Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 – Программная инженерия
Магистерская программа Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«15» 07 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Разработка аппаратно-программного обеспечения системы SCHome и ее модулей по технологии «Internet of Things»

Исполнитель студент группы 857-ом	08.07.2020 (подпись, дата)	А.Е. Демьяненко
Руководитель доцент, канд.техн.наук	09.07.2020 (подпись, дата)	С.Г. Самохвалова
Руководитель магистерской программы профессор, доктор техн. наук	14.07.2020 (подпись, дата)	И.Е. Еремин
Нормоконтроль доцент, канд. техн. наук	10.07.2020 (подпись, дата)	В.В. Еремина
Рецензент	11.07.20 (подпись, дата)	О.Г. Какаулин

Благовещенск 2020

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«02» 09 2020 г.

ЗАДАНИЕ

К магистерской диссертации студента Демьяненко Александра Евгеньевича.
1. Тема магистерской диссертации: Разработка аппаратно-программного обеспечения системы SCHome и ее модулей по технологии «Internet of Things»
(утверждено приказом от _____).
2. Срок сдачи студентом законченной работы 16.07.2020 г.
3. Исходные данные к магистерской диссертации: отчет по преддипломной практике, ГОСТы, научные публикации, дополнительная литература.
4. Содержание магистерской диссертации: анализ деятельности объекта автоматизации, исследование вопросов информационной безопасности, разработка прототипа автоматизированной информационной системы.
5. Перечень материалов приложения: - .
6. Дата выдачи задания: 02.09.2019 г.
Руководитель магистерской диссертации: Самохвалова Светлана Геннадьевна, доцент, канд.техн.наук.

Задание принял к исполнению _____ А.Е. Демьяненко
(подпись студента)

РЕФЕРАТ

Магистерская диссертация содержит 72 с., 34 рисунка, 1 таблица, 20 источников, 1 приложение.

ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА, СИСТЕМА, ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ, ЗАЩИТА ИНФОРМАЦИИ, ИНТЕРНЕТ ВЕЩЕЙ, УМНЫЙ ДОМ, УДАЛЕННЫЙ ДОСТУП, ПРОТОТИПИРОВАНИЕ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

Целью магистерской диссертации является разработка информационной системы «SCHome» для удаленного управления оборудованием в доме или на предприятии.

В настоящей работе разработана информационная система «SCHome», обеспечивающая удаленное управление оборудованием в доме. Так же была проанализирована работа локальной вычислительной сети (ЛВС) системы, проанализированы виды, классификации и способы предотвращения угроз ЛВС. Разработано программное обеспечение модулей системы, прототипы модулей SCH-Base, SCH-Switch1, SCH-Switch2, SCH-Switch4, SCH-Plant, SCH-Atmos, SCH-Solar, а также прототип для наглядной демонстрации работы системы «SCHome».

Информационная система «SCHome» позволит значительно снизить тепло- и электро-затраты, а также автоматизировать некоторые аспекты жизни человека. Реализованный удаленный доступ позволит безопасно управлять домом или получать подробную информацию о его состоянии. Программная подсистема работы по расписанию позволит убрать участие человека в рутинных процессах.

Реализация сборных комплектов системы с различными модулями с информацией по сборке, документацией и бесплатным программным обеспечением, позволит вовлечь школьников, студентов, любителей радиотехники и программирования в изучение технологий «Интернета вещей», и «Умного дома».

СОДЕРЖАНИЕ

Введение 10

1 Развитие технологий «Умный дом» и «Интернет Вещей» 12

1.1 Технология «Умный дом» 12

1.2 Технология «Интернет Вещей» 14

1.2.1 Достоинства технологии Интернета Вещей 19

1.2.2 Недостатки технологии Интернета Вещей 20

2 Программное, аппаратное и алгоритмическое обеспечение задачи 21

2.1 Анализ предметной области 21

2.1.1 Постановка задачи 21

2.1.2 Назначение программного обеспечения 22

2.1.3 Обзор существующего программного обеспечения решающего поставленную задачу 22

2.2 Аппаратное обеспечение решения задачи. 23

2.2.1 Обоснование выбора архитектуры системы 23

2.2.2 Обоснование выбора аппаратных средств 24

2.3 Программное обеспечение решения задачи 28

2.3.1 Платформа Blynk 28

2.3.2 Обоснование выбора языка программирования 31

3 Анализ возможных типов атак и модели нарушителя, осуществляющего атаки на локальную сеть системы «SCHome» 32

3.1 Анализ сетевого трафика 32

3.2 Подмена доверенного объекта или субъекта ЛВС 32

3.3 Ложный объект ЛВС 33

3.3.1 Внедрение в ЛВС ложного объекта путём навязывания ложного маршрута 33

3.3.2 Использование ложного объекта для организации удалённой атаки на ЛВС 34

4 Защита локальной сети системы 36

4.1 Главные цели сетевой безопасности 36

4.2 Анализ методов и средств защиты информации, применяемых в локальных сетях 37

4.3 Способы защиты информации 38

4.4 Идентификация и аутентификация 40

4.5 Управление доступом 42

5 Программная реализация системы управления оборудованием и техникой «SCHome» 44

5.1 Модели жизненного цикла программного обеспечения 44

5.1.1 Описание процессов жизненного цикла спиральной модели 44

5.1.2 Обоснование выбора модели жизненного цикла для разрабатываемого программного средства 45

5.2 Проектирование основного модуля SCH-Base 47

5.2.1 Проектирование подсистемы работы модулей по расписанию 50

5.2.2 Проектирование подсистемы работы модулей в ручном режиме 51

5.3 Проектирование web-интерфейса модулей 53

5.4 Проектирование модуля SCH-Switch1 55

5.5 Проектирование модуля SCH-Switch2 56

5.6 Проектирование модуля SCH-Switch4 56

5.7 Проектирование модуля SCH-Plant 58

5.8 Проектирование модуля SCH-Solar 58

5.9 Проектирование модуля SCH-Atmos 60

5.10 Проектирование модуля SCH-Cam 61

5.11 Проектирование модуля SCH-Wind 62

5.12 Проектирование модуля SCH-FaceId 62

5.13 Проектирование модуля SCH-Broadcast 62

5.14 Создание прототипа системы SCHome 62

Библиографический список 67
Приложение А 71

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей магистерской диссертации использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 19.001-77 ЕСПД	Общие положения;
ГОСТ 19.004-80 ЕСПД	Термины и определения;
ГОСТ 19.101-77 ЕСПД	Виды программ и программных документов;
ГОСТ 19.102-77 ЕСПД	Стадии разработки;
ГОСТ 19.103-77 ЕСПД	Обозначение программ и программных документов;
ГОСТ 19.104-78 ЕСПД	Основные надписи;
ГОСТ 19.105-78 ЕСПД	Общие требования к программным документам;
ГОСТ 19.106-78 ЕСПД	Требования к программным документам, выполненным печатным способом;
ГОСТ 19.401-78 ЕСПД	Текст программы. Требования к содержанию и оформлению.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

IoT – Internet of Things;

ЛВС – Локальная вычислительная сеть;

ПК – Персональный компьютер;

БД – База данных;

ПО – Программное обеспечение.

ВВЕДЕНИЕ

Процесс развития информационных технологий не стоит на месте. На данный момент производство техники и оборудования различного назначения наращивает обороты. Все больше техники появляется в жилых помещениях, на предприятиях, образовательных учреждениях. С увеличением количества техники так же увеличиваются энергозатраты и появляются проблемы с ее управлением. Для комфорта использования обычно используют «умную» технику, основанную на технологии «Internet of Things».

Целью магистерской диссертационной работы является создание аппаратно-программного обеспечения системы SCHome и ее модулей по технологии «Internet of Things» для управления различной техникой и оборудованием.

Для достижения данной цели были поставлены и решены следующие задачи:

1) провести анализ технологий «Умный дом» и «Интернет Вещей»;

2) провести анализ угроз системам, основанным на технологии «Интернет Вещей»;

3) разработать прототип информационной системы демонстрирующий автоматизацию и удаленное управление различного оборудования.

Предмет исследования – информационная система (аппаратно-программный продукт) управления техникой и оборудованием путем удаленного доступа к системам «умной» техники через их интерфейсы, а также через модули разрабатываемой системы.

Научная новизна диссертационной работы заключается в разработке автоматизированной информационной системы, состоящей из модулей универсальной по отношении к управляемым данной системой устройствам.

Практическая значимость исследования заключается в разработке прототипа системы SCHome демонстрирующий автоматизацию и удаленное управление оборудованием и техникой Центра развития современных компетенций детей

«АмурТехноЦентр» «Дом научной коллаборации имени академика РАН М.Т. Луценко» (далее - ДНК).

Основные этапы исследования: исследование проводилось с 2018 по 2020 года в несколько этапов:

На первом этапе формулировалась тема исследования, проводился сбор информации по теме исследования из различных источников, осуществлялась формулировка гипотезы, постановка цели, задач.

Второй этап – в ходе данного этапа осуществлялся анализ типовых угроз системам, основанным на технологии «Интернета Вещей», а также проводился анализ средств и компонентов, представленных на IT рынке, осуществлялось написание и публикация научных статей по теме исследования.

Третий этап заключался в том, что осуществлялась разработка программного и аппаратного обеспечения системы SCHome, разработка прототипа для демонстрации работы системы и последующего внедрения в ДНК, тестирование прототипа, а также последующая оценка полученных результатов.

В первой главе рассматриваются технологии «Умный дом» и «Интернет Вещей», их достоинства и недостатки.

Во второй главе проводится анализ предметной области, анализ аппаратного и программного обеспечения решения поставленной задачи.

В третьей главе проводится анализ возможных типов атак системы, основанной на локальной вычислительной сети

В четвертой главе рассматриваются способы защиты локальной сети исходя из анализа угроз, представленных в третьей главе.

В пятой главе рассматривается модель жизненного цикла разрабатываемой системы, а также описание разрабатываемых модулей и прототипа системы SCHome.

1 РАЗВИТИЕ ТЕХНОЛОГИЙ «УМНЫЙ ДОМ» И «ИНТЕРНЕТ ВЕЩЕЙ»

1.1 Технология «Умный дом»

Человек всегда стремился сделать свою жизнь максимально комфортной, и появление системы «умный дом» (smart home) стало закономерным результатом этого желания. Если еще десять лет назад автоматизация жилого пространства была прерогативой исключительно обеспеченных людей, то сегодня она доступна практически всем жителям развитых стран благодаря широкому удешевлению информационных технологий. Вместе с тем на постсоветском пространстве такие системы только начинают внедряться в повседневную жизнь людей, и многие либо относятся к ним с недоверием, либо вообще о них ничего не знают.

Под понятием умный дом подразумевается автоматизация жилого пространства, благодаря которой все повседневные задачи (или их часть), связанные с проживанием, решаются без участия человека. Впервые идеи создания такой системы стали появляться еще в начале 20-века, одновременно с широкой электрификацией и появлением первых моделей привычных сегодня бытовых приборов – пылесосов, стиральных машин, холодильников, тостеров, утюгов с регулируемой температурой и т. д.

Первые системы управления «умными домами» были разработаны и реализованы еще в 50-60-х годах прошлого столетия. В 70-х в США был разработан стандарт управления домашними устройствами X10 – тогда же на рынок была выпущена относительно недорогая система автоматизации, которая получила довольно большое распространение. В последующие годы шло постепенное развитие этой технологии не только в США и Европе, но и в странах Азии (прежде всего Японии и Южной Корее). Особую роль в этом сыграло появление в и широкое распространение смартфонов и планшетов, с помощью которых контроль над домашними системами стал максимально комфортным.

Сегодня под этим термином подразумевается единая система автоматизированного управления бытовыми устройствами, организованная на уровне домашней локальной сети.

Основные функции «умного дома» сводятся к следующему:

– Поддержание максимального комфорта повседневной жизни человека за счет контроля характеристик жилого пространства (микроклимата, освещения) и автоматизации бытовых операций (уборки, стирки, мойки посуды и т. д.).

– Обеспечение безопасности жильцов и имущества путем предотвращения нештатных ситуаций и злонамеренных действий третьих лиц или своевременного предупреждения о них.

– Эффективное и экономное расходование ресурсов – прежде всего электроэнергии и воды – за счет их рационального использования.

Сегодня системы «умный дом» часто интегрируются в общую информационную сеть жилища. Для максимальной эффективности они объединяются с другими системами – видеонаблюдением, охранными и пожарными сигнализациями и т. д. В таком случае управление всеми компонентами осуществляется из единого центра.

В самом общем виде система домашней автоматизации состоит из трех основных категорий компонентов:

– Управляющие устройства. Контроллеры (хабы) осуществляют управление системой (прием и обработку сигналов, выдачу команд) в автоматическом режиме по задаваемым пользователем настройкам. В зависимости от конфигурации системы таких управляющих устройств может быть одно или несколько – во втором случае они также соединены друг с другом и подключены к общему терминалу, роль которого может выполнять специализированное оборудование или домашний ПК.

– Датчики. Назначение этих устройств – контроль над состоянием среды (жилого пространства). Они регистрируют изменения ее характеристик (например, температуры, влажности, наличие движения, давление в отопительной системе и т. д.) и подают соответствующие сигналы на контроллер, который после их обработки выдает управляющие команды, направленные на приведение параметров жилья к установленным или оповещение пользователя о возможной опасности.

– Исполнительные устройства. Наиболее многочисленная группа приборов, задача которых – выполнить определенное действие при подаче контроллером управляющей команды. К исполнительным устройствам относятся автоматические выключатели и розетки, оповещатели (сирены), запорные клапаны, кондиционеры, вентиляционные установки и т. д.

– Дополнительное оборудование. Для слаженной работы всей системы и ее отдельных элементов необходимо обеспечить их надежную коммуникацию друг с другом. С этой целью используются маршрутизаторы, автономные источники питания устройств, сетевые распределители, коммуникационные кабели, устройства беспроводной связи и т. д.

Система «умный дом» работает как единая локальная сеть. Все устройства подключены друг к другу по проводным и/или беспроводным (Wi-Fi, Bluetooth и т. д.) каналам. Большое внимание при этом уделяется шифрованию данных, предотвращающему взлом оборудования злоумышленниками, для чего производители используют общепринятые или специализированные протоколы безопасности. Для того, чтобы пользователь мог контролировать систему, находясь вне дома, она подключается к глобальной информационной сети Интернет.

Один из основных вопросов, связанных с внедрением данной технологии, заключается в ее стоимости. Однозначного ответа здесь нет – все зависит от масштаба и функционала конкретной системы управления домашним пространством. При желании, серьезно ограничив возможности «умного дома» лишь базовыми, можно уложиться в достаточно скромные 50-60 тысяч рублей (не считая затрат на установку и подключение). Более серьезные системы с широким набором функций будут стоить намного дороже – но и уровень комфорта они обеспечат куда больший. Впрочем, даже ограничившись климат-контролем и автоматическим освещением, можно серьезно повысить удобство собственного дома.

1.2 Технология «Интернет Вещей»

Термин IoT был придуман в 1999 году Кевином Эштоном, одним из трех основателей Центра автоматической идентификации Массачусетского универси-

тета. Существует несколько определений этого термина, и каждое из них недостаточно точное. Определение, предложенное компанией Gartner (той самой, которая придумала термин ERP): «Интернет вещей - это сеть физических объектов, имеющих встроенные технологии, позволяющие им взаимодействовать с внешней средой, передавать информацию о своем состоянии и получать данные извне». Составной частью Интернета вещей является Индустриальный (или Промышленный) интернет вещей (Industrial Internet of Things, IIoT). И уже появился новый термин: «Интернет всего» (Internet of Everything, IoE), который придет на смену Интернету вещей в недалеком будущем.

В 1990 году Джон Ромки, один из создателей протокола TCP/IP, подключил свой тостер к Интернету и заставил его включаться и выключаться дистанционно. Это устройство и стало первой в мире «интернет-вещью». В период с 2008 по 2009 год, по оценке аналитиков корпорации Cisco, количество устройств, подключённых ко Всемирной паутине, превысило численность населения Земли.

Современный Интернет состоит из тысяч корпоративных, научных, правительственных и домашних компьютерных сетей. Объединение сетей разной архитектуры и топологии осуществляется с помощью протокола IP. Каждому участнику Сети (или группе участников) присваивается IP-адрес, постоянный или временный (динамический).

Аналогичным образом Интернет вещей сегодня состоит из множества слабо связанных между собою сетей, каждая из которых решает свои задачи. Например, в офисном здании может быть развернуто сразу несколько сетей: для управления кондиционерами, системой отопления, освещением, безопасностью и т.д. Эти сети могут работать по разным стандартам, и объединение их в одну сеть представляет собою нетривиальную задачу. Кроме того, существующая (четвертая) версия протокола IP (IPv4) позволяет использовать всего лишь 4,22 миллиарда адресов, из-за чего возникла проблема их исчерпания. И хотя не каждому устройству, подключенному к Сети, необходим уникальный IP-адрес (но

все равно необходим уникальный идентификатор), в связи с бурным ростом Интернета вещей проблема нехватки адресов может стать ограничивающим фактором. Кардинально решить ее поможет шестая версия протокола, IPv6, которая обеспечит возможность использования каждым жителем Земли более 300 млн. IP-адресов.

