

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА
Рецензент

_____ 2020 г.
«__» _____

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко

«__» _____ 2020 г.

Разработка прототипа системы связи Li-Fi для передачи данных между
мобильными устройствами Android (передающая часть)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ И.Л. Кафтанников
«__» _____ 2020 г.

Автор работы,
студент группы КЭ-222
_____ А.В. Козлова
«__» _____ 2020 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«__» _____ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Г.И. Радченко
«___» _____ 2020 г.

ЗАДАНИЕ
на выпускную квалификационную работу магистра
студенту группы КЭ-222
Козловой Анастасии Владимировне
обучающемуся по направлению
09.04.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка прототипа системы связи Li-Fi для передачи данных между мобильными устройствами Android (передающая часть)»
утверждена приказом по университету от 24 апреля 2020 г. № 627
- 2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.
- 3. Исходные данные к работе:** статьи, книги.
– Wu, X., Safari. Access point selection for hybrid Li-Fi and Wi-Fi networks / X. Wu, M. Safari, H. Haas // IEEE Transactions on Communications. - 2017. – V. 65, №2. – P. 5375–5385.
– Modeling the random orientation of mobile devices: Measurement, analysis and LiFi Use Case / M.D. Soltani, A.A. Purwita, Z. Zeng, H. Haas, M. Safari // IEEE Transactions on Communications. – 2019. – V. 67, №3. – P. 2157–2172.
– Haas, H. A guide to wireless networking by light / H. Haas, C. Chen, D. O'Brien // Progress in Quantum Electronics. – 2017. – V. 55. – P. 88 –111.

– Петрусь, И. П. Современные методы построения коммуникационных линий в видимой части спектра электромагнитных волн / И. П. Петрусь // Международная конференция студентов, аспирантов и молодых ученых "Информационные технологии, телекоммуникации и системы управления": сборник докладов. – Екатеринбург: [УрФУ]. – 2015. – С. 212-216.

4. Перечень подлежащих разработке вопросов:

- анализ существующих решений и научных проектов систем Li-Fi;
- анализ основных технологических решений реализации передатчика;
- определение требований к системе;
- разработка архитектуры решения;
- выбор и обоснование аппаратных и программных средств;
- разработка принципиальной электрической схемы устройства приемопередатчика;
- разработка печатной платы устройства приемопередатчика;
- разработка алгоритмов работы для аппаратной части;
- разработка алгоритмов работы мобильного приложения;
- реализация алгоритмов;
- тестирование.

5. Дата выдачи задания: 1 декабря 2019 г.

Руководитель работы _____ /И.Л. Кафтанников /

Студент _____ /А.В. Козлова /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка модели, проектирование	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы _____ /И.Л. Кафтанников/

Студент _____ /А.В. Козлова/

Аннотация

А. В. Козлова. Разработка прототипа системы связи Li-Fi для передачи данных между мобильными устройствами Android (передающая часть). – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 130 с., 39 ил., библиогр. список – 75 наим.

Работа посвящена проектированию и разработке передающего модуля прототипа программно-аппаратного комплекса системы связи Li-Fi для передачи данных между мобильными устройствами Android. Li-Fi является актуальной технологией в современном мире, так как для обмена данными она использует видимый свет, что позволяет достичь высоких скоростей при передаче.

Аппаратная часть основана на использовании микроконтроллера STM32F100, он обеспечивает компромисс между высокой производительностью, энергоэффективностью и финансовыми затратами. Аппаратная часть представляет из себя приемопередатчик, который подключается к мобильному устройству через современный аудиоразъем jack 3.5. Мобильное приложение поддерживается устройствами на операционной системе Android 6.0 и выше в виду распространенности данной операционной системы для мобильных устройств.

Работа состоит из введения, 5 разделов, заключения, библиографического списка и приложений. В разделе 1 проведен анализ найденных на рынке систем Li-Fi, описаны их достоинства и недостатки, в разделе 2 проведен анализ основных технологических решений, определены функциональные и нефункциональные требования к программной системе, в разделе 3 выполнено проектирование программной и аппаратной частей: приведены UML-диаграммы, архитектура системы, подобраны компоненты для приемопередатчика, описаны основные алгоритмы передающего модуля системы и т.д. В разделе 4 показаны результаты реализации, в 5 – результаты тестирования модуля.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	11
1.1 ОБЗОР АНАЛОГОВ	15
1.2 ВЫВОД	19
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	23
2.1 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ	23
2.2 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	35
2.3 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	39
3. ПРОЕКТИРОВАНИЕ	41
3.1 ДИАГРАММА КЛАССОВ ПРИЛОЖЕНИЯ	41
3.2 СТРУКТУРНАЯ СХЕМА ПЕРЕДАЮЩЕГО МОДУЛЯ	42
3.3 АРХИТЕКТУРА СИСТЕМЫ	43
3.4 ПОДБОР КОМПОНЕНТОВ	45
3.5 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА	56
3.6 РАЗВОДКА ПЕЧАТНОЙ ПЛАТЫ	57
3.7 ПРОЕКТИРОВАНИЕ КОРПУСА УСТРОЙСТВА	58
3.8 ВЫБОР ИНСТРУМЕНТОВ РАЗРАБОТКИ	60
3.9 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ	61
3.10 ОПИСАНИЕ ДАННЫХ	69
4. РЕАЛИЗАЦИЯ.....	70
4.1 ПРИЛОЖЕНИЕ ANDROID	70
4.2 УСТРОЙСТВО ПРИЕМОПЕРЕДАТЧИКА	86
5. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ	90
6. ЗАКЛЮЧЕНИЕ	93
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	94

ПРИЛОЖЕНИЕ А СХЕМА ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПИАЛЬНАЯ	102
ПРИЛОЖЕНИЕ Б ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ К СХЕМЕ ЭЛЕКТРИЧЕСКОЙ ПРИНЦИПИАЛЬНОЙ	103
ПРИЛОЖЕНИЕ В ПЕЧАТНАЯ ПЛАТА	105
ПРИЛОЖЕНИЕ Г ИСХОДНЫЕ КОДЫ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	106
ПРИЛОЖЕНИЕ Д ИСХОДНЫЕ КОДЫ ДЛЯ STM32	129

ВВЕДЕНИЕ

Устройства, использующие свет для передачи данных, помнят владельцы первых телефонов с ИК-портами (инфракрасный порт). Это технология соединения появилась в 1993 году. До существования современных беспроводных технологий ИК-порты были практически единственным возможным способом передать файлы между мобильными телефонами. Для этого применяли невидимые глазу ИК-волны — принцип, широко используемый в пультах управления. Технология требовала много времени и минимального расстояния между телефонами для передачи файлов, но стабильно работала. Однако слишком низкая скорость передачи данных, делает данный интерфейс устаревшим.

Следующим этапом связи между мобильными устройствами стал Bluetooth (1998 год). Одним из главных недостатков сетей Bluetooth является обеспечиваемый уровень безопасности. Современная Bluetooth-технология не предлагает никакого средства опознавания пользователя, что делает Bluetooth-устройства особенно уязвимыми к так называемым spoofing-нападениям (радиодезинформации) и неправильному применению опознавательных устройств. Особенно слабым аспектом Bluetooth является процесс «спаривания» (pairing) устройств, при котором происходит обмен ключами в незакодированных каналах. Если нападающий перехватит передачу процесса инициализации обмена, то он сможет получить ключ инициализации путем калькуляции этих ключей для любого возможного варианта пароля и сравнения результатов с перехваченной передачей. Также к недостаткам стоит отнести среднюю скорость обмена данными.

Следующий шаг – использование Wi-Fi для коммуникации устройств. Для этого нужно организовать локальную сеть и лишь через нее одно устройство сможет передать данные на второе (оба должны быть подключены к этой локальной сети), что не очень удобно, тем более для этого способа передачи необходимо дополнительное оборудование, например, маршрутизатор для

«раздачи» Wi-Fi. К тому же не везде разрешено/возможно использовать Wi-Fi (как и Bluetooth).

Спустя годы, несмотря на доступность и распространенность Bluetooth и Wi-Fi, потребовалось расширение каналов частотных диапазонов, поэтому человечество вновь вернулось к вопросу передачи данных с помощью света, на этот раз видимого, что является потенциалом для развития высокоскоростной коммуникации устройств и концепта Интернета вещей (Internet of Things, IoT).

В 2011 году немецкий физик Харальд Хаас впервые упомянул термин Li-Fi (light fidelity) на одной из своих лекций. В 2015 году он представил технологию передачи данных с помощью световых импульсов, которая и была названа Li-Fi [1]. Основное преимущество этой технологии – высокая скорость передачи данных и в несколько раз меньшие затраты по организации данной системы по сравнению с Wi-Fi. Li-Fi относится к VLC-технологиям (Visible Light Communication) – связь по видимому свету.

Первая в России сеть передачи данных с помощью света была запущена в Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики (ИТМО) в июле 2017 года [2], в настоящий момент там же разрабатывается комплекс беспроводной системы передачи данных по технологии Li-Fi для Интернета вещей и интеллектуальной световой среды в городском пространстве [20]. Больше в отечественных информационных источниках не было найдено данных о разработке, поддержке, использовании Li-Fi сетей в России. Новизна технологии Li-Fi в России и ее преимущества определяют актуальность исследовательской темы о передаче данных с помощью света. Так как мобильные устройства на сегодняшний день распространены повсеместно, целесообразно реализовать коммуникацию между мобильными устройствами. Очевидно, что в рамках дипломной работы не представляется возможным разработать полноценную систему Li-Fi для коммуникации устройств, поэтому цель магистерской диссертации - это

разработка программно-аппаратного модуля для передачи данных, который является частью прототипа системы Li-Fi для обмена данными между устройствами Android. В аппаратную часть входит разработка стороны передачи для приемопередатчика, в программную – разработка алгоритма работы передатчика, написание мобильного приложения для общения с передатчиком.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

1. Провести обзор и анализ существующих аналогичных или похожих решений.
2. Провести анализ способов реализации аппаратной и программной части системы Li-Fi для мобильных устройств Android.
3. Подобрать компоненты для аппаратной части.
4. Разработать архитектуру программно-аппаратного комплекса, схему электрическую принципиальную, выполнить трассировку печатной платы приемопередатчика, разработать основные алгоритмы и представить их конечную реализацию.
5. Оценить работоспособность системы.

Основные источники информации в данной работе – это зарубежные и российские статьи по теме Li-Fi и VLC.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Совсем скоро вокруг людей появится огромное количество всевозможных «умных» устройств: не только телефоны, планшеты и часы, но также очки, кольца, браслеты, колонки, игрушки и прочие устройства [3]. В таком кибернетическом мире потребуется очень широкий и надежный канал передачи данных, который сможет справиться с огромным объемом передаваемой информации. Не исключено, что в ближайшие 10 лет на каждого пользователя смартфонов, компьютеров, «умных» устройств потребуется выделять высокочастотные каналы, количество передаваемых данных в которых сможет измеряться десятками Гбит/с. Чтобы качественно увеличить пропускную способность передаваемой информации, а также научить устройства общаться друг с другом без участия серверов, многие ученые уже сейчас активно работают над технологией передачи данных с помощью света. С недавних пор это стало реальным благодаря массовому внедрению светодиодов.

В настоящее время есть альтернативные способы высокоскоростной передачи данных – проводные: использование волоконно-оптического кабеля (оптоволокно), витая пара категории CAT6A и выше. Беспроводные: 5G. Главным недостатком оптоволокна является высокая стоимость его прокладки и эксплуатации [4]: необходимо иметь специальное оборудование для преобразования сигнала, а также при разрыве, волокно практически не подлежит ремонту (обычно менять приходится целый участок кабеля). По сравнению с волоконно-оптическим кабелем, кабель витой пары более тонкий, более гибкий и его проще устанавливать. Он также недорог. Однако, витая пара обладает следующими недостатками: сильное воздействие внешних электромагнитных наводок, возможность утечки информации и сильное затухание сигналов (потеря мощности сигналов). Затухание растет с увеличением частоты сигнала и удлинением кабеля передающей среды. Проводники витой пары подвержены поверхностному эффекту - при высокой частоте тока, электрический ток

вытесняется из центра проводника, что приводит к уменьшению полезной площади проводника и дополнительному ослаблению сигнала [5]. Недостатки сетей 5G: необходимость разработок новых стандартов, более мощных микропроцессоров и программного обеспечения, для плотного покрытия понадобится установить большое количество станций сотовой связи [6].

Li-Fi представляет из себя высокоскоростную систему связи типа точка - многоточка (окружающая среда) но в ограниченном пространстве. В общем случае, в качестве роутера в такой системе выступают светодиоды либо светодиодные лампы. Li-Fi – это оптическая версия Wi-Fi, которая, в отличие от последнего, имеет значительно большую ширину полосы пропускания частот, благодаря использованию света видимого спектра. Для сравнения, спектр радиоволн, используемых в радиотехнике, лежит в диапазоне от 3 КГц до 3000 ГГц, спектр видимого излучения варьируется от 430 до 790 ТГц, что показано на рисунке 1 [7]. В результате этого потенциальная эффективность передачи данных посредством Li-Fi значительно выше, чем у Wi-Fi (соответственно, и чем у Bluetooth и ИК-излучения).

