах, которые организация назначает пользователю, ресурсу или приложению.

Предприятие гарантирует максимально безопасное состояние всех принадлежащих ему устройств, и отслеживает активы, чтобы гарантировать их максимальную безопасность. «Максимально возможное безопасное состояние» означает, что устройство находится в наиболее практичном безопасном состоянии и все еще выполняет действия, соответствующие его миссии.

Все ресурсы аутентификации и авторизации являются динамическими и строго контролируются. Это постоянный цикл получения доступа, мониторинга и оценки угроз, адаптации и переоценки доверия к текущей связи. Предполагается, что предприятие, реализующее ZTA, будет иметь все необходимые системы управления учетными данными, активами и доступом, включая многофакторную аутентификацию.

Предприятие собирает максимум информации о текущем состоянии сетевой инфраструктуры и коммуникаций, используя ее для повышения собственной безопасности, а также данные о сетевом трафике и запросах доступа, необходимые для улучшения создания и применения политики.

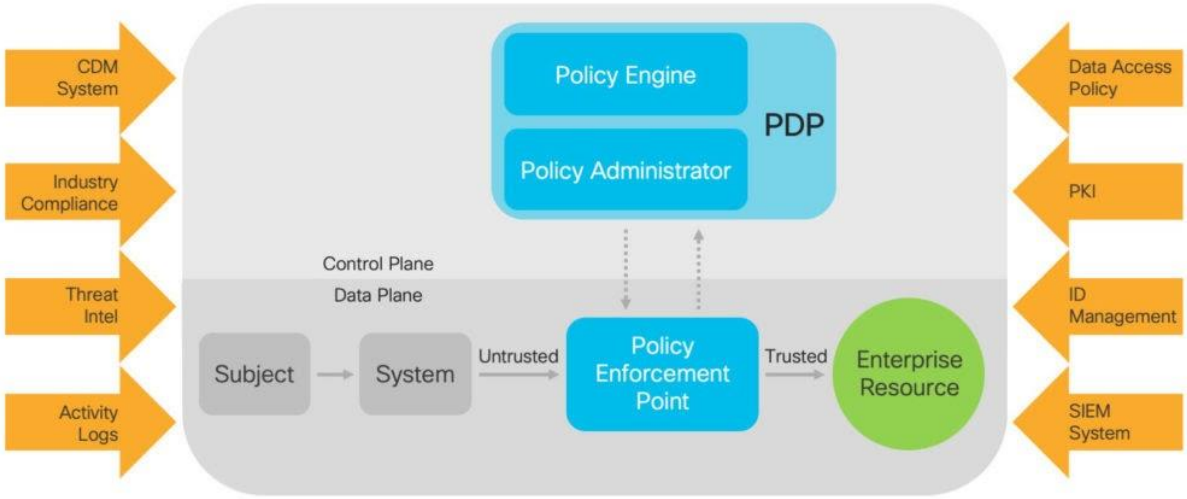


Рисунок 48 - Компоненты архитектуры

Здесь (рис. 48) для удобства приводится не оригинальный рисунок NIST, а версия из статьи Cisco «Making a Deliberate Cybersecurity Lifestyle Choice».

Существует множество логических компонентов, из которых формируется архитектура Zero Trust на предприятии. Эти компоненты могут работать как локальная служба или через облако. На рисунке выше представлена «идеальная модель», демонстрирующая логические компоненты и их взаимодействие.

Интеграция информации о ресурсах компании, о пользователях, о потоках данных и о рабочих процессах с политикой правил формирует необходимые вводные для принятия решения о доступе к ресурсам.

При инициировании пользователем (субъектом) процедуры аутентификации, цифровая идентификация строится вокруг него. На рисунке эта процедура представлена с блока Subject. Еще один термин для такого пользователя — принципал (Principal), то есть клиент, для которого разрешается аутентификация.

Представленная выше схема сети делится на несколько уровней трафика. Уровень управления (Control Plane) отделяется от другой части сети, которая может быть видна пользователю. С точки зрения принципала существует только уровень данных этой сети.

На Control Plane находится точка принятия решения о доступе (PDP), состоящая из двух логических компонентов:

Policy Engine (PE), отвечающего за решение о предоставлении доступа к ресурсу для данного субъекта. Данный компонент использует политику предприятия, а также входные данные из внешних источников (например, службы анализа угроз) для предоставления, отклонения или отзыва доступа к ресурсу;

Policy Administrator (PA), отвечающего за установление и/или закрытие канала связи между субъектом и ресурсом. При создании канала связи он связывается с Policy Enforcement Point (PEP), который находится на уровне данных (Data Plane).

PEP отвечает за включение, мониторинг, вызов PDP и правильную обработку его ответа, и, в конечном итоге, за разрыв соединений между субъектом и корпоративным ресурсом. За пределами PEP находится неявная зона доверия, в которой размещен корпоративный ресурс.

Все остальные поля (на рисунке слева и справа) демонстрируют компоненты безопасности, которые могут представлять информацию, необходимую для принятия решения о доступе в PDP/PEP. К ним относится, к примеру, система непрерывной диагностики и мониторинга (CDM), собирающая информацию о текущем состоянии активов предприятия.

Идентификация и микросегментация

При разработке ZTA в качестве ключевого компонента создания политики доступа используется идентичность действующих лиц. Под идентичностью подразумеваются атрибуты аутентификации и атрибуты пользователя в сети, то есть данные, которые могут быть проверены для гарантии правомерности доступа.

Конечная цель управления идентификацией на предприятии состоит в том, чтобы ограничить представление каждого пользователя сети исключительно теми ресурсами, на которые у него есть права.

Предприятие может защищать ресурсы в своем собственном сегменте сети устройствами Next-Generation Firewall (NGFW), используя их в качестве Policy Enforcement Point. NGFW динамически предоставляют доступ по отдельным запросам от клиентов. Этот подход применяется к различным случаям использования и моделям развертывания, поскольку защитное устройство действует как PEP, а управление этими устройствами — как компонент PE/PA.

Для реализации ZTA также можно использовать оверлейные сети. Такой подход иногда называют моделью с программно-определяемым периметром (SDP) и часто включает в себя концепции из программно-определяемой сети (SDN). Здесь Policy Administrator действует как сетевой контроллер, который устанавливает и реконфигурирует сеть на основе решений, принятых Policy Engine.

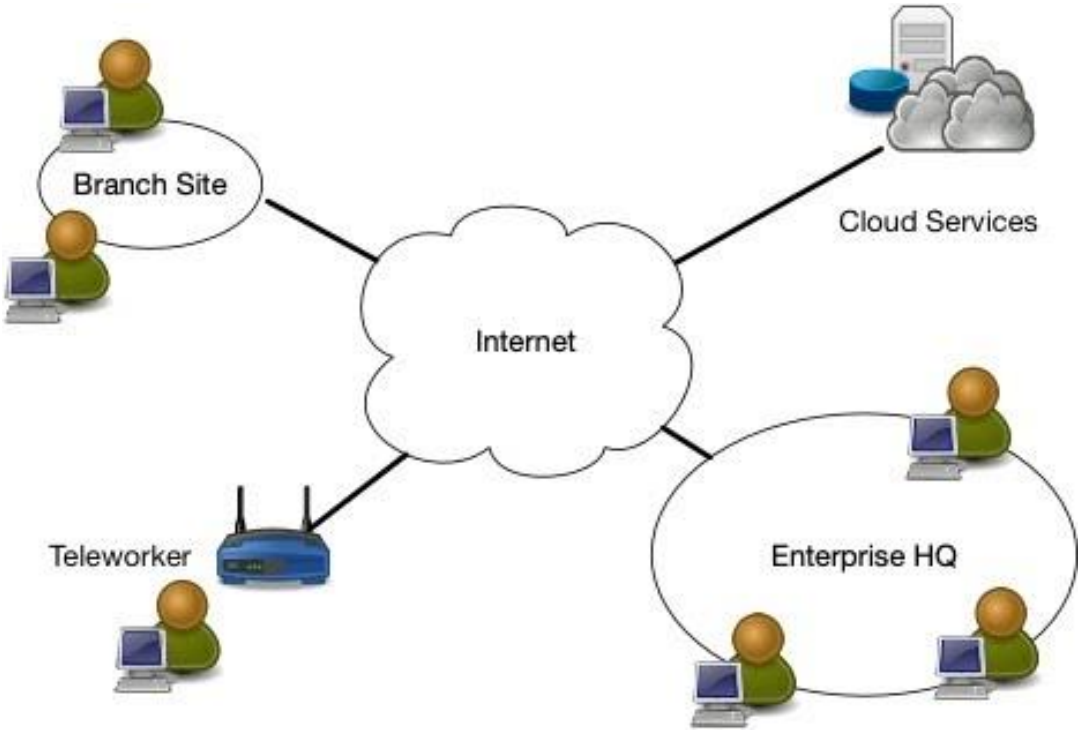


Рисунок 49 – Сеть предприятия

Наиболее распространенный сценарий развертывания ZTA относится к предприятию (**рис.49**), имеющему главный офис и несколько географически распределенных локаций, связанных друг с другом сторонними, не принадлежащими предприятию, сетевыми каналами.

В этой схеме сотрудникам, работающим удаленно (teleworker), по-прежнему требуется полноценный доступ к корпоративным ресурсам, и блок PE/PA часто развертывается в виде облачной службы.

По мере перехода предприятия на большее количество облачных приложений и сервисов, подход с нулевым доверием требует размещения PEP в точках доступа каждого приложения и источника данных.

PE и PA могут быть расположены в облаке, либо даже у третьего облачного провайдера (за пределами Cloud Provider A и Cloud Provider B).

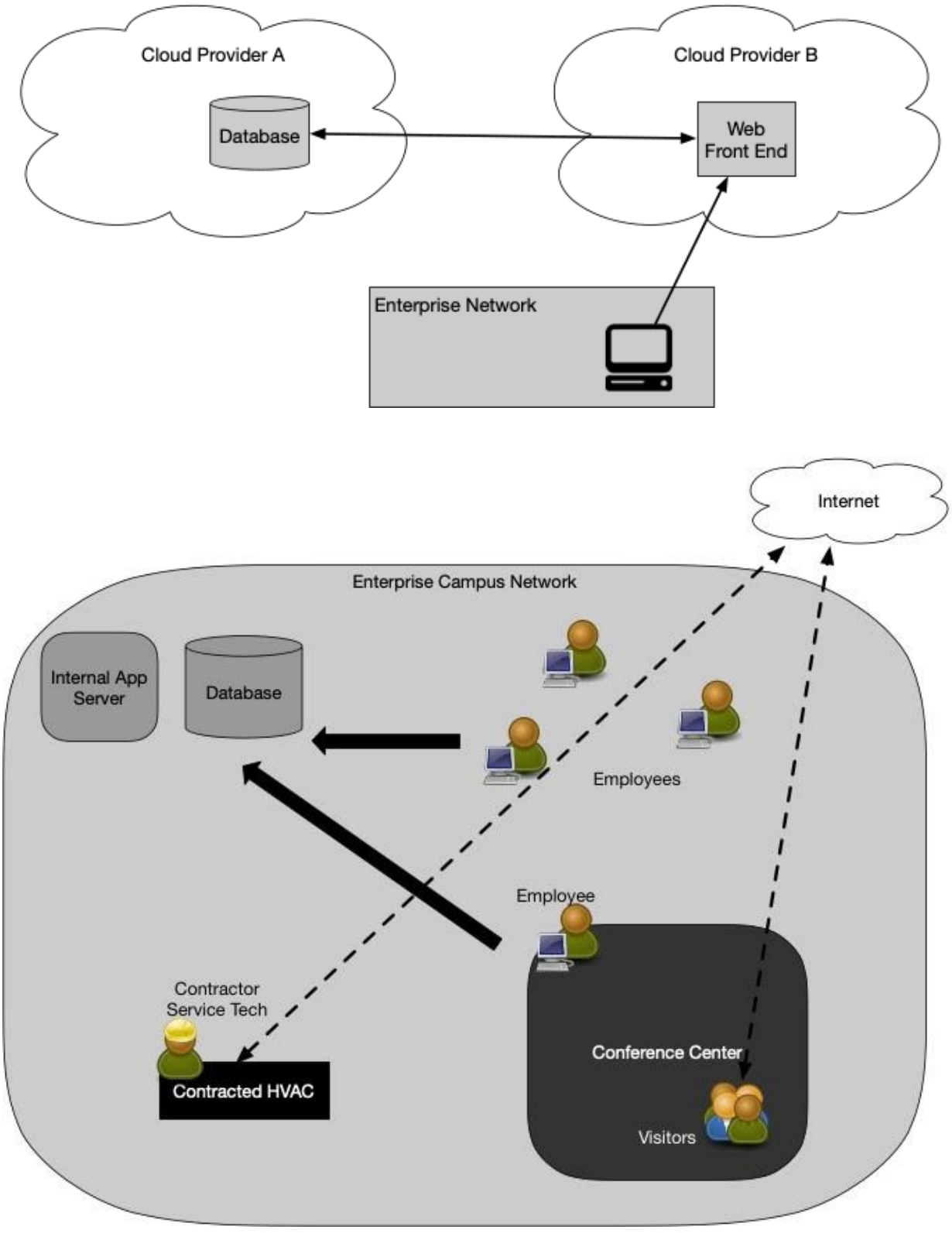


Рисунок 50 – Сценарии сетевого использования

Другой распространенный сценарий — предприятие с посетителями и/или контрактниками, которым требуется ограниченный доступ к корпоративным ресурсам. В этом примере организация также имеет конференц-центр, где посетители взаимодействуют с сотрудниками.

С помощью подхода ZTA Software-Defined Protection посетители могут выйти в интернет, но не могут получить доступ к корпоративным ресурсам. Порой они даже не имеют возможности обнаруживать корпоративные сервисы посредством сканирования сети.

Здесь PE и PA могут быть размещены в виде облачной службы или в локальной сети. PA гарантирует, что все активы, не принадлежащие предприятию, получат доступ к интернету, но не к локальным ресурсам.

Семь рисков реализации Zero Trust

Воздействие на процесс принятия решений

В ZTA компоненты Policy Engine и Policy Administrator являются ключевыми для всего предприятия. Любой администратор, имеющий доступ к настройкам правил PE, может вносить несанкционированные изменения или совершать ошибки, которые нарушат работу. Скомпрометированный PA может предоставить доступ ко всем защищенным ресурсам. Для снижения рисков компоненты PE и PA должны быть правильно настроены и проверены.

Отказ в обслуживании

PA является ключевым компонентом для доступа к ресурсам — без его разрешения невозможно установить связь. Если в результате DoS-атаки или перехвата трафика злоумышленник нарушает или запрещает доступ к PEP или PA, это может отрицательно повлиять на работу предприятия. У предприятия есть возможность смягчить угрозу, разместив PA в облаке или реплицировав его в нескольких местах.

Украденные учетные данные

Злоумышленники могут использовать фишинг, социальную инженерию или комбинацию атак для получения учетных данных ценных учетных записей. Реализация многофакторной аутентификации может снизить риск доступа из скомпрометированной учетной записи.

Видимость в сети

Часть трафика (возможно, бóльшая) в сети предприятия может быть непрозрачной для традиционных инструментов сетевого анализа.

Это не означает, что предприятие не в силах анализировать зашифрованный трафик — можно собирать метаданные и использовать их для обнаружения подозрительной активности. Методы машинного обучения позволяют исследовать трафик на глубоком уровне.

Хранение сетевой информации

Сетевой трафик и метаданные, используемые для построения контекстных политик, могут стать целью хакерских атак. Если злоумышленник получит доступ к информации о трафике, он может получить представление об архитектуре сети и определить векторы дальнейших атак.

Другим источником разведывательной информации для злоумышленника является инструмент управления, используемый для кодирования политик доступа. Как и хранимый трафик, этот компонент содержит политики доступа к ресурсам и может показать, какие учетные записи представляют наибольшую ценность для компрометации.

Опора на собственные форматы данных

ZTA использует несколько различных источников данных для принятия решений о доступе. Часто ресурсы, используемые для хранения и обработки этой информации, не имеют общего открытого стандарта взаимодействия. Если у одного провайдера есть проблема или у него нарушена безопасность, то предприятие порой не имеет возможности перейти к другому провайдеру без чрезмерных затрат.

Как и DoS-атаки, этот риск не является уникальным для ZTA, но поскольку ZTA сильно зависит от динамического доступа к информации, нарушение может повлиять на основные бизнес-функции. Для снижения рисков предприятиям следует оценивать поставщиков услуг на комплексной основе. Допуск объектов, не являющихся физическим лицом (Non-Person Entity, NPE), к компонентам управления:

Нейросети и другие программные агенты используются для управления проблемами безопасности в корпоративных сетях и могут взаимодействовать с критически важными компонентами ZTA (например, Policy Engine и Policy Administrator).

Остается открытым вопрос аутентификации NPE на предприятии с ZTA. Предполагается, что большинство автоматизированных технологических систем для доступа к API все же будут использовать какие-то средства аутентификации (например, код API Key).

Наибольший риск при использовании автоматизированной технологии для настройки и применения политик — это вероятность ложноположительных (безобидные действия, ошибочно принятые за атаки) и ложноотрицательных (атаки, ошибочно принятые за нормальную активность) реакций. Их число можно уменьшить с помощью регулярного анализа реакций.

Заключение

ZTA сегодня больше похожа не на чертеж надежной крепости, а на карту с обозначенными ключевыми точками для путешествия. Тем не менее многие организации уже имеют элементы ZTA в корпоративной инфраструктуре. Согласно выводам NIST, организации должны стремиться постепенно внедрять принципы нулевого доверия. Еще долгое время большинство корпоративных инфраструктур будут работать в гибридном режиме с нулевым доверием/периметром.

Для дальнейшего изучения темы применения концепции Zero Trust обратите внимание на следующие материалы:

«Zero Trust Networks: Building Secure Systems in Untrusted Networks»;
«Building Zero Trust networks with Microsoft 365»;
«Insider Threat Monitoring for Zero Trust with Microsoft Azure»;
«Draft (2nd) SP 800-207, Zero Trust Architecture.

Реализация архитектуры безопасности с нулевым доверием: вторая редакция (SELDON news)

Лабораторная работа №4: Проектирование информационных систем. Применение стратегии безопасности для ККС.

Цель: Сформулировать меры реализации плана внедрения политики «нулевого доверия» для IoT-системы.

Оборудование: ПК, ПО: Cisco Packet Tracer версии 7.1 и выше.

Объекты исследования: Корпоративные компьютерные сети, IoT-инфраструктура, Smart City, Smart Campus, стратегия информационной безопасности «Zero Trust», модели и политики информационной безопасности, контекстные диаграммы, диаграммы потоков данных (DFD), функциональные модели информационных систем.

Исходя из **факта выполнения** лабораторной работы №3 «Разработка сети Smart-Campus, которая моделирует университетскую среду, где, наряду с традиционными сетями корпусов и аудиторий (в общем виде), существует сеть IoT необходимо предоставить/сформулировать/задекларировать необходимую структуру документации ИС:

! Требования к отчету (пункты 1 – 5.x **формулируются** по итогам предыдущей работы

1. Общий дизайн информационной системы:

1.0. Общий дизайн информационной системы (схема)

1.1. Общий дизайн информационной системы (пояснение)

2. Список используемого оборудования

3. Список используемого программного обеспечения

4. Детализированная схема размещения оборудования:

4.0. Детализированная схема размещения оборудования (Серверная) (схема)

4.1. Детализированная схема размещения оборудования (Серверная) (пояснение)

4.2. Детализированная схема размещения оборудования (Точки коммутации) (схема)

5. Сервис физических коммуникаций:

5.0. Территориальная схема физических коммуникаций (схема)

5.1. Территориальная схема физических коммуникаций (пояснение)

! Вероятней всего у вас 4 точки коммутации (см. условия в лабораторной №3)

5.2. Схема подключения физических коммуникаций (Точка коммутации №0) (схема)

5.3. Схема подключения физических коммуникаций (Точка коммутации №0) (пояснение)

5.4. Схема подключения физических коммуникаций (Точка коммутации №1) (схема)

5.5. Схема подключения физических коммуникаций (Точка коммутации №1) (пояснение)

5.6. Схема подключения физических коммуникаций (Точка коммутации №2) (схема) (если есть)

5.7. Схема подключения физических коммуникаций (Точка коммутации №2) (пояснение) (если есть)

5.8. Схема подключения физических коммуникаций (Точка коммутации №3) (схема) (если есть)

5.9. Схема подключения физических коммуникаций (Точка коммутации №3) (пояснение) (если есть)

Создать Логическую схему **сервисов (в том числе и IoТуслуг) ИС, предоставив структуры по плану**:

6.0. Логическая схема размещения сервисов ИС (схема)

6.1. Логическая схема размещения сервисов ИС (пояснение)

Информационная система (ИС) компании являет собой комплекс средств автоматизации процессов обработки, хранения информации и взаимодействия компонентов, включает в себя ИТ-инфраструктуру компании, информацию компании, персональные компьютеры (ПК), прочие ИТ-активы.

ИТ-инфраструктура (Инфраструктура) – совокупность взаимосвязанных примененных технологий, аппаратных и программных средств, систем связи и коммуникаций, составляющих и/или обеспечивающих основу ИС.

Реализовать схемы (6.0 и 6.1) посредством методологии графического структурного анализа, диаграмм потоков данных (DFD)

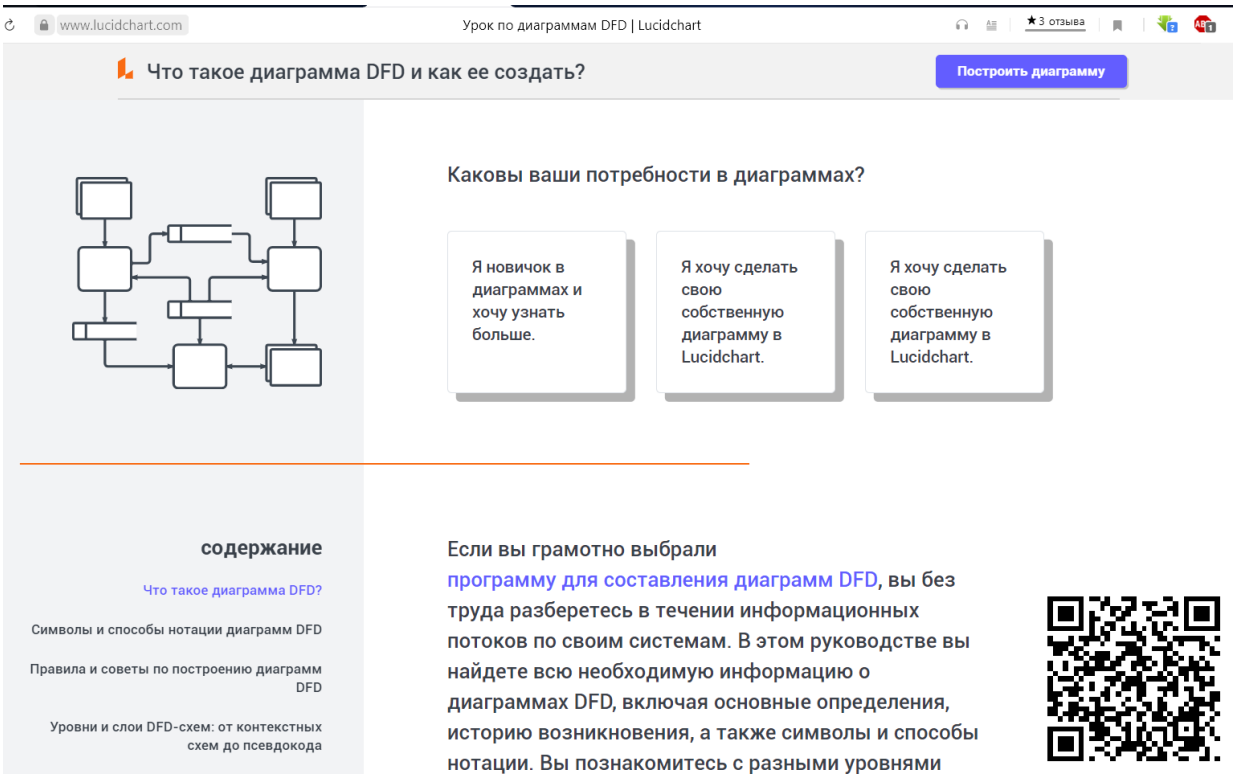
Диаграмма DFD наглядно отображает течение информации в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами «доставки» в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки.

Диаграммы DFD варьируются от простейших набросков процессов (включая нарисованные вручную) до подробных многоуровневых схем с глубоким анализом способов обработки данных.

Диаграммы DFD применяются для **анализа существующих** и **моделирования новых систем**.

Посредством интер-отклика указана ссылка на обучающий и практический курс «**Что такое диаграмма DFD и как ее создать»** от **LucidChart** - веб-проприетарной платформы, которая используется для совместной работы пользователей при составлении, пересмотре и совместном использовании диаграмм и диаграмм.

Регистрация на сервисе бесплатна.



Построение диаграмм потоков данных (которые, естественно, (как вы узнаете из курса), можно, представляются в виде все того же ориентированного графа, как мы раньше это делали (при построении устройства), позволит вам приступить к обобщению целевого назначения вашей инф. структуры, **но, что важно**, в рамках одной системы.

Ведь ранее вы просто реализовывали несколько подсистем (например: умный полив газона первого корпуса/система «умной защиты и т.д/ локальная сеть аудитории) по принципу – «есть в этом нужда», «необходимо агрегировать все в рамках одной структуры», а сейчас, осознанно определите зависимости - абсолютный аналог смысла «связь» в изучаемых вами СУБД реляционного типа) которые можно связать с организационной структурой предприятия (учреждения), как объекта проектирования сложной сетевой информационной инфраструктуры.

Построенная DFD-диаграмма вряд ли сможет отражать все информационные потоки (а, следовательно, и обмены) внутри системы. Но многое станет явней, в частности, если перед разработкой DFD, разработанную сеть протестировать через симуляцию и Packet Generator в ручном и автоматическом режимах.

На основании умозаключений вы можете сформировать и описать **пул сервисов** (например, под сервисами имеются в виду и такие «простые» и понятные, но как показывает практика, незаслуженно упускаемые в отчетах и схемах, такие как: «ISP предоставляет услугу доступа к глобальной сети» / «одна из нескольких IoT-подсистем реализует делегирование полномочий связанных с контролем энергообеспечения» и т.д)

Затем вы должны реализовать в отчете пункты 7.1-7.2:

7.1. Принципиальная схема организационных подразделений (схема) **(в упрощенном виде)**

7.2. Принципиальная схема организационных подразделений (пояснение) **(в упрощенном виде)**

На **основании проделанной работы** [1 - 7.2] можно формировать стратегию информационной безопасности «Zero Trust», основанную на описанных принципах, на рассматриваемой системе:

Основываясь на **методике по оценке рисков информационной безопасност**и произвести оценку вашей сети (риски/угрозы и весь необходимый теоретический материал вы можете найти в списке источников (например: «банк угроз инф. безопасности») идущей после методологии, и, затем, на финальной стадии**, описать возможности применения «Zero Trust» в рассматриваемом случае.**

Стратегия должна оформляться в виде тезисов, пунктов. Нумерация сплошная.

Обязательно отразить в отчете риски появления новых (явных) угроз безопасности при частичном (неполном) выполнении принципов **«Zero Trust».**

И помните, что ваша оценка (анализ) должна быть максимальной полной:

«...Ведь если своевременно обнаружить зарождающийся недуг, что дано лишь мудрым правителям, то избавиться от него нетрудно, но если он запущен так, что всякому виден, то никакое снадобье уже не поможет.»

Макиавелли «Государь»

§ ПРАКТИКУМ: СЛАУ в IoT-ИНФРАСТРУКТУРЕ

Система линейных алгебраических уравнений (линейная система, также употребляются аббревиатуры СЛАУ, СЛУ) — система уравнений, каждое уравнение в которой является линейным — алгебраическим уравнением первой степени.

В классическом варианте коэффициенты при переменных, свободные члены и неизвестные считаются вещественными числами, но все методы и результаты сохраняются (либо естественным образом обобщаются) на случай любых полей, например, комплексных чисел.

Решение систем линейных алгебраических уравнений — одна из классических задач линейной алгебры, во многом определившая её объекты и методы. Кроме того, линейные алгебраические уравнения и методы их решения играют **важную роль во многих прикладных направлениях**, в том числе в линейном программировании, эконометрике.

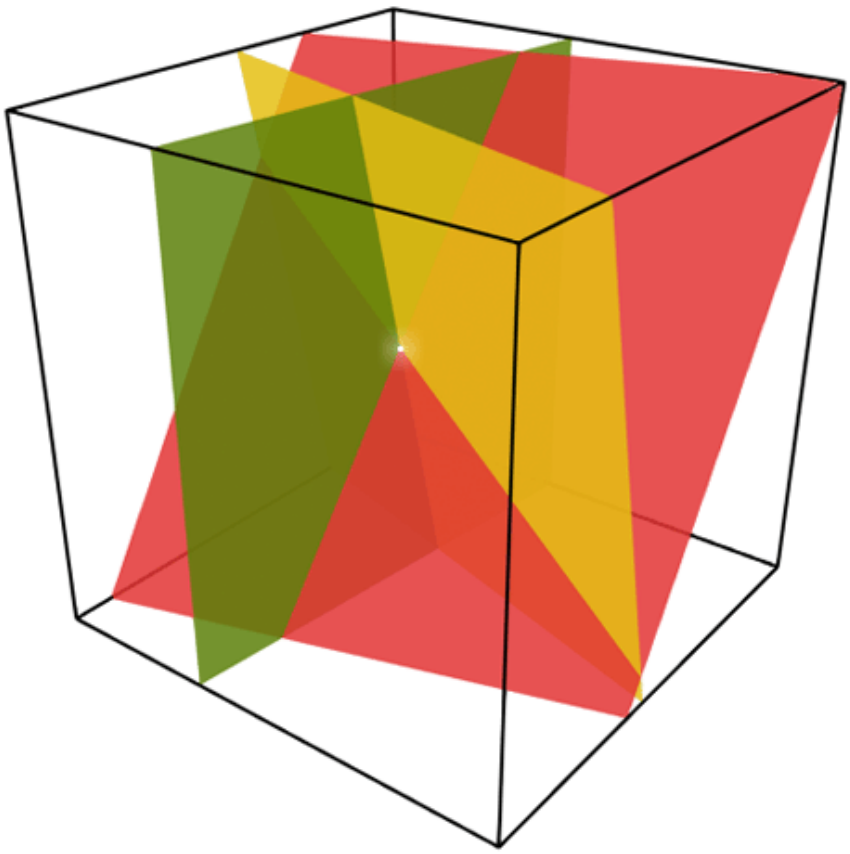


Рисунок 51 – Система линейных уравнений от трёх переменных определяет набор плоскостей. Точка пересечения является решением.

Лабораторная работа №5: Решение задачи IoT при помощи СЛАУ.

Цель работы: изучение методов решения систем линейных уравнений и определение эффективности их использования в решении прикладных задач.

Предмет исследования: способы и алгоритмы решения систем линейных алгебраических уравнений.

Методы исследования: теоретическое изучение материала и практическое решение задач.

Задача:

Для построения заданной PAN (Personal Area Network) сети необходимо 360 устройств с полной функциональностью(А), 300 устройств с малой функциональностью (Б) и 675 конечных устройств (В).

Применяются три способа коммутации.

При первом способе коммутации получается сеть второго ранга с 3 устройствами типа А, 1 устройствами типа Б и 4 устройствами типа В, при втором способе получается сеть второго с 2 устройства типа А, 6 – типа Б и 1 - типа В, при третьем способе получается сеть второго ранга, в которую входит: 1 устройство типа - А, 2 - типа Б и 5 устройства типа В.

Найти затраты устройств при каждом из указанных способов коммутации.

Типовое решение:

Условие задачи запишем в виде таблицы.

Тип устройства	Количество, по способам коммутации			Необходимое количество устройств
	I способ	II способ	III способ	
А	3	2	1	360
Б	2	6	2	300
В	4	1	5	675

Обозначим через x, y, z количество устройств, «раскраиваемых» соответственно первым, вторым и третьим способами. Используя данные таблицы запишем систему:

$$3x + 2y + z = 360,$$
$$x + 6y + 2z = 300,$$
$$4x + y + 5z = 675.$$

Система уравнений – это математическая модель условия выполнения всего задания по коммутациям А, Б и В.

Решить систему трех линейных уравнений с тремя неизвестными можно любым известным методом (матричным, Крамера, Гаусса).

Матрица системы квадратная, если ее определить не равен нулю, то данную систему можно **решить методом Крамера.**

Варианты заданий задаются через переменные: N [1..9] M [1..9] H [1..9] K [1..30] J [1..30] Y [21..40]

G [100..999] T [100..999] Y [100..999]

Тип устройства	Количество, по способам коммутации			Необходимое количество устройств
	I способ	II способ	III способ	
А	N	M	K	G
Б	H	J	M	T
В	M	H	Y	Y

Требования к отчету:

1. Алгоритм выдачи вариантов через предопределение посредством рандомайзера (**обязательна программная реализация** для всех переменных в заданных диапазонах допускается на любом языке высокого уровня с листингом программы, скриншотами результата компиляции). Относительная уникальность вашей комбинации для переменных достигается за счет скрининга (особенности генератора псевдослучайных величин).

2. Ручной просчет методом Крамера (или, если необходимо, любым другим способом.

Лабораторная работа №6

Цель работы: Поиск кратчайшего пути в графе с помощью алгоритма Дейкстры. Решение типовой задачи о соединении городов (алгоритмы Прима и Краскала).

Таким образом, лабораторная состоит из двух задач. Варианты заданий формируются сходным способом, как и в лабораторной работе №5:

Вариант задания для Задачи «Поиск кратчайшего пути в графе с помощь алгоритма Дейкстры». В формальном виде вам необходимо сформировать через генератор псевдослучайных чисел матрицу чисел в диапазоне от 1 до N, где N = {количество букв вашего полного имени в инфинитиве + количество букв вашего отчества в родительском падеже * целое число, являющееся номером положения первой буквы вашей фамилии в русском алфавите (например, если ваша фамилия начинается с буквы Г - то 4}. А сам просчет оформить по инструкции в ручном виде, обязательно продемонстрировав сформированную матрицу (алгоритм и скриншот, который будет аргументов освоения материала и являться доказательством истинности «входных данных»).

Вариант задачи с алгоритмами Прима и Краскаля не требует от вас автоматической генерации значений, исходный граф предоставляется преподавателем. Просчет ручной, итерационный (как в методологии).

Вся методология и дидактический материал предоставлен посредством интер-отклика:

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Наталья Мещерякова, НИУ ВШЭ
Материалы к семинарам по Дискретной математике

§ ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

Выпускные квалификационные работы (ВКР) студентов, получающих специальность «информатик-экономист» «информатик», «специалист по специализированным компьютерным системам/сетям», могут иметь как научно-исследовательскую направленность (НИР, когда работа связана с исследованием характеристик экономических процессов, протекающих в социально-экономических системах, проведением экономико-математического моделирования с применением специализированных пакетов прикладных программ и т. п.) и опытно-конструкторский характер (ОКР, когда большой удельный вес в работе составляет разработка новых программных приложений и программных комплексов, баз данных, автоматизированных ин-формационных систем, ориентированных на решение экономических задач и т.п.).

Независимо от направленности и характера ВКР при ее выполнении необходимо осуществить экономическое обоснование принимаемых решений, поскольку одним из важных критериев прогрессивности создаваемых образцов и видов научно-технической продукции является экономическая эффективность соответствующих решений.

Экономическое обоснование выполняется в форме бизнес-плана, являющегося основным системным документом реализации нового проекта, или технико-экономического обоснования.

Технико-экономическое обоснование (ТЭО) – это анализ, расчет, оценка экономической целесообразности осуществления предлагаемого проекта, в данном случае – проекта по разработке и исследованию экономико-информационной системы.

ТЭО основано на сопоставительной оценке затрат и результатов, установлении эффективности использования, срока окупаемости вложений.

Технико-экономическое обоснование является необходимым для каждого инвестора исследованием, в ходе подготовки которого проводится ряд работ по изучению и анализу всех составляющих инвестиционного проекта и разработке сроков возврата вложенных в бизнес средств.

Технико-экономическое обоснование проекта имеет много общего с бизнес-планом. Отличия заключаются в следующем:

обычно ТЭО пишется для проектов внедрения новых технологий, процессов и оборудования на уже существующем, работающем предприятии, поэтому анализ рынка, маркетинговая стратегия, описание компании и продукта, а также анализ рисков в нем часто отсутствуют;

в ТЭО приводится информация о причинах выбора предлагаемых технологий, процессов и решений, принятых в проекте, результаты от их внедрения и экономические расчеты эффективности.

Следовательно, можно говорить о более узком, специфическом характере ТЭО по сравнению с бизнес-планом.

Экономическая часть любого проекта, содержащая технико-экономическое обоснование, должна:

являться логическим продолжением основной части ВКР (как пример);

быть связана с профилем специальности студента;

быть достаточно современной, актуальной; содержать в себе новые методические положения, действующие расценки, нормативы, рыночные ориентиры.

Все расчеты должны сопровождаться соответствующими пояснениями, ссылками на источники получения исходных данных.

Формулы должны приводиться с расшифровкой условных обозначений. В расчетах следует использовать текущие рыночные цены и тарифы на продукцию, работы, услуги, сырье, действующие на момент разработки проекта, курсы иностранных валют для пересчета валютной выручки и цен в иностранной валюте.

Большинство расчетов целесообразно представлять в табличной форме в соответствии с требованиями действующих стандартов. Основные результаты и расчеты экономического раздела могут быть представлены в презентации для доклада на защите ВКР.

Состав расчетов технико-экономического обоснования, выполняемых в экономической части ВКР, включает следующие положения:

1) обоснование целесообразности разработки проекта;

2) оценка уровня качества разрабатываемого программного продукта;

3) организация и планирование работ по разработке проекта;

4) расчет затрат на разработку проекта;

5) расчет эксплуатационных затрат;

6) оценка эффективности разработанного проекта.

В первом пункте необходимо отразить актуальность, необходимость и значимость проведения исследований, изложить цели, задачи и специфические особенности выполняемого исследования.

Во втором пункте необходимо описать базовый вариант. Выбор базового варианта производится совместно студентом и руководителем ВКР на основе патентного поиска, обзора литературы по заданному направлению, анализа информации, найденной в Интернет и других источниках.

После выбора базового варианта (выдается преподавателем на основании проделанной работы по практикуму) необходимо провести анализ и сравнение с разрабатываемым продуктом по показателям качества. Показатели качества могут быть различными в зависимости от поставленной экономико-информационной задачи.

В третьем пункте, прежде всего, необходимо правильно установить оптимальный объем работ по теме и разбить работы по этапам. Наиболее ответственной частью работ в этом пункте является расчет трудоемкости отдельных видов проводимых работ, так как трудозатраты составляют основную часть стоимости научно-исследовательских и опытно-конструкторских работ (НИОКР). Общее количество дней, затрачиваемое на все работы, должно быть равно времени, отводимому на преддипломную практику и дипломирование. В некоторых случаях (в зависимости от сложности проекта) работа над программным продуктом (которым может стать разработанный вам проект управления IoT-инфраструктуры) начинается задолго до преддипломной практики, и это следует учесть в расчетах.

Загрузка в днях у основного исполнителя (выполняющего функции программиста-разработчика) должна быть равна расчетной ожидаемой длительности времени. В связи с этим, для него загрузка в процентах по каждой работе равна 100.

Так как руководитель помогает студенту и консультирует его по различным вопросам, то его время также должно быть учтено для расчета затрат на оплату труда. Время руководителя определяется исходя из фактических или предполагаемых затрат.

Загрузка в процентах у руководителя может быть определена следующим способом.

Предположим, что постановку задачи осуществляет одновременно руководитель и студент. Студент расходует на эту работу 3 дня, и все это время занят только своей работой, а руководитель может в течение этих дней уделять данной проблеме только треть своего рабочего времени каждый день. Тогда его загрузка будет равна 33% и для оплаты следует принять только один день (33 % от трех дней).

Важно верно определить величину заработной платы руководителя и разработчика программы.

Руководителем разработки может быть профессор, доцент, старший преподаватель, ассистент, аспирант или специалист, работающий на конкретном предприятии. Необходимо в расчеты закладывать их основную и дополнительную заработную плату. Аналогично определяется оклад программиста-разработчика.

Необходимо также учитывать районный и северный коэффициенты при расчете дополнительной заработной платы.

В пятом пункте производится расчет эксплуатационных (текущих) затрат. К ним относятся затраты, связанные с использованием программного продукта в течение первого года эксплуатации.

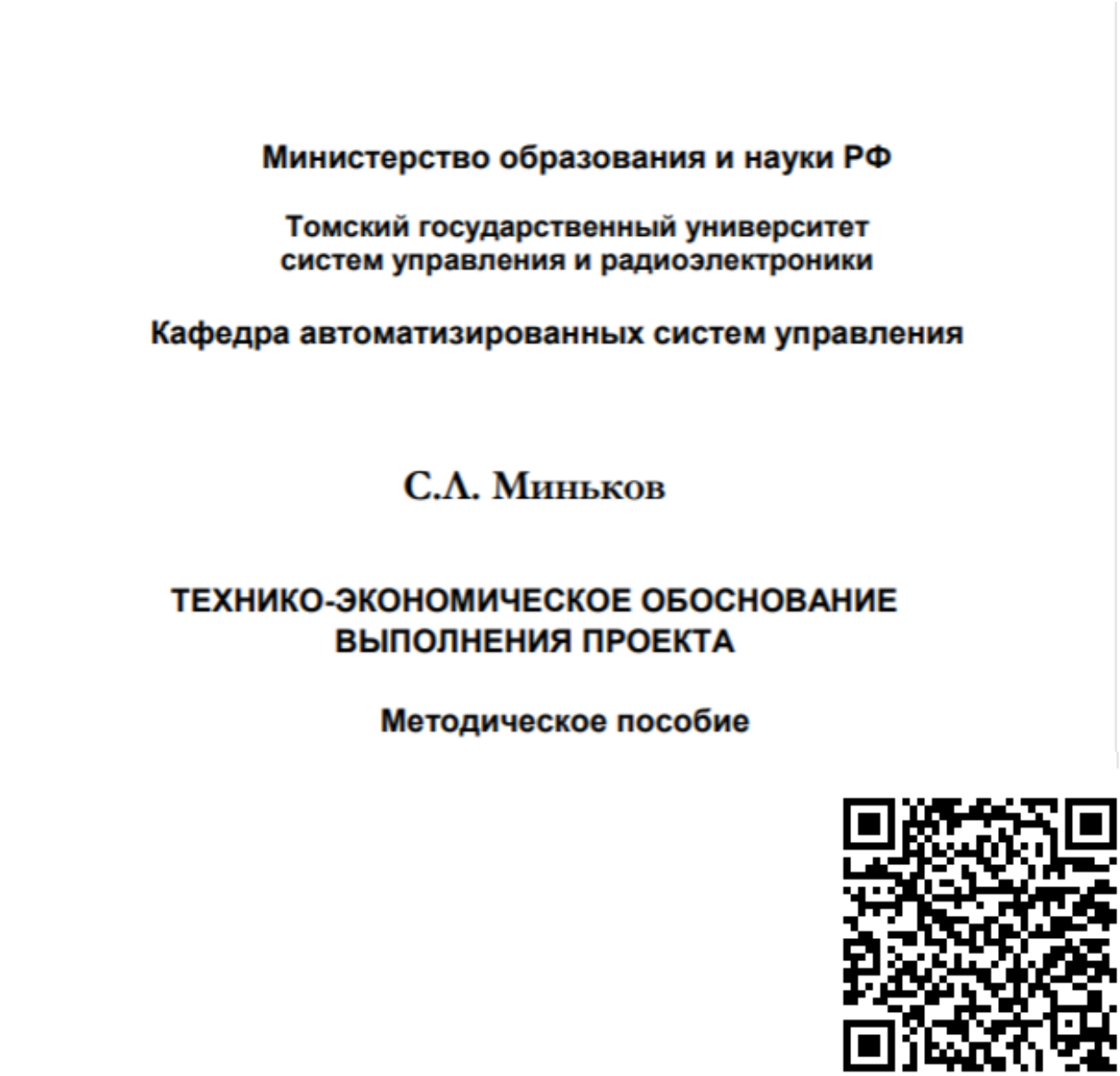
Здесь важно определить время использования программного продукта. Предположим, что созданный продукт будет использоваться двумя пользователями. Один из них будет работать с программой два часа в день. Рабочая неделя включает пять рабочих дней. В году 52 недели. Исключаем 104 дня (52*2) выходных плюс 12 праздничных дней. Итого 116 дней. Остается 249 дня. Умножаем на 2 час и получаем 498 час. Если рабочий день равен 8 часам, то общее число полных рабочих дней для пользователя в году будет равно 62,25 (498 час. / 8 час.). Аналогично производим расчет по второму пользователю.

Обычно разрабатываемый продукт позволяет ускорить время выполнения работ. Поэтому время пользования продукта-аналога немного больше, чем у нового продукта. Допустим, что это время составляет 70 дней. Появляется экономия во времени, и эта экономия прослеживается по всем статьям расходов.

В шестом пункте на основе результатов выполнения четвертого и пятого пунктов рассчитываются годовой экономический эффект, фактический коэффициент экономической эффективности разработки, срок окупаемости затрат на разработку проекта.

Сама методология представлена посредством интер-отклика (с примером реализации ТЭО программного продукта информационной системы):

Методическое пособие предназначено для студентов высших учебных заведений, выполняющих выпускную квалификационную работу по специальности 09.03.03 «Прикладная информатика в экономике».


Министерство образования и науки РФ

Томский государственный университет
систем управления и радиоэлектроники

Кафедра автоматизированных систем управления

С.Л. Миньков

ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ
ВЫПОЛНЕНИЯ ПРОЕКТА

Методическое пособие

Технико-экономическое обоснование выполнения проекта: методическое пособие / С.Л. Миньков. – Томск: ТУСУР, 2014. – 30 с.

§ NODE-RED - ГРАФИЧ. КОНФИГУРАТОР ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ

Одним из наиболее важных и сложных вопросов при запуске системы автоматизации (умного дома) является процесс первоначальной конфигурации системы и её отладки. Сложность заключается в том, что необходимо написать значительную часть в текстовом редакторе. При этом разработчик остаётся практически один на один с задачей, т.к. почти не существует средств автоматизации для данного этапа. В то же время система умного дома сложная и многоуровневая по своей структуре, что также усложняет процесс, т.к. необходимо учесть множество факторов одновременно. Отсутствие подобного инструментария стоит на пути массового внедрения систем умного дома у большинства пользователей, т.к. порог входа в данном случае достаточно высок. Решением в данном случае является специализированная среда разработки, которая позволяет упростить некоторые из типовых задач.

Для решения этой задачи был разработано несколько средств, один из которых - Node-RED, позволяющий через браузер построить схему взаимодействия устройств между собой и внешними системами.

Данное решение удобно как промежуточное для связи устройств различного типа между собой и/или же с системой автоматизации или, например, СУБД или иным облаком. С использованием дополнительных пакетов Node-RED можно использовать для создания простых систем автоматизации умного дома, но решение будет относительно ограниченных ввиду неполного покрытия функциональных потребностей умного дома.

Node-RED работает на Node.JS, и был разработан для работы на относительно малопроизводительных системах, таких как: Raspberry Pi; BeagleBone; Arduino.

С учётом озвученных факторов Node-RED удобно использовать на шлюзах между различными сетями устройств интернета вещей функционирующих на собственных, как правило, более простых протоколах и традиционным интернетом, построенных на TCP/IP, UDP. В этом случае он позволит более оптимально использовать свободные ресурсы шлюза, работающего, как правило, на Linux.

Лабораторная работа №7

ПРАКТИКУМ: СОЗДАНИЕ ПЕРВОГО IoT-приложения

Цель работы: создание IoT-приложения для мониторинга удаленных серверов или распределенной сети.

Оборудование и инструменты: ПК, ОС Windows 10, веб-браузер с поддержкой WebKit; Flow-based Development tool for Visual Programming: Node-RED; облачная коммуникационная платформа Twilio; и облачная коммуникационная платформа с поддержкой function-as-a-service IBM Bluemix.

Задание №1

С помощью инструмента Node-RED и сервиса Twilio создаем на платформе Bluemix приложение для мониторинга удаленных серверов или распределенной сети.

Вы создадите полное IoT-решение, которое осуществляет мониторинг удаленного сервера, а затем уведомляет пользователя, если его компьютер или его сеть не функционирует.

Методология размещена посредством интер-отклика:

«Создаем первое IoT-приложение»
Мохамед Хассан
Дина С Ахмед

Задание №2

Bluemix — мощная платформа, которая позволяет быстро и без проблем создавать, разворачивать и администрировать приложения в облаке. Bluemix — это реализация архитектуры **IBM Open Cloud Architecture** на основе открытого ПО Cloud Foundry, работающего по принципу «платформа как услуга» (**Platform as a Service – PaaS**).