Ожидается, что 2020 год в мире будет от 30 до 50 млрд. объединенных в сеть вещей, а возможности адресации протокола IPv6 позволят практически без ограничений идентифицировать в Сети любую вещь.

В основе Интернета вещей лежат следующие технологии:

Средства идентификации. Каждый объект физического мира, участвующий в Интернете вещей, пусть даже не подключенный к Сети, все равно должен иметь уникальный идентификатор. Для автоматической идентификации предметов могут использоваться различные уже существующие системы: радиочастотная, при использовании которой к каждому объекту прикрепляется радиочастотная метка, оптическая (штрих-коды, Data Matrix, QR-коды), инфракрасные метки и т.д. Но в обеспечение уникальности идентификаторов различных типов придется провести работу по их стандартизации.

Средства измерения. Задача средств измерения – обеспечить преобразование информации о внешней среде в данные, пригодные для передачи их средствам обработки. Это могут быть как отдельные датчики температуры, освещенности и т.п., так и сложные измерительные комплексы. Для достижения автономности средств измерения желательно обеспечить электропитание датчиков за счет средств альтернативной энергетики (солнечные батареи и т.п.), чтобы не тратить время и средства на подзарядку аккумуляторов или замену батарей.

Средства передачи данных. Для передачи данных может быть использована любая из существующих технологий. В случае применения беспроводных сетей особое внимание уделяют повышению надежности передачи данных. При использовании проводных сетей активно используют технологию передачи данных по линиям электропередачи, поскольку многие «вещи» (такие как торговые автоматы, банкоматы и т.п.) подключены к электросетям.

Средства обработки данных. Тридцать и более миллиардов устройств, которые, по прогнозам, будут подключены в 2020 году к Интернету, сгенерируют 44 миллиарда терабайтов данных. Это примерно в семь раз превышает количество оцифрованной информации во всем мире по состоянию на 2010-е годы.2 Поэтому в компании Microsoft полагают, что главная часть Интернета вещей — это не датчики и средства передачи данных, а облачные системы, обеспечивающие высокую пропускную способность и способные быстро реагировать на определенные ситуации (например, уметь по показаниям датчиков выяснять, что в доме уже пять минут никого нет, а входная дверь осталась открытой). Помогут справиться с огромными потоками информации также туманные вычисления, которые будут не конкурировать с облачными, а эффективно их дополнять.

Исполнительные устройства. Это устройства, способные преобразовывать цифровые электрические сигналы, поступающие от информационных сетей, в действия. Например, для того чтобы через смартфон можно было включить систему отопления в доме, она должна иметь соответствующее устройство. Исполнительные устройства зачастую конструктивно совмещаются с датчиками.

Предполагается, что к 2020 году Интернет вещей будет применяться в самых различных отраслях. Прежде всего это промышленность (см. статью Промышленный интернет вещей), транспорт (220 млн. подключенных автомобилей), умный дом, коммунальные службы (миллиард датчиков, существенное снижение потерь энергии), здравоохранение (646 млн. устройств, собирающих данные о здоровье людей), аграрный сектор (75 млн. датчиков для мониторинга состояния почвы). Кроме того, Интернет вещей будет применяться в торговле, логистике, общепите, гостиничном бизнесе, банковской системе, строительстве и в вооруженных силах (126 тыс. военных дронов и роботов).

Поскольку Интернет вещей — молодой и потенциально очень емкий рынок, многие крупные компании спешат занять эту нишу:

- Google обещает разработать голосовой интерфейс, благодаря которому домашняя утварь (например, холодильник) научится понимать естественную речь человека

- Intel анонсировала платформу Intel IoT Platform, предназначенную, как следует из названия, для Интернета вещей
- Apple предлагает платформу HomeKit, которая предназначена для управления домашней электроникой (бытовой техникой, освещением, сигнализацией, дверями гаража и т.д.)
- Microsoft адаптирует свои облачные сервисы Azure для Интернета вещей

Как это обычно бывает на молодых перспективных рынках, может начаться «война стандартов». Дабы избежать ее, уже сейчас прилагаются немалые усилия.

В частности, два общедоступных высокотехнологичных концерна из разряда крупнейших — AllSeen Alliance и Alljoyn от Qualcomm — объединили усилия с Open Interconnect Consortium (OIC) в рамках новой организации Open Connectivity Foundation (OCF).

С задачей совместимости на корпоративном уровне должен справиться стандарт OneM2M, которому следуют уже 230 компаний, в том числе такие известные, как Amazon, Cisco, Huawei, Intel, NEC, Qualcomm, Samsung и многие другие.

Эксперты считают, что «в настоящее время безопасной экосистемы Интернета вещей не существует». Из-за того, что во многих устройствах, подключенных к Интернету, не шифруется беспроводной трафик, не предусмотрены пароли достаточной сложности, а также из-за многих других факторов хакеры могут, например, включать и отключать чужие посудомоечные и стиральные машины, запирать хозяев в их собственном доме или даже наблюдать за их домашней жизнью с помощью, например, видеокамеры, установленной на роботе-пылесосе. Для повышения безопасности предлагается введение обязательной сертификации устройств, рассчитанных на подключение к Интернету, установка на них специальных унифицированных чипов и другие меры.

В отдаленной перспективе «умными» станут не только дома, но и города, и даже (некоторые) государства. Но на данном этапе развития технологий и общества Интернет вещей активно внедряется не в глобальных масштабах, а внутри компаний, занимающихся производством товаров, энергии, транспортными перевозками и т.п. — там, где за счет новых технологий ожидается повышение производительности и конкурентоспособности. Сложность масштабирования этого опыта обусловлена тем, что необходимо интегрировать в единое целое многие системы от разных поставщиков, а наладить их слаженную работу — задача сложная.

1.2.1 Достоинства технологии Интернета Вещей

Комфорт. Интернет вещей обеспечивает удобство человеческой жизни: кофе наливается сам собой, а автомобиль следит за положением дел на дороге. Отслеживание данных также повышает качество нашей жизни, ведь вы можете быть уверены, к примеру, что заказанные холодильником продукты – свежие.

Эффективность и безопасность. Использование технологий IoT минимизирует чрезвычайные ситуации и травмоопасные условия, что особенно актуально при использовании системы в промышленных и коммерческих целях. При этом объемы производства могут вырасти в несколько раз, ведь автоматизируется масса операций.

Помощь в принятии социальных, экономических и других решений. Будет легче принимать более эффективные решения, когда наши идеи станут подкрепляться собранной информацией. Эмпирические данные помогают определять причинно-следственные связи и предугадывать разные тенденции, что сводит к минимуму необходимость в ручном анализе данных для бизнеса.

Сокращение временных затрат и увеличение доходов. Концепция интернета вещей обеспечивает быстрые отклики устройств, тем самым сокращая время, необходимое для выполнения любой работы. При высокой эффективности бизнеса, ориентированного на данную технологию, доходы должны возрасти, так как появляется больше шансов для конкурентоспособности и реализации идей. Экономия времени находит отражение даже в частной жизни. Вам не

нужно больше мониторить информацию самостоятельно и, к примеру, ездить оплачивать счета или покупать продукты.

1.2.2 Недостатки технологии Интернета Вещей

Необходимость огромной предварительной подготовки. Нужно не только научить устройства идентифицировать и маркировать различные объекта, но и организовать общение устройств разной марки. В большинстве случаев гаджеты сегодня могут взаимодействовать только с продукцией того же производителя.

Из первого пункта вытекает проблема совместимости и интеграции данных. Сотни стандартов передачи и обработки данных ограничивают возможности взаимосвязи. Сейчас всё еще не установлено, должны ли устройства работать по одному стандарту или правила надо разрабатывать исходя из конкретного поставщика интернет-услуг.

Наиболее важным недостатком в отношении внедрения IoT является вопрос о конфиденциальности. Умные домашние устройства получают много данных о пользователе. Эта информация включает в себя личные графики, потребительские привычки, расписание приема лекарств и даже местоположение пользователя в любой момент времени. Если эти данные попадут не в те руки, людям может быть нанесен большой вред и ущерб. Поэтому шифрование – важный момент развития Интернета вещей.

Нужно также отметить, что данная концепция относится к сложным системам. И у них всегда есть вероятность отказа от работы или совершения ошибки. К примеру, устройство может случайно уведомить всех членов семьи об отсутствии продуктов в холодильнике или сломанной лампочке. Или неправильно воспримет сигналы принтера и закажет цветные чернила вместо черно-белых.

2 ПРОГРАММНОЕ, АППАРАТНОЕ И АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЗАДАЧИ

2.1 Анализ предметной области

2.1.1 Постановка задачи

Разработать программно-аппаратный комплекс (систему) для нужд Центра развития современных компетенций детей «АмурТехноЦентр» «Дом научной коллаборации имени академика РАН М.Т. Луценко» (далее - ДНК). Программное обеспечение должно реализовывать автоматическое управление (по расписанию) и удаленное управление оборудованием и техникой, размещенное в помещениях ДНК. В программном обеспечении предусмотреть ввод расписания работы модулей системы, его редактирования администратором и ответственными за расписание. Предусмотреть удаленный доступ к управлению системой и редактированию расписания. Также предусмотреть возможность масштабирование всей системы – возможность включения в состав системы новых модулей на основе реле, для запуска их по расписанию (к примеру, управление устройством обеззараживания воздуха), модулей-анализаторов, которые могут сами управлять другими модулями. Обеспечить вывод отчета по прошедшей неделе для отслеживания удаленного доступа к системе, а именно время, дата, имя пользователя, который редактировал расписание.

Удаленный доступ, т.е. ручное управление модулями должно осуществляться из приложения со смартфона, а также из браузеров устройств, размещенных в локальной сети ДНК.

Первый этапом должна стать разработка рабочего прототипа, для демонстрации работы системы. Прототип должен осуществлять функционал удаленного управления лампой дневного света, сетевыми розетками, актуаторами (электронными замками), анализ влажности почвы, влажности, температуры, относительную загрязненность воздуха внутри помещения, автополив растений в зависимости от внешних условий.

Второй этап – внедрение системы обладающей функционалом прототипа, и дальнейшее постепенное расширение этого функционала.

2.1.2 Назначение программного обеспечения

Разрабатываемое программное обеспечение предназначено для управления оборудованием и техникой ДНК. Основная задача программного обеспечения – автоматизация работы (по расписанию, по событию), удаленное управление системой. Удаленное управление осуществляется администратором системы. Вывод информации о доступе к системе осуществляется в виде таблиц и экранных форм.

2.1.3 Обзор существующего программного обеспечения решающего поставленную задачу

Для выполнения поставленных задач не существует отдельного аппаратно-программного обеспечения. На рынке представлены либо автономные модули, каждый из которых необходимо настраивать вручную, что не соответствует задачам масштабирования системы и удаленного управления, либо дорогостоящими системами умных домов, которые ориентированы на управление оборудованием частных домов, и обладающих функционалом, не требующимся ДНК. Тем не менее, были найдены варианты систем, которые выполняют максимальное число вышеперечисленных задач.

Сравнение различных систем «умный дом» представлено в таблице 1.

Таблица 1 – Сравнительные характеристики систем

	Достоинства	Недостатки	Стоимость базового набора
1	2	3	4
Ростелеком	личные данные, видеоматериалы и аналитика датчиков хранятся на платформе Ростелеком в зашифрованном виде; возможность покупки комплекта по сниженным ценам, с учетом акций и скидок пользователям Ростелеком; возможность работы с другими операторами.	отсутствие возможности подключения бытовых приборов; отсутствие возможности включения устройств от других производителей; завышенная цена базового комплекта.	12 тыс. рублей

Продолжение таблицы 1

1	2	3	4
Xiaomi	реализация множества полезных функций уже со стартовым набором элементов; совместимость со смартфонами на ОС Android и iOS; настроено все «из коробки».	китайские розетки не пригодны для европейских вилок, требуется приобретение переходника;	5 тыс. рублей
Ajax StarterKit	защищенный радиоканал; простота настройки и управления; быстрое оповещение; резервное питание контроллера.	исключительно охранные функции.	13 тыс. рублей
Gal SH	простота установки; доступная цена;	отсутствие возможности подключения бытовых приборов и «умной» ётехники;	5 тыс. рублей

2.2 Аппаратное обеспечение решения задачи.

2.2.1 Обоснование выбора архитектуры системы

В целях выполнения поставленной задачи была выбрана клиент-серверная структура программного обеспечения.

Одной из свойств ПО, указанных заказчиком, была масштабируемость. То есть управляемых устройств может быть несколько. Для соединения всех устройств потребуется единый узел, сервер, который будет реализовывать как протокол обмена информацией между пользователем и микроконтроллером, отвечающим за звонок, так хранение и обработку заданного пользователем расписания.

На рисунке 1 представлена схема клиент-серверного приложения, где под «основным ПО» понимается ПО, реализующим обработку расписания, протокол обмена данных с управляемым микроконтроллером, а также Web-интерфейс пользователя и администратора.

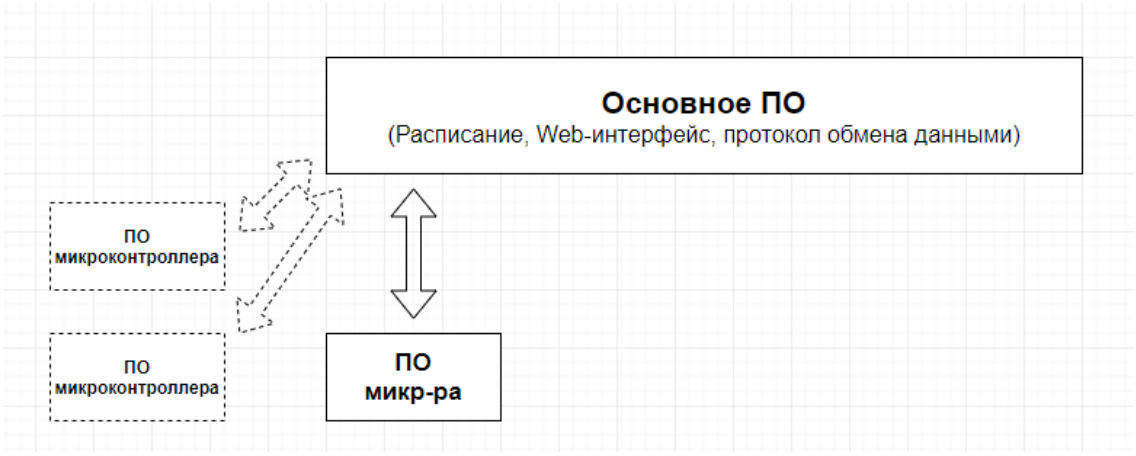


Рисунок 1 – Архитектурная схема клиент-серверного приложения.

На рисунке 2 изображена подобная схема, где часть ПО, а именно ответственная за передачу данных выделена в отдельный сервер.

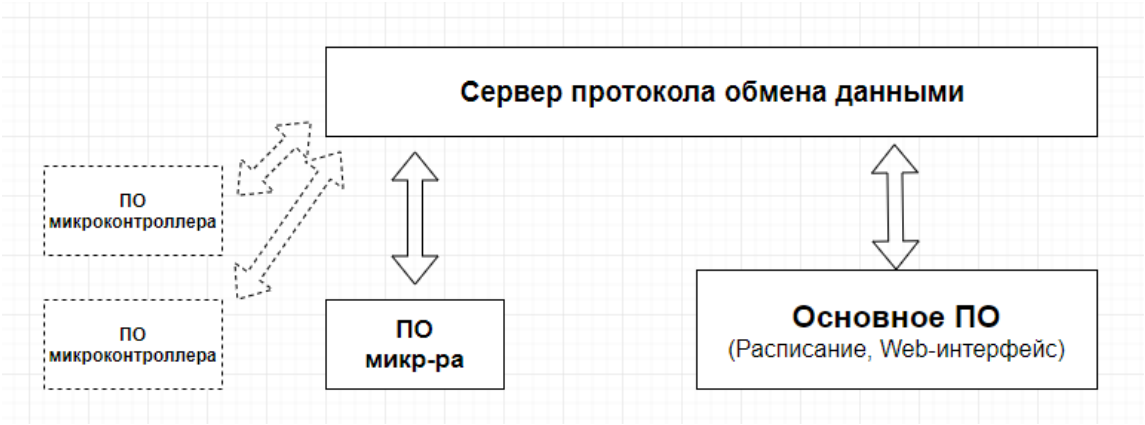


Рисунок 2 – Схема приложения с выделенным сервером протокола обмена данными

Для выполнения поставленных задач подходит вторая схема, так как она позволяет снизить затраты на разработку ПО. Существует множество готовых решений реализации протокола обмена данных между микроконтроллерами и пользователями, в том числе и свободно распространяемых и бесплатных.

2.1.2 Обоснование выбора аппаратных средств

Так как система должна соответствовать стандартам разработки по технологии IoT, обмен данными между модулями должен происходить через беспроводной канал связи.

Проанализировав информацию в интернете, мы обнаружили подходяще типы беспроводной связи на основе 4 модулей (Рис. 3):

LoRaWAN – Беспроводной модуль предачи данных по радиоканалу частотой 868МГц. Особенностью является очень большая дальность передачи (свыше 10 км) за счет низкой скорости передачи данных (250-5470 бит/сек), высокая цена.

