МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**

Кафедра **«Прикладная математика и информатика»**

02.03.03 Математическое обеспечение и администрирование информационных систем

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

БАКАЛАВРСКАЯ РАБОТА

на тему Разработка модуля защищённой передачи данных в информационной системе «Умный дом»

Студент В.С. Помозов

Руководитель А.В Очеповский

Допустить к защите
Заведующий кафедрой к.тех.н, доцент, А.В. Очеповский _____

«_____» _____ 2016 г.

Тольятти 2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**

Кафедра **«Прикладная математика и информатика»**

УТВЕРЖДАЮ
Зав. кафедрой «Прикладная
математика и информатика»
_____ А.В. Очеповский

«____»_____ 2016 г.

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Помозов Владимир Сергеевич

1. Тема Разработка модуля защищенной передачи данных в информационной системы «Умный дом»

2. Срок сдачи студентом законченной выпускной квалификационной работы 16.06.2016

3. Исходные данные к выпускной квалификационной работе: аналоговые показания в зоне размещение аппаратной платформы, требования к функциональным характеристикам: обеспечение канала связи с web-сервером, криптографическая защита для передачи данных.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов):
Введение
Глава 1 Описание информационной системы «Умный дом»
Глава 2 Моделирование и проектирование информационной системы «Умный дом»
Глава 3 Реализация программного обеспечения для аппаратной платформы информационной системы «Умный дом»
Заключение

5. Ориентировочный перечень графического и иллюстративного материала презентация, включающая блок-схемы работы информационной системы, графики, диаграммы.

6. Дата выдачи задания « 11 » января 2016 г.

Менеджер по проектам ООО «Сто линий» _____ С.А. Мальцев

Руководитель выпускной квалификационной работы _____ А.В. Очеповский

Задание принял к исполнению _____ В.С. Помозов

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**

Кафедра **«Прикладная математика и информатика»**

УТВЕРЖДАЮ
Зав. кафедрой «Прикладная
математика и информатика»
_____ А.В. Очеповский

«____»_____ 2016 г.

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

СтудентаПомозов Владимир Сергеевич_____
по теме Разработка модуля защищенной передачи данных в информационной системе «Умный дом»_____

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
Выбор и утверждение темы ВКР	14.01.2016	14.01.2016	Выполнено	
Анализ состояния вопроса	20.02.2016	20.02.2016	Выполнено	
Разработка алгоритма-анализа и передачи информации с датчиков	15.03.2016	15.03.2016	Выполнено	
Анализ алгоритмов шифрования	20.03.2016	20.03.2016	Выполнено	
Программная реализация предложенных решений	29.03.2016	29.03.2016	Выполнено	
Апробация предложенных решений на реальных данных	30.04.2016	30.04.2016	Выполнено	
Оформление текста бакалаврской работы	5.05.2016	5.05.2016	Выполнено	
Подготовка презентации к выступлению	10.06.2016	10.06.2016	Выполнено	
Предварительная защита	14.06.2016	14.06.2016	Выполнено	

ВКР				
Корректировка ВКР согласно сделанным замечаниям	15.06.2016	15.06.2016	Выполнено	
Проверка ВКР в системе «Антиплагиат.ВУЗ»	16.06.2016	16.06.2016	Выполнено	
Сдача пояснительной записки ВКР и реализованного программного приложения	16.06.2016	16.06.2016	Выполнено	

Руководитель выпускной квалификационной работы _____ А.В. Очеповский

Задание принял к исполнению _____ В.С. Помозов

Аннотация

Тема выпускной квалификационной работы: Разработка модуля защищённой передачи данных в информационные системы «Умный дом»

Ключевые слова: информационная система, умный дом, аппаратная платформа

Объектом исследования при написании работы послужил процесс сбора данных о помещении посредством системы «Умный дом».

Предметом исследования работы стала аппаратная часть данного информационной системы, разрабатываемая по заказу компании ООО «100 Линий».

В бакалаврскую работу входит введение, три главы, итоговое заключение.

Во введении раскрывается актуальность исследования по выбранному направлению, ставится проблема, цель и задачи исследования, определяются объект, предмет научных поисков, формулируется гипотеза, ставятся цель и задачи.

В главе первой происходит анализ информационной системы «Умный дом», анализ типовой компании заказчика и рассматривается модель как она есть.

В главе второй осуществлено моделирование используемых решений при создании аппаратной платформы, таких как рассмотрение информационной системы модели, как она должна быть, алгоритма шифрования данных, а также структура аппаратной платформы

В третей главе происходит выбор технических средств для реализации, а также кодирование и тестирование программного продукта

В заключение происходит вывод о проведённой работе.

Объём бакалаврской работы составляет 47 страниц, на которых размещены 32 рисунка и 3 таблицы. При написании выпускной квалификационной работы использовалось 20 источников.

Оглавление

Введение

В настоящее время у множества компаний, связанных с инновационными разработками в сфере IT и робототехники, возникает потребность в разработке информационной системы «Умный дом».

«Умный дом» — это высокотехнологичная система, позволяющая объединить все коммуникации в одну и поставить её под управление искусственного интеллекта, программируемого и настраиваемого под все потребности, и пожелания хозяина.

Проекты «Умный дом» имеют ряд неоспоримых преимуществ, одним из которых является их экономичность. Они позволяют существенно снизить затраты на электроэнергию (экономия до 30%), отопление (50%) и воду (41%). Вопрос экономии обретает особую актуальность ввиду постоянно растущих цен на электроэнергию и газ.

Вопрос рационально выделяемых ресурсов для обслуживания живых зон так же встаёт и у более крупных организаций, таких как жилищно-коммунальное хозяйство. При управлении процессом выделения ресурсов в большом многоквартирный дом, регулировка ресурсов с помощью систем умного дома даже на 10%, может позволить достичь экономии средств в миллионы рублей.

Часто подобные комплексы представляют собой различные аппаратные устройства, делающие акцент на функциональных характеристиках, предоставляя тем самым пользователю более широкий спектр применения конкретного устройства. Не редко разработчики данных комплексов уделяют гораздо меньше внимания системному подходу к комплексу «Умный дом» и безопасности личных данных своих пользователей. В результате чего «Умный дом» становится не средством автоматизации процессов или же оптимизации расхода ресурсов, а легкомысленными устройствами, позволяющими включить чайник с мобильного устройства.

В общем итоге, можно сказать, что **актуальность** выпускной квалификационной работы, является насущная необходимость в экономии

выделяемых ресурсов при обслуживание типовых многоквартирных домов,посредством внедрения информационной системы «Умный дом», которая будет собирать данные о теплообмене в многоквартирном доме.

Объектом данной выпускной квалификационной работы является процесс сбора данных о помещении посредством системы «Умный дом».

В качестве **предмета** выпускной квалификационной работы будет рассмотрена аппаратная часть данного информационной системы, разрабатываемая по заказу компании ООО «100 Линий».

Целью данной выпускной квалификационной работы является реализация программного обеспечения для аппаратной части информационной системы «Умный дом», выявлению актуальных проблем, возникающих при организации системного подхода к комплексу «Умный дом» и обеспечение защиты персональных данных для собственника данного инженерного комплекса.

Для достижения указанной цели поставлены следующие **задачи**: проанализировать существующие аппаратные платформы для информационной системе «Умный дом», для выявления наиболее подходящей платформы, реализовать актуальные методы шифрования, затем произвести тестирование системы и её внедрение.

Выполнение данных задач позволило бы включить, реализуемую в ходе выполнения выпускной квалификационной работы, аппаратную часть информационной системы «Умный дом» в единую систему, состоящую из нескольких схожих аппаратных блоков, нескольких пользователей и единого сервера, а также обеспечить сохранение данных в конфиденциальности. Что в свою очередь могло бы существенно повлиять на точность собираемых данных.

В ходе выполнения выпускной квалификационной работы планируется выполнять анализ литературы и технической документации относительно аппаратных платформ с целью нахождения наиболее оптимальной для разработки программного обеспечения способного выполнять поставленные задачи. Выполнить систематизацию информации относительно систем

шифрования данных с целью сравнить и выявить наиболее подходящий алгоритм шифрования для выполнения поставленной задачи в выпускной квалификационной работе. Произвести теоретический анализ существующих способов передачи данных на сервер.

На данный момент большинство существующих комплексов работают по системе сбора информации о теплообмене с единого узла в доме, находящегося в основном в технических помещениях здания и не обладают информации о конечном результате теплообмена в типовом многоквартирном доме.

Практическая значимость данной выпускной квалификационной работы заключается в том, чтобы усовершенствовать инженерный комплекс «Умный дом», посредством реализации программного обеспечения, позволяющего включать данную аппаратную часть в систему с многими пользователями или устройствами и реализовать защищённую передачу данных на сервер.

Структура выпускной квалификационной работы обусловлена предметом, целью и задачами исследования. Работа состоит из введения, трёх глав, заключения и приложения.

Введение раскрывает актуальность, определяет степень научной разработки темы, объект, предмет, цель, задачи и методы исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе будет произведён анализ предметной области, будет выбрана для рассмотрения типовая организация которая будет использовать создаваемую информационную систему «Умный дом».

Во второй главе будет выполнено моделирование и проектирование информационной системы «Умный дом»

В третьей главе будут определены аппаратные решения для выполнения поставленных задач, и рассмотрены уникальные алгоритмы программного продукта, а также его тестирование и внедрение.

В заключении подводятся итоги исследования, формулируются выводы по, рассматриваемой в ходе выполнения работы, теме.

Глава 1 Описание информационной системы «Умный дом»

1.1 Общие сведения о информационной системе «Умный дом»

Система «Умный дом» - продукт развития высоких технологий, способный объединить все устройства и коммуникации в единую структуру для максимально простого управления. Основную управляющую функцию берёт на себя искусственный интеллект, имеется возможность его настройки и программирования для точного исполнения функций, что обеспечивает в доме уют и комфорт.

Технология широко распространена во всём мире, многие граждане развитых стран не могут представить без этой системы свою жизнь, однако для жителей России термин «Умный дом» до сих пор остаётся неизвестным или малопонятным. Существует множество причин, по которым подобные системы не находят широкого распространения, основной из них можно назвать слабое понимание сути использования технологии, приносимой ей экономии.

Под архитектурой системы умного дома следует понимать совокупность модулей необходимых для её полноценной работы. Данные модули должны быть связанны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы.

Система состоит из отдельных элементов, некоторые из них относительно хорошо распространены, особенно в квартирах жителей крупных мегаполисов, среди них световые датчики движения и устройства за контролем над протечками воды. Всё более популярны становятся и другие важные элементы, среди которых разнообразные устройства для контроля за безопасностью, управления световыми приборами, домашние кинотеатры, системы распределения видео и звука внутри дома и за его пределами.

Система состоит из отдельных элементов, некоторые из них относительно хорошо распространены, особенно в квартирах жителей крупных мегаполисов, среди них световые датчики движения и устройства за контролем над протечками воды. Всё более популярны становятся и другие важные элементы,

среди которых разнообразные устройства для контроля за безопасностью, управления световыми приборами, домашние кинотеатры, системы распределения видео и звука внутри дома и за его пределами.

Цели использования продуктов данной сферы достаточно разнообразны. Система помогает создать возможность для эффективного управления освещением, отоплением, электричеством, водным обеспечением, видео наблюдением, вентиляцией, кондиционированием и другими жизненно необходимыми функциями. После её применения пользователь получает исчерпывающую информацию о доме, включая работоспособность тех или иных приборов и коммуникаций [5].

Система может найти применение в различных областях, она в равной степени эффективна для частных строений, коттеджей, офисных помещений, многоквартирных домов. Универсальность является важнейшим преимуществом технологии.

1.2 Модель работы информационной системы AS-IS

В роли потенциального заказчика продукта «Умный дом», разрабатываемого в ходе выпускной квалификационной работы, рассматривается типовая организация Жилищно-коммунального хозяйства (в дальнейшем «ЖКХ»).

ЖКХ представляет собой комплекс экономических отраслей, занимающийся обеспечением работоспособности инфраструктуры в различных поселениях. Рассматриваемый нами комплекс проводит обслуживание помещений разного назначения и важных городских объектов. В число обслуживаемых объектов включаются типовые многоквартирные дома, объединённые в жилые районы [14, 20].

Более подробная структура типовой организации ЖКХ представлена на рисунке 1.1.