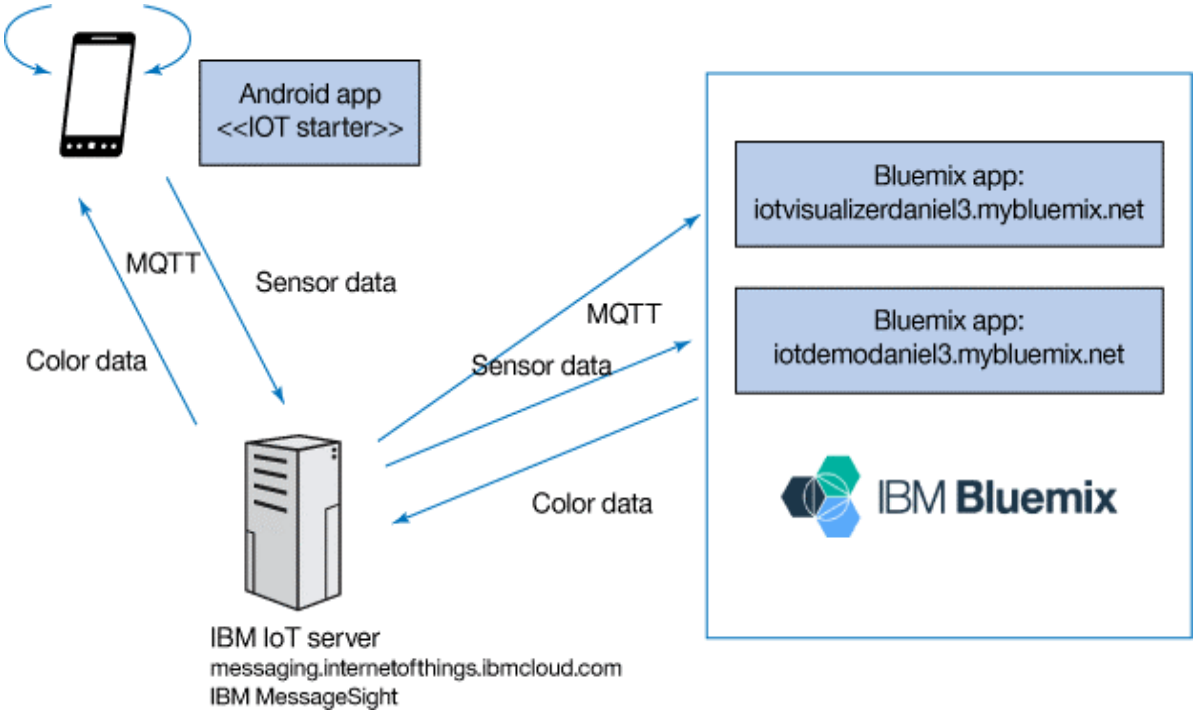
При помощи этой платформы и службы IBM Watson IoT Paltform разработать Bluemix-приложение (**благодаря пошаговой методологии**), которое позволит обрабатывать, визуализировать и сохранять данные, полученные со смартфона.

Что необходимо для разработки приложений?

1. Учетная запись в Bluemix (есть возможность регистрации для бесплатного пробного использования).

2. Также необходимо загрузить и установить интерфейс командной строки Cloud Foundry.

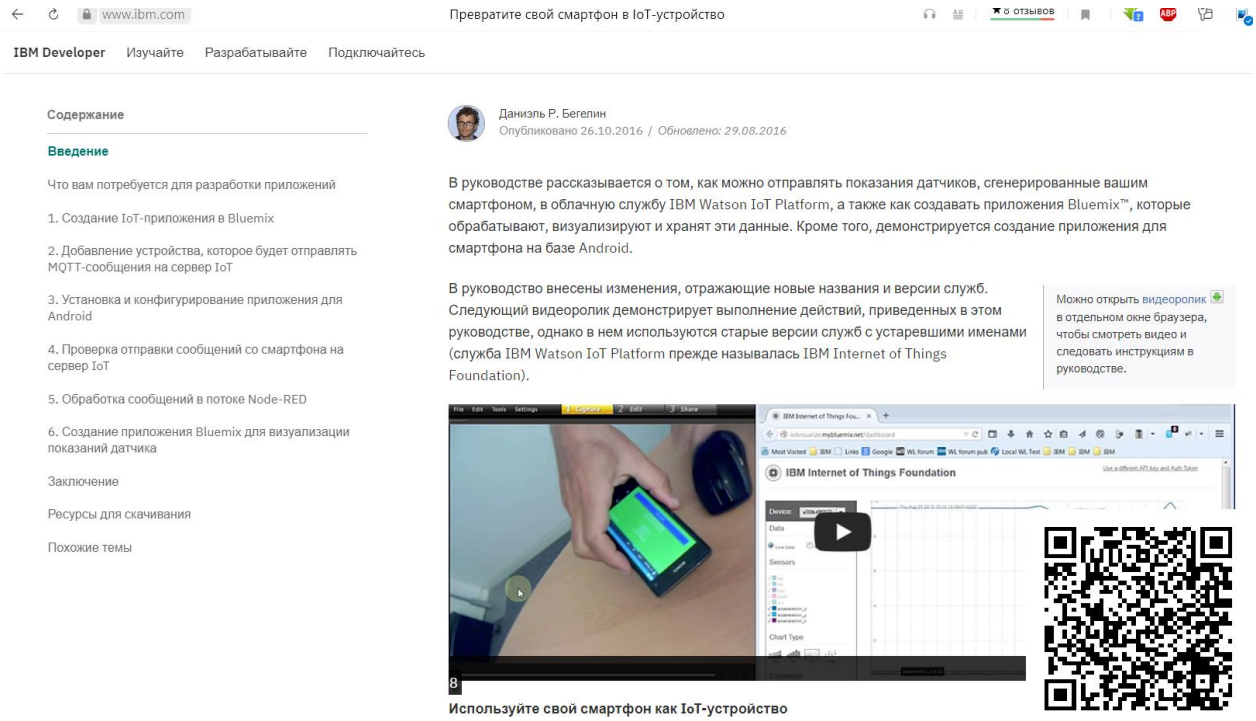
3. Смартфон с ОС Android или iOS.



В руководстве, по ссылке, размещенной в QR-коде, рассказывается о том, как можно отправлять показания датчиков, сгенерированные вашим смартфоном, в облачную службу IBM Watson IoT Platform, а также как создавать приложения Bluemix, которые обрабатывают, визуализируют и хранят эти данные. Кроме того, демонстрируется создание приложения для смартфона на базе Android.

Отчетность по факту выполнения задания в виде публикации видео-захвата экрана вашего компьютера на платформе youtube, в которой демонстрируется факт развернутая панель управления (общий хронометраж видео (минимальные требования к видеоролику: 720p, 30 fps) не должен составлять более 2х минут для каждого задания в этой лабораторной. Индивидуальные задания (дополнения) выдаются преподавателем.

Методология предоставлена посредством интер-отклика с портала IBM Developer



Создание приложений Bluemix для обработки, визуализации и хранения показаний датчиков, отправляемых и получаемых смартфоном

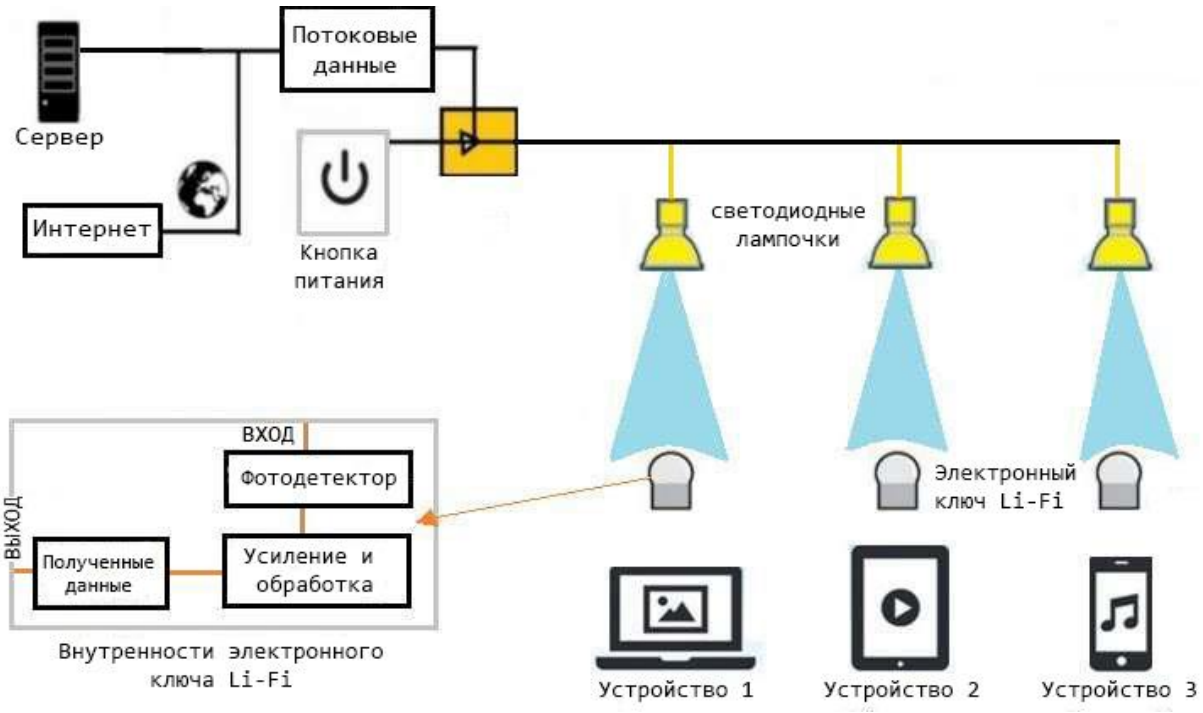
Даниэль Р. Бегелин

§ Li-Fi: БУДУЩЕЕ ИНТЕРНЕТА*

Представьте себе время, когда каждая из лампочек в вашем доме будет источником интернета. Вообразите сценарий, когда, простояв под лампочкой в течение лишь одной минуты, вы скачали бы около 5 фильмов в формате HD. Звучит круто, верно? Но благодаря технологии Li-Fi эта мечта может стать реальностью. С помощью данной технологии мы можем переосмыслить роль света как такового.

Li-Fi — это система связи видимого света (VLC), которая использует свет для отправки беспроводных данных, встроенных в его луч. Устройство с поддержкой Li-Fi преобразует луч света в электрический сигнал. Затем сигнал преобразовывается обратно в данные. Этот термин был придуман немецким физиком Харальдом Хаасом (Harald Haas) во время TED Talk в 2011 году. Он предвидел идею использования лампочек в качестве беспроводных маршрутизаторов.

Лампы Li-Fi оснащены чипом, который незначительно модулирует свет для оптической передачи данных. Данные передаются бытовыми светодиодными (LED) лампами и принимаются фоторецепторами.



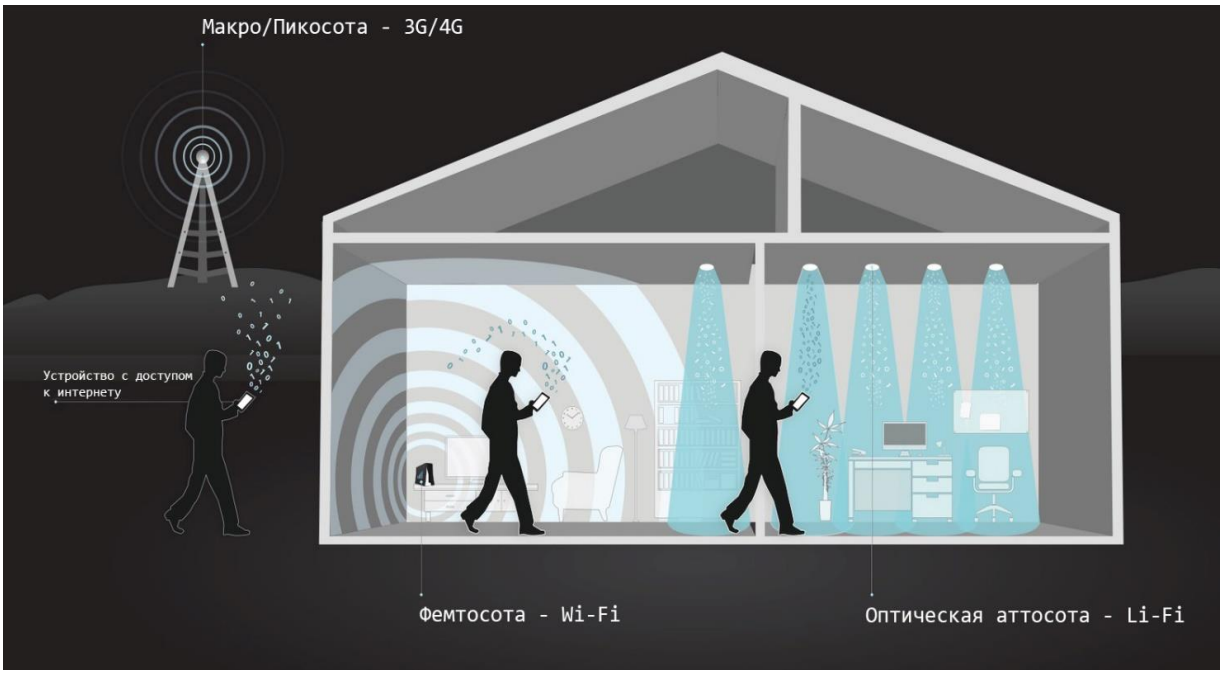
При детальном внедрении системы, Li-Fi может достигать скоростей передачи, которые примерно в 100 раз превышают современный традиционный Wi-Fi, работающий на радиоволнах (т.е. скорость может достигать более 1 гигабита в секунду).

Как это работает?

Светодиодные лампочки можно диммировать (регулировать по мощности) на очень высоких скоростях, неразличимых человеческому глазу. Короткие импульсы при быстром диммировании LED ламп затем преобразуются «приемником» в электрический сигнал. После этого, сигнал преобразуется обратно в поток двоичных данных, который мы получаем в виде веб-, видео- и аудиофайлов, на наших устройствах с выходом в интернет.

Достоинства и недостатки по сравнению с Wi-Fi (IEEE 802.11):

Наиболее отличительной особенностью Li-Fi является то, что в отличие от Wi-Fi, она не интерферирует с радиосигналами, что ставит ее в более выигрышные позиции с точки зрения стабильности скорости интернета. Это еще без учета той огромной разницы в скоростях двух видов сравниваемых сетей.



К недостаткам стоит отнести: расстояние покрытия Li-Fi составляет 10 метров, в то время как для Wi-Fi — 32 метра.

Помимо этого, технология Li-Fi не может быть развернута на улице при солнечном свете или в любых нестабильных условиях, она не может работать в темноте при отсутствии светодиодных ламп

Области применения:

1. Военная промышленность

Покрытие Li-Fi может быть ограничено небольшой освещенной областью, например, такой, как палатка. Таким образом, это может ограничивать доступ к конфиденциальной информации при определенных условиях и в тех местах, где мобильные телефоны не могут быть использованы, например, на складах боеприпасов.

2. Интернет вещей

Благодаря своей впечатляющей скорости, Li-Fi может оказать огромное влияние на Интернет вещей. Учитывая то, что данные передаются на гораздо более высоком уровне, еще большее число подключенных к интернету устройств смогут взаимодействовать друг с другом.

3. Информационная безопасность

У Li-Fi радиус действия меньше, чем у Wi-Fi, и поэтому он более безопасен в этом плане. Хотя этот параметр и был учтен в минусах, стоит отметить, что с точки зрения безопасности передачи данных, меньший радиус действия можно рассматривать и как положительную сторону. Это может быть очень полезно в отраслях, которые обрабатывают большое количество конфиденциальных данных, например, в здравоохранении.

Будущее Li-Fi

В скором времени, каждое из наших устройств, будет постоянно подключено к интернету, поскольку мы вступаем в т.н. эру «Интернета Всего» (**IoE**). Справится ли Wi-Fi с задачей обработки всего этого интернет-трафика в одиночку? Не думаю.

Учитывая постоянно растущий спрос на средства связи, технология Li-Fi имеет хорошие шансы на скорое внедрение, т.к. сможет сочетать освещение и беспроводную передачу данных.

Компания, основанная профессором Геральдом Хаасом в 2012 году, известная как pureLifi, проводит эксперименты и активно исследует достижения в этой области. Стартап Velmenni, находится на передовой этой технологической революции в Индии.

По моему мнению, эта технология имеет достаточный потенциал стать повсеместной, так что будьте готовы к ней…

ПРИЛОЖЕНИЕ. Обзор открытых IoT-платформ

Интернет вещей (IoT) продолжает активно расти и развиваться, при этом ему приходится сталкиваться с обычными для молодого рынка проблемами — противостоянием отраслевых организаций, за спиной которых обычно скрывается та либо иная группа техногигантов, продвигающих свои протоколы и стандарты коммуникаций. Впрочем, в последние годы наметилась тенденция к ослаблению «проприетарной хватки» — производители техники и даже потребители начали отдавать предпочтение открытым проектам.

Нужно заметить, в таком решении присутствует рациональное зерно, поскольку Open Source-проекты доказали свою живучесть, тогда как многие компании с проприетарными разработками бесследно исчезли, прихватив заодно свою интеллектуальную собственность.

Издание Linux.com опубликовало подготовленный Эриком Брауном обзор основных открытых проектов, которые работают над созданием софта для «умного» дома и промышленной автоматизации (в предыдущем обзоре можно было ознакомиться с открытыми RTOS и Linux OS для устройств IoT). В нем рассматривается больше двадцати открытых проектов, два из которых — AllSeen (AllJoyn) и OCF (IoTivity) — курируются Linux Foundation, но все они преследуют одну цель — создание фреймворков для работы конечных IoT-устройств типа датчиков или сенсоров с сетевыми шлюзами и облачными сервисами.

Обзор помимо значительных IoT-проектов включает и нишевые. К слову, становится все труднее провести границу между ПО для IoT и обычным программным обеспечением. Нужно также отметить, что все выбранные проекты работают по модели Open Source, созданы на ядре Linux или позаимствовали у этой ОС один или несколько компонентов. Большинство фреймворков используют Linux для построения собственной экосистемы — начиная с рабочего окружения на ПК и заканчивая облачными службами для управления сетевыми шлюзами и датчиками. Выпуск адаптированного ПО для Raspberry Pi, BeagleBone, Arduino — ещё одна отличительная черта почти всех проектов.

Конечно, в этой сфере есть мощные закрытые (запатентованные) технологии, среди которых такие корпоративные платформы, как Apple HomeKit для «умного» дома или облачная IoT-платформа для построения SaaS-приложений Microsoft Azure IoT Suite. Но даже такие платформы предлагают частично открытый доступ к коду либо инструментам разработки. Примером такого фреймворка является Verizon ThingSpace — он создан для написания приложений для управления «умным» городом. Его ядро — проприетарный продукт, но открытый API позволяет запускать ThingSpace на одноплатных компьютерах. Amazon AWS IoT — ещё один IoT-проект корпоративного уровня с частично открытым комплектом средств разработки.

Частично открытым можно признать и проект Thread Group. Его запуском ведала компания Nest, впоследствии купленная Google. Thread Group отвечает за разработку маломощной, безопасной и масштабируемой беспроводной mesh-сети на базе протокола 6LoWPAN. Что касается открытости, то ей соответствуют такие проекты Google, как Brillo или протокол обмена данными между IoT-устройствами Weave. В мае Nest выпустила открытую версию Thread — OpenThread.

Open source для IoT

AllSeen Alliance (AllJoyn). AllJoyn — открытая программная архитектура, позволяющая IoT-устройствам и приложениям взаимодействовать друг с другом. При этом речь идёт не только о взаимодействии цифровых устройств в рамках одной операционной системы, посредством протокола AllJoyn «говорить» между собой на одном языке могут устройства на разных платформах, например, Windows и Android, к тому же AllJoyn предусматривает их подключение к бытовой технике.

AllSeen Alliance насчитывает 160 участников, среди которых Qualcomm, Lenovo, LG, Symantec, Sony и Panasonic. Совместными усилиями они создают свой набор ПО на основе кода AllJoyn, который разработали и передали группе инженеры Qualcomm.

AllJoyn можно назвать наиболее распространённым фреймворком Open Source. В октябре AllSeen Alliance объединился с организацией Open Connectivity Foundation (OCF).

Рабочая группа создаёт инструментарий для создания приложений и сервисов, которые могут автоматически группироваться в P2P-сеть, подключаясь к соседним IoT-устройствам по Wi-Fi или Bluetooth.

Bug Labs dweet/freeboard. Изначально (с 2007 г.) проект выпускал различного рода модульные компьютеры в стиле Lego. Электронные конструкторы состояли из базы BUGbase (в нее встроен процессор, чип Wi-Fi, Ethernet-контроллер, интерфейс USB, небольшой ЖК-экран и аккумулятор) и дополнительных элементов «периферии» — GPS-приёмника, сенсорной панели, датчика движения и пр. По мере развития аппаратных возможностей развивалась и софтверная направляющая Bug Labs, вылившись в итоге в IoT-платформу для бизнеса. В основе Bug Labs dweet/freeboard лежит модуль ПО dweet.io. Этот инструмент, по сути, представляет собой сервис для обмена сообщениями для всего, что может подключаться к Интернету.

При подключении устройства к сервису dweet.io, эта платформа увидит его и присвоит ему имя, а затем начнет пересылать данные или отправлять «двиты» (твиты между IoT-устройствами) в облако. В то же время сервис Freeboard, работающий как информационная панель на смартфоне, может отслеживать состояние окружающей среды, следить за безопасностью и потреблением электроэнергии в доме. Эти данные он получает из «двитов». Например, датчик влажности, встроенный в коробку для хранения сигар, может регулярно поставлять данные о влажности, что позволит сохранить сигары в пригодном для употребления состоянии.

DeviceHive. Эта платформа использует сетевые и облачные технологии и позволяет реализовывать решения для обмена информацией между устройствами по модели M2M (Machine-to-Machine, межмашинное взаимодействие). Инфраструктуру DeviceHive разработала компания DataArt, в её основе лежит протокол AllJoyn. Хостинг платформы может осуществляться на таких сервисах, как Azure, AWS, Apache Mesos или OpenStack. Облачная платформа DeviceHive поддерживает Apache Spark и Spark Streaming, что позволяет запускать пакетную обработку поверх данных устройств, отслеживать процессы в реальном времени и использовать машинное обучение.

DeviceHive 2.0 была дополнена шинами передачи сообщений (Apache Kafka) и узлов хранилища (Cassandra). IoT-фреймворк DeviceHive интегрирован с Ubuntu Snappy Core, что превращает этот Linux-дистрибутив в модульную платформу. Такая трансформация позволяет запускать специфические приложения в облаке DeviceHive, а адаптеры — подключать к низкоуровневой аппаратуре и проксимальным сетям.

DSA. Фреймворк Distributed Switch Architecture (DSA) предоставляет возможность управления разветвленной сетью IoT-девайсов как единым устройством, задавать пути прохождения, контролировать логику трафика и работу приложений в системе. DSA отвечает за разработку библиотеки Distributed Service Links (DSLinks), необходимой для трансляции протоколов и интеграции данных со сторонними источниками. DSA предлагает масштабируемую сетевую топологию, состоящую из нескольких DSLinks-библиотек, работающих на конечных IoT-устройствах. Суть технологии DSA можно кратко свести к введению дополнительного уровня адресации в рамках узлов/устройств всей сети.

Eclipse IoT (Kura). Некоммерческая организация Eclipse Foundation (EF) — ещё один игрок на поле IoT. В качестве инструментов разработки сообщество применяет Java-ориентированный движок Kura — он требуется для создания IoT-шлюзов и оснащен гибким пользовательским интерфейсом на базе Bootstrap, упрощающим управление шлюзами с мобильных устройств. Kura может взаимодействовать с фреймворком Apache Camel с целью обеспечить простую маршрутизацию сообщений в рамках бизнес-логики приложений. В рамках EF на базе спецификации OSGi развивается модульная платформа OM2M. В ней реализован облегченный REST API, к которому можно подключиться через множество сетевых привязок, включая протоколы HTTP и CoAP, платформа поддерживает различные форматы контента, например, XML и JSON.

EF также развивает субпроект Paho и фреймворк SmartHome. Первый основан на стандарте OASIS MQTT (Message Queue Telemetry Transport). MQTT — нетребовательный к ресурсам протокол обмена сообщениями, хорошо подходящий для подключения небольших устройств к Интернету. Eclipse Paho — реализация этого протокола со стороны клиента. Eclipse SmartHome — это фреймворк для проектирования технических решений для «умных» домов с упором на среды с разнородным оборудованием.

Kaa. Проект компании CyberVision предлагает масштабируемую, с возможностью сквозного обеспечения услуг платформу для высоконагруженных, подключенных к облаку IoT-сетей. Она включает клиент-серверную архитектуру REST (Representational State Transfer, передача репрезентативных состояний) для развёртывания сервисов, решения аналитических задач и управления данными.

Координация кластерных узлов производится на базе Apache Zookeeper. Комплект SDK Kaa включает Java, C++ и Си. Он позволяет гибко регулировать межпроцессорное клиент-серверное взаимодействие между программами, настройки аутентификации, шифрования, а также хранение и сортировку данных. Пакет SDK включает графические схемы для обвязки специфического для IoT кода. Эти схемы определяют семантику и абстрактные функции различных групп устройств в подключённой сети.

Macchina.io. Проект предоставляет веб-ориентированное, модульное и расширяемое окружение на JavaScript и Си для разработки сетевых приложений для IoT, работающих на одноплатных компьютерах. Macchina.io поддерживает широкий набор датчиков и технологий подключения, в том числе микроплаты Tinkerforge, сенсоры XBee ZB, ресиверы Global Navigation Satellite System (GNSS)/GPS, серийные и подключаемые GPIO-девайсы, акселерометры.

GE Predix. Predix — это PaaS-сервис промышленного гиганта General Electric, созданный для большого объема именно индустриальных данных и аналитики. Технология работы Predix предусматривает непосредственное подключение промышленных установок и систем управления технологическими процессами к Интернету через облако, в котором исполняются приложения реального времени по обработке огромного количества данных. В 2017 г. станет доступен для коммерческой реализации Predix на Azure. GE и Microsoft планируют интегрировать Predix с Azure IoT Suite и Cortana Intelligence Suite, а также с приложениями Microsoft для бизнеса, такими как Office 365, Dynamics 365 и Power BI, чтобы соединить промышленные данные с бизнес-процессами и аналитикой.

Интересные подробности о Predix: платформа проводит профилактическое обслуживание 35 000 реактивных авиадвигателей в реальном времени: они передают данные в контрольные центры, где на базе индустриальной платформы разработаны стандартные модели поведения оборудования. Если обнаруживаются еле заметные расхождения в поведении объекта от расчетов по модели, аналитические приложения позволяют спрогнозировать поломку задолго до ее возникновения.

Home Assistant. Система домашней автоматизации работает на Python 3, объединяет все домашние устройства в единую сеть и позволяет управлять ими как традиционным образом — с помощью выключателей, так и с экрана смартфона, планшета или компьютера, из любой точки планеты. Home Assistant может взаимодействовать с роутерами OpenWrt, Tomato, Netgear, DD-WRT, а также Google Chromecasts, Music Player Daemon, Logitech Squeezebox и др.

Mainspring. Это Java-ориентированный фреймворк M2MLabs для обслуживания M2M-коммуникаций, включая удаленный мониторинг и телеметрию различного оборудования, управление «умными» электросетями и парком локального оборудования: контроллерами, климатическим оборудованием, системами контроля энергоснабжения, СКУД и видеокамерами, многочисленными датчиками и др. Работу фреймворка дополняет веб-сервис на базе REST — он требуется для конфигурирования девайсов и настройки инструментов.

Node-RED. Одним из наиболее важных факторов, ограничивающих развитие IoT, является отсутствие удобных средств разработки правил взаимодействия устройств IoT. Для решения этой задачи был разработан фреймворк Node-RED, позволяющий через браузер построить схему взаимодействия устройств между собой и с внешними системами и распределить IoT-узлы. Используя JSON, отдельные узлы можно быстро развёртывать в готовую среду исполнения. Node-RED может работать на Linux-платах, с его помощью производится обмен данными с сервисами Docker, IBM Bluemix, AWS и Azure.

Open Connectivity Foundation (IoTivity). IoTivity — это открытый фреймворк для обеспечения бесшовного связывания различных устройств в соответствии с концепцией Интернета вещей.

Проект распространяется под свободной лицензией Apache 2.0. Обеспечена интеграция с платформами Windows, Ubuntu, Arduino, Tizen, Android, OS X и iOS. Цель IoTivity — стать эталонным стандартом IoT. За продвижение IoTivity отвечает консорциум Open Interconnect Consortium (создан в июле 2014 г., включает более 50 участников).

Стандарт обеспечивает решение таких задач, как обнаружение устройств, управление устройствами, организация передачи данных, аутентификация, разграничение доступа и управление данными. IoTivity не привязан к конкретным технологиям организации канала связи и может использовать такие каналы передачи данных, как Bluetooth, WiFi Direct, ZigBee, Z-Wave и ANT+.

openHAB. openHAB предлагает структуру, основанную на спецификации динамической плагинной шины для создания приложений (OSGi). openHAB имеет модульную архитектуру, что позволяет участникам добавлять устройства, в том числе на основе устаревших протоколов и поддерживает компоненты для «умных» домов, создавая решение, которое позволяет использовать единые пользовательские интерфейсы.

В рамках проекта Eclipse SmartHome инфраструктура openHAB может применяться для корпоративного использования. Eclipse SmartHome пытается привести в соответствие фрагментированную экосистему «умного» дома с общими интерфейсами API для создания пользовательских интерфейсов.

OpenIoT. Это облачная Java-ориентированная платформа, в рамках которой можно создавать приложения, собирающие и обрабатывающие данные с датчиков. В случае необходимости другие приложения, работающие в той же OpenIoT, на основе этих данных будут предпринимать какие-либо действия — например, уведомлять родителей о том, что их ребенок проснулся, или, скажем, при наблюдении за больным автоматически размещать заказ на лекарства, если их количество в холодильнике будет ниже допустимого.

Для этого датчики подключаются к узлам сбора информации, поступающие данные собираются, комбинируются, и им присваиваются метки.

OpenRemote. Продукт компании OpenRemote с одноименным названием позволяет создавать мобильные приложения для «умного» дома без программирования, при этом в одном приложении могут использоваться разные технологии: Z-Wave, KNX, X10, ZigBee, управление компьютером по ssh и др. OpenRemote — это сервер, выполняющий любые команды, и конструктор интерфейсов переключателей, надписей и др., этим элементам назначаются команды, например, http-запросы на выполнение JavaScript функций на сервере Z-Wave.

OpenThread. Приобретенная Google и входящая сейчас в группу Alphabet компания Nest Labs опубликовала в мае 2015 г. исходный код библиотеки OpenThread, реализующей протокол связи для устройств Интернета вещей под названием Thread. Протокол Thread используется в разработанной в Nest Labs системе Weave, предназначенной для связи между интеллектуальной домашней техникой.

Учитывая, что Thread основывается на 6LoWPAN, который, в свою очередь, использует IEEE 802.15.4, при наличии исходного кода добавление поддержки Thread требует минимальных усилий от разработчиков устройств. При использовании OpenThread обеспечивают доступ к облаку и шифрование по алгоритму AES. Проект OpenThread является очередной попыткой предложить сфере IoT универсальный язык общения. Thread уже активно используется множеством компаний, которые разрабатывают подключаемые продукты.

Eddystone/Physical Web. Проект Google Eddystone разрабатывает кросс-платформенный формат Bluetooth LE для Bluetooth-маяков. Маяками называют Bluetooth-передатчики, отправляющие какие-то данные информационного или рекламного характера, которые могут принимать смартфоны и планшеты в радиусе действия передатчика. Например, автобусная остановка может транслировать таким образом график маршрутов, магазин — рекламные акции и предложения, музей — режим работы выставок и т. д. Сообщение, приходящее в формате оповещения, может содержать ссылку, ведущую на веб-страницу.

Важным отличием Eddystone от аналогов является поддержка нескольких так называемых «типов фреймов» (блоков загруженных данных).

Предыдущие решения от Apple (iBeacon) и самой Google (The Physical Web) поддерживают только один тип, что ограничивает их функциональность. Поддержка Eddystone уже встроена в Nearby API на Android в составе Google Play Services. Формат также можно использовать в iOS в качестве библиотеки. Код Eddystone доступен на GitHub по лицензии Apache v2.0.

PlatformIO. Этот проект включает в себя утилиту командной строки, через которую можно запускать компиляцию и загрузку программ на несколько семейств микроконтроллеров (Atmel AVR, Atmel SAM, ST STM32, TI MSP430 и др.). При этом поддерживаются разные наборы библиотек: Arduino, Energia, mbed, а также нативный код для Atmel AVR, espressif, MSP430.

PlatformIO может быть востребован разработчиками, которые пишут ПО для одноплатных компьютеров на разных процессорах или архитектурах — их код будет компилироваться под разные платы. Проект имеет интегрированную среду разработки. Для работы с PlatformIO требуется скачать языковой пакет Python и текстовый редактор SublimeText. Поддерживается более 200 плат.

Thing System. Программная платформа автоматизации дома на базе Node.js. Особенность платформы в том, что она работает на базе ИИ-движка, может самообучаться и обрабатывать запросы уровня M2M не требуя вмешательства человека. Отсутствие подключения к облачным сервисам повышает безопасность платформы и конфиденциальность хранящихся в ней данных.

ThingSpeak. Данная платформа позволяет писать приложения для обработки данных, собранных с датчиков. К основным возможностям ThingSpeak можно отнести: сбор данных в реальном времени, анализ данных и их визуализация. ThingSpeak API позволяет не только отправлять, хранить и получать доступ к данным, но и предоставляет различные статистические методы их обработки.

Основу платформы составляют каналы, в которые и посылаются данные для хранения и визуализации. Каждый канал включает в себя восемь полей для любого типа данных, три поля для местоположения (широта, долгота, высота), и одно поле состояния.

После регистрации в ThingSpeak своего канала в него сразу можно отправлять данные, обрабатывать их и получать к ним доступ через корпоративные приложения. Каналы поддерживают форматы данных JSON, XML и CSV.

SiteWhere. Проект позволяет бизнесу создавать решения IoT без специфичного программирования. Оборудование может быть привязано к физическим или логическим ресурсам с помощью подключаемых готовых коммерческих модулей. SiteWhere поддерживает многочисленные протоколы, такие как MQTT, JSON, AMQP, XMPP, Stomp, JMS и WebSockets при помощи опубликованных API. Данные могут быть сохранены в облаке или на локальных серверах. SiteWhere предоставляет интерфейсы для настраиваемых по событиям триггеров, поиск и аналитику, а также техподдержку для быстрой интеграции индивидуальных данных устройства в другие платформы.

Zetta. В основе Zetta лежит программная платформа Node.js, которая помогает связывать устройства с облаком для создания георасределенной сети. Zetta сочетает интерфейсы REST, WebSockets и реактивное программирование, что подходит для сборки устройств в решение для обработки больших объёмов данных в реальном времени. Zetta служит основой таких коммерческих платформ, как Apigee.

PC Week №20 (919) 22 ноября 2016
Сергей Стельмах | itWeek

Учебно-практическое издание

«Визуальное программирование (FBD) для микропроцессорных систем и IoT»

Практикум

2021 г.

GLÜCKSRITTER
MUNISTER

§ INSCRIPTUM

Устойчивое желание обрести как можно больше профессиональных навыков и повысить общий уровень компетенций по тому или иному вопросу требуют более высокого уровня функциональности и комплексности в методологии преподаваемых дисциплин и междисциплинарных модулей.

Лабораторный практикум - существенный элемент учебного процесса в профессиональном учебном заведении, в ходе которого обучающиеся фактически впервые сталкиваются с самостоятельной практической деятельностью в конкретной области. Лабораторные занятия, как и другие виды практических занятий, являются средним звеном между углубленной теоретической работой обучающихся на лекциях, семинарах и применением знаний на практике. Эти занятия удачно сочетают элементы теоретического исследования и практической работы. Выполняя лабораторные работы, студенты лучше усваивают программный материал, так как многие определения и формулы, казавшиеся отвлеченными, становятся вполне конкретными, происходит соприкосновение теории с практикой, что в целом содействует уяснению сложных вопросов науки и становлению обучающихся как будущих специалистов.

Само значение слов «лаборатория», «лабораторный» (от латинского labor труд, работа, а laboro - трудиться, стараться, хлопотать, преодолевать затруднения) указывает на сложившиеся понятия, связанные с применением умственных и физических усилий к изысканию ранее неизвестных путей и средств для разрешения научных и жизненных задач.

Неслучайно слово «практикум», применяемое для обозначения определенной системы практических (преимущественно лабораторных) учебных работ, и выражает ту же основную мысль (греческое - praktikos), означает «деятельный», это значит, что предполагаются такие виды учебных занятий, которые требуют от обучающихся усиленной деятельности.

В целях создания интегрированного курса, связывающего между собой основы алгоритмизации и программирования (а также реверс-инжиниринга), электронику и электротехнику, был создан данный лабораторный практикум.

Полагаю, что разработанный мной практикум, базирующийся на концепции обучения посредством визуального программирования и встроенным программам интродукции в архитектуру аппаратных средств, широко используемых во встроенных компьютерных системах, помогут вам в профессиональной деятельности.

Мунистер В.Д.

§ СОДЕРЖАНИЕ

INSCRIPTUM 273
СОДЕРЖАНИЕ КУРСА 275
ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВ ДИАГРАММ (FBD). 277
ОБЗОР ТИПОВ СРЕД ПРОГРАММИРОВАНИЯ СИСТЕМ 280
СРЕДА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ (FLProg) 287
ОБЗОР АППАРАТНО-ПРОГРАММНЫХ СРЕДСТВ ARDUINO 288
ПРАКТИКУМ: ПЕРВЫЙ ПРОЕКТ В FLPROG и ARDUINO IDE 297
ПРАКТИКУМ: СОЗДАНИЕ СИСТЕМЫ ОГРАНИЧЕНИЯ ДОСТУПА С ПРИМЕНЕНИЕМ РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ 313
ПРАКТИКУМ: СОЗДАНИЕ ВСТРАИВАЕМОЙ СИСТЕМЫ ПО ТЕХНИЧЕСКОМУ ЗАДАНИЮ 320
RemoteXY и Bluetooth ВЗАИМОДЕЙСТВИЕ С ПРОГРАММОЙ FLPROG: 321
ПРАКТИКУМ: БЕСПРОВОДНАЯ КЛАВИАТУРА ДЛЯ КОМПЬЮТЕРА НА СМАРТФОНЕ. 328
ОБЗОР SoC ESP8266/ESP32 как аппаратной основы IoT 330
ПРАКТИКУМ: СОЗДАНИЕ WEB-ИНТЕРФЕЙСА НАСТРОЙКИ ACCESS POINT ESP8266 В FLProg. 343
HMI (Human-Machine-Interface) 371
FLProg + Nextion HMI. …. 372
ОБЗОР ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ 374

§ СОДЕРЖАНИЕ КУРСА

Как говорил еще Конфуций: задача учителя — открывать новую перспективу размышлениям ученику.

Раскрытие перспективы современных, многофункциональных и доступных к применению в практической деятельности микропроцессорных систем может достигаться за счет вовлечения в результат удивительного сплава мысли в области программной инженерии – визуального программирования.

Визуальное (графическое) программирование — способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста. Его часто представляют, как следующий этап развития текстовых языков программирования.

В последнее время визуальному программированию стали уделять больше внимания — в связи с развитием мобильных сенсорных устройств и средств, обеспечивающих Human-machine interface (человеко-машинный интерфейс) на уровне взаимодействия оператора с органами управления системы. Визуальное программирование в основном используется для создания программ с графическим интерфейсом для операционных систем с графическим интерфейсом пользователя.

Среда визуального программирования позволяет написать Web-приложение для браузера; создать консольное приложение для программирования микроконтроллеров, программируемых микросхем.

В учебно-практическом пособии «Визуальное программирование (FBD) для микропроцессорных систем и IoT» уделено большое внимание изучению и применению в прикладных задачах распространенных микропроцессорных систем (MCU/SoC) посредством их конфигурирования (данных управляющих устройств) на графическом языке программирования FBD (Function Block Diagram) под конкретные целевые задачи.

Издание предназначено для студентов, изучающих дисциплины «Алгоритмизация и программирование», «Микропроцессорные системы», «Проектирование микропроцессорных систем», «Аппаратно-программные комплексы», «Вычислительные машины системы и сети», «Вычислительная техника и сети в отрасли», «Основы визуального программирования».

Практикум является элементом интегрированного курса «Интернет вещей. Межмашинное взаимодействие. Программирование в компьютерных системах и сетях» в который входит учебно-теоретическое издание (хрестоматия): «Компьютерные сети. IoT и межмашинное взаимодействие», и учебно-практическое издание «Дом, который построил сам себя. Сетевой практикум. IoT»

Данное учебно-практическое издание должно выработать у студента навыки осознанного применения теоретических знаний о беспроводных технологиях взаимодействия и идентификации, изученных в ранее указанных изданиях рассматриваемого интегрированного курса по сетевому взаимодействию вместе с рассматриваемыми технологиями и средствами со смежных учебных курсов.

Потенциал рассматриваемой здесь предметной области (в сфере программирования) наиболее сильно раскрывается в прикладной деятельности и может рекомендоваться в качестве дополнения к дисциплинам:

«Эксплуатация информационных систем на транспорте/производстве (по отраслям)», «Инструментальные средства информационных систем (по отраслям)» по направлениям подготовки ВПО ОКУ «Бакалавр» 09.03.01 – Информатика и вычислительная техника, 09.03.02 – «Информационные системы и технологии», 15.03.04 – «Автоматизация технологических процессов и производств», 19.06.02 – «Эксплуатация транспортно-технологических машин и комплексов»,

Материалы издания безо всех ограничений могут быть использованы, дополнены инженерами и аспирантами, занимающимися задачами автоматизации технологических процессов, автоматизированного управления и мониторинга, синтеза дискретных систем, так как являются адаптированными версиями материалов (стандартов и иной документации), изложенных на открытых источниках. Автор данного учебного издания несет ответственность за корректность перевода и адаптации стандартов IEEE/IETF, которые на момент публикации не были русифицированы.

Содержание данного учебного курса было апробировано на рассматриваемых микропроцессорных устройствах в условиях эксперимента и тестирования.

§ ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВ ДИАГРАММ (FBD)

FBD (англ. Function Block Diagram) — графический язык программирования стандарта **МЭК 61131-3**.

ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р МЭК 61131-3— 2016

КОНТРОЛЛЕРЫ ПРОГРАММИРУЕМЫЕ

Часть 3

Языки программирования

(IEC 61131-3:2013, IDT)

Издание официальное

Стандарт **МЭК 61131-3** описывает 5 языков программирования, явившихся результатом изучения наиболее удачных фирменных разработок мировых лидеров рынка ПЛК. Языки ПЛК весьма оригинальны и существенно отличаются от известных языков программирования для компьютеров. Рассмотрим один из языков, получивший признание инженеров в области автоматизации технологических процессов: FBD.

Развитие идеи программной реализации электрических схем привело к появлению языка функциональных блоков и диаграмм (ФБД), FBD (Function Block Diagram).

Диаграмма FBD очень **напоминает принципиальную схему электронного устройства на микросхемах.** Выходы экземпляров функциональных блоков могут быть поданы на входы других блоков либо непосредственно на выходы ПЛК. Сами блоки, представленные на схеме как функциональные модули, могут выполнять стандартные и специальные функции.

FBD схемы ясно отражают взаимосвязь входов и выходов диаграммы. **Если алгоритм хорошо описывается с позиции сигналов, то его FBD-представление всегда получается значительно нагляднее, чем в текстовых языках.**

FBD предназначен для программирования программируемых логических контроллеров (ПЛК). Программа образуется из списка цепей, выполняемых последовательно сверху вниз. Цепи могут иметь метки.

Инструкция перехода на метку позволяет изменять последовательность выполнения цепей для программирования условий и циклов.

При программировании используются наборы библиотечных блоков и собственные блоки, также написанные на FBD или других языках **МЭК 61131-3.**

Блок (элемент) — это подпрограмма, функция или функциональный блок (И, ИЛИ, НЕ, триггеры, таймеры, счётчики, блоки обработки аналогового сигнала, математические операции и др.).

Каждая отдельная цепь представляет собой выражение, составленное графически из отдельных элементов. К выходу блока подключается следующий блок, образуя цепь.

Внутри цепи блоки выполняются строго в порядке их соединения. Результат вычисления цепи записывается во внутреннюю переменную либо подается на выход ПЛК (рис.1):

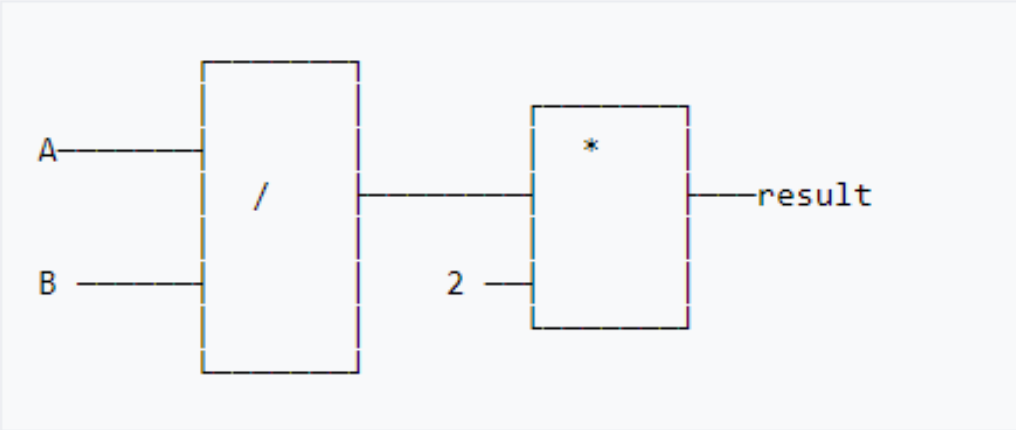


Рис. 1 — Пример фрагмента программы на FBD

Пример формализации исходного предиката: «A поделить на B, умножить на 2 и записать в переменную result» предоставлен выше. Та же самая функция на псевдокоде: result:= 2*A/B.

При необходимости управления вызовом блоков в них добавляются специальные входы EN (enable) и выходы ENO. Логический ноль на входе EN запрещает вызов блока. Выход ENO используется для индикации ошибки в блоке и позволяет прекратить вычисление остатка цепи.

Язык FBD прост в изучении, нагляден и удобен для прикладных специалистов, не имеющих специальной подготовки в области информатики. Жесткая последовательность выполнения приводит к простой внутренней структуре команд, которая транслируется в быстрый и надежный код. Существует много практических реализаций языка FBD с определенными расширениями или ограничениями.

Одним из вариантов FBD является язык программирования CFC (Continuous Function Chart). Он позволяет произвольно задавать порядок выполнения блоков. Диаграммы CFC дают программисту большую свободу действий, но платой за это является несколько большая вероятность допустить ошибку и более объемный код.

Язык функциональных блоковых диаграмм (FBD)
и его применение
Онлайн-журнал: "Электрик Инфо".

§ ОБЗОР ТИПОВ СРЕД ПРОГРАММИРОВАНИЯ СИСТЕМ

Среды программирования (среды разработки) — это программы, в которых программисты пишут свои программы.

Иными словами, среда программирования служит для разработки (написания) программ и обычно ориентируется на конкретный язык или несколько языков программирования (в этом случае языки, обычно, принадлежат одной языковой группе, например, Си-подобные).

Интегрированная среда программирования (IDE) содержит в себе все необходимое для разработки программ:

Редактор — в нем программист пишет текст программы, так называемый программный код;

Компилятор — он, как мы уже с вами знаем, транслирует программу, написанную на высокоуровневом языке программирования в машинный язык (машинный код), непосредственно понятный компьютеру. Язык С++ относится к компилируемым языкам, поэтому для обработки текстов его программ служит компилятор, иногда вместо компилятора (либо вместе с ним) используется интерпретатор, для программ, написанных на интерпретируемых языках программирования;

Отладчик — служит для отладки программ. Как мы все знаем, ошибки в программах допускают абсолютно все: и новички, и профессионалы - они могут быть синтаксическими (обычно они выявляются еще на стадии компиляции) и логическими. Для тестирования программы и выявления в ней логических ошибок служит отладчик.

Мы рассмотрели базовую комплектацию среды программирования, но иногда в них присутствуют еще и такие компоненты, как система управления версиями, различные инструменты для конструирования графического интерфейса программы, браузер классов, инспектор объектов и другие.

Рис. 2 — Состав системы программирования

Таким образом, инструментальная среда (система) программирования включает, прежде всего, текстовый редактор, позволяющий конструировать программы на заданном языке программирования, а также инструменты, позволяющие компилировать или интерпретировать программы на этом языке, тестировать и отлаживать полученные программы.

Кроме того, могут быть и другие инструменты, например, для статического или динамического анализа программ. Взаимодействуют эти инструменты между собой через обычные файлы с помощью стандартных возможностей файловой системы.

Различают следующие классы инструментальных сред программирования (см. рис.2):

· среды общего назначения,

· языково-ориентированные среды.

Инструментальные среды программирования общего назначения содержат набор программных инструментов, поддерживающих разработку программ на разных языках программирования (например, текстовый редактор, редактор связей или интерпретатор языка целевого компьютера) и обычно представляют собой некоторое расширение возможностей используемой операционной системы. Для программирования в такой среде на каком-либо языке программирования потребуются дополнительные инструменты, ориентированные на этот язык (например, компилятор).

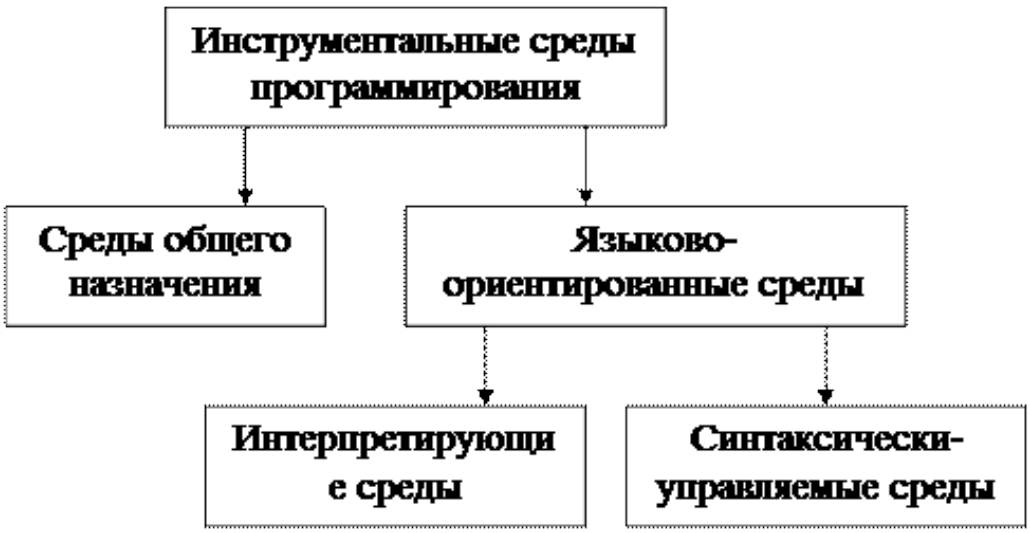


Рис. 3 — Классификация инструментальных сред программирования.