HC-06 - Беспроводной модуль для приема/передачи данных в Arduino проектах по протоколу Bluetooth. Радиус действия до 10 м, Скорость передачи данных составляет 1200–1382400 бод, рекомендуемое напряжение 6В.

nRF24L01+ - Модуль NRF24L01 позволяет передавать данные через радиоканал частотой 2,4ГГц, скоростью 250–2000 Кбит/сек. Особенностью данного модуля является его низкая цена, широкий диапазон каналов, дальность приема сигнала до 100м по прямой видимости, наличие «спящего» режима, незащищенное соединение «точка-точка» со вторым аналогичным модулем.

ESP8266– Беспроводной модуль передачи данных по WiFi (протокол 802.11 b/g/n) частотой 2,4ГГц. Имеет разные режимы работы (в том числе и низкого энергопотребления), высокую скорость передачи данных, низкая цена, WPA/WPA2 шифрование.

Из всех вышеперечисленных модулей наиболее подходит модуль ESP8266 (далее ESP). Во-первых, ESP обладает низкой ценой. Во-вторых, канал передачи данных по радиоканалу шифруется на уровне протокола WiFi. В-третьих, в отличие от остальных модулей, ему не требуется приемное устройство, так как он подключается напрямую к существующей локальной сети WiFi или может выступать в роли точки доступа WiFi.

На данный момент на рынке представлено несколько микроконтроллеров с Wi-Fi модулем в своем составе. Подавляющее большинство использует именно ESP8266 и ESP-32.

ESP82266 является недорогим WiFi модулем и специально разрабатывался для IoT устройств (рис. 4). Сейчас можно хорошо распространены версии модуля ESP-12E построенного на базе чипа ESP82266. Программное обеспечение модуля хранится в микросхеме flash-памяти формата SOP-210mil. При каждом включении питания данное программное обеспечение автоматически помещается в чип ESP8266. Объем flash-памяти - 4 МБ и этого объема вполне достаточно, чтобы

хранить полноценные программные приложения, управляемые обширным набором текстовых AT-команд (Attention - команды), и для реализации сложных алгоритмов шифрования и аутентификации на основе сертификатов безопасности WPA2-Enterprise. При этом процессорное ядро обладает достаточной мощностью для работы сложных пользовательских приложений цифровой сигнальной обработки. Модуль ESP-12E снабжен встроенным кварцевым резонатором, полностью обеспечивающим работу процессорного ядра и периферии при подаче питания. 22 вывода, расположенных вдоль двух противоположных краев модуля, позволяют модулю ESP-12E взаимодействовать с внешними устройствами. Модуль имеет возможность работы с различными интерфейсами: 2-контактный UART (RX и TX) для обмена данными и AT-командами, Вывод GND (земля), Питание (VCC), Chip Enable (CH_PD) для управления питанием модуля с внешнего микроконтроллера, вывод Reset для принудительной перезагрузки модуля, 17 GPIO (в том числе и выводы интерфейса SPI) и АЦП (1 вывод). В модуле ESP-12E предусмотрена встроенная антенна типа PCB с коэффициентом усиления 3dBi, улучшенной производительности. Программный код для модуля разрабатывается в среде разработки Arduino IDE и с помощью встроенного в IDE загрузчика загружается во flash-память.

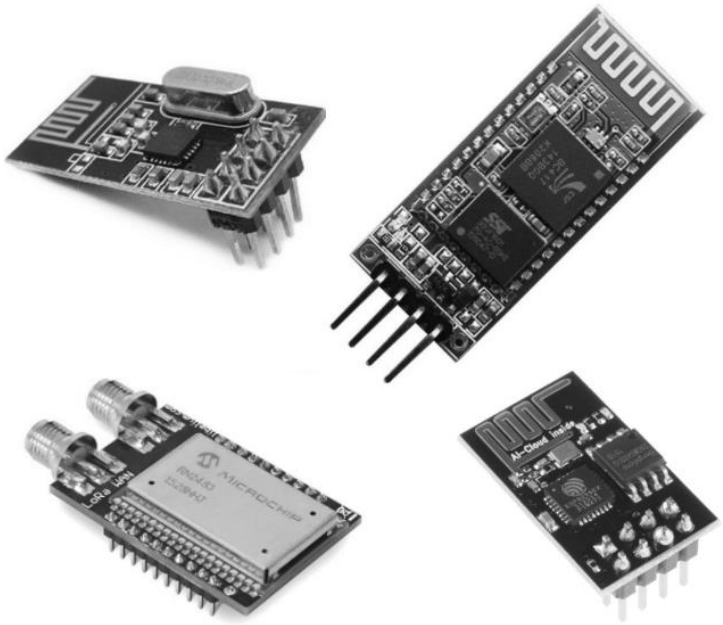


Рисунок 3 – Беспроводные модули nRF24L01+, HC-06, LoRaWAN module,

ESP32 является «старшим братом» ESP8266, так как разработан той же фирмой Espressif и имеет улучшенные характеристики (рис. 5).. Главным достоинством является наличие 15 контактов АЦП, что позволяет считывать показания с большего количества аналоговых датчиков.

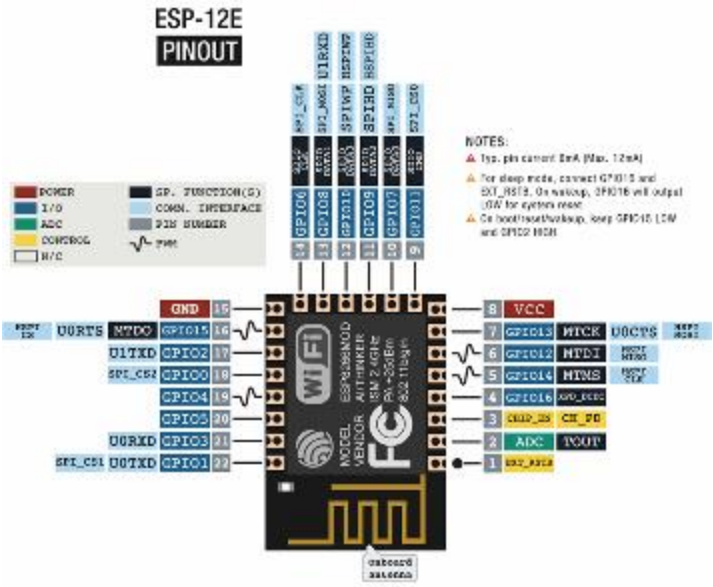


Рисунок 4 – Распиновка контактов ESP8266

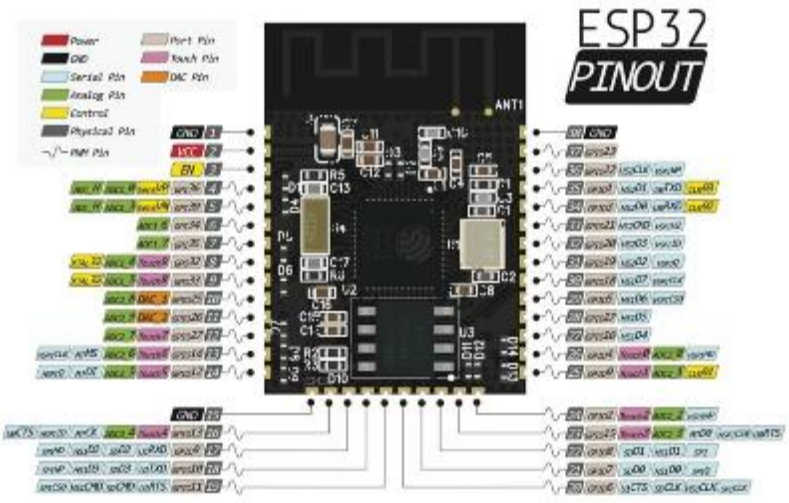


Рисунок 5 – Распиновка контактов ESP32

Важную роль при выборе аппаратного обеспечения играет цена и размеры. ESP32 имеет цену на 30-40% большую, чем ESP8266, но это оправдано лучшими характеристиками, а так же размерами, что может быть критичным показателем при встраивании модуля.

Для быстрой разработки прототипа были выбраны не просто модули ESP82266 и ESP32, но платы разработки (NodeMCU и ESP32DEV), основанные

на них, и имеющие в своем составе USB-контроллер и разъемы для подключения к ПК. Данные платы позволяют быстро подключать модули и перепрошивать их.

2.3 Программное обеспечение решения задачи

Для реализации принципа технологии IoT необходима сеть. Для создания системы управления оборудованием в отдельно взятом помещении наилучшим способом подойдет локальная вычислительная сеть, развернутая при помощи WiFi.

Работа системы SCHome заключатся в обмене данных между модулями по протоколу Blynk. Данный протокол реализуется установленной на микрокомпьютере локальным сервером Blynk и клиентской частью на микроконтроллерах и смартфоне.

2.3.1 Платформа Blynk

Платформа Blynk состоит из трех основных компонентов:

- Blynk App – Android/iOS приложение, позволяет создавать интерфейсы для любых проектов, используя различные виджеты.
- Blynk Server – отвечает за все коммуникации между смартфоном и аппаратным обеспечением. Имеется возможность использовать облачный сервер условно-бесплатно (Blynk Cloud) или запустить свой частный сервер Blynk локально (Blynk Server).
- Blynk Libraries – Библиотеки для всех популярных аппаратных платформ, которые обеспечивают связь с сервером и обрабатывают все входящие и исходящие команды.

Blynk Server - это Java-сервер с открытым исходным кодом Netty, отвечающий за пересылку сообщений между мобильным приложением Blynk и различными платами микроконтроллеров (например, Arduino, Raspberry Pi. и т.д.). На рисунке 6 изображена схема платформы Blynk.

Когда оборудование подключается к Blynk Cloud, оно открывает либо keep-alive SSL/TLS соединение на порту 443 (9443 для локальных серверов), либо keep-alive TCP/IP соединение на порту 8080. Blynk App открывает взаимное соединение SSL/TLS с Blynk Cloud на порту 443 (9443 для локальных серверов). Blynk Cloud

(или Blynk Server) отвечает за пересылку сообщений между оборудованием и приложением. В обоих (App и аппаратном) соединениях Blynk использует собственный двоичный протокол, описанный ниже.

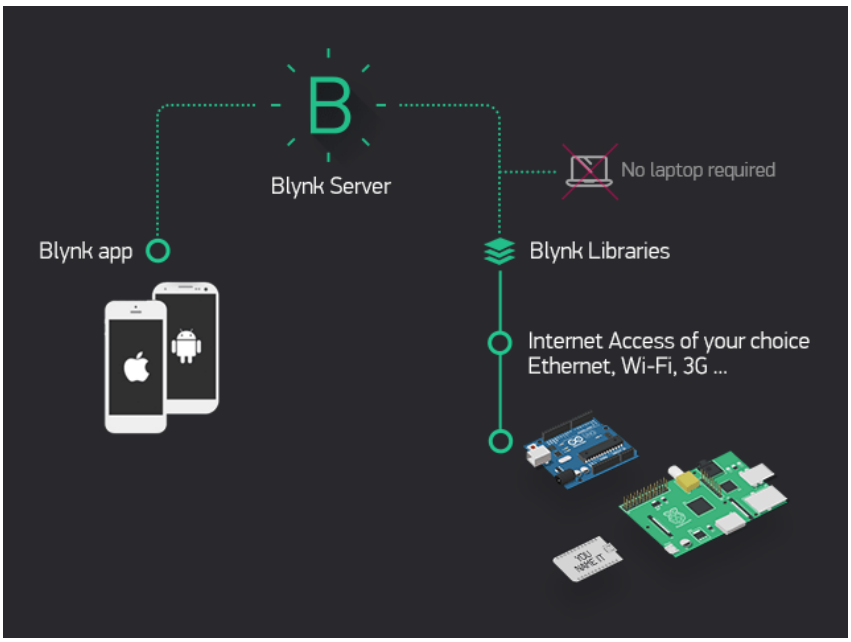


Рисунок 6 – Схема платформы Blynk

По протоколу для аппаратной части Blynk передает бинарные сообщения между сервером и оборудованием со следующей структурой, представленной на рисунке 7.

Команда	**Id сообщения**	**Длина/Статус**	**Содержание**
1 байт	2 байта	2 байта	Переменная

Рисунок 7 – Структура сообщения, между микроконтроллером и Blynk Server

По протоколу на стороне мобильного приложения Blynk передает бинарные сообщения между сервером и мобильным приложением со следующей структурой, показанной на рисунке 8.

Команда	**Id сообщения**	**Длина/Статус**	**Содержание**
1 байт	2 байта	4 байта	Переменная

Рисунок 8 – Структура сообщения, между Blynk App и Blynk Server

По протоколу web-сокетов (внешний API) Blynk передает бинарные сообщения между сервером и websockets (для web) со следующей структурой, представленной на рисунке 9.

Заголовок web-сокета	**Команда**	**Id сообщения**	**Содержание**
	1 байт	2 байта	Переменная

Рисунок 9 – Структура сообщения API

Когда код команды равен 0, то структура сообщения следующая, представленная на рисунке 10.

Заголовок web-сокета	**Команда**	**Id сообщения**	**Код ответа**
	1 байт	2 байта	4 байта

Рисунок 10 – Структура сообщения API

Возможные варианты ответов представлены на рисунке 11.

```
public static final int OK = 200;
public static final int QUOTA_LIMIT = 1;
public static final int ILLEGAL_COMMAND = 2;
public static final int USER_NOT_REGISTERED = 3;
public static final int USER_ALREADY_REGISTERED = 4;
public static final int USER_NOT_AUTHENTICATED = 5;
public static final int NOT_ALLOWED = 6;
public static final int DEVICE_NOT_IN_NETWORK = 7;
public static final int NO_ACTIVE_DASHBOARD = 8;
public static final int INVALID_TOKEN = 9;
public static final int ILLEGAL_COMMAND_BODY = 11;

public static final int NOTIFICATION_INVALID_BODY = 13;
public static final int NOTIFICATION_NOT_AUTHORIZED = 14;
public static final int NOTIFICATION_ERROR = 15;

public static final int NO_DATA = 17;
public static final int SERVER_ERROR = 19;
public static final int ENERGY_LIMIT = 21;
public static final int FACEBOOK_USER_LOGIN_WITH_PASS = 22;
```

Рисунок 11 – Коды ответов

Основные особенности платформы Blynk:

1) Одинаковый API и UI для всех поддерживаемых устройств и оборудования

2) Подключение к облаку с помощью:
    a) WiFi
    b) Bluetooth и BLE
    c) Ethernet
    d) USB (Serial)
    e) GSM
3) Набор простых в использовании виджетов
4) Прямая манипуляция пинами микроконтроллеров без записи кода
5) Простота интеграции и добавления новых функциональных возможностей с помощью виртуальных пинов
6) Мониторинг истории данных с помощью виджета Super Chart
7) Связь между устройствами с помощью Bridge Widget
8) Отправка электронных писем, твитов, push-уведомлений и т. д.

2.3.2 Обоснование выбора языка программирования

На сегодняшний день, программирование микроконтроллеров осуществляется при помощи IDE. IDE – это интегрированная среда разработки, комплекс программных средств, используемый программистами для разработки ПО.

Среда разработки может включать в себя:

- текстовый редактор,
- транслятор (компилятор и/или интерпретатор),
- средства автоматизации сборки,
- отладчик.

Для разработки ПО для микроконтроллеров ESP8266 и ESP32 была выбрана Arduino IDE.

3 АНАЛИЗ ВОЗМОЖНЫХ ТИПОВ АТАК И МОДЕЛИ НАРУШИТЕЛЯ, ОСУЩЕСТВЛЯЮЩЕГО АТАКИ НА СЕТЬ СИСТЕМЫ SCHOME

3.1 Анализ сетевого трафика

Анализ сетевого трафика – это изучение логики работы локальной вычислительной сети (ЛВС), а также получение событий, происходящих в системе, в тот момент, когда эти события происходят, и команд, передаваемых участниками системы друг другу. Изучение происходит после перехвата обменных пакетов на канальном уровне.

Кроме того, анализ сетевого трафика позволяет перехватывать потоки информации, которыми обмениваются участники локальной сети. Таким образом, в случае успешной атаки злоумышленник получает на удаленный объект несанкционированный доступ ко всем данным, которые передаются между двумя сетевыми объектами. Однако злоумышленник не имеет возможности изменять трафик. Информация, которая может быть перехвачена злоумышленником, например, может быть незашифрованным именем пользователя и паролем или ключом идентификации микроконтроллера, передаваемым по сети.

3.2 Подмена доверенного объекта или субъекта ЛВС

Одной из главных проблем безопасности локальной сети является идентификация и аутентификация, а точнее их недостаточность, удаленных объектов. Необходимо выполнить однозначную идентификацию сообщений, передаваемых между объектами взаимодействия.

Каждый объект в локальной сети имеет свой уникальный сетевой адрес (на канальном уровне модели OSI это аппаратный адрес сетевого адаптера; на сетевом уровне адрес определяется в зависимости от используемого протокола сетевого уровня (например, IP-адрес). Однако не рекомендуется использовать сетевой адрес для идентификации объектов сети, так как сетевой адрес очень легко подделать. Но возможно использовать его в сочетании с другими способами идентификации объекта.