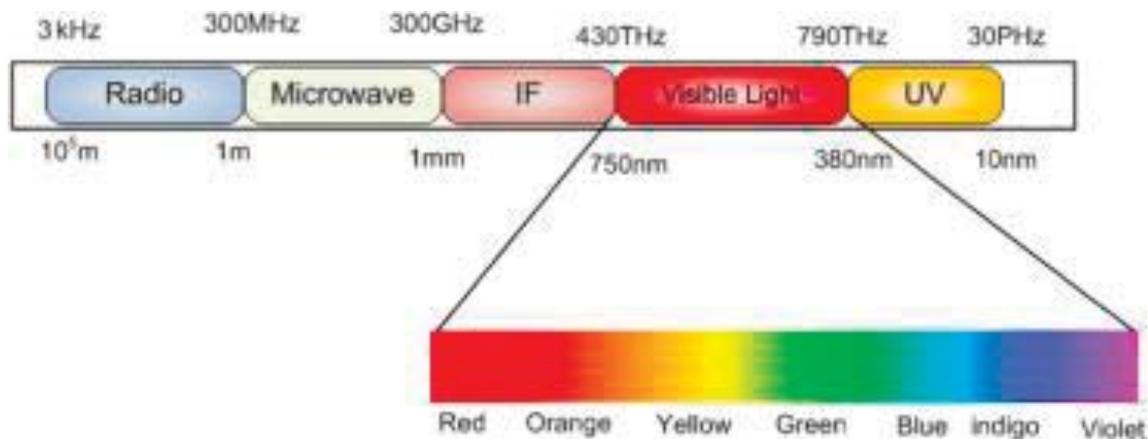


Рисунок 1 – Частотный спектр VLC

Кроме этого, естественный свет, излучаемый светодиодами или светодиодной лампой – это то, с чем сталкивается человек на протяжении всей жизни без опасения для своего здоровья, поэтому Li-Fi в этом плане абсолютно безопасен для человека, в отличие от электромагнитных волн, исходящих от Wi-

Fi [8]. Помимо безопасности для здоровья, характерной особенностью такой системы является безопасность передаваемой информации: если пользователь не видит источник света, находясь в другом помещении, он не имеет доступа к передаваемым данным. В результате этого, такие системы могут использоваться в военных целях, где радиочастотные коммуникации могут быть подвергнуты прослушиванию. Еще одно преимущество Li-Fi – организация коммуникации с помощью этой системы является дешевой [8]. Li-Fi не создает помехи устройствам, поэтому разрешено использовать там, где запрещен Wi-Fi, кроме того, Li-Fi можно совмещать с освещением помещения, так как он использует светодиоды. В таблице 1 сопоставлены возможности перечисленных выше высокоскоростных технологий связи с Li-Fi. На рисунке 2 приведено изображение, которое отражает применение Li-Fi в общем виде.



Рисунок 2 – Применение Li-Fi

Таблица 1 – Возможности высокоскоростных технологий

Наименование решения	Максимальная частота	Ширина сигнала	Максимальная скорость
Оптоволокно [62]	100 ПГц	От 100 ТГц до 100 ПГц	50000 Гбит/с

Продолжение таблицы 1

Наименование решения	Максимальная частота	Ширина сигнала	Максимальная скорость
Витая пара [5]	700 МГц	От 500 МГц до 700 МГц	100 Гбит/с
5G сети [6]	100 ГГц	От 24 ГГц до 100 ГГц	20 Гбит/с
Li-Fi ([7], [63])	790 ТГц	От 430 ТГц до 790 ТГц	224 Гбит/с

Невысокие скорости обмена данными по ИК-порту, Bluetooth, проблемы безопасности Bluetooth-технологии [9], дороговизна организации коммуникаций по Wi-Fi, преимущества Li-Fi и другие обстоятельства выше обосновывают необходимость разработки прототипа системы Li-Fi в варианте для передачи, что входит в данную работу, и приема данных между мобильными устройствами Android, так как устройства на ОС (операционной системе) Android распространены на сегодняшний день повсеместно, что отражено на рисунке 3 [10]. Данная система должна увеличить пропускную способность, безопасность и удобство использования при обмене информацией между упомянутыми устройствами, а в дальнейшем использоваться для подключения к сети Интернет и для Интернета вещей.

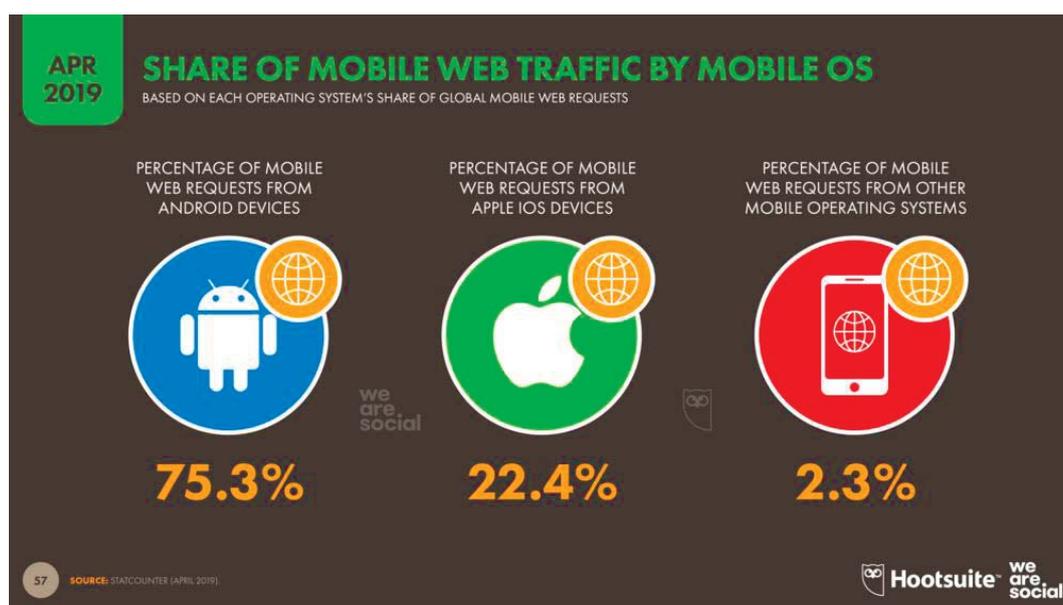


Рисунок 3 – Статистика Web-запросов с различных мобильных ОС

1.1 ОБЗОР АНАЛОГОВ

При поиске существующих решений было обнаружено малое количество похожих программно-аппаратных комплексов, использующихся как для обмена данными между устройствами, так и для доступа к Интернету. Далее кратко описаны наиболее похожие найденные решения.

1. Решения от французской компании Oledcomm, которая в данный момент является одним из лидеров в области световых решений [11]. Oledcomm специализируются на создании продуктов, которые оказывают геолокационные услуги, а также – на решениях для выхода в Интернет с помощью света. Основной их продукт, похожий на разрабатываемую систему в данной работе - это LiFiMAX. LiFiMAX представляет собой оптическую систему беспроводной связи, обеспечивающую до 16 одновременных пользователей высокоскоростным, высоконадежным и беспроводным доступом к сети благодаря невидимому свету [16]. Система LiFiMAX состоит из точки доступа LiFiMAX, которую можно закрепить на потолке, например, конференц-зала. Основные характеристики: скорость 100 Мбит/с по восходящему каналу и 40 Мбит/с по нисходящему, совместима с Linux, Windows, MacBook с USB-портами типа А и С, высокая безопасность, так как свет не проходит через стены. На рисунке 4 [16] приведена схема системы LiFiMAX. Пакет LiFiMAX стоит 900 \$ (~55 173 рубля), поставки этого решения начались еще до сентября 2019 года [17].

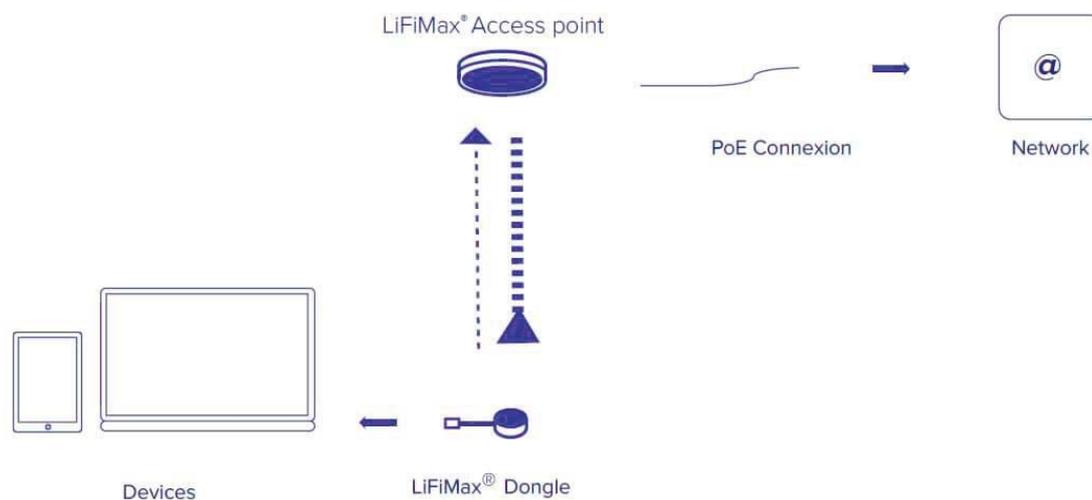


Рисунок 4 – Схема установки системы LiFiMAX

2. Решение индийской компании Velmenni: mesh-сеть (вид топологии компьютерной сети, построенной на принципе ячеек, в которой рабочие станции сети соединяются друг с другом и способны принимать на себя роль коммутатора для остальных участников [18]) на основе технологии VLC, в mesh-сети пользователь может передвигаться без обрывов сети [12]. Анонс решения был обещан на конец 2016 года, но он до сих пор не состоялся. Известные технические характеристики: скорость передачи данных до 1 Гб/с, приемопередатчик представляет из себя USB-ключ [19].
3. Экспериментальная разработка университета ИТМО - локальная сеть передачи данных с помощью света [2]. Известные технические параметры: в лабораторных условиях удалось достичь скорости 50 Мб/с при передаче видео с одного ноутбука на другой, поддерживается только на компьютерах, приемопередатчик – USB-ключ.
4. Решения шотландской компании PureLiFi, основанной профессором Харальдом Хаасом, который ввел понятие «Li-Fi»: разрабатывается технология «Gigabit LiFi» для мобильных устройств, подобная технология уже может быть интегрирована в ноутбуки [13]. Известные технические параметры: пропускная способность 1 Гбит/с, встроенный

приемопередатчик, полный дуплекс. На рисунке 5 [13] показан встраиваемый приемопередатчик.



Рисунок 5 – Интегрированный приемопередатчик Gigabit LiFi для мобильных устройств

5. Решения нидерландской компании Signify (Philips Lighting до 2018 года) – светильники Trulifi с поддержкой Li-Fi. Есть три версии этих светильников. Серия 6001, которая доступна еще с 2018 года, известные технические характеристики: скорость соединения 30 Мбит/с, обмен данными со светильником происходит через USB-ключ, используется видимый свет максимальная высота потолка – 2 м [21]. Серия Trulifi 6002 использует двусторонний ИК-свет со скоростью до 150 Мбит/с, максимальная высота потолка – 2 м, обмен данными со светильником через USB-ключ [21]. Trulifi 6013 (стала доступна осенью 2019 года) использует видимый и ИК-свет для создания соединения со скоростью 250 Мбит/с, максимальная высота потолка – 2 м, ширина покрытия до 8 м, благодаря ИК-излучению соединение может быть установлено даже при выключенном свете, обмен данными со светильником через USB-ключ [21]. в настоящий момент, решение этой компании обеспечивает самую широкую зону покрытия в мире [14]. На рисунке 6 [14] показана схема системы Li-Fi от компании Signify, которую можно считать общей при организации системы Li-Fi для выхода в Интернет. Цены на свои продукты компания Signify не хранит в открытом доступе.



Рисунок 6 – Организация системы Li-Fi для доступа в Интернет от Signify

6. Решение французской компании Lucibel: система Li-Fi состоит из светодиодного светильника (на сайте компании представлены различные их виды) и USB-ключа, используемого для передачи и приема информации. Известные технические параметры: USB-ключ может быть подключен к таким ОС, как Windows 7, 10, MacOS, Linux, скорость соединения – 54 Мб/с [15]. Стоимость системы отсутствует в открытом доступе. На рисунке 7 [15] показан светильник «LIFICUP», на рисунке 8 [15] – USB-ключ «CLÉ USB LIFI».



Рисунок 7 – Светодиодный светильник для Li-Fi от компании Lucibel



Рисунок 8 – USB-ключ для Li-Fi от компании Lucibel

1.2 ВЫВОД

На основании информации о найденных аналогах выявлены их достоинства и недостатки, которые сведены в таблицу 3.