Языково-ориентированная инструментальная среда программирования предназначена для поддержки разработки ПС на каком-либо одном языке программирования и знания об этом языке существенно использовались при построении такой среды.

Вследствие этого в такой среде могут быть доступны достаточно мощные возможности, учитывающие специфику данного языка. Такие среды разделяются на два подкласса:

· интерпретирующие среды,
· синтаксически-управляемые среды.

Интерпретирующая инструментальная среда программирования обеспечивает интерпретацию программ на данном языке программирования, т.е. содержит, прежде всего, интерпретатор языка программирования, на который эта среда ориентирована. Такая среда необходима для языков программирования интерпретирующего типа (таких, как Лисп), но может использоваться и для других языков.

Синтаксически-управляемая инструментальная среда программирования базируется на знании синтаксиса языка программирования, на который она ориентирована. В такой среде вместо текстового используется синтаксически-управляемый редактор, позволяющий пользователю использовать различные шаблоны синтаксических конструкций (в результате этого разрабатываемая программа всегда будет синтаксически правильной). Одновременно с программой такой редактор формирует (в памяти компьютера) ее синтаксическое дерево, которое может использоваться другими инструментами.

В данном практикуме в основном мы будем использовать две среды общего назначения, **связь которых между собой** несет в себе характерные свойства интерпретирующих инструментальных сред, так и синтаксически-управляемых инструментальных сред (связь станет очевидной при комплексном анализе).

Основной средой разработки в практикуме станет IDE **FLProg**, результат работы в которой будет интерпретировано и транслировано в формальную форму, применяемую в другой среде разработки – **Arduino IDE**.

Arduino — это прежде всего торговая марка аппаратно-программных средств для построения простых систем автоматики и робототехники, ориентированная на начинающих пользователей.

Программная часть состоит из бесплатной программной оболочки (IDE) (об этом пойдет чуть позже) для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть представляет собой набор смонтированных печатных плат, продающихся как официальным производителем, так и сторонними производителями. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino.

Arduino используется как для создания автономных объектов, так и подключения к программному обеспечению через проводные и беспроводные интерфейсы.

Среды программирования Arduino-совместимых плат можно разделить на следующие типы:

- Интегрированные среды разработки
- Графические среды, визуализирующие структуру кода.
- Графические среды, отображающие код в виде графики.
- Визуальные среды программирования, не использующие кода.

Рассмотрим каждый тип – начав по порядку:

Arduino IDE — интегрированная среда разработки для Windows, MacOS и Linux, разработанная на C и C ++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

Исходный код для среды выпущен под общедоступной лицензией GNU версии 2. Поддерживает языки Си и C ++ с использованием специальных правил структурирования кода.

Arduino IDE предоставляет библиотеку программного обеспечения из проекта Wiring, которая предоставляет множество общих процедур ввода и вывода. Для написанного пользователем кода требуются только две базовые функции для запуска эскиза и основного цикла программы, которые скомпилированы и связаны с заглушкой программы main () в исполняемую циклическую программу с цепочкой инструментов GNU, также включённой в дистрибутив IDE. Использует программу avrdude для преобразования исполняемого кода в текстовый файл в шестнадцатеричной кодировке, который загружается в плату Arduino программой-загрузчиком во встроенном программном обеспечении платы.

Проектирование программы для контроллера в ней происходит на языке Processing/Wiring, который является диалектом языка Си (скорее Си++). Эта среда представляет собой, по сути, обычный текстовый редактор с возможностью загрузки написанного кода в контроллер

Альтернативой Arduino IDE является среда разработки от производителя микроконтроллеров Atmel — AVRStudio.

Программирование в ней ведётся на чистом C, и она уже имеет намного больше возможностей и более похожа на серьёзные IDE (Visual Studio).

Эти два типа рассчитаны на опытных программистов, которые хорошо знают язык и могут с помощью них создавать серьёзные проекты.

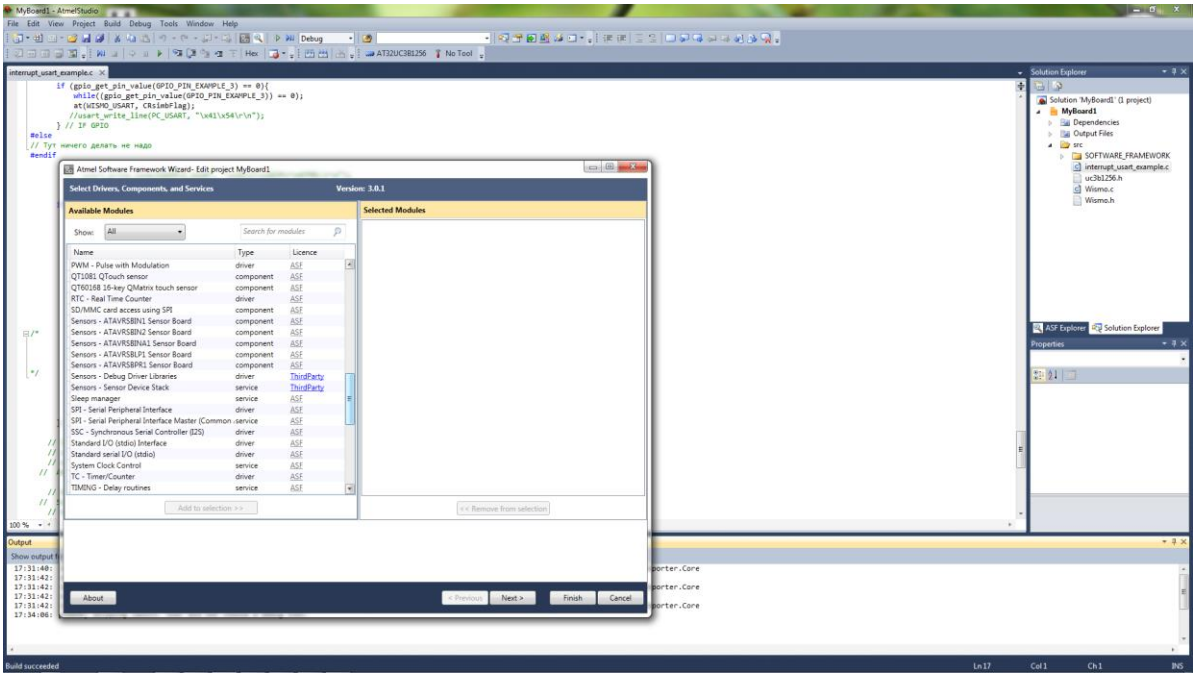


Рис. 4 — UI IDE AVRstudio

Графические среды, визуализирующие структуру кода.

Это программы, которые, по сути, являются расширением форматирования для обычного текстового редактора кода. В нем программа так же пишется на языке C, но в более удобном варианте. Сейчас таких сред очень много, самые яркие примеры: Scratch, S4A, Ardublock (рис.5).

Они очень хорошо подходят для начального обучения программированию на языке C, поскольку отлично показывают структуру и синтаксис языка.

Но для больших серьёзных проектов программа получается громоздкой.

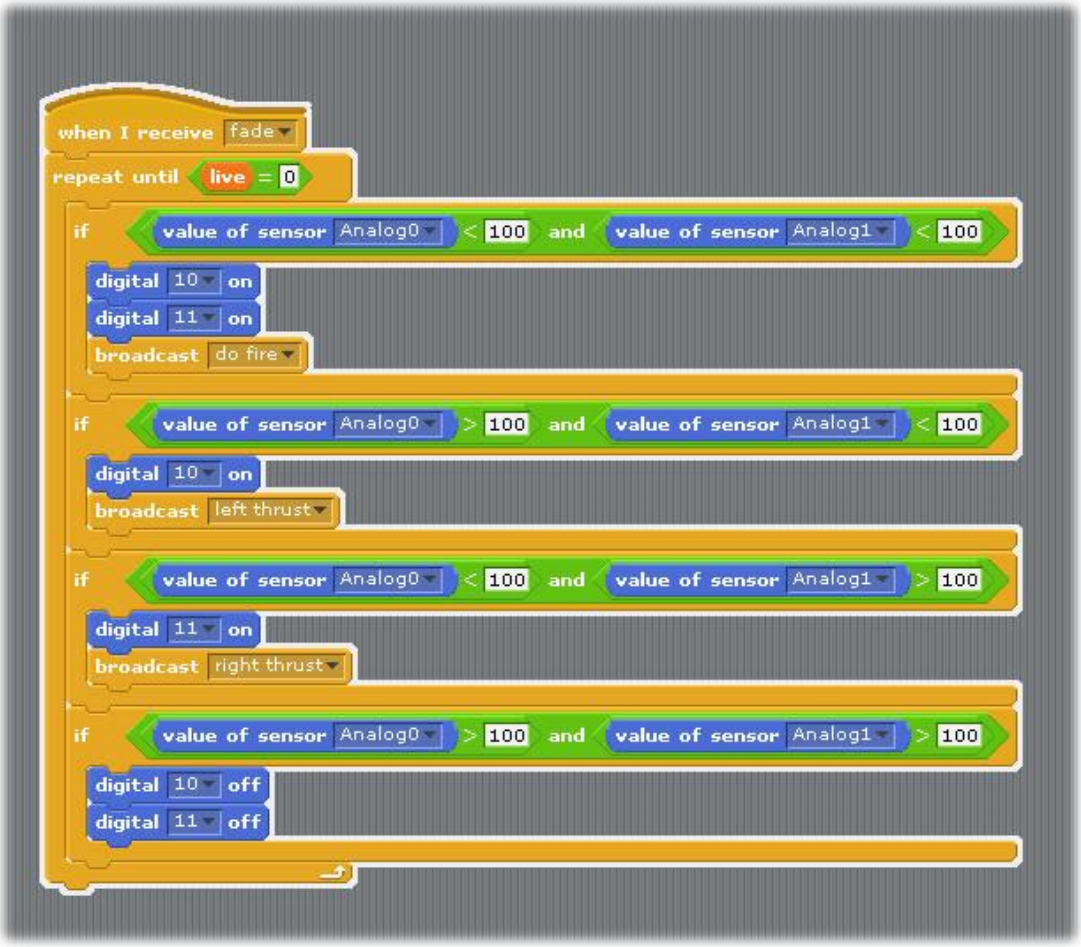


Рис. 5 — Графическая среда, визуализирующая структуру кода.

Графические среды, отображающие код в виде графики.

Это программы, скрывающие код и заменяющие его графическими аналогами. В них так же повторяется структура языка, формируются циклы, переходы, условия. Так же очень хорошо подходят для обучения построению алгоритмов, с последующим переходом на программирование на классических языках. И так же не подходят для построения больших проектов ввиду громоздкости получаемого отображения. Пример таких программ: **MiniBlog, Algorithm Builder, Flowcode.**

Описанные выше типы программ рассчитаны на full-stack программистов или на тех, кто решил изучать классическое программирование. Но для изготовления конечного устройства кроме непосредственно программирования контроллера обычно требуется разработка внешней обвязки платы, разработка и расчет силовой части, входных развязок и многого другого. С этим у программистов часто возникают проблемы. Зато с этим прекрасно справляются электрики и электронщики.

Но среди них мало программистов, которые смогли бы составить программу для контроллера. Сочетание программиста и электронщика – достаточно редкий случай. В результате такой ситуации реальных, законченных проектов на основе плат Arduino (да и других контроллеров) единицы. Для решения этой проблемы и служат программы последнего типа:

Визуальные среды программирования, не использующие кода.

Данные программы реализуют принцип, который уже много лет применяется практически всеми производителями контроллеров промышленного применения. Он заключается в создании программ для контроллера на языках FBD или LAD. Собственно говоря, как таковыми языками они не являются. Это, скорее, графические среды для рисования принципиальных или логических схем. Вспомним, что процессоры далеко не всегда были микропроцессорами, а создавались на базе цифровых микросхем. Поэтому тем, кто привык работать с цифровой техникой, больше понравится работа на них, чем написание кода на классических языках программирования. Примером таких программ являются проекты Horizont и **FLProg**. Программы этого типа хорошо подходят как для изучения построения импульсной и релейной техники, так и для создания серьезных проектов.

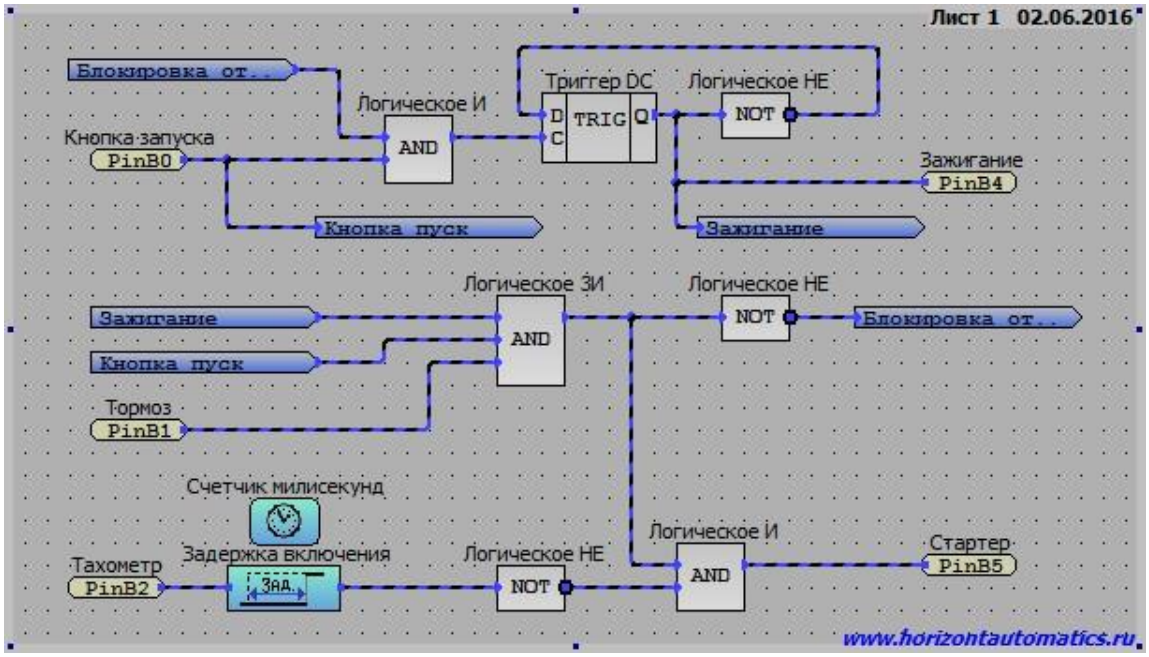


Рис. 6 — Horizont Configurator, визуальная среда построения алгоритмов работы устройств, в том числе и микроконтроллеров. объединяются друг с другом непосредственно линиями связи – графическими связями.

§ СРЕДА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ FLProg

FLProg IDE — среда разработки, предназначенная для создания транслируемого в C/C++ и компилируемоего кода посредством методологии графического (визуального) программирования.

С помощью этой программы можно запрограммировать контроллер не зная текстовых языков программирования, а выглядит это как рисование электронной или электрической схемы.

Визуальные языки программирования FBD и Ladder, с помощью которых пишется программа, используются для программирования практически всех логических реле, и части промышленных контроллеров во всем мире.

Для создания FLProg был использован опыт программистов фирм Siemens, ABB, Schneider Electric и наработки в их средах программирования. При этом был несколько расширен классический функционал языков для работы с промышленными контроллерами путём добавления функциональных блоков, отвечающих за работу с внешними устройствами. Программа работает на компьютерах под управлением ОС Windows и Linux.

Пользовательский интерфейс FLProg устроен так, что проект представляет собой набор виртуальных плат, на каждой из которых собран законченный модуль разрабатываемой системы.

Каждая плата имеет наименование и снабжена комментариями. Для экономии места в рабочей зоне её можно свернуть, если работа над ней закончена, а при необходимости вновь развернуть и внести коррективы.

Разработанную «принципиальную схему» FLProg переводит на язык Processing/Wiring. По завершении компиляции автоматически открывается программа Arduino IDE с загруженным скетчем проекта. В Arduino IDE необходимо указать COM-порт компьютера, к которому подключён микроконтроллерный модуль, выбрать тип модуля и загрузить программу в его микроконтроллер.

Первый урок по работе с программой FLProg (обзор)
https://flprog.ru/uchebnyj-centr/articles/znakomstvo-s-flprog/pervyj-urok-po-rabote-s-programmoj-flprog/

§ ОБЗОР АППАРАТНО-ПРОГРАММНЫХ СРЕДСТВ ARDUINO

Петин В. А. — Проекты с использованием контроллера Arduino.

Появление первых микроконтроллеров ознаменовало начало новой эры в развитии микропроцессорной техники. Наличие в одном корпусе большинства системных устройств сделало микроконтроллер подобным обычному компьютеру. В отечественной литературе они даже назывались однокристальными микроЭВМ. Соответственно и желание использовать микроконтроллеры как обычные компьютеры появилось практически с их появлением.

Но желание это сдерживалось многими факторами. Например, чтобы собрать устройство на микроконтроллере, необходимо знать основы схемотехники, устройство и работу конкретного процессора, уметь программировать на ассемблере и изготавливать электронную технику. Потребуются также программаторы, отладчики и другие вспомогательные устройства.

В итоге без огромного объема знаний и дорогостоящего оборудования не обойтись. Такая ситуация долго не позволяла многим использовать микроконтроллеры в своих проектах. Сейчас, с появлением устройств, дающих возможность работать с микроконтроллерами без наличия серьезной материальной базы и знания многих предметов, все изменилось. Примером такого устройства может служить проект Arduino итальянских разработчиков.

Arduino и его клоны представляют собой наборы, состоящие из готового электронного блока и программного обеспечения. Электронный блок здесь — это печатная плата с установленным микроконтроллером и минимумом элементов, необходимых для его работы.

Фактически электронный блок является аналогом материнской платы современного компьютера. На нем имеются разъемы для подключения внешних устройств, а также разъем для связи с компьютером, по которому и осуществляется программирование микроконтроллера. Особенности используемых микроконтроллеров ATmega фирмы Atmel позволяют производить программирование без применения специальных программаторов. Все, что нужно для создания нового электронного устройства, — это плата Arduino, кабель связи с компьютером.

Второй частью проекта Arduino является программное обеспечение для создания управляющих программ.

Оно объединило в себе простейшую среду разработки и язык программирования, представляющий собой вариант языка C/C++ для микроконтроллеров. В него добавлены элементы, позволяющие создавать программы без изучения аппаратной части. Так что для работы с Arduino практически достаточно знания только основ программирования на C/C++. Создано для Arduino и множество библиотек, содержащих код, работающий с различными устройствами.

В чем преимущество Arduino?

Пользователь современного компьютера не задумывается о функционировании отдельных частей ПК. Он просто запускает нужные программы и работает с ними.

Точно так же и Arduino позволяет пользователю сосредоточиться на разработке проектов, а не на изучении устройства и принципов функционирования отдельных элементов. Нет надобности и в создании законченных плат и модулей. Разработчик может использовать готовые платы расширения или просто напрямую подключить к Arduino необходимые элементы. Все остальные усилия будут направлены на раз работку и отладку управляющей программы на языке высокого уровня.

В итоге доступ к разработке микропроцессорных устройств получили не только профессионалы, но и просто любители что-то сделать своими руками. Наличие готовых модулей и библиотек программ позволяет непрофессионалам в электронике создавать готовые работающие устройства для решения своих задач. А варианты использования Arduino ограничены только возможностями микроконтроллера и имеющегося варианта платы, ну и, конечно, фантазией разработчика.

История создания Arduino

В 2002 году программист Массимо Банци (Massimo Banzi) был принят на работу в должности доцента в Институт проектирования взаимодействий города Иврея (Interaction Design Institute Ivrea, IDII) для продвижения новых способов разработки интерактивных проектов. Однако крошечный бюджет и ограниченное время доступа к лабораторной базе сводили его усилия практически на нет.

В проектах Банци использовал устройство BASIC Stamp, разработанное калифорнийской компанией Parallax. Stamp представлял собой небольшую печатную плату с размещенными на ней источником питания, микроконтроллером, памятью и портами ввода/вывода для соединения с различной аппаратурой. Программирование микроконтроллера осуществлялось на языке BASIC.

BASIC Stamp имел две проблемы: недостаток вычислительной мощности и достаточно высокую цену — плата с основными компонентами стоила около 100 долларов. И команда Банци решила самостоятельно создать плату, которая удовлетворяла бы всем их потребностям.

Банци и его сотрудники поставили себе целью создать устройство, представляющее собой простую, открытую и легкодоступную платформу для разработки, с ценой — не более 30 долларов — приемлемой для студенческого кармана. Хотели они и выделить чем-то свое устройство на фоне прочих.

Поэтому в противовес другим производителям, экономящим на количестве выводов печатной платы, они решили добавить их как можно больше, а также сделали свою плату синей, в отличие от обычных зеленых плат. Продукт, который создала команда, состоял из дешевых и доступных компонентов — например, базировался он на микроконтроллере ATmega328. Но главная задача состояла в том, чтобы гарантировать работу устройства по принципу plug-and play, чтобы пользователь, достав плату из коробки и подключив к компьютеру, мог немедленно приступить к работе.

Первый прототип платы был сделан в 2005 году, она имела простейший дизайн и еще не называлась Arduino. Чуть позже Массимо Банци придумал назвать ее так — по имени принадлежащего ему бара, расположенного в городе Ивреа. Бренд "Arduino" без какой-либо рекламы и привлечения средств маркетинга быстро приобрел высокую популярность в Интернете.

С начала распространения продано более 250 тыс. комплектов Arduino, и это, не учитывая множества клонов. В мире насчитывается более двухсот дистрибьюторов продукции Arduino — от крупных фирм, таких как SparkFun Electronics, до мелких компаний, работающих на местный рынок.

На сегодня платформа Arduino представлена не одной платой, а целым их семейством.

В дополнение к оригинальному проекту, называемому Arduino Uno, новые модели, имеющие на плате более мощные средства, носят название Arduino Mega, компактные модели — Arduino Nano, платы в водонепроницаемом корпусе — LilyPad Arduino, а новая плата с 32-разрядным процессором Cortex-M3 ARM — Arduino Due.

Своим успехом проект Arduino обязан существовавшим до него языку Processing и платформе Wiring. От этих проектов Arduino унаследовал одну сильную черту — удобную для пользователя среду разработки.

Due — плата на базе 32-битного ARM микропроцессора Cortex-M3 ARM SAM3U4E;

Leonardo — плата на микроконтроллере ATmega32U4;

Uno — самая популярная версия базовой платформы Arduino;

Duemilanove — плата на микроконтроллере ATmega168 или ATmega328;

Diecimila — версия базовой платформы Arduino USB;

Nano — компактная платформа, используемая как макет. Nano подключается к компьютеру при помощи кабеля USB Mini-B;

Mega ADK — версия платы Mega 2560 с поддержкой интерфейса USB-host для связи с телефонами на Android и другими устройствами с интерфейсом USB;

Mega2560 — плата на базе микроконтроллера ATmega2560 с использованием чипа ATMega8U2 для последовательного соединения по USB-порту;

Mega — версия серии Mega на базе микроконтроллера ATmega1280;

Arduino BT — платформа с модулем Bluetooth для беспроводной связи и программирования;

LilyPad — платформа, разработанная для переносимых изделий, может зашиваться в ткань;

Fio — платформа разработана для беспроводных применений. Fio содержит разъем для радио XBee, разъем для батареи LiPo и встроенную схему подзарядки;

Mini — самая маленькая платформа Arduino;

Pro — платформа, разработанная для опытных пользователей, может являться частью большего проекта;

Pro Mini — как и платформа Pro, разработана для опытных пользователей, которым требуется низкая цена, меньшие размеры и дополнительная функциональность.

Рассмотрим более подробно некоторые из этих плат.

Arduino Uno R3 — флагманская платформа для разработки на базе микроконтроллера ATmega328P.

На Arduino Uno предусмотрено всё необходимое для удобной работы с микроконтроллером: 14 цифровых входов/выходов (6 из них могут использоваться в качестве ШИМ-выходов), 6 аналоговых входов, кварцевый резонатор на 16 МГц, разъём USB, разъём питания, разъём для внутрисхемного программирования (ICSP) и кнопка сброса.

Микроконтроллер ATmega328P

Сердцем платформы Arduino Uno является 8-битный микроконтроллер семейства AVR — ATmega328P. А микроконтроллер ATmega16U2 обеспечивает связь микроконтроллера ATmega328P с USB-портом компьютера. При подключении к ПК Arduino Uno определяется как виртуальный COM-порт. Прошивка микросхемы 16U2 использует стандартные драйвера USB-COM, поэтому установка внешних драйверов не требуется.

Порты ввода/вывода

Цифровые входы/выходы: пины 0–13;

Логический уровень единицы — 5 В, нуля — 0 В.

Максимальный ток выхода — 40 мА. К контактам подключены подтягивающие резисторы, которые по умолчанию выключены, но могут быть включены программно.

ШИМ: пины 3,5,6,9,10 и 11

Позволяют выводить 8-битные аналоговые значения в виде ШИМ-сигнала.

АЦП: пины A0–A5

6 аналоговых входов, каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 значений). Разрядность АЦП — 10 бит.

TWI/I²C: пины SDA и SCL

Для общения с периферией по синхронному протоколу, через 2 провода. Для работы — используйте библиотеку Wire.

SPI: пины 10(SS), 11(MOSI), 12(MISO), 13(SCK).

Через эти пины осуществляется связь по интерфейсу SPI. Для работы — используйте библиотеку SPI.

UART: пины 0(RX) и 1(TX)

Эти выводы соединены с соответствующими выводами микроконтроллера ATmega16U2, выполняющей роль преобразователя USB-UART. Используется для коммуникации платы Arduino с компьютером или другими устройствами через класс Serial.

Имя светодиода	Назначение
RX и TX	Мигают при обмене данными между Arduino Uno и ПК.
L	Светодиод вывода `13`. При отправке значения HIGH светодиод включается, при отправке LOW – выключается.
ON	Индикатор питания на плате.

Рис. 7 — Светодиодная индикация на плате Arduino UNO R3

Характеристики Arduino UNO R3

Микроконтроллер: ATmega328
Тактовая частота: 16 МГц
Напряжение логических уровней: 5 В
Входное напряжение питания: 7–12 В
Портов ввода-вывода общего назначения: 20
Максимальный ток с пина ввода-вывода: 40 мА
Максимальный выходной ток пина 5V: 800 мА
Портов с поддержкой ШИМ: 6 Портов, подключённых к АЦП: 6 Разрядность АЦП: 10 бит Flash-память: 32 КБ EEPROM-память: 1 КБ Оперативная память: 2 КБ Габариты: 69×53 мм

Arduino Due — плата микроконтроллера на базе процессора Atmel SAM3X8E ARM Cortex-M3.

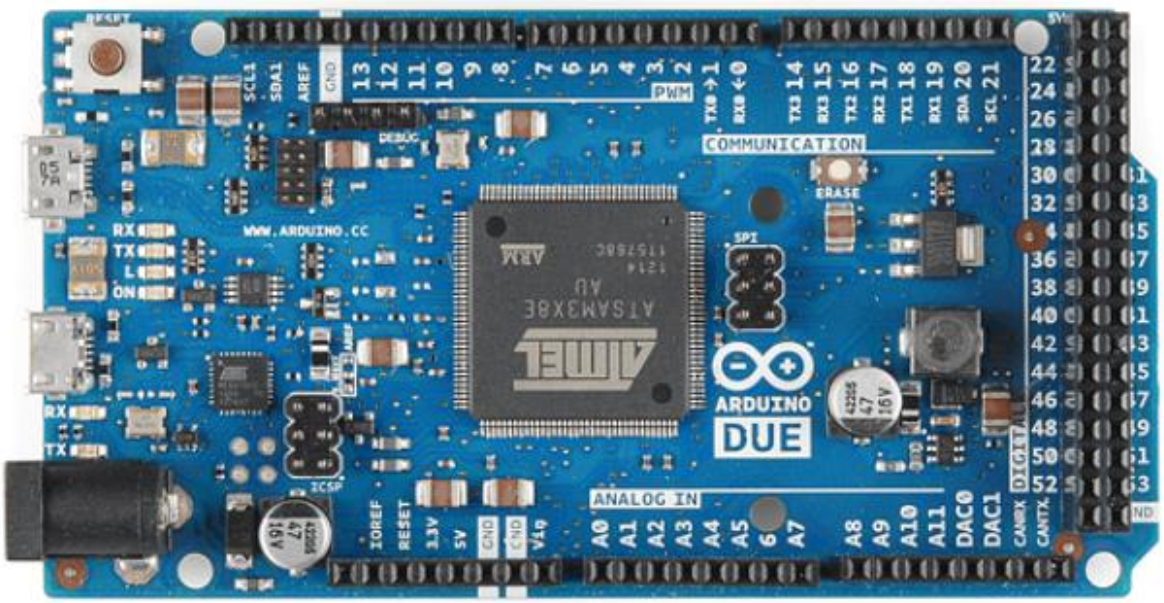


Рис. 8 — Arduino Due

Это первая плата Arduino на основе 32-битного микроконтроллера с ARM ядром. На ней имеется 54 цифровых вход/выхода (из них 12 можно задействовать под выходы ШИМ), 12 аналоговых входов, 4 UARTa (аппаратных последовательных порта), а генератор тактовой частоты 84 МГц, связь по USB с поддержкой OTG, 2 ЦАП (цифро-аналоговых преобразователя), 2 TWI, разъем питания, разъем SPI, разъем JTAG, кнопка сброса и кнопка стирания.

Arduino Due работает от 3,3 В. Максимальное напряжение, которое выдерживают вход/выходы составляет 3,3 В. Подав более высокое напряжение, например, 5 В, на выводы Arduino Due, можно повредить плату.

Плата содержит все, что необходимо для поддержки микроконтроллера. Чтобы начать работу с ней, достаточно просто подключить её к компьютеру кабелем микро-USB, либо подать питание с AC/DC преобразователя или батарейки. Due совместим со всеми платами расширения Arduino, работающими от 3,3 В, и с цоколевкой Arduino 1.0.

Порты ввода/вывода

В отличие от других плат Arduino, Arduino Due работает от 3,3 В. Максимальное напряжение, которое могут выдержать вход/выходы составляет 3,3 В. Подав напряжение, например: 5 В, на выводы Arduino Due, можно вывести плату из строя.

Цифровые входы/выходы: контакты 0–53. Работают на напряжении 3,3 В. В режиме выхода могут выдавать ток 3 или 15 мА (в зависимости от контакта); в режиме входа — принимать ток 6 или 9 мА (в зависимости от контакта). К контактам также подключены подтягивающие резисторы по 100 кОм, которые по умолчанию выключены, но могут быть включены программно.

Аппаратные последовательные порты (RX/TX): 0/1, 19/18, 17/16, 15/14. Передача данных осуществляется на уровне 3,3 В. Первая пара также соединена с чипом ATmega16U2, отвечающим за подключение через USB к компьютеру.

Широтно-имульсная модуляция (ШИМ/PWM): контакты 2–13. Дают возможность выдавать аппаратный шим с разрешением 8 бит (256 градаций).

SPI — отдельная группа контактов 2×3. На Arduino Due используется только для общения по SPI-интерфейсу с другими устройствами. Он не может быть использован для программирования контроллера, как на других Arduino. По расположению он в точности совпадает с расположением на Arduino Uno, Arduino Mega 2560, Arduino Leonardo, а следовательно даёт возможность работы с платами расширения его использующими, таких как Ethernet Shield.

CAN-шина: контакты CANRX и CANTX. Позволяют использовать Arduino Due в промышленных сетях. Поддержка с программной стороны пока не реализована производителем.

Встроенный светодиод: контакт 13 (L). Для простой индикации. В отличии от Arduino Uno и Mega, он поддерживает ШИМ.

Шины **TWI/I²C**: 20(SDA)/21(SCL), SDA1/SCL1. Для общения с периферией по синхронному протоколу, через 2 провода.

Аналоговые входы: контакты A0–A11. Принимают сигнал до 3,3 В. Большее напряжение может вывести процессор из строя. Аналоговые входы предоставляют разрешение до 12 бит (4096 градаций), хотя по умолчанию настроены на разрешение в 10 бит для совместимости со скетчами для других моделей Arduino.

Цифро-аналоговый преобразователь: контакты DAC1 и DAC2. Позволяют выдавать настоящий аналоговый сигнал с 12-битным разрешением (4096 градации), например, для устройств, связанных с обработкой звука.

Сброс процессора: RESET. Позволяет аппаратно перезагружать плату.

Входное напряжение: Vin. Выдаёт напряжение, поданное внешним источником, либо может являться входом для внешнего питания.

Стабилизированные 5 В: контакт 5V. Позволяет получать ровные 5 В и ток до 800 мА.

Стабилизированные 3,3 В: контакт 3.3V. Позволяет получать ровные 3,3 В и ток до 800 мА.

Общая земля: GND.

Опорное напряжение для плат расширения: IOREF. Платы расширения должны «советоваться» с этим контактом, чтобы правильно определять родное напряжение родительской платы. Arduino Due выдаёт на IOREF 3,3 В.

Характеристики Arduino Due

Микроконтроллер: AT91SAM3X8E
Тактовая частота: 84 МГц
Портов ввода-вывода общего назначения: 54
Максимальный ток с пина ввода-вывода: 3 или 15 мА (в зависимости от вывода)
Максимальный ток с пина ввода-вывода: 40 мА
Максимальный выходной ток пина 5V: 800 мА
Портов с поддержкой ШИМ: 12 Портов, подключённых к АЦП: 12 Разрядность АЦП: 12 бит Flash-память: 512 КБ Оперативная память: 96 КБ Габариты: 101×53 мм

http://arduino.ru/Hardware/ArduinoDue

§ ПРАКТИКУМ: ПЕРВЫЙ ПРОЕКТ В FLPROG и ARDUINO IDE

Оборудование: ПК, кабель AM/microBM 5p Cablexpert Pro (CCP-mUSB2-AMBM-1.0M или аналоги), Arduino UNO R3.

Программное обеспечение: IDE FLProg, IDE Arduino, интерактивный справочник (exeBook) FLProg_Старт.exe

Цель: ознакомиться с интерфейсом инструментальных средств, настроив их для работы соответствующим образом, получить навыки работы в интегрированных средах разработки, получить готовый к реализации FBD-скетч (прошивку), запрограммировать микропроцессорную систему.

Ход работы:

1. Открыть руководство FLProg_старт (рис.9) и программу FLProg. Изучить содержимое руководства (стр. 57 – 67).
2. Выполнить по инструкции содержание урока (1) (стр. 67 – 85). В качестве отчета предоставить видео-ролик продолжительностью до 25 секунд с демонстрацией работы устройства.

- Качество – не менее 720p.
- Платформа размещения – youtube.
- При размещении видео-ролика, демонстрирующего работу **светодиода определенного цвета** по заданному сценарию, не **соответствующему выбранному** студентом/преподавателем на начальном этапе работа не принимается.

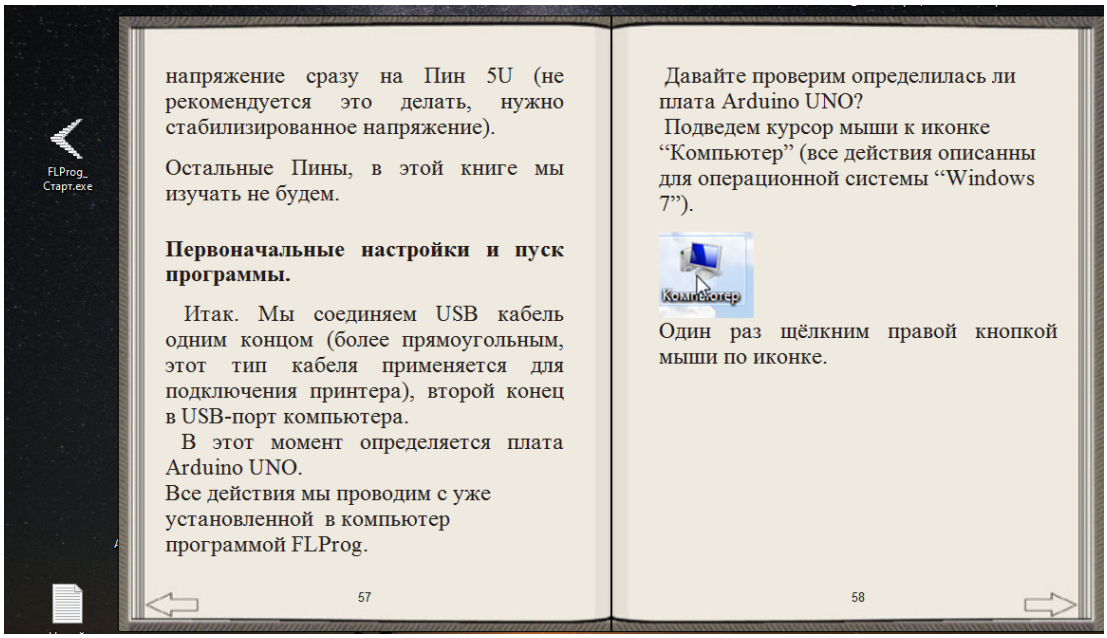


Рис. 9 — FLProg_старт.

3. У программистов в качестве первого урока принято использовать“Hello World”, у программистов микроконтроллеров помигать светодиодом (что мы уже сделали), а у электриков и электронщиков собрать схему управления контактором. Поскольку основными пользователями программы FLProg как раз они и являются, собирать на первом уроке будем как раз данную схему.

Контактор (лат. contāctor «соприкасатель») — двухпозиционный электромагнитный аппарат, предназначенный для частых дистанционных включений и выключений силовых электрических цепей в нормальном режиме работы. **Контактор** является разновидностью **электромагнитного** реле — важнейшего переключательного компонента во всей **электронике**, в том числе в **IoT**.

Принцип работы контактора любой модели заключается в том, что группа подвижных контактов постоянно сходится и расходится с неподвижными фиксированными контактами, пропуская и ограничивая течение электрического тока. Можно показаться, что это простой переключатель, но с рядом особенностей.

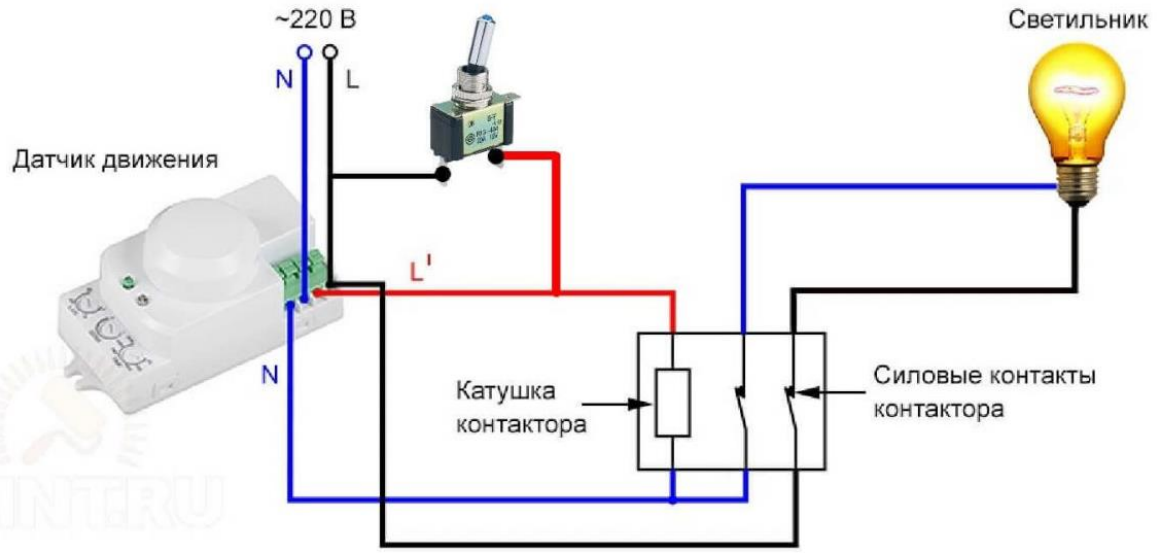


Рис. 11 — Схема управления освещением на контакторе

Первое и самое главное — для обеспечения безопасности нормальное положение контактных групп (режим покоя) разомкнутое. В таких приспособлениях нет никаких механических функций для воздействия на контакты, чтобы они всегда были во включенном состоянии. Они смыкаются лишь тогда, когда на них подают напряжение.

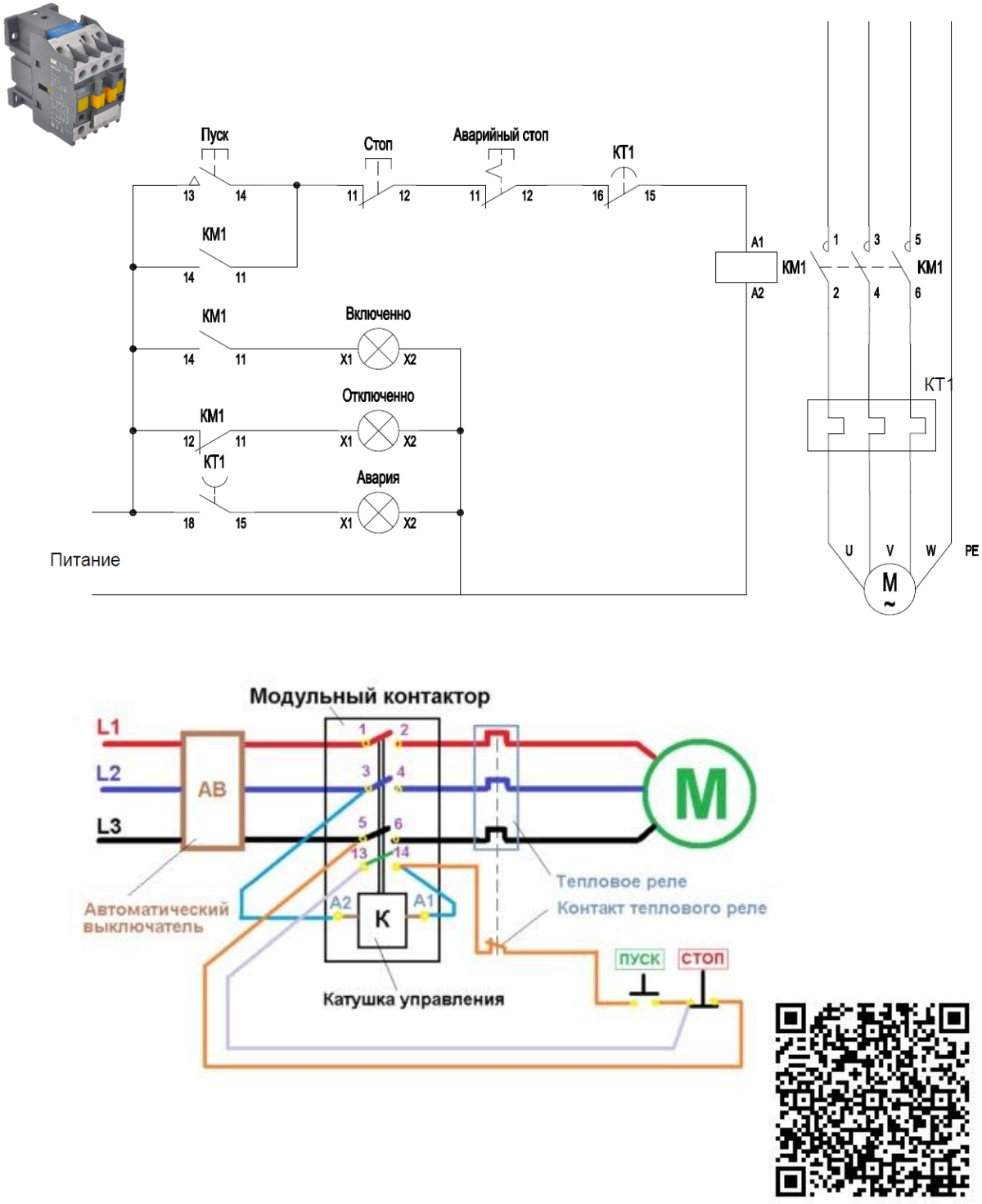


Рис. 11 — Стандартная схема управление контактором

Заменим эту схему контроллером Ардуино. Оставим в стороне вопросы помехозащищённости и экранирования устройства. Наша цель — создать в программе FLProg соответствующую дискретную логику. Поэтому оформим схему подключения.

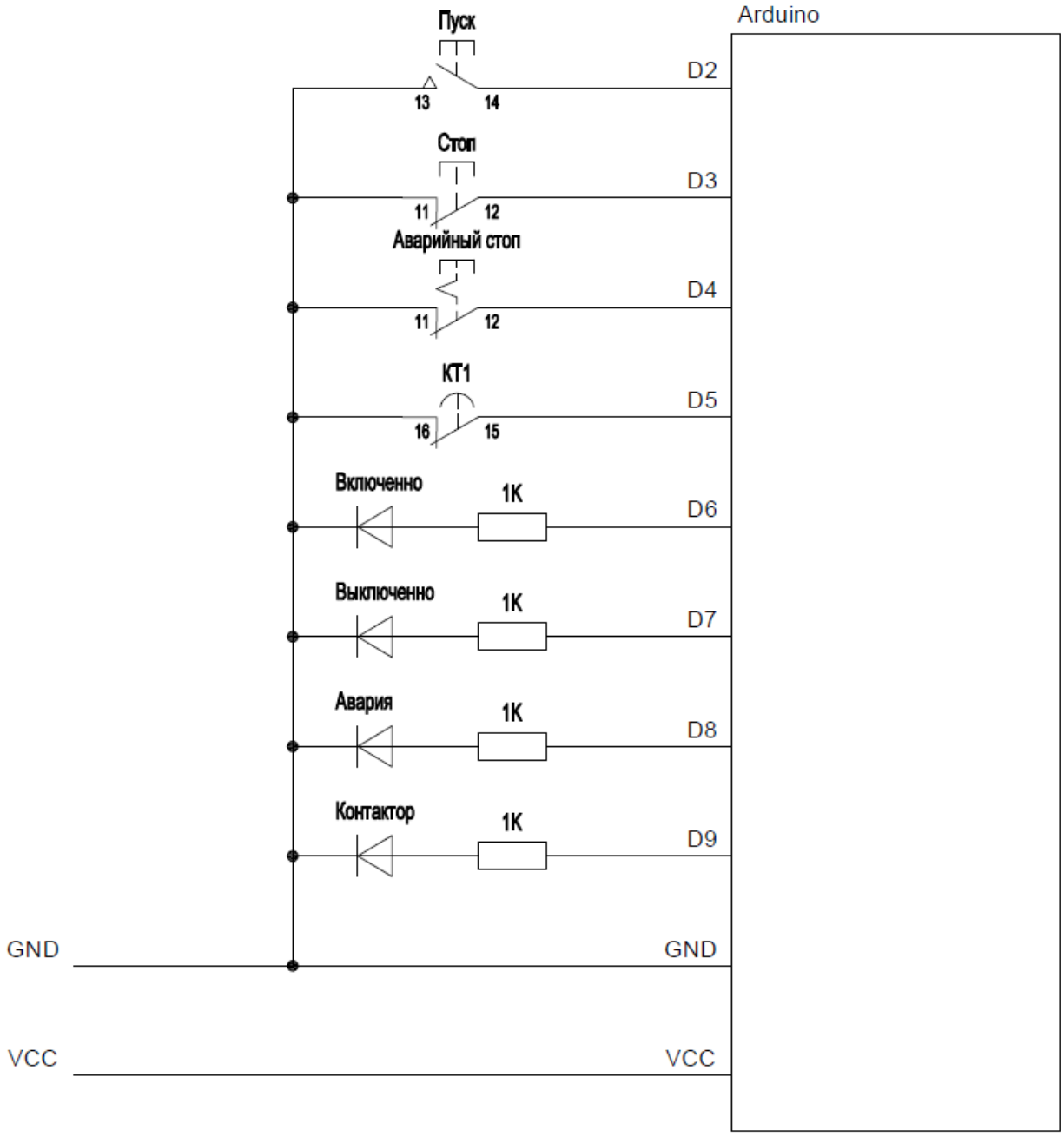


Рис. 11 — Тестовая схема управления под Arduino

Разберем схему чуть подробнее: D0…D9 – цифровые порты выбранной платы. Под обозначением 1К мы подразумеваем резисторы с таким номиналом. Индикация на портах вывода D6…D9 светодиодная, о чем свидетельствует надписи. GND – заземление. VCC – питание.

То есть, роль контактора в данной тестовой схеме выполняет светодиод «Контактор». Теперь попробуем запрограммировать контроллер.

Запускаем программу FLProg, нажимаем кнопку «Создать новый проект».

Откроется окно выбора контроллера и языка программирования проекта.