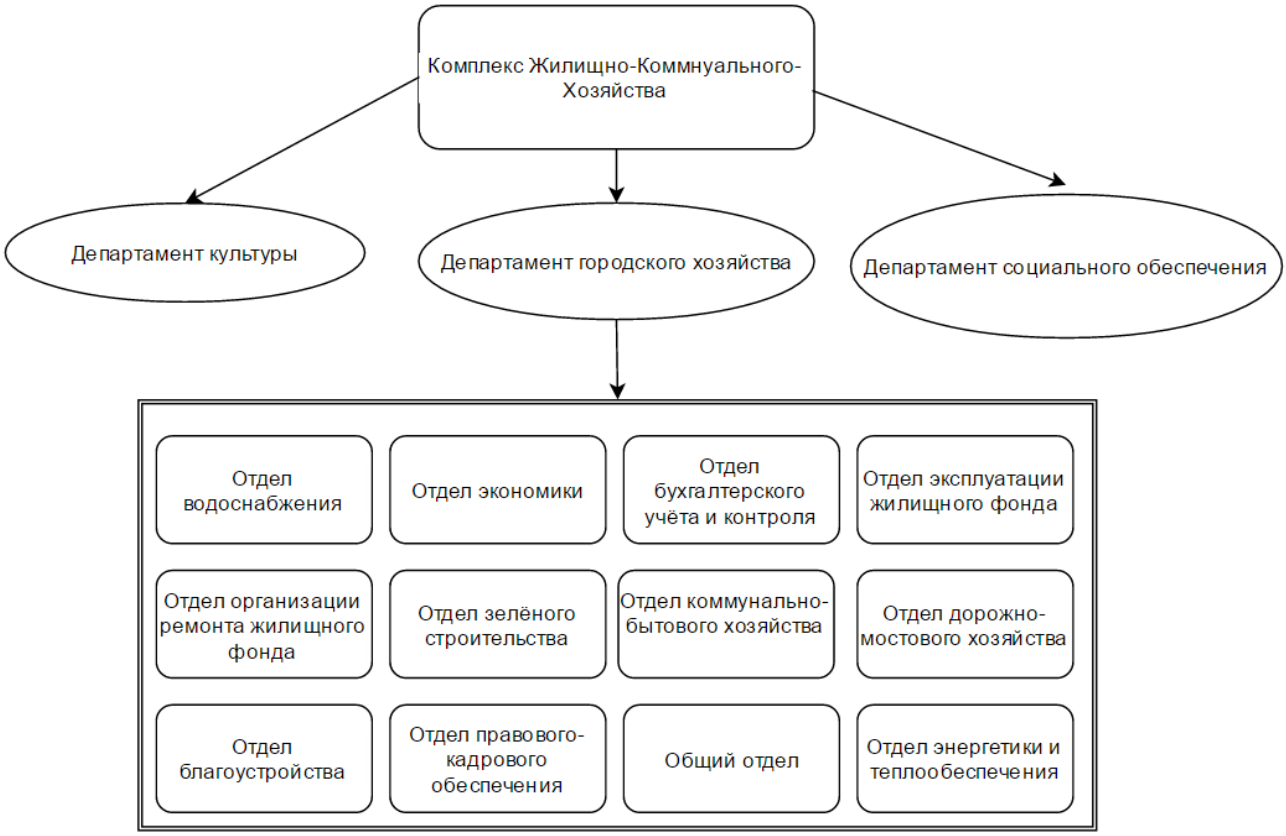

Рисунок 1.1 – Структура типового ЖКХ

В общем случае типовой ЖКХ выполняет следующие задачи:

- энергоснабжение – обеспечение обсуживаемых помещений соответствующим типом энергии (газ, электричество);
- теплоснабжение – обеспечение обсуживаемых помещений поставками горячей воды и тепловой энергии, обеспечение работы котельных;
- водоснабжение – прокладка, ремонт и обслуживание водопроводных труб, водозабор и отчистка воды в многоквартирных домах;
- составление, хранение и использование цифровых карт города, включающих: сети коммунальных ресурсов и канализации, транспортные пути, здания, сети коммуникаций и др.;
- системы учета и расчетов за потребление коммунальных ресурсов и жилищно-коммунальных услуг, включая интернет порталы ГИС ЖКХ России в каждом регионе с личными кабинетами каждого пользователя ресурсами и услугами ЖКХ;

— вентилирование – централизованное кондиционирования воздуха и вентиляция;

— вывоз мусора;

— лифтовое хозяйство – обеспечение стабильной работы лифтов в обслуживаемых помещениях;

— капитальный ремонт и модернизация зданий;

— уборка дорог и содержание прилегающих к дому территорий.

ЖКХ выполняет, как приведённые в списке мероприятия по благоустройству жилых зон, так и множество других.

В текущий момент времени в ЖКХ имеется отдел Энергетики и обращений по вопросам теплоснабжения, выполняющий широкий круг задач. В сферу его деятельности включено обеспечение квартир электроэнергией. Ещё одной важной задачей является сбор и анализ большого количества сопутствующей выполнению основных функций информации. Качественная терморегуляция требует проведения всестороннего обслуживания, важно вовремя выявить необходимость профилактических операций.

Данный департамент занимается следующими задачами в процессе своей работы:

1) Создание и воплощение стратегии развития городского хозяйства на основе прогрессивных технологий.
2) Формирование муниципального заказа в сферах ЖКХ, электроснабжения, благоустройства, озеленения, утилизации отходов; выбор исполнителей и осуществление контроля за исполнением заказа на территории городского округа.
3) Создание конкурентной среды в сфере оказания жилищно-коммунальных услуг.
4) Разработка и осуществление мер по охране окружающей среды в целях уменьшения негативного воздействия вредных выбросов на окружающую среду.

5) Обеспечение нормативно-правовой базы для деятельности по охране окружающей среды в рамках имеющихся полномочий.

Организация системы контроля за соблюдением городскими жителями требований муниципальных нормативных актов в сфере вывоза, утилизации и переработки отходов и организации, специально отведенных для этих целей мест

Проведение мероприятий, направленных на развитие у жителей бережного отношения к природе, повышение их осведомленности в вопросах экологии

Контроль за природоохранной деятельностью в пределах городского округа. Исходя из данных задач, можно сделать вывод о том, что внедрение информационной системы «Умный дом» может благотворно сказаться на деятельность данного департамента. В частности, внедрение данной информационной системы будет производится на типовом отделе «Энергетики и обращений по вопросам теплоснабжения». Данный отдел выбран в качестве наиболее подходящего подразделения департамента ЖКХ для внедрения разрабатываемой информационной системы[14].

В данный момент отдел «Энергетики и обращений по вопросам теплоснабжения» занимается помимо прочего, обеспечением многоквартирных домов электроэнергией, а также сбором информации о многоквартирных домах, для проведения соответствующих мероприятий по работам, связанным с терморегуляций в многоквартирных домах.

На текущий момент система сбора и обработки температурных данных о домах происходит по следящим принципам, представленным на рисунке 1.2 и 1.3 в виде модели «AS-IS» [1]. Система в данный момент использует единственный контроллер, служащий для съёма показаний относительно передачи тепла в многоквартирные дома. Данная технология отличается простотой, она осуществляет температурный контроль в соответствии с принятыми нормативными документами, так что терморегуляция полностью

выполняет требования действующего регламента. Но такая система не в состоянии выполнять ряд важных функций, она не соответствует современным требованиям, следствием её применения являются лишние материальные затраты и дискомфорт жильцов.

Рисунок 1.2 – Концептуальная модель сбора данных «AS-IS»

Несовершенность технологии обусловлена следующими причинами:

- отсутствует контроль за распределением тепловых ресурсов по различных участках здания;
- статистика после используемого сбора информации имеет серьёзные погрешности;
- отсутствуют средства обработки полученной информации, комфортные для работы пользователя.

Температура снимается на узлах передачи тепловой энергии, что не позволяет получить данные относительно показателей температуры в квартирах потребителей. Реальные и получаемые показатели приводят к серьёзным

коллизиям, когда при заморозках в помещения поступает недостаточно тепла, а в жаркую погоду отопление выдаёт большое количество энергии.

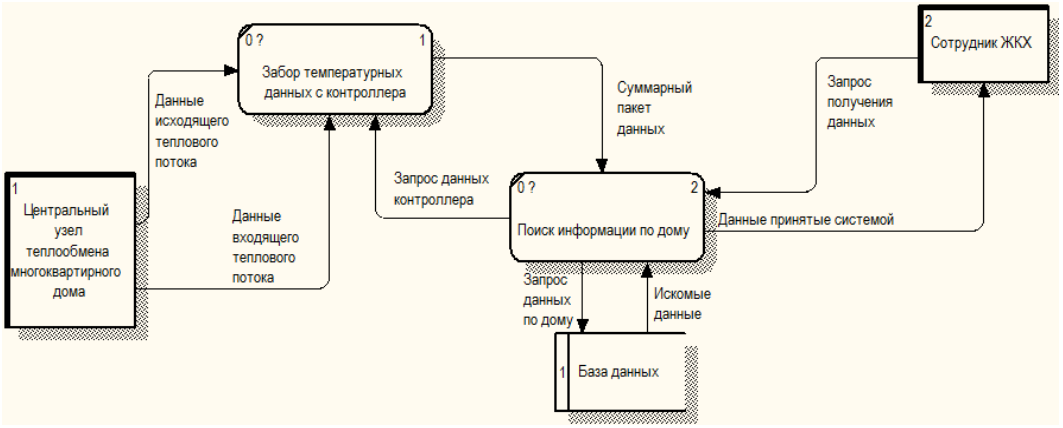

Рисунок 1.3 – DFDдиаграмма информационной системы «AS-IS»

Обе проблемы приносят жильцам сильное ощущение дискомфорта, к тому же за лишнее тепло следует серьёзная переплата.

Для решения данной проблемы необходимо применять рациональное распределение ресурсов, что при текущей системе сбора и анализа информации является невыполнимой задачей. Отсутствие необходимой информации не позволяет реализовать многие важные функции, поток выделяемой энергии носит однородный характер, он распределяется равномерно, хотя для разных участков требуется различное количество тепла.

Недостаточный сбор статистики в помещениях не позволяет собирать данные для более рационального распределения выделяемых ресурсов.

Незащищённый способ передачи данных предоставляет возможность нанести вред жильцам дома злоумышленниками.

Все недостатки системы, использующейся в текущий момент времени, объясняются её техническим несовершенством. Технология не позволяет реализовать современную информационную среду, которой было бы удобно пользоваться операторам и потребителям. Собираемые с одного контроллера данные дают весьма опосредованную информацию относительно отдельных участков дома, ведение статистики с помощью подобной системы не представляется возможным.

В результате анализа недостатков текущей модели информационной системы для сбора показаний в многоквартирном доме можно сделать вывод о необходимости изменения реализации метода сбора информации и самой информационной системы.

1.3 Формирование требований к новой информационной системе

В основном, выделенные проблемы в модели которая используется в данный момент касается именно прибора снимающего показания в многоквартирном доме. Для улучшения системы и решения поставленных проблем следует разработать технологический комплекс средств по улучшению данной системы. В текущий момент наиболее удачным решением может является дополнение уже готовой системы – новой, которая в свою очередь, будет работать вместе со старой, дополняя нехватку данных получаемых ранее.

Технологический комплекс «Умный дом» станет лучшим решением в сложившейся ситуации, его информационные продукты отличаются высокой степенью наглядности, аппаратная часть и Web-сервер постоянно сохраняют связь, поэтому обработка данных происходит в режиме реального времени.

Принципиальное отличие от используемой ранее системы заключается в совершенно новом подходе к обработке информации, появляется возможность размещать данные в открытом для конечных потребителей доступе. Каждый житель дома, установивший у себя подобную систему, может контролировать свои расходы. Большое количество собираемых данных позволяет добиться высокой точности результатов.

Помимо получение конечных данных о температуре внутри жилой зоны, гибкость аппаратной части информационной системы «Умный дом» позволят собирать множество данных для статистики и при правильной организации полученных данных можно будет сделать выводы для более рационального распределения выделяемых ресурсов на многоквартирный дом.

На основании проведённого выше анализа представляется возможным выдвинуть функциональные требования к аппаратной части разрабатываемой информационной системе:

— аппаратная часть информационной системы должна быть размещена в жилой зоне;
— аппаратная часть должна обладать функционалом для сбора температурных данных о помещении;
— аппаратная часть должна иметь достаточно гибкий функционал, для осуществления сбора дополнительной информации о помещении;
— аппаратная часть должна иметь достаточно вычислительных мощностей для обеспечения шифрования данных;
— аппаратная часть должна обладать подключением к внешней сети.