Таблица 2 – Преимущества и недостатки имеющихся решений

Наименование решения	Преимущества	Недостатки
1. LiFiMAX от компании Oledcomm [11]	<ul style="list-style-type: none"> - мультипользовательская система – до 16 пользователей; - поддержка основных ОС для ПК; - выход в Интернет; - компактный отдельный ключ-приемопередатчик; - USB-ключ типа А и С. 	<ul style="list-style-type: none"> - не поддерживается Android-устройствами; - низкая скорость для световой передачи – до 100 Мб/с; - высокая стоимость; - нет режима коммуникации между устройствами.

Продолжение таблицы 2

Наименование решения	Преимущества	Недостатки
<p>2. mesh-сеть на основе VLC от компании Velmenni [12]</p>	<ul style="list-style-type: none"> - высокая эффективность, если много подключенных пользователей; - независимость от провайдера [22]; - выход в Интернет - компактный отдельный ключ-приемопередатчик; - режим коммуникации между устройствами. 	<ul style="list-style-type: none"> - эффективность достигается только при большом числе пользователей; - низкая скорость для световой передачи – до 1 Гб/с; - отсутствует в продаже; - USB-ключ типа А (не на всех устройствах есть такой порт); - негарантированное качество связи и ширина канала [22]; - первоначальный запуск mesh-сети сложен [22].
<p>3. Разработка ИТМО [2]</p>	<ul style="list-style-type: none"> - наличие компактных отдельных модемов-приемопередатчиков; - режим коммуникации между устройствами. 	<ul style="list-style-type: none"> - экспериментальная разработка; - не поддерживается мобильными устройствами; - низкая скорость для световой передачи – до 50 Мб/с; - USB-ключ типа А (не на всех устройствах есть USB-порт типа А).

Продолжение таблицы 2

Наименование решения	Преимущества	Недостатки
4. Gigabit LiFi от компании PureLiFi [13]	<ul style="list-style-type: none"> - встроенный приемопередатчик; - выход в Интернет; - средняя скорость для световой передачи - 1 Гб/с. 	<ul style="list-style-type: none"> - дополнительные затраты на интегрирование приемопередатчиков в оборудование, что отразится на конечной стоимости системы; - только разрабатывается для мобильных устройств; - нет режима коммуникации между устройствами.
5. Trulifi от компании Signify [14]	<ul style="list-style-type: none"> - доступ в Интернет; - режим коммуникации между устройствами; - самая большая площадь покрытия в мире; - компактный отдельный ключ-приемопередатчик. 	<ul style="list-style-type: none"> - низкая скорость для световой передачи до 250 Мб/с; - USB-ключ типа А (не на всех устройствах есть USB-порт типа А); - не поддерживается мобильными устройствами.
6. LiFi от компании Lucibel [15]	<ul style="list-style-type: none"> - поддержка основных ОС для ПК; - выход в Интернет; - компактный отдельный ключ-приемопередатчик. 	<ul style="list-style-type: none"> - не поддерживается мобильными устройствами; - низкая скорость для световой передачи - 54 Мб/с; - USB-ключ типа А; - нет режима коммуникации между устройствами.

Коммерческие решения Li-Fi систем из проведенного обзора еще не представлены на российском рынке, к тому же обладают следующими недостатками – почти все:

1. не поддерживают мобильные устройства либо устройства Android (самая популярная ОС для мобильных устройств);
2. имеют USB-ключ только типа А (кроме компании Oledcomm и PureLiFi, которая встраивает приемопередатчик в устройства), что при дальнейшем развитии системы и добавлении поддержки мобильных устройств окажется неудобным из-за того, что большинство мобильных устройств не имеют вход USB типа А;
3. имеют слишком низкую скорость для световой передачи.

Половина из представленных аналогов не имеют режима обмена данными между устройствами. Поэтому, чтобы разрабатываемая система была востребована и уникальна требуется:

- разработать универсальный приемопередатчик, который мог бы подключаться как к мобильным устройствам, так и к ПК при разработке системы в будущем;
- написать приложение для платформы Android, так как это самая популярная ОС для мобильных устройств в настоящее время;
- обеспечить режим передачи данных с одного мобильного устройства на другое.

2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Для реализации программно-аппаратного модуля передачи данных для прототипа системы Li-Fi для мобильных устройств Android необходимо определить требования к системе в целом, потому что полнота требований играют ключевую роль в успехе всего проекта. Основные группы пользователей проекта:

- люди, имеющие смартфоны Android и желающие передавать и принимать данные с другого устройства за минимальное время;
- пассажиры самолетов, моряки-подводники и другие люди в подобных условиях, где не везде разрешено или есть возможность использовать Wi-Fi или Bluetooth (не рассматривается ИК-соединение из-за его очень маленькой скорости);
- люди, заинтересованные в развитии концепта IoT.

Общее требование: графический интерфейс клиентского приложения – отображение и настройка имени и ключа безопасности в клиентском приложении, визуальное отображение этапов процесса передачи файла, получение такой информации об ведущем или ведомом устройствах, как имя устройства и ключ в мобильном приложении.

2.1 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

Функционал передачи данных с помощью света на мобильных устройствах Android можно реализовать 3 способами: на встроенном светодиоде для вспышки камеры устройства, с помощью фонарика устройства и на отдельном устройстве. Выбран последний способ, так как при использовании светодиода вспышки требуется держать телефон вертикально, что неудобно при направлении светового потока данных в приемник, кроме того и вспышка очень медленна, потому что необходимо заряжать конденсатор для ее использования, фонарик же есть не на всех устройствах, отдельный приемопередатчик более универсален – в дальнейшем развитии системы его можно будет применять также и на ПК (на ПК

нет светодиода для вспышки и фонарика, поэтому в любом случае потребуются реализовывать передачу иным способом)

Для подключения передатчика к мобильному устройству Android выбран разъем jack 3.5, так как он доступен, дешев и достаточен для целей учебного образца, кроме того он есть и на ПК, и на мобильных устройствах, что делает устройство более универсальным. Во многих аналогах используется разъем USB разных типов, но разъема USB A почти нет на мобильных устройствах Android, а USB C нет на ПК. Минусы использования jack 3.5 – медленнее USB и имеет недостаточное питание на выходе в мобильных устройствах (около 2 В), его не хватит, чтоб запитать передатчик, поэтому необходимы батарея или стабилизатор напряжения.

В качестве питания устройства можно использовать аккумулятор, батарею или стабилизатор напряжения. Аккумулятор имеет более большие габариты, чем батарея или стабилизатор, когда как устройство передатчика предполагается быть компактным. Аккумулятор самое дорогое решение из предложенных. Батарейки нужно периодически менять. Оптимальным решением является использование стабилизатора напряжения, он выигрывает и по цене и удобству использования конечным пользователем.

Передача данных в системе основана на использовании модуляции интенсивности светового потока – наиболее простой способ для реализации данной модуляции - это использование ЦАП. Чтобы управлять источником светового сигнала можно использовать широтно-импульсную модуляцию (ШИМ), цифро-аналоговый преобразователь (ЦАП) либо операционный усилитель (ОУ) в режиме компаратора.

При ШИМе модуляция светового сигнала происходит за счёт изменения ширины импульсов [23]. Возможные проблемы: при использовании ШИМ светодиод фактически часть времени горит, а часть нет, поэтому при приеме сигнала, устройство приема не сможет адекватно принять данные, поэтому

данный способ управления светодиодом не подходит для разрабатываемой системы. Для ШИМА требуется постоянная синхронизация приема и передачи, что будет сильно сказываться на скорости передачи. Пример сигнала на выходе ШИМ приведен на рисунке 9 [25].

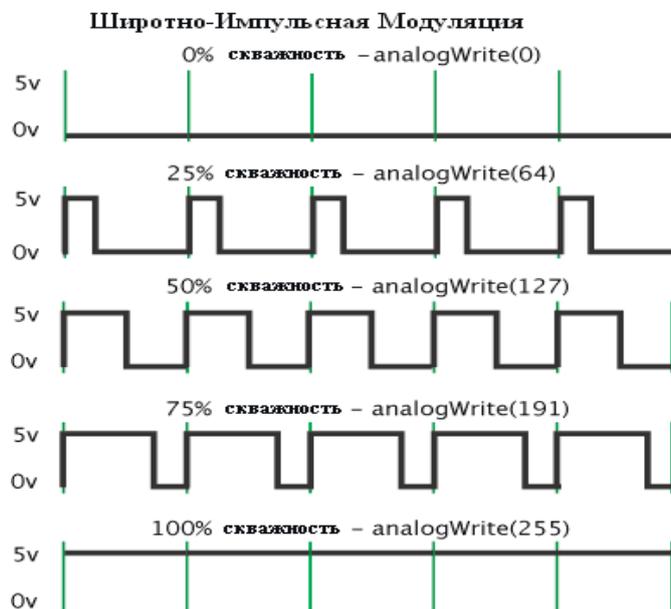


Рисунок 9 – Сигнал на выходе ШИМ

Операционный усилитель в режиме компаратора может управлять источником светового сигнала следующим образом: поступает сигнал, который сравнивается с опорным напряжением и в зависимости от результата сравнения на светодиод подается 0 или 1. Недостаток для разрабатываемой системы: при 0 светодиод подается 0 или 1. Недостаток для разрабатываемой системы: при 0 светодиод будет полностью отключен (если не используется инверсная логика, иначе при 1), тем самым на стороне приема будет неясно закончилась ли передача данных или до сих пор продолжается посылка, которая содержит 0, поэтому передающий светодиод должен иметь 3 состояния – 1) соответствующее 0, 2) соответствующее 1 и 3) соответствующее отсутствию передачи, из-за этого ОУ в режиме компаратора не подходит для управления источником светового сигнала в данной системе.

ЦАП преобразовывает некоторые цифровые данные, определенные обычно в виде двоичного кода, в их аналоговое представление, тем самым позволяет

относительно плавно изменять напряжение [24]. Для разрабатываемой системы он подходит больше всего, так как лишен проблем ШИМа и ОУ в режиме компаратора, к тому же он не уступает им по быстродействию. Пример вида сигнала на выходе ЦАП приведен на рисунке 10 (красная ломаная). Время преобразования ЦАП зависит от его разрядности, шага квантования, точности преобразования: чем выше эти параметры, тем качественнее преобразование, но и тем самым оно медленнее. В разрабатываемой системе применяются 2 ЦАП: 1-ый – для подачи данных мобильного устройства в разъем jack, 2-ой для подачи данных с передатчика в приемник. Параметры ЦАП могут варьироваться от конкретной модели мобильного устройства и микроконтроллера. В ходе исследования не было найдено конкретных моделей ЦАП, которые используются в мобильных устройствах на сегодняшний день.

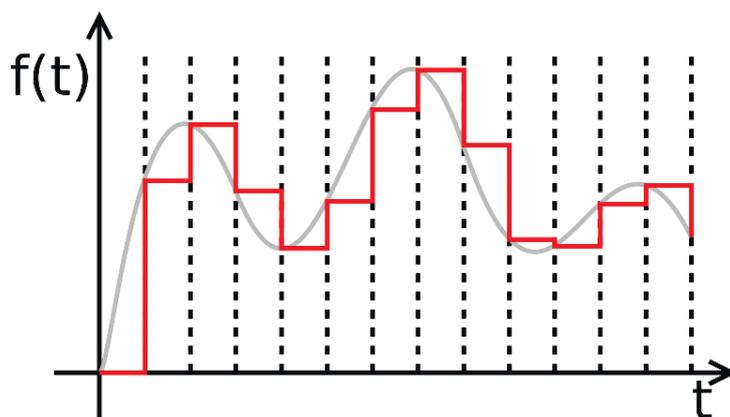


Рисунок 10 – Вид сигнала с ЦАП на фоне идеального сигнала

Еще одним аспектом проектирования варианта передающего модуля является выбор способа записи передаваемой информации в ЦАП. Передавать данные в ЦАП в микроконтроллере можно:

- в основном цикле программы, что нецелесообразно, так как микроконтроллер большую часть простаивает;
- в обработчике таймера микроконтроллера через равные промежутки времени – таймер приостанавливает работу всей программы и передает данные ЦАПу;

- через прямой доступ к памяти (ПДП). Назначение ПДП в том, чтобы по команде от периферии или ядра взять и скопировать участок памяти с одного места на другой [37]. Для этого в настройки ПДП нужно загрузить адрес откуда копировать, адрес куда копировать, какой размер блока данных копируемый за раз (байт, два байта — слово, два слова и т.д.) и сколько блоков надо скопировать. Далее контроллер ПДП ждет команды или разрешения на работу. В момент, когда получит такую команду, он начинает работу в фоновом режиме. По завершении подаст прерывание, установит флаг [37].