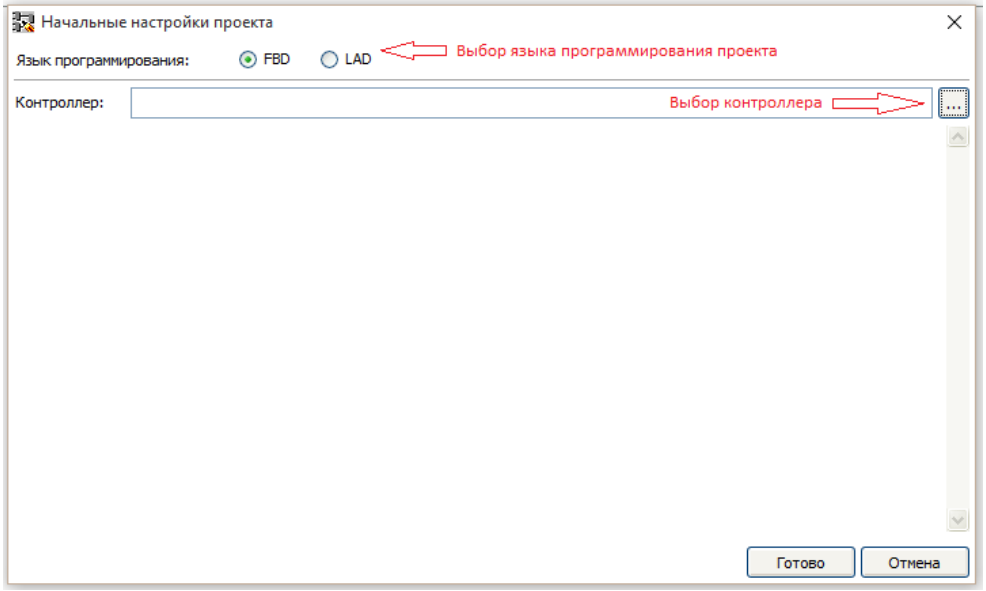


Рис. 12 — Начальные настройки проекта

Для создания проекта можно использовать любой из двух языков программирования (FBD и LAD) являющимися стандартами в области программирования промышленных контроллеров. **Выбираем язык FBD и контроллер Arduino UNO.**

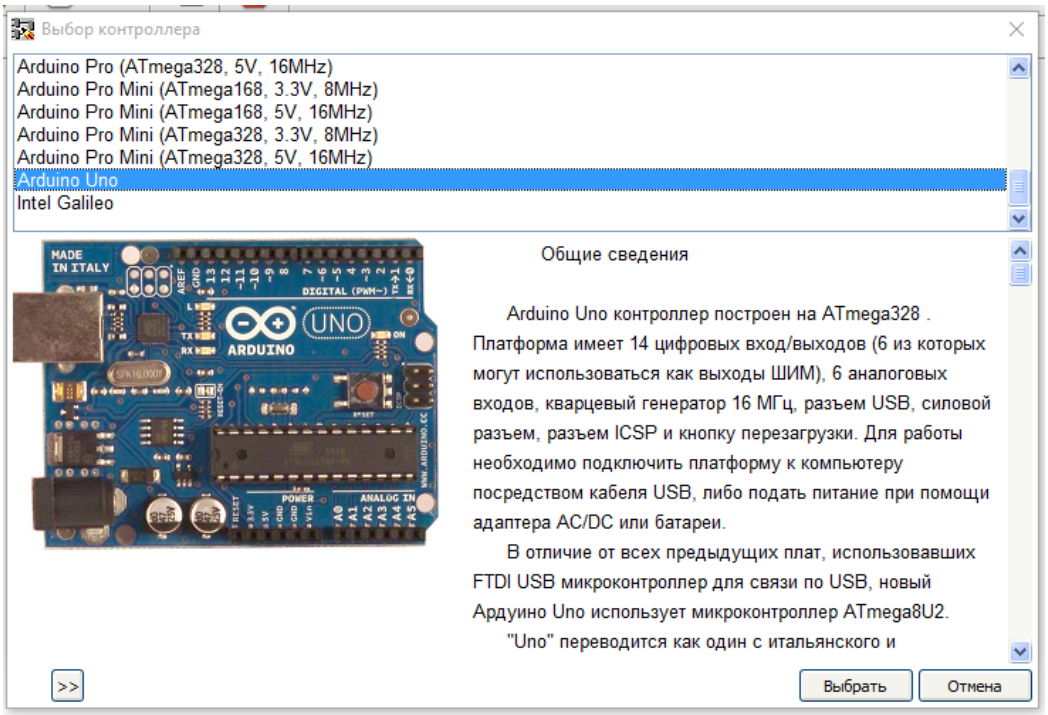


Рис. 13 — Окно выбора контроллера в FLProg

Рабочее окно программы FLProg (рис.14) на языке FBD состоит из нескольких полей:

1. Основное меню программы
2. Дерево проекта
3. Дерево установленного оборудования (**дерево тэгов**, если речь идет о разработке в FBD). В нём представлено оборудование (промежуточные реле, реле времени, генераторы…), которое используется в проекте. В новом проекте в нём присутствуют только входы и выходы контроллера.
4. Библиотека блоков. В ней находится оборудование, которое возможно применить в проекте. В данном уроке нас будет интересовать только папка «Базовые блоки»
5. Область схемы, в которой и будет собственно рисоваться схема. Схема в FLProg представляет собой набор плат с оборудованием.

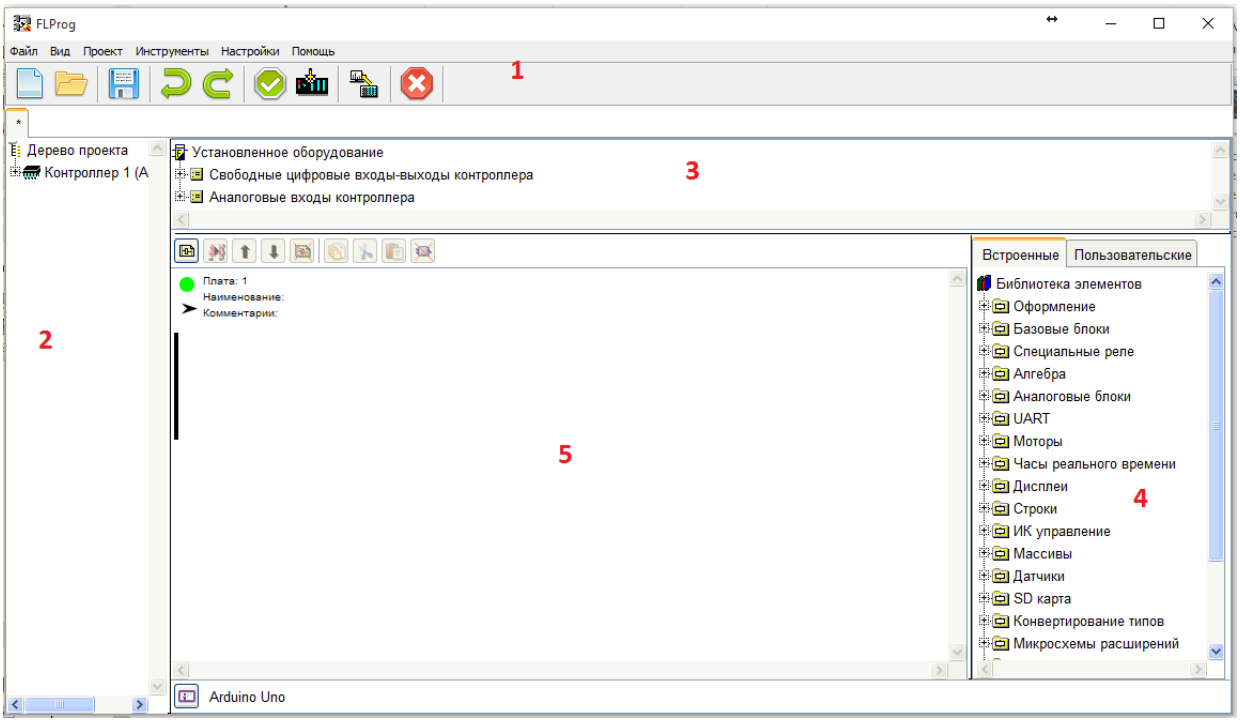


Рис. 14 — Интерфейс IDE FLProg

Для начала вытащим на область схемы контакты кнопок. Это возможно сделать двумя путями.

Перетащить соответствующий вход из папки (рис.15) «Свободные входы–выходы контроллера» дерева установленного оборудования на область схемы:

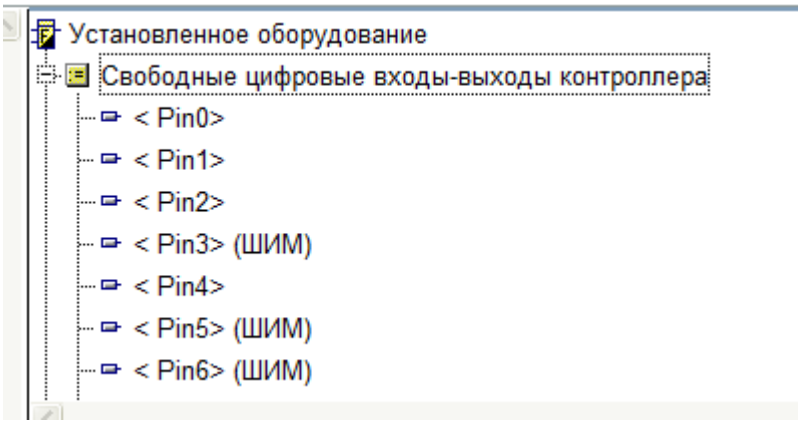


Рис. 15 — «Свободные входы–выходы контроллера»

Или (рис.16) перетащив блок «Контакт» из папки «Базовые элементы» библиотеки блоков:

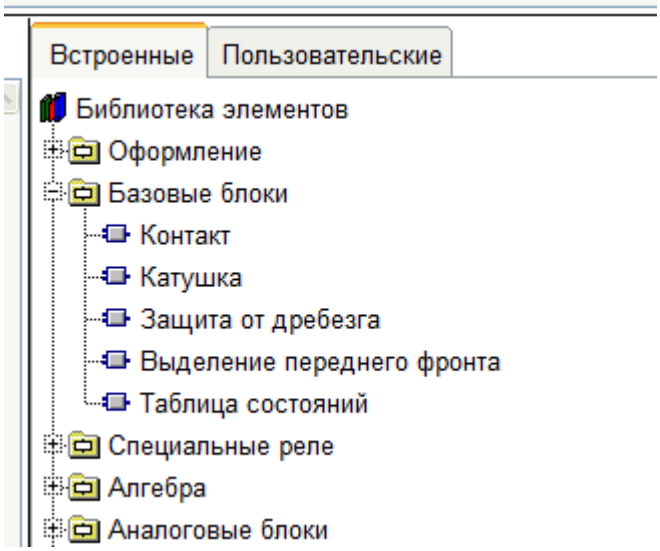


Рис. 16 — «Библиотека элементов»

В результате на схеме появится УГО (условно – графическое обозначение) контакта. В случае перетаскивания его из дерева установленного оборудования контакт окажется сразу привязанным к цифровому входу – выходу платы.

Если блок контакта был вытащен из библиотеки элементов, он будет абстрактным контактом без какой – либо привязки (рис.17):

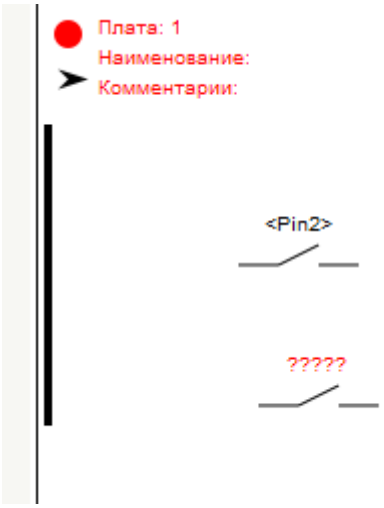


Рис. 17 — Абстрактный контакт

И любом случае контакты необходимо параметрировать. Для этого делаем двойной клик на контакте. Открывается окно редактирования блока (рис.18):

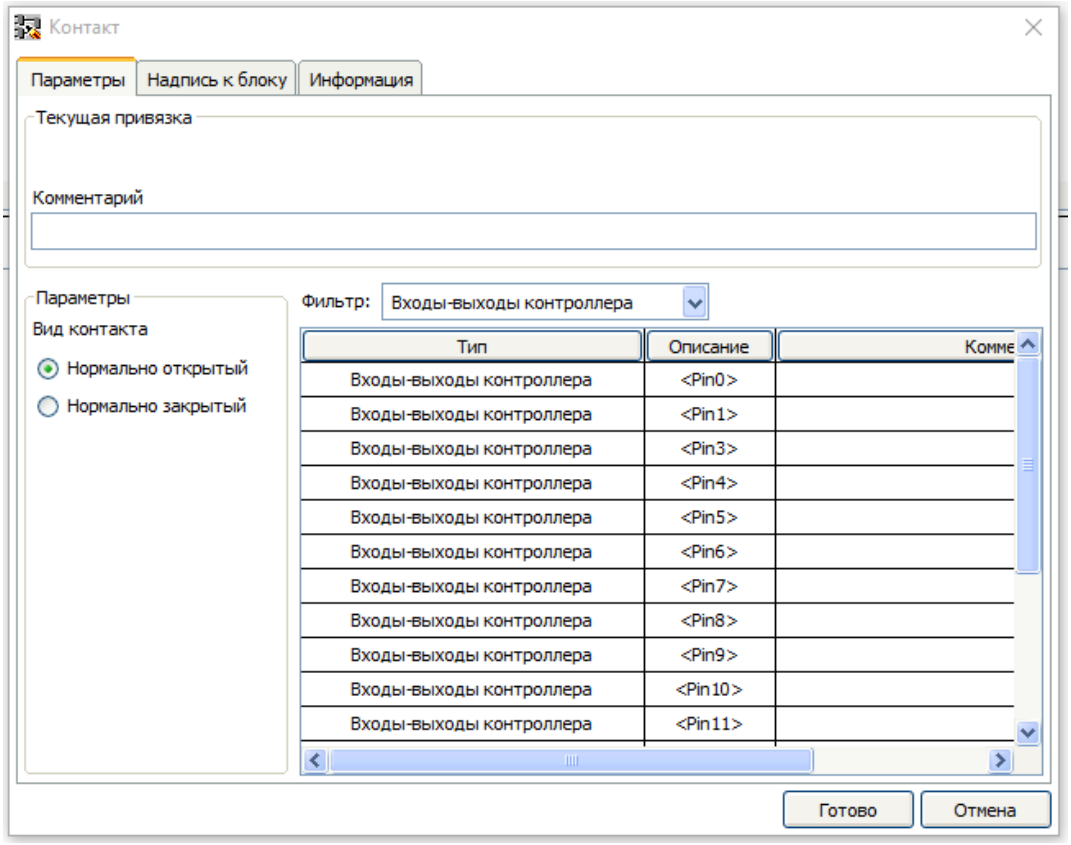


Рис. 18 — Параметризация элемента (контакта)

Вернее к заданию – в первую очередь «создадим» входы:

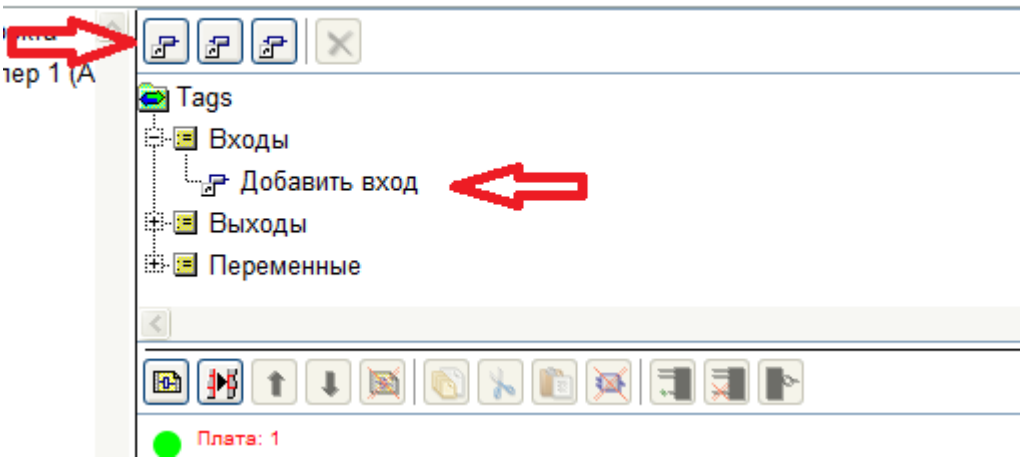


Рис. 19 — Панель создания входов в дереве тэгов.

Выбираем цифровой вход, появляются новые параметры. Записываем название входа, выбираем нужный вход платы Ардуино, и ставим галочку «Включить подтягивающий резистор».

Таким же образом добавляем все необходимые входы:

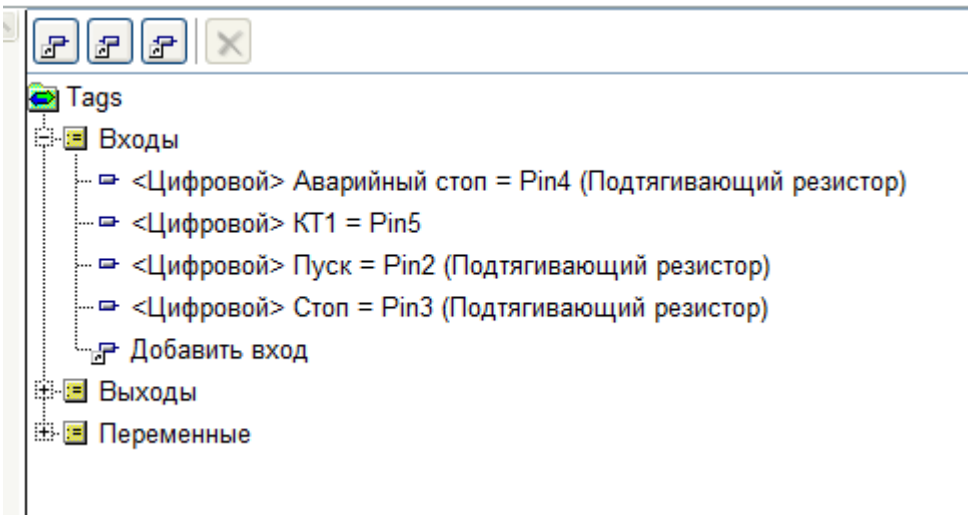


Рис. 20 — Созданные входы

Затем создаем переменную, отвечающую за состояние контактора. Для этого либо нажимаем на кнопку «Добавить переменную», либо делаем двойной клик на пункте «Добавить переменную» в дереве тэгов.

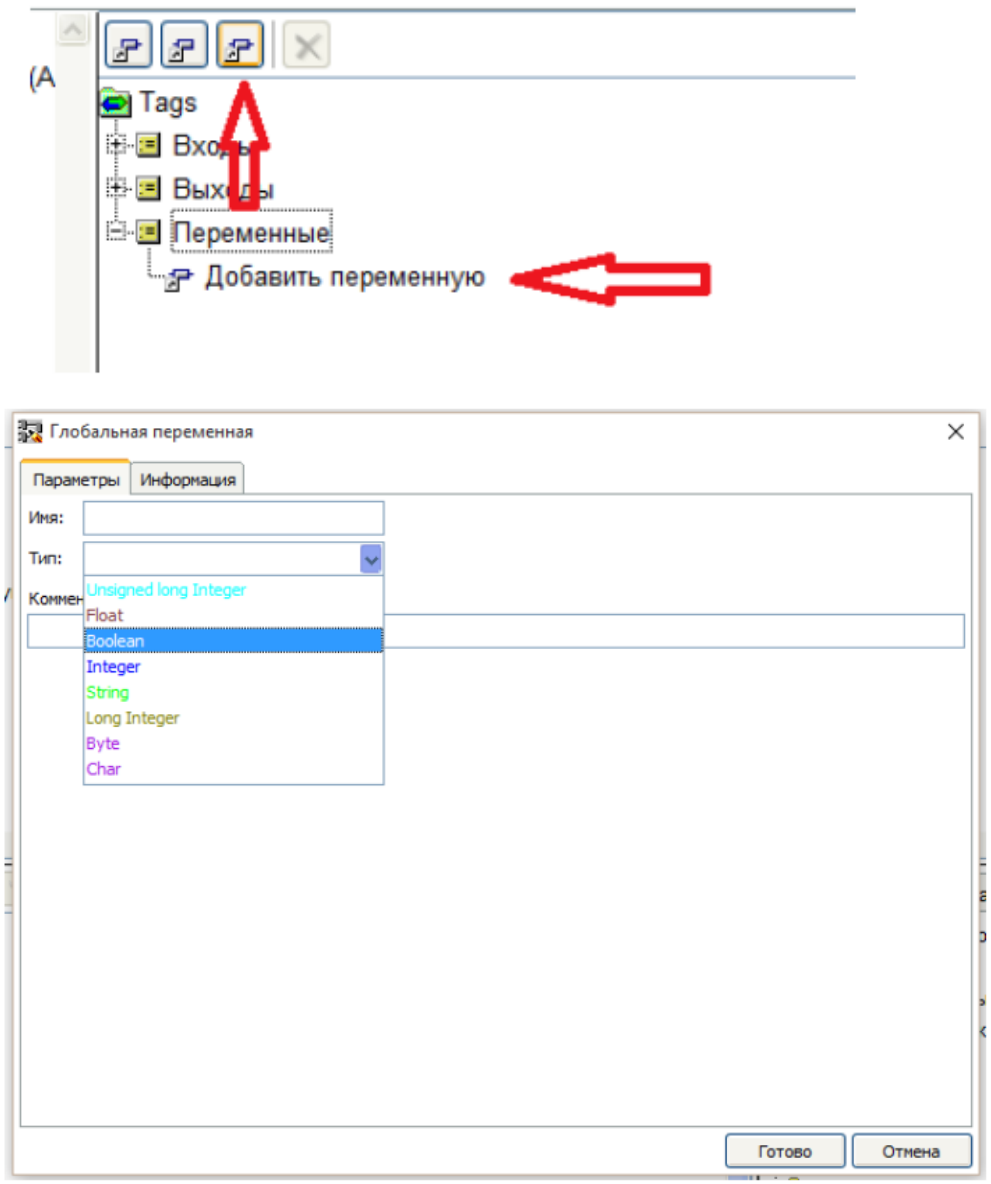


Рис. 20 — Окно настройки типа переменной (выбираем – Boolean)

Блоки входа на языке FBD соответствуют реальным выходам платы, следующим образом. Когда на реальном входе 0 – на выходе блока – False, когда на входе платы 5В на выходе блока True.

Для запоминания состояния контактора используем **RS триггер**. Его надо перетащить из папки «Триггеры» библиотеки блоков на рабочее поле схемы.

RS-триггер, или SR-триггер — триггер, который сохраняет своё предыдущее состояние при нулевых входах и меняет своё выходное состояние при подаче на один из его входов единицы.

При подаче единицы на вход S (от англ. Set — установить) выходное состояние становится равным логической единице.

А при подаче единицы на вход R (от англ. Reset — сбросить) выходное состояние становится равным логическому нулю.

При логическом нуле на обоих входах на выходе удерживается последнее состояние. При логических единицах на обоих входах в случае RS триггера выход устанавливается в логический ноль, а в случае SR триггера в логическую единицу.

Для того что бы включился контактор необходимо подать на вход S сигнал со входа «Пуск». Для этого перетаскиваем из дерева тэгов вход «ПУСК» на рабочую область схемы. Если вспомнить о том, что при нажатии кнопки Пуск на вход платы подаётся логический 0, то понятно, что необходимо инвертировать сигнал с кнопки. Для этого наведём курсор на вход S триггера и кликнем правой кнопкой мыши (рис.21):

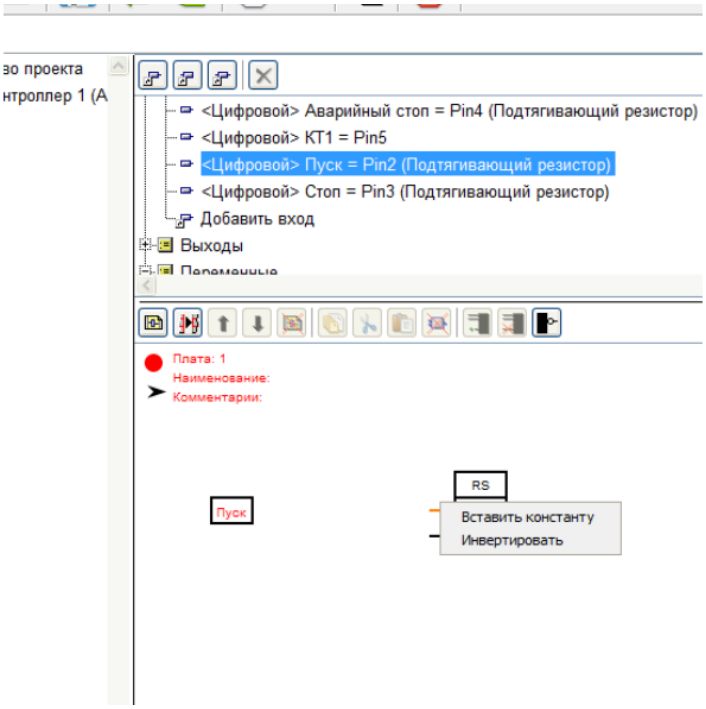
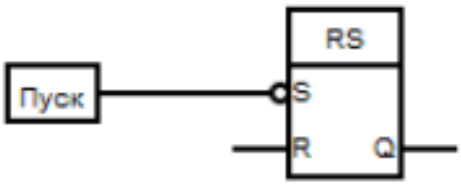


Рис. 21 — В открывшимся меню в пункт выберем «Инвертировать»

После чего соединяем вход S триггера с выходом блока входа «Пуск».



Остановка контактора происходит если:

Нажата кнопка «СТОП» (лог.1 на блоке входа «Стоп») ИЛИ нажата кнопка «АВАРИЙНЫЙ СТОП» (лог.1 на блоке входа «Аварийный стоп») ИЛИ сработало тепловое реле (лог.1 на блоке входа «КТ1»).

Значит, нам нужен блок ИЛИ с тремя входами. Перетаскиваем его из библиотеки блоков из папки «Базовые блоки».

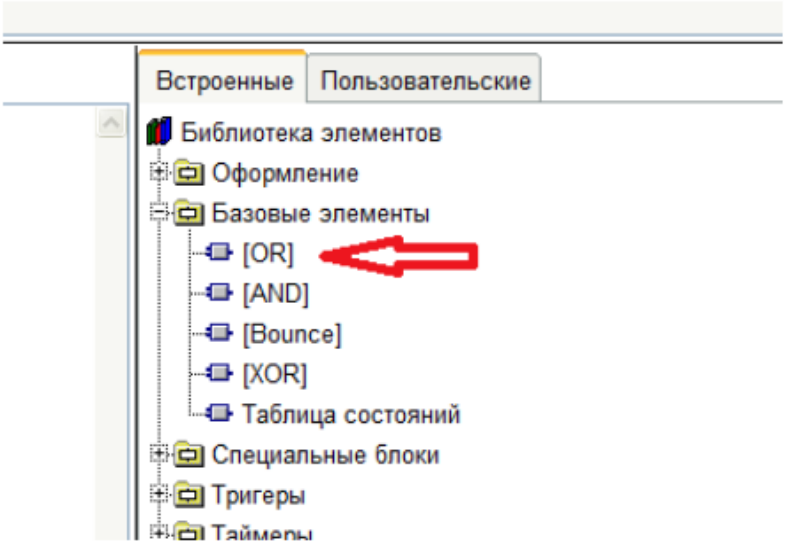


Рис. 22 — Базовый блок OR (ИЛИ)

По умолчанию у блока ИЛИ два входа. Для того что бы добавить третий, выделяем блок и нажимаем кнопку «Добавить вход» (рис.23):

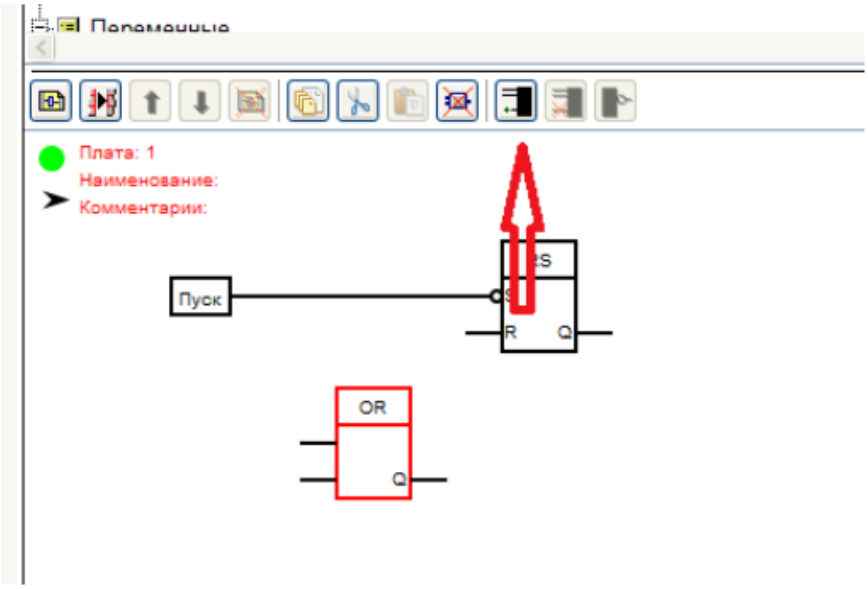


Рис. 23 — Увеличение ходов у логического блока ИЛИ

Переносим необходимые входы из дерева тэгов и соединяем со входами блока ИЛИ. А выход блока ИЛИ соединяем с входом R триггера.

Затем забираем из дерева тэгов переменную «Состояние контактора» и выход триггера соединяем со входом этой переменной:

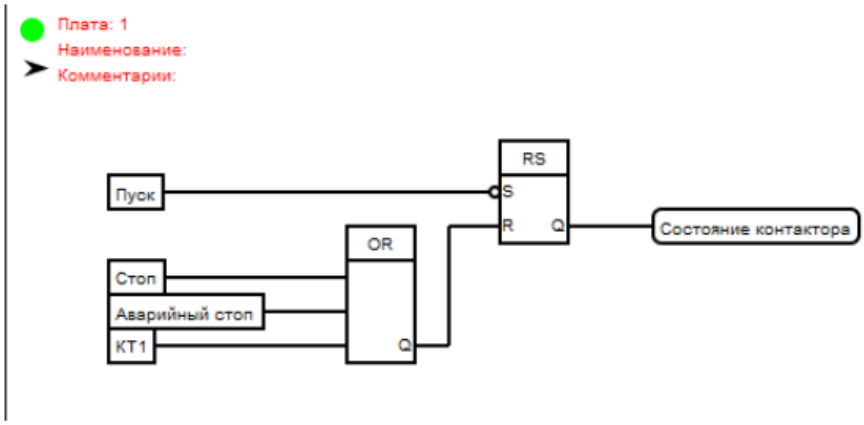


Рис. 24 — «Плата управления»

На этом закончим первую плату – «Плата управления».

После чего создадим новую плату и сразу назовём её «Управление выходами».

Далее создадим выходы платы в соответствии со схемой. Для этого надо кликнуть на кнопку «Добавить выход» дли сделать двойной клик на пункте «Добавить выход» в дереве тэгов. Выходы создаём цифрового типа.

Перетащим на вторую плату созданные выходы, вход КТ1 и переменную «Состояние контактора» Затем соединим блоки в соответствии со схемой. Необходимые входы блоков инвертируем.

Рис. 25 — «Плата управление выходами (выводами)»

Обратите внимание, что при перетаскивании на схему блоков входа, выхода или переменной изначально у них нет входов или выходов. Они появляются при подведении курсора к блоку в месте их будущего расположения.

С созданием схем закончили. Теперь надо залить программу в контроллер. Для этого нажимаем кнопку «Компилировать проект».

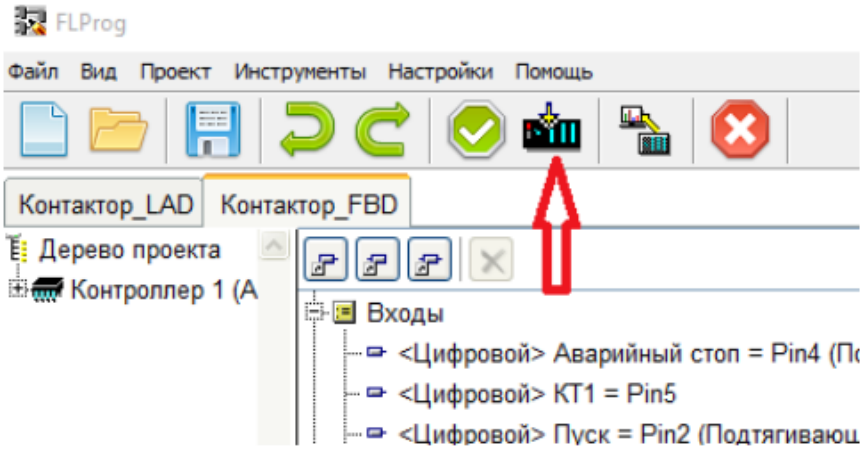


Рис. 26 — Кнопка компиляции проекта в FLProg

После компиляции должна открыться IDE Arduino. Изучаем транслированный в привычный текстовый вид код на языке С.

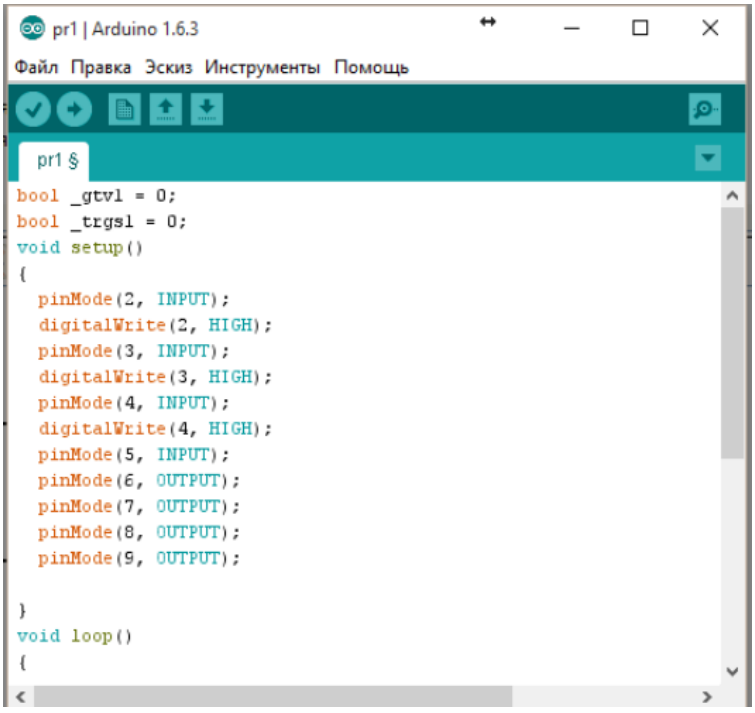


Рис. 27 — Arduino-IDE с открытым скетчем, в который была преобразована созданная схема.

После проделанных шагов вам необходимо перейти к настройке интегрированной среды разработки Arduino.

Подключите плату Arduino UNO R3 к любому из доступных и удобных вам портов USB. Определите номер виртуального COM-порта (в реальности - USB), на котором находится плата (левый нижний угол или диспетчер устройств, там должен быть драйвер CH340g.

Микросхема CH340G – преобразователь интерфейса USB в UART (мост USB-UART). Характеристики, условия эксплуатации, типовые схемы включения.

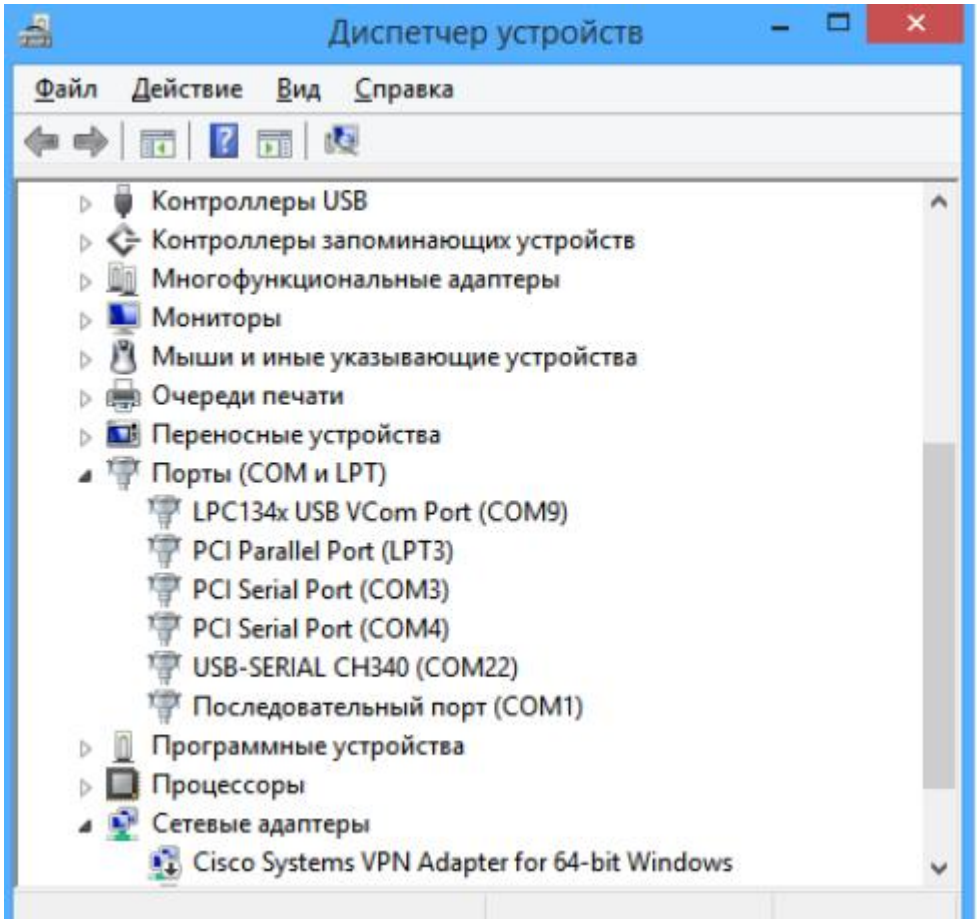


Рис. 27 — Определение номера COM-порта (в данном случае COM22)

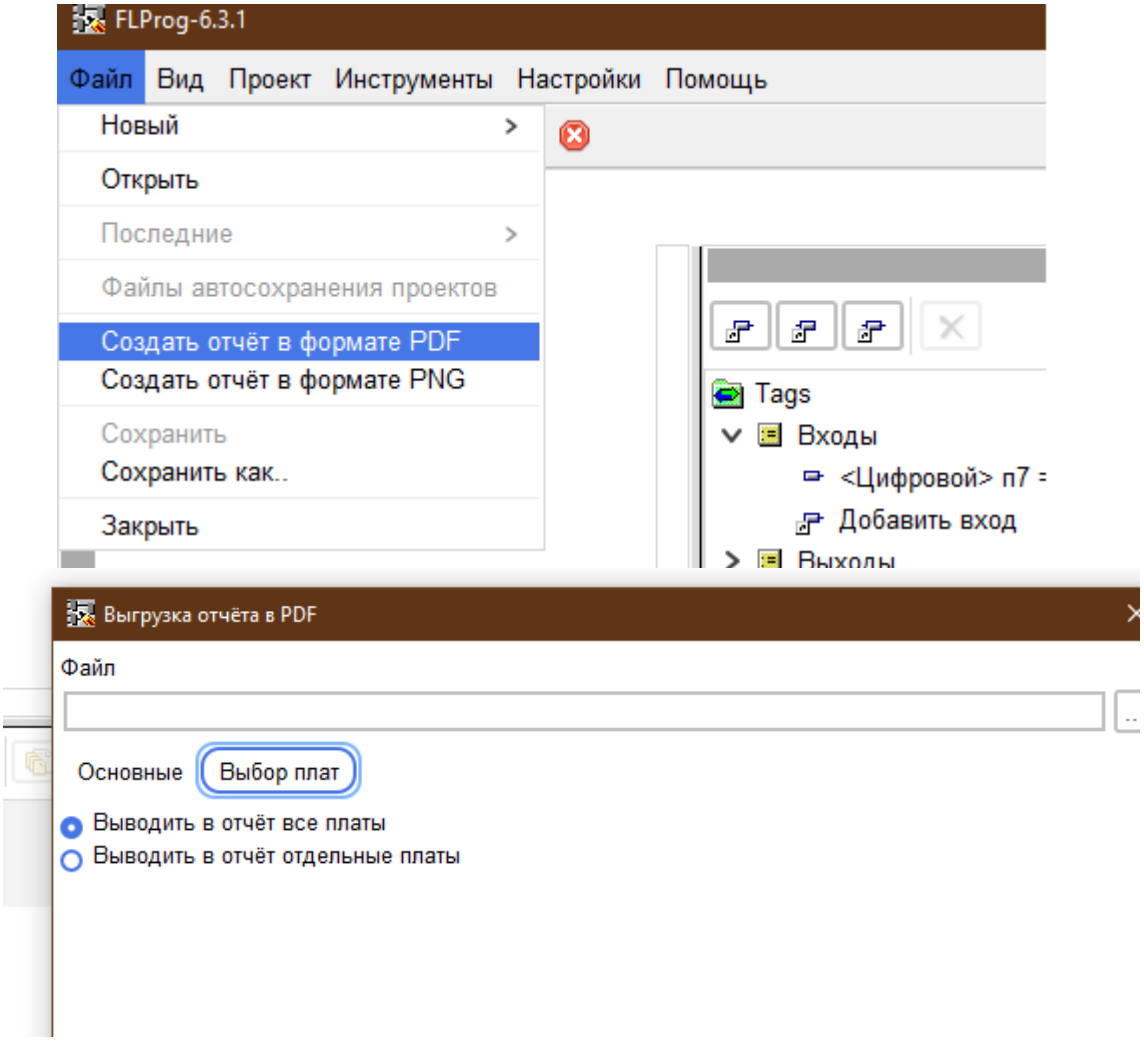
Вкладка: **Инструменты** – **Порт** (последовательный порт): выбираем в нашем случае 22 (обычно цифры меньше – 1-9).

Тип платы: **Arduino UNO**.

Перед компиляцией в Arduino IDE (процессом программирования самого устройства) необходимо удостовериться в правильности настройки (воспользуйтесь обязательно помощью преподавателя или ассистента (лаборанта).

Требования к отчету:

1. Ссылка на предоставляемый открытый доступ к видео-демонстрации по заданным требованиям.
2. Копия готового проекта в FLProg в расширении .flp («Сохранить как» с именованием следующего вида: Пр1_ФАМИЛИЯ_ГРУППА)
3. Отчет в PDF по всем платам (с аналогичным именованием);
4. Код программы в оформленном в Microsoft Word/LibreOffice виде, в типологии отчета с титульной страницей.
5. Скриншот (в отчет) описания базового элемента OR, которая находится в встроенной справке программы FLProg.



§ ПРАКТИКУМ: СОЗДАНИЕ СИСТЕМЫ ОГРАНИЧЕНИЯ ДОСТУПА С ПРИМЕНЕНИЕМ RFID.

Оборудование: ПК, кабель AM/microBM 5p Cablexpert Pro (CCP-mUSB2-AMBM-1.0M или аналоги), Arduino UNO R3, RFID модуль RFID RC522, соединительные провода.

Программное обеспечение: IDE FLProg, IDE Arduino.

Цель: научиться реализовывать простую систему контроля и управления доступом, получить навыки работы с радиочастотной идентификацией посредством FBD.

Теоретические сведения:

Система контроля и управления доступом, СКУД (англ. Physical Access Control System, PACS) — совокупность программно-аппаратных технических средств контроля и средств управления, имеющих целью ограничение и регистрацию входа-выхода объектов (людей, транспорта) на заданной территории через «точки прохода»: двери, ворота, КПП.

Основная задача — управление доступом на заданную территорию (кого пускать, в какое время и на какую территорию), включая также:

ограничение доступа на заданную территорию;

идентификацию лица, имеющего доступ на заданную территорию.

Дополнительные задачи:

учёт рабочего времени;

расчет заработной платы (при интеграции с системами бухгалтерского учёта);

ведение базы персонала / посетителей;

интеграция с системой безопасности, например:

с системой видеонаблюдения для совмещения архивов событий систем, передачи системе видеонаблюдения извещений о необходимости стартовать запись, повернуть камеру для записи последствий зафиксированного подозрительного события;

с системой охранной сигнализации (СОС), например, для ограничения доступа в помещения, стоящие на охране, или для автоматического снятия и постановки помещений на охрану.

с системой пожарной сигнализации (СПС) для получения информации о состоянии пожарных извещателей, автоматического разблокирования эвакуационных выходов и закрывания противопожарных дверей в случае пожарной тревоги.

Идентификатор Основные типы исполнения — карточка, брелок, метка. Является базовым элементом системы контроля доступа, поскольку хранит код, который служит для определения прав («идентификации») владельца.

Это может быть Touch memory, бесконтактная карта (например, **RFID-метка**), или устаревающий тип карт с магнитной полосой. В качестве идентификатора могут выступать также коды, вводимый на клавиатуре, или отдельные биометрические признаки человека — отпечаток пальца, рисунок сетчатки или радужной оболочки глаза, трехмерное изображение лица.

Надежность (устойчивость к взлому) системы контроля доступа в значительной степени определяется типом используемого идентификатора: например, наиболее распространенные бесконтактные карты proximity могут подделываться в мастерских по изготовлению ключей на оборудовании, имеющемся в свободной продаже. Поэтому для объектов, требующих более высокого уровня защиты, подобные идентификаторы не подходят. Принципиально более высокий уровень защищенности обеспечивают RFID-метки, в которых код карты хранится в защищённой области и шифруется.

Кроме непосредственного использования в системах контроля доступа, RFID-метки широко применяются и в других областях. Например, в локальных расчетных системах (оплата обедов в столовой и других услуг), системах лояльности и так далее.

Контроллер - это «мозг» системы: именно контроллер определяет, пропустить или нет владельца идентификатора в дверь, поскольку хранит коды идентификаторов со списком прав доступа каждого из них в собственной энергонезависимой памяти. Когда человек предъявляет (подносит к считывающему устройству) идентификатор, считанный из него код, сравнивается с хранящимся в базе, на основании чего принимается решение об открытии двери.

Сетевой контроллер объединяется в единую систему с другими контроллерами и компьютером для возможности централизованного контроля и управления. В таком случае решение о предоставлении доступа может приниматься как контроллером, так и программным обеспечением головного компьютера.

Чаще всего объединение контроллеров в сеть осуществляется посредством промышленного интерфейса RS-485 или локальной сети Ethernet.

В случаях, когда необходимо обеспечить работу контроллера при авариях электросети, блок контроллера обеспечивается собственным аккумулятором, либо внешним блоком резервного питания. Время работы от аккумулятора может составлять от нескольких часов до нескольких суток.

Ход работы:

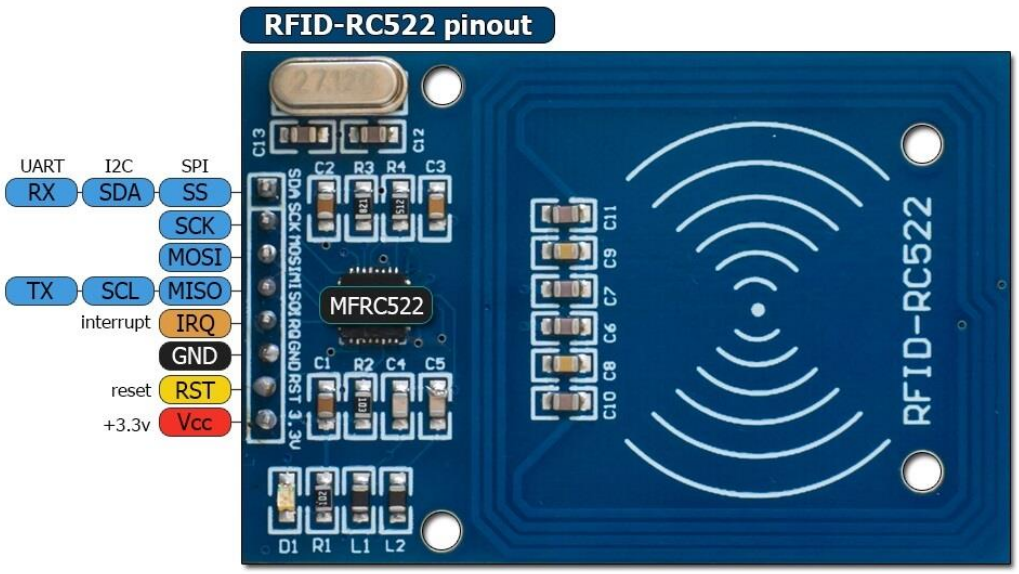
1. Подключить Arduino UNO, определив номер отождествляемого COM-порта.
2. Запустить и настроить Arduino IDE на соответствующий порт, плату.
3. Изучить характеристики модуля RFID RC522:

Технические характеристики RFID-модуля RC522

Напряжение питания: **3.3V**;
Потребляемый ток :13-26mA;
Рабочая частота: 13.56MHz;
Дальность считывания: 0 - 60 мм;
Интерфейс: SPI;
Скорость передачи: максимальная 10МБит/с;
Размер: 40мм x 60мм;

Назначение выводов интерфейса SPI:

SDA – выбор ведомого; SCK –сигнал синхронизации;
MOSI – передача от master к slave; MISO – передача от slave к master;
RST – вывод для сброса; IRQ – вывод прерывания;
GND – земля; Vcc –питание **3.3 В.**



4. Сканер бесконтактных меток RFID-RC522 без меток позволяет обнаружить и считать идентификаторы бесконтактных карт, меток, пропусков стандарта 13,56 МГц на расстоянии до 6 см. Благодаря данному сканеру можно сделать ряд интересных проектов: пропускные системы, электронные замки, складской учету и много другое. Предложить прикладной вариант применения данного модуля.
5. Запустить IDE FLprog. Создать проект в FBD для Arduino Uno (Atmega328). Перейти к изучению функциональных блоков:

В программе FLProg реализовано два вида функциональных блоков для работы со сканером.