Довольно часто для служебных сообщений в локальную сеть передаются не требующие подтверждения одиночные сообщения, при этом виртуальное соединение не устанавливается. В этом случае злоумышленник отправляет свое сообщение (служебное сообщение) от имени устройства управления сетью (роутера, маршрутизатора и т. д.), тем самым осуществляя несанкционированное управление объектом системы. Такие действия могут привести к серьезным проблемам с безопасностью объекта и локальной сети в целом.

3.3. Ложный объект ЛВС

Если в ЛВС не решены описанных выше проблемы, а именно идентификации управляющих устройств (например, маршрутизаторов, микроконтроллеров), то такая ЛВС будет подвергнута следующей атаке, которая заключается в изменении маршрутизации и введении в систему ложного объекта. Кроме того, при использовании алгоритмов удаленного поиска для взаимодействия с объектами, существует высокий риск несанкционированного внедрения ложного объекта в систему.

3.3.1 Внедрение в ЛВС ложного объекта путём навязывания ложного маршрута

Сегодня глобальные сети представляют собой совокупность сегментов сети, соединенных между собой (через узлы сети), и данные передаются от источника к получателю по маршруту, представляющему собой последовательность узлов сети. Каждый маршрутизатор имеет таблицу маршрутизации, в которой хранится оптимальный маршрут до каждого пункта назначения. Таблицы маршрутизации также существуют для любых хостов в глобальной сети. Для реализации оптимальной и эффективной маршрутизации в локальной сети существуют специальные протоколы управления для обмена информацией между маршрутизаторами, такие как OSPF (Open Shortest Path First), RIP (Routing Internet Protocol). ICMP (Internet Control Message Protocol) также используется для уведомления хостов о новом маршруте, а SNMP (Simple Network Management Protocol) - для удаленного управления маршрутизаторами. Все эти

протоколы являются протоколами сетевого управления, что, в свою очередь, означает удаленные изменения маршрутизации в Интернете.

При этом типе атаки злоумышленник пытается достичь определенной цели-изменить исходную маршрутизацию на объекте локальной сети до состояния, при котором измененный маршрут проходил через объект злоумышленника (ложный объект).

В одном из этих способов злоумышленник отправляет специальные служебные сообщения по сети для изменения маршрутизации. Эти сообщения определяются протоколами сетевого управления и отправляются от имени устройств сетевого управления (таких как маршрутизатор). В случае успеха злоумышленник получает полный контроль над потоком данных, передаваемых между двумя объектами локальной сети, и это открывает дополнительные возможности для приема, анализа и передачи сообщений, полученных от объектов локальной сети.

Этот тип атаки также включает в себя пример развертывания поддельной точки доступа Wi-Fi. Точка имеет то же имя, что и реальная, и объект сети может незаметно для пользователя подключиться к ней, «попав в руки» злоумышленника.

3.3.2 Использование ложного объекта для организации удалённой атаки на ЛВС

После перехвата потока данных ложным объектом, эта информация может быть использована злоумышленником в разных целях. Информация может сохраняться на ложном объекте ЛВС, а затем передана злоумышленнику, дла дальнейшей селекции (извлечении паролей, токенов, конфиденциальной информации и т.п.). Так же ложный объект может изменять информацию или подменять её полностью. Кроме того, ложный объект может вызывать отказы в обслуживании. Если система не предусматривает использование средств аутентификации объекта системы, то ложный объект может отправлять бесконечное колличество анонимных запросов на другой объект сети, что может привести к отказу атакованного объекта или нарушению работоспособности системы в целом (DoS).

Отказ в обслуживании (DoS) - злоумышленник отправляет неверный запрос атакуемому объекту. В этом случае процедура обработки запроса может быть зациклена и система может зависнуть.

4 ЗАЩИТА ЛОКАЛЬНОЙ СЕТИ СИСТЕМЫ

Обеспечение информационной безопасности локальной сети - достаточно сложная задача. Это связано с тем, что существует множество различных опасностей, которые угрожают информации, размещённой в локальной сети. Поэтому очень легко упустить любую из угроз при разработке комплекса защитных мер. Это, в свою очередь, приводит к высокому риску нанесения ущерба пользователям и владельцу системы.

4.1 Главные цели сетевой безопасности

На данный момент эксперты выделили три основные цели при построении системы сетевой безопасности. Они включают в себя конфиденциальность, целостность и доступность данных. Кроме того, при разработке проекта определяются дополнительные задачи. Однако они не будут рассматриваться в рамках данной выпускной квалификационной работы, в отличие от основных целей, обязательных для каждого проекта защиты локальной сети.

Первая цель - конфиденциальность информации. Сегодня многие люди выдвигают эту задачу на первый план, так как они предрасположены переоценивать угрозу, исходящую от хакеров. Это мнение в корне неверно, так как не соответствует статистике. Исходя из последних исследований, немного больше половины случаев потери важных данных происходит в результате сбоев в системе электроснабжения. Ну, а большинство оставшихся инцидентов исходят от различных технических сбоев. Если же информация и теряется в результате действий хакеров и других злоумышленников, то на них приходится небольшая доля всех инцидентов.

Целостность данных означает полное сохранение информации в её первоначальном виде. Информация должна иметь гарантию того, что она не будет уничтожена в результате технических сбоев или действий хакеров. Данные не должны быть изменены в случае мошенничества или какой-либо неисправности.

Наиболее сложной задачей в сетевой безопасности является обеспечение целостности информации, так как необходимо учитывать достаточно большое количество различных угроз.

Третья цель любой системы сетевой безопасности - доступность данных. Каждый пользователь должен иметь доступ в любой момент времени к той информации, которая ему необходима для работы. Первая задача для достижения этой цели - это обеспечение стабильной работы аппаратных средств: серверов, принтеров, рабочих станций и т. п. Вторая - разделение доступа к информации между различными группами пользователей.

Доступность данных является третьей целью любой системы сетевой безопасности. Это означает, что каждый пользователь в любой момент времени (не считая особых случаев, таких как тех. обслуживание и т. д.) имеет доступ ко всей необходимой информации для своей работы. Эта цель обычно делится на две задачи. Первая - обеспечение стабильной работы оборудования: серверов, принтеров, рабочих станций и т. п. Вторая - разделение доступа к информации между различными группами пользователей.

4.2 Анализ методов и средств защиты информации, применяемых в локальных сетях

Для того чтобы обеспечить надёжную защиту ЛВС, в системе информационной безопасности должны быть реализованы самые прогрессивные и перспективные технологии информационной защиты. К ним относятся:

- технологии криптографической защиты данных;
- технологии аутентификации;
- технологии межсетевых экранов;
- технологии виртуальных защищённых каналов и сетей VPN;
- технологии токенов (smart-карт, touch-memory, RFID- и NFC-карт и т. п.);
- технологии обнаружения вторжений (Intrusion Detection);
- технологии защиты от вирусов (программы антивирусы);

– технологии централизованного управления системой информационной безопасности.

Метод защиты от вирусов с использованием программ-антивирусов особенно важен касательно сервера системы Blynk. В операционной системе Raspbian (Debian-подобная), из которой запускается сервер должна присутствовать антивирусная программа.

4.3 Способы защиты информации

Способы защиты информации в локальной сети делятся на несколько типов, показанных на рис. 12:

1. Препятствие - физический барьер на пути злоумышленника к защищаемой информации (к территории и помещениям с оборудованием).

2. Управление доступом - защита информации, при которой выполняется урегулирование использования ресурсов системы (программных, технических средств и т.п.).

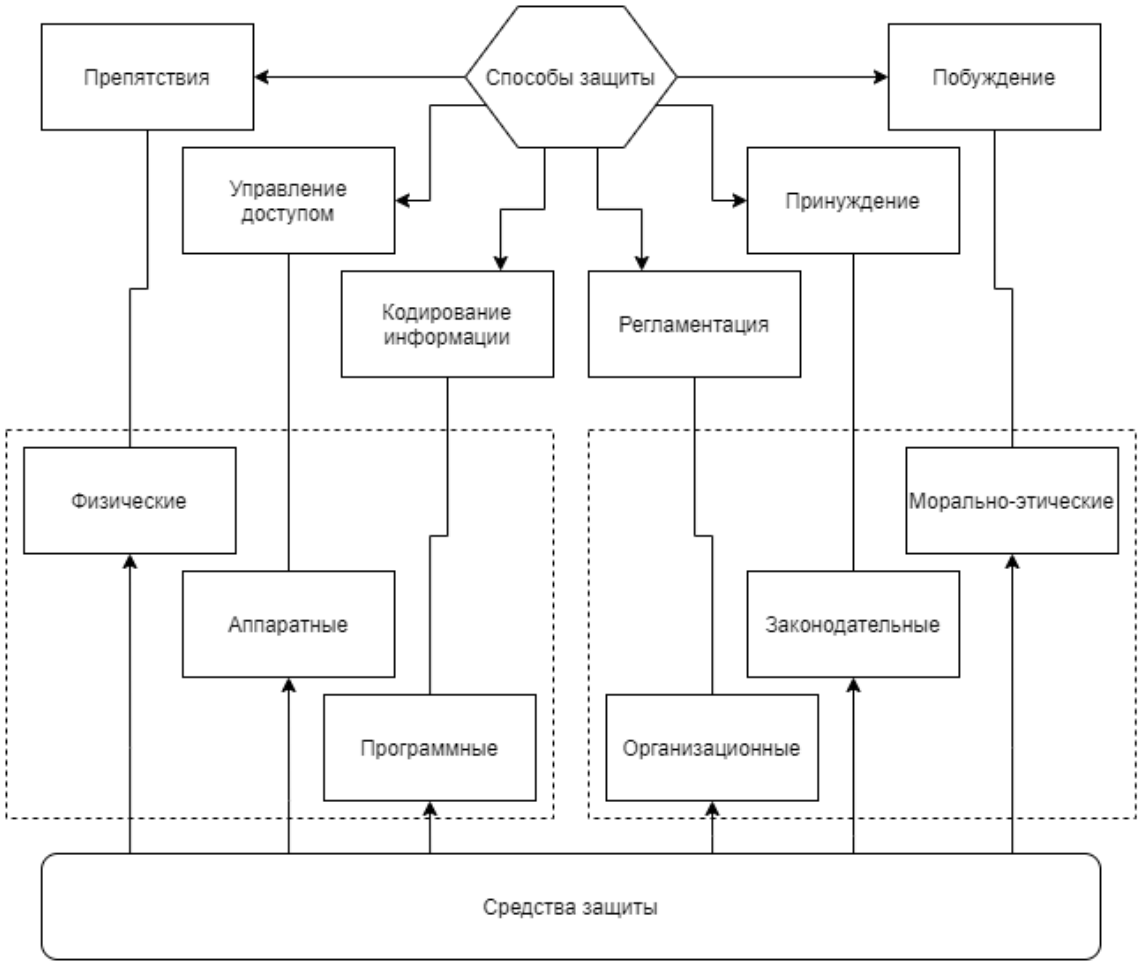


Рисунок 12 - Способы защиты информации в ЛВС

3. Кодирование информации - информация в ЛВС передаётся в закодированном виде, то есть каждый бит передаваемой информации соответствует своему собственному набору уровней электрических сигналов.

4. Регламентация - разработка и внедрение в процессе эксплуатации ЛВС комплекса мероприятий, создающих условия для обработки и хранения информации в ЛВС, приводящих к снижению возможности несанкционированного доступа к ней. Необходимо строго регламентировать структуру ЛВС (архитектуру зданий, размещение технических средств, оснащение помещений), организацию и работу всего персонала по обработке информации, для эффективной защиты.

5. Принуждение - персонал и пользователи локальной сети следуют правилам обработки и использования защищённой информации. В случае нарушения правил наступает материальная, административная или уголовная ответственность.

Все вышеперечисленные методы защиты информации реализуются с использованием различных средств защиты. Средства защиты подразделяются на программные, технические, законодательные, организационные и морально-этические средства.

Организационные средства защиты - организационно-правовые меры по обеспечению защиты информации, осуществляемые в ходе разработки, внедрения и эксплуатации ЛВС. Организационные мероприятия охватывают все структурные элементы ЛВС на всех этапах: строительство помещений, проектирование систем, монтаж и наладка оборудования, испытания и проверки, эксплуатация.

Законодательные средства защиты - правовые акты страны, регулирующие правила использования и обработки информации ограниченного доступа и устанавливающие меры ответственности за нарушение этих правил.

Морально-этические средства защиты - различные нормы, которые традиционно развивались или развиваются по мере распространения компьютеров в стране или обществе. Эти нормы обязательны, в отличие от законодательных мер, но если человек их не соблюдает, то он, как правило, теряет свой авторитет и престиж.

4.4 Идентификация и аутентификация

Идентификация и аутентификация считаются основой программно-аппаратных средств обеспечения безопасности. Идентификация и аутентификация могут быть представлены как «проходная» на предприятии, так как они почти всегда выполняются первыми.

Идентификация - это процесс присвоения имени субъекту (пользователю или процессу, действующему от имени этого пользователя). Аутентификация - проверка, с помощью которой вторая сторона проверяет подлинность предмета. Обычно субъект подтверждает свою подлинность, предъявляя одну из сущностей:

- "что-то, что он знает" - личный идентификационный номер, пароль, ключ и т. д.;

- "что-то, чем он владеет" - личная карточка или любое другое устройство аналогичного назначения;

- "что-то, что является частью его самого" - его биометрические характеристики, такие как отпечатки пальцев, голос и т.п.

Однако не всегда удаётся выполнить надёжную идентификацию и аутентификацию по ряду причин.

- существует вероятность того, что любые объекты аутентификации могут быть скомпрометированы, украдены или подделаны;
- система основана на полученной информации, источник которой неизвестен;
- чем выше надёжность средства защиты, тем выше и его стоимость;
- существует обратная зависимость между надёжностью аутентификации и удобством пользователя (или системного администратора).

Необходимо найти компромисс между доступностью, надёжностью и простотой использования и администрирования. Как правило, этот компромисс достигается с помощью первых двух основных механизмов аутентификации, перечисленных выше.

Пароли являются наиболее распространённым средством аутентификации. Система сравнивает введённый пароль и пароль, заранее определённый для конкретного пользователя; если пароли совпадают, проверка считается пройдённой, и субъект доказал свою подлинность. Но в последнее время секретные криптографические ключи становятся все более популярными - они обеспечивают наибольшую эффективность, часто являясь одновременно и идентификационной, и аутентификационной информацией.

Аутентификация паролём проста и привычна пользователю, так как она уже давно встроена в операционные системы и сервисы. На данный момент пароли распространены среди многих организаций, так как они обеспечивают приемлемый уровень безопасности для этих организаций. Однако это самый слабый инструмент аутентификации. Надежность паролей зависит от способности человека запоминать их и хранить в тайне. Тем не менее пароль может быть подсмотрен, пароль может быть подобран брутфорсом (перебор паролей, символов). Файл с паролями может быть зашифрован, но открытым для чтения, и тогда он может быть передан на компьютер злоумышленнику, чтобы подобрать пароль полным перебором.

Пароли также уязвимы для электронного перехвата - это самый фундаментальный недостаток, который не может быть компенсирован обучением пользователей или улучшением администрирования. Использование криптографии является единственным выходом при передаче данных по линиям связи.

Однако существует ряд действий, повышающих надёжность парольных систем. Такие как наложение технических ограничений (длина пароля больше 8 знаков, содержит буквы, цифры, спецсимволы), управление сроком действия паролём, ограничение числа неудачных попыток входа, обучение и воспитание пользователей.

В конкретных организациях с высокими требованиями к безопасности используются устройства для контроля биометрических характеристик. Такие устройства сложны и имеют высокую цену.

Администрирование службы идентификации и аутентификации - важная, но в то же время сложная задача. Хороший администратор постоянно поддерживает конфиденциальность, целостность и доступность информации, хранящейся в локальной сети, что довольно сложно, поскольку локальная сеть часто представляет собой разнородную сетевую среду. Важно отметить, что наряду с автоматизацией целесообразно максимально использовать централизацию информации, используя средства централизованного администрирования или выделенные серверы аутентификации. Сетевые сервисы, предоставляемые некоторыми операционными системами, могут служить основой для централизации данных. Централизация также может облегчить работу пользователей, так как позволяет реализовать концепцию единого входа в локальную сеть. Пользователь проходит проверку подлинности один раз и в пределах своих полномочий получает доступ ко всем сетевым ресурсам.

4.5 Управление доступом

Средства управления доступом - это средства, которые определяют и управляют действиями, которые пользователи и процессы могут выполнять с информацией и ресурсами в системе. Логическое управление доступом является основным механизмом многопользовательских систем. Он обеспечивает целостность, конфиденциальность объектов и их доступность, запрещая обслуживание неавторизованных пользователей. Логическое управление доступом определяет набор допустимых операций для каждой пары (субъект-объект), а также контролирует выполнение установленного в системе порядка.

Права доступа контролируются различными частями программной среды-системой управления базами данных (СУБД), ядром операционной системы, промежуточным программным обеспечением, дополнительными средствами защиты и т. д.