Очевидно, что помещение информации в ЦАП в фоновом режиме, более предпочтительно, чем использование таймера, который, чтобы поместить данные в ЦАП, приостановит выполнение всей программы. В данной работе источником для контроллера ПДП являются значения, которые получаются, после обработки данных, приходящих из мобильного устройства через jack 3.5 и преобразующихся в цифровой код с помощью АЦП. В качестве приемника данных из ПДП выступает входной регистр ЦАП. ПДП контроллер во время работы сам копирует, сам увеличивает адрес, при заполнении буфера – обнуляется и начинает переписывать буфер по кругу заново [37]. На стороне мобильного устройства данные можно поместить в ЦАП при помощи специального AudioTrack API.

В системах Li-Fi тип источника света влияет на скорость обмена данными. На сегодняшний день доступны следующие технологии освещения для Li-Fi – светодиоды (например, синие) с фосфорным покрытием, RGB-светодиоды, микродиоды, которые тоже в основном состоят из цветовой модели RGB и RGB-лазеры [26]. Зависимость скорости передачи данных с помощью Li-Fi от выбора конкретного освещения приведена на рисунке 11 из [26].

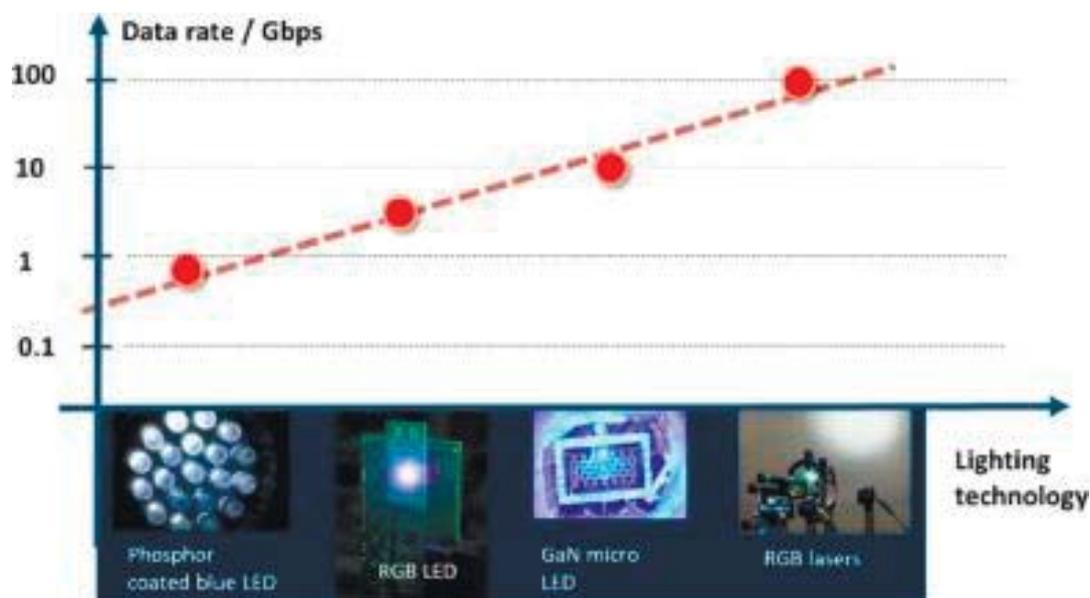


Рисунок 11 – Зависимость скорости передачи от технологии освещения

В таблице 3 дана сравнительная характеристика перечисленных выше источников света. Технологии освещения сравниваются между собой по критериям: достоинства и недостатки, цена, скорость передачи данных. Критерии достоинств и недостатков взяты следующие: распространенность на рынке, эффективность получения света (частота обновления), технические и технологические характеристики (изнашивание оборудования, энергопотребление, требования к микроклимату). В рамках работы создания прототипа системы Li-Fi для мобильных устройств Android, использовать освещение с помощью микродиодов и RGB-лазеров не представляется возможным, ввиду их цены, законодательной власти для лазера (требуется разрешение органов власти для лазерной установки) [27] и отсутствия распространенности микродиодов на рынке. Харальд Хаас в работе «LiFi – это парадигма технологии 5G» пишет, что использование светодиодов с фосфорным покрытием самый простой и экономичный способ [26]. Ввиду относительно жестких требований к нагреванию RGB-светодиодов, а также с экономической точки зрения были выбраны для системы светодиоды с фосфорным покрытием.

Таблица 3 – Сравнение технологий освещения для Li-Fi

Технология освещения	Достоинства	Недостатки	Скорость передачи, Гбит/с	Цена, руб.
Светодиоды с фосфорным покрытием	<ul style="list-style-type: none"> - широко распространены; - при нагревании свет не меняется; - нет критичных требований к микроклимату; - время переключения [64] 10^{-7} - 10^{-9} с. 	<ul style="list-style-type: none"> - часть света рассеивается в слое фосфора; - фосфор подвержен деградации [28]. 	~ 1	От 10
RGB-светодиоды	<ul style="list-style-type: none"> - высокая эффективность получения света; - нет критичных требований к микроклимату; - время переключения каждой компоненты [64] 10^{-7} - 10^{-9} с. 	<ul style="list-style-type: none"> - жесткие требования к системам питания и контролю, т.к. при нагревании «уплывает» цвет [28]. 	~ 5	От 12
Микродиоды	<ul style="list-style-type: none"> - устойчивы к старению; - высокая эффективность получения света; - нет критичных требований к микроклимату; - низкое энергопотребление [29]; - время переключения каждой компоненты [66] 10^{-9} с. 	<ul style="list-style-type: none"> - нет в массовом производстве 	~ 10	От 196

Продолжение таблицы 3

Технология освещения	Достоинства	Недостатки	Скорость передачи, Гбит/с	Цена, руб.
RGB-лазеры	- самая высокая эффективность получения света; - время переключения [65] 10^{-9} с.	- жесткие требования по охлаждению; - требования к влажности микроклимата; - необходимо разрешение органов власти для установки мощных лазеров [30].	~ 100	От 8003

Передача данных в системе основана на использовании модуляции света. Модуляция в VLC отличается от модуляции радиочастотной связи, поскольку свет нельзя закодировать по фазе или амплитуде светового сигнала [7]. Модуляция в VLC достигается с использованием изменений интенсивности света, соответствующего информации в сигнале сообщения. Если источником светового сигнала в дальнейшем планируется освещать помещение, то изменения яркости модулированного света должны выполняться таким образом, чтобы они не приводили к колебаниям, воспринимаемым человеком. Исследователем Л.У. Кханом из департамента электротехники Университета Инженерии и Технологий в его работе [7] были отмечены следующие возможные виды модуляции для VLC: амплитудная манипуляция (On-Off Keying, OOK), модуляция с помощью манипуляции цветом (Color Shift Keying, CSK), методы импульсной модуляции –

фазово-импульсная модуляция (Pulse Position Modulation, PPM), многоимпульсная позиционно-импульсная модуляция (Multipulse Pulse-Position Modulation, MPPM), экспериментальная импульсно-позиционная модуляция (expurgated pulse position modulation, EPPM), MEPPM (совокупность последних двух видов модуляции).

1. ООК - светодиоды включены или выключены в соответствии с битами в потоке. Например, 1 обозначает состояние on, а 0 – off, при этом светодиод не выключается полностью в состоянии off, но выполняется значительное снижение уровня интенсивности. Основным преимуществом использования ООК является его простая реализация. Возможен димминг (уменьшение силы света) с сохранением исходной скорости передачи данных, что позволит экономить электроэнергию, но при низком уровне света надежность связи и дальность приема уменьшается. Максимальная скорость передачи данных при данном виде модуляции достигается при уровне уменьшения силы света на 50%, предполагающем равное количество единиц и нулей в среднем; увеличение или уменьшение яркости светодиодов более 50 % будет вызывать снижение скорости передачи [31]. Есть два популярных формата ООК модуляции: с возвратом к 0 – RZ (return zero) и NRZ – без возврата к 0. Формат модуляции RZ становится все более популярным для высокоскоростной передачи данных благодаря превосходной производительности по сравнению с обычным форматом NRZ [33]. Также в работе [33] был специально проведен анализ, который показал, что вероятность битовой ошибки может быть уменьшена, если вместо NRZ используется кодирование RZ. В случае формата сигнала NRZ длинные строки в 1 с или 0 с приведут к полному отсутствию каких-либо переходов, что затруднит приемнику восстановление тактового сигнала. Формат RZ в некоторой степени смягчает эту проблему, поскольку длинные строки, состоящие из 1, по-прежнему будут производить переходы, благодаря наличию 0 после 1. С помощью ООК RZ достигается скорость

передачи 1 Гбит/с [33]. Форматы импульсов при использовании RZ и NRZ проиллюстрированы на рисунке 12 [34].

2. CSK модуляция выполняется с использованием интенсивности трех цветов в источнике RGB светодиода. CSK был предложен в IEEE 802.15.7 для повышения скорости передачи данных, является сложным в реализации методом модуляции.

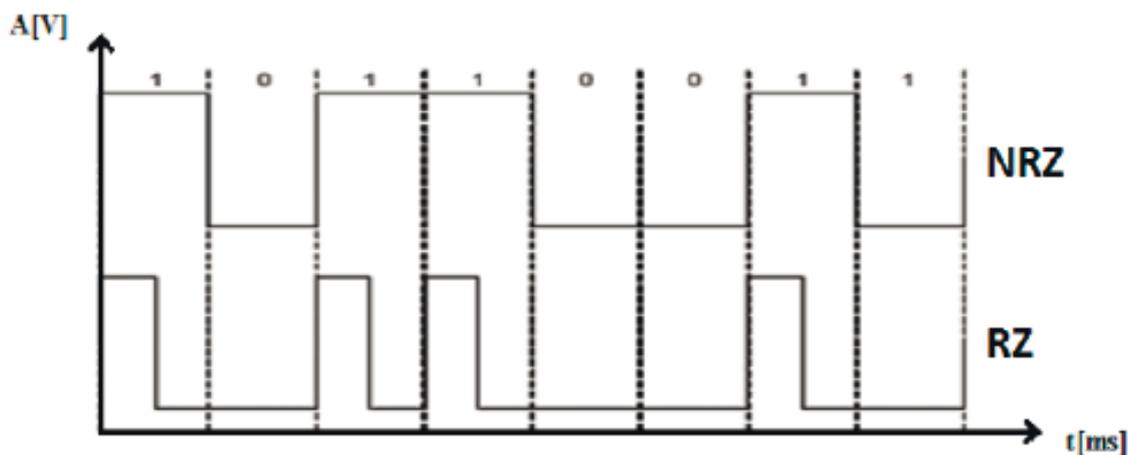


Рисунок 12 – Формы сигнала OOK NRZ и OOK RZ

3. PPM модуляция и его потомки разрабатывались из-за ограничения скорости OOK. PPM – это объединение широтно-импульсной модуляции (PWM) и DMT (Discrete Multi Tone – многочастотный способ модуляции, использующий несколько несущих частот и позволяющий организовать такое же количество каналов передачи данных [32]). PPM была основана на положении импульса. Суть: разделение длительности передачи на равные интервалы и выполнение передачи импульса в любом из интервалов. Но так как в каждом интервале присутствует только один импульс, передача данных имеет низкую скорость [7], но более высокую энергоэффективность по сравнению с OOK.
4. MPPM модуляция – представляет собой многоимпульсную PPM. Суть та же, что и у PPM, но в каждый интервал может передавать несколько импульсов.

5. EPPM и MEPPM – это многоуровневые способы модуляции, которые были введены в PPM для повышения спектральной эффективности (отношение скорости передачи данных к полосе пропускания). Рисунок 13 из [7] иллюстрирует разницу передачу данных у описанных импульсных методов модуляции.

Modulation	Transmitted Pattern	Number of Symbols
PPM		Q
MPPM		$\binom{Q}{K}$
EPPM		Q
MEPPM		$\binom{Q+N}{N}$

Q = Time slots N = Number of Levels

Рисунок 13 – Передаваемые данные импульсных методов модуляции

Авторы работ по Li-Fi (в [7], [26]) не останавливаются лишь на каком-либо одном виде модуляции, который стоит использовать, но указывают, что при использовании RGB-светодиодов лучше применять независимую модуляцию компонент R, G, B. Так как нет ограничений на использование ООК-модуляции, то выбрана ООК-модуляция вида RZ, которая поможет уменьшить ошибки на стороне приема и повысить скорость передачи для разрабатываемой системы. ООК является широко распространенной схемой модуляции, используемой для оптической связи из-за простоты реализации, эффективного использования полосы пропускания, а также экономической эффективности [33].

Приложение для Android может быть реализовано с поддержкой определенной версии Android и выше. При реализации поддержки версий 2.3.3-2.3.7 приложение будет работать на всех устройствах Android, но при этом инструментарий для разработки будет сильно ограничен, так как данная версия

Android имеет API 10. Разрабатывать для последней версии Android 9 тоже не имеет смысла, потому что сейчас число ее пользователей мало, и приложение не будет поддерживаться более ранними версиями. На 2019 год наиболее распространены ОС Android версии 6.0, что показано в таблице 4 [35]. Поэтому для разработки мобильного приложения минимальной версией следует выбрать именно 6.0.