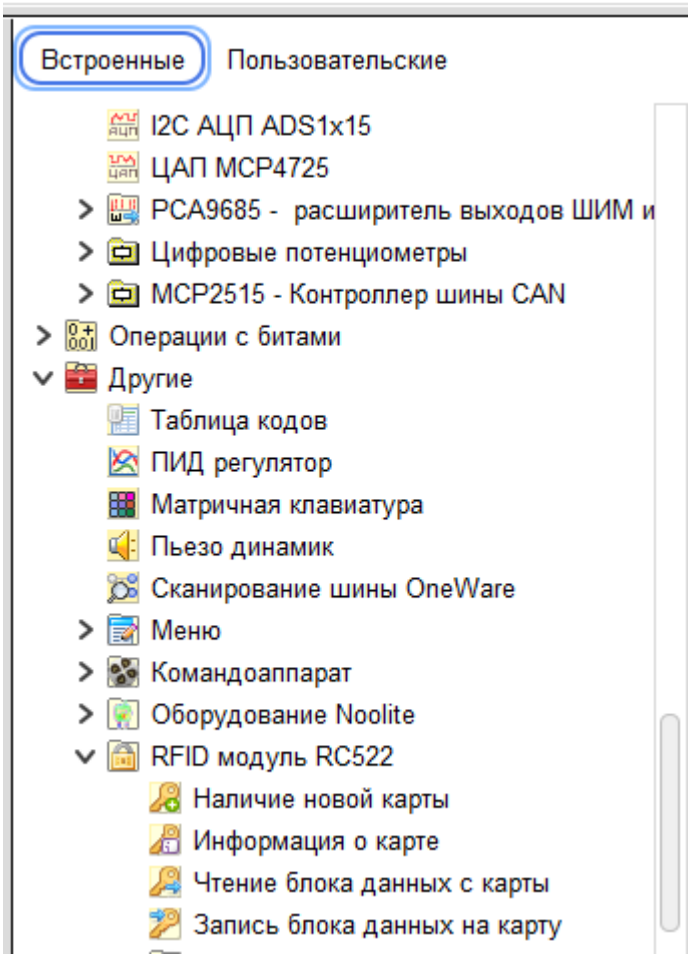
Блоки непосредственно связанные со сканером

«Наличие новой карты»

«Информация о карте»

«Чтения бока данных на карту»

«Запись блока данных на карту»



Блоки, связанные со сканером косвенно и предназначенные для работы с хранилищем UUID карт.

В программе реализовано хранилище UUID карт, представляющее собой набор ячеек каждая из которых хранит непосредственно UUID карты, и статус ячейки.

Три статуса зарезервировано программой:
0x00 – ячейка свободна;
0X01 – в ячейке хранится UUID но она заблокирована;
0x02 — в ячейке хранится UUID и она активна;

UUID (англ. universally unique identifier «универсальный уникальный идентификатор») — это стандарт идентификации, используемый в создании программного обеспечения, стандартизированный Open Software Foundation (OSF) как часть DCE — среды распределённых вычислений.

Остальные коды статуса (3 … 255) пользователь может использовать по своему усмотрению.

Хранилище может располагаться в оперативной памяти контроллера (**при снятии питания или перезагрузке оно очистится**), или в **EEPROM**. Хранилищ может быть несколько, и они могут быть разных типов. При расположении хранилища в EEPROM размеры его ограниченны. Для Arduino Uno – это максимум 85 ячеек (во всех хранилищах EEPROM в сумме), для Arduino Mega – 341 ячейка.

Блоки для работы с хранилищем
«Сохранить UUID карты в хранилище»
«Прочитать UUID карты из хранилища»
«Статус ячейки в хранилище»
«Записать статус ячейки в хранилище»
«Блокировка/разблокировка ячейки»
«Поиск UUID в хранилище»
«Свободные ячейки хранилища»
«Очистка ячейки в хранилище»
«Очистка всего хранилища»

Техническое задание:

Пусть в устройстве будут два хранилища расположенных EEPROM. В первом хранятся так называемые Master Card. С помощью них можно записывать обычные карты во второе хранилище.

Для записи Master Card существует «Секретная кнопка». Так же допускается дисплей для отображения необходимой информации, и обычная управляющая кнопка. Создать систему разграничения доступа по следующей схеме (рис.28):

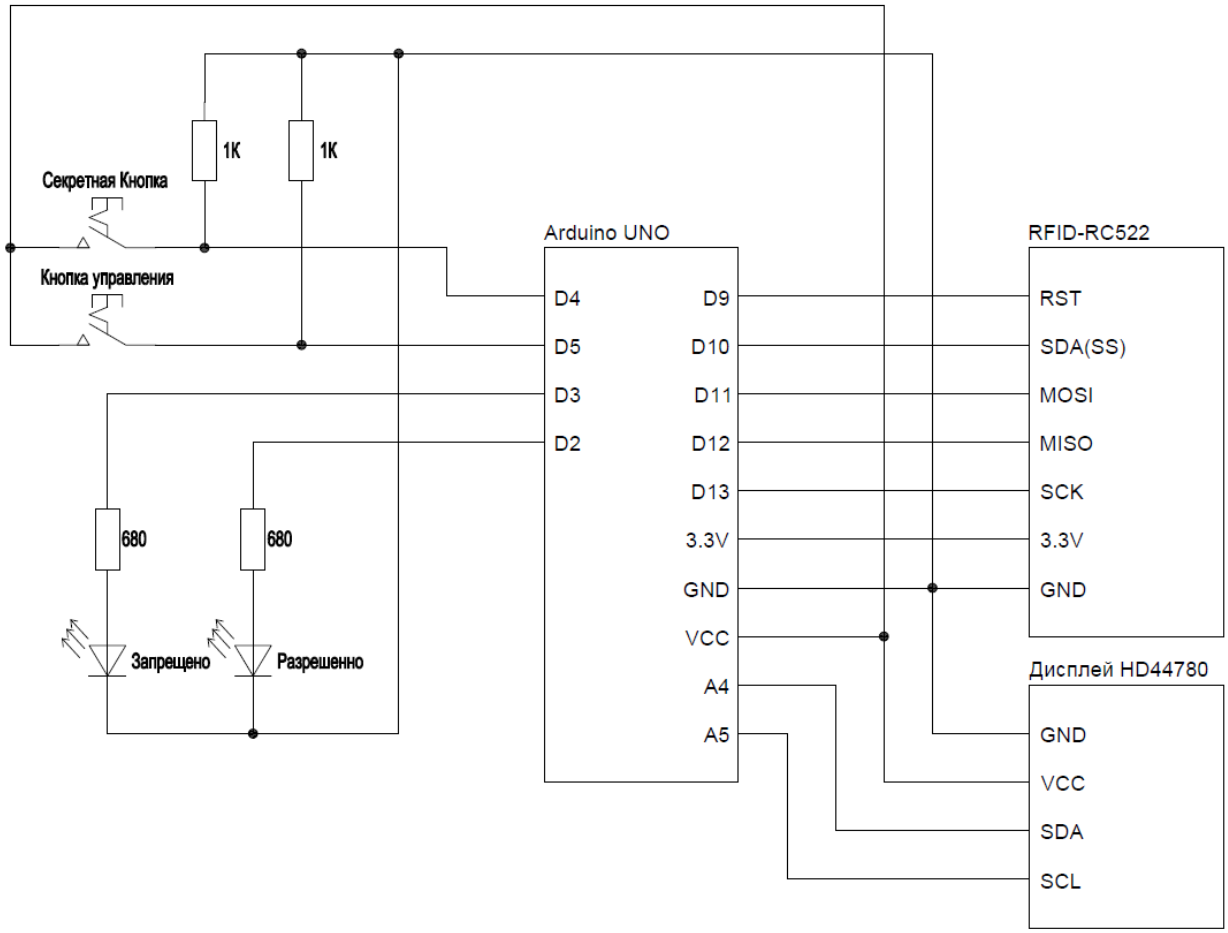
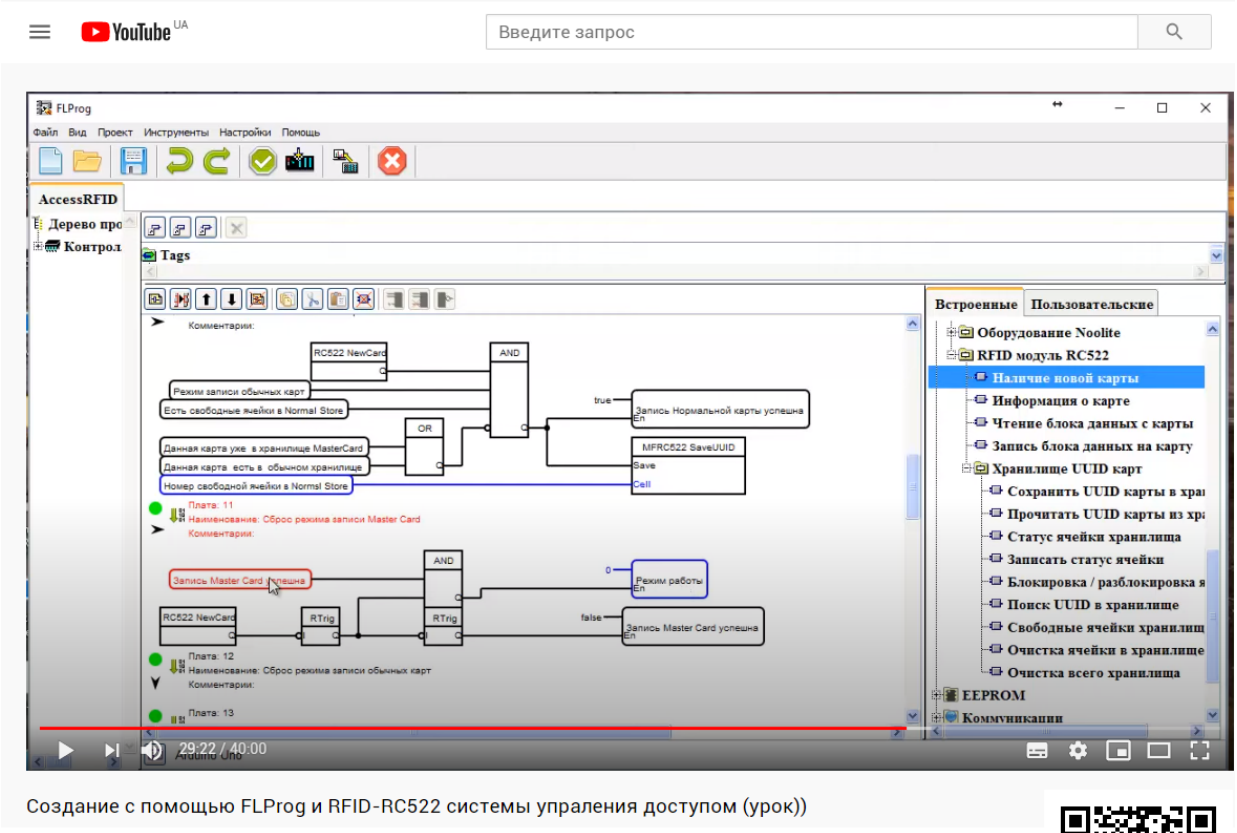


Рис. 28 — Схема разрабатываемого устройства

Карты и брелки используемые вместе со сканером имеют «на борту» 1 кБ памяти которые можно использовать в собственных целях. Давайте расширим техническое задание и будем записывать, какую-либо информацию.

Пошаговая видео-демонстрация выполнения практического задания в открытом доступе по ссылке: https://youtu.be/AXWDZEUY-6I



Создание с помощью FLProg и RFID-RC522 системы упраления доступом (урок))

Требования к отчету:

1. Привести код программы и FBD-грамму.
2. Описать все выходы и входы устройства по вышеизложенной схеме.
3. Привести имена типов карт (RFID-типы), поддерживающих данный модуль, указать частоту. Дать кратких обзор типов RFID-меток.
4. Подготовить реферат на защиту по темам (для лиц, упомянутых преподавателем по списку):

    1. Дальняя идентификация (Идентификация на пассивных метках дальнего считывания);

    2.Альтернативные методы автоматической идентификации;

    3. Информационная безопасность RFID. Вирусы и атаки в радиочастотной идентификации;

    4. RFID и IoT/IoE.

§ ПРАКТИКУМ: СОЗДАНИЕ ВСТРАИВАЕМОЙ СИСТЕМЫ ПО ТЕХНИЧЕСКОМУ ЗАДАНИЮ

Оборудование: персональный компьютер.

Программное обеспечение: IDE FLProg, IDE Arduino.

Цель: научиться реализовывать конечные проекты встраиваемых компьютерных систем посредством среды визуального программированию по техническому заданию.

Теоретические сведения:

Встраиваемая система (встроенная система, англ. embedded system) — специализированная микропроцессорная система управления, контроля и мониторинга, концепция разработки которой заключается в том, что такая система будет работать, будучи встроенной непосредственно в устройство, которым она управляет.

Шаблон задания:

Разработать встраиваемую систему на базе любого микроконтроллера семейства Arduino/ESP, которая будет считывать и записывать в постоянную память показания температуры и влажности воздуха. Обеспечить звуковую или светодиодную индикацию при достижении уровня влажности (относительной или абсолютной – на выбор) более 79%.

Ход работы:

1. Задание выбирается индивидуальным образом – должен соблюдаться подход равноценности и равнозначности выбираемого задания с учетом предпочтений студента. Для этого рекомендуется провести семинар по теме анализа компонентов (датчиков, исполнительных устройств, средств идентификации и аутентификации, специальных устройств ввода-вывода и обработки информации, представляемых в среде FLProg.
2. Студент предоставляет Datasheet по каждому из необходимых устройств (за исключением главного микроконтроллера) в печатном виде, в которой обязательно должна быть оформлена первая титульная печатная страница для рецензирования преподавателем.
3. Студент предоставляет собранную схему в FLProg в электронном формате (flp), код программы на C++ на защиту.

§ RemoteXY и Bluetooth ВЗАИМОДЕЙСТВИЕ С ПРОГРАММОЙ FLPROG.

RemoteXY - сервис для организации удаленного беспроводного и проводного управления устройствами на базе arduino, esp8266, esp32 и chipKIT. Сервис состоит из онлайн-редактора мобильного интерфейса, приложения для устройств Android и iOS, сервера для удаленного управления устройствами, а также библиотеки для ArduinoIDE.

Приложений для Android устройств два - бесплатное и платное. Бесплатное приложение имеет ограничение в 5 элементов интерфейса. Если же элементов интерфейса больше, то приложение позволяет протестировать работу устройства в течении ограниченного времени (всего 30 секунд за сеанс).

Работа с сервисом RemoteXY организована следующим образом:

- в онлайн-редакторе необходимо выбрать программируемое устройство и способ его подключения, затем настроить способ подключения;
- при помощи онлайн-редактора необходимо создать интерфейс программируемого устройства, для этого предусмотрены элементы управления и индикации, такие как кнопки, переключатели, поля ввода и вывода текста, графики;
- получить сгенерированный онлайн-редактором исходный код, перенести его в ArduinoIDE (также поддерживаются FLProg и MPIDE), дополнить необходимым кодом и загрузить в программируемое устройство;
- при помощи приложения подключиться к устройству.

Отметим, что идентификатор интерфейса хранится непосредственно в программируемом устройстве в виде массива чисел, и загружается в приложение для смартфона при каждом подключении. Таким образом, отпадает необходимость настройки интерфейса на каждом подключаемом к устройству смартфоне, необходимо лишь настроить связь между смартфоном и запрограммированном устройстве.

Редактор интерфейса разделен на 3 части. В центре основное окно **(рис.29)**, в котором отображается внешний вид созданного интерфейса. Слева раскрывающиеся списки с элементами интерфейса, сгруппированные по категориям. Справа расположены вкладки конфигурации подключения устройства, настройки экрана и контекстно-зависимая вкладка настройки выбранного элемента интерфейса.

Работа с онлайн-редактором строится по принципу drag-and-drop, также, как и в проводнике компьютера. Просто перетаскивайте необходимый элемент с левой области в центральную, на макет смартфона.

При клике на элемент на макете смартфона он выделяется синей рамкой и теперь его можно перемещать по макету, а также масштабировать, хватая и перемещая маркеры по углам рамки. Обратите внимание, над рамкой выбранного элемента отображается имя переменной, по которому в дальнейшем будет осуществляться доступ к выбранному элементу.

рис. 29 — Основное окно RemoteXY

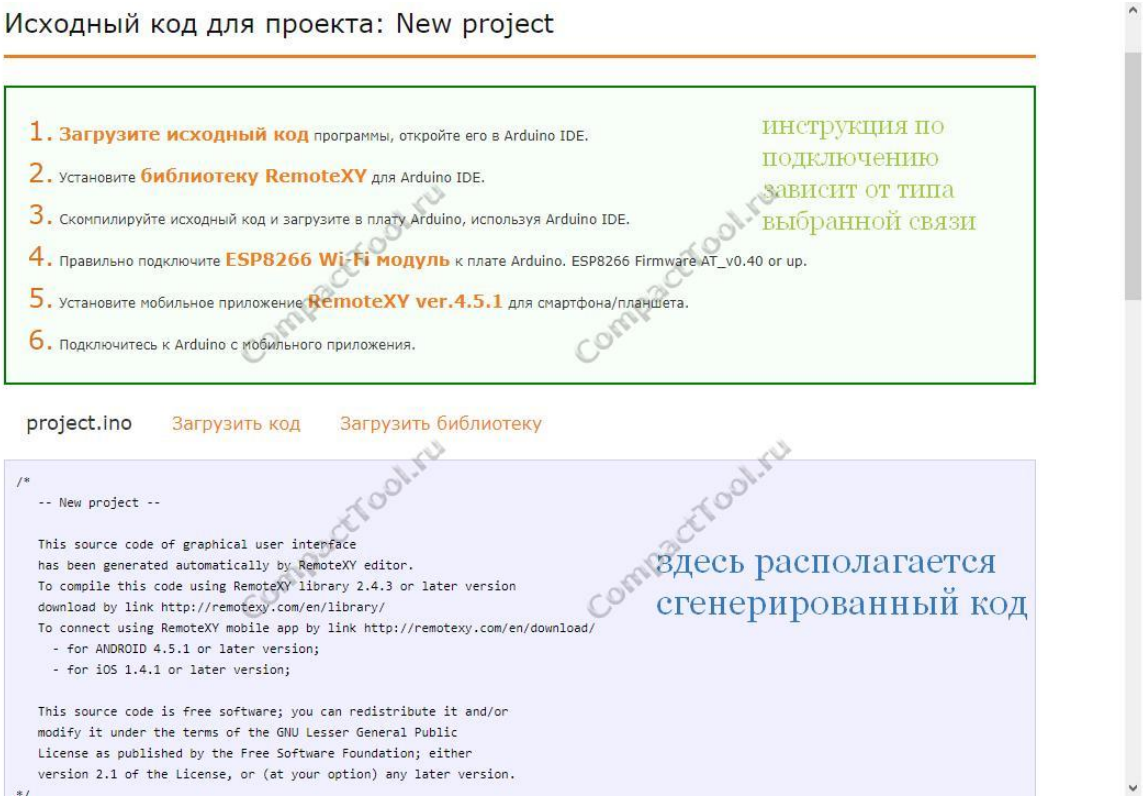
Начать знакомство с RemoteXY разработчиками программы рекомендуется с изучения способов связи приложения и устройств, которыми необходимо управлять. Но перед этим создадим простейший интерфейс следующим образом:

- перетащите на макет смартфона элемент “выключатель”;
- убедитесь, что в настройках выключателя в выпадающем списке “привязать к выводу” выбран вывод с надписью “LED”;

Выглядеть это должно так (рис.30):

рис. 30 — Созданный простейший интерфейс

Для получения сгенерированного кода необходимо нажать большую зеленую кнопку “Получить исходный код”:



Для доступа к вариантам совместимых соединений необходимо раскрыть вкладку “конфигурация” в правой части онлайн редактора и нажать мышкой на любой из появившихся значков.

Список всех возможных соединений приведен на рис.31:

рис. 31 — Совместимые соединения в среде RemoteXY

Соединение при помощи кабеля требует **наличие функции OTG** в смартфоне. К сожалению, не все смартфоны могут похвастаться наличием такой функции. Соединение такого типа доступно для всех плат прототипирования, которые имеют на плате распаянный переходник USB-UART. При отсутствии распаянного переходника USB-UART (как в случае с arduino pro mini) можно подключить внешний.

Продемонстрируем соединение «через кабель» на примере Arduino Mega2560.

Используем следующую конфигурацию подключения:

- соединение: USB OTG
- устройство: Arduino Mega2560
- модуль: USB-UART
- среда разработки: ArduinoIDE.

Далее необходимо выбрать программный или аппаратный протокол, выберем аппаратный (Hardware Serial). Выбираем Serial port к которому подключен переходник USB-UART, который размещен на плате **Mega2560.**

Напоследок устанавливаем скорость соединения 9600 бод. Напомним, что в интерфейсе сейчас находится один элемент управления - переключатель, который зажигает расположенный на плате светодиод. После загрузки скомпилированного кода в микропроцессорную систему мы подключаем её к своему смартфону.

Процесс подключения показан на следующих скриншотах.

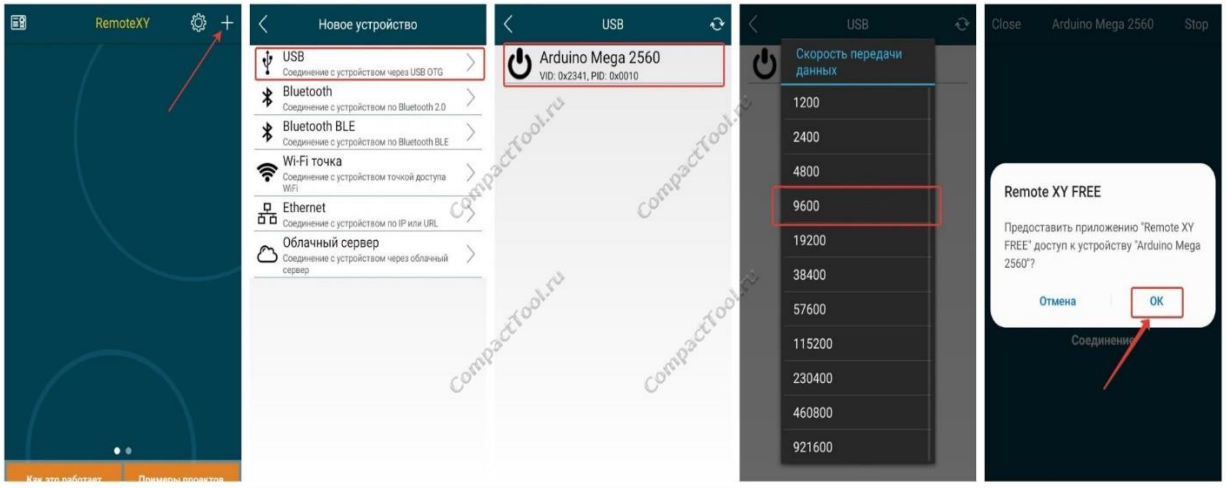


рис. 32 — Подключение Arduino Mega 2560 в RemoteXY

Это очень специфичный способ связи смартфона и управляемого устройства. Для **постоянного соединения годится только** в некоторых случаях, но очень подходит для не частого подключения, например, для настройки управляемого устройства.

Примечание: этот способ соединения не работает с Arduino Leonardo.

Следующий **тип связи** - **BlueTooth**. Возможна работа с модулями **HC-05, HM-10** и встроенным в ESP32 bluetooth on chip.

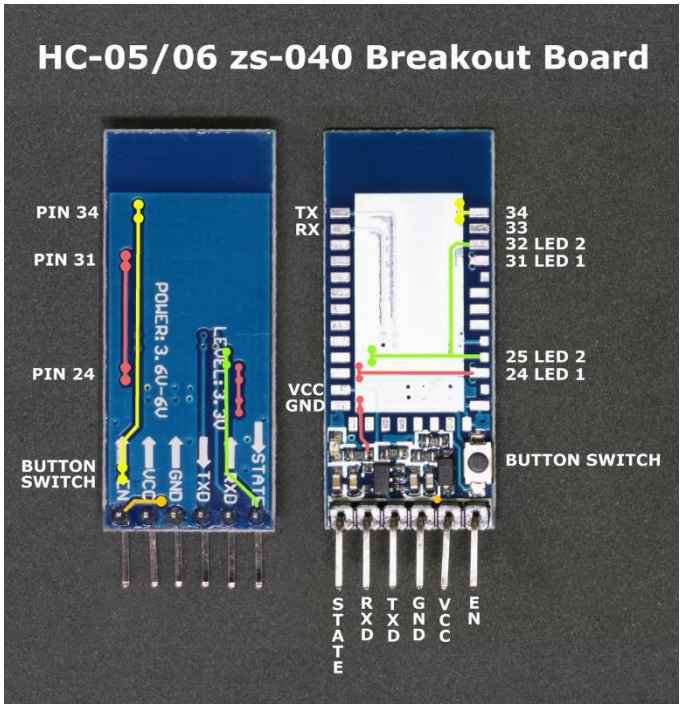
Данный способ связи годится для использования на небольших расстояниях, ограниченных дальностью связи модуля bluetooth. Отлично подойдет для устройств умного дома без доступа к сети интернет, отдельных “умных” устройств, управляемых со смартфона.

Особенностью данного способа связи является то, что смартфон может поддерживать связь с несколькими устройствами одновременно, что не достижимо при помощи следующего еще одного способа взаимодействия - точки доступа (AP).

Arduino UNO + Bluetooth HC-05(06) в RemoteXY

К контроллеру **Arduino UNO** необходимо подключить модуль Bluetooth HC-05, HC-06 (рис. или совместимый. Смартфон или планшет должен поддерживать Bluetooth.

Примечание: Устройства с ОС iOS не поддерживают модули HC-05(06). Вместо них вы можете использовать модуль Bluetooth BLE HM-10.



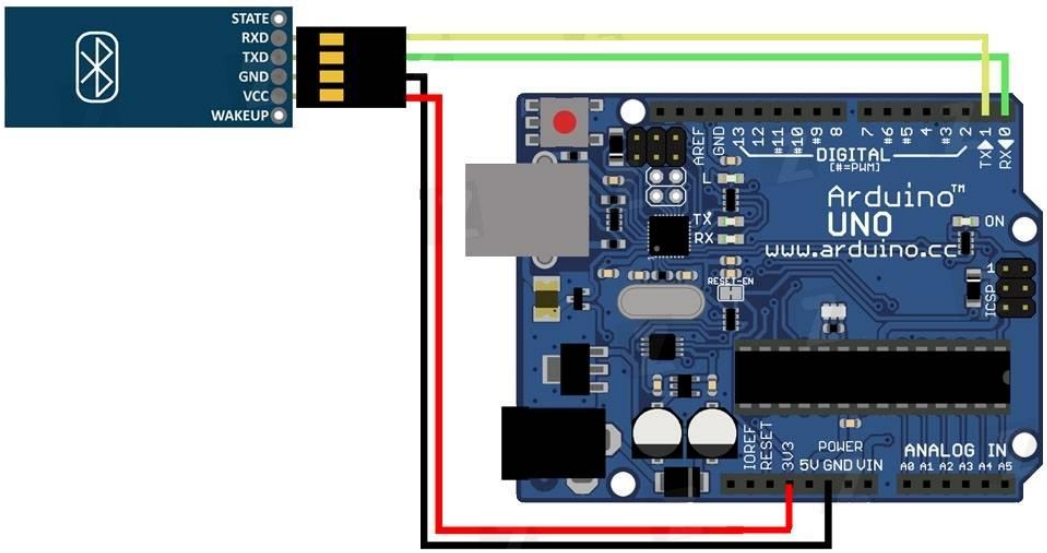


рис. 32 — Bluetooth HC-05(06), breakout и схема подключения.

Шаг 1. Создайте проект графического интерфейса.

Войдите в редактор RemoteXY. Установите в поле смартфона одну кнопку. Выделите эту кнопку, затем в правой панели во вкладке «Элемент» выберите свойство «Привязать к выводу» в значение 13(LED).

Шаг 2. Настройте конфигурацию проекта

В правой панели во вкладке «Конфигурация», выберите следующие настройки.

В правой панели во вкладке «Подключение модуля» установите следующие настройки. В настройках указывают, что модуль HC-05(06) подключается к Arduino через программный последовательный порт SoftwareSerial используя контакты 2 и 3 на скорости 9600. Это стандартная скорость передачи для модулей HC-05(06).

§ ПРАКТИКУМ: БЕСПРОВОДНАЯ КЛАВИАТУРА ДЛЯ КОМПЬЮТЕРА НА СМАРТФОНЕ.

Оборудование: персональный компьютер, смартфон с ОС Android версии 6.0 и выше, доступ к Internet.

Программное обеспечение: RemoteXY, IDE Arduino, IDE FLProg.

Цель: научиться реализовывать конечные проекты встраиваемых компьютерных систем посредством среды визуального программированию по техническому заданию.

Туториал и весь дидактический материал (в виду его большого объема) находится располагается по ссылке:

https://habr.com/ru/company/flprog/blog/400655/

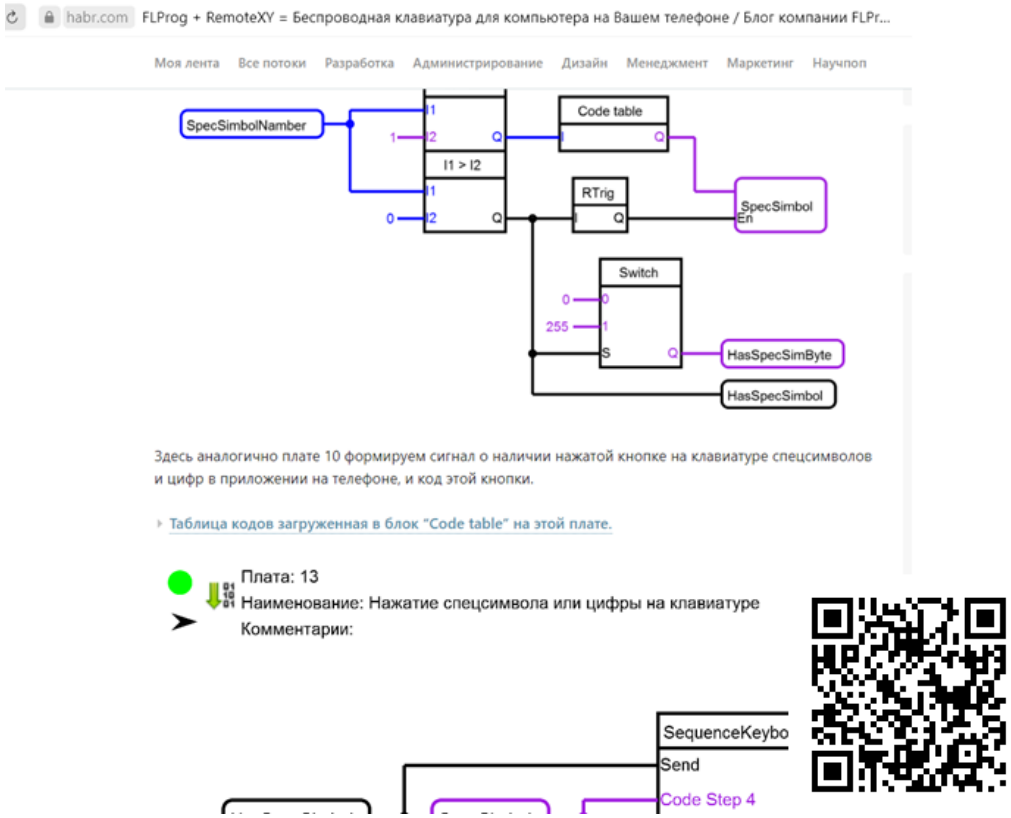


рис. 33 — Статья-туториал на портале Хабрахабр «FLProg + RemoteXY. Беспроводная клавиатура для компьютера»

В виду сложностей реализации проекта, связанных, прежде всего с временными рамками в учебном процессе, задание с реализацией беспроводного управления компьютером имеет демонстрационный характер (достаточно показать его реализацию и обсудить ее в рамках семинара).

На фактическое выполнение предлагается следующее задание. Целевое предназначение, которого направление на освоение последнего из рассматриваемых способов связи (взаимодействий), т.е, посредством IEEE 802.11:

ПРАКТИЧЕСКОЕ КОМПЛЕКСНОЕ ЗАДАНИЕ
«Интернет вещей с RemoteXY: конфигурации подключения.»
Ход работы: https://cxem.net/arduino/arduino213.php
Форма отчетности: скриншоты на всех этапах заданий

Первая часть задания: создать соединение со следующими параметрами:

- Соединение - Wi-Fi access point;
- Устройство – на ваш выбор (я буду использовать Mega);
- Модуль – ESP8266 Wi-Fi module;
- Среда программирования – ArduinoIDE;
- Интерфейс подключения – Hardware Serial;
- Порт – Serial (владельцы Mega могут попробовать любой другой доступный порт, я буду использовать Serial2);
- Скорость обмена – 115200 бод;
- Имя (SSID) – оставить по умолчанию;
- Пароль – оставить по умолчанию;
- Порт – оставить по умолчанию.
- Разместить на рабочем поле редактора кнопку, свойства кнопки не менять. Исходный код скомпилировать в ArduinoIDE и загрузить в контроллер.
- Далее необходимо подключить смартфон к созданной точке доступа, подключить программу RemoteXY к устройству. Если при нажатии на кнопку загорается светодиод на плате контроллера, значит всё сделано правильно.
- **Вторая часть задания:** изменить проект так, чтобы он использовал Software Serial, использовать выводы на свое усмотрение.
- **Третья часть задания:** изменить проект для подключения платы NodeMCU.

ОБЗОР SoC ESP8266/ESP32 как аппаратной основы IoT

ESP32 и ESP8266 — это недорогие микропроцессорные системы идеально подходящие для проектов в области интернета вещей (IoT) и домашней автоматизации.

ESP8266 — микроконтроллер китайского производителя Espressif Systems с интерфейсом Wi-Fi. Помимо Wi-Fi, микроконтроллер отличается отсутствием флеш-памяти в SoC, программы пользователя исполняются из внешней флеш-памяти с интерфейсом SPI.

Характеристики:

- 80 MHz 32-bit процессор Tensilica Xtensa L106.
- IEEE 802.11 b/g/n Wi-Fi. Поддерживается WEP и WPA/WPA2.
- 14-17 портов (варируется от модификации) ввода-вывода(из них возможно использовать 11), SPI, I²S, UART, 10-bit АЦП. I²C возможен только через bit-banging.
- Питание 2,2…3,6 В. Потребление до 215 мА в режиме передачи, 100 мА в режиме приема, 70 мА в режиме ожидания. Поддерживаются три режима пониженного потребления, все без сохранения соединения с точкой доступа: Modem sleep (15 мА), Light sleep (0.4 мА), Deep sleep (15 мкА).

Микроконтроллер не имеет на кристалле пользовательской энергонезависимой памяти. Исполнение программы ведется из внешней SPI ПЗУ путём динамической подгрузки требуемых участков программы в кэш инструкций.

Подгрузка идет аппаратно, прозрачно для программиста. Поддерживается до 16 МБ внешней памяти программ. Возможен Standard, Dual или Quad SPI интерфейс.

Производитель не предоставляет документации на внутреннюю периферию микроконтроллера. Вместо этого он дает набор библиотек, через API которых программист получает доступ к периферии.

Рис. 34 — ESP8266 NodeMCU

Рассмотрим одну из наиболее знаковых и распространенных модификаций ESP 8266 – ESP 12E. Данный модуль имеет 128 КБ ОЗУ и 4 МБ флеш-памяти (для хранения программ и данных), достаточных, чтобы справиться с большими строками, которые составляют веб-страницы, данными в JSON/XML и всем, что мы сегодня добавляем на устройства IoT.

ESP8266 содержит встроенный приемопередатчик Wi-Fi 802.11b/g/n HT40, поэтому он может не только подключаться к сети Wi-Fi и взаимодействовать с интернетом, но и устанавливать собственную сеть, позволяя другим устройствам подключаться напрямую к нему. Это делает ESP8266 NodeMCU еще более универсальным.

Требования к питанию:

Поскольку диапазон рабочего напряжения ESP8266 составляет **от 3 В до 3,6 В**, данная плата для поддержания постоянного напряжения на уровне 3,3 В поставляется с LDO стабилизатором напряжения. Он может надежно обеспечивать ток до 600 мА, чего должно быть более чем достаточно, поскольку ESP8266 во время радиочастотных передач потребляет до 80 мА.

- Рабочее напряжение: от 2,5 до 3,6 В
- Встроенный стабилизатор: 3,3 В, 600 мА
- Рабочий ток: 80 мА
- Потребление в спящем режиме: 20 мкА

Выход стабилизатора также выводится на выводы на сторонах платы и обозначен как 3V3. Эти выводы можно использовать для подачи питания на внешние компоненты.

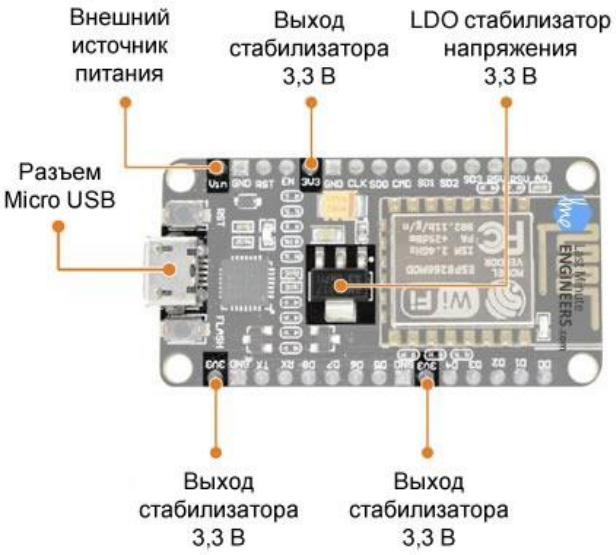


Рис. 35 — Элементы питания ESP8266 NodeMCU

Питание к ESP8266 NodeMCU подается через встроенный USB-разъем MicroB. В качестве альтернативы, если у вас есть стабилизированный источник напряжения 5 В, можно использовать вывод VIN для непосредственного питания ESP8266 и его периферии.

ESP8266 требует 3,3 В для питания и логические уровни 3,3 В для связи. Контакты GPIO не допускают напряжение 5 В! **Если вы хотите соединить плату со схемами 5 В (или выше), то необходимо реализовать согласование логических уровней.**

Периферия и ввод/вывод

ESP8266 NodeMCU имеет в общей сложности 17 выводов GPIO, выведенных на разъемы с обеих сторон отладочной платы. Эти выводы могут использоваться для выполнения различных периферийных задач, в том числе:

- вход АЦП – канал 10-разрядного АЦП;
- интерфейс UART – интерфейс UART используется для загрузки кода по последовательной связи;
- выходы ШИМ – выводы ШИМ могут использоваться для регулировки яркости светодиодов или управления двигателями;
- интерфейсы SPI, I2C – интерфейсы используются SPI и I2C для подключения всевозможных датчиков и периферийных устройств;

Мультиплексируемые выводы ввода/вывода

- 1 канал АЦП
- 2 интерфейса UART
- 4 выхода ШИМ
- Интерфейсы SPI, I2C и I2S

Рис. 36 – Мультиплексируемые выводы GPIO платы ESP8266 NodeMCU

В ESP8266 используется функция мультиплексирования выводов (несколько периферийных устройств мультиплексируются на один вывод GPIO). Это означает, что один вывод GPIO может действовать как PWM/UART/SPI.

Кнопки и светодиодный индикатор на плате:

На плате ESP8266 NodeMCU находятся две кнопки. Одна из них, помеченная как RST, расположенная в верхнем левом углу, представляет собой кнопку сброса, которая, конечно же, используется для сброса микросхемы ESP8266. Другая кнопка, FLASH, в левом нижнем углу – это кнопка загрузки, используемая при обновлении прошивки.

Кнопки и индикаторы

- RST – сброс чипа ESP8266
- FLASH – загрузка новой программы
- Синий светодиод - программируется пользователем

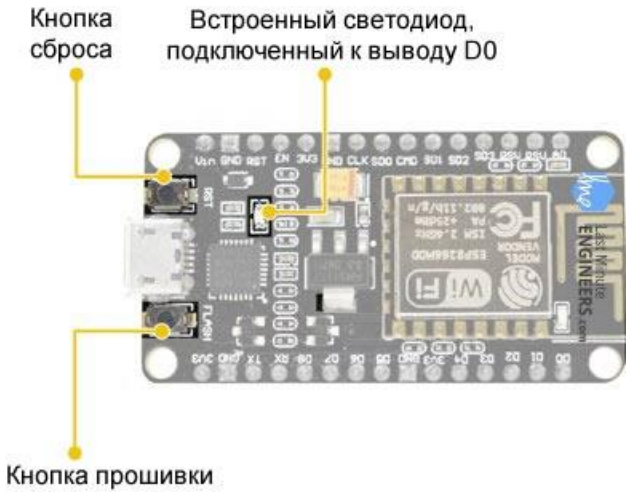


Рис. 37 – Кнопки и светодиоды на плате ESP8266 NodeMCU

На плате также имеется светодиодный индикатор, который программируется пользователем и подключен к выводу D0 платы.

Последовательная связь

На плате установлен контроллер USB-UART CP2102/или CH340 от Silicon Labs, который преобразует USB сигнал в сигнал последовательного порта и позволяет компьютеру программировать и взаимодействовать с микросхемой ESP8266.

- USB-UART преобразователь CP2102/или CH340
- Скорость связи 4,5 Мбит/с
- Поддержка управления потоком

Для простоты мы сгруппируем выводы с аналогичными функциями.

Выводы питания – на плате расположено четыре вывода питания, а именно: один вывод VIN и три вывода 3.3V. Если у вас есть стабилизированный источник напряжения 5 В, вывод VIN можно использовать для непосредственного питания ESP8266 и его периферии. Выводы 3.3V – это выходы встроенного стабилизатора напряжения. Эти выводы могут использоваться для подачи питания на внешние компоненты. GND – это вывод земли отладочной платы ESP8266 NodeMCU.

Выводы I2C используются для подключения всех видов датчиков и периферийных устройств на шине I2C в вашем проекте. Поддерживаются и I2C Master, и I2C Slave.

Работа интерфейса I2C может быть реализована программно, а тактовая частота составляет максимум 100 кГц. Следует отметить, что тактовая частота I2C должна быть выше самой низкой тактовой частоты из ведомых устройств.

Выводы GPIO. На ESP8266 NodeMCU имеется 17 выводов GPIO, которые можно назначать программно на различные функции, такие как I2C, I2S, UART, PWM, дистанционное инфракрасное управление, светодиодный индикатор и кнопка. Каждый включенный вывод GPIO может быть настроен либо на внутреннюю подтяжку к земле или к шине питания, либо установлен на высокоимпедансное состояние. При конфигурировании на вход для генерирования прерываний процессора он может быть настроен на срабатывание либо по фронту, либо по спаду.

Вывод ADC подает сигнал на имеющийся в NodeMCU, встроенный 10-разрядный прецизионный аналого-цифровой преобразователь последовательного приближения (SAR ADC). С помощью этого АЦП могут быть реализованы две функции: проверка напряжения питания на выводе VDD3P3 и проверка входного напряжения на выводе TOUT (но не одновременно).

Выводы UART ESP8266 NodeMCU имеет 2 интерфейса UART, то есть UART0 и UART1, которые обеспечивают асинхронную связь (RS232 и RS485) и могут обмениваться данными со скоростью до 4,5 Мбит/с. Для связи можно использовать UART0 (выводы TXD0, RXD0, RST0 и CTS0), который поддерживает управление потоком. UART1 (вывод TXD1) поддерживает только сигнал передачи данных, поэтому он обычно используется для печати журнала событий.

Выводы SPI ESP8266 имеет два интерфейса SPI (SPI и HSPI), поддерживающих и ведомый (slave), и ведущий (master) режимы. Эти интерфейсы SPI также поддерживают следующие функции SPI:

- 4 режима синхронизации передачи SPI;
- до 80 МГц и тактовые частоты, полученные делением 80 МГц;
- до 64 байт FIFO.

Вывод**ы SDIO ESP8266** имеет защищенный цифровой интерфейс ввода/вывода (SDIO, Secure Digital Input/Output Interface), который используется для прямого подключения карт SD. Поддерживаются 4-битный 25 МГц SDIO v1.1 и 4-битный 50 МГц SDIO v2.0.

Выводы PWM. На плате имеется 4 канала широтно-импульсной модуляции (PWM). Выход ШИМ может быть реализован программно, и использован для управления двигателями и светодиодами. Частотный диапазон ШИМ регулируется от 1000 мкс до 10000 мкс, то есть от 100 Гц до 1 кГц.

Выводы управления используются, как ни странно, для управления ESP8266. Эти выводы включают в себя вывод включения микросхемы EN, вывод сброса RST и вывод пробуждения WAKE.

Вывод EN – микросхема ESP8266 включена, когда на вывод EN подается высокий логический уровень. При низком логическом уровне микросхема работает на минимальной мощности.

Вывод RST используется для сброса микросхемы ESP8266.

Вывод WAKE используется для вывода чипа из глубокого сна.

"Обзор платы NodeMCU ESP8266 и ее использование в Arduino IDE"
Электронный портал: radioprog.ru

Сетевая инфраструктура

Типовое применение ESP8266 как аппаратной основы Internet of Things чаще всего подразумевает установку в домах или офисах. При этом сетевое подключение осуществляется к домашней/офисной локальной сети с выходом в интернет через роутер. Пользователь устройства может контролировать его с помощью планшета или компьютера через свою локальную сеть либо удаленно, через Интернет.

ESP8266 может работать как в роли точки доступа так и оконечной станции. При нормальной работе в локальной сети ESP8266 конфигурируется в режим оконечной станции. Для этого устройству необходимо задать SSID Wi-Fi сети и, в закрытых сетях, пароль доступа. Для первоначального конфигурирования этих параметров удобен режим точки доступа. В режиме точки доступа устройство видно при стандартном поиске сетей в планшетах и компьютерах.

Остается подключиться к устройству, открыть HTML страничку конфигурирования и задать сетевые параметры. После чего устройство штатно подключится к локальной сети в режиме оконечной станции.

В случае исключительно местного использования возможно всегда оставлять устройство в режиме точки доступа, что снижает необходимые усилия пользователя по его настройке.

После подключения к Wi-Fi сети устройство должно получить IP-параметры локальной сети. Эти параметры можно задать вручную вместе с параметрами Wi-Fi либо активизировать какие-либо сервисы автоматического конфигурирования IP-параметров (например, DHCP).

После настройки IP параметров обращение к серверу устройства в локальной сети обычно осуществляется по его IP адресу, сетевому имени (в случае если имена поддержаны какой-либо технологией, например, NBNS) или сервису (в случае если поддержан автоматический поиск сервисов, например через протокол SSDP).

Зачастую доступ к устройству требуется из Интернета.

Например, пользователь с мобильного телефона удаленно проверяет состояние своего «умного дома», обращаясь напрямую к устройству. В этом случае устройство работает в режиме сервера, к которому обращается внешний клиент.

Как правило, устройство на основе ESP8266 находится в локальной сети офиса или дома. Выход в Интернет обеспечивает роутер, подключенный с одной стороны к локальной сети, а с другой, к сети провайдера интернета. Провайдер назначает роутеру свой статический или динамический IP адрес и роутер осуществляет трансляцию адресов локальной сети в сеть провайдера. По умолчанию правила этой трансляции обеспечивают свободную видимость интернет-адресов из локальной сети, но не позволяют обратиться к локальным адресам со стороны Интернета. Есть несколько способов обойти это ограничение.

Конфигурирование NAT

Большинство современных роутеров позволяют задать дополнительные правила трансляции сетевых адресов между локальной и глобальной сетями. Как правило для этого используются технологии Virtual server или DMZ. Обе технологии позволяют обратиться к серверу в локальной сети из глобальной сети, зная лишь IP адрес, выданный роутеру провайдером.

В случае статического IP адреса роутера – это, зачастую, может быть удовлетворительным решением для ограниченного круга пользователей системы. Однако такой подход не всегда удобен: необходимо вручную конфигурировать роутер и выяснять IP-адрес роутера, который может регулярно меняться. Относительно легко решить проблему неизвестного IP адреса можно с помощью механизма DDNS.

DDNS

Чтобы обратиться к серверу устройства конечный пользователь должен знать IP адрес, по которому находится устройство. Однако получить у провайдера Интернета для устройства статический IP адрес не всегда возможно, да и пользоваться таким адресом неудобно. Для решения этой проблемы были созданы специальные интернет-сервисы под общим наименованием динамический DNS. Эти сервисы работают как специальные серверы с фиксированными именами в интернете. Разработчик заводит на таком сервисе свой аккаунт с уникальным именем. Параметры этого аккаунта он прописывает в устройстве. Устройство в режиме клиента периодически обращается к серверу сервиса, сообщая ему имя своего аккаунта и свой текущий IP адрес. Конечный пользователь в интернете обращается к этому же сервису и получает от него текущие IP параметры устройства.

В таком случае устройство в сети видно с доменным именем третьего уровня, например, esp8266.ddns.org. Существует множество DDNS сервисов. Основная проблема DDNS сервисов это гарантии существования конкретного сервиса. Как правило, гарантируется только коммерческий сервис, когда за его использование взимается плата.

Внешние IoT сервисы для ESP8266

Чтобы облегчить проблему доступности устройства в Интернете и сделать инсталляцию устройства легкой для пользователя были разработан ряд решений. Механизм этих решений базируется на существовании в Интернете специального сервера, к которому может подключиться как IoT устройство, так и планшет/компьютер пользователя.

При этом устройство работает в режиме клиента, никаких специальных настроек роутера или особых навыков от инсталлятора и пользователя устройства не требуется. Обмен данными с устройством осуществляется при посредничестве этого специального сервиса, параметры которого в устройство должен заложить разработчик.