В момент принятия решения о предоставлении доступа анализируется следующая информация:

- идентификатор субъекта;
- атрибуты субъекта;

- время действия;
- место действия;
- внутренние ограничения сервиса.

Часто над средствами логического управления доступом устанавливается ограничивающий интерфейс, который делает невозможным выполнение пользователем несанкционированных действий, исключая из списка видимых ему объектов те объекты, к которым у него нет доступа.

5 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ ОБОРУДОВАНИЕМ И ТЕХНИКОЙ SCHOME

Система является комплексом аппаратно-программных средств и состоит из модулей, разделенных физически и обладающих различным функционалом.

Для полноценной работы системы потребуется локальная сеть WiFi, т.е. в помещении, где внедряется система должен присутствовать роутер.

5.1 Модели жизненного цикла программного обеспечения

Стандарт ГОСТ Р ИСО/МЭК 12207-2010 предназначен для представления определенной совокупности процессов, облегчающих связи между приобретающими сторонами, поставщиками и другими правообладателями в течение жизненного цикла программных продуктов.

Модель жизненного цикла программного обеспечения – структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

Модель жизненного цикла программного обеспечения – это структура, содержащая процессы, действия и задачи, выполняемые в процессе разработки, использования и обслуживания программного продукта.

Эти модели можно разделить на 3 основные группы:

- современные технологии быстрой разработки.
- с учётом специфики задачи;
- инженерный подход;

В соответствии со стандартом различают модели:

- каскадная модель жизненного цикла программного обеспечения;
- итерационная модель;
- спиральная модель жизненного цикла программного обеспечения.

5.1.1 Описание процессов жизненного цикла спиральной модели

Спиральная модель - это процесс разработки программного обеспечения, который сочетает в себе как проектирование, так и пошаговое прототипирование, чтобы объединить преимущества восходящей и нисходящей концепции разработки программного обеспечения (Рисунок 13).

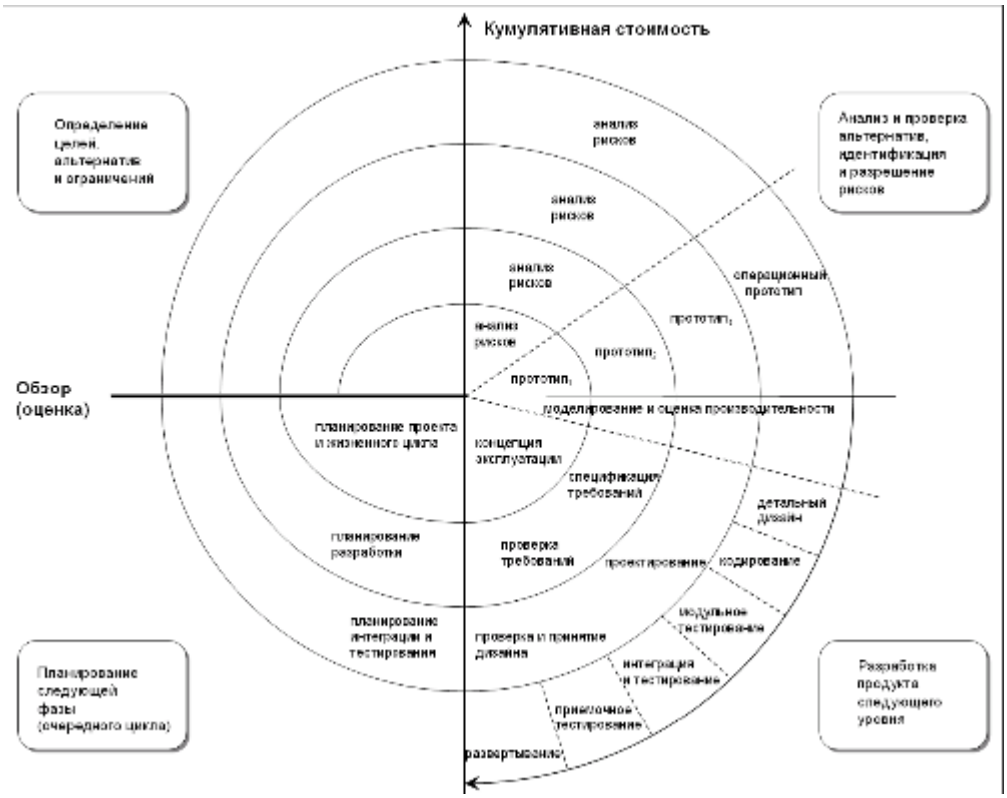


Рисунок 13 – Спиральная модель жизненного цикла

Жизненный цикл на каждом витке спирали – могут применяться разные модели процесса разработки программного обеспечения. В итоге на выходе получается готовый продукт. Модель сочетает в себе возможности модели прототипирования и каскадной модели. Итерационная разработка отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет перейти к следующему этапу, не дожидаясь полного завершения работ на текущем этапе. Основная задача заключается в том, чтобы как можно быстрее предоставить пользователям системы работоспособный продукт, активизировав тем самым процесс уточнения и дополнения требований.

5.1.2 Обоснование выбора модели жизненного цикла для разрабатываемого программного средства

Преимущества спиральной модели:

- позволяет быстро показать пользователям системы работоспособный продукт, тем самым активизируя процесс обновления и актуализации требований;

- позволяет изменять требования при разработке ПО, что характерно для большинства разработок, в том числе и стандартных;

- модель обеспечивает возможность гибкого проектирования, поскольку она воплощает в себе преимущества каскадной модели, и в то же время допускает итерации по всем фазам одной и той же модели;

- позволяет получить более надёжную и стабильную систему. По мере развития программного обеспечения ошибки и слабые места обнаруживаются и исправляются на каждой итерации;

- позволяет пользователям активно участвовать в планировании, анализе рисков, разработке и оценке мероприятий;

- риски клиента снижаются. Заказчик может завершить разработку бесперспективного проекта с минимальными финансовыми потерями;

- обратная связь от пользователей к разработчикам осуществляется с высокой частотой и на ранних стадиях разработки модели, что обеспечивает создание желаемого качественного продукта.

Недостатки спиральной модели:

- жизненный цикл модели имеет сложную структуру, поэтому разработчикам, менеджерам и заказчикам может быть трудно её применять;

- спираль может продолжаться бесконечно, так как каждый отклик клиента на созданную версию может породить новый цикл, который задерживает окончание работы над проектом;

- большое количество промежуточных циклов может привести к необходимости в обработке дополнительной документации;

- использование модели может быть дорогостоящим и даже запредельным с точки зрения времени, т.к. время, затраченное на планирование, переопределение целей, анализ рисков и создание прототипов, может оказаться чрезмерным;

- может быть трудно определить цели и этапы, которые указывают на готовность продолжать процесс развития на следующем этапе.

Основная задача спирального цикла – определить момент перехода к следующему этапу. Для решения этой задачи на каждый этап жизненного цикла накладываются временные ограничения и переход осуществляется в соответствии с планом, даже если не все запланированные работы выполнены. Планирование базируется на статистических данных, полученных в предыдущих проектах, и личном опыте разработчиков.

Применение спиральной модели целесообразно в следующих случаях:

- при разработке новой серии продуктов или систем;
- при разработке проектов с использованием новых технологий;
- при разработке проектов, требующих демонстрации качества и версий системы или продукта через короткий промежуток времени;
- при разработке проектов с ожидаемыми существенными изменениями или дополнениями требований;
- для выполнения долгосрочных проектов;
- при разработке проектов, требующих расчёта затрат, связанных с оценкой и разрешением рисков.

В проекте была выбрана спиральная модель жизненного цикла программного обеспечения, так как детальная формализация программного обеспечения не была предоставлена заказчиком. Его требования были поверхностными. Так же спиральная модель является наиболее подходящей к требованию масштабируемости. Заказчик предполагает дополнение системы отдельными модулями, в том числе и программными.

Таким образом, данная модель является наиболее подходящей для решения поставленной задачи.

5.2 Проектирование основного модуля SCH-Base

Основной модуль называется SCH-Base. Данный модуль является основным и главным в системе. Аппаратная часть состоит из микрокомпьютера Raspberry Pi и блока питания к нему. Микрокомпьютер имеет операционную систему Raspbian, основанную на дистрибутиве Debian.

Raspberry Pi - одноплатный компьютер компактного размера (Рис. 14). Имеет разъём HDMI для подключения монитора, USB-порты для подключения USB устройств, GPIO разъём для подключения низкоуровневой периферии, Ethernet-порт для подключения к сети. Используемая модель – RaspberryPi Model B имеет процессор ARM 700Ghz, 512Мб оперативной памяти.

Raspberry Pi - это компактный одноплатный компьютер (рис. 14). Он имеет разъем HDMI для подключения монитора, порты USB для подключения USB-устройств, разъем GPIO для подключения низкоуровневого периферийного устройства и порт Ethernet для подключения к сети. Используемая модель-Raspberry Pi Model B имеет процессор ARM 700 ГГц и 512 МБ оперативной памяти.

Рисунок 14 – Микрокомпьютер Raspberry Pi

Для отладки данного модуля установлен VNC Server. VNC-Virtual Network Computing (VNC) - это система удалённого доступа к рабочему столу компьютера с использованием протокола RFB (англ. Remote FrameBuffer, удалённый буфер кадров). Управление осуществляется путём передачи нажатий клавиш клавиатуры и движений мыши с одного компьютера на другой и ретрансляции содержимого экрана по компьютерной сети.

VNC - независимая от платформы система: клиент VNC под названием VNC viewer, работающий на одной операционной системе, может подключаться к серверу VNC, работающему на любой другой операционной системе.

Именно в этом модуле установлен Blynk Server.

Данный модуль должен иметь стабильное подключение по WiFi. Имеется возможность подключения Ethernet-кабеля от роутера.

Так же, по аналогии с другими микроконтроллерами, на Raspberry может запускаться программа, написанная на C++ и использующая Blynk Libraries для Raspberry. В этом случае, микрокомпьютер выступает в той же роли, что микроконтроллеры, подсоединенные к серверу через протокол Blynk.

На рисунке 1 в приложении А приведен интерфейс программы работы по расписанию и конфигурационный файл с его настройками.

Интерфейс приложения модуля представлен на рисунке 15.

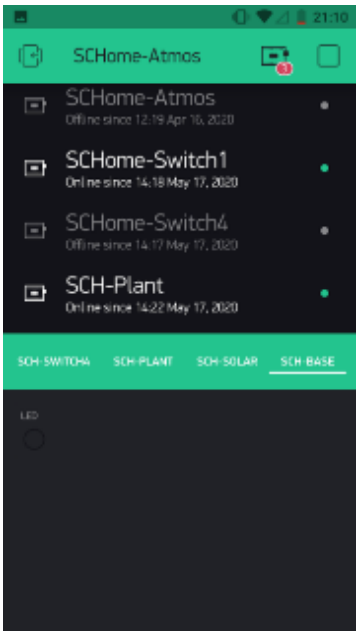


Рисунок 15 – Интерфейс модуля SCH-Base в приложении.

В мобильном приложении отображается список модулей, находящихся в системе и их состояния.

Электрическая схема модуля представлена на рисунке 16.

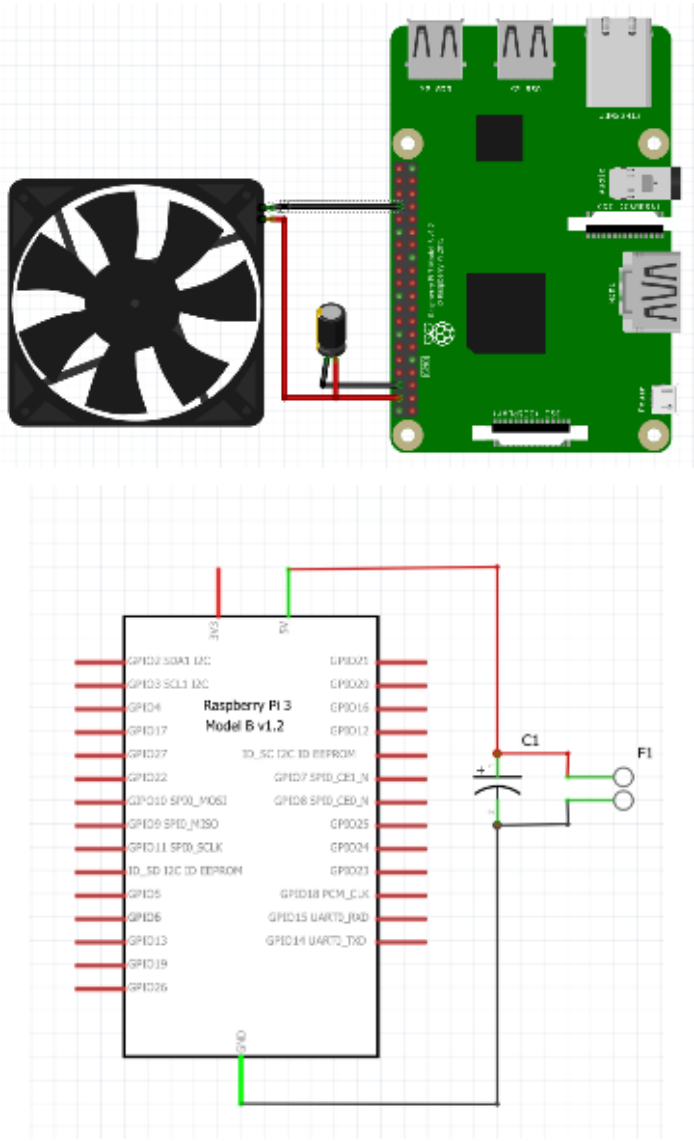


Рисунок 16 – Электрическая схема SCH-Base

5.2.1 Проектирование подсистемы работы модулей по расписанию

Для реализации работы системы по расписанию, была разработана программа, которая постоянно запущена в фоновом режиме на Raspberry Pi.

Языком программирования данной программы был выбран Python, так как язык является высокоуровневым и средства запуска программ на нем были уже предустановлены в операционной системе Raspberry Pi.

Еще одной особенностью повлиявшей на выбор языка было наличие готовых библиотек для работы с файловой системой, текстовыми и конфигурационными файлами, с системой запланированных задач, что ускоряет разработку программы в целом.

Модуль обработки расписания содержит в себе компоненты интерфейсов администратора и пользователя. Оба позволяют создавать и редактировать расписания работы модулей системы, и отправлять запросы на микроконтроллеры. Компонент интерфейса администратора реализует функции добавления новых пользователей, редактирования их прав, а также формирования отчета по отправке запросов обычных пользователей.

Схема модуля обработки расписания представлена на рисунке 17

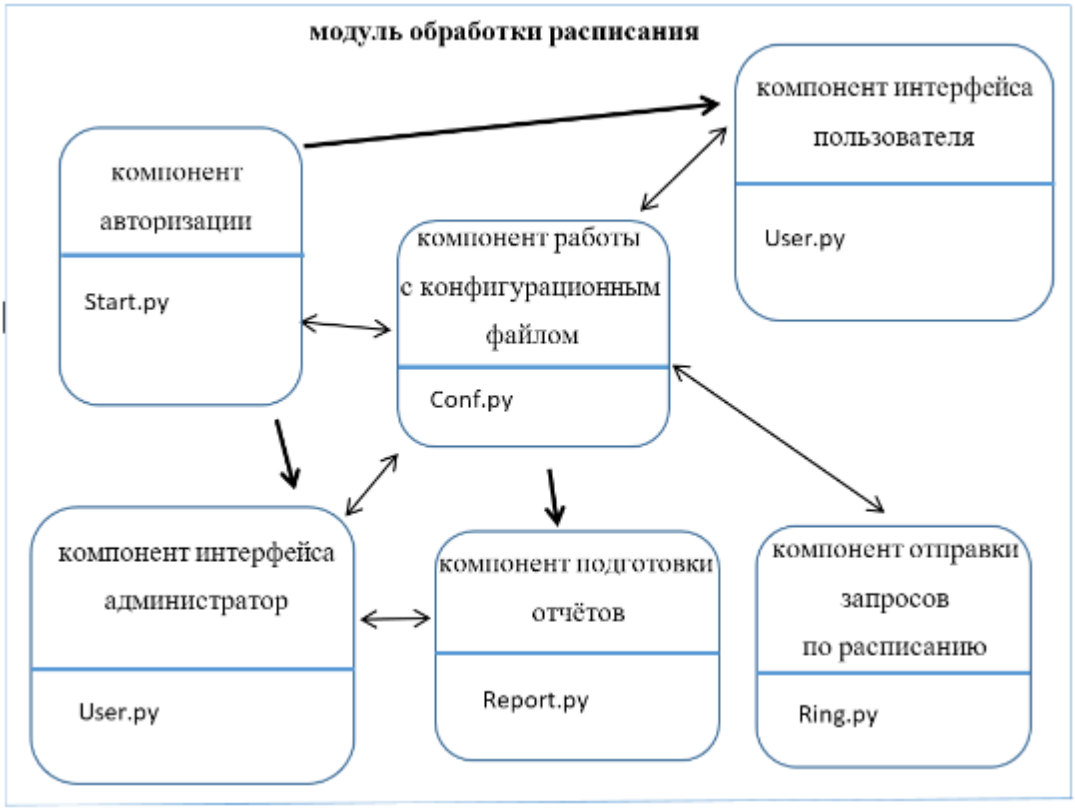


Рисунок 17 – Схема взаимодействия компонентов в модуле обработки расписания

Компонент отправки запросов по расписанию обрабатывает текущее расписания, создает события, привязанные к определённым дате и времени, указанным в расписании и отправляет запросы на микроконтроллеры модулей.