Таблица 4 – Распространенность различных версий Android на 2019 год

Версия	Название	API	Распространенность
2.3.3 - 2.3.7	Gingerbread	10	0,3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0,3%
4.1.x	Jelly Bean	16	1,2%
4.2.x		17	1,5%
4.3		18	0,5%
4.4	KitKat	19	6,9%
5.0	Lollipop	21	3,0%
5.1		22	11,5%
6.0	Marshmallow	23	16,9%
7.0	Nougat	24	11,4%
7.1		25	7,8%
8.0	Oreo	26	12,9%
8.1		27	15,4%
9	Pie	28	10,4%

Приложение для Android может быть многопоточным и однопоточным. Основные преимущества многопоточности – выполнение долгих операций в отдельном потоке, что позитивно сказывается на производительности, а также позволяет реализовать отзывчивый интерфейс. Ввиду преимуществ многопоточности, ее следует реализовать для мобильного приложения системы.

В таблицу 5 сведены выбранные пути реализации программно-аппаратного модуля системы Li-Fi передачи данных для мобильных устройств Android.

Таблица 5 – Выбранные способы реализации

Аппаратная часть		Программная часть	
Вид передатчика:	отдельное устройство	Минимальная версия Android:	6.0
Тип подключения к мобильному устройству:	аудио-разъём jack 3.5		
Тип модулятора:	ЦАП	Свойство приложения:	многопоточное
Источник светового сигнала:	светодиод с фосфорным покрытием		
Вид модуляции:	ООК RZ		

2.2 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Для всех групп пользователей, учитывая недостатки аналогов (см. раздел 1.3), требуется следующая функциональность модуля передачи:

- передача и прием одного или нескольких файлов на другое устройство Android через устройство передатчика;
- выбор режима коммуникации и необходимых файлов;
- настройка данных о устройстве в мобильном приложении: ввод имени устройства, задание ключа;
- просмотр справки в мобильном приложении;
- прерывание передачи данных;
- просмотр информации об устройстве.

На рисунке 14 показана диаграмма вариантов использования системы в целом. Варианты использования системы описаны универсальным языком моделирования UML. На диаграмме отражены сценарии работы с системой,

которые приводят к желаемому пользователем результату (ПС Li-Fi МУ – краткое название разрабатываемой системы).

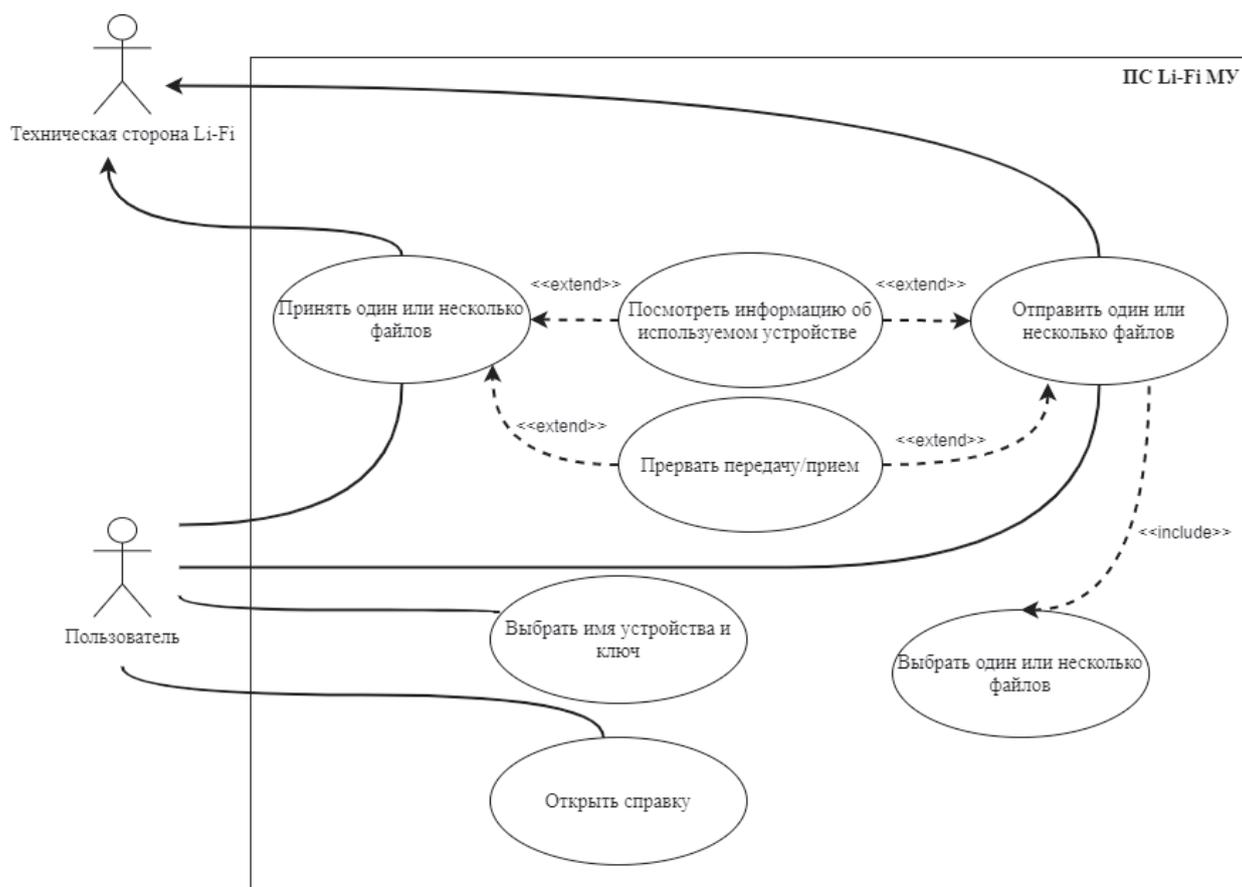


Рисунок 14 – Диаграмма прецедентов системы

В системе есть 2 актера: «Пользователь» (лицо из группы пользователей, определенных ранее), взаимодействующий с системой и «Техническая сторона Li-Fi» - аппаратная часть системы. Основные потоки событий для каждого варианта использования:

- отправить один или несколько файлов: пользователь вставляет приемопередатчик в разъем jack 3.5 -> нажимает кнопку «Режим передачи» -> на следующем экране нажимает кнопку «Включить передачу» -> нажимает кнопку «Выбрать файл(ы)» -> нажимает кнопку «Отправить» -> ключ безопасности, имя устройства и следом за ним посылки данных в зашифрованном виде отправляются на передающий модуль приемопередатчика -> передающий модуль модулирует источником света

ключ и посылки данных; также прецедент отправки файла(ов) включает в себя поведение прецедентов прерывания передачи, просмотра информации об устройстве (ключ безопасности и его имя) и выбора файла(ов); детальная технологическая цепочка отправки файла приведена на рисунке 15;

- выбрать один или несколько файлов: пользователь, находясь в режиме передачи нажимает кнопку «Выбрать файл(ы)» -> файлы преобразовываются в бинарный формат -> данные зашифровываются -> формируется посылка;
- принять файл(ы): пользователь вставляет приемопередатчик в разъем jack 3.5 -> нажимает кнопку «Режим приема» -> на следующем экране нажимает кнопку «Включить прием» -> на принимающее устройство приходит ключ передающего устройства -> если ключ на передающей и принимающей стороне одинаков, то начинается прием данных принимающим модулем приемопередатчика -> данные декодируются с помощью специальных методов в мобильном приложении-> данные собираются методами мобильного приложения в исходный файл; прецедент приема файла(ов) включает в себя поведение прецедентов прерывания передачи и просмотра информации о подключенном устройстве (от которого был успешно принят хотя бы один файл);



Рисунок 15 – Технологическая цепочка отправки файла

- посмотреть информацию об устройстве: находясь в режиме приема или передачи пользователь нажимает на имя устройства -> отображается его

ключ безопасности, используемый для установки соединения с принимающим устройством;

- отменить передачу: находясь в режиме приема или передачи, пользователь переключает switch в положение off -> если отмена со стороны передачи – передающему модулю перестает передаваться от мобильного устройства информация и на стороне приема будет приходить непрерывно 0, что будет интерпретировано как конец передачи, если отмена со стороны приема – принимающий модуль отключает устройство, которое служит для приема и удаляет данные, которые передавались;
- выбрать имя устройства и ключ: пользователь нажимает кнопку «Настройки» -> на следующем экране вводит желаемое имя устройства и ключ безопасности -> нажимает кнопку «Сохранить» -> введенные данные записываются мобильным приложением в файл, чтобы восстанавливать их при каждом запуске приложения;
- открыть справку: пользователь нажимает кнопку «Справка» -> на следующем экране отображается справочная информация по системе.

2.3 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Для всех групп пользователей можно выделить программные и аппаратные нефункциональные требования. Данные требования определяют свойства либо ограничения системы. К аппаратным требованиям относятся:

- мерцание источника светового сигнала не должно быть заметно человеческому глазу;
- приемный модуль должен иметь быстродействующий фоторезистор;
- передатчик образует с приемником единое устройство – приемопередатчик;
- приемопередатчик должен быть оснащен высокопроизводительным и энергоэффективным микроконтроллером;

- в учебных целях приемопередатчик должен использовать интерфейс TRRS jack 3.5 для общения с мобильным устройством, потому что jack 3.5 дешев и распространен;
- приемопередатчик должен быть эргономичен;
- приемопередатчик должен быть легким и компактным.

Программные нефункциональные требования:

- поддержка устройств с ОС Android версии 6.0 и выше;
- при коммуникации данные должны шифроваться;
- устройства должны обмениваться данными только после совпадения защитного ключа принимающей и передающей сторон;
- пользовательский интерфейс мобильного приложения должен быть эргономичен;
- должна быть поддержка вертикальной и горизонтальной ориентации экрана мобильным приложением.

Со стороны 3 группы пользователей, кроме вышеперечисленных требований, включается требование по грамотной реализации проекта, чтобы можно было развивать и дополнять систему в будущем, а не начинать с нуля.

3. ПРОЕКТИРОВАНИЕ

3.1 ДИАГРАММА КЛАССОВ ПРИЛОЖЕНИЯ

Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. На основе диаграммы использования, представленной на рисунке 14, составлена диаграмма классов для мобильного приложения разрабатываемой системы (см. рисунок 16). Диаграмма классов была описана языком UML. Взаимосвязи объектов классов на диаграмме:

- ассоциация - показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому; классы «Меню» и «Передача», «Меню» и «Прием», «Меню» и «Настройка», «Меню» и «Справка»;
- агрегация - разновидность ассоциации при отношении между целым и его частями; классы «Прием» и «Инф-ия об используемом уст-ве», «Передача» и «Инф-ия о используемом уст-ве»; в данном случае при подключении одного устройства к другому они обмениваются собственной информацией;
- композиция - более жесткий вариант агрегации, имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов. Если контейнер будет уничтожен, то всё его содержимое будет также уничтожено; классы «Настройка» и «Инф-ия о используемом уст-ве»; если пользователь удалит данные о своем устройстве, то объект класса «Инф-ия об используемом уст-ве», хранящий аналогичные данные, также будет удален.