Распространение использования таких сервисов сдерживается необходимостью длительно поддерживать свой сервис в Интернете или пользоваться чужими сервисами с непонятными перспективами длительного существования бесплатных возможностей или регулярной оплатой коммерческих вариантов.

Internet of Things

Основное применение ESP8266 находит в управлении разнообразными бытовыми приборами через беспроводные сети. Концепцию такого управления часто называют «Internet of Things» (IoT, «интернет вещей»). **Верхний уровень IoT** представлен разнообразными приложениями под популярные платформы (Android, iOS, Windows, …). Эти приложения позволяют разработчику прибора адаптировать приложение под управление его прибором и передать пользователю готовое решение. Существует несколько популярных реализаций концепции IoT в плане обмена данными по сети:

HTML сервер на ESP8266: Контроль и управление устройством ведется через браузер. Тяжеловесное решение, подходит автономным устройствам автоматики.

AllJoyn — набирающий популярность открытый IoT протокол крупного альянса производителей цифровой техники «Allseen». Поддержка встроена в Windows 10. На русском можно почитать здесь.

HTTP запросы с использованием протоколов типа REST, XML-RPC (SOAP). Для этого на ESP8266 запускают упрощенный HTTP сервер, без HTML. Достоинство метода — отсутствие проблем с настройкой файрволлов, HTTP обычно открыт всегда.

MQTT. Это простой протокол поверх TCP/IP. Очень популярное решение. Существует большое количество IoT приложений верхнего уровня для Android, iOS и других платформ, поддерживающих этот протокол.

SNMP. Расширяемый протокол управления сетевыми устройствами. Основной недостаток в том, что в большинстве сетей файрволлы блокируют прохождение SNMP.

ModBus и другие протоколы промышленной автоматизации.

Интересные проекты ПО **верхнего уровня** с решениями на базе ESP8266:

Majordomo — русскоязычный открытый проект домашней автоматизации.

Blynk — облачная платформа для IoT, которая имеет приложения для iOS и Android и поддерживает управление микроконтроллерами ESP8266, Arduino, Raspberry Pi, SparkFun и д.р. через Интернет.

SUPLA — открытый проект систем автоматизации зданий, использующий ESP8266.

В рамках данного курса мы будем использовать плату **Witty Cloud (ESP8266)** на базе модуля ESP-12F.

Для работы Witty Cloud не нужен внешний микроконтроллер или другое управляющее устройство, так как помимо Wi-Fi модуля в ESP-12F уже встроен 32-битный микроконтроллер с тактовой частотой 80 МГц, а также чип флеш-памяти на 4МБ.

Модуль Witty Cloud - это отличное решения для использования в "интернете вещей", системах удаленного мониторинга или управления и т.д. Плата предлагает несколько вариантов работы с Wi-Fi сетями, в том числе может выступать и как клиент Wi-Fi сети, и сам создавать точку доступа.

Использование Witty Cloud **вместо "голого" ESP8266 модуля существенно упрощает работу с платформой**, так как в него уже встроены USB-UART конвертер CH340, стабилизатор напряжения, припаяна PLS-планка со стандартным шагом 2.54 мм, а также, с программой стороны - в чип ESP8266 залита прошивка со встроенным интерпретатором Lua.

Witty Cloud состоит из двух отдельных плат. На верхней расположен сам Wi-Fi модуль ESP8266, стабилизатор на 3.3В, RGB светодиод, фоторезистор и программируемая кнопка. На нижней - USB-UART конвертер на чипе CH340 с обвязкой.

С помощью программатора (нижней платы) можно прошивать и другие платы из серии ESP-12, установив их на совместимый по распиновке адаптер, и припаяв на него стабилизатор.

Технические характеристики: ESP32 против ESP8266

ESP32 является преемником **ESP8266**. Он имеет дополнительное ядро процессора, более быстрый Wi-Fi, больше GPIO и поддерживает Bluetooth 4.2 и Bluetooth с низким энергопотреблением. Кроме того, ESP32 поставляется с сенсорными контактами, которые можно использовать для пробуждения ESP32 из глубокого сна, встроенным датчиком эффекта Холла и встроенным датчиком температуры (последние версии ESP32 больше не поставляются со встроенным датчиком температуры).

Обе платы очень дешевы, но ESP32 стоит немного дороже. В то время как ESP32 может стоить от 6 до 12 долларов, ESP8266 может стоить от 4 до 6 долларов (но это зависит от того, где вы их приобретаете).

В следующей таблице показаны основные различия между чипами ESP8266 и ESP32:

Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	-40℃ ~ 125℃	-40℃ ~ 125℃

Рис. 38 – Основные различия между чипами ESP8266 и ESP32

Использовать просто чипы ESP32 или ESP8266 сложно и непрактично, особенно при тестировании и создании прототипов. В большинстве случаев вы захотите использовать платы разработки ESP32 и ESP8266.

Эти платы поставляются со всеми необходимыми схемами для питания чипа, подключения его к компьютеру, схемой для легкой загрузки кода, контактами для подключения периферийных устройств, встроенными светодиодами питания и управления и другими полезными функциями.

Платы для разработки ESP32 и ESP8266, которые используют чаще всего — это плата разработки ESP32 DEVKIT DOIT и комплект ESP8266 ESP-12E NodeMCU Kit. Однако есть много других моделей плат для разработки, из которых вы можете выбирать.

В систему интегрирован радиочастотный тракт: балун (симметрирующий трансформатор), встроенные антенные коммутаторы, радиочастотные компоненты, малошумящий усилитель, усилитель мощности, фильтры и модули управления питанием.

ESP32 создан и разработан компанией Espressif Systems, китайской компанией, расположенной в Шанхае, а производится компанией TSMC по техпроцессу 40 нм.

ESP32 имеет большее количество GPIO чем ESP8266. Вы можете сами решать, какими выводами будут UART, I2C, SPI – для этого вам просто нужно прописать их в коде. Это возможно благодаря функции мультиплексирования микросхемы ESP32, которая позволяет назначать несколько функций одному и тому же выводу.

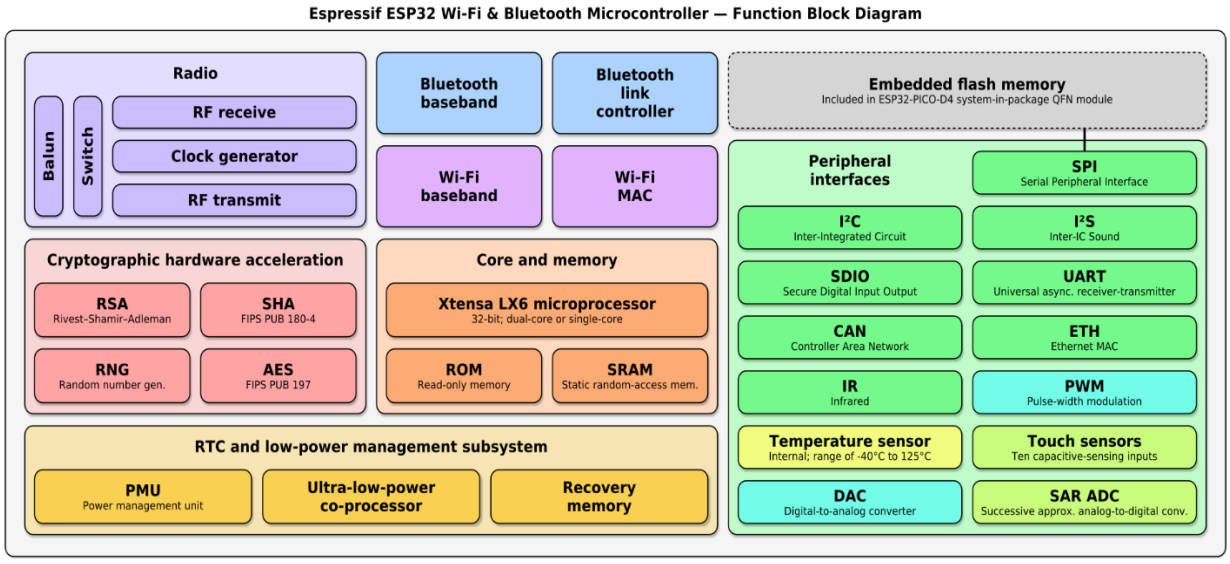


Рис. 39 – Функциональная блок-схема ESP32

§ ПРАКТИКУМ: СОЗДАНИЕ WEB-ИНТЕРФЕЙСА НАСТРОЙКИ ACCESS POINT ESP8266 В FLProg.

Оборудование: персональный компьютер, кабель AM/microBM 5p Cablexpert Pro (CCP-mUSB2-AMBM-1.0M или аналоги), микропроцессорная система ESP8266 (Witty Cloud ESP12F).

Программное обеспечение: IDE FLProg, IDE Arduino.

Цель: научиться реализовывать конечные проекты встраиваемых компьютерных систем посредством среды визуального программированию, овладеть навыками визуального программирования и конфигурирования микропроцессорных систем посредством Wi-fi.

Ход работы:

Работа над проектом начинается с выбора контроллера.

ESP8266 -> Boards -> ESP8266 NodeMCU v3

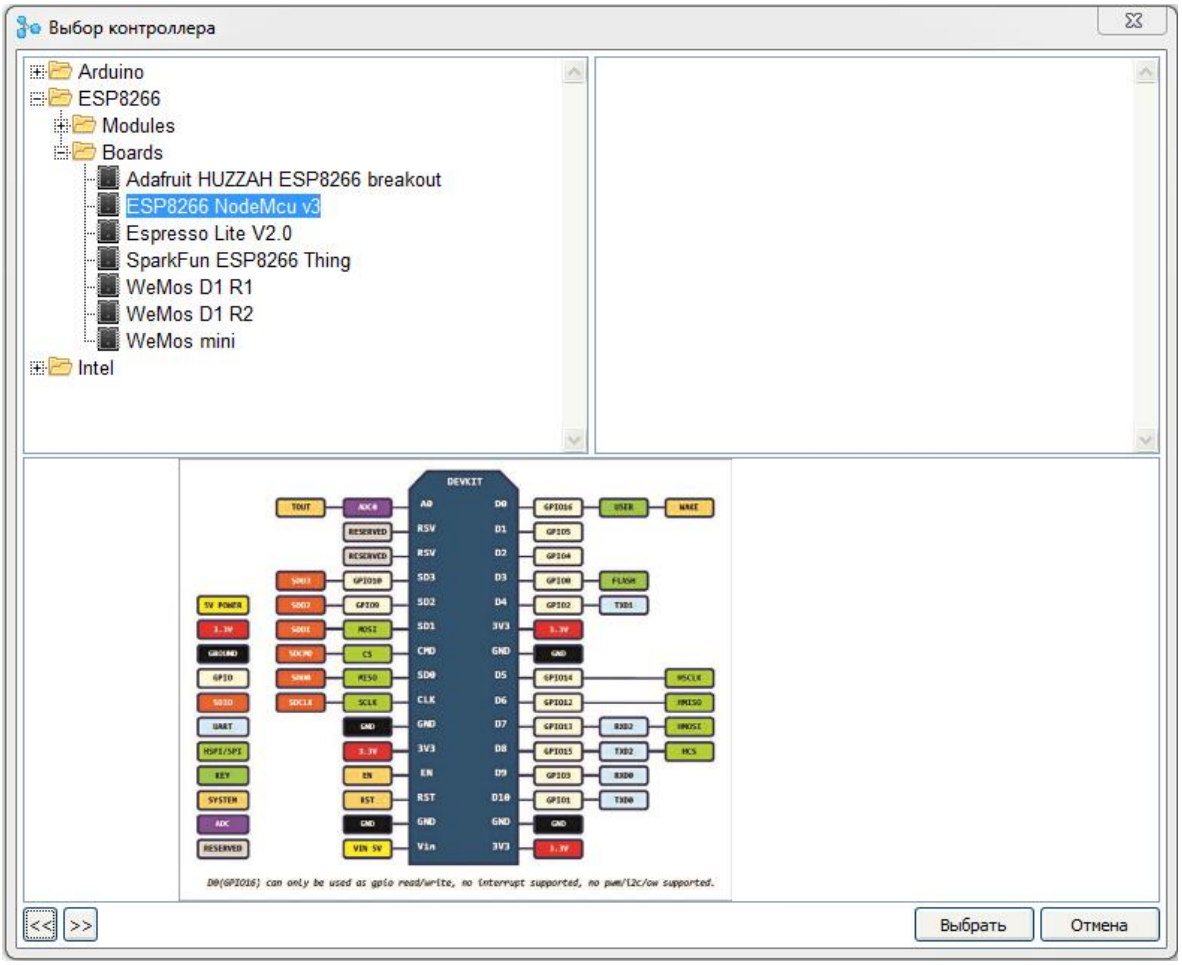


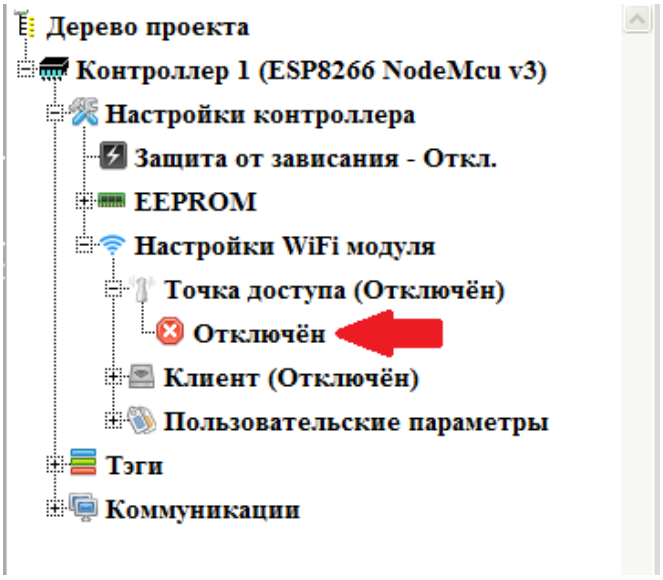
Рис. 40 – плата Witty Cloud является копией ESP8266 NodeMCU v3

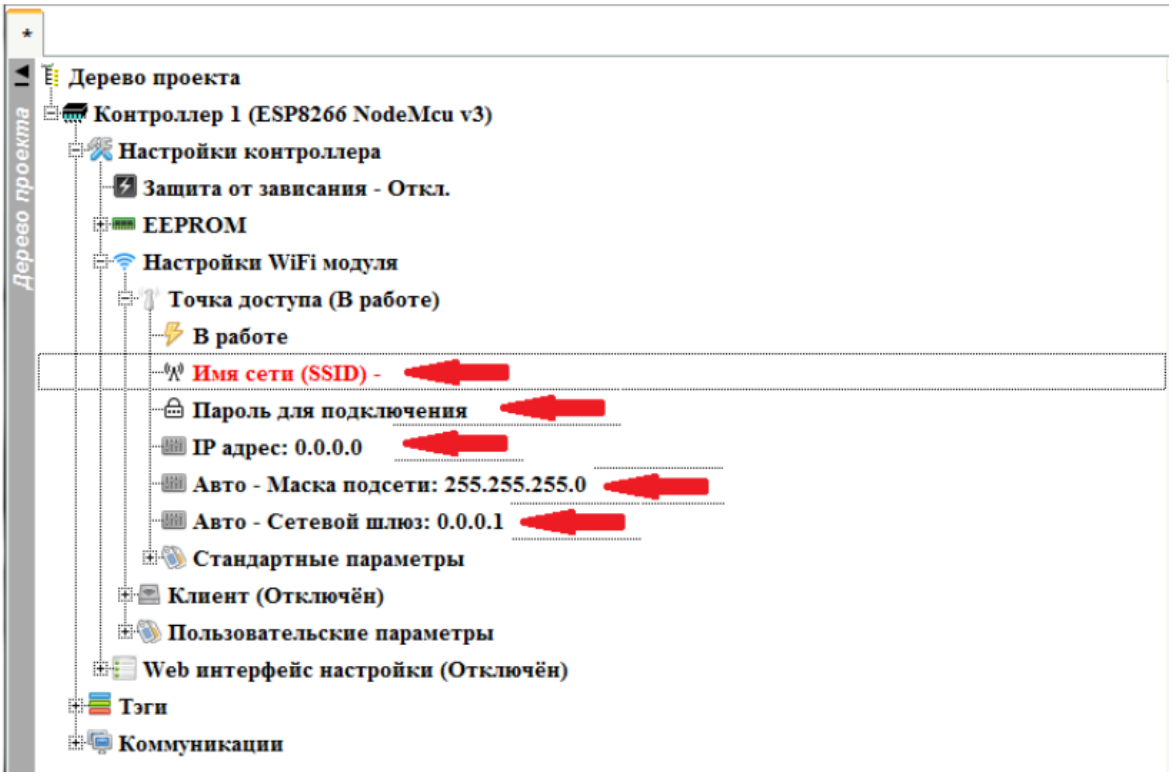
При выборе конкретного контроллера или платы можно посмотреть его изображение, распиновку, а также технические характеристики.

Основная работа при создании web-интерфейса настройки производится в дереве проекта.

Для начала настроим точку доступа. Раскрываем дерево проекта до пункта «Точка доступа» включительно и двойным кликом по ветке «Отключён» включаем точку доступа в работу.

В открывшихся ветках настраиваем параметры точки доступа. Для изменения необходимого параметра производим двойной клик на соответствующей ветке.





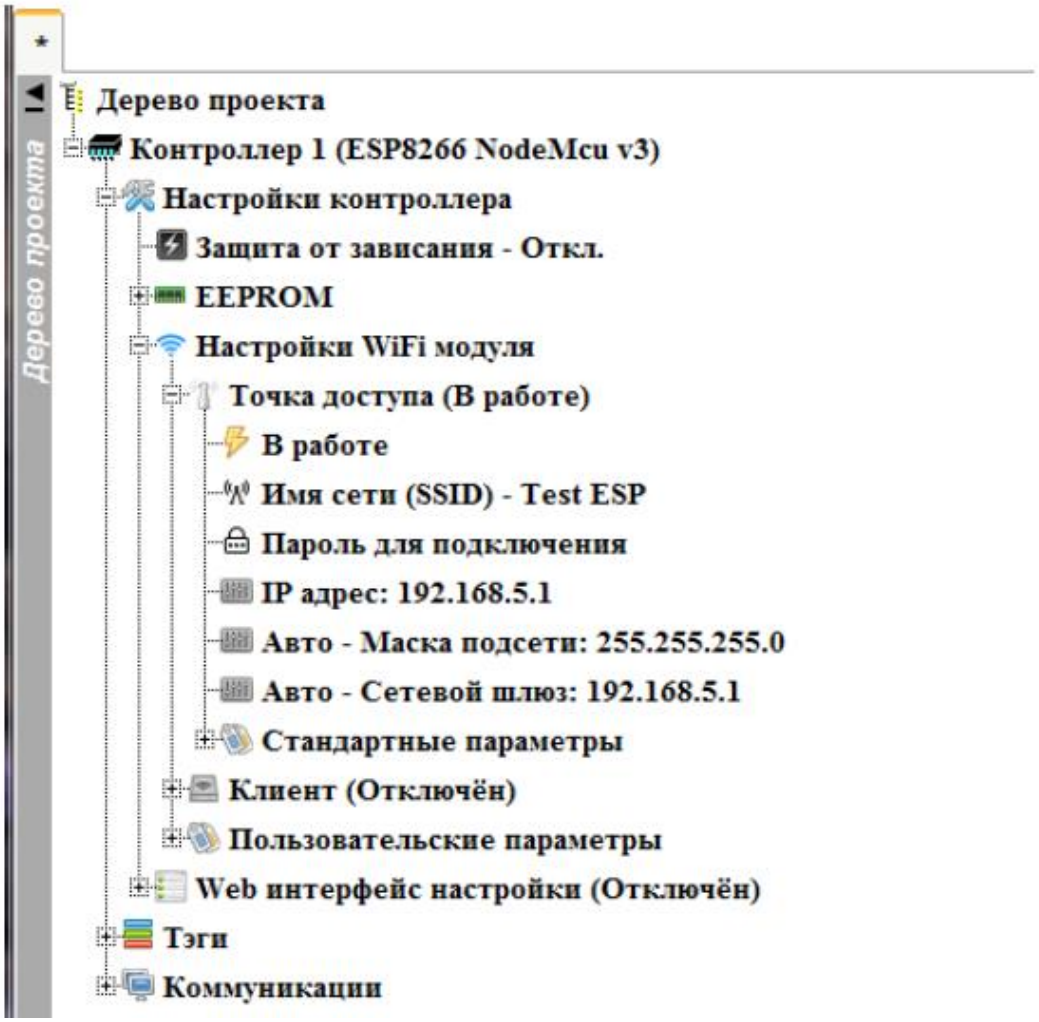
Имя сети (SSID) – Имя сети которую будет организовывать точка доступа.

Пароль для подключения – пароль для подключения к точке доступа. Если оставить пустым, то точка доступа будет без пароля со свободным подключением.

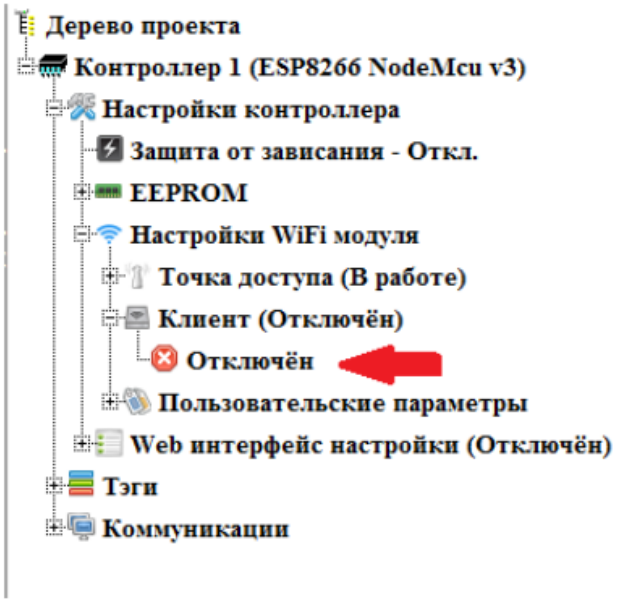
IP адрес – IP адрес который будет иметь контроллер в сети созданным точкой доступа. По этому адресу потом можно будет подключиться к контроллеру.

Остальные параметры (Маска подсети и Сетевой шлюз) заполнятся автоматически после установки IP адреса, но при необходимости их можно изменить, если требуются нестандартные значения.

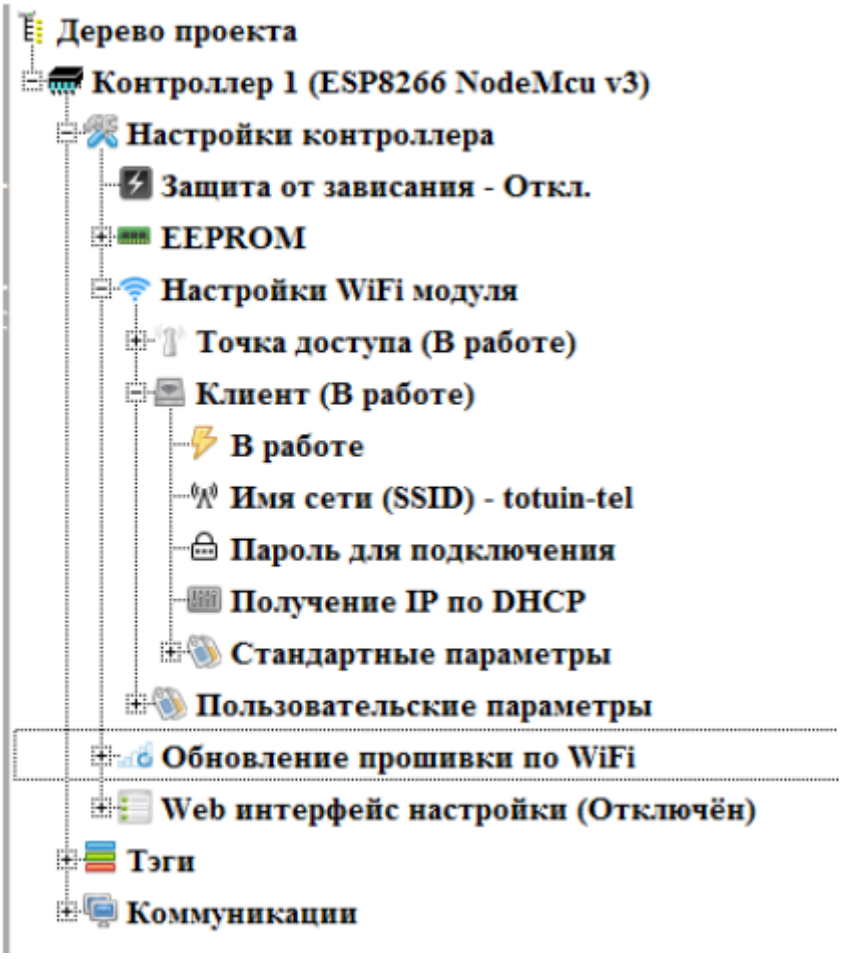
В результате должно получиться, что-то схожее:



С точкой доступа закончили, можем свернуть этот узел, и переходим к клиенту. Так же разворачиваем его узел, и включаем его в работу двойным кликом по ветке «Отключён».



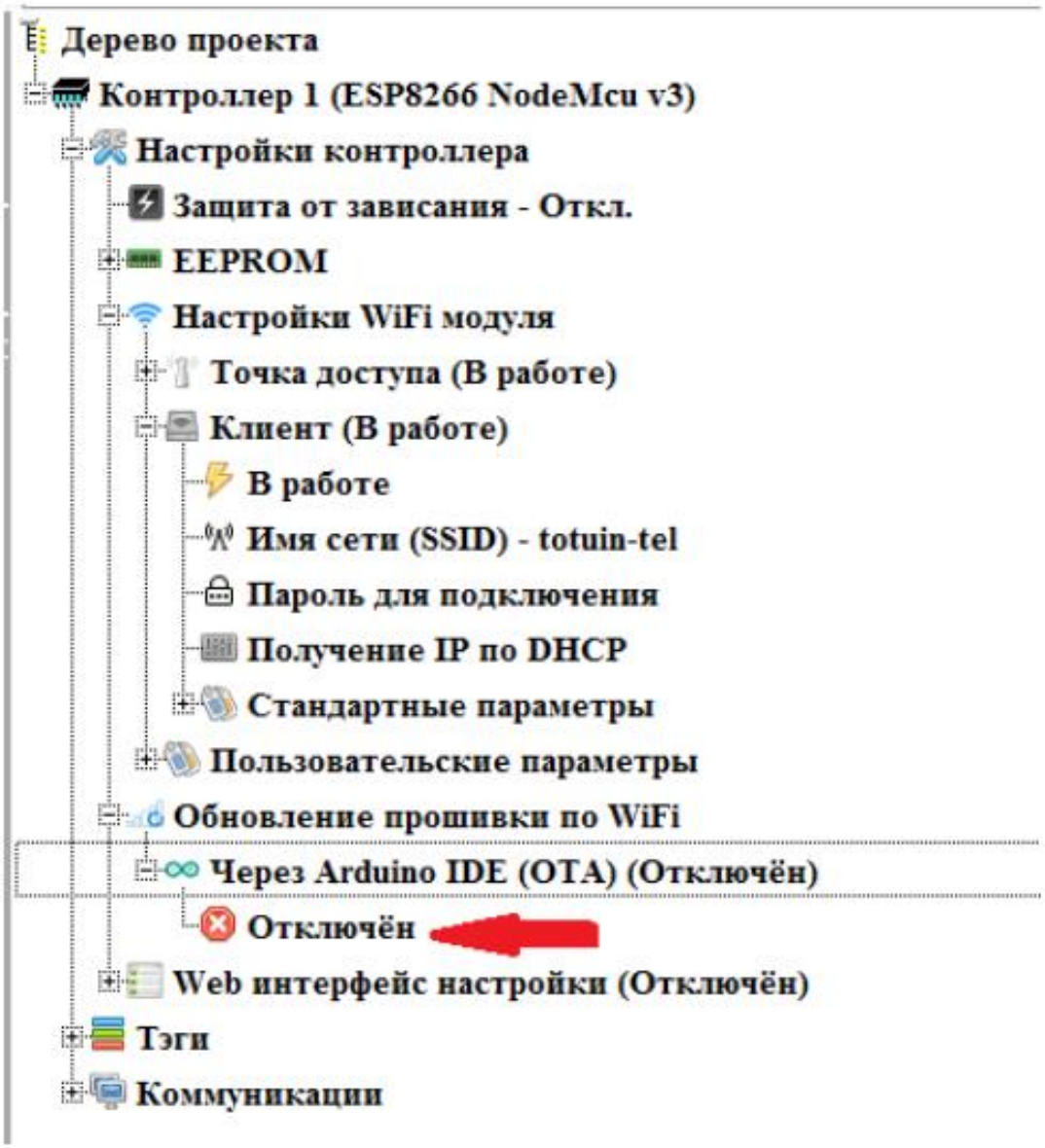
Настраиваем клиента. Возможны два варианта настройки клиента. Непосредственное задание настроек сети, и получение настроек по DHCP. Для начала используем второй вариант.



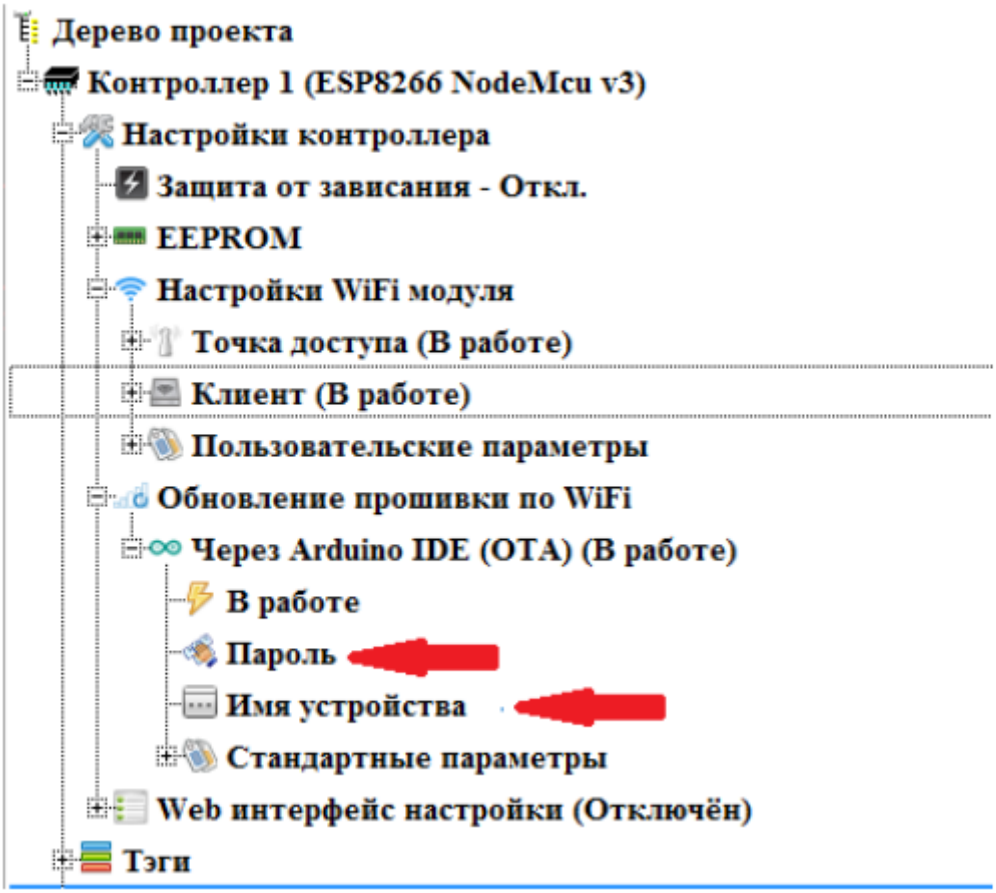
Обратите внимание, что в целях безопасности пароль подключения в дереве проекта не показывается.

С настройкой Wi-Fi интерфейсов закончили. Сворачиваем (если нужно) узел клиента и переходим к настройке режима обновления прошивки по Wi-Fi (если это требуется). Данный узел появляется только при включённом в работу клиенте.

Включаем этот режим двойным кликом по ветке “**Отключён**”



Задаём необходимые параметры (изменение значения параметров производится с помощью двойного клика на соответствующей ветке).

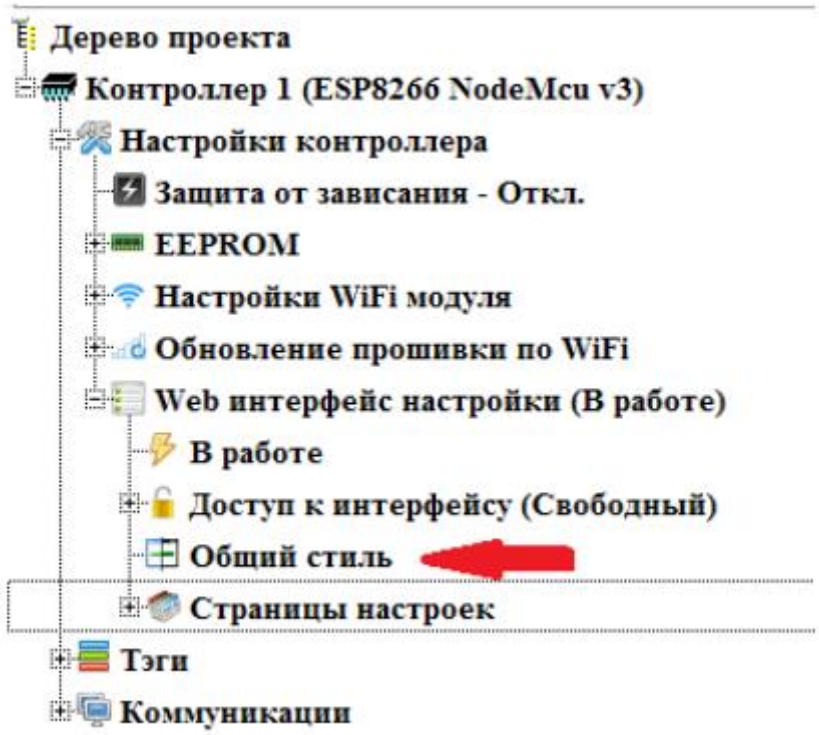


Пароль – при задании пароля перед заливкой новой прошивки потребуется его ввод.

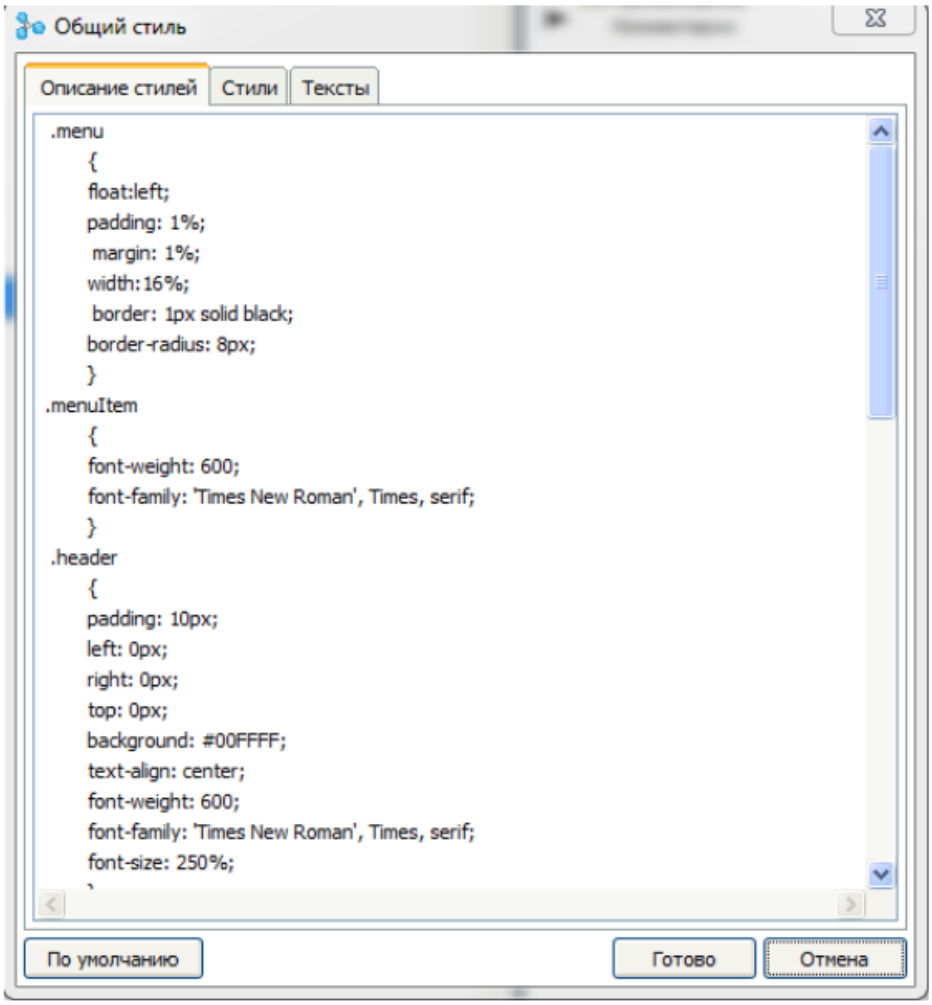
Имя устройства – Это имя будет фигурировать в названии порта соединения в Arduino IDE.

Теперь переходим непосредственно к созданию web интерфейса настройки. Открываем узел “Web интерфейс настройки” и включаем его двойным кликом по ветке “Отключён”.

Web интерфейс настроек представляет собой набор страниц с параметрами. Если страниц более одной, автоматически формируется меню для доступа к ним. Для каждой страницы можно задать собственные стили CSS, если использовать общие стили для всего Web интерфейса настройки.
Для настройки общих стилей CSS для всего web интерфейса совершаем двойной клик по ветке «Общий стиль».

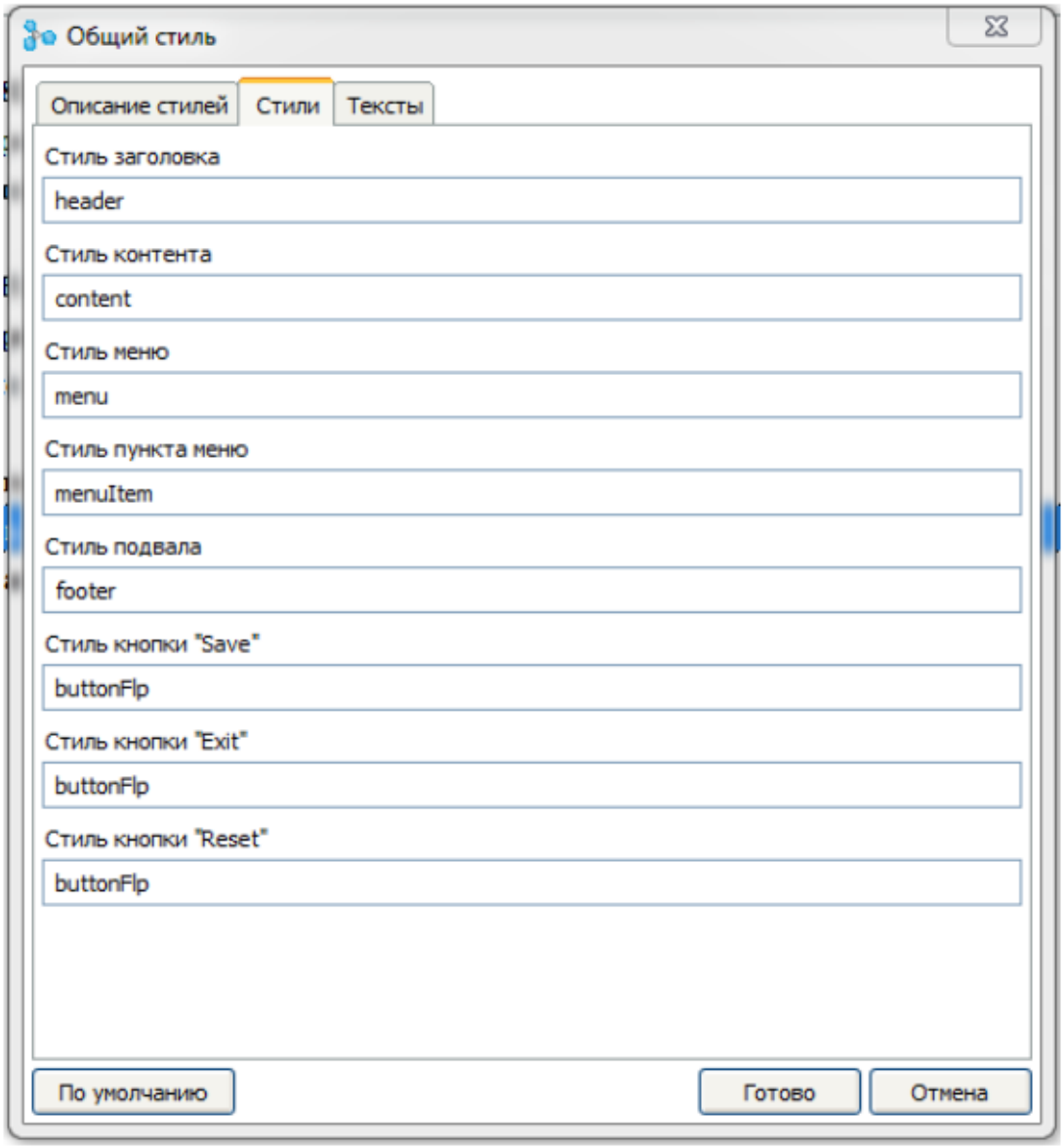


Откроется окно настроек общего стиля.

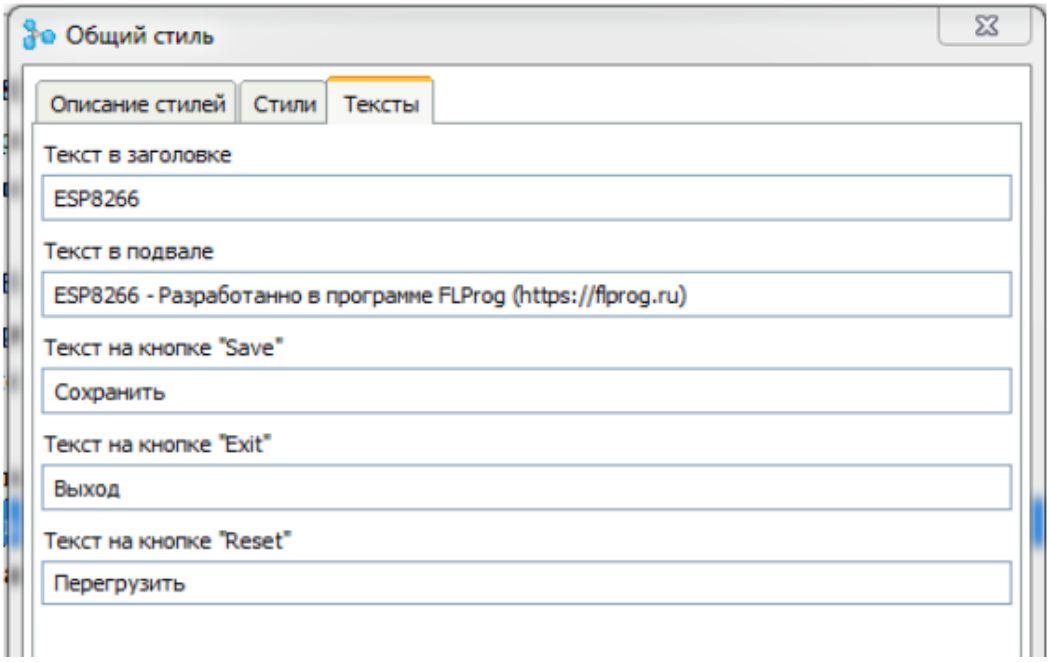


На закладке «Описание стилей» находится поле ввода непосредственно описания стилей применяемых для всех страниц настройки. По умолчанию это поле уже заполнено стилями для создания стандартного интерфейса. Но если есть желание изменить дизайн страниц, то можно их изменить.

На закладке «Стили» можно задать названия стилей, используемых для конкретных элементов страницы.

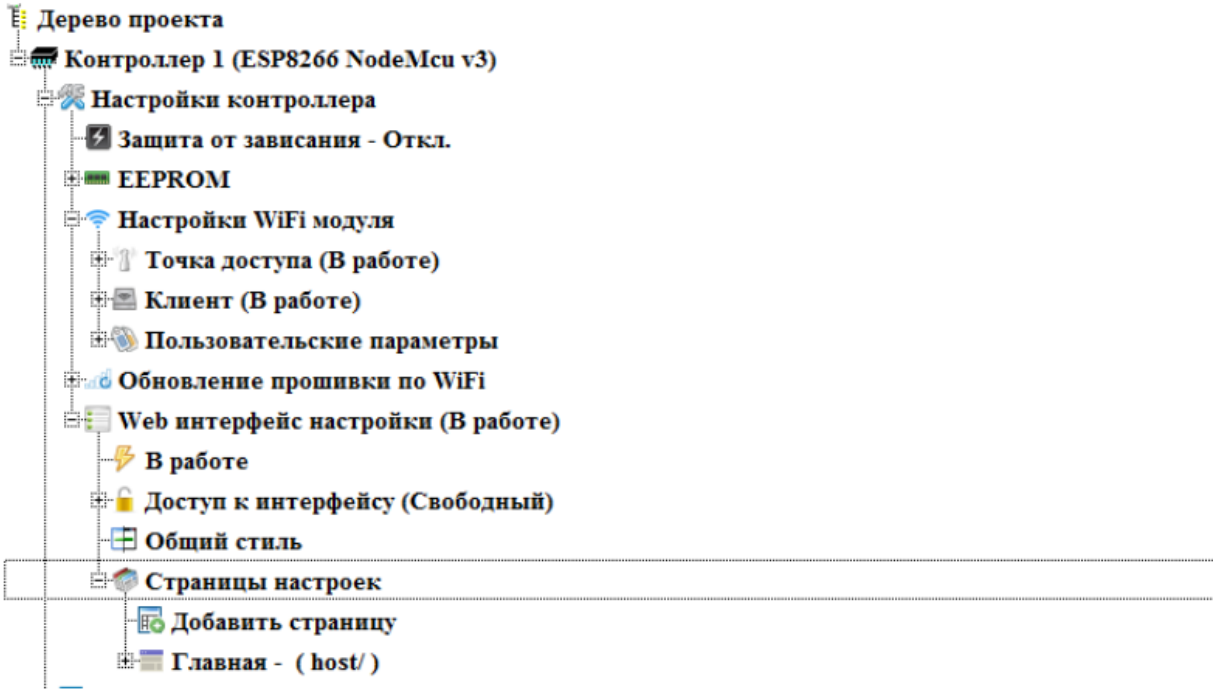


Эта закладка так же заполнена по умолчанию. На закладке «Тексты» можно ввести тексты основных элементов, используемых на странице.

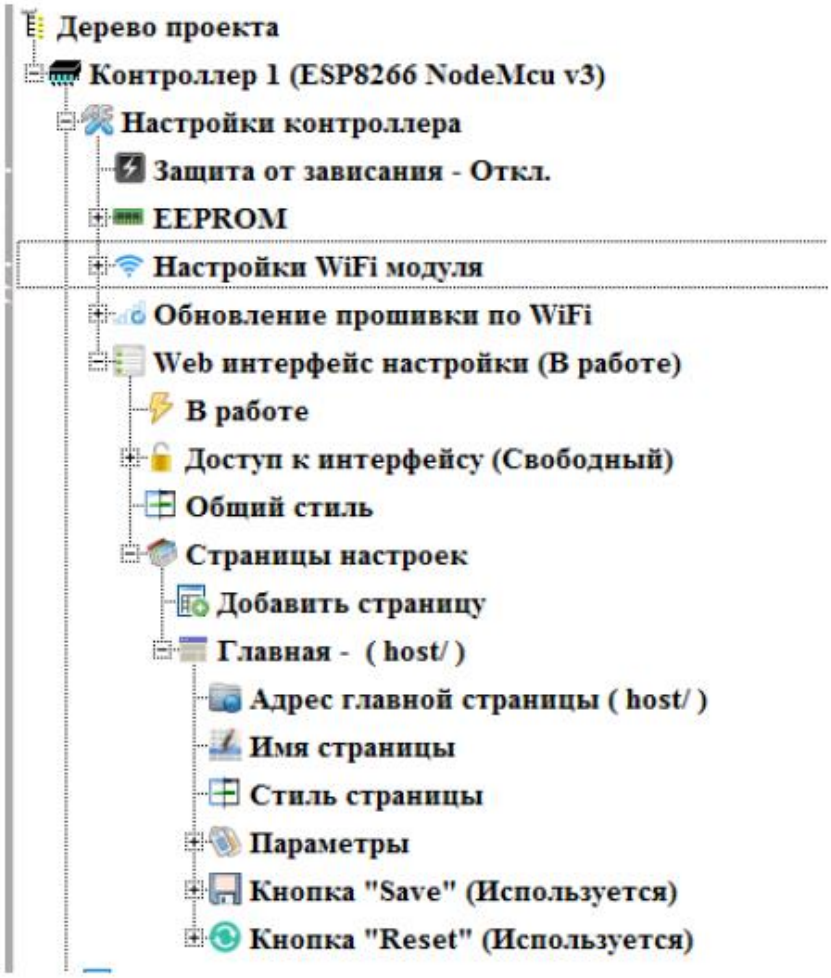


Для восстановления всех значений данного диалога значениями по умолчанию, можно воспользоваться кнопкой «По умолчанию». Настройки стилей и текстов, заданные в общих стилей, применяются на всех страницах настроек, если они не перекрыты настройками стилей конкретной страницей (это рассмотрим ниже)

Страницы показаны в узле «Страницы настроек». Раскрываем его:



По умолчанию всегда присутствует одна страница. Раскрыв её узел, получаем доступ к её настройкам.