5.2.2 Проектирование подсистемы работы модулей в ручном режиме

Управление модулями в ручном режиме осуществляется через интерфейс Android-приложения и web-интерфейс.

Приложение является клиентом, обменивающийся данными с системой по протоколу Blynk.

При первом включении необходимо зарегистрироваться в системе или войти, если пользователь уже создан администратором. Экранная форма окна авторизации представлена на рисунке 18.

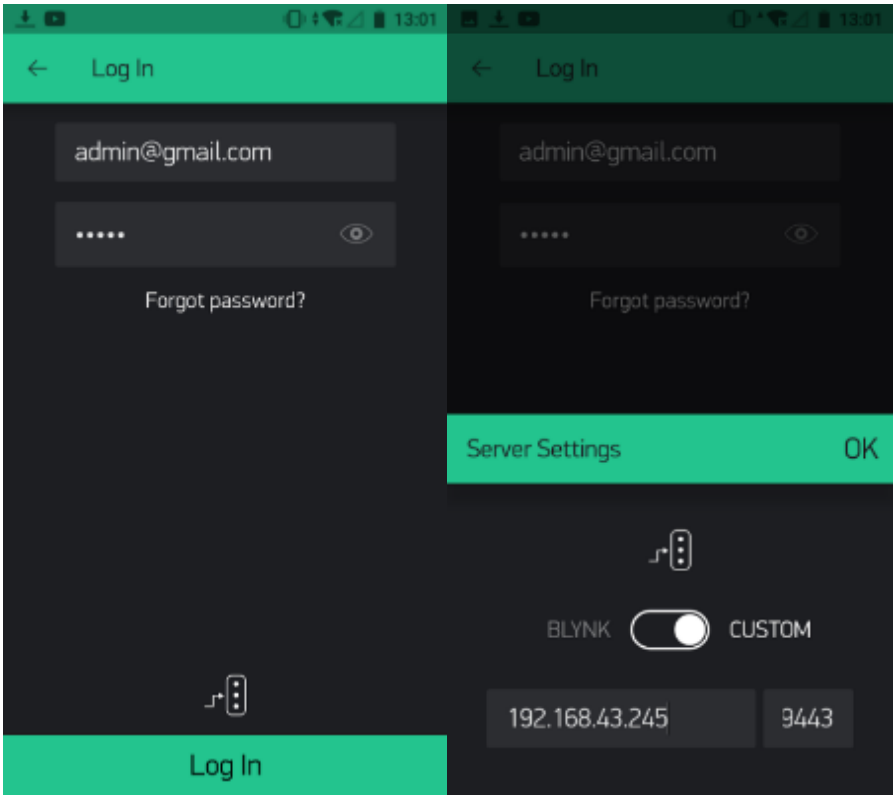


Рисунок 3.1 – Авторизация в Blynk App

По умолчанию система работает в автоматическом режиме. Интерфейс приложения состоит из кнопок, переключателей, слайдеров, по нажатию на которые отправляются команды на модуль через сервер Blynk, а также из полей данных, графиков, терминалов, куда выводится информация с модулей. Данный интерфейс настраивается при включении модулей в систему и может в любой момент изменен пользователем. Для удобства, интерфейс разных модулей разнесен на несколько вкладок (Tabs). В случае необходимости ручного управления (к примеру, включить лампу во время, не предусмотренное расписанием) по нажатию кнопки (DEBUG_LED) модуль переходит в ручное управление, и при повторном нажатии возвращается в автоматический режим.

Интерфейсы различных модулей – SCH-Atmos, SCH-Switch1, SCH-Switch4, SCH-Plant, SCH-Solar, SCH-Base представлены в Приложении В.

5.3 Проектирование web-интерфейса модулей

Для удобства настройки модуля и его включения в систему был разработан web-интерфейс модуля. При первом запуске, микроконтроллер переходит в режим конфигурирования – ESP вместо подключения к WiFi-сети, сам поднимает свою WiFi точку доступа, со стандартными для модуля названием сети (SSID) и паролем. При подключении к этой точке доступа с любого устройства, будь то ПК или смартфон с браузером, необходимо перейти по адресу http://192.168.220.1/. На микроконтроллере в режиме конфигурации поднят web-сервер, который по этому адресу отображает страницу с изменяемыми параметрами.

Пример страницы конфигурации модуля SCH-Atmos представлен на рисунке 19.

Рисунок 19 - Web-интерфейс модулей

На странице присутствует форма с полями ввода данных и кнопка Save, для отправки содержимого этих полей на web-сервер микроконтроллера. Одинаковыми для всех модулей полями являются:

– «WiFi SSID» - первый SSID первой WiFi сети системы.

– «PWD» - пароль к первой WiFi сети.

– «WiFi SSID» - второй SSID WiFi сети системы, который используется в случае невозможности подключения к первой WiFi сети.

– «Blynk Server» - IP-адрес или доменное имя сервера, на котором развернут сервер протокола Blynk (например, 192.168.43.245 или blynk-cloud.com).

– «Token» - аутентификационный ключ (токен), данного модуля.

– «Blynk Server1» - второй IP-адрес или доменное имя сервера, на котором развернут сервер протокола Blynk (используется в случае, если нет подключения к первому).

– «Token1» - аутентификационный ключ (токен), данного модуля (на каждом сервере он уникальный, даже если это одно и тоже устройство).

– «Port» - порт, используемый для протокола Blynk (8080 или 9443).

– «Board Name» - имя данного модуля, которое можно изменить, в случае если в системе присутствуют два модуля одного типа.

Значения вышеперечисленных полей заданы по умолчанию в прошивке. Так же, для каждого типа модуля присущи собственные поля, которые нужно изменить или задать при первоначальной настройке, но нет необходимости изменять в дальнейшем, и добавлять для них кнопки в интерфейсе приложения Blynk App. Это могут быть значения интервала работы модуля (опроса датчиков), номера контактов микроконтроллера, к которому подключен датчик, название кнопки управления в Blynk App (Label) и т.д.

При отправке формы, значения сохраняются как значения по умолчанию в файловую систему ESP.

Переход в режим конфигурации осуществляется и при других условиях:

– нет возможности подключиться к WiFi с параметрами по умолчанию, неправильно введен SSID и/или пароль (через 10 секунд ESP перезагружается в режим конфигурации).

– два раза нажата кнопка сброса (Reset) ESP (ESP сразу перезагружается в режим конфигурации).

5.4 Проектирование модуля SCH-Switch1

Модуль SCH-Switch1 является самым простым модулем, так как состоит из двух компонентов – микроконтроллера NodeMCU и модуля реле.

Функция модуля заключается в управлении нагрузкой, выступая в роли электронного реле. После приема команды от других модулей или смартфона, микроконтроллер подает сигнал на реле, и контакты реле замыкаются. На прототипе модуль управляет работой лампы 220В.

Для питания модуля можно использовать как блок питания на 5V (на прототипе) или AC/DC преобразователь, чтобы брать питание напрямую с сети 220В как на примере SCH-Switch4.

Физическая и электрическая схема представлены на рисунках 20-21.

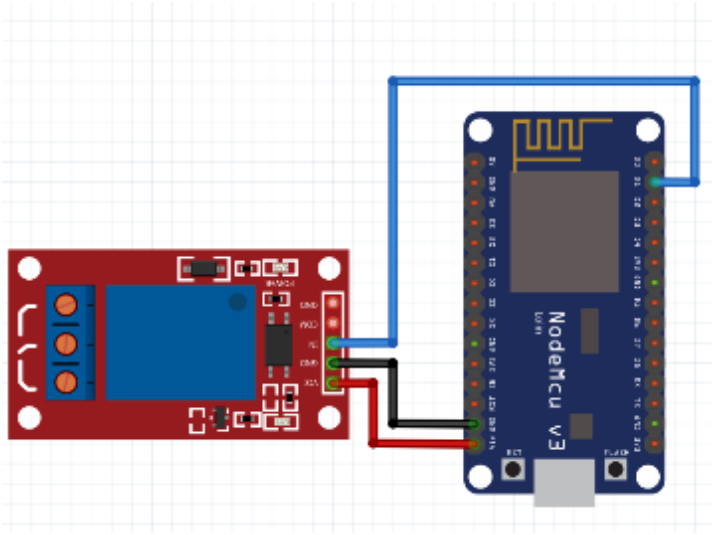


Рисунок 20 – Физическая схема SCH-Switch1

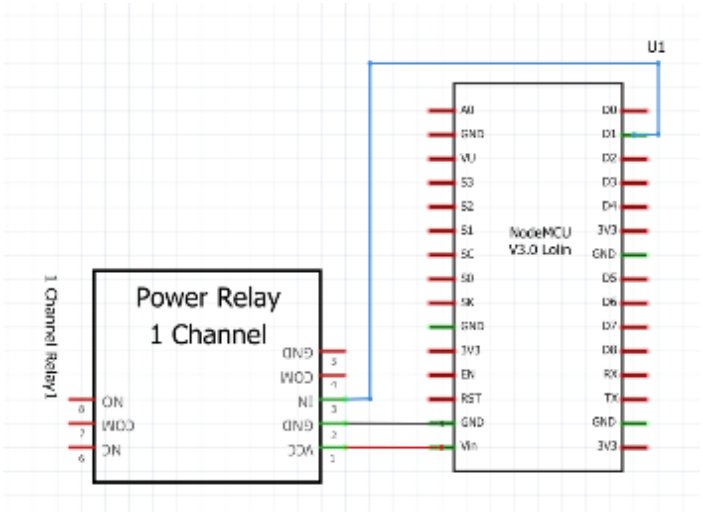


Рисунок 21 – Электрическая схема SCH-Switch1

5.5 Проектирование модуля SCH-Switch2

Модуль SCH-Switch2 аналогичен модулю SCH-Switch1, за исключением присутствия возможности управления большим количеством оборудования. На модуле присутствует 2 реле.

Физическая и электрическая схема представлены на рисунках 22-23.

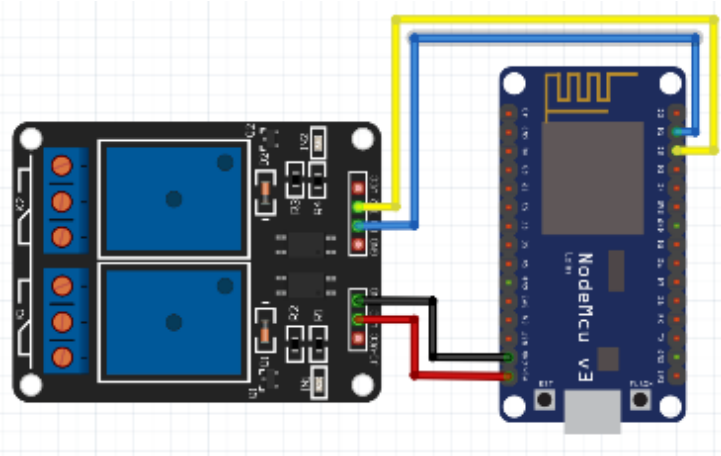


Рисунок 22 – Физическая схема SCH-Switch2

5.6 Проектирование модуля SCH-Switch4

Модуль SCH-Switch4 аналогичен модулям SCH-Switch1 и SCH-Switch2, за исключением возможности управления еще большим количеством оборудования. На модуле присутствует 4 реле.

Физическая и электрическая схема представлены на рисунках 24-25.

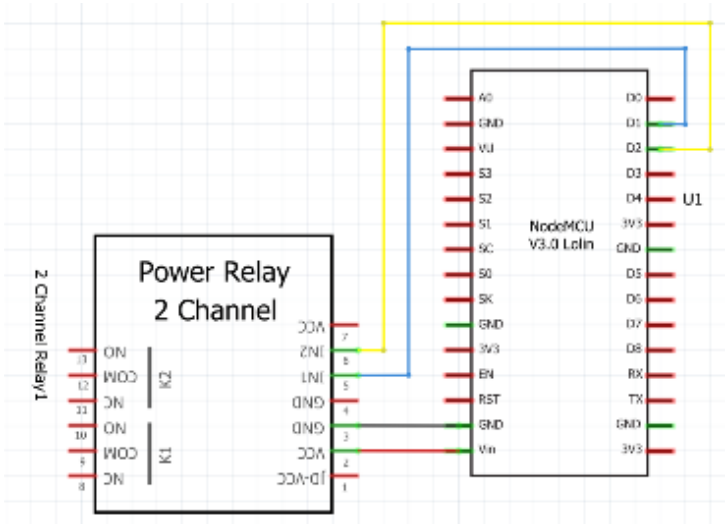


Рисунок 23 – Электрическая схема SCH-Switch2

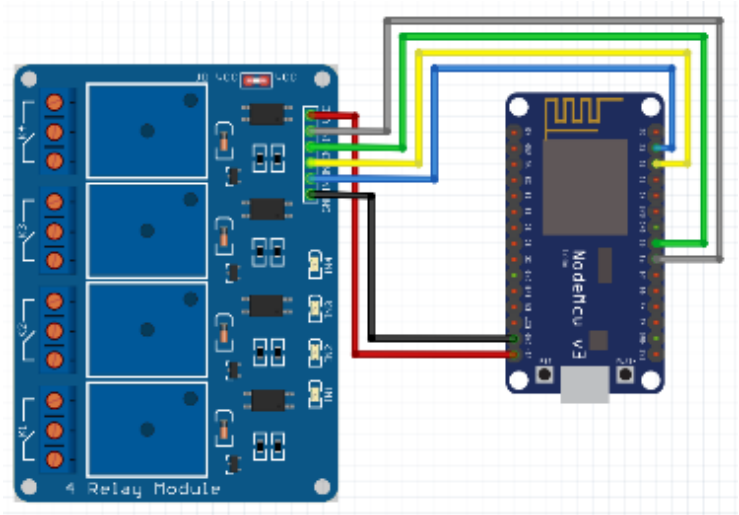


Рисунок 24 – Физическая схема SCH-Switch4

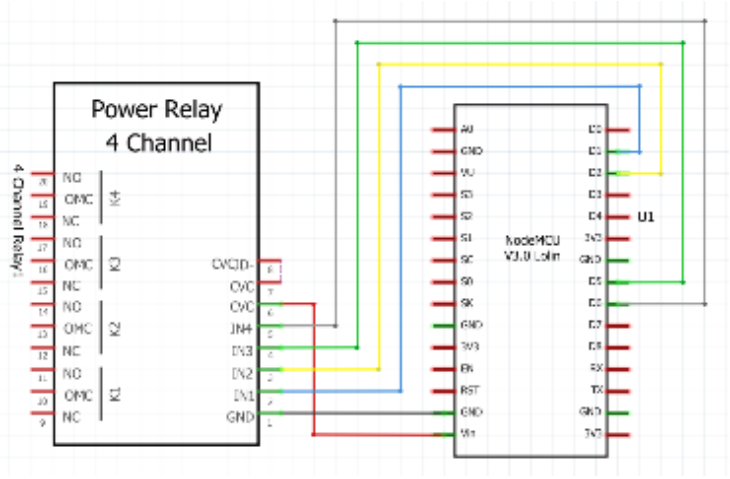


Рисунок 25 – Электрическая схема SCH-Switch4

5.7 Проектирование модуля SCH-Plant

Модуль SCH-Plant является микроконтроллером NodeMCU с двумя подключенными датчиками. Датчик влажности почвы, размещается в почве и замеряет ее влажность, передавая показания микроконтроллеру. Датчик BMP280 измеряет влажность и температуру воздуха (так же имеется возможность измерять атмосферное давление).

Питание данного модуля может осуществляться с помощью блока питания на 5V, что удобно для систем, размещенных в помещении. Также модуль может дополнен аккумулятором, что позволяет размещать в любом месте большой теплицы, вне доступности от проводов. Единственный недостаток в том, что данный вариант модуля придется время от времени заряжать. В случае низкого разряда аккумулятора, модуль отправляет оповещение на смартфон. Так же, во избежание передачи другим модулям неверных показаний (когда вытаскиваем датчик из земли), предусмотрено включение режима ожидания со смартфона, замораживающего работу модуля.

Данные со датчиков передаются и сохраняются на сервере Blynk.

Физическая и электрическая схема представлены на рисунках 26-27.