В результате выдвинутых функциональных требований следует сделать выбор о том каким образом лучше осуществить подход для разработки аппаратной части для информационной системы «Умный дом».

В данный момент существует множество готовых решений по технологии умный дом, в качестве решения поставленных задач можно использовать уже готовый вариант с целью дальнейшего преобразования его под нужды информационной системы.

1.4 Выбор метода для создания аппаратной части информационной системы «Умный дом»

В итоге следует сделать выбор о том, как доработать аппаратную часть для усовершенствования текущей информационной системы. Следует ли совершенствовать уже полностью готовую аппаратную платформу или же разрабатывать свою. Для более наглядного выбора средства разработки следует рассмотреть таблицу 1.1, в которой производится сравнение методов улучшение информационной системы «Умный дом» в зависимости от требований к функциональной части.В результате анализа приведённой

таблицы следует сделать вывод о необходимости разработки своей аппаратной части для информационной системы «Умный дом»

Таблица 1.1 – Сравнение методов улучшения системы «Умный дом»

Требования к модулю информационной системы «Умный дом»	Готовые решения для систем «Умный дом»	Разработка своей аппаратной платформы для системы «Умный дом»
Функционал сбора температурных данных	Данный функционал присутствует, но является крайне трудной задачей настроить нужный формат сбора данных в соответствии с временными рамками.	Реализовать сбор температурных данных можно в любом виде удобном для информационной системы.
Гибкость настройки входных данных	Не представляется возможным добавление нужных датчиков по сбору информации или отключить лишние	Можно подключить и запрограммировать практически любой датчик совместимой с основой аппаратной платформы.
Гибкость настройки выходных данных	В большинстве своём пересылка данных идёт на собственного сервера сторонних систем	Достаточная гибкость при выборе способа отправки данных и их типа
Шифрование	Либо отсутствует, либо идёт исключительно в совокупности с расшифровкой на собственных серверах	Можно запрограммировать любой подходящий способ шифрования данных
Цена	Высокая стоимость аппаратной платформы для информационной системы	Соотношение цена/функционал одно из самых низких в сегменте

В результате анализа, произведённого в первой главе, были выявлены недостатки системы «как она есть», были сформулированы требования к новой информационной системе, а также был выбран метод изменения данной информационной системы.

Глава 2 Моделирование и проектирование информационной системы «Умный дом»

2.1 Общая архитектура информационной системы «Умный дом»

Под архитектурой системы «Умный дом» следует понимать совокупность модулей необходимых для её полноценной работы. Данные модули должны быть связанны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы. Общая структура данной информационной системы, отображены с помощью концептуальной модели, представленной на рисунке 2.1 и DFDмодели на рисунке 2.2.

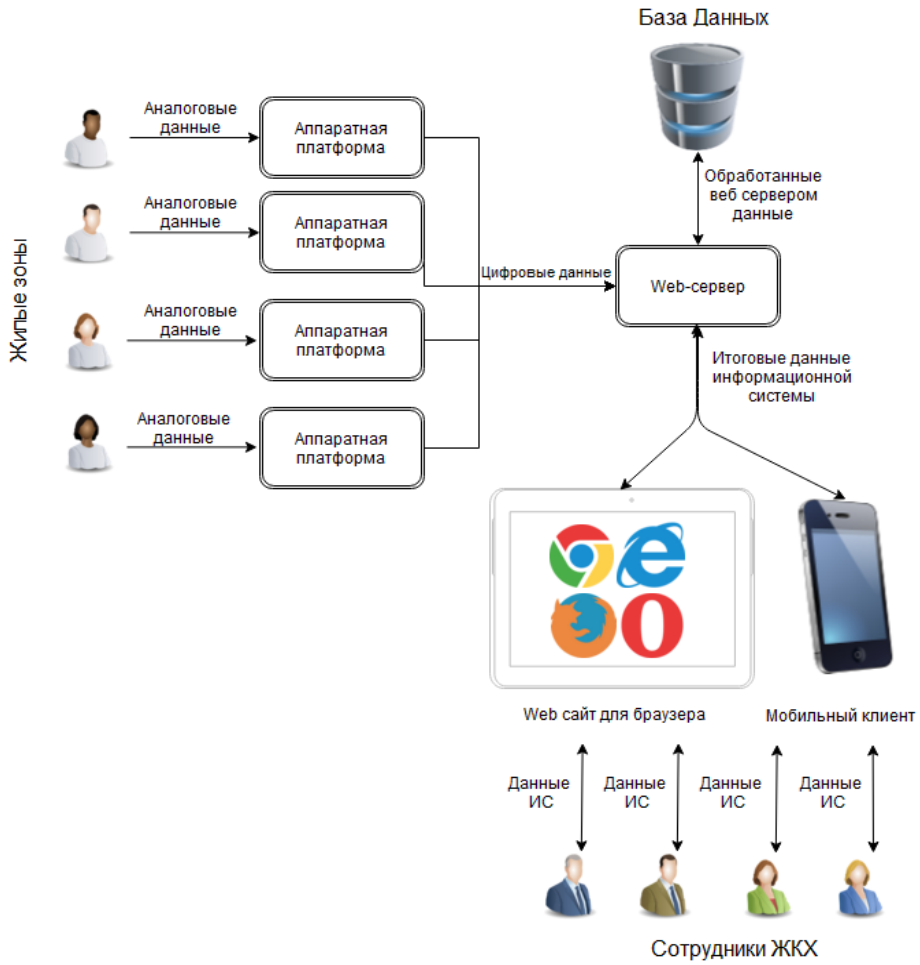

Рисунок 2.1 – Структура разрабатываемойинформационной системы «Умный дом»

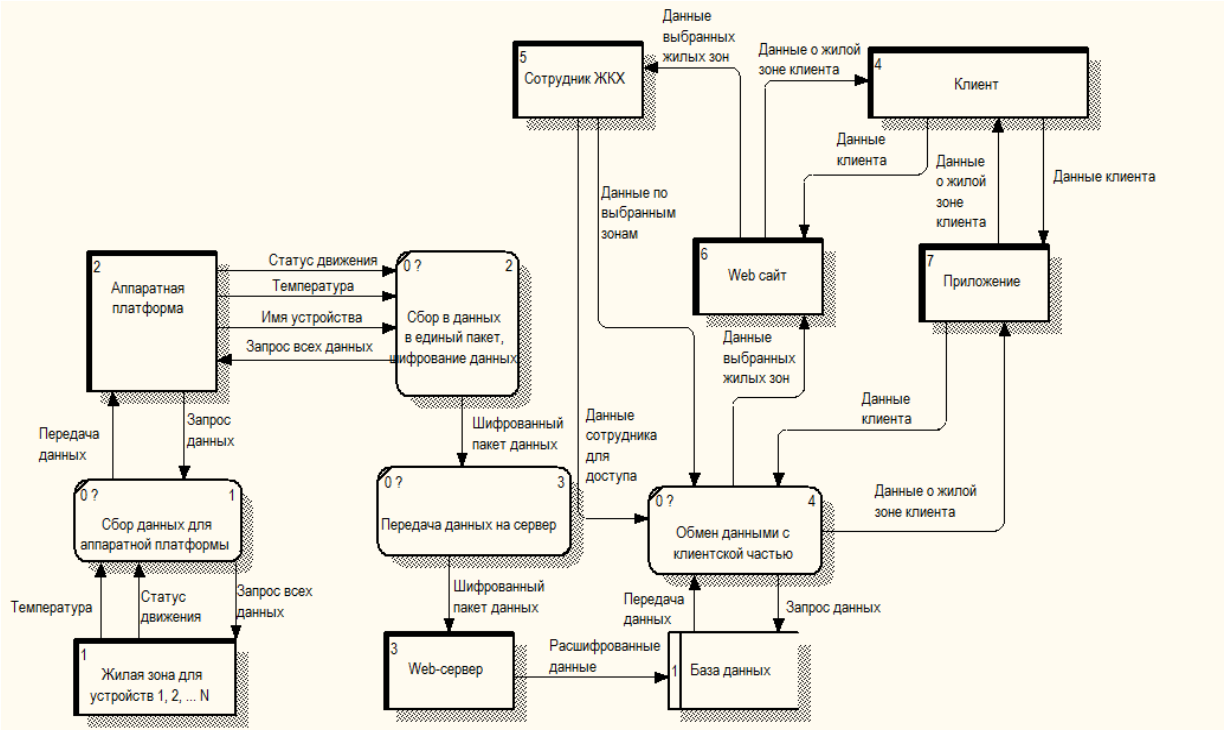

Рисунок 2.2 –Модель DFDинформационной системы «Умный дом, TOBE»

Как видно из представленныхмоделей информационной системы «Умный дом» её можно условно разделить на три основных модуля: аппаратная платформа, сервер, клиент.

Каждый из этих модулей берёт на себя равную по значимости роль в работе всей информационной системы.

Аппаратная платформа представлена контроллером, размещенным в жилой зоне многоквартирного дома, обеспечивающим контроль итоговой температуры, вместо сбора температурных данных на узле в технической части здания. Точность сбора показаний будет зависеть от количества аппаратных платформ размещённых в доме, но даже одна подобная аппаратная платформа может помочь передать информацию о нерациональном теплообмене внутри многоквартирного дома.

В результате работы контроллера размещённого в жилой зоне происходит процесс перевода таких аналоговых данных, как температура или влажность в

цифровые данные для их дальнейшей шифровки, формулировки исходящего пакета в требуемом для сервера виде и их дальнейшей передачи.

Web-сервер представлен программным продуктом, размещённом на удалённом сервере. Данный программный продукт обеспечивает приём входящих от аппаратной платформы данных. Расшифровку этих данных и заполнение полученной информации в базу данных сервера. Так же на сервере может производиться различный анализ данных с целью выявления проблем с теплообменом внутри многоквартирного дома или же анализ других полученных от аппаратной части данных в зависимости от их наличия.

Помимо описанных задач Web-сервер выполняет роль промежуточного звена между клиентской частью и данными прошедшими обработку сервером. Он обрабатывает запросы пришедшие с клиентской части и формирует данные для передачи формам веб-страниц или приложений.

Клиентская часть в информационной системе представлена сайтом

2.2 Общая архитектура аппаратной части системы «Умный дом»

Под архитектурой аппаратной части системы «Умный дом» следует понимать совокупность модулей необходимых для полноценной работы аппаратной платформы. Данные модули должны быть связанны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы. Концептуальная модель аппаратной платформы для информационной системы «Умный дом» представлена на рисунке 2.3.

В результате анализа, произведённого ранее, аппаратной частью системы умного дома будет являться одно полностью автономное устройство, в нём будет весь необходимый набор датчиков, предназначенный для сбора статистики о помещении, в котором оно размещено [8, 10, 17]. Так же в нём будут находиться некоторые модули предназначенные для обеспечения безопасности самого помещения.