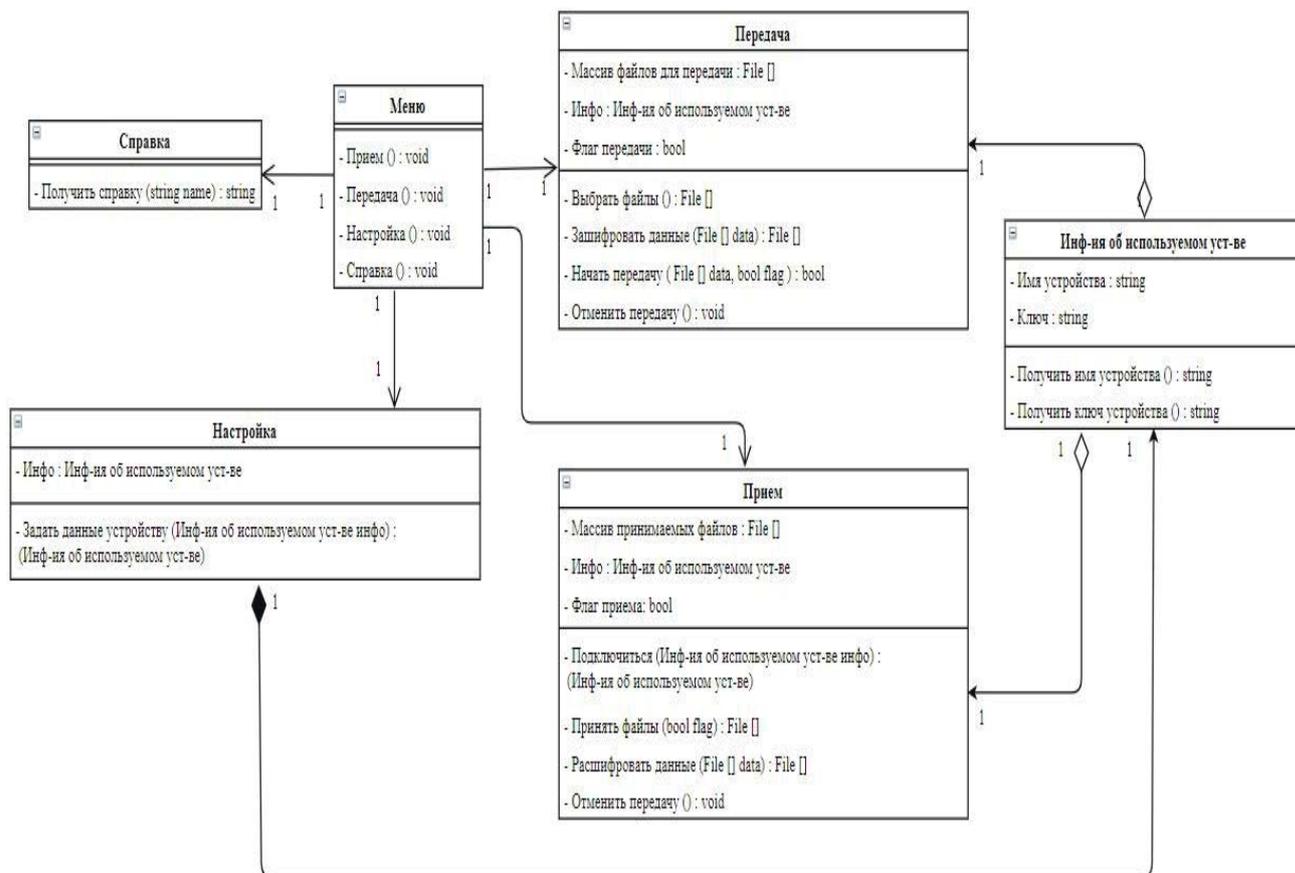


Рисунок 16 – Диаграмма классов мобильного приложения ПС Li-Fi МУ

3.2 СТРУКТУРНАЯ СХЕМА ПЕРЕДАЮЩЕГО МОДУЛЯ

Аппаратная часть разрабатываемой системы состоит из различных электронных элементов, для выбора которых необходимо выделить основные функциональные модули аппаратной части. В ее состав входит:

- система управления – микроконтроллер;
- питание устройства;
- обмен информацией с системой;
- источник светового сигнала;
- фотодетектор.

В работе разрабатывается передающий модуль, в него входят все вышеперечисленные модули кроме фотодетектора. Структурная схема передающего модуля представлена на рисунке 17.



Рисунок 17 – Структурная схема передающего модуля

3.3 АРХИТЕКТУРА СИСТЕМЫ

Общая аппаратная часть комплекса состоит из следующих компонентов: 1) источник питания, 2) микроконтроллер, который включает в себя передающий и принимающий модули, 3) источник светового сигнала, 4) фотодетектор, 5) интерфейс для обмена данными с мобильным устройством. Программная часть состоит из таких модулей, как: 1) ядро мобильного приложения, 2) графическое ядро - UI, 3) API для работы с аудиоразъемом jack 3.5, 4) файл, в котором хранится имя устройства пользователя и ключ безопасности. Архитектура программно-аппаратного комплекса представлена на рисунке 18.

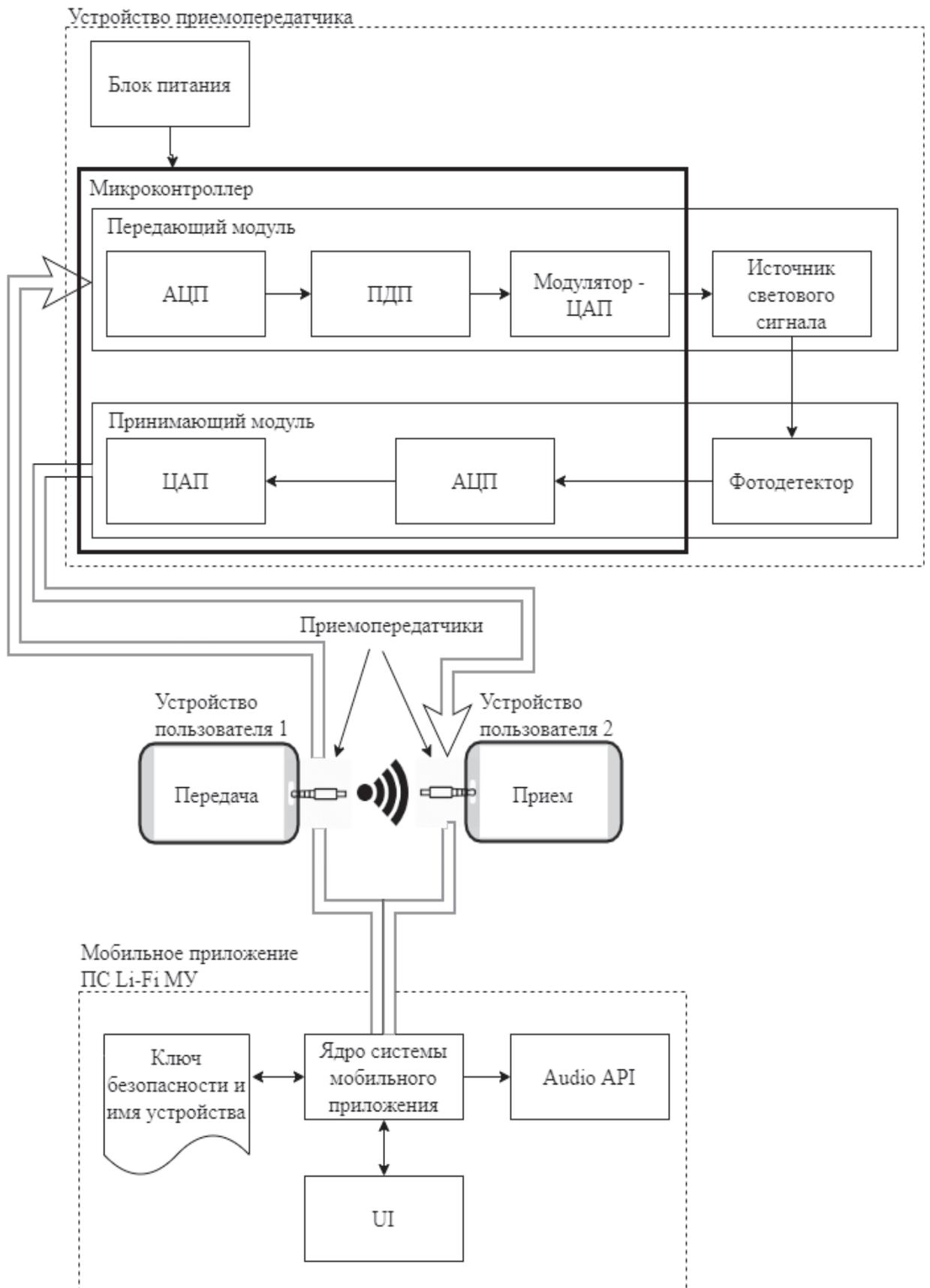


Рисунок 18 – Диаграмма компонентов системы

3.4 ПОДБОР КОМПОНЕНТОВ

Добавление интеллектуальных функций в современную радиоэлектронную аппаратуру базируется на использовании микропроцессоров (МП), программируемых логических интегральных схем, приборов типа «система на кристалле» и других современных цифровых устройств [36]. Подобная интеграция позволяет автоматизировать процессы измерения, управления, контроля, регулирования и обработки информации, а также обеспечить такие свойства приборных комплексов, как многофункциональность, модифицируемость, адаптивность, обучаемость и ряд других. В отличие от персональных компьютеров, вычислительным ядром которых служат универсальные высокопроизводительные микропроцессоры, для интеграции в различные устройства применяют микроконтроллеры (МК). В связи с широкими возможностями МК, это один из основных компонентов аппаратной части разрабатываемой системы.

Li-Fi – это высокоскоростная технология передачи данных, поэтому необходима высокая скорость реагирования приемопередатчика, поэтому требуется высокопроизводительный МК. Кроме того, Li-Fi считается дешевой технологией, поэтому при выборе МК будет учитываться его стоимость, чтобы минимизировать затраты, но сохранить надежность работы устройства и обеспечить достаточную производительность. В ходе поиска в Интернете было найдено множество высокопроизводительных МК, в рамках работы провести сравнительный анализ для всех не представляется возможным, поэтому для анализа выбраны наиболее распространенные:

- 8-разрядные микроконтроллеры AtTiny13 производства Atmel;
- 8-разрядные микроконтроллеры ATmega48 производства Atmel;
- 32-разрядные микроконтроллеры серии ATSAM E70 фирмы Atmel;
- высокопроизводительные микроконтроллеры PIC16C7 фирмы Microchip;
- 32-битные микроконтроллеры Infineon XMC1100 и XMC4100;

- семейство 32-разрядных микроконтроллеров STM32F100 производства компании STMicroelectronics;
- энергосберегающие 32-разрядные микроконтроллеры EFM32GG компании Silicon Labs.

В таблице 6 приведен их сравнительный анализ по критериям: частота микроконтроллера в МГц, что является критерием производительности, набору необходимой периферии – несколько каналов ЦАП и АЦП, ПДП, наличию низкого потребления энергии и цене.

Таблица 6 – Сравнительный анализ микроконтроллеров

МК	Частота, МГц	Число каналов ЦАП	Число каналов АЦП	ПДП	Низкое потребление энергии	Цена, руб.
AtTiny13 [38]	20	-	4	-	+	от 35
ATmega48 [39]	16	-	8	-	+	от 50
ATSAM E70 [40]	до 300	2	24	+	-	от 780
PIC16C7 [41]	24	-	5	-	-	от 211
ХМС1100 [42]	32	-	6-12 в зависимости от модели	-	-	от 145
ХМС4100 [43]	80	2	суммарно 9	+	-	от 275

Продолжение таблицы 6

МК	Частота, Мгц	Число каналов в ЦАП	Число каналов АЦП	ПДП	Низкое потребление энергии	Цена, руб.
STM32F100 [44]	24	суммарно 4	10-16 в зависимости от модели	+	+	от 160
EFM32GG [45]	48	1	7	+	+	от 423

Как видно из таблицы 5, для данного проекта по всем показателям подходят только МК из семейств STM32F100 и EFM32GG. Так как STM32F100 более дешевые МК и весьма быстродействующие, то для реализации передатчика выбран STM32F100RBT6 (см. рисунок 19) по цене 160 рублей. ЦАП данной модели МК имеет следующие характеристики, влияющие на быстродействие [44]:

- время пробуждения из выключенного состояния – 6,5 – 10 мкс;
- время установления (между самым высоким и самым низким значением) – 3-4 мкс.

Характеристики АЦП микроконтроллера, влияющие на быстродействие [44]:

- время пробуждения из выключенного состояния – 1 мкс;
- время калибровки АЦП – 83 мкс;
- время преобразования по регулярным каналам – 2 мкс;
- время преобразования по инжектированным каналам – 3 мкс;
- время выборки – 239,5 мкс.

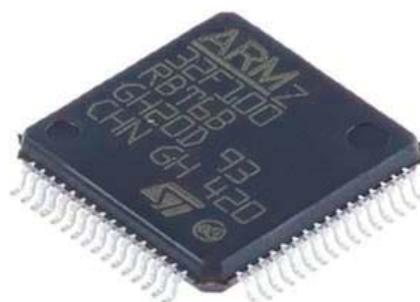


Рисунок 19 - STM32F100RBT6

Для тактирования МК могут применяться внешний высокочастотный генератор (HSE) и внутренний высокочастотный генератор (HSI) [51]. HSI является RC цепочкой с частотой 8МГц. Точность HSI значительно ниже внешнего генератора HSE, при этом на нее сильно влияют изменения температуры. Источником тактирования для HSE служит внешний тактовый сигнал, который может быть от 4 до 32 МГц. Это может быть кварцевый резонатор, тактовый генератор и т.д. HSE отличается от HSI высокой температурной стабильностью, точностью частоты и долговечностью, исходя из этого, для тактирования МК был выбран HSE на 8 МГц, так как он указан в даташите устройства [44]. Была куплена модель HSE - 8000 МГц (усечен.) HC-49S (US), Кварцевый резонатор с параметрами [46]:

- резонансная частота – 8 МГц;
- нагрузочная емкость – 32 пФ;
- рабочая температура - -20...70 °С;
- длина корпуса – 11,05 мм;
- диаметр (ширина) корпуса – 4,65 мм;
- вес – 1 г;
- цена – 15 р.