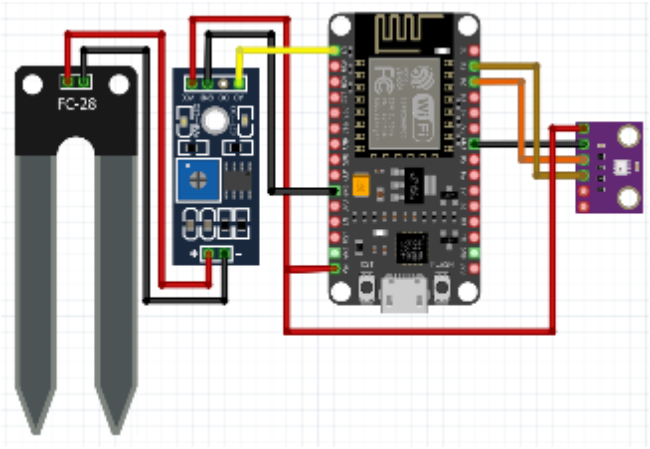


Рисунок 26 – Физическая схема SCH-Plant

5.8 Проектирование модуля SCH-Solar

SCH-Solar это модуль управления солнечными панелями с отслеживанием солнца и вырабатываемой электроэнергии. Данный модуль состоит из микроконтроллера ESP, и датчиков тока и напряжения. Измеряемые характеристики

подключаемой цепи: сила тока до 5А и напряжение постоянного тока 25В.Данный модуль имеет контакты для подключения двух сервомоторов, управляемых по ШИМ или одного драйвера шагового мотора. Данная возможность позволить управлять солнечными панелями малых размеров по двум осям или большими солнечными панелями по одной оси.

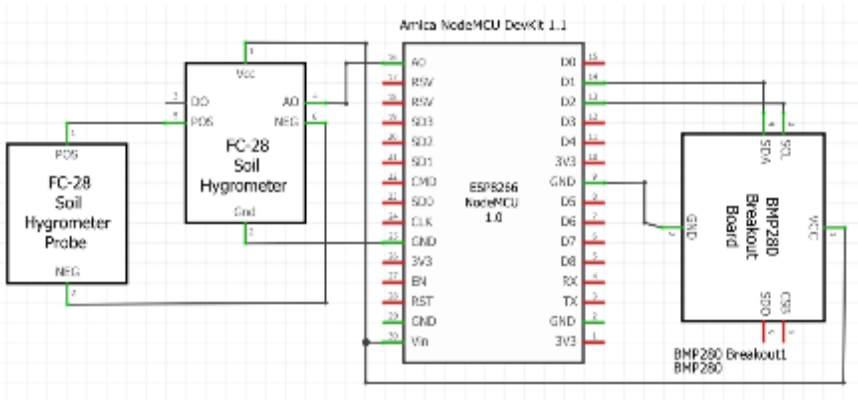


Рисунок 27 – Электрическая схема SCH- Plant

Физическая и электрическая схема представлены на рисунках 28-29.

Для данного модуля были разработаны 3D-модели деталей механизма поворота 6 солнечных панелей размерами 110мм на 60мм мощностью 1Ватт каждая подсоединенных паралельно-последовательно. Данные модели можно распечатать на 3D-принтере и собрать поворотную контрукцию (Рис. 30).

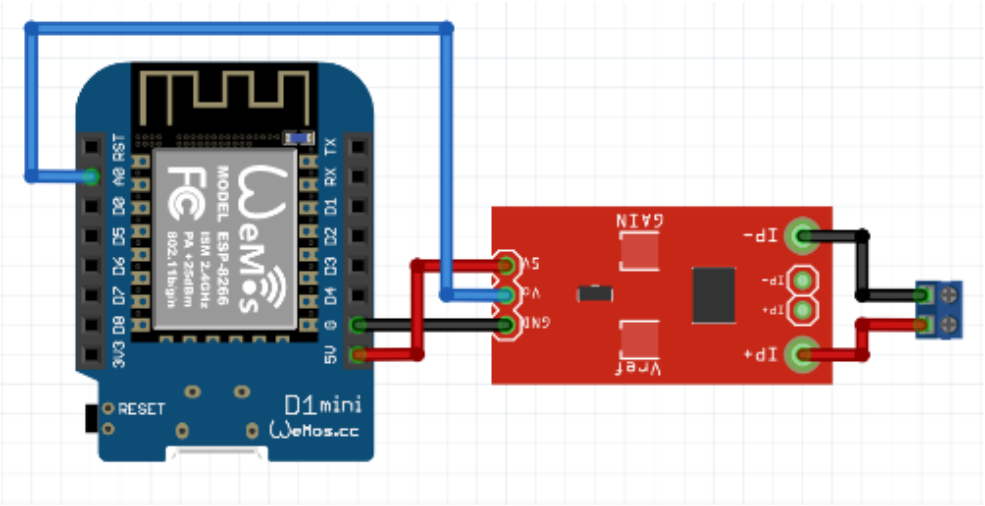


Рисунок 28 – Физическая схема SCH-Solar

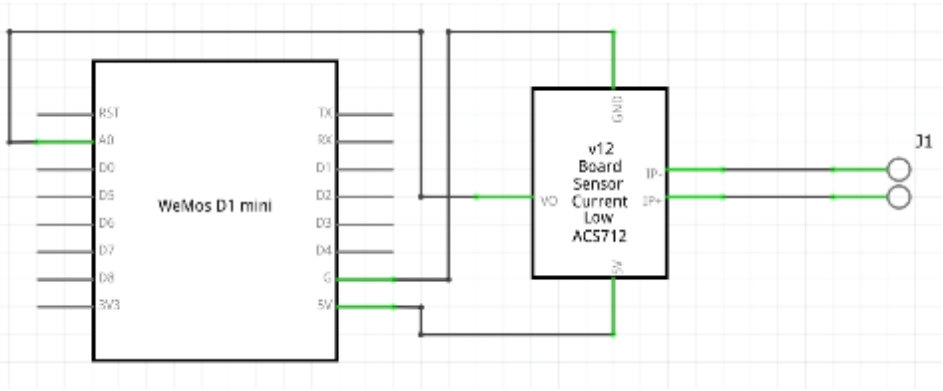


Рисунок 29 – Электрическая схема SCH- Solar

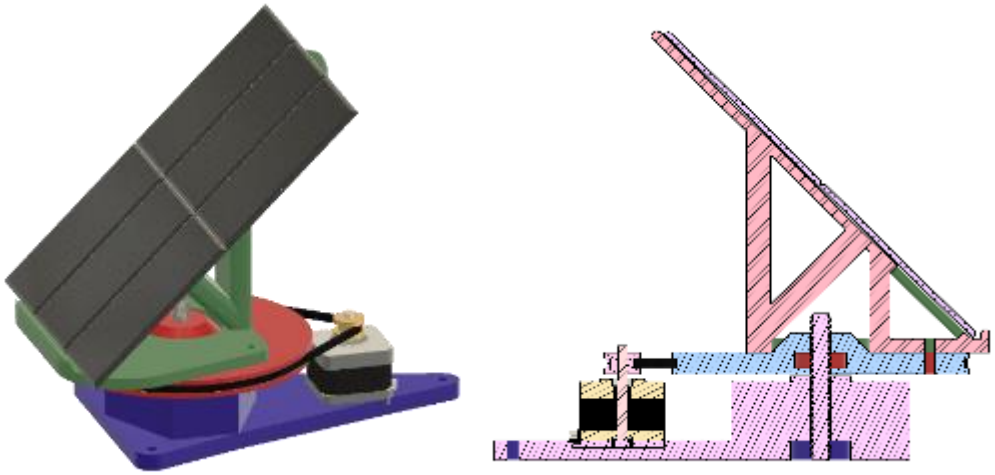


Рисунок 30 – 3D модель поворотного стола с солнечными панелями, управляемого с модуля SCH-Solar

Модуль SCH-Solar на данный момент находится в разработке.

5.9 Проектирование модуля SCH-Atmos

Модуль SCH-Atmos является модулем анализатором, так как состоит из микроконтроллера ESP и датчиками, собирающими данные о температуре, влажности, и относительной загрязнённости воздуха. Данные передаются в систему SCHome, и на их основе принимается решения о передачи запросов другим модулям.

Физическая и электрическая схема представлены на рисунках 31-32.

Прототип находится в активной стадии разработки, а именно, собираются экспериментальные данные от датчиков-газоанализаторов MQ, чтобы выяснить,

какой из них является наиболее подходящим для выполнения функции мониторинга относительной загрязнённости воздуха.

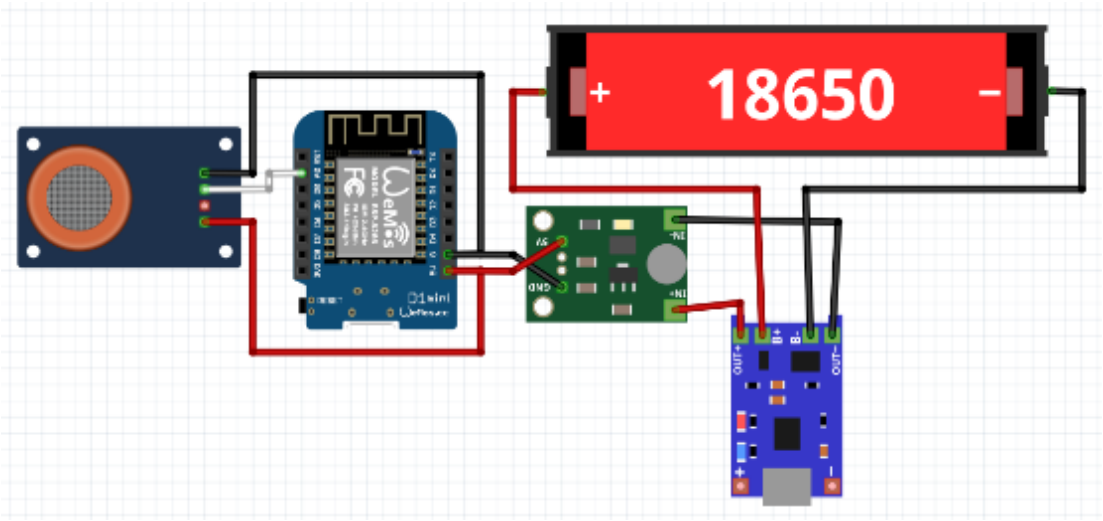


Рисунок 31 – Физическая схема SCH-Atmos

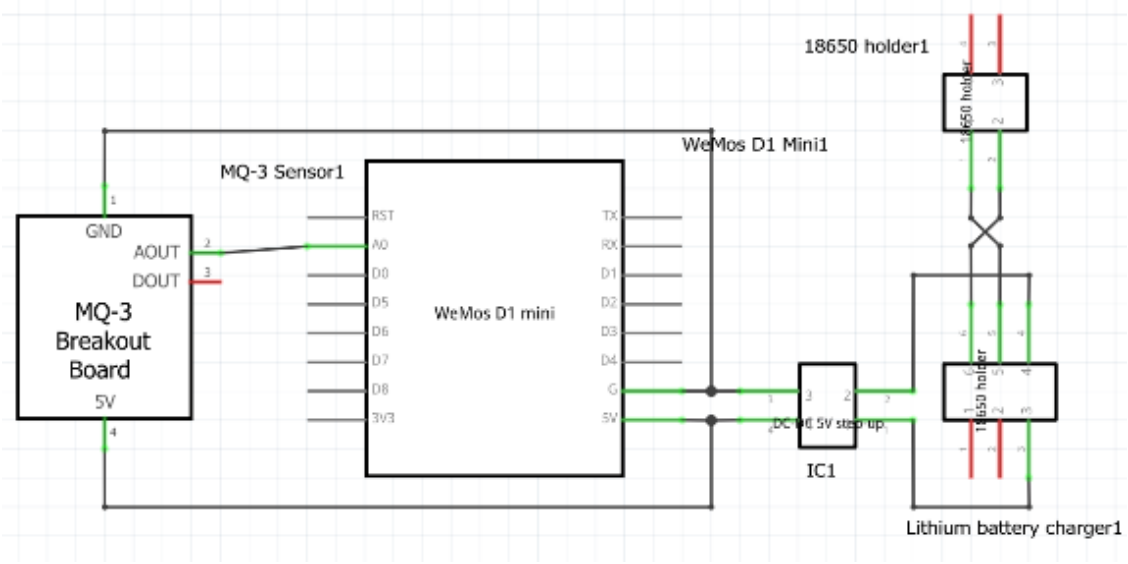


Рисунок 32 – Электрическая схема SCH- Solar

5.10 Проектирование модуля SCH-Cam

SCH-Camera - модуль поворотной камеры. Он используется для обеспечения безопасности в местах, где требуется дополнительная видимость к основной системе видеонаблюдения.

Состоит из микроконтроллера ESP32 с подключенной камерой OV7670. Данный модуль, как и SCH-Solar имеет возможность подключения двух

сервомоторов или одного драйвера шагового двигателя. Относится ко второму этапу разработки системы.

5.11 Проектирование модуля SCH-Wind

SCH-Windmill – модуль для отслеживания электроэнергии, вырабатываемой вертикальными ветрогенераторами (с низким уровнем шума). В отличие от модуля SCH-Solar не имеет возможности подключения сервомоторов или драйвера шагового двигателя. Относится ко второму этапу разработки системы.

В дальнейшем, планируется переработать модули SCH-Solar и SCH-Wind, создав из них универсальный модуль отслеживания вырабатывания и потребления электроэнергии. Данный модуль можно будет встроить не только в сеть с солнечными панелями или ветряными электрогенераторами, но и в сеть с различной техникой и оборудованием для отслеживания энергопотребления данными приборами.

5.12 Проектирование модуля SCH-FaceId

SCH-FaceId - это подсистема распознавания лиц на контрольно-пропускном пункте предприятия или при входе в помещение. Позволяет отслеживать время прибытия авторизованных пользователей на вход в помещение, распознавая лица и соотнося их с данными, хранящимися в системе. На основе модуля SCH-Cam. Данный модуль относится ко второму этапу развития системы.

5.13 Проектирование модуля SCH-Broadcast

SCH-Broadcast – модуль для обеспечения голосовой радиосвязи в помещениях между рабочими местами. Режимы работы модулей могут быть изменены вручную администратором (или ответственным) или по расписанию. Режимы "один ко многим" и "один к одному". Эти модули позволяют организовать систему оповещения, а также голосовую связь между людьми в разных местах. Данный модуль относится ко второму этапу развития системы.

5.14 Создание прототипа системы SCHome

Для демонстрации работы системы был разработан прототип, который состоит из следующих модулей: SCH-Base, SCH-Switch1, SCH-Switch4, SCH-Atmos, SCH-Plant SCH-Solar и WiFi- .

Прототип планируется установить в ДНК для демонстрации работы системы и в образовательных целях.

Система, основанная на функционале прототипа должна быть развернута в масштабе помещения ДНК. При этом предварительно должны быть доработаны перечисленные модули, а также разработаны и напечатаны на 3D-принтере их корпуса.

Внешний вид данного прототипа представлен на рисунке 33.

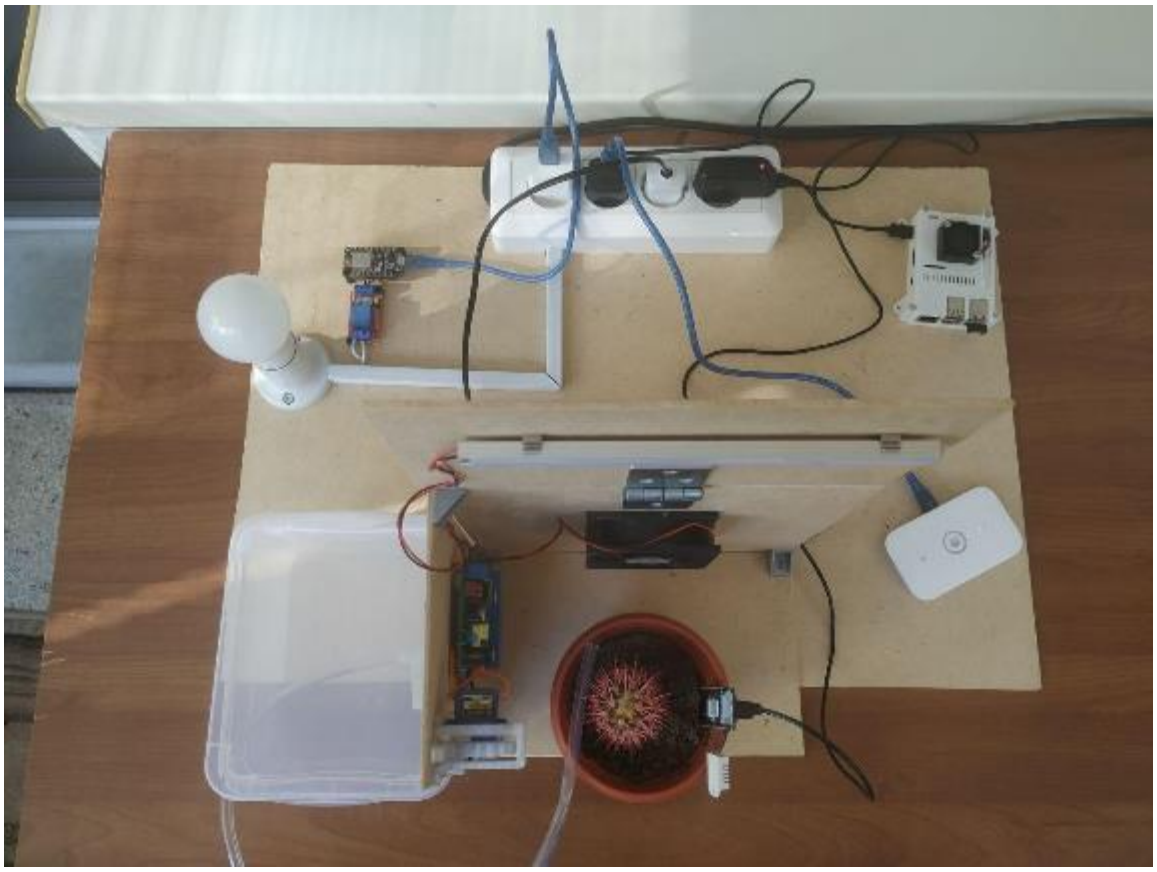


Рисунок 33 – Внешний прототипа SCHome

Внешний вид модуля SCH-Base представлен на рисунке 34.