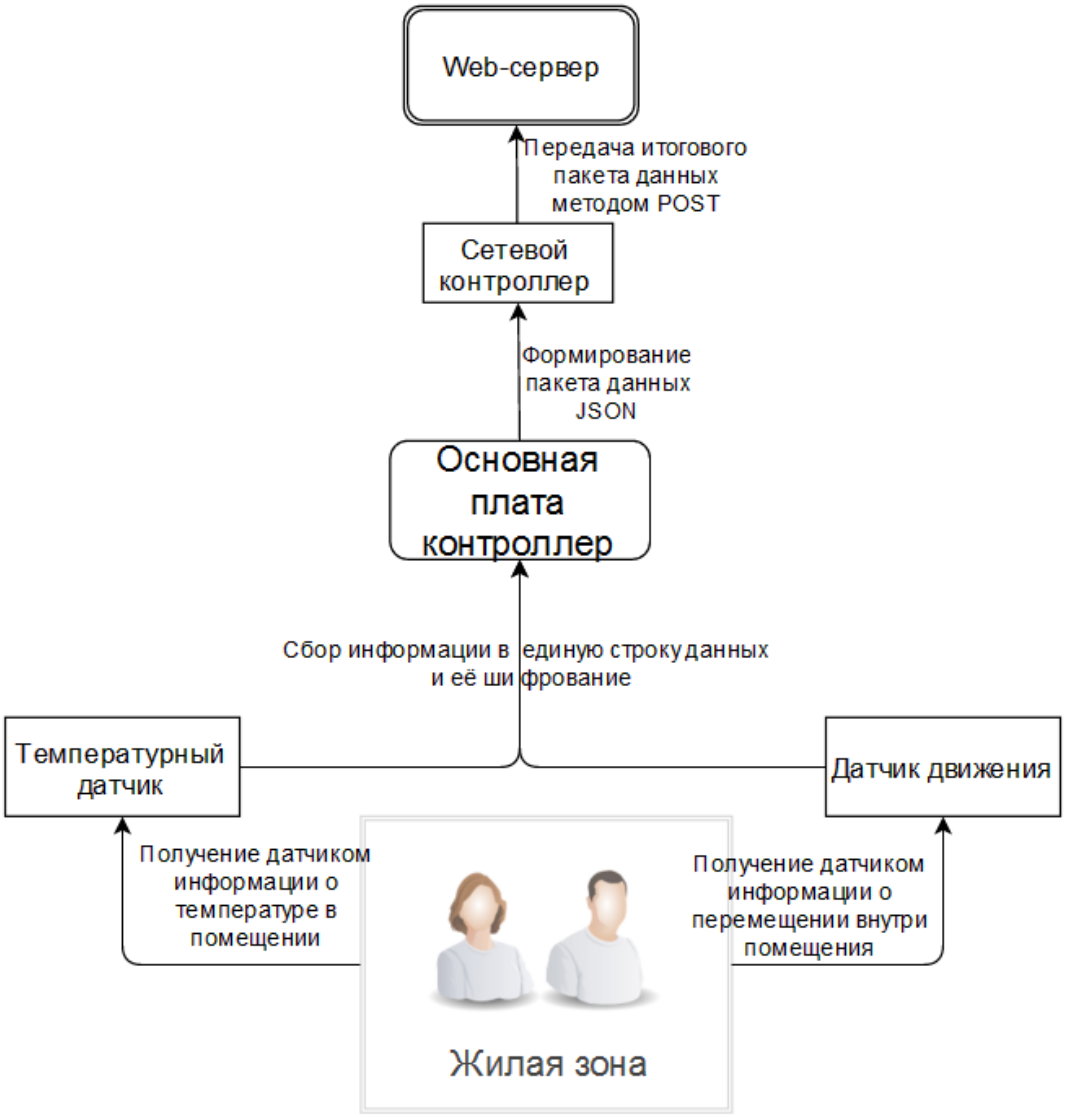

Рисунок 2.3 – Концептуальная модель для аппаратной платформы системы «Умный дом»

Автономность данной системы будет создана благодаря наличию независимого источника питания, таким образом устройство должно получать энергию не только от домашней сети, но и от портативных источников энергии, таких как батареи или аккумуляторы.

Так же устройство должно быть самодостаточным в отношении связи и даже при отсутствии доступа к сети, устройство как минимум должно сигнализировать конечного клиента или сервер о неисправности сети [2]. Данный критерий позволяет заключить, что устройство должно содержать в

себе GSM модуль для передачи информации, осуществления SMS передач или же звонка.

Сбор статистики о помещении, скорее всего сбор статистики будет производится по принципу размещённых на самом устройстве температурных датчиков, датчиков влажности, а потенциально и датчиков, отвечающих за контроль над расходом электроэнергии.

Защиту помещения в свою очередь будут обеспечивать датчики движения, работающие по принципу системы безопасности, будут так же и датчики, которые будут контролировать безопасность самого помещения, например, датчики протечек и возгораний, аналогичную же роль могут выполнять датчики влажности и температуры при соответствующим программном обеспечении.

Аппаратная часть системы «Умный дом» должна быть централизованной — должно иметься главное устройство для выполнения поставленных задач сбора, шифрования и отправки данных, и набор различных датчиков, подключенных к этому главному модулю. Также допускается наличие вспомогательного модуля, который будет включаться в работу, когда главное устройство неработоспособно вследствие неисправности или обновления ПО. Вспомогательный модуль может обеспечивать возможность управления всеми периферийными устройствами по одному, заранее заданному алгоритму, доступ человека к функциям управления системой не требуется и будет невозможен до ввода в строй основного компьютера.

Температурный датчик должен контролировать температуру помещения автоматизации, так же может потребоваться добавление датчика влажностис цельюинформирования о состоянии возможности протечки, а также потенциально можно внедрить датчик расхода электроэнергии.

Модуль контроля наличия людей на объекте автоматизации должен информировать сервер или конечного пользователя о активации датчика движения, в зависимости от чего web-сервер будет принимать решение о

дальнейших действиях либо собирать статистику с целью оптимизации затрат ресурсов.

Модуль осуществления интерактивной связи с сервером или конечным пользователем, должен информировать о технических сбоях или других критических событиях посредством SMS-сообщений или телефонных звонков.

Модуль управления системами связи с внешним миром должен осуществлять доступ аппаратной части во внешнюю сеть для связи оборудования с сервером.

Входные данными аппаратной части устройства будут являться данные окружающей среды помещения автоматизации, с помощью ряда датчиков эти аналоговые данные будут преобразованы и отправлены на сервер цифровым пакетом предварительно зашифрованных данных в виде выходных данных устройства.

2.3 Описание алгоритм шифрования данных в информационной системе «Умный дом»

Так как в систему внесены персональные данные пользователей и некоторая информация по жилой площади собираемая в реальном времени, сохранение этих данных в конфиденциальности является одной из важнейших задач.

Шифрование используется для хранения важной информации в ненадёжных источниках и передавать свои незащищенные каналы. Эта передача данных представляет собой два взаимно обратный процесс:

— перед передачей данных по линии связи или перед хранением, они подвергаются кодированию;

— чтобы восстановить исходные данные из зашифрованной процедуры к ним применяют дешифрацию.

Шифр представляет собой пару алгоритмов, реализующих каждое из этих изменений. Данные алгоритмы применяются к объекту шифрования с использованием ключа.

На данный момент существует множество методов шифрования. В основном эти методы разделяются, в зависимости от структуры используемых ключей для методов симметричного и асимметричного методов. Кроме того, методы шифрования могут иметь различные криптографические и другого входных данных процесса - блочные шифры и потоковые шифры [13].

Для решения поставленных задач был выбран симметричный, блочный тип шифрования AES. Его общий алгоритм представлен на рисунке 2.4.

Данный алгоритм AdvancedEncryptionStandard, хорошо зарекомендовал себя в плане сохранности информации.

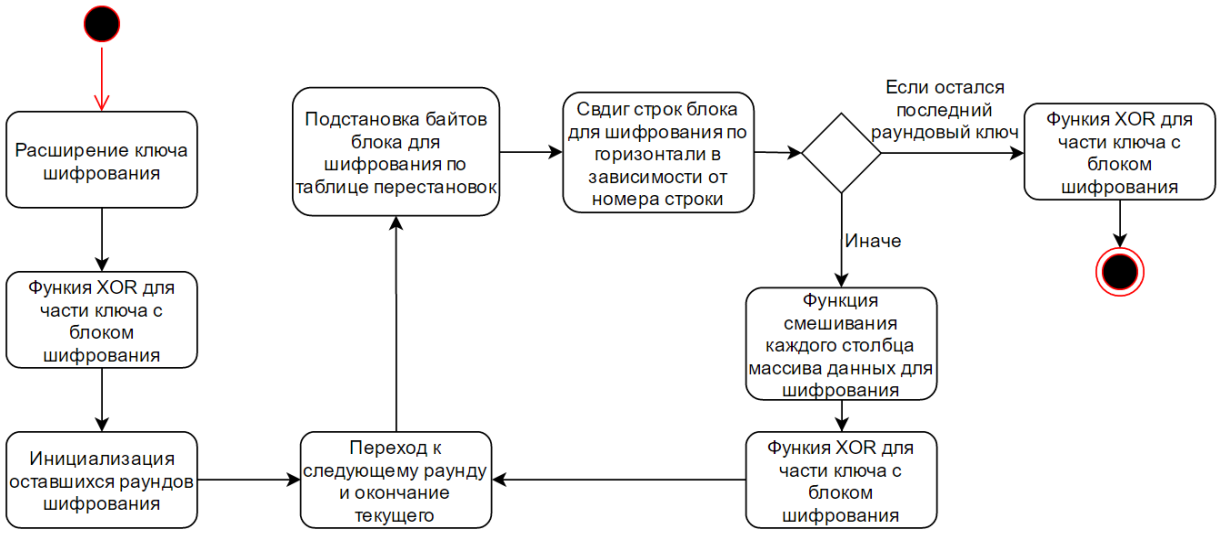

Рисунок 2.4 – Алгоритм шифрования AES

Размер входящих блоков равен 128 битам, а размер ключа может быть 128 бит, 192 бита или же 256 бит. Применимо к нашей задачей размерности ключа и блока данных будет составлять всего 32 бита, алгоритм допускает и такую длину ключа и ключевого файла в результате чего происходит значительно меньше итераций и, следовательно, уменьшается нагрузка на вычислительные мощности аппаратной платформы.

В результате изменения длинны ключа, уменьшится и крипто стойкость алгоритма шифрования данных, но, если взглянуть на таблицу 2.1 можно сделать вывод о том, что представленное количество вариантов ключа обеспечит должный уровень защиты от прямого перебора.

Таблица 2.1 – Зависимость количество вариантов ключа, от размера ключа

Размер ключа в битах	Количество возможных вариантов
1 бит	2
2 бита	4
4 бита	16
8 бит	256
16 бит	65536
32 бит (используемый в ВКР)	$4.2x10^{9}$
56 бит (DES)	$7.2x10^{16}$
64 бит	$1.8x10^{19}$
128 бит (AES)	$3.4x10^{38}$
192 бит (AES)	$6.2x10^{57}$
256 бит (AES)	$1.1x10^{77}$

Как видно из представленной блок схемы алгоритма шифрования, изменение количества бит в данных для шифрования и ключе может существенно сократить количество итераций, а, следовательно, снизить нагрузку на аппаратную честь. В тоже время выбранная длина ключа обеспечивает хороший уровень защиты данных, позволяющий избежать взлома методом прямого перебора ключа. А сам алгоритм шифрования AES (Rijndael) в 2009 признан самым распространённым алгоритмом шифрования и вплоть до текущего момента не зафиксирован не единый случай взлома данного алгоритма при длине ключа в 128 бит и более.

В результате моделирования произведённого во второй главе, была проанализирована архитектура информационной системе «как должно быть»,

были сформулированы требования к структуре аппаратной части информационной системы. Так же был проанализирован используемый алгоритм шифрования и выявлены его преимущества.

Глава 3 Реализация программного обеспечения для аппаратной платформы информационной системы «Умный дом»

3.1 Выбор аппаратной платформы для разработки системы «Умный дом»

В настоящее время инженеры-любители, увлекающиеся конструированием в духе «сделай сам», не испытывают недостатка в специальных устройствах, позволяющих оборудовать любые изделия хорошей электронной начинкой. Одно из самых популярных решений такого рода — это дешевый микрокомпьютер Raspberry Pi, система на кристалле (SoC), использующая полнофункциональную версию ОС Linux (этот компьютер разрабатывался в обучающих целях). Также существует платформа Arduino — микроконтроллер, обладающий внушительной технической поддержкой (целое сообщество разработчиков) и имеющий сотни схем-расширений (так называемых «шилдов»).

На самом деле, ниши применения обоих устройств отличаются, у каждого из них есть свои достоинства и недостатки, а также спектр задач, которые решаются по-разному [7].

Рассмотрим следующие аппаратные платформы, подходящие в качестве аппаратной платформы информационной системе: Arduino и RaspberryPi

Arduino представляет собой электронный конструктор – средство для проектирования электронных устройств, которым присуще более тесное взаимодействие с материальным миром, чем обычным современным компьютерам (действующим в рамках виртуальной реальности). Данное средство является платформой с открытым исходным кодом, разработанной для «physical computing». Конструктор строится на простой печатной плате с современной средой для создания ПО.

Сфера применения этого средства – разработка технических устройств, оснащенных функцией подключения широкого спектра датчиков и способных

принимать от них сигналы, а также позволяющих управлять другими техническими средствами.

Устройства, спроектированные конструктором, способны функционировать без сторонних программ либо при взаимодействии с приложениями, установленными на ПК. Возможна как сборка плат собственными силами, так и приобретение готовых вариантов. Рассматриваемая среда распространяется свободно.

Язык программирования Arduino схож с решением для «physical computing» под названием Wiring. Основой последнего служит среда для создания ПО под названием Processing.

Для «physical computing» разработано большое количество микроконтроллеров и платформ. Данные устройства собирают разрозненные данные о программировании воедино, предоставляя их в удобной для использования сборке.

Использование рассматриваемого программного средства позволяет получить наилучшие результаты, затратив при этом минимум средств.