Источник светового сигнала является вторым основным компонентом передатчика. В главе 1 для достижения цели работы было определено использование светодиодов с фосфорным покрытием. Для качественной модуляции данных и дальнейшего корректного их приема был выбран сверхъяркий светодиод холодного свечения TDS-P001L4U15 STAR. Светодиод имеет встроенную рассеивающую тепло пластину, так как из-за высокой яркости сильно греется. Его стоимость – 95 рублей, что бюджетно, по сравнению с остальными подобными светодиодами. TDS-P001L4U15 STAR имеет следующие параметры: [46]:

- максимальный ток $I_{led} = 350$ мА;

- мощность $P_{\text{led}} = 1$ Вт;
- напряжение питания $U_{\text{led}} = 5$ В;
- световой поток $\Phi = 100$ лм;
- габариты - $8 (\pm 0,2) \times 14,5 (\pm 0,3) \times 5,1 (\pm 0,3)$ мм;
- вес – 1,6 г.

Изображение светодиода представлено на рисунке 20.



Рисунок 20 - TDS-P001L4U15 STAR

В соответствии со световым потоком, имеем площадь покрытия, рассчитанную по формуле (2), выраженную из формулы (1) [68].

$$\Phi = E * S * k_h, \quad (1)$$

где: E - нормативное значение освещенности помещения, в данном случае в учебных целях выбрано 300 Лк (тип помещения – офис общего назначения), S - площадь покрытия, м. кв., k_h - коэффициент высоты потолка (до 2,7 м – 1; 2,7-3 м – 1,2; 3-3,5 м – 1,5; 3,5-4 – 2).

При условии, что потолки до 2,7 м, а источник света только в приемопередатчике площадь покрытия:

$$S = \frac{\Phi}{E * k_h} = \frac{100}{1,2 * 300} = 0,28 \text{ кв. м} \quad (2)$$

Источником питания устройства передатчика в главе 1 был выбран стабилизатор напряжения. Микроконтроллер STM32F100RBT6 питается от 3,3 В, с разъема jack 3.5 может выходить от ~ 1,8 до 5 В, в зависимости от того подключено устройство к смартфону или ПК, следовательно, необходим стабилизатор напряжения с выходным напряжением 3,3 В и питанием от 1,8 В.

Кроме того, для максимальной яркости светодиода требуется 5 В, поэтому необходим второй стабилизатор с выходным напряжением 5 В и питанием от 3,3 В – чем ярче горит светодиод, тем лучше данные будут восприниматься на стороне приема. Из-за малых входных напряжений стабилизаторы будут иметь более высокую стоимость, чем остальные. Выбор стабилизаторов был остановлен на типе LDO – линейных стабилизаторах с малым падением напряжения. Они имеют небольшие размеры, низкий уровень шумов на выходе и высокую точность выходного напряжения [47]. Для выбора стабилизаторов использовалась информация из рисунка 21 [48]. Были выбраны следующие фиксированные стабилизаторы с наименьшими габаритными размерами и стоимостью:

- LD39015M33R с основными параметрами [46]:

$$U_{\text{inputMin}} = 1,5 \text{ В};$$

$$U_{\text{inputMax}} = 5,5 \text{ В};$$

$$U_{\text{output}} = 3,3 \text{ В};$$

габариты - 3x1,75x1,3 мм;

вес – 0,05 г;

цена – 43 рубля.

- STLQ50C50R с основными параметрами [48]:

$$U_{\text{inputMin}} = 2,3 \text{ В};$$

$$U_{\text{inputMax}} = 12 \text{ В};$$

$$U_{\text{output}} = 5 \text{ В};$$

габариты – 2,2x1,35x1 мм;

вес – 0,013 г;

цена – 76 рублей.

Наименование	Входное напряжение, В	Выходное напряжение, В	Вых. ток, мА	Падение напряжения ¹ , мВ	Потреб. ток (тип), мкА	SVR ² , дБ	Шум на выходе ³ , мкВ _{RMS} /В	Enable /Power Good	Рекомендуемые характеристики вых. конденсатора		Корпус
									Емкость, мкФ	ESR, Ом	
LD3985	2,5...6	1,22; 1,8; 2,5; 2,6; 2,7; 2,8; 2,9; 3,0; 3,3; 4,7	150	0,4...60	85	50	30	+/-	1...22	0,005...5	SOT23-5L, TSOT23-5L, CSP (1,57×1,22 мм)
LDS3985	2,5...6	1,5; 1,8; 2,5; 2,8; 3,0; 3,3; 5,0	300	0,4...150	85	50	30	+/-	2,2...22	0,005...5	SOT23-5L, DFN6 (3×3 мм)
LD39015	1,5...5,5	0,8; 1,0; 1,2; 1,25; 1,5; 1,8; 2,5; 3,3	150	до 80	18	62	29	+/-	0,33...22	0,15...2	SOT23-5L, SOT666, CSP (1,1×1,1 мм)
LD59015	2,4...5,5	0,8; 1,2; 1,5; 1,8; 2,5; 3,0; 3,3	150	до 150	31	76	20	+/-	0,33...22	0,05...8	SOT323-5L
LD39020	1,5...5,5	0,8...5,0	200	до 200	20	65	45	+/-	0,22...22	0,05...0,9	DFN4 (1×1 мм)
LD39115J	1,5...5,5	1,0; 1,2; 1,4; 1,5; 1,8; 2,5; 2,8; 3,0; 3,3	150	80 (100 мА)	20	67	30	+/-	1...22	0,1...1,8	CSP4 (0,8×0,8 мм)
LD3913054	1,5...5,5	1,0; 1,2; 1,8; 2,5; 2,9; 3,0; 3,3; 4,1; Adj	300	до 300	55 (1)	65 (48)	38 (100)	+/-	0,33...22	0,1...4	CSP4 (0,69×0,69 мм)/DFN6 (1,2×1,3 мм)
LD39050	1,5...5,5	2,5; 3,3; Adj	500	до 200	20	62	30	+/+	1...22	0,05...0,8	DFN6 (3×3 мм)
LD39100	1,5...5,5	1,2; 2,5; 3,3; Adj	1000	до 200	20	65	85	+/+	1...22	0,05...0,15	DFN6 (3×3 мм)
LD39200	1,25...6,0	3,3; Adj	2000	до 135	100	50	24	+/+	1...22	0,05...1,2	DFN6 (3×3 мм), DFN8 (4×4 мм)
LDK120	1,9...5,5	0,8; 1,0; 1,1; 1,2; 1,5; 1,8; 2,5; 2,8; 2,9; 3,0; 3,1; 3,2; 3,3; 3,5; Adj	200	до 150	30	55	51	+/-	1...22	0...10	SOT23-5L, SOT323-5L, DFN6 (1,2×1,3 мм)
LDK150	1,9...5,5	0,8; 1,1; 1,2; 1,5; 1,8; 2,5; 2,9; 3,0; 3,2; 3,3; Adj	300	до 200	30	55	51	+/-	1...22	0...10	SOT23-5L, SOT323-5L, DFN6 (1,2×1,3 мм)
LDK220	2,5...13,2	1,2...1,8; 2,5...3,3; 3,6; 4,0; 4,2; 5,0; 6,0; 8,5; 9,0; Adj	200	до 200	40	45	20	+/-	1...22	0,05...0,9	SOT23-5L, SOT323-5L, DFN6 (1,2×1,3 мм)
LDLN015	2,1...5,5	1,0; 1,2; 1,5; 1,8; 2,5; 2,8; 3,0; 3,3	150	до 86	17	89	6,3...9,9	+/-	0,33...10	0,05...0,6	DFN6 (2×2 мм)
LDCL015	1,8...5,5	3,3; Adj	150	до 70	120	51	40	+/-	Любая	Любой	SOT23-5L
STLQ50	2,3...12	1,8; 2,5; 3,3; 5,0; Adj	50	до 350	3	30	560	-/-	0,22...4,7	0...10	SOT323-5L
STLQ015	1,5...5,5	1,2; 1,5; 1,8; 2,5; 2,8; 3,0; 3,1; 3,3	150	до 112	1	30	75	+/-	0,47...10	0,056...6	SOT666
ST715	2,5...24	2,5; 3,3; Adj	85	до 500	4,15	45	95	-/-	0,47...1	0...1,5	SOT23-5L, SOT323-5L, DFN8 (3×3 мм)

Рисунок 21 - Основные электрические характеристики LDO-стабилизаторов

ST

Важно подобрать конденсаторы для стабилизаторов и кварцевого резонатора, которые будут выступать в роли фильтров от помех. Для данной системы наиболее подходящими конденсаторами будут являться керамические конденсаторы. Они отличаются низким последовательным сопротивлением (ESR) и минимальной погрешностью номинала по сравнению с электролитическими или танталовыми конденсаторами, являются неполярными – можно применять в схемах с переменным током, имеют низкую цену по сравнению со слюдяными конденсаторами, обладают миниатюрным размером [50]. Был выбран кер. чип.

Конденсатор 0402 X5R 2.2мкФ 2.5В 20%, GRM155C80E225ME15D в количестве по 2 штуке для каждого стабилизатора на каждый вывод, с параметрами [46]:

- номинальная емкость – 2,2 мкФ;
- рабочее напряжение – 2,5 В;
- габариты - 1.0x0,5x0,55 мм;
- цена – 1,20 руб. за 1 шт.;
- вес – 0,65 мг.

Для кварцевого резонатора выбран кер. чип. конденсатор 1206 X6S 22 мкФ 4В 10%, GRM31CC80G226KE19L в количестве 2 штук для каждого вывода, с параметрами [46]:

- номинальная емкость - 22 мкФ;
- рабочее напряжение – 4 В;
- габариты – 3,2x1,6x1,5 мм;
- цена – 4 руб. за 1 шт.;
- вес – 0,95 мг.

После добавления стабилизатора для светодиода, светодиод будет всегда гореть максимально ярко, так как на нем постоянное напряжение. Чтобы управлять яркостью светодиода необходимо добавить транзистор, на базу которого будет идти напряжение с ЦАП, таким образом, будет управление величиной протекающего тока через светодиод. Эмпирически было установлено, что падение напряжение на светодиод $U_{VD} = 2,9$ В. Отталкиваясь от значения параметра тока $I_{led} = 100$ мА, при котором светодиод горит уже с максимальной яркостью, подобран транзистор КТ645Б. Его основные параметры [46]:

- $U_{кэ} = 40$ В – напряжение коллектор – эмиттер;
- $U_{кб} = 40$ В – напряжение коллектор – база;
- $U_{эб} = 5$ В – напряжение эмиттер – база;
- $I_{max} = 300$ мА – максимально допустимый постоянный ток;
- $\beta = 80 \geq$ - коэффициент передачи тока;

- $U_{\text{нкэ}} = 0,05 \text{ В}$ – напряжение насыщения коллектор-эмиттер;
- $U_{\text{нбэ}} = 1,2 \text{ В}$ – напряжение насыщения база-эмиттер;
- цена – 4 рубля;
- вес – 0,3 г.

Необходимо рассчитать номинальные значения резисторов для правильной работы схемы устройства. Для схемы устройства, исходя из 2-го закона Киргофа [49], имеем следующие уравнения (3) и (4):

$$5\text{В} = I_{\text{к}} * R_2 + U_{\text{VD}} + U_{\text{нкэ}}, \quad (3)$$

$$U_{\text{упр}} = I_{\text{б}} * R_1 + U_{\text{нбэ}}, \quad (4)$$

где $U_{\text{упр}}$ – напряжение с выхода микроконтроллера, которое управляет светодиодом, оно находится в пределах $\sim 2,5 - 3,2 \text{ В}$ ($2,5$ – логически 0, $3,2$ – логическая единица), выбираем максимальное значение – $3,2 \text{ В}$.

Имея все данные для уравнения (3), найдем минимальное значение сопротивления для резистора R_2 , расчет проведен в выражении (5):

$$R_2 = \frac{5 - 0,05 - 2,9}{0,1} = 21 \text{ Ом} \quad (5)$$

Вычислив ток базы $I_{\text{б}}$ в выражении (7), найдем минимальное сопротивление для резистора R_1 в выражении (8), выразив его значение из уравнения (4). Ток базы $I_{\text{б}}$ КТ645Б при токе светодиода 100 мА рассчитывается по формуле (6):

$$I_{\text{б}} = \frac{I_{\text{max using}}}{\beta}, \quad (6)$$

где $I_{\text{max using}}$ соответствует I_{led} , равно $0,1 \text{ А}$, β – коэффициент усиления транзистора.

$$I_{\text{б}} = \frac{100}{80} = 1,25 \text{ мА}, \quad (7)$$

$$R_1 = \frac{3,2 - 1,2}{1,25} = 1600 \text{ Ом} = 1,6 \text{ кОм}, \quad (8)$$

Основываясь на проведенных вычислениях, выбраны маломощные SMD резисторы мощностью 1 Вт с номинальными сопротивлениями $1,8 \text{ кОм}$ и 22 Ом :

- $0,062\text{Вт } 0402 \text{ } 1,8 \text{ кОм}, 1\%, \text{ Чип резистор (SMD) [46]}$;
- $0,062\text{Вт } 0402 \text{ } 22 \text{ Ом}, 1\%, \text{ Чип резистор (SMD) [46]}$.