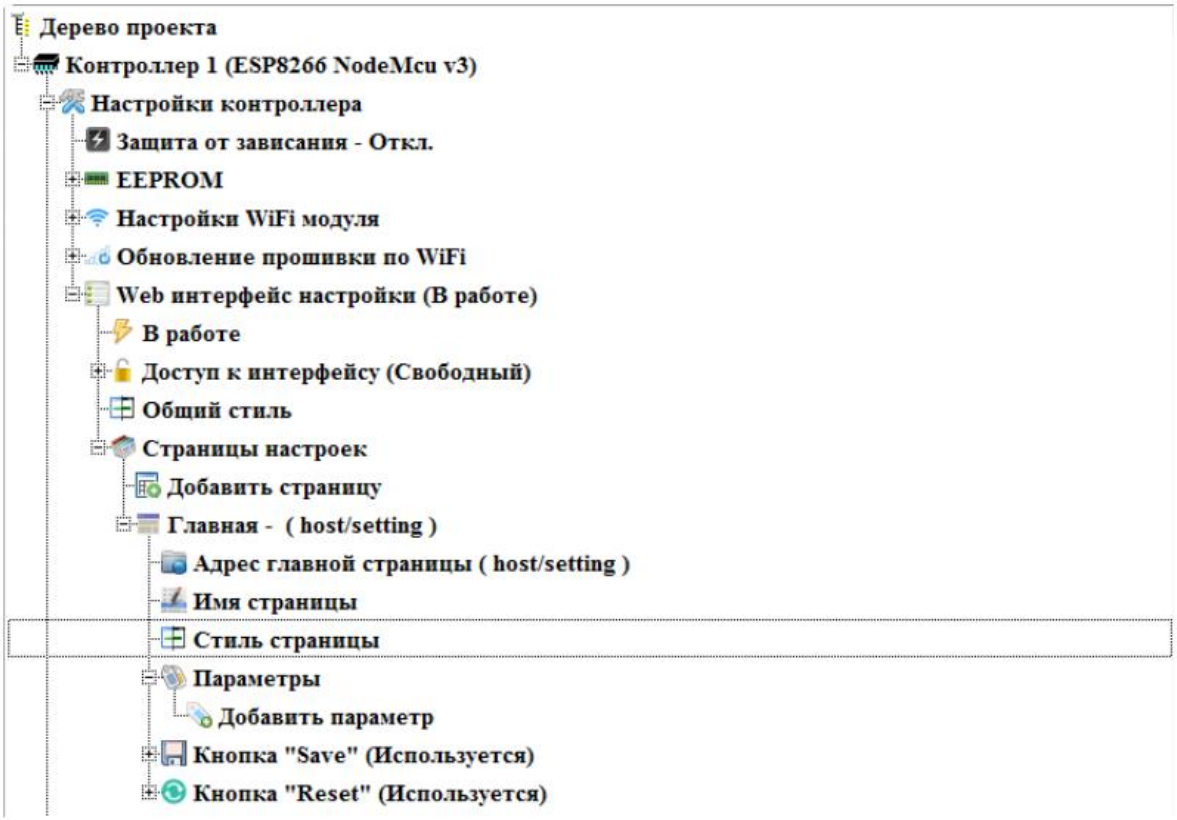
Адрес главной страницы – адрес основной страницы настроек. По умолчанию – host – то есть адрес контроллера в сети. При необходимости можно сменить. Сменим его на адрес host/setting (двойной клик по данной ветке).

Имя страницы – название страницы в меню. Оставим ей название – "Главная".

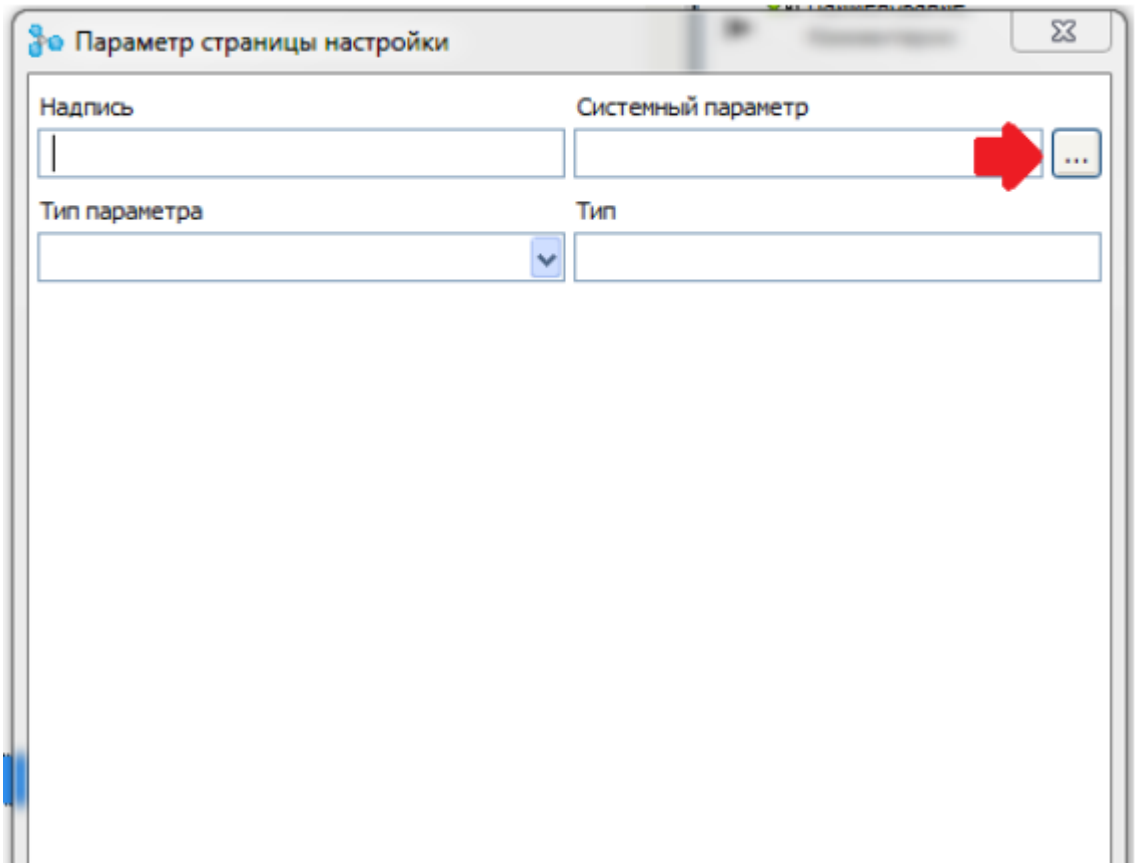
Стиль страницы — двойной клик по данной ветке вызывает диалог настройки стиля для конкретно этой страницы.

В этом диалоге можно дописать дополнительные стили CSS для данной страницы и назначить стили и тексты для элементов дизайна. Так же можно переопреописать стили, описанные в диалоге общих стилей.

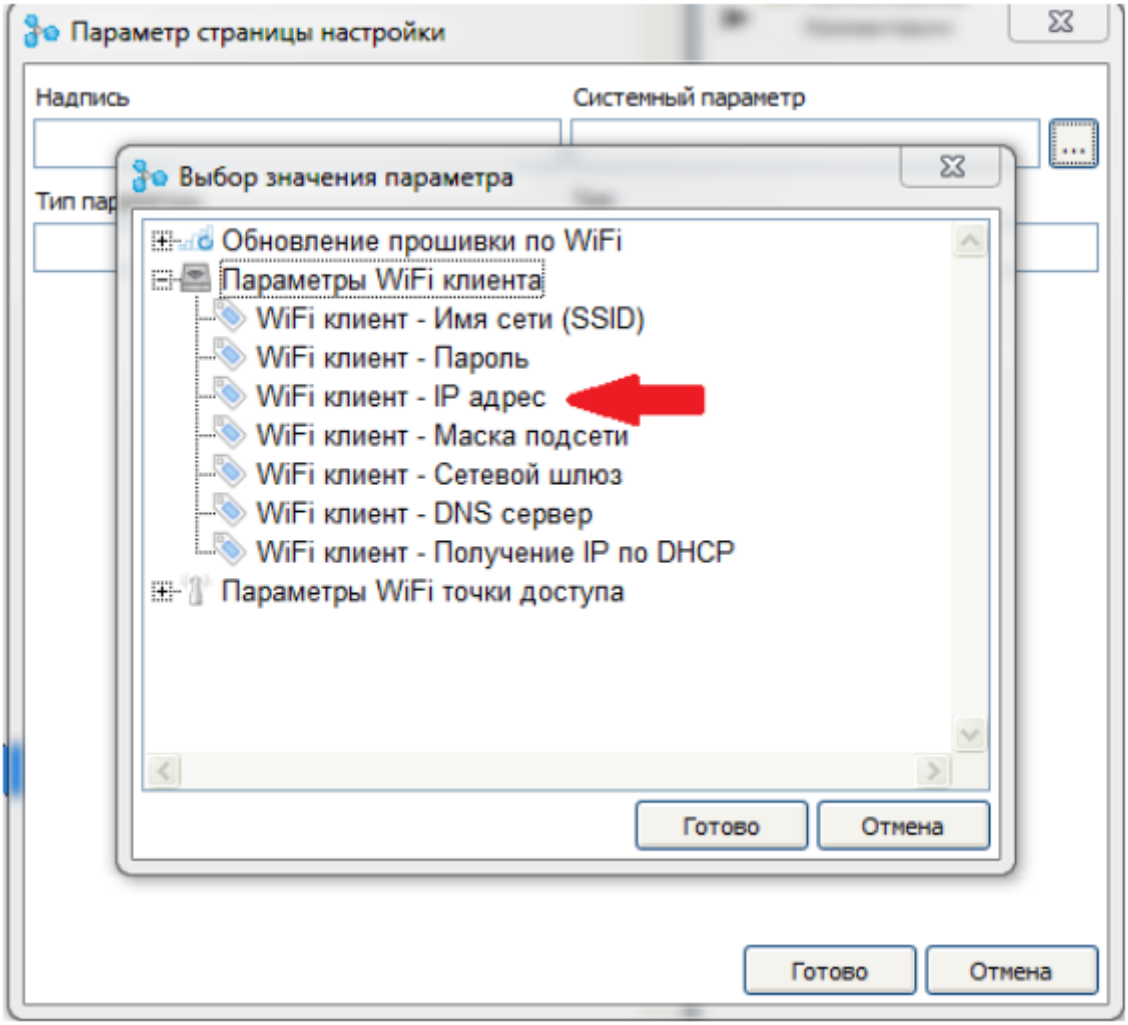
В узле “**Параметры**” задаются параметры, отображаемые на странице. На главной странице мы зададим отображение IP адреса полученного от роутера по DHCP в виде простого текста. Параметр добавляется с помощью двойного клика по ветке «Добавить параметр».



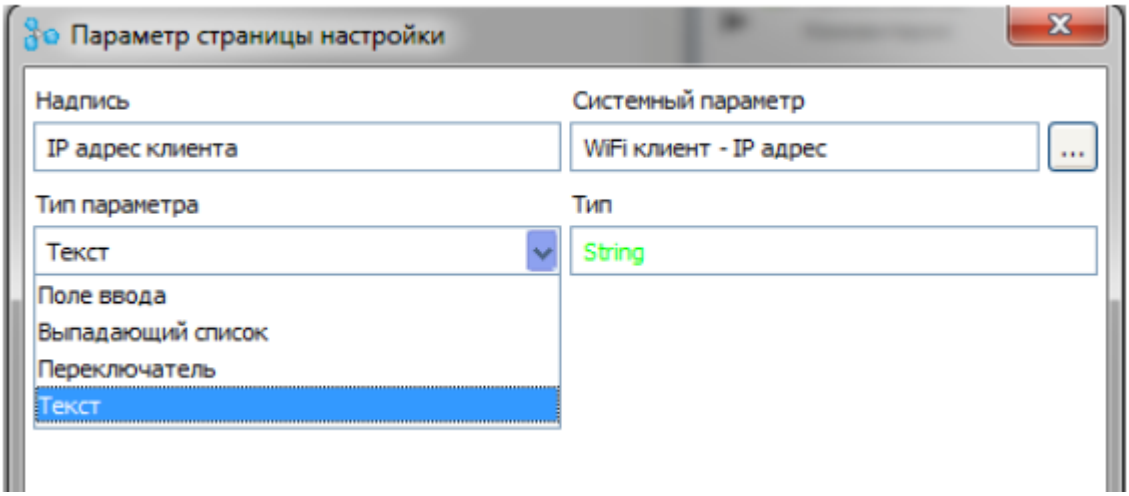
Открывается диалог создания нового параметра. В нём для начала нажимаем кнопку выбора системного параметра.



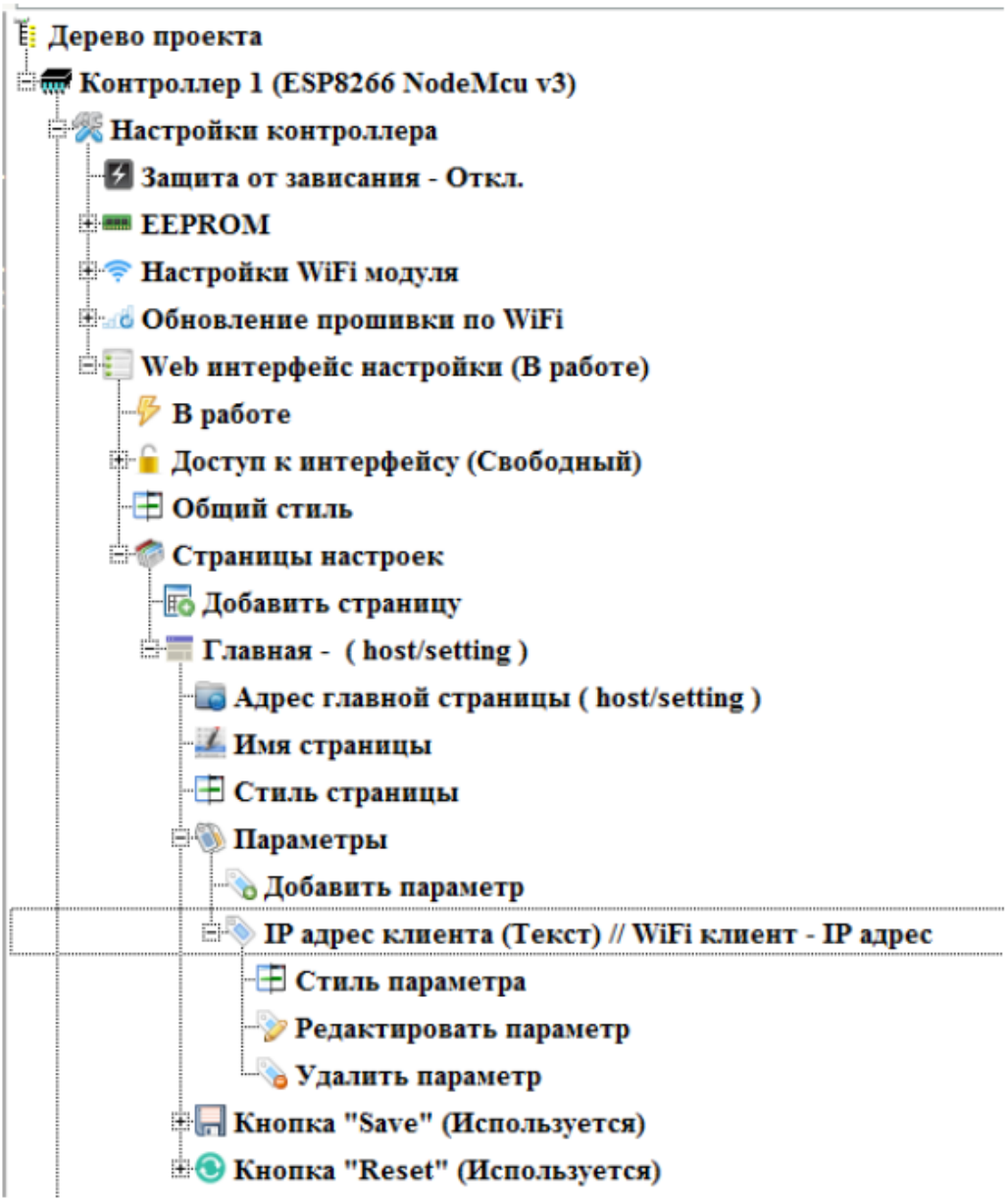
Откроется список доступных системных параметров. Выбираем параметр “Wi-Fi клиент – IP адрес”.



В поле надписи вводим текст лейблы для этого параметра, а в поле тип параметра выберем значение “Текст”.

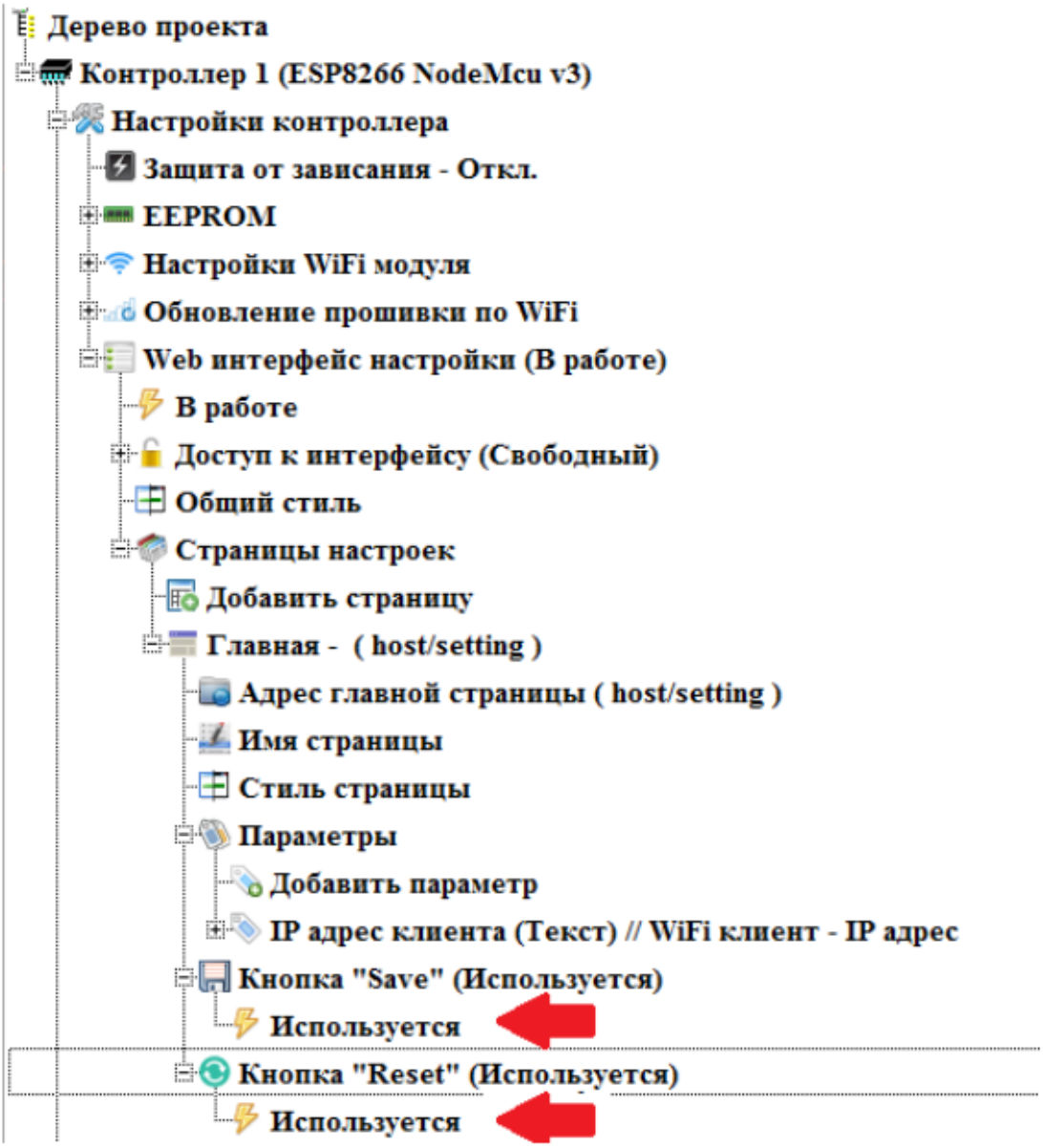


После создания параметра появится новый узел дерева проекта, в котором можно настроить стиль для данного параметра, изменить настройки параметра или удалить его.

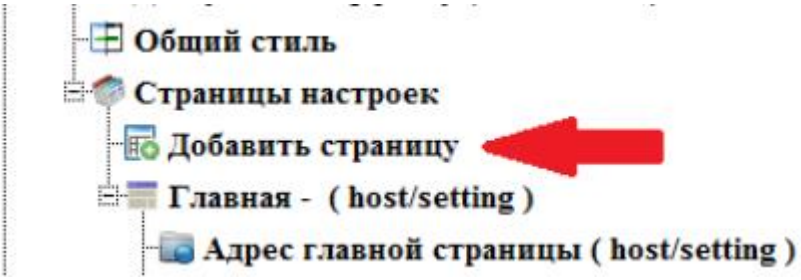


Узлы «Кнопка “Save”» и «Кнопка “Reset”» задают наличие кнопок сохранения данных и перезагрузки контроллера на странице.

Поскольку никаких изменяемых данных на главной странице у нас нет, отключим эти кнопки на странице двойным кликом на ветке «Используется» (по умолчанию кнопки присутствуют на странице).

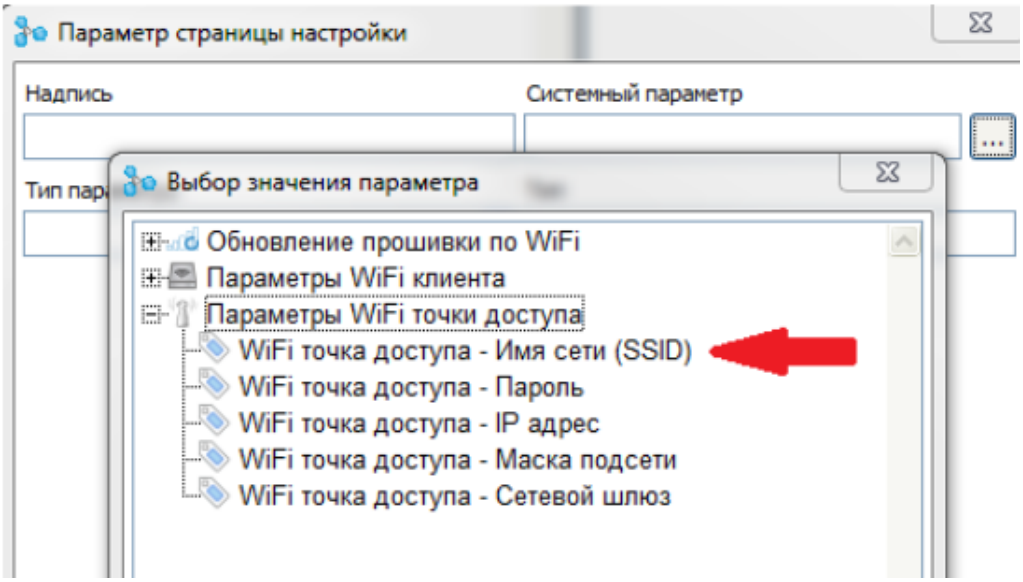


Добавим новую страницу двойным кликом по ветке «Добавить страницу»:

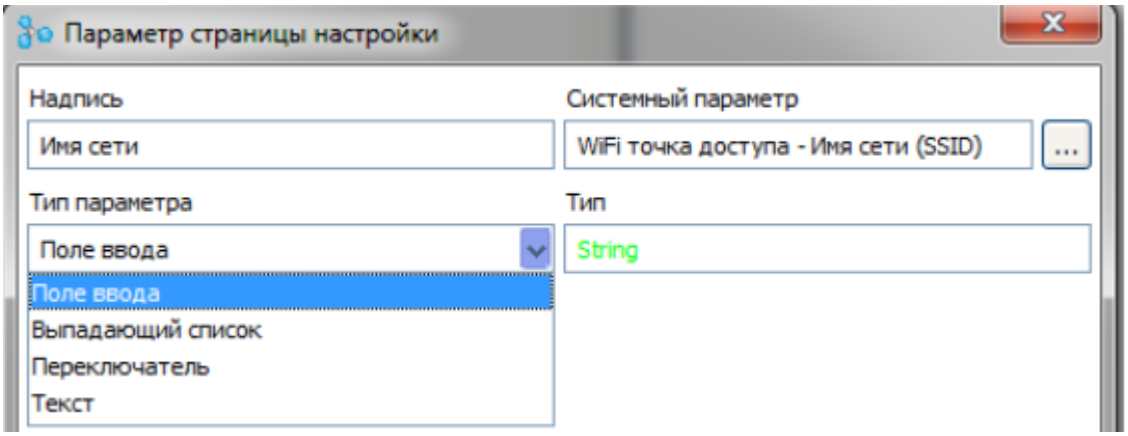


Откроется диалог создания новой страницы. В нём заполним имя страницы (как она будет называться в меню интерфейса настройки) и адрес. Адреса вспомогательных страниц всегда продолжают адрес главной страницы. На данной странице мы будем настраивать параметры точки доступа.

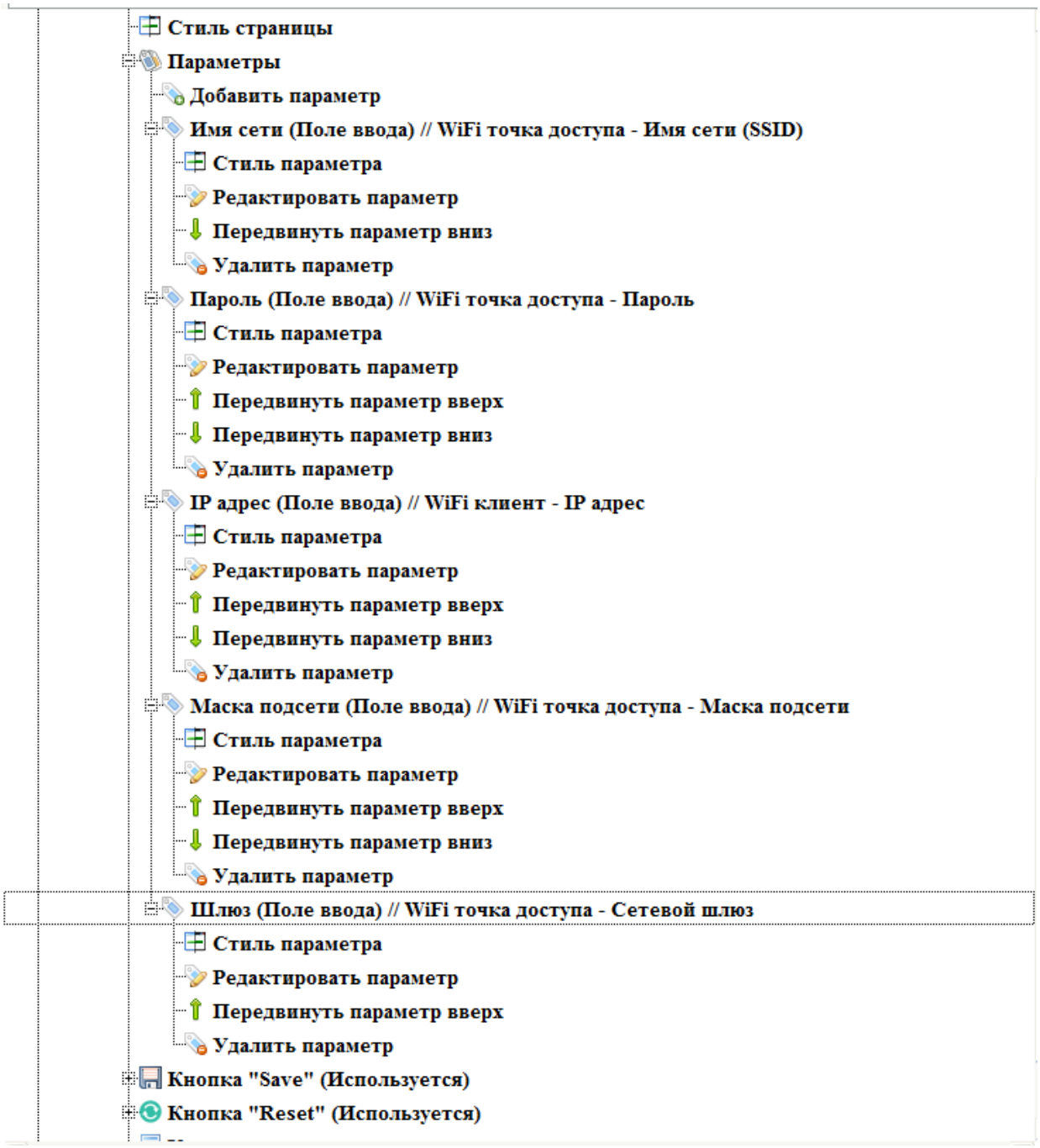
Добавим на страницу параметр “Wi-Fi точка доступа – имя сети (SSID)”.



И выберем для него тип «Поле ввода» и лейблу «Имя сети»

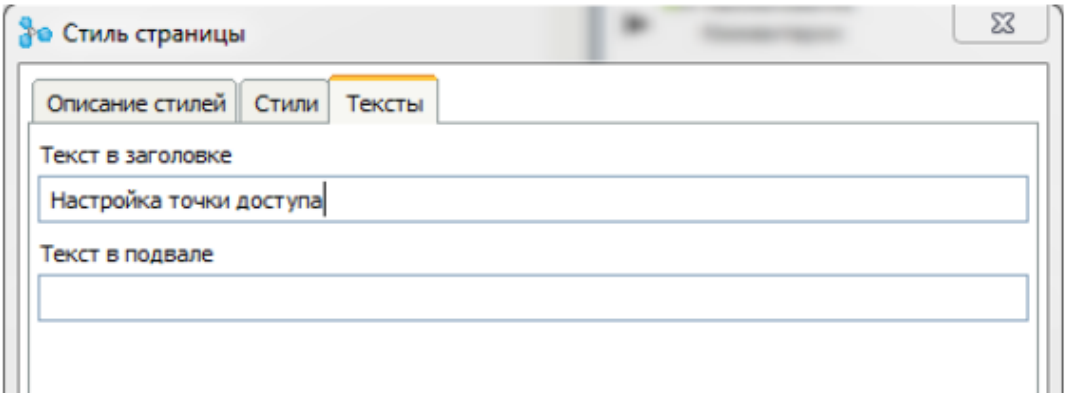


Таким же образом добавим остальные параметры точки доступа:

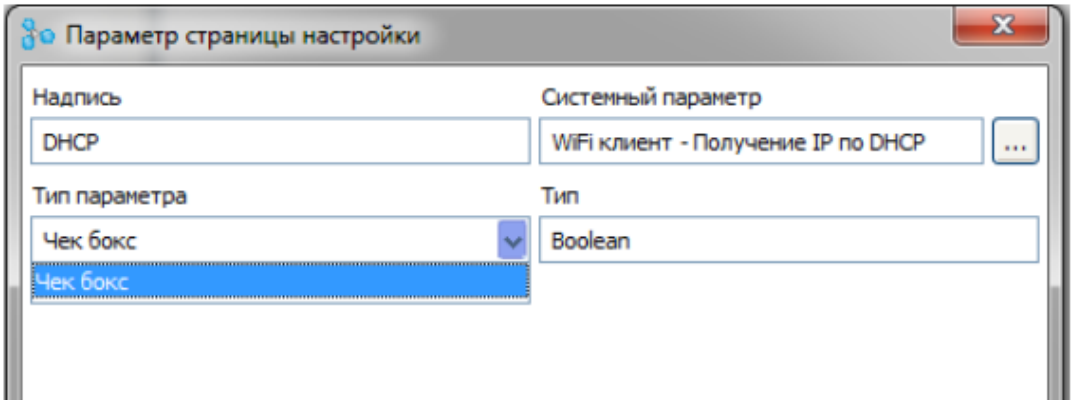


Если параметров на странице более одного в узлах параметров появляются ветки, позволяющие изменить порядок вывода параметров на странице. Поскольку на странице есть изменяемые параметры, оставим на ней кнопки сохранения и перезагрузки контроллера.

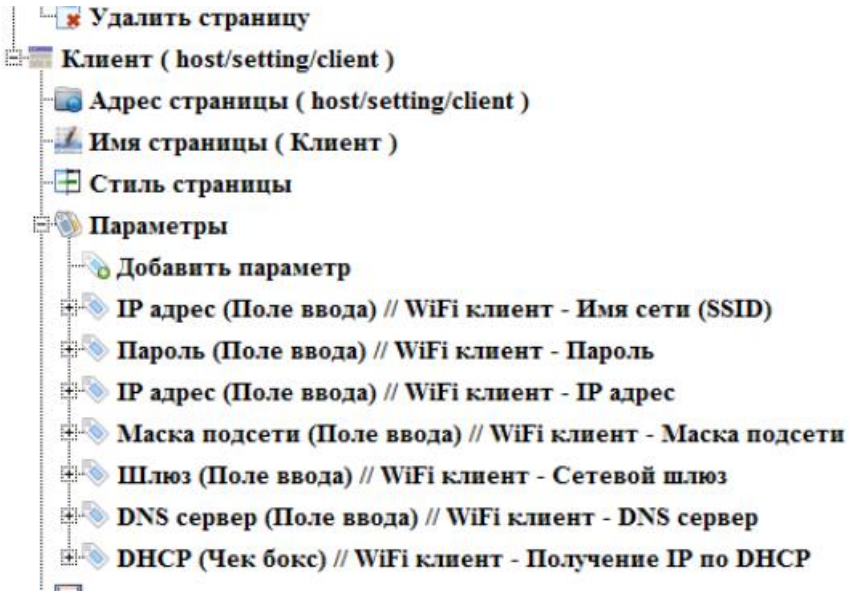
Для данной страницы в диалоге стилей зададим заголовок для данной страницы.



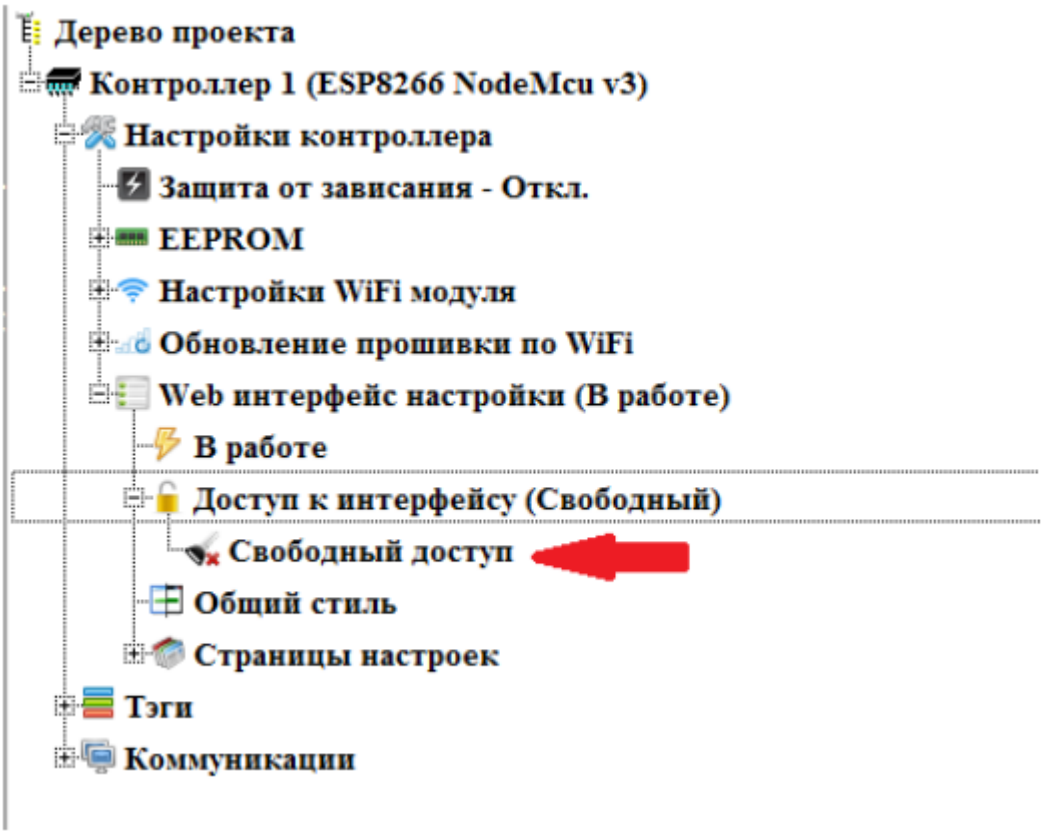
По такому же сценарию создадим страницу с настройками клиента. Для параметра «Wi-Fi клиент – получение IP по DHCP» зададим тип параметра «Чек бокс».



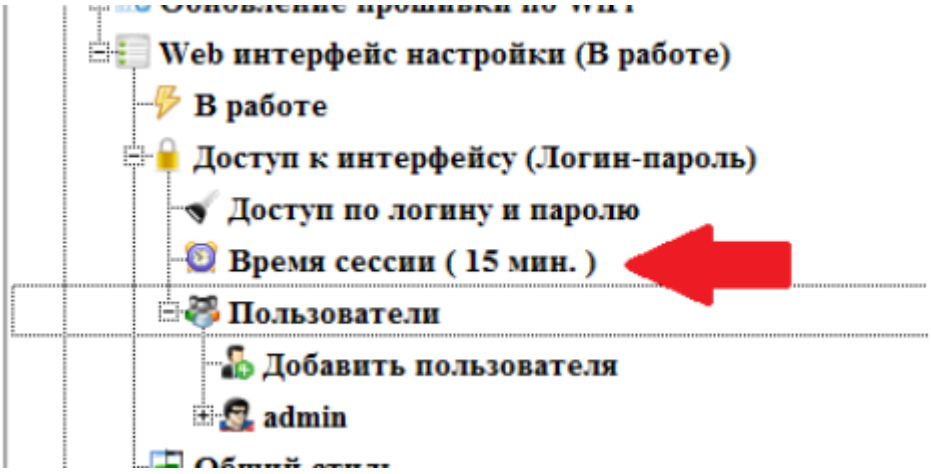
При наличии в интерфейсе более двух страниц в узлах страниц так же появляются ветки, управляющие положением страниц в меню интерфейса. Но главная страница всегда остаётся первой.



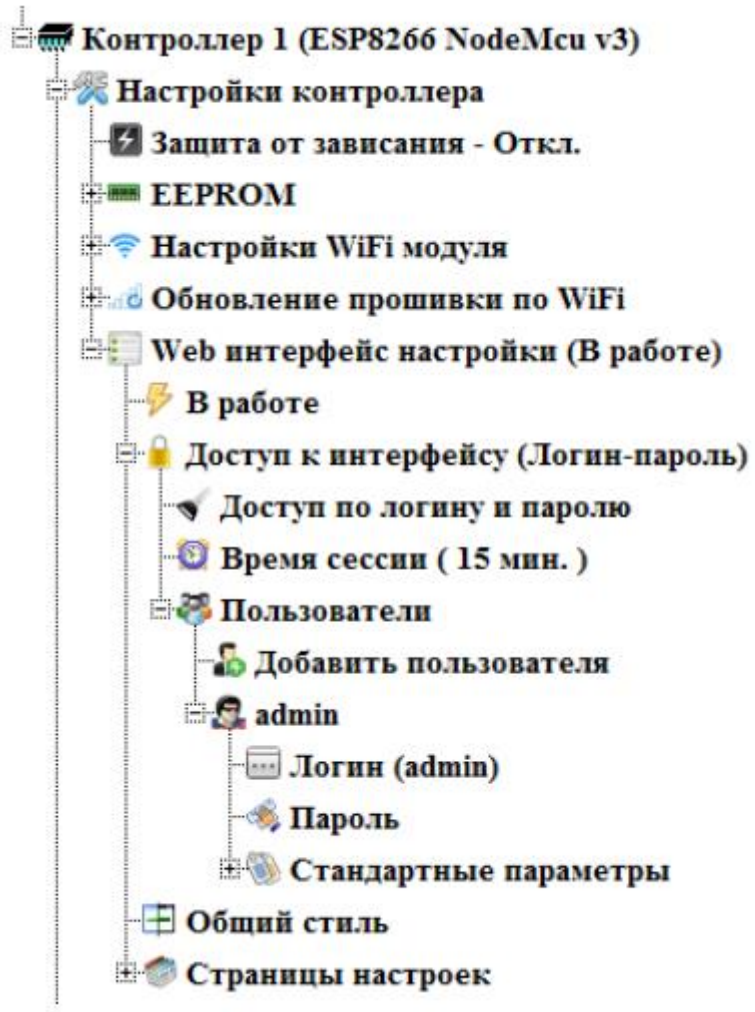
В данный момент доступ к интерфейсу и всем страницам свободный. Введём ограничение доступа. Для этого сделаем двойной клик по ветке «Свободный доступ» узла «Доступ к интерфейсу»:



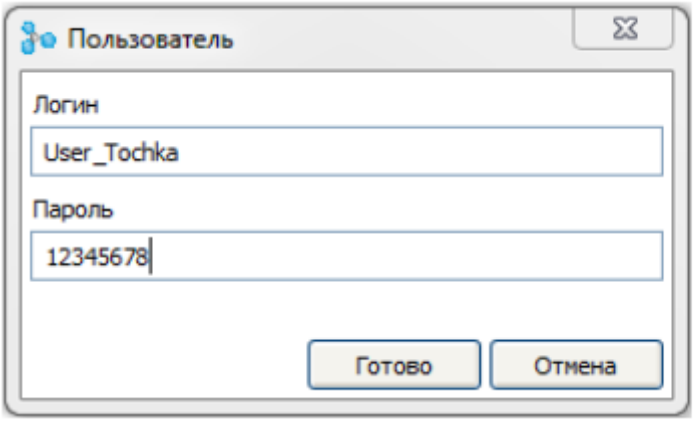
Теперь для доступа к интерфейсу необходимо будет авторизоваться. Время активной авторизации можно задать. По умолчанию оно составляет 15 минут. Через это время после последней активности пользователя будет произведён автоматический сброс текущего пользователя. Это время можно изменить путём двойного клика по соответствующей ветке.



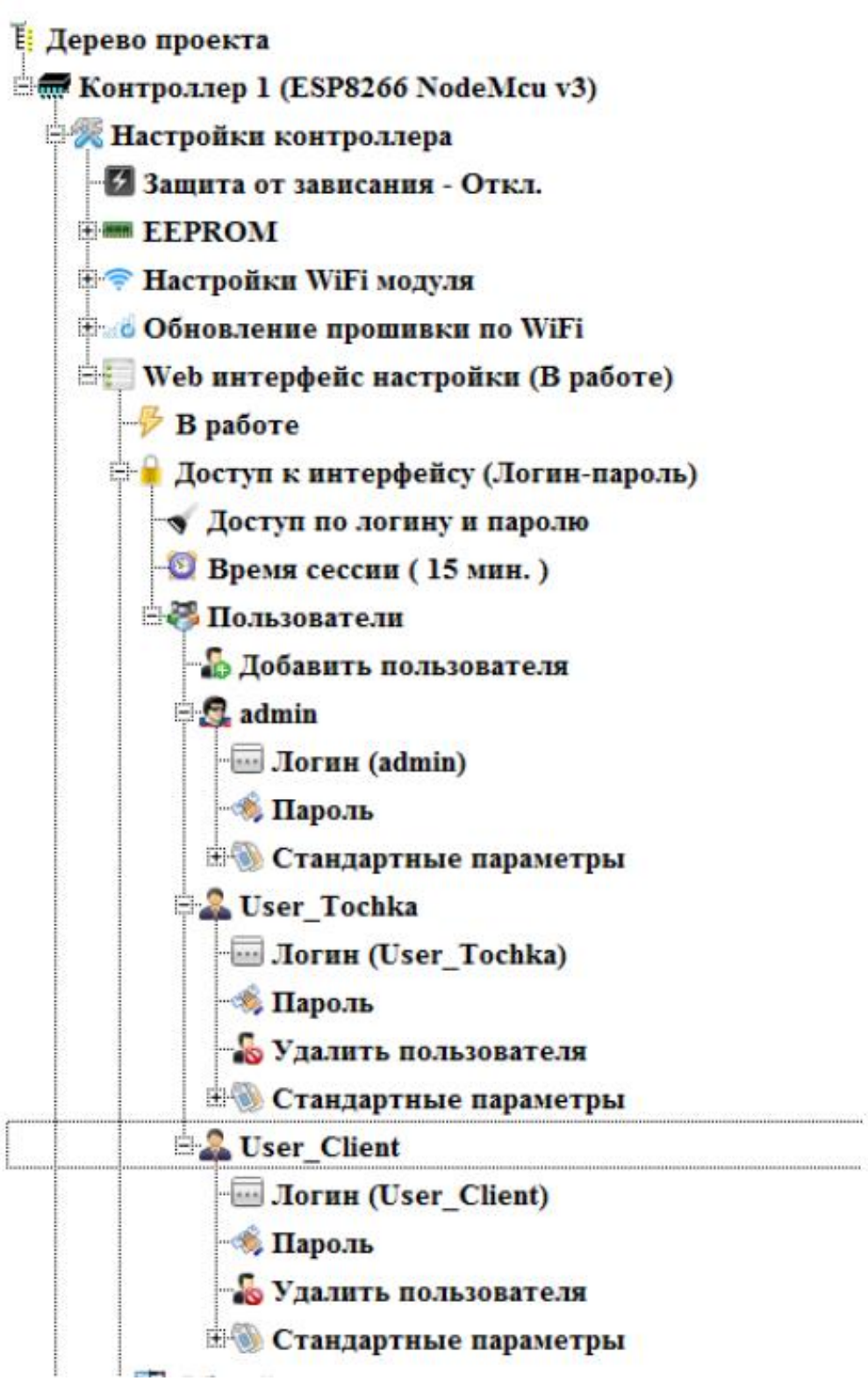
В узле «Пользователи» можно задать необходимое количество пользователей. Там всегда присутствует супер-пользователь (по умолчанию логин — admin). Ему всегда доступны все страницы и параметры. Как и для любого пользователя в его узле можно задать логин и пароль.



Создадим нового пользователя, который будет иметь право настраивать точку доступа. Для этого произведём двойной клик по ветке «Добавить пользователя». Откроется диалог добавления пользователя. В нем мы зададим логин и пароль нового пользователя.

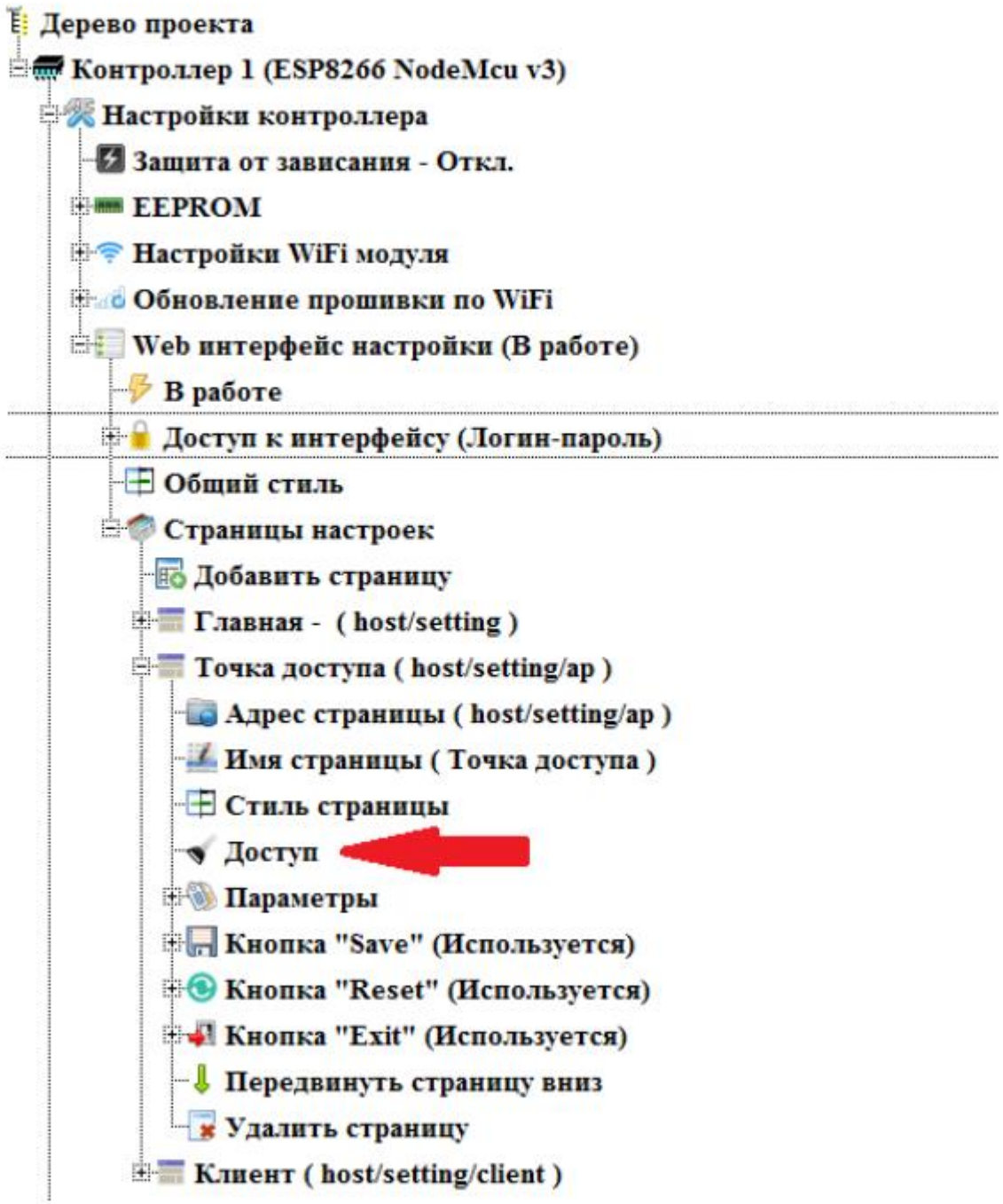


И создадим ещё одного пользователя, который сможет настраивать клиента.