Невысокая цена. Стоимость плат Ардуино ниже, чем аналогичных платформ. Наиболее бюджетная версия может быть собрана самостоятельно, решения в сборке обойдутся менее, чем в $50.

Кроссплатформенность. Arduino совместим с операционными системами Windows, Linux и MacOs, тогда как большая часть микроконтроллеров могут работать лишь с Windows.

Простота и понятность среды программирования. Работа со средой Arduino окажется несложной даже начинающим пользователям. Она создана на языке программирования Processing; это обеспечивает удобство для преподавателей, поскольку обучающиеся, использующие его, получают и навыки работы с Arduino.

Рассматриваемое ПО имеет открытый исходный код и возможность расширения, что позволяет опытным пользователям дополнять его. Для дополнения могут быть использованы библиотеки C++. Те, кому необходимо

разобраться в технических аспектах, могут перейти на язык AVR-C, на котором основан C++. Это означает возможность добавления кода из среды AVR-C в ПО Arduino.

Дополнительное преимущество обеспечивают и микроконтроллеры, лежащие в основе Arduino – они обладают открытыми принципиальными схемами и могут быть расширены. Лицензия Creative Commons, с которой они выпускаются модульные схемы, позволяют опытным специалистам расширять и дополнять их, разрабатывая новые версии. Создание тестовых образцов доступно и для рядовых пользователей – это позволит сэкономить средства и понять принцип работы.

Raspberry Pi в свою очередь, является полнофункциональным компьютером. Он обладает всеми атрибутами настоящего компьютера: выделенным процессором, памятью и графическим драйвером для вывода через HDMI. На нем даже работает специальная версия операционной системы Linux. Поэтому на Raspberry Pi легко установить большинство программ для Linux. Стоит немного потрудиться — и Raspberry Pi можно использовать как полноценный медиа-сервер или эмулятор видеоигр.

Хотя в Pi и отсутствует внутреннее хранилище данных, на этом компьютере можно использовать смарт-карты в качестве флэш-памяти, обслуживающей всю систему. Таким образом, можно быстро выгружать для отладки различные версии операционной системы или программных обновлений. Поскольку это устройство обеспечивает независимую соединяемость по сети, его можно настраивать и для доступа по SSH, либо пересылать на него файлы по протоколу FTP.

Рассмотрев обе аппаратные платформыArduino и RaspberryPi можно сделать вывод о том, что для реализации системы «умный дом» платформа Arduino является наиболее приемлемым выбором. Данный тезис может подтвердить таблица 3.1.

Таблица 3.1 – Сравнение аппаратный платформ RaspberryPiи Arduino

Параметр для сравнения	Raspberry Pi	Arduino
Стоимость	3-5 тысяч рублей	0.5-3 тысячрублей
Выбор датчиков и модулей расширения	Количество модулей расширений и датчиков довольно ограничено	Огромный выбор модулей расширения и датчиков различного функционала
Общий функционал платформы	Платформа имеет лишний для реализуемых задач функционал, который увеличивает вероятность сбоя всей системы	Функционал идеально подходит для реализации поставленных задач
Комьюнити пользователей платформы	Довольно узкое комьюнити российских пользователей, что делает некомфортной работу над поставленной задачей.	Широкое комьюнити пользователей в российском сегменте, что позволяет контактировать со сторонними разработчиками и совместно искать пути решения возникающих проблем
Реализации проектов в открытом доступе	Большинство проектов предоставляются по технологии платного распространения	Много открытых совместных проектов, элементы которых можно в дальнейшем использовать

Вывод о преимуществах платформы Arduino сделать на основании следующих факторов:

— меньшая стоимость аппаратной платформы;

— больший выбор датчиков и различных модулей совместимых с платформой Arduino;

— отсутствие лишнего функционала, имеющегося в платформе RaspberryPi, дающую возможность возникновения дополнительных ошибок не связанных с работой основных модулей;

— огромное количество информации о платформе Arduino в свободном доступе;

— большее сообщество пользователей данной аппаратной платформы.

В ходе выполнения бакалаврской работы будут использованы такие платы расширения как:

— ESP8266 – сетевой модуль для аппаратной платформы Arduino, обеспечивает связь аппаратной платформы с сетью посредством беспроводной связи Wi-Fi;

— HC-SC501 – данный датчик является инфракрасным датчиком движения;

— DHT22 – датчик влажности и температуры;

— SIM9000 – GSM датчик.

3.2 Выбор среды программирования аппаратной платформы «Умный дом»

Интегрированная среда разработки, IDE (англ. Integrated development environment) — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО). Выбор среды разработки при написании программ для Arduino во многом определяет функционал, которым будет обладать программа. В зависимости от IDE меняется формат написание программы. Различные среды разработки под Arduino требуют совершенно разного уровня знания как синтаксиса языка, так и принципов объектно-ориентированного программирования в целом.

Программирование в среде Arduino IDE осуществляется через родную оболочку, которую можно скачать на портале Arduino. Она включает в себя следующие элементы:

— редактор;

— препроцессор;

— компилятор;

— менеджер проектов.

Помимо этого, она содержит средства, позволяющие загружать программу в микроконтроллер. Программа создана на языке Java, ее базой служит проект

Processing. Она совместима с операционными системами Windows, Linux и Mac OS X.

Язык программирования Arduino представляет собой стандартный C++, однако у него присутствует ряд особенностей, благодаря которым новым пользователям окажется проще добиться своих первых результатов.

Программные средства, разработанные программистом Ардуино, носят название набросков ("скетчи"). Эти наброски имеют расширение. ino. Компиляции предшествует их обработка препроцессором Arduino. Можно также создавать и использовать в проекте стандартные файлы C++. Функция main() создается в данной среде самостоятельно, после чего к ней добавляются нужные «черновые» действия.

Пользователю требуется создать 2 функции, которые должны присутствовать при использовании рассматриваемого языка. Одна из них – setup() - запускается 1 раз при старте. Выполнение другой, loop(), осуществляется циклично.

Пользователю нет необходимости вставлять в текст создаваемой программы заголовочные файлы стандартных библиотек, которые он использует – это будет осуществлено препроцессором. Что касается пользовательских библиотек, то их указание обязательно.

Менеджеру проекта присущ необычный способ добавления библиотек. Сохранение последних в форме исходных текстов, составленных на стандартном языке C++, осуществляется в отдельной папке, размещающейся в рабочем каталоге среды. Их наименования заносятся в перечень, расположенный в меню среды. Необходимые библиотеки отмечаются программистом, после чего они добавляются в перечень компиляции.

В рассматриваемой среде не предлагается настроек для компилятора; кроме того, ей минимизируются другие настройки. Благодаря этому новым пользователям проще начинать работу, также снижается риск появления проблем.

3.3 Кодирование аппаратной платформы информационной системы «Умный дом»

Для начала каждый датчик должен быть запрограммирован на сбор информации (рис. 3.1).

```
#include "DHT.h"
#define DHTPIN 2
DHT dht(DHTPIN, DHT22);
void setup() {
Serial.begin(9600);
dht.begin();
}
void loop() {
delay(2000);
float h = dht.readHumidity();
float t = dht.readTemperature();
if (isnan(h) || isnan(t)) {
Serial.println("Не удается считать показания");
return;
}
Serial.print("Влажность: "+h+" %\t"+"Температура: "+t+" *C ");
}
```

Рисунок 3.1 – Реализация сбора данных с температурного датчика

В большинстве своём каждый датчик совместимый с аппаратной платформой Arduino имеет собственные библиотеки для более комфортного программирования работы самого датчика, поэтому программисту не нужно знать того как программируется микроконтроллер каждого датчика. В данном случае видно, что при программировании температурного датчика подключалась библиотека DHT.H, содержащая набор функций для обращения к датчику. Посредством разработанных библиотек в программном коде, происходит опрос статуса работы датчика и считывания информации с него. Считанные данные записываются [4, 6].

Подобную реализацию можно наблюдать не только в температурном датчике, но и в датчике движения (рис 3.2).

```
#include Wire.h
#include DS3231.h
#include LiquidCrystal.h
int pirPin = 8;
int val;
void setup() {
    pinMode(pirPin,INPUT);
    Serial.begin(9600);
}
void loop() {
    val = digitalRead(pirPin);
    if (val == LOW) {
      Serial.println("No motion    }
    else {
      Serial.println("Motion!");
    }
    delay(1000);
}
```

Рисунок 3.2 – Реализация сбора данных с датчика движения

Следующим важным модулем программного продукта является подключение к внешней сети интернет. Для подключения могут быть использованы два различных модуля, базовое Ethernet соединение основной платформы ArduinoMega.Исходный код программы прошивки Arduinопредставлен в приложении на рисунке В.1 (Приложение В) И сетевой контроллер беспроводного подключения к интернету – ESP8266. При отсутствии доступа к беспроводной сети устройство начинает опрашивать кабельное соединение [9, 11].

В представленной ниже части исходного кода модуля ESP8266 можно наблюдать опрос датчика (рис 3.3).

```
bool ESP8266wifi::begin() {
    msgOut[0] = '\0';
    msgIn[0] = '\0';
    flags.connectedToServer = false;
    flags.localServerConfigured = false;
    flags.localApConfigured = false;
    serverRetries = 0;

    bool statusOk = false;
    byte i;
    for(i =0; i<HW_RESET_RETRIES; i++){
        readCommand(10, NO_IP); //Cleanup
        digitalWrite(_resetPin, LOW);
        delay(500);
        digitalWrite(_resetPin, HIGH); // select the radio
        // Look for ready string from wifi module
        statusOk = readCommand(3000, READY) == 1;
        if(statusOk)
            break;
    }
    if (!statusOk)
        return false;

    writeCommand(CWMODE_1, EOL);
    if (readCommand(1000, OK, NO_CHANGE) == 0)
        return false;

    if(flags.echoOnOff)//if echo = true
        writeCommand(ATE1, EOL);
    else
        writeCommand(ATE0, EOL);
    if (readCommand(1000, OK, NO_CHANGE) == 0)
        return false;

    writeCommand(CIPMUX_1, EOL);
    flags.started = readCommand(3000, OK, NO_CHANGE) > 0;
    return flags.started;
}

bool ESP8266wifi::isStarted(){
    return flags.started;
}

bool ESP8266wifi::restart() {
    return begin()
        && (!flags.localApConfigured || startLocalAp())
        && (!flags.localServerConfigured || startLocalServer())
        && (!flags.apConfigured || connectToAP())
        && (!flags.serverConfigured || connectToServer());
}
```

Рисунок 3.3 – Реализация сетевого контроллера

При успешном опросе контроллера, выполняется решение о начале функций соединения с сетью, описанных ниже (рис. 3.4).

```
bool ESP8266wifi::connectToAP(){
    writeCommand(CWJAP);
    _serialOut -> print(_ssid);
    writeCommand(COMMA_2);
    _serialOut -> print(_password);
    writeCommand(DOUBLE_QUOTE, EOL);
    readCommand(15000, OK, FAIL);
    return isConnectedToAP();
}
bool ESP8266wifi::isConnectedToAP(){
    writeCommand(CIFSR, EOL);
    byte code = readCommand(350, NO_IP, ERROR);
    readCommand(10, OK); //cleanup
    return (code == 0);
}
char* ESP8266wifi::getIP(){
    msgIn[0] = '\0';
    writeCommand(CIFSR, EOL);
    byte code = readCommand(1000, STAIP, ERROR);
    if (code == 1) {
        // found staip
        readBuffer(&msgIn[0], sizeof(msgIn) - 1, '"');
        readCommand(10, OK, ERROR);
        return &msgIn[0];
    }
    readCommand(1000, OK, ERROR);
    return &msgIn[0];
}

char* ESP8266wifi::getMAC(){
    msgIn[0] = '\0';
    writeCommand(CIFSR, EOL);
    byte code = readCommand(1000, STAMAC, ERROR);
    if (code == 1) {
        // found stamac
        readBuffer(&msgIn[0], sizeof(msgIn) - 1, '"');
        readCommand(10, OK, ERROR);
        return &msgIn[0];
    }
    readCommand(1000, OK, ERROR);
    return &msgIn[0];
}
```

Рисунок 3.4 – Реализация программы для сетевого контроллера

В случае неработоспособности модуля беспроводного соединение происходит опрос кабельного подключения на аппаратной платформе ArduinoMega 2566 (рис. 3.5) [18].

```
IPAddress server(94,19,113,221);
char macbuf[13];

EthernetClient client;
OneWire ds(DS18B20_PIN);
unsigned long lastConnectionTime = 0;
boolean lastConnected = false;
int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;
char replyBuffer[160];                                  /
int CountSensors;
long Pressure = 0;
float Humidity = 0;
 void setup() {
   if (Debug)
  {
    Serial.begin(9600);
  }
  delay(1000);
  if (Ethernet.begin(mac) == 0)
  {
    if (Debug)
    {
      Serial.println("Failed to configure Ethernet using DHCP");
    }
     for(;;);
  }
```

Рисунок 3.5 – Исходный код программы для работы сетевого модуля на плате Arduino

Для реализации алгоритма шифрования был выделен специальный файл - aes.cpp, подключённый к аппаратной платформе ArduinoMega. В нём содержится полный алгоритм шифрования написанный на чистом языке C++, в качестве входных данных программа принимает блоки информации предварительно разбитые для дальнейшего шифрования. Напоминаю, что в текущей версии шифрования, в связи с довольно слабой вычислительной мощностью аппаратной платформы программа принимает блоки данных и ключ размером в 4 байта (32 бита) [13].