Рисунок 34 – Внешний вид модуля SCH-Base

ЗАКЛЮЧЕНИЕ

В результате выполнения магистерской диссертации были разработаны информационная система управления оборудованием и техникой SCHome, модули и прототип системы.

Перед разработкой были сформулированы требования и функциональные возможности системы. Помимо этого, был произведен анализ рисков возможных видов атак и модели нарушителя, осуществляющего атаки на разрабатываемую систему. В дальнейшем, все это учитывалось при разработке, и были предприняты меры по защите системы как от атак изнутри (ложный объект), так и атак извне (атаки на локальную сеть).

Развёрнуты сервера протокола Blynk, базы данных PosgreSQL, интерпретаторы скриптов Python, C++ и WEB сервер CGI. Настроено взаимодействие развёрнутого ПО. Написание скриптов осуществлялось в бесплатной среде разработки Arduino IDE и Visual Studio. С модулем SCH-Base был установлен удалённый доступ к рабочему столу компьютера.

Разработаны прототипы модулей SCH-Base, SCH-Switch1, SCH-Switch2, SCH-Switch4, SCH-Plant, SCH-Atmos, SCH-Solar и программный код для каждого из этих модулей. В ходе написания программной части были сформированы несколько вариантов программ с отличающимся функционалом: вариант программы с минимальными объёмами занимаемой памяти; вариант для удобного чтения сторонними разработчиками; вариант с максимальным приведением к универсальности и независимости от модулей, на которых данный программный код запускается.

Программный код находится в открытом доступе (open-source) и размещён на платформе GitHub: https://github.com/Picjavard/SCHome.

В дальнейшем планируется доработать систему, а именно разработать новые модули системы, смоделировать корпусы модулей и распечатать прототипы на 3D-принтере. Так же требуется создать подсистему обработки ошибок и отка-

зов системы, реализовать сохранение данных в локальной базе данных, организовать простую систему принятий решений на основе внедрения нейронной сети, разработать голосовой интерфейс для основного модуля или смартфона. Разработка спецификаций программного обеспечения и технического задания на разработку системы для ЦРСКД «АмурТехноЦентр» «ДНК им. акад. РАН М.Т. Луценко», а также руководство пользователя и документации является первостепенной задачей. Планируется разработка различных комплектов с моделями для распространения, а также шаблоны программного кода с подробной документацией к ним, для возможности разработки, модификации, доработки модулей другими разработчиками. Выполнение данной задачи необходимо для привлечения разработчиков программного обеспечения в команду. После выполненных вышеперечисленных задач планируется налаживание контактов и сотрудничества с фирмами-поставщиками и производителями электроники для последующего производства комплектов системы SCHome и распространения на рынке.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Грингард, С. Интернет вещей. Будущее уже здесь / С. Грингард. - М.: Альпина Паблишер, 2016. - 188 с.

2 Дейтел, П. Android для программистов: создаем приложения / П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. – СПб.: Питер, 2013. – 560 с.

3 Дементьев, А. «Умный» дом XXI века / А. Дементьев : Издательские решения, 2016. - 196 стр.

4 Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – М.: Издательство «Русская Редакция»; СПб.: Питер, 2008. – 896 с.

5 Олифер, В.Г. Основы сетей передачи данных / В.Г. Олифер, Н.А. Олифер. – СПб: Питер, 2009. – 663 с.

6 Петрушин, С.А. Источники энергии для индивидуальных домов / С.А. Петрушин, В.А. Глушков: LAP Lambert Academic Publishing, 2014. 196 стр.

7 Архитектура клиент-сервер или Web: выбор разработчика [Электронный ресурс]. – Режим доступа: http://www.interface.ru/home.asp?artId=22674 – 25.05.2019.

8 Жизненный цикл тестирования ПО [Электронный ресурс]. – Режим доступа: http://ru.qatestlab.com/knowledge-center/qa-testing-materials/5-distinctions-between-a-client-server-and-web-application/ – 12.05.2020.

9 Особенности проектирования интерфейсов информационных систем [Электронный ресурс]. – Режим доступа: https://revolution.allbest.ru/programming/00781099_0.html – 05.04.2020.

10 Цели и задачи проектирования [Электронный ресурс]. – Режим доступа: http://msd.com.ua/chelovecheskij-faktor/celi-i-zadachi-proektirovaniya/ – 12.04.2020.

11 Проектирование информационной системы [Электронный ресурс]. – Режим доступа: https://finswin.com/projects/proektirovanie/informacionnyh-sistem.html.

12 Проектирование программного обеспечения [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/edison/blog/267569/

13 Жизненный цикл программного обеспечения [Электронный ресурс]. – Режим доступа: https://qaevolution.ru/zhiznennyj-cikl-programmnogo-obespecheniya/ - 24.10.2019.

14 Documentation for Blynk, the most popular IoT platform for businesses [Электронный ресурс]. – Режим доступа: https://docs.blynk.cc/#blynk-server – 27.05.2020.

15 Espressif Systems - Wi-Fi and Bluetooth chipsets and solutions [Электронный ресурс]. – Режим доступа: https://www.espressif.com/ – 05.11.2019.

16 Installing operating system images - Raspberry Pi Documentation [Электронный ресурс]. – Режим доступа: https://www.raspberrypi.org/documentation/installation/installing-images/README.md – 12.10.2018.

17 khoih-prog/Blynk_WM: Blynk and WiFiManager Library [Электронный ресурс]. – Режим доступа: https://github.com/khoih-prog/Blynk_WM – 23.05.2020.

18 Arduino совместимые платы и робототехника [Электронный ресурс]. – Режим доступа: https://www.chipdip.ru/catalog/arduino-boards – 11.12.2019.

19 Fritzing [Электронный ресурс]. – Режим доступа: https://fritzing.org/learning/get-started – 26.01.2019.

20 BlackBox: Код. Связь WeMos и Arduino по UART [Электронный ресурс]. – Режим доступа: https://it-blackbox.blogspot.com/2018/10/wemos-arduino-uart_85.html – 14.11.2019.

Библиографический список авторских работ по теме диссертации

1. Демьяненко А.Е. Проект разработки информационной системы «Конфигурируемый дом». // Молодёжь XXI века: шаг в будущее: материалы XVIII региональной научно- практической конференции (18 мая 2017 года) – Благовещенск: Изд-во БГПУ, 2017. – С. 1012-1013.

2. Самохвалова С.Г., Демьяненко А.Е. Разработка интернет-магазина готовых решений системы «Конфигурируемый дом». // Экономика и социум: международный научно-практический электронный журнал – Институт управления и социально-экономического развития, 2017. - № 12 (43).

3. Самохвалова С.Г., Демьяненко А.Е. Разработка программного обеспечения для системы «Конфигурируемый дом» // Современные проблемы науки: материалы Российской национальной научной конференции с международным участием (22 декабря 2017 г.). – Часть I. – Благовещенск Амурский гос. ун-т, 2017. – С. 128-130

4. Самохвалова, С.Г., Демьяненко, А.Е. Разработка устройства отладки для отображения текущего состояния системы «Конфигурируемый дом» // Вестник Амурского государственного университета. – Серия «Естественные и экономические науки», Выпуск 81. – Благовещенск, Изд-во АмГУ, 2018 – С. 40-43

5. Самохвалова, С.Г., Демьяненко, А.Е. Прототипирование отладочного устройства для системы «Конфигурируемый дом» // Молодёжь XXI века: шаг в будущее: материалы XIX региональной научно- практической конференции (23 мая 2018 года) – Благовещенск: Изд-во ДальГАУ, 2018. – С. 177-178

6. Самохвалова, С.Г., Демьяненко, А.Е. Прототипирование мобильной платформы с компьютерным зрением для системы «SCHome» // Молодёжь XXI века: шаг в будущее: материалы XX региональной научно- практической конференции (24 мая 2019 года) – Благовещенск: Изд-во АмГУ, 2019.

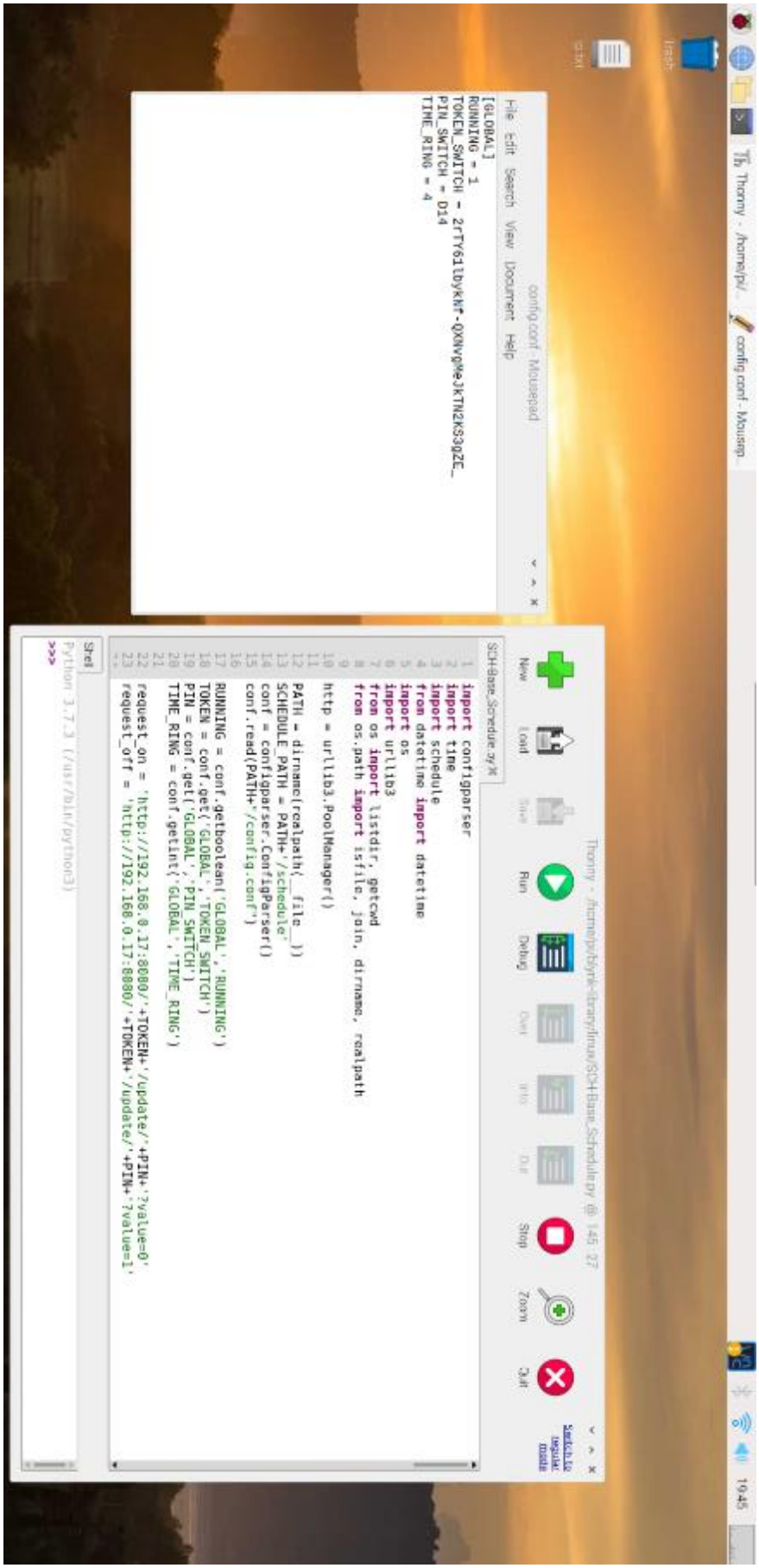
7. Самохвалова, С.Г., Демьяненко, А.Е. Автоматизированная система оповещения для общеобразовательных учреждений на основе системы «SCHome» // Вестник Амурского государственного университета. – Благовещенск, Изд-во АмГУ, 2019

8. Самохвалова, С.Г., Демьяненко, А.Е. Разработка модуля системы «SCHome» для анализа показателей атмосферы «SCH-Atmos» // В кн.: «Молодёжь XXI века: шаг в будущее»: материалы XXI региональной научно-практической конференции (20 мая 2020 года) – Благовещенск: Изд-во АГМА, 2020. – С. 114-115

9. Самохвалова, С.Г., Демьяненко, А.Е. Разработка аппаратно-программного обеспечения системы SCHome и ее модулей по технологии «Internet of Things» // «Научные тенденции: Вопросы точных и технических наук»: Сборник

научных трудов по материалам XXVIII международной научно-практической конференции 12 июня 2020 г. Изд. ЦНК МОАН, 2020. – С. 10-11

Приложение А



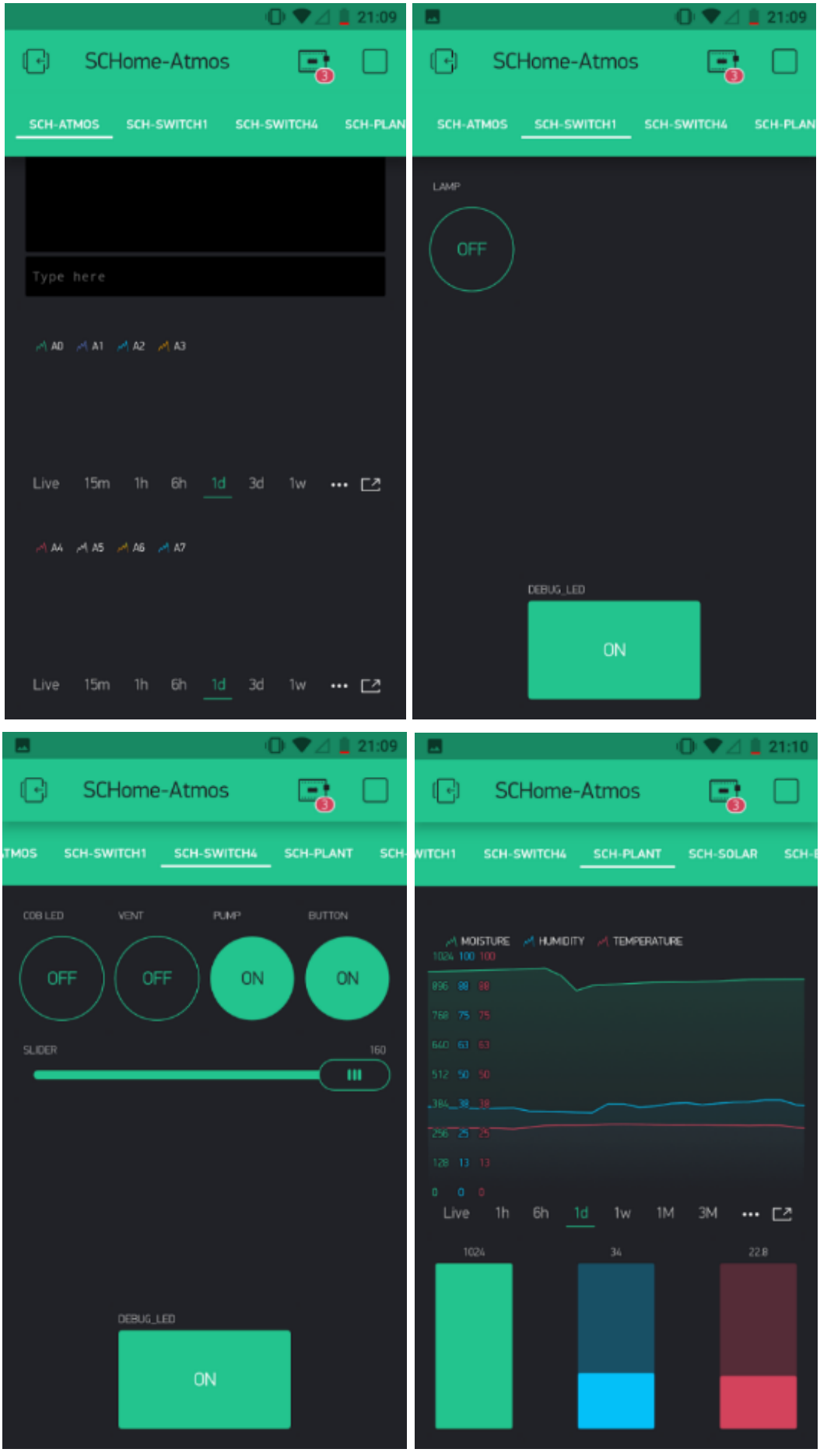


Рисунок А.2 – Интерфейс приложения Blynk App