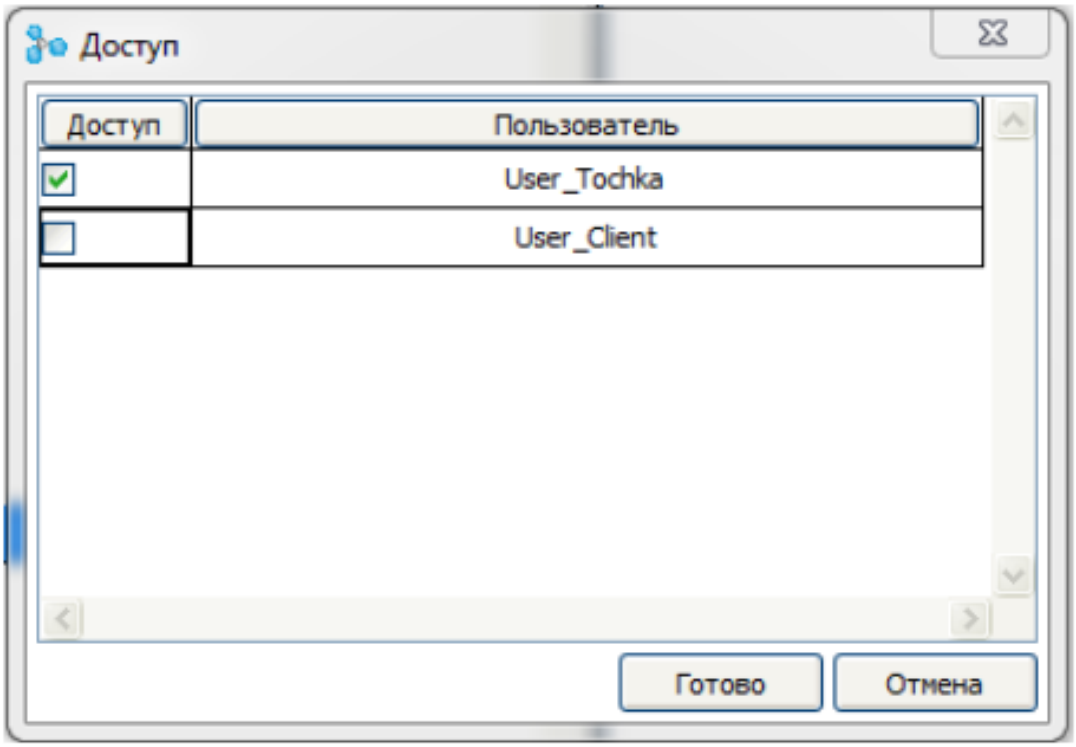


После включения доступа по логину и паролю на всех страницах появилась кнопка «Exit», которая служит для выхода текущего пользователя. Оставим ее на всех страницах.

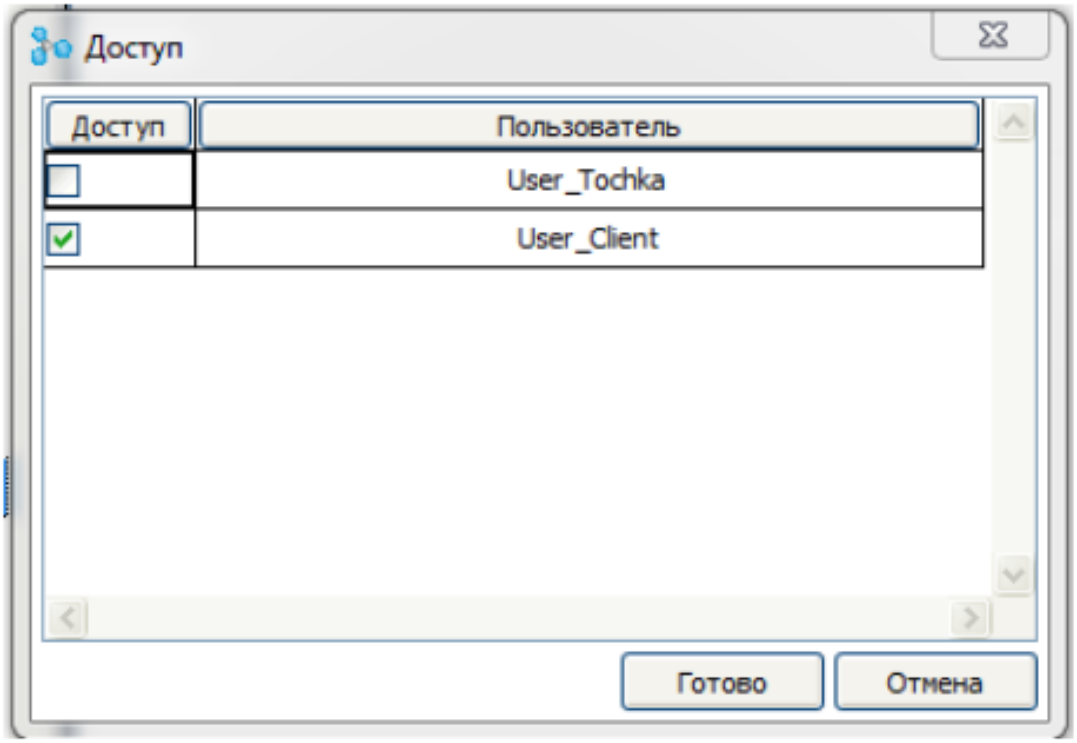
Теперь настроим доступ на страницах. Главная страница всегда доступна всем пользователям. А в узлах остальных страниц появилась новая ветка «Доступ». Дважды кликнем на этой ветке в узле страницы «Точка доступа».



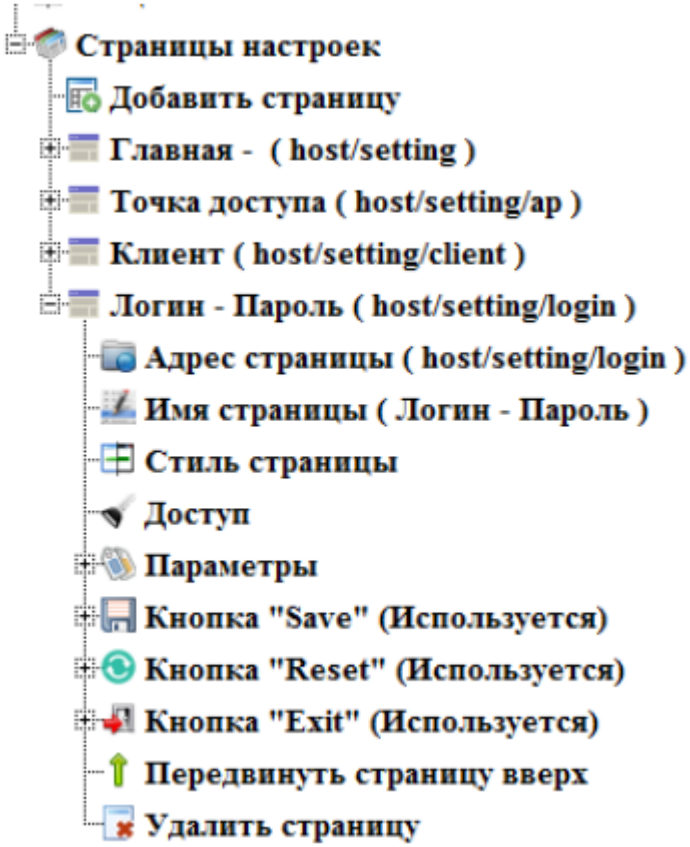
Откроется диалог выбора пользователей для доступа к странице. По умолчанию доступ открыт всем пользователям. Снимем галочку с пользователя «User_Client» тем самым ограничив его доступ к данной странице.



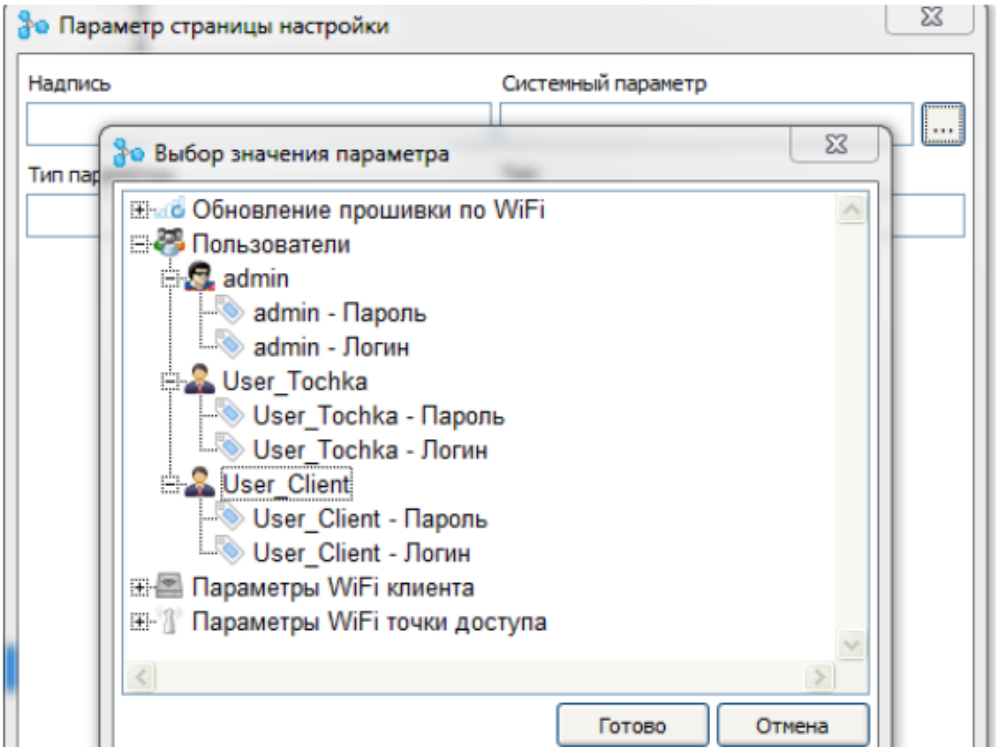
Таким же способом ограничим доступ к странице настроек клиента пользователю «User_Tochka».



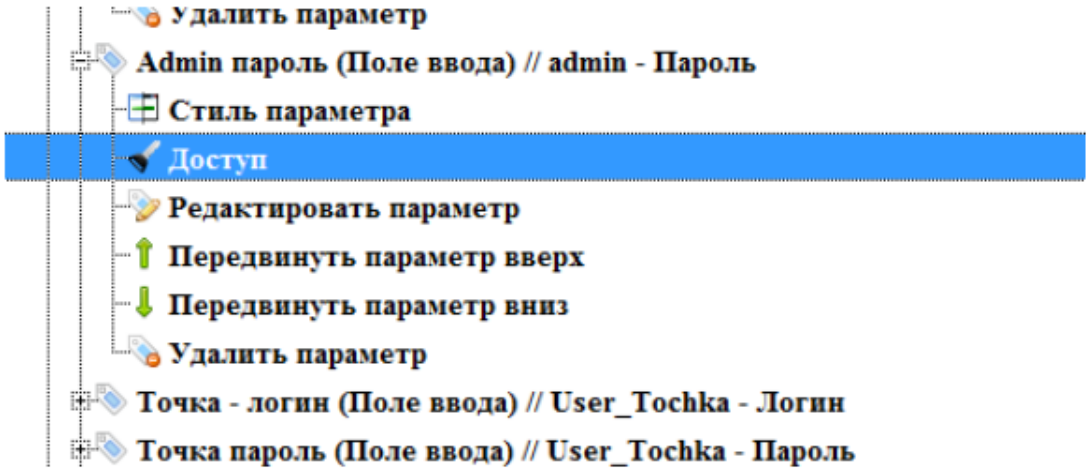
Создадим ещё одну страницу для настроек логинов и паролей:



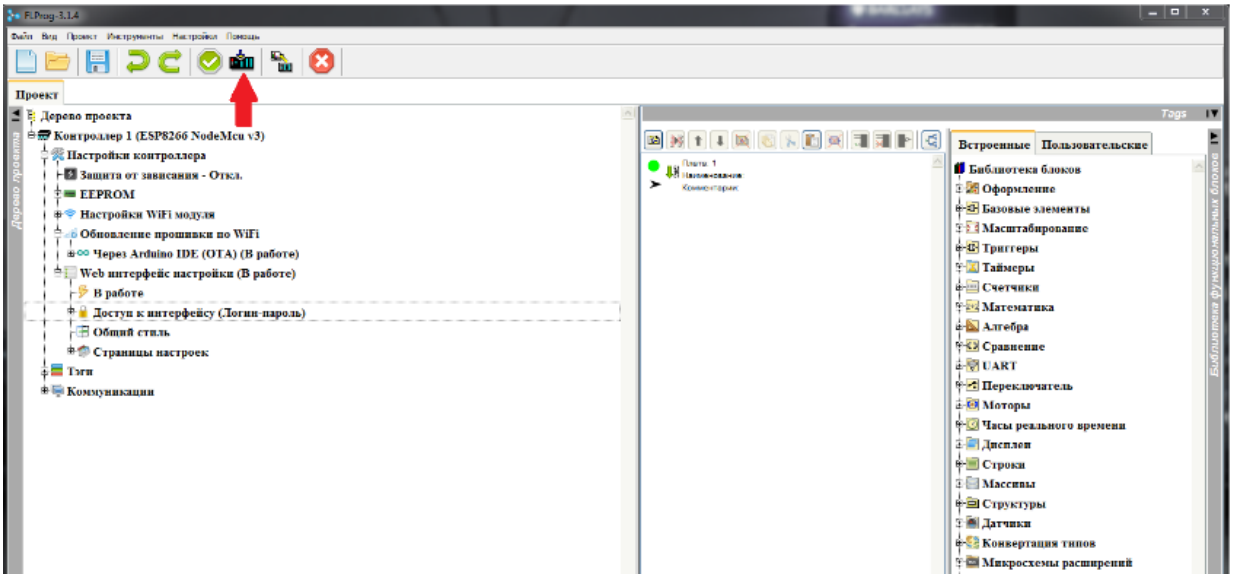
Добавим на эту страницу параметры логинов и паролей для всех пользователей. Эти параметры появились в списке системных параметров после включения доступа по логину и паролю.



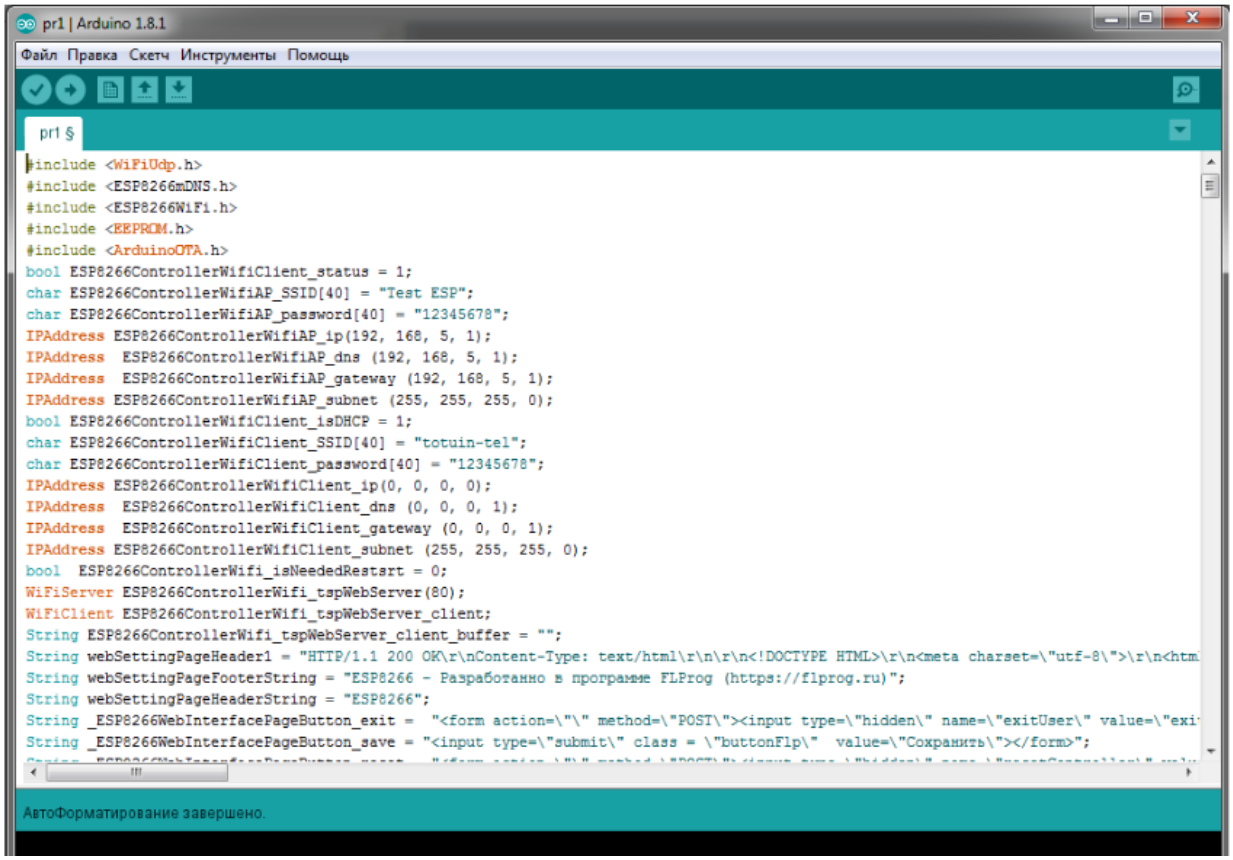
Для данной страницы не будем настраивать ограничение доступа, а настроим ограничение непосредственно на параметры. В узлах параметров появились ветки настройки доступа аналогичные настройкам доступа к страницам.

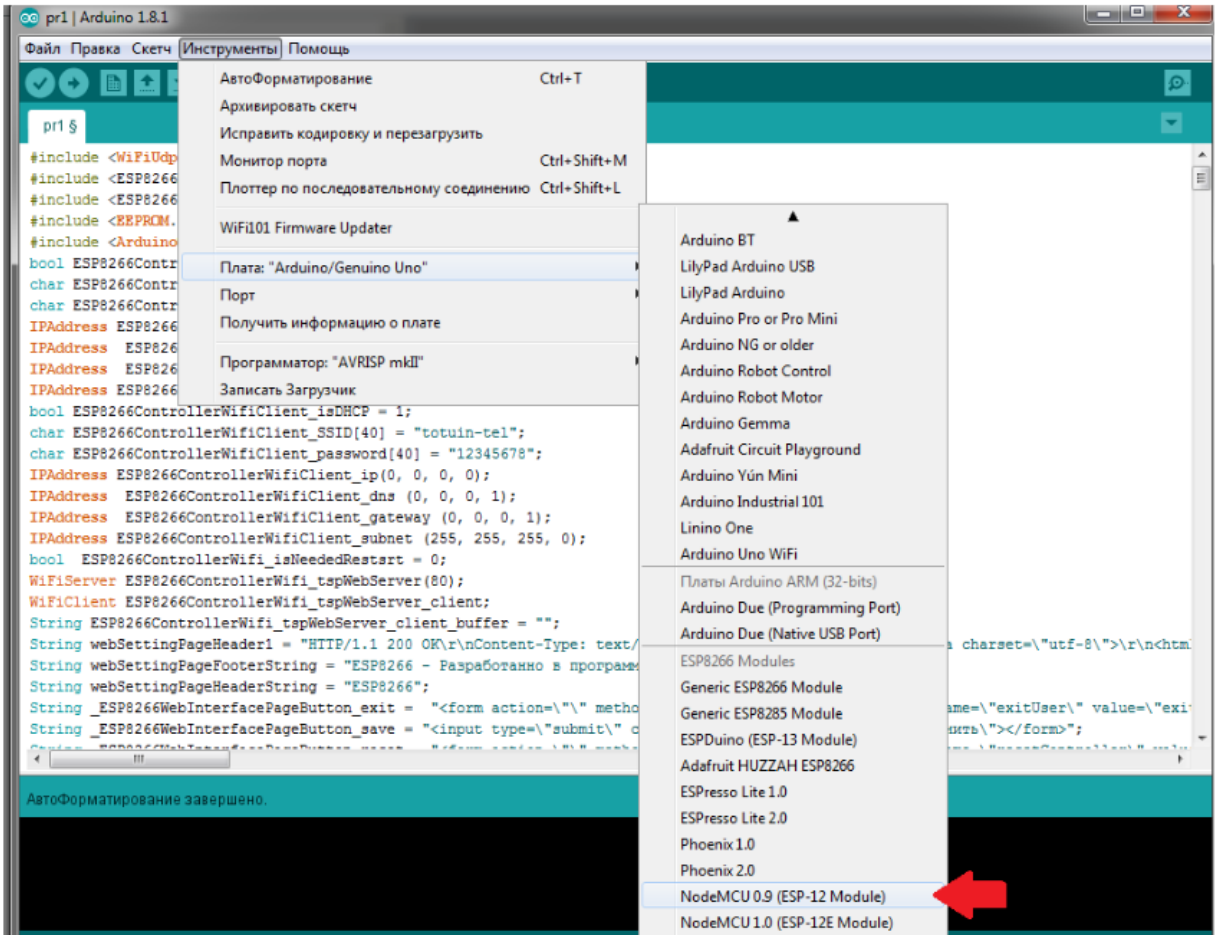


К параметрам админа запретим доступ обеим пользователям, а к параметрам пользователей оставим доступ только тому пользователю, которому эти параметры принадлежат. На этом настройку web интерфейса считаем законченной. Нажимаем кнопку «Компилировать проект» в главном интерфейсе программы и получаем готовый скетч в Arduino IDE:

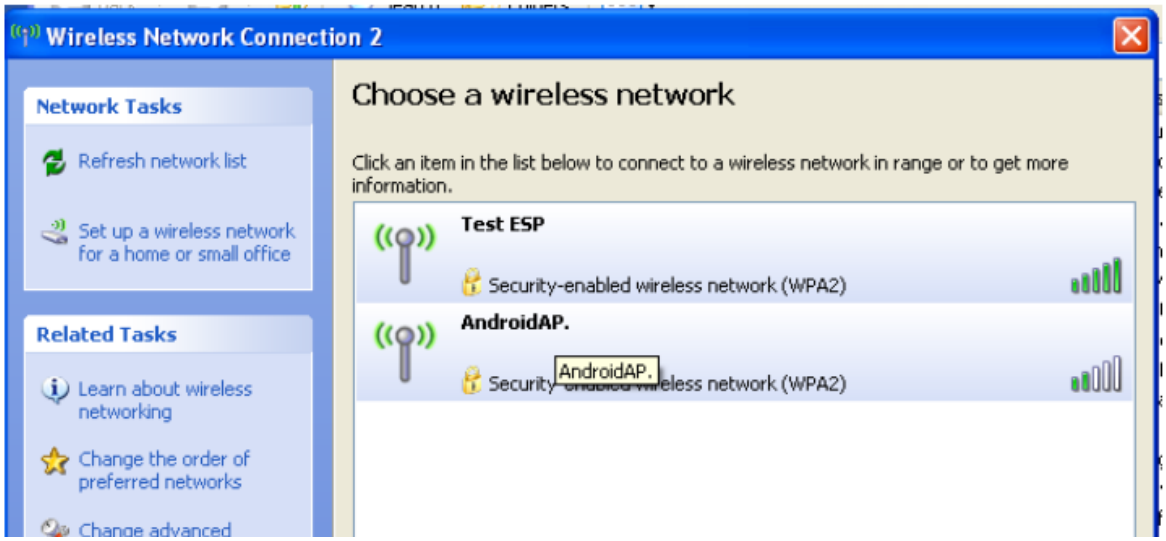


В Arduino IDE выбираем нашу плату. И порт, к которому подключен контроллер. После чего заливаем прошивку в контроллер:

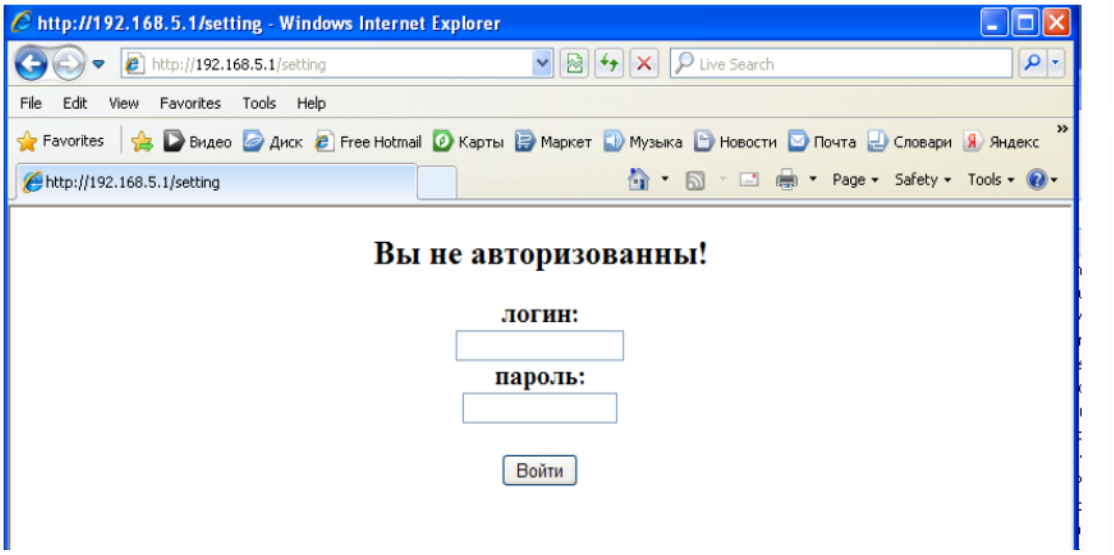

pr1 | Arduino 1.8.1
Файл Правка Скетч Инструменты Помощь
pr1 §
#include <WiFiUdp.h>
#include <ESP8266mDNS.h>
#include <ESP8266WiFi.h>
#include <EEPROM.h>
#include <ArduinoOTA.h>
bool ESP8266ControllerWifiClient_status = 1;
char ESP8266ControllerWifiAP_SSID[40] = "Test ESP";
char ESP8266ControllerWifiAP_password[40] = "12345678";
IPAddress ESP8266ControllerWifiAP_ip(192, 168, 5, 1);
IPAddress ESP8266ControllerWifiAP_dns (192, 168, 5, 1);
IPAddress ESP8266ControllerWifiAP_gateway (192, 168, 5, 1);
IPAddress ESP8266ControllerWifiAP_subnet (255, 255, 255, 0);
bool ESP8266ControllerWifiClient_isDHCP = 1;
char ESP8266ControllerWifiClient_SSID[40] = "totuin-tel";
char ESP8266ControllerWifiClient_password[40] = "12345678";
IPAddress ESP8266ControllerWifiClient_ip(0, 0, 0, 0);
IPAddress ESP8266ControllerWifiClient_dns (0, 0, 0, 1);
IPAddress ESP8266ControllerWifiClient_gateway (0, 0, 0, 1);
IPAddress ESP8266ControllerWifiClient_subnet (255, 255, 255, 0);
bool ESP8266ControllerWifi_isNeededRestart = 0;
WiFiServer ESP8266ControllerWifi_tspWebServer(80);
WiFiClient ESP8266ControllerWifi_tspWebServer_client;
String ESP8266ControllerWifi_tspWebServer_client_buffer = "";
String webSettingPageHeader1 = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<meta charset=\"utf-8\">\r\n<htm
String webSettingPageFooterString = "ESP8266 - Разработанно в программе FLProg (https://flprog.ru)";
String webSettingPageHeaderString = "ESP8266";
String _ESP8266WebInterfacePageButton_exit = "<form action=\"\" method=\"POST\"><input type=\"hidden\" name=\"exitUser\" value=\"exi
String _ESP8266WebInterfacePageButton_save = "<input type=\"submit\" class = \"buttonFlp\" value=\"Сохранить\"></form>";
АвтоФорматирование завершено.

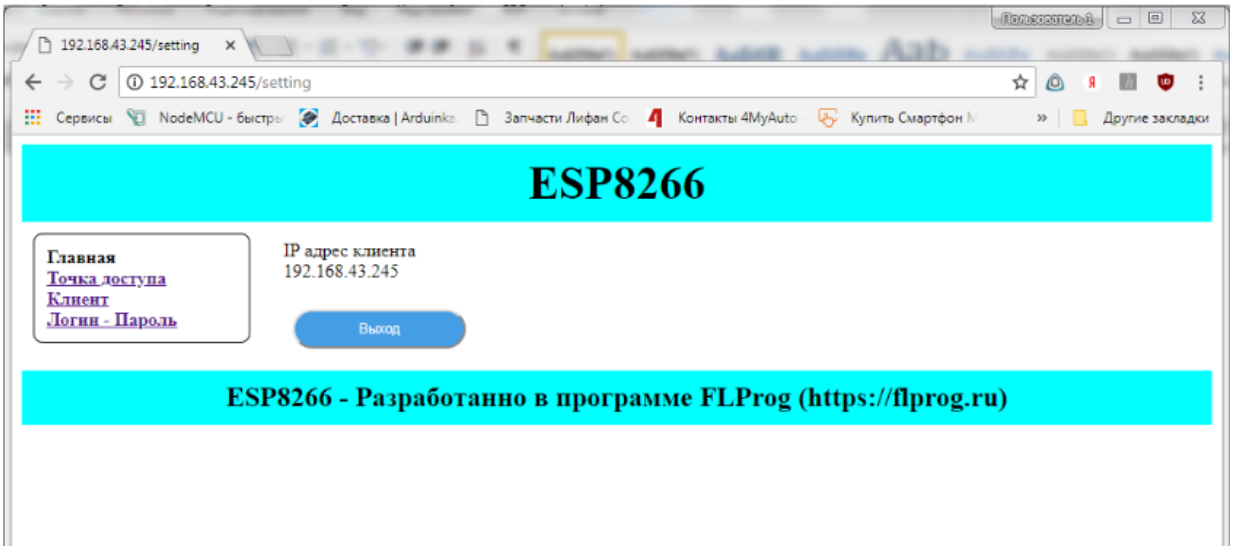

pr1 | Arduino 1.8.1
Файл Правка Скетч Инструменты Помощь
АвтоФорматирование Ctrl+T
Архивировать скетч
Исправить кодировку и перезагрузить
Монитор порта Ctrl+Shift+M
Плоттер по последовательному соединению Ctrl+Shift+L
WiFi101 Firmware Updater
Плата: "Arduino/Genuino Uno"
Порт
Получить информацию о плате
Программатор: "AVRISP mkII"
Записать Загрузчик
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi
Платы Arduino ARM (32-bits)
Arduino Due (Programming Port)
Arduino Due (Native USB Port)
ESP8266 Modules
Generic ESP8266 Module
Generic ESP8285 Module
ESPDuino (ESP-13 Module)
Adafruit HUZZAH ESP8266
ESPresso Lite 1.0
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 2.0
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
АвтоФорматирование завершено.

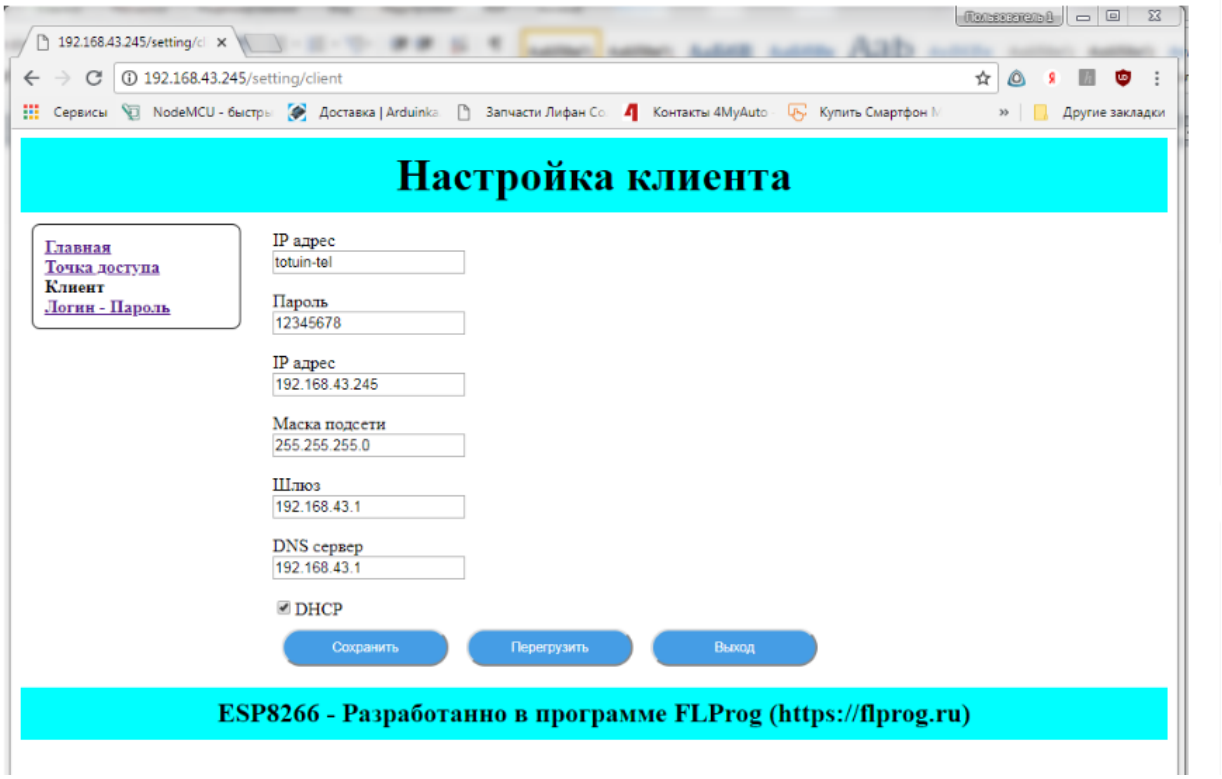
После загрузки появляется новая точка доступа:



Подключаемся к ней и заходим по заданному нами в проекте адресу. Появляется страница авторизации:

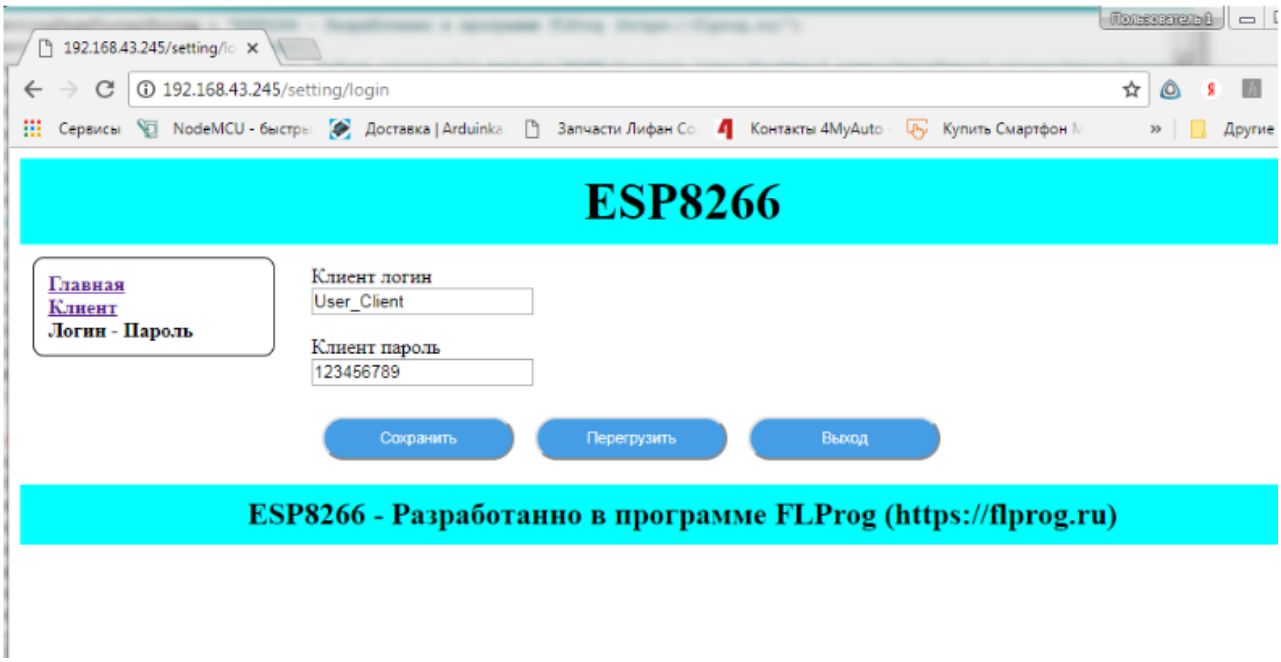




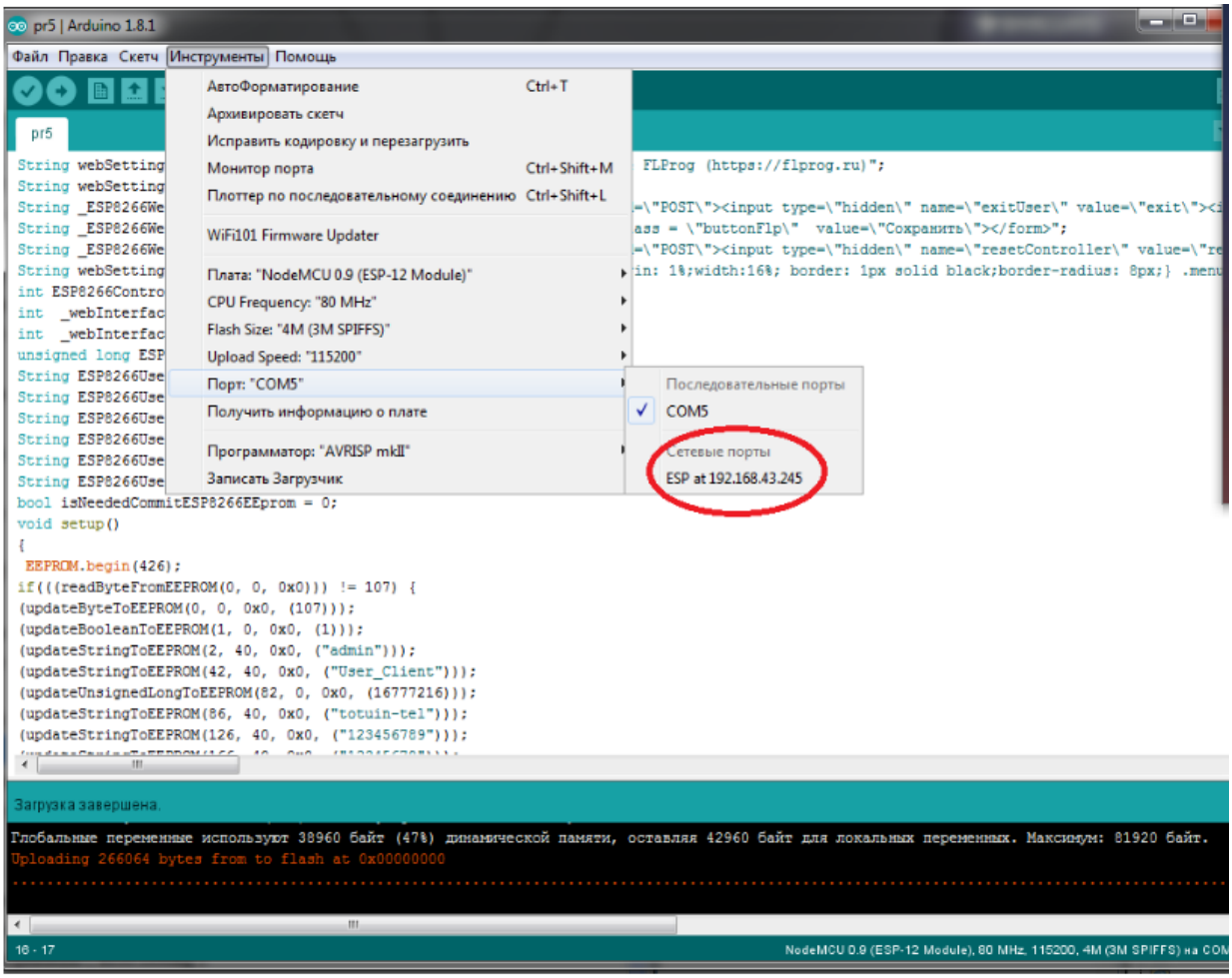


А на странице логина и пароля все параметры. Выходим и логинимся под **User_Tochka**. Видим в меню только доступные страницы, и только свой логин и пароль.

Перелогиниваемся под **User_Client** и картина соответствующая – видим только то, что положено.



Открываем Arduino IDE и в настройках порта видим, что контроллер готов к обновлению прошивки «по воздуху».



Требования к отчету:

Индивидуальное задание и требование к отчету будут реализованы вместе МДК «Организация и принципы построения компьютерных систем» или МДК «Защита информации в компьютерных системах».

Форма отчетности – ответы на вопросы по реализации проекта (устная форма), код программы на языке C/C++.

HMI (Human-Machine-Interface)

Человеко-машинный интерфейс (ЧМИ) (англ. Human-machine interface, HMI) — широкое понятие, охватывающее инженерные решения, обеспечивающие взаимодействие человека-оператора с управляемыми им машинами.

Создание систем человеко-машинного интерфейса тесно увязано с понятиями эргономика и юзабилити.

Проектирование ЧМИ включает в себя:

создание рабочего места: кресла, стола, или пульта управления, размещение приборов и органов управления (устройства ввода данных) (соответствием всего этого физиологии человека занимается эргономика), освещение рабочего места и, возможно, микроклимат.

далее рассматриваются взаимодействие оператора со всеми органами управления: их доступность и необходимые усилия, эффективность и скорость доступа, согласованность (непротиворечивость) управляющих воздействий (в том числе т. н. «защита от дурака»), расположение дисплеев и размеры надписей на них (всё это входит в сферу юзабилити).

Юзабилити (от англ. usability — «удобство и простота использования, степень удобства использования»), также удобство использования, пригодность использования, эргономичность — способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях (**ISO/IEC 25010**); свойство системы, продукта или услуги, при наличии которого конкретный пользователь может эксплуатировать систему в определенных условиях для достижения установленных целей с необходимой результативностью, эффективностью и удовлетворённостью (**ISO 9241-210**).

Удобство (пригодность) использования системы не сводится только к тому, насколько её легко эксплуатировать. В соответствии со стандартами серии ISO 9241 эту характеристику следует понимать более широко, учитывая личные цели пользователя, его эмоции и ощущения, связанные с восприятием системы, а также удовлетворённость работой. Свойства, необходимые для обеспечения пригодности использования, зависят также от задачи и окружающей среды.

Пригодность использования — не абсолютное понятие, оно может различным образом проявляться в определённых условиях эксплуатации

Одной из наиболее сложных задач является создание эффективного ЧМИ рабочих мест сложных машин с множеством органов управления, например, пилотов самолёта и космических кораблей.

В промышленных условиях ЧМИ чаще всего реализуется с использованием типовых средств: операторских панелей, компьютеров и типового программного обеспечения.

Вашему вниманию предлагается изучение научной статьи **«Разработка человеко-машинного интерфейса и его применение в системах управления»** Сверчкова Дениса Сергеевича.

В статье (ссылка на нее расположена посредством интер-отклика) описаны различные типы человеко-машинных интерфейсов, области применения каждого интерфейса, этапы разработки человеко-машинного интерфейса и приведен пример реализации человеко-машинного интерфейса (ЧМИ) для системы управления (СУ).

Рекомендуется также ознакомиться с материалом, адаптированным под разработку и изучение в рамках **IDE FLProg** на портале «Хабрахабр»:

- FLProg + Nextion HMI. Урок 1
- FLProg + Nextion HMI. Урок 2
- FLProg + Nextion HMI. Урок 3
- (https://habr.com/ru/company/flprog/blog/392561/)

Дисплеи **Nextion** — это модули с цветными сенсорными экранами и контроллерами, в которые Вы можете записывать свои программы. На модулях дисплеев Nextion имеется разъём UART и выводы GPIO, что позволяет использовать дисплеи Nextion как совместно с Arduino/ESP (подключая дисплей к микропроцессорной системе по шине UART), так и отдельно (подключая кнопки, светодиоды, реле и т.д. напрямую к выводам GPIO дисплеев).

В завершении данного учебно-практического издания вашему вниманию предлагается для изучения **презентации «Обзор современных программируемых логических контроллеров. SCADA системы в автоматизированных производствах» состоящая из 38 слайдов.**

Программа презентации подготовлена таким образом, что в ней раскрываются технические особенности устройств (ПЛК), понятные студентам, осваивающих образовательные программы по укрупненной группе специальностей 09.00.00 «Информатика и вычислительная техника», а не только осваивающим учебные программы по укрупненной группе "Автоматизация технологических процессов и производств (по отраслям)".

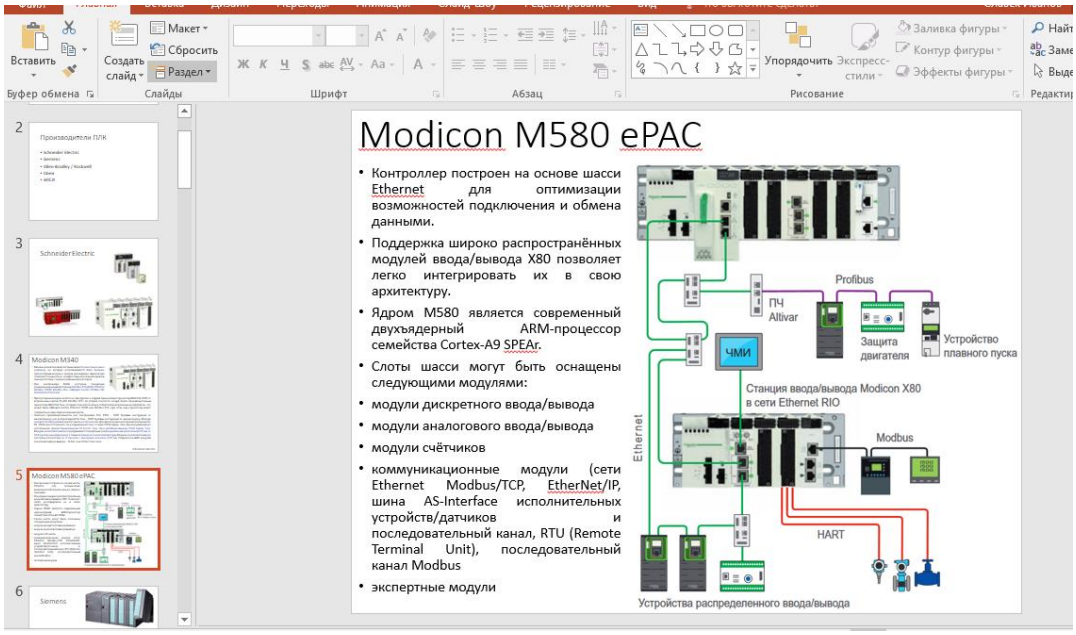


Рис. 40 – Скриншоты из предлагаемой для изучения презентации.

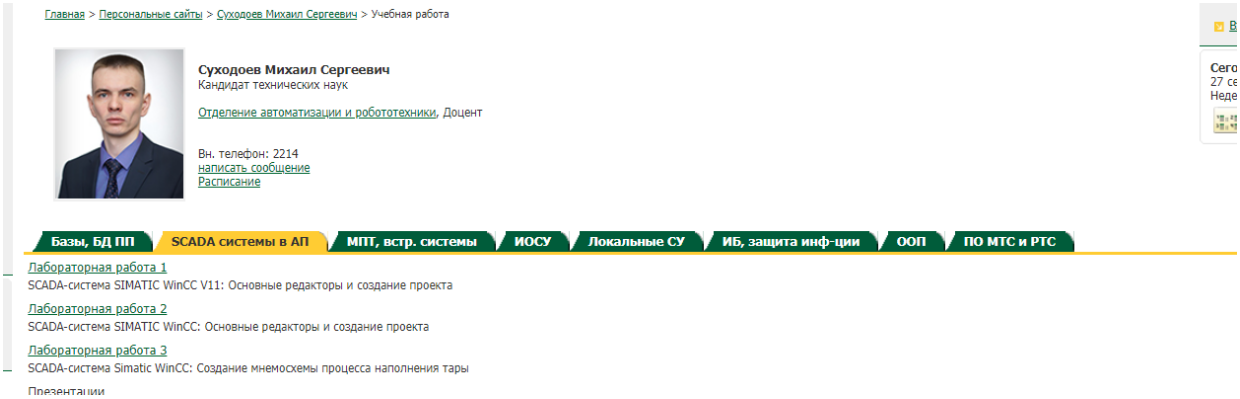


Рис. 41 – Составитель презентации – доцент Томского политехнического университета Суходоев Михаил Сергеевич

Таким образом вся программа данного издания построена на вовлечение будущих специалистов укрупненной группы 09.00.00 в смежные направления, для приобретения тех или иных профессиональных компетенций и первичных навыков, расширяющих область работы будущих специалистов.

Подводя итог, еще раз акцентируем на область применения данного практикума: ознакомление с принципами визуального программирования с несколькими видами микропроцессорных систем, получение теоретических навыков работы с специализированной документацией, приобретением умения работать с техническими заданиями в нескольких интегрированных средах разработки, ознакомлением с понятием человеко-машинного интерфейса, навыков дифференцирования микропроцессорных систем, программируемых логических контроллеров и иной управляющей и исполнительной цифровой техники в рамках концепции Internet of Things/Internet of Everything.

Презентация «Обзор современных программируемых логических контроллеров. SCADA системы в автоматизированных производствах)
https://yadi.sk/i/AtlkjUUVeV4cXA