Процедура SubBytes, перестановка байтов по таблице замен, общая структура замен представлена на рисунке 3.6.

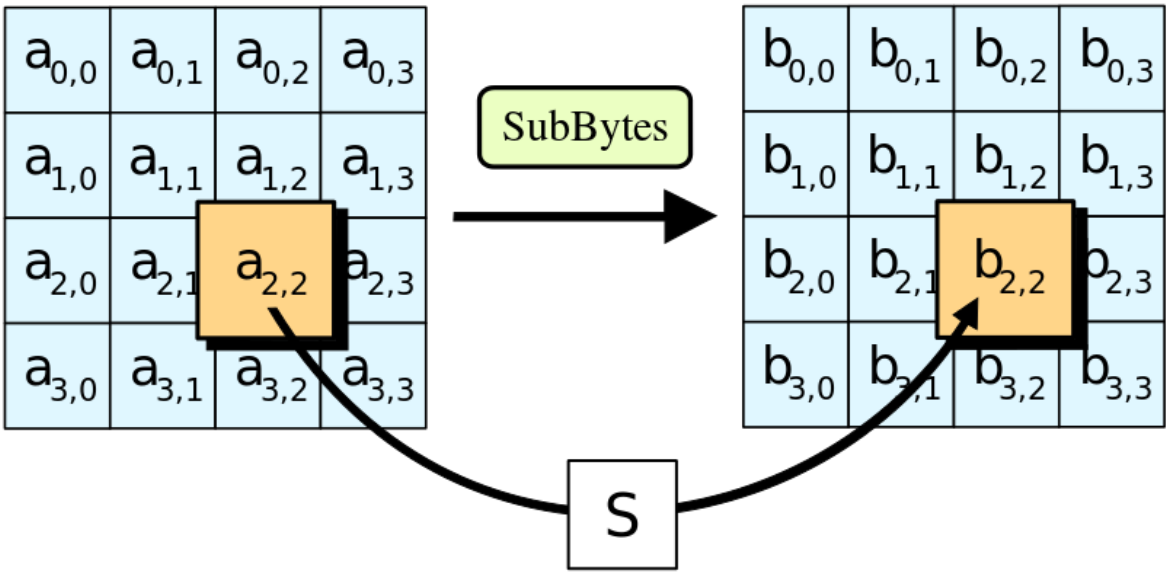

Рисунок 3.6 – Операция SubBytes

В коде программы данная операция представлена следующим образом (рис. 3.7).

```
1  void SubBytes()
2  {
3  int i,j;
4  for(i = 0; i < 4; i++)
5  {
6  for(j = 0; j < 4; j++)
7  {
8  state[i][j] = getSBoxValue(state[i][j]);
9  }
10 }
11 }
```

Рисунок 3.7 – Операция SubBytes

Процедура SubBytes() обрабатывает каждый байт состояния, независимо производя нелинейную замену байтов используя таблицу замен (S-box). Такая операция обеспечивает нелинейность алгоритма шифрования [15].

Следующая основная функция алгоритма шифрования является ShiftRows, основной её принцип действия представлен на рисунке 3.8. Исходный код для работы данной функции представлен на рисунке 3.9.

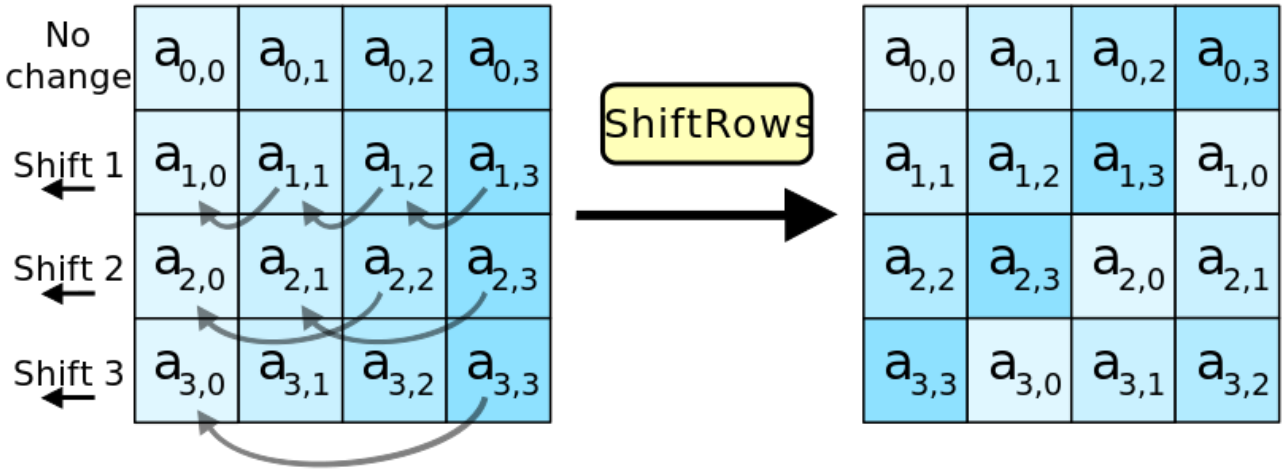

Рисунок 3.8 – Принцип работы функции ShiftRows

```
1  void ShiftRows()
2  {
3  unsigned char temp;
4  temp=state[1][0];
5  state[1][0]=state[1][1];
6  state[1][1]=state[1][2];
7  state[1][2]=state[1][3];
8  state[1][3]=temp;
9  temp=state[2][0];
10 state[2][0]=state[2][2];
11 state[2][2]=temp;
12 temp=state[2][1];
13 state[2][1]=state[2][3];
14 state[2][3]=temp;
15 temp=state[3][0];
16 state[3][0]=state[3][3];
17 state[3][3]=state[3][2];
18 state[3][2]=state[3][1];
19 state[3][1]=temp;
20 }
```

Рисунок 3.9 – Принцип работы функции ShiftRows

При этой трансформации строки состояния циклически сдвигаются на r байт по горизонтали, в зависимости от номера строки. Для нулевой строки r = 0, для первой строки r = 1 и так далее. Таким образом, каждая колонка выходного состояния после применения процедуры ShiftRows состоит из байтов из каждой колонки начального состояния.

Ещё одной важной процедурой для шифрования AES является функция MixColumns. Общий принцип которой, представлен на рисунке 3.10.

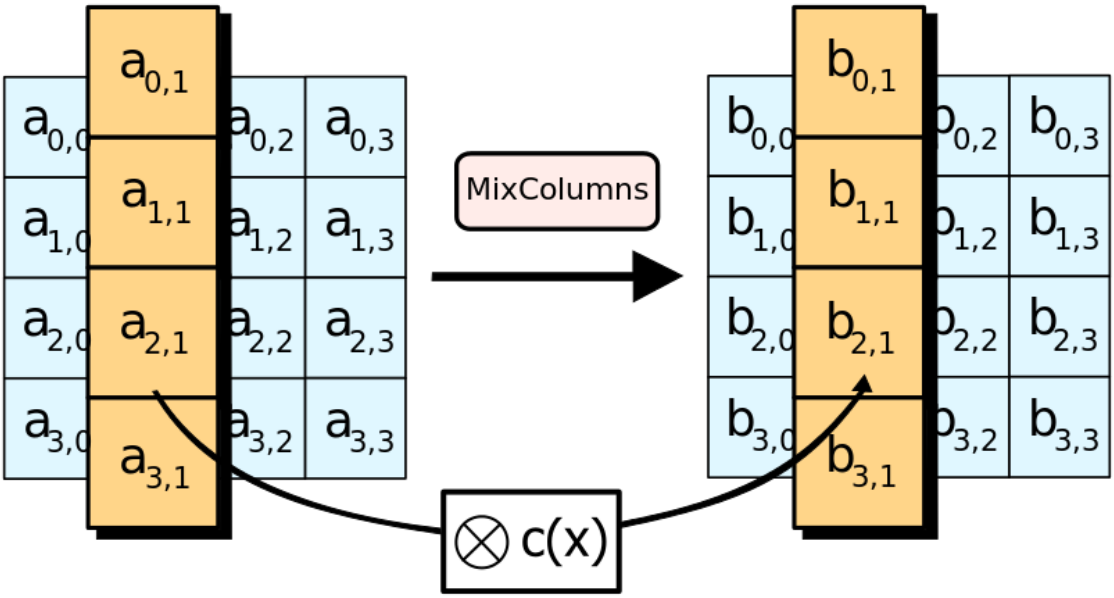

Рисунок 3.10 – Функция MixColumns

В процедуре MixColumns, каждая колонка состояния перемножается с фиксированным многочленом c(x), что в коде программы отображено следующим образом (рис. 3.11).

```
1  void MixColumns()
2  {
3  int i;
4  unsigned char Tmp,Tm,t;
5  for(i = 0; i < 4; i++)
6  {
7  t=state[0][i];
8  Tmp = state[0][i] ^ state[1][i] ^ state[2][i] ^ state[3][i] ;
9  Tm = state[0][i] ^ state[1][i] ; Tm = xtime(Tm); state[0][i] ^= Tm ^ Tmp ;
10 Tm = state[1][i] ^ state[2][i] ; Tm = xtime(Tm); state[1][i] ^= Tm ^ Tmp ;
11 Tm = state[2][i] ^ state[3][i] ; Tm = xtime(Tm); state[2][i] ^= Tm ^ Tmp ;
12 Tm = state[3][i] ^ t ; Tm = xtime(Tm); state[3][i] ^= Tm ^ Tmp ;
13 }
14 }
```

Рисунок 3.11 – Функция MixColumns

И одной из важнейших функций данного алгоритма шифрования является функция AddRoundKey представленная на рисунке 3.12. Исходный код данной функции представлен на рисунке 3.13.

В процедуре AddRoundKey каждый байт состояния объединяется с RoundKey, используя операцию XOR (⊕).

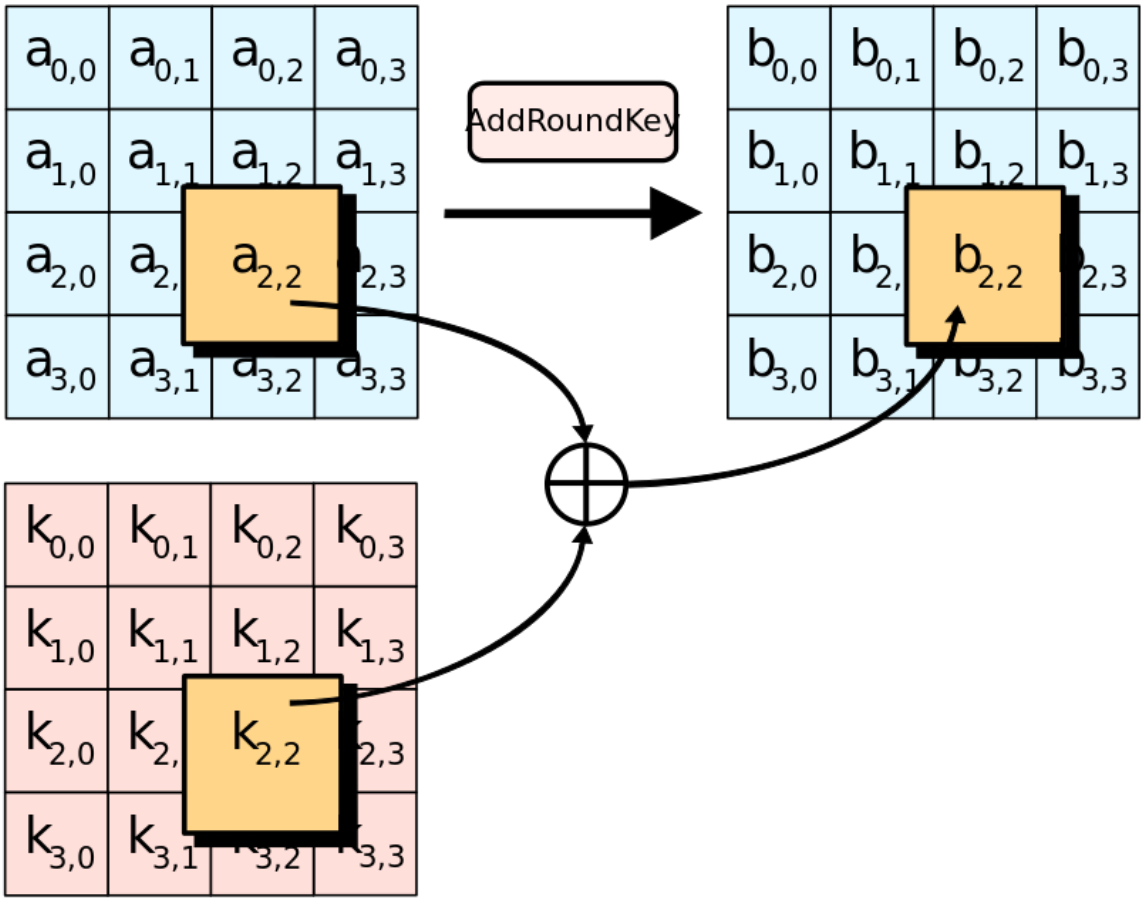

Рисунок 3.12 – Функция AddRoundKey

```
AddRoundKey(0);
// There will be Nr rounds.
// The first Nr-1 rounds are identical.
// These Nr-1 rounds are executed in the loop below.
for(round = 1;round < Nr; round++)
{
SubBytes();
ShiftRows();
MixColumns();
AddRoundKey(round);
}
// The last round is given below.
// The MixColumns function is not here in the last round.
SubBytes();
ShiftRows();
AddRoundKey(Nr);
// The encryption process is over.
// Copy the state array to output array.
for(i = 0; i < 4; i++)
{
for(j = 0; j < 4; j++)
{
out[i * 4 + j]=state[j][i];
}
}
}
```

Рисунок 3.13 – Функция AddRoundKey

3.4 Тестирование аппаратной платформы «Умный дом»

Существует несколько видов тестирования программного продукта. Тестирование на соответствие требованиям к программному продукту и к реализации всех заложенных в данный продукт характеристик называют функциональным тестированием. Оно проводится как правило вручную функциональными тестировщиком [3, 12].

Основным методом тестирования программного обеспечения аппаратной платформы по сбору данных и передачи их на сервер будет ручное функциональное тестирование по методу «Чёрного ящика» - это вид тестирования, проводимый без знания внутренних механизмов работы продукта. Производится на основании внешних проявлений работы продукта. В терминах программного обеспечения под тестированием "черного ящика" обычно подразумевают тестирование через интерфейс пользователя, не имея доступа к исходному коду продукта.

Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций.

Для тестирования алгоритма шифрования будут производится модульное тестирования, более известное как Unit-тестирование.

Процесс Unit тестирование заключает в себя процесс написание программного теста для каждой нетривиальной функции программы. В качестве примера рассмотрим блок, специально созданный для теста алгоритма шифрования AES (рис. 3.14).

Следующий блок используется специально для тестирования и проверяет корректность шифрации, дешифровки, а также скорость работы каждого модуля программы при использовании ключей 128, 192 и 256 бит [6, 16].

Полный код тестирования программы, файлаAESTest.cpp представлен в приложении на рисунке А.1 (Приложение А).

```
#ifdef AES_TEST
    #ifdef LIBST
        #include "libst.cpp"
        typedef long clock_t;
        clock_t clock(){ return GetTickCount(); }
        const clock_t CLOCKS_PER_SEC = 1000;
        extern "C" int _fltused = 0;
    #else
        #include <string.h>
        #include <stdio.h>
        #include <math.h>
        #include <time.h>
    #endif
#endif
aes::u32 inline aes::rot_left_8( u32 value )
{
#ifdef  _M_IX86
    return  _rotl(value, 8);                // MSVC и Intel C++ делают RC
#else
    return  value >> 24 | value << 8;
#endif
}
```

Рисунок 3.14 – Часть исходного теста для Unittестирования программы

Для проверки работы сбора тепловых показаний и датчиков будет применятся технология тестовых кейсов.

Составим тест кейсы для тестирования платформы:

Тест-кейс №1. Подключение аппаратной платформы к заранее определённой сети.

Шаги:

1) Подключить питание к аппаратной платформе.
2) Убедиться в том, что устройство находится в зоне действия заранее определённой беспроводной сети
3) Перейти на страницу аутентификации.Shdiplom.ru
4) Войти в систему с ранее определёнными данными для выбранного устройства.

Ожидаемый результат:

Пользователь будет перемещён на страницу статуса устройства, где отображена его текущая активность

Тест-кейс №2. Определения изменения температуры в доме

Шаги:

1) Зайти на сайт s Shdiplom.ru
2) Войти в систему с ранее зарегистрированным пользователем.
3) Перейти к форме отображения статистики
4) На 30 минут изменить температуру в помещении посредством включения системы кондиционирования или дополнительного отопления.
5) Отслеживать текущие температурные данные.

Ожидаемый результат:

Пользователь будет видеть изменения текущего температурного состояния и графика.

Тест-кейс №3. Определение работоспособность датчика движения.

Шаги:

1) Зайти на сайт Shdiplom.ru.
2) Войти в систему с ранее зарегистрированным пользователем.
3) Перейти к форме отображения статистики о состоянии датчика движения.
4) Выйти на 30 минут из помещения, где расположена аппаратная часть системы, а затем войти в неё.
5) Отслеживать статистику датчика движения.

Ожидаемый результат:

Пользователь будет успешно обнаружит изменения показаний датчика движения на аппаратной платформе.

Перейдем к прохождению описанных тест-кейсов:

Результат выполнения тест-кейса №1, №2 и №3 представлен на рисунке 3.4

Рисунок 3.4 – Результат тест-кейса 1, 2 и 3

3.5 Разработка плана внедрения аппаратной платформы в информационную систему «Умный дом»

Для работы разработанной аппаратной платформы была разработана методика развертывания системы.

Требования к развертыванию программного продукта, включают в себя следующие пункты:

— обеспечения аппаратной платформы доступом к источнику питания по линии 12 В;

— доступ аппаратной платформы к пользовательской Wi-Fi сети.

Для внедрения устройства, требуется выполнить описанные процедуры:

— внесение в систему данных о местоположении устройства (физический адрес с точностью до квартиры);

— добавление ID устройства в БД информационной системы;

— внесение данных о домашней сети в программу (SSID и пароль);

— подключение устройства к сети.

Заключение

В данной выпускной квалификационной работе был выделен объект исследования, проанализирована предметная область, определены цели и задачи работы. Проведен обзор программных средств, позволяющих реализовать серверное приложения для информационной системы «Умный дом».

Анализ программных средств показал, что для решения задач данной выпускной квалификационной работы требуется разработать свой программный продукт.

Был создан программный продукт для аппаратной части информационной системы умный дом. Данный программный продукт, предназначен, для организации сбора данных о жилом помещении с целью передачи этих данных защищённым шифровкой пакетом данных в типовую организацию ЖКХ. Данные предоставляемые созданным программным продуктом, могут позволить улучшить жилищные условия пользователей, а также оптимизировать выделения ресурсов со стороны ЖКХ.

Подобная система в дальнейшем может расширять свой функционал, расширяя тем самым актуальность применения данной системы в реальных условиях.

Даже в условиях выполнения бакалаврской работы удалось достичь сбора достаточно точных конечных температурных данных о жилом помещении.

Обеспечить должный уровень защиты передаваемой информации, а так же предоставить подходящую для дальнейшей разработки аппаратную платформу с необходимым функционалом для реализации программной части, практически любой сложности относительно поставленных ранее задач.

В результате выполнения бакалаврской работы удалось выполнить все поставленные задачи и подтвердить знания материала преподаваемого в процессе обучения специальности математическое обеспечение и администрирование информационных систем.

Список используемой литературы

Учебники и учебные пособия

1. Галямина, И.Г. Управление процессами : учеб. пособие / И. Г. Галямина. – СПб..: Питер, 2013. – 304 с.
2. Карвинен, Т.К. Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi: учеб. пособие / Т.К. Карвинен. – Вильямс, 2015. – 445 с.
3. Клейн Т.С. Дневник охотника за ошибками. Путешествие через джунгли проблем безопасности программного обеспечения: учеб. пособие / Т.С. Клейн. – ДМК Пресс, 2013. – 242 с.
4. Мейерс, С. Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14: учеб. пособие / С. Мейерс. – Вильямс, 2015. – 304 с.
5. Олейник, П.П. Корпоративные информационные системы: учеб. пособие / П. П. Олейник. – СПб..: Питер, 2012. – 176 с.
6. Ошероув, Р. Искусство автономного тестирования с примерами на C++, 2-е издание: учеб. пособие / Р. Ошероув. – ДМК Пресс, 2014. – 360 с.
7. Петин, В.Н. Arduino и RaspberryPi в проектах InternetofThings: учеб. пособие / В.Н. Петин. – БХВ-Петербург, 2016. – 320 с.
8. Петин, В.Н. Проекты с использованием контроллера Arduino, 2-е издание: учеб. пособие / В.Н. Петин. – БХВ-Петербург, 2015. – 448 с.
9. Сильвен, Р. Android NDK. Разработка приложений под Android на C/C++: учеб. пособие / Р. Сильвен. – ДМК Пресс, 2012. – 496 с.
10. Соммер, У.К. Программирование микроконтроллерных плат Arduino/Freeduino: учеб. пособие / У.К. Соммер. – БХВ-Петербург, 2012. – 238 с.
11. Стивенс Р. Алгоритмы. Теория и практическое применение: учеб. пособие / Р. Стивенс. – Эксмо, 2016. – 544 с.

Электронные ресурсы

12. Автор Неизвестен. Модульное тестирование, 2011 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: https://ru.wikipedia.org/wiki/Модульное_тестирование
13. Мозер. Д. AdvancedEncryptionStandard, 2014 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard
14. Сафаров. Г.С. Жилищно-коммунальное хозяйство, 2016 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: https://ru.wikipedia.org/wiki/Жилищно-коммунальное_хозяйство
15. Шнайер. Д. Шифрование, 2012 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: https://ru.wikipedia.org/wiki/Шифрование

Литература на иностранном языке

16. Allain. A. Digman, Jumping into C++, Cprogramming.com, 2013.
17. Amariei. C. Arduino Development Cookbook, Packt Publishing, 2015.
18. Bayle. J. C++ Programming for Arduino, Packt Publishing, 2013.
19. Chuang Lin, Peng Zhang. Digman, Security in Network Coding (Wireless Networks), Springer, 2016.
20. Verona. J. Practical DevOps, Packt Publishing, 2016.

Приложение А

Листинг файла AESTest.cpp

```
#ifdef AES_TEST
/**
 *  Функция используется при компиляции тестового исполняемого файла.
 *  При тестировании проверяется корректность (де)шифрации
 *  и скорость различных методов класса aes при использовании
 *  ключей 128, 192 и 256 бит.
 */
int main()
{
    /**
     *  Данные для тестирования взяты из fips-197.pdf
     *  http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
     */

    static const byte plaintext[aes::block_size] =
    {
        0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,
        0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0xee, 0xff
    };

    // Для "коротких" ключей используется начальная часть всего ключа.
    static const byte key[] =
    {
        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
        0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
        // 192 бит
        0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
        // 256 бит
        0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f
    };

    static const byte ciphertext[3][aes::block_size] =
    {
        //  для ключа 128 бит
        {0x69, 0xc4, 0xe0, 0xd8, 0x6a, 0x7b, 0x04, 0x30,
         0xd8, 0xcd, 0xb7, 0x80, 0x70, 0xb4, 0xc5, 0x5a},
        //  для ключа 192 бит
        {0xdd, 0xa9, 0x7c, 0xa4, 0x86, 0x4c, 0xdf, 0xe0,
         0x6e, 0xaf, 0x70, 0xa0, 0xec, 0x0d, 0x71, 0x91},
        //  для ключа 256 бит
        {0x8e, 0xa2, 0xb7, 0xca, 0x51, 0x67, 0x45, 0xbf,
         0xea, 0xfc, 0x49, 0x90, 0x4b, 0x49, 0x60, 0x89}
    };

    aes     crypto;
    byte    buf[crypto.block_size];

    printf("\n AES test:\n");

    for( int i0  = 0; i0 <= 2; i0++ )
    {
```

Рисунок А.1 – Листинг программы AESTest,часть 1

```
50      for( int i0  = 0; i0 <= 2; i0++ )
51      {
52          int key_length = 128 + 64 * i0;
53          printf("\n %d bit key... ", key_length);
54
55          if( crypto.expand_key(key, key_length) != crypto.Ok )
56              printf("error: can't expand key.");
57          else
58          {
59              crypto.encrypt(plaintext, buf);
60              if( memcmp(&ciphertext[i0], buf, crypto.block_size) )
61                  printf("error: encryption failed.");
62              else
63              {
64                  crypto.decrypt(buf, buf);
65                  if( memcmp(plaintext, buf, crypto.block_size) )
66                      printf("error: decryption failed.");
67                  else
68                      printf("Ok.");
69              }
70          }
71      }
72
73      printf("\n\n Speed test:");
74      printf("\n\n   key   | key expansions/sec | encryption, bytes/sec | decryption, bytes/sec");
75      for( int i1  = 0; i1 <= 2; i1++ )
76      {
77          const int times = 1024 * 1024;
78          int key_length = 128 + 64 * i1;
79          printf("\n %3d bit |", key_length);
80
81          clock_t start = clock();
82          for( int t0 = times; t0; t0-- )
83          {
84              crypto.expand_key(key, key_length);
85          }
86          printf("    %10d      |",
87             (unsigned)((double)times  * CLOCKS_PER_SEC / (clock() - start))
88             );
89
90          crypto.encrypt(plaintext, buf);
91
92          start = clock();
93          for( int t1 = times; t1; t1-- )
94          {
95              crypto.encrypt(buf, buf);
96          }
97          printf("       %10d      |", (unsigned)
98             ((double)crypto.block_size * times * CLOCKS_PER_SEC / (clock() - start))
99             );
100
```

Рисунок А.2 – Листинг программы AESTest,часть 2

```
101         start = clock();
102         for( int t2 = times; t2; t2-- )
103         {
104             crypto.decrypt(buf, buf);
105         }
106         printf("       %10d", (unsigned)
107            ((double)crypto.block_size * times * CLOCKS_PER_SEC / (clock() - start))
108            );
109 
110         crypto.decrypt(buf, buf);
111         if( memcmp(plaintext, buf, crypto.block_size) )
112         {
113             printf("\n\n error: Monte Carlo test failed.\n");
114             return ~0;
115         }
116     }
117     printf("\n\n Ok.\n");
118     return 0;
119 }
120 #endif
```

Рисунок А.3 – Листинг программы AESTest,часть 3

Приложение Б

Листинг файла Arduino.cpp

```
#include <SPI.h>
#include <Ethernet.h>
#include <OneWire.h>
#include <Wire.h>
#include <DHT.h>

bool Debug = false; //режим отладки

//************************************************************************************************
byte mac[] = { 0xDE, 0xAD, 0xBE, 0x00, 0x00, 0x00 }; //MAC-адрес Arduino
#define DHT_EXIST 1             // наличие датчика влажности
#define DHTPIN 6                // пин подключения датчика влажности DHT22
#define DHTTYPE DHT22           // тип датчика влажности DHT22/DHT11
#define postingInterval 600000  // интервал между отправками данных в миллисекундах (10 минут)
//************************************************************************************************

IPAddress server(94,19,113,221); // IP сервера ...
char macbuf[13];

EthernetClient client;
OneWire ds(DS18B20_PIN);

#if DHT_EXIST == 1
  DHT dht(DHTPIN, DHTTYPE);
#endif

unsigned long lastConnectionTime = 0;           // время последней передачи данных
boolean lastConnected = false;                  // состояние подключения
int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;
char replyBuffer[160];                          // буфер для отправки
int CountSensors;                               // количество найденных датчиков температуры
long Pressure = 0;
float Humidity = 0;

void setup() {

  if (Debug)
  {
    Serial.begin(9600);
  }

  // секунда для инициализации Ethernet
  delay(1000);
  // Пробуем подключиться по Ethernet
  if (Ethernet.begin(mac) == 0)
  {
    if (Debug)
    {
      Serial.println("Failed to configure Ethernet using DHCP");
    }
```

Рисунок В.1 – Листинг программы Arduino.cpp,часть 1

```
    }
    // ничего не делаем
    for(;;);
  }

 //Узнаём количество термодатчиков
  CountSensors = DsCount();
  if (Debug)
  {
    Serial.print("Found ");
    Serial.print(CountSensors);
    Serial.println(" sensors.");
  }

  #if DHT_EXIST == 1
    dht.begin();
  #endif

  lastConnectionTime = millis()-postingInterval+15000; //первое соединение через 15 секунд после запуска
}

void loop()
{
  //Если вдруг нам случайно приходят откуда-то какие-то данные,
  //то просто читаем их и игнорируем, чтобы очистить буфер
  if (client.available())
  {
    client.read();
  }

  if (!client.connected() && lastConnected)
  {
    if (Debug)
    {
      Serial.println();
      Serial.println("disconnecting.");
    }
    client.stop();
  }

  //если не подключены и прошло определённое время, то делаем замер,
  //переподключаемся и отправляем данные
  if (!client.connected() && (millis() - lastConnectionTime > postingInterval))
  {

    //формирование HTTP-запроса
    memset(replyBuffer, 0, sizeof(replyBuffer));
    strcpy(replyBuffer,"ID=");

    memset(macbuf, 0, sizeof(macbuf));
    //Конвертируем MAC-адрес
```

Рисунок В.2 – Листинг программы Arduino.cpp,часть 2

```
100     //Конвертируем MAC-адрес
101     for (int k=0; k<6; k++)
102     {
103       int b1=mac[k]/16;
104       int b2=mac[k]%16;
105       char c1[2],c2[2];
106
107       if (b1>9) c1[0]=(char)(b1-10)+'A';
108       else c1[0] = (char)(b1) + '0';
109       if (b2>9) c2[0]=(char)(b2-10)+'A';
110       else c2[0] = (char)(b2) + '0';
111
112       c1[1]='\0';
113       c2[1]='\0';
114
115       strcat(macbuf,c1);
116       strcat(macbuf,c2);
117     }
118     strcat(replyBuffer, macbuf);
119
120     //Сбрасываем поиск датчиков (кол-во нам уже известно)
121     ds.reset_search();
122     //Теперь в цикле опрашиваем все датчики сразу
123
124     for (int j=0; j<CountSensors; j++)
125     {
126
127       byte i;
128       byte present = 0;
129       byte data[12];
130       byte addr[8];
131
132       if ( !ds.search(addr))
133       {
134         ds.reset_search();
135         return;
136       }
137
138       ds.reset();
139       ds.select(addr);
140       ds.write(0x44,1);
141
142       delay(1000);
143
144       present = ds.reset();
145       ds.select(addr);
146       ds.write(0xBE);
147
148       for ( i = 0; i < 9; i++) // we need 9 bytes
149       {
150         data[i] = ds.read();
```

Рисунок В.3 – Листинг программы Arduino.cpp,часть 3

```
    data[i] = ds.read();
  }

  LowByte = data[0];
  HighByte = data[1];
  TReading = (HighByte << 8) + LowByte;
  SignBit = TReading & 0x8000;  // test most sig bit
  if (SignBit) // negative
  {
    TReading = (TReading ^ 0xffff) + 1; // 2's comp
  }
  Tc_100 = (6 * TReading) + TReading / 4;    // multiply by (100 * 0.0625) or 6.25

  Whole = Tc_100 / 100;  // separate off the whole and fractional portions
  Fract = Tc_100 % 100;

  char temp[3];

  itoa(Whole,temp);
  strcat(replyBuffer,"&");

  //конвертируем адрес термодатчика
  for (int k=7; k>=0; k--)
  {
    int b1=addr[k]/16;
    int b2=addr[k]%16;
    char c1[2],c2[2];

    if (b1>9) c1[0]=(char)(b1-10)+'A';
    else c1[0] = (char)(b1) + '0';
    if (b2>9) c2[0]=(char)(b2-10)+'A';
    else c2[0] = (char)(b2) + '0';

    c1[1]='\0';
    c2[1]='\0';

    strcat(replyBuffer, c1);
    strcat(replyBuffer, c2);
  }
  strcat(replyBuffer,"=");
  if (SignBit) //если температура отрицательная, добавляем знак минуса
  {
    strcat(replyBuffer,"-");
  }
  strcat(replyBuffer,temp);
  strcat(replyBuffer,".");
  if (Fract<10)
  {
    strcat(replyBuffer,"0");
  }
  itoa(Fract,temp);
```

Рисунок В.4 – Листинг программы Arduino.cpp,часть 4

```
200       itoa(Fract,temp);
201       strcat(replyBuffer,temp);
202     }
203
204     char temp[8];
205     long p_100, h_100;
206
207     #if DHT_EXIST == 1
208       Humidity = dht.readHumidity();
209       strcat(replyBuffer, "&");
210       strcat(replyBuffer, macbuf);
211       strcat(replyBuffer, "02=");
212       h_100 = Humidity*100;
213       Whole = h_100 / 100;
214       Fract = h_100 % 100;
215       itoa(Whole, temp);
216       strcat(replyBuffer, temp);
217       strcat(replyBuffer, ".");
218       if (Fract<10)
219       {
220         strcat(replyBuffer,"0");
221       }
222       itoa(Fract, temp);
223       strcat(replyBuffer, temp);
224     #endif
225
226     strcat(replyBuffer,'\0');
227
228     if (Debug)
229     {
230       Serial.println(replyBuffer);
231       Serial.print("Content-Length: ");
232       Serial.println(len(replyBuffer));
233     }
234
235     //отправляем запрос
236     httpRequest();
237
238   }
239   //храним последнее состояние подключения
240   lastConnected = client.connected();
241 }
242
243 void httpRequest()
244 {
245   if (client.connect(server, 80))
246   {
247     if (Debug)
248     {
249       Serial.println("connecting...");
250     }
```

Рисунок В.5 – Листинг программы Arduino.cpp,часть 5

```
250       }
251       // отправляем HTTP POST запрос:
252       client.println("POST http://shdiplom.ru/post.php HTTP/1.0");
253       client.println("Host: shdiplom.ru");
254       //client.println("User-Agent: arduino-ethernet");
255       //client.println("Connection: close");
256       client.println("Content-Type: application/x-www-form-urlencoded");
257       client.print("Content-Length: ");
258       client.println(len(replyBuffer));
259       client.println();
260       client.println(replyBuffer);
261       client.println();
262
263       lastConnectionTime = millis();
264     }
265     else
266     {
267       if (Debug)
268       {
269         Serial.println("connection failed");
270         Serial.println("disconnecting.");
271       }
272       client.stop();
273     }
274   }
275
276   //Количество термодатчиков на шине
277   int DsCount()
278   {
279     int count=0;
280     bool thatsall = false;
281     byte addr[8];
282     do
283     {
284       if ( !ds.search(addr))
285       {
286         ds.reset_search();
287         thatsall = true;
288       }
289       count++;
290     } while(!thatsall);
291     return (count-1);
292   }
293
294   int len(char *buf)
295   {
296     int i=0;
297     do
298     {
299       i++;
300     } while (buf[i]!='\0');
```

Рисунок В.6 – Листинг программы Arduino.cpp,часть 6

```
300   } while (buf[i]!='\0');
301   return i;
302 }
303
304 void reverse(char s[])
305 {
306   int i, j;
307   char c;
308
309   for (i = 0, j = strlen(s)-1; i<j; i++, j--)
310   {
311     c = s[i];
312     s[i] = s[j];
313     s[j] = c;
314   }
315 }
316
317 void itoa(int n, char s[])
318 {
319   int i, sign;
320
321   if ((sign = n) < 0)       /* записываем знак */
322     n = -n;                 /* делаем n положительным числом */
323   i = 0;
324   do {                      /* генерируем цифры в обратном порядке */
325     s[i++] = n % 10 + '0';  /* берем следующую цифру */
326   } while ((n /= 10) > 0);  /* удаляем */
327   if (sign < 0)
328     s[i++] = '-';
329   s[i] = '\0';
330   reverse(s);
331 }
```

Рисунок В.7 – Листинг программы Arduino.cpp,часть 7