Выбранные резисторы имеют одинаковые параметры, такие как:

- длина корпуса – 1 мм;
- ширина (диаметр) корпуса – 0,5 мм;
- вес – 0,01 г;
- цена – 0,9 рубля.

SMD резисторы значительно меньше по размеру, чем их традиционный аналог. Малый размер необходим для сборки компактного приемопередатчика.

Устройство передатчика подсоединяется к смартфону пользователя через аудио разъем jack 3.5. Необходим штекер jack 3.5 для передатчика. Данный штекер может быть моно или стерео. Для разрабатываемой системы подойдет лишь стерео с 4 контактами: по левому каналу будут приходить данные из смартфона на АЦП МК, через правый - смартфон будет принимать полученные данные с ЦАПа МК, через линию микрофона напряжение будет подаваться на стабилизатор, который запитает устройство, и общий вывод будет заземлен. Изображение распиновки штекера jack 3.5 представлено на рисунке 22 [53]. Был выбран штекер с наименьшей стоимостью из представленных [52]:

- штекер jack 3.5 стерео, пластик 4 контакта, СТИА;
4 контакта;
цена – 40 рублей.

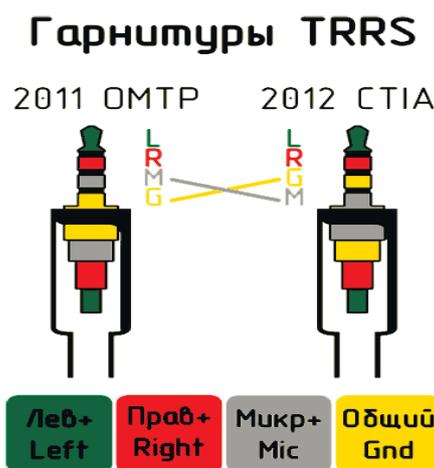


Рисунок 22 – Распиновка jack 3.5

Финансовые затраты на разработку аппаратной части устройства приведены в таблице 7, в затраты не входит стоимость изготовления корпуса устройства и печатной платы.

Таблица 7 – Затраты на аппаратное обеспечение

Обозначение на схеме	Устройство	Номинал	Цена 1 товара, руб.	Кол-во	Итоговая цена
U1	МК STM32	STM32F100RBT6B	160,00	1	160,00
D1	Светодиод	TDS-P001L4U15, серия STAR 7000-8000K	95,00	1	95,00
U3	Стабилизатор напряжения	LD39015M33R, $U_{input} = 1,5 \dots 5,5 \text{ В}$, $U_{output} = 3,3 \text{ В}$	43,00	1	43,00
U2	Стабилизатор напряжения	STLQ50C50R, $U_{input} = 2,3 \dots 12 \text{ В}$, $U_{output} = 5 \text{ В}$,	76,00	1	76,00
C3 – C6	Керамический SMD конденсатор	0402 X5R 2,2мкФ 2,5В 20%, GRM155C80E225M E15D	1,20	4	4,80
C1 - C2	Керамический SMD конденсатор	1206 X6S 22мкФ 4В 10%, GRM31CC80G226K E19L	4,00	2	8,00
X1	Кварцевый резонатор	HC-49S, 8 МГц	15,00	1	15,00
Q1	Транзистор	КТ645Б	4,00	1	4,00

Продолжение таблицы 7

Обозначение на схеме	Устройство	Номинал	Цена 1 товара, руб.	Кол-во	Итоговая цена
R1	Резистор SMD	0,062 Вт 0402 1,8 кОм, 1%,	0,9	1	0,9
R2	Резистор SMD	0,062 Вт 0402 22 Ом, 1%,	0,9	1	0,9
JACK1	Аудио - штекер	3,5мм, стерео, 4 контакта, СТИА	40	1	40
Итоговая стоимость без учета корпуса, руб.					447,6

3.5 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА

После выбора всех компонентов передающего модуля была спроектирована схема электрическая принципиальная для всей аппаратной части в целом. Она представлена на рисунке А.1 в приложении А. Спецификация всех компонентов в схеме представлена в приложении Б.

Для разработанной принципиальной схемы была рассчитана максимальная потребляемая мощность. Она образуется из суммы максимальных потребляемых мощностей каждого элемента. Потребляемые мощности элементов указаны в даташитах к ним. Также для расчетов всех элементов, кроме конденсаторов, можно применять формулу (9) [69]. Для расчета мощностей конденсаторов применяют формулу (10) [70]:

$$P = I * U, \quad (9)$$

$$P = \frac{W}{t} = \frac{C * U^2}{2 * t}, \quad (10)$$

где P – максимальная потребляемая мощность элемента, I – сила тока, U – напряжение питания, C – емкость конденсатора, W – энергия заряженного конденсатора, t – единица времени.

В результате расчетов ниже представлены значения мощностей для всех элементов схемы:

- транзистор КТ645Б – 0,5 Вт;
- фоторезистор GL5537 – 0,1 Вт;
- светодиод TDS-P001L4U15 STAR – 1 Вт;
- резисторы SMD 0402 – 0,062 Вт;
- стабилизатор LD39015M33R – 0,099 мВт;
- стабилизатор STLQ50C50R – 0,0069 мВт;
- кварцевый резонатор HC-49S – 0,001 Вт;
- микроконтроллер STM32F100RBT6B – 540 мВт;
- конденсатор GRM155C80E225ME15D – 6,875 мкВт;
- конденсатор GRM31CC80G226KE19L – 176 мкВт.

Просуммировав полученные значения, получили потребляемую мощность устройства, которая составляет 2,33 Вт.

3.6 РАЗВОДКА ПЕЧАТНОЙ ПЛАТЫ

Чтобы сделать устройство более компактным по размерам, выбран вариант двухслойной платы. Преимущество двухсторонних печатных плат - наличие двух проводящих слоев, что позволяет плотнее компоновать элементы и увеличивает трассировочную способность печатной платы. Компоненты расположены таким образом, чтобы в результате плата имела минимальные габариты.

Разводка печатной платы выполнена в САПР PROTEUS ARES. Схема разведенной печатной платы представлена на рисунке В.1 в приложении В. Плата имеет размеры 47x50 мм. Минимальная ширина проводника 0,25 мм, что соответствует 3 классу точности печатной платы [54]. Для изготовления платы

взят стеклотекстолит двухсторонний Rexant СФ1.5-2-0.35 09-4021, толщиной 1,5 мм, размерами 10x10 мм и стоимостью 87 рублей [55].

3.7 ПРОЕКТИРОВАНИЕ КОРПУСА УСТРОЙСТВА

Устройство передатчика, которое совместно с приемником образует приемопередатчик, должно быть заключено в корпус, отвечающий функционалу и конструктивным особенностям. Корпус должен иметь 3 отверстия:

- для разъема TRRS jack 3.5, с помощью которого происходит подключение к мобильному устройству Android;
- для светодиода, который передает данные в виде световых сигналов;
- для фотодетектора, который принимает световые сигналы.

Отверстия расположены в следующем порядке: на противоположной стороне от отверстия для TRRS jack 3.5 находятся два отверстия для источника светового сигнала слева и фотодетектора справа. Эти отверстия должны быть защищены от попадания влаги или других механических повреждений прозрачным стеклом, которое не будет влиять на качество передаваемых и принимаемых данных. Все компоненты должны быть надежно закреплены.

Корпус необходимо выполнить из черного непрозрачного материала, чтобы уменьшить влияние помех из внешнего мира. Также материал должен быть достаточно крепким, чтобы выдерживать, например, падение устройства с небольшой высоты. Материалы используемые для изготовления корпуса должны быть эргономичны. Корпус имеет габариты 50x53x17,8 мм. На рисунках 23-24 представлена модель полученного корпуса в САПР SolidWorks.

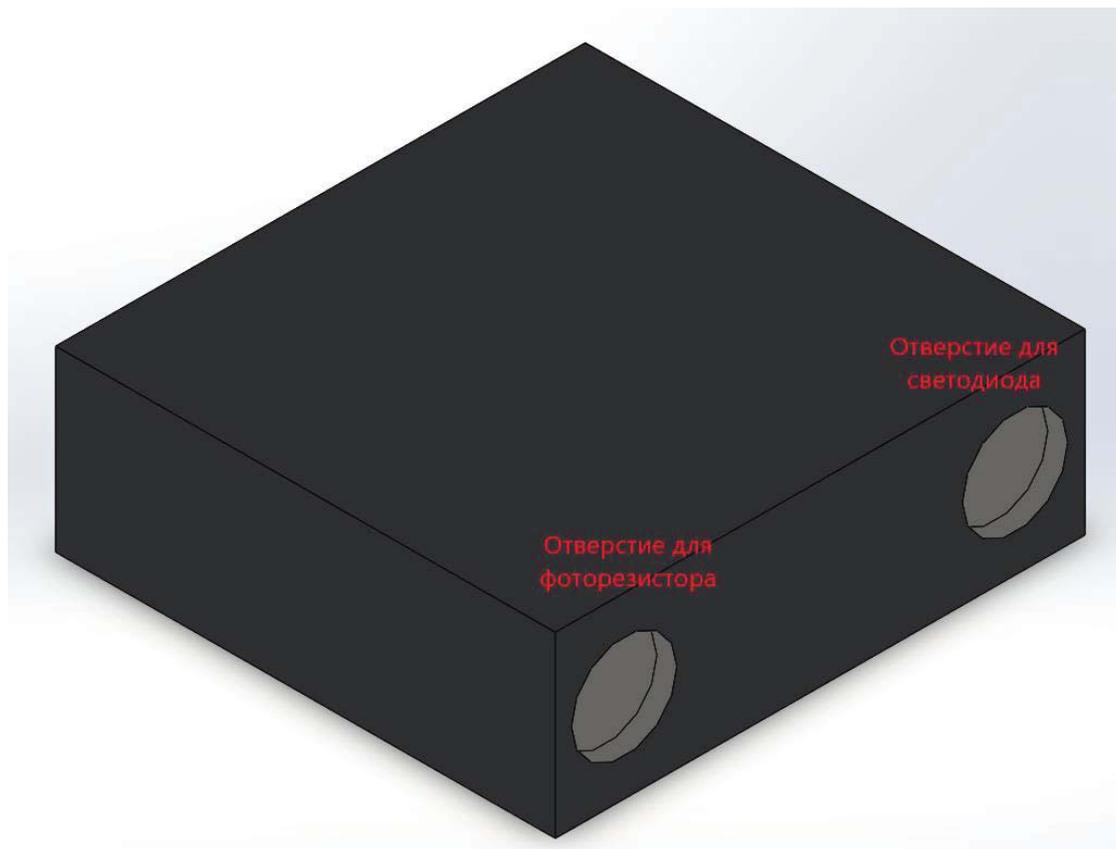


Рисунок 23 – Корпус устройства с внешней стороны

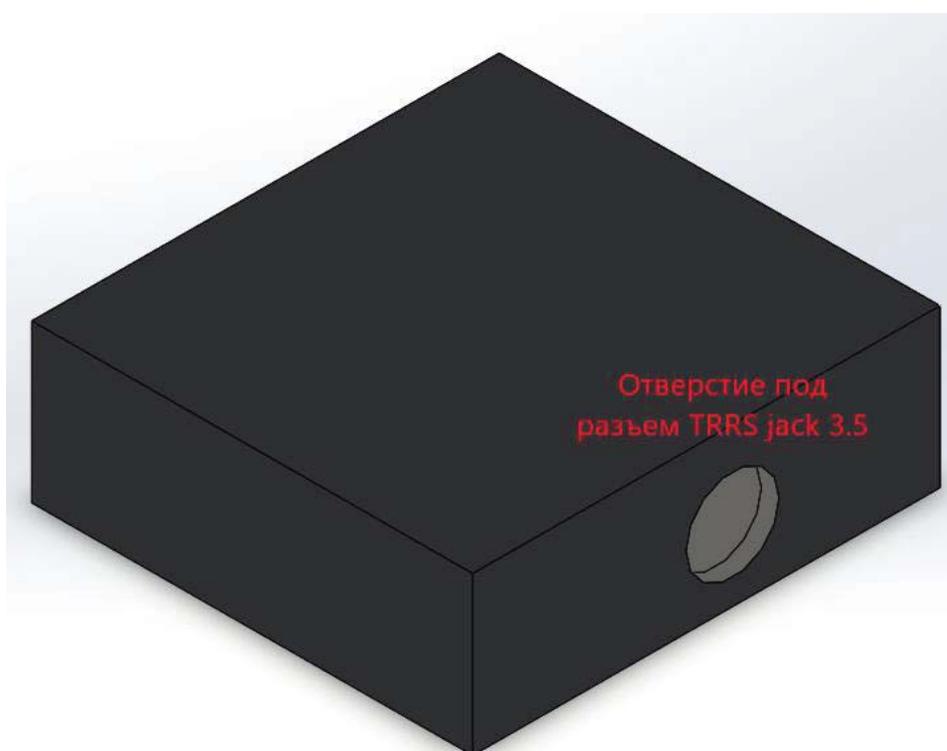


Рисунок 24 – Корпус устройства со стороны штекера

3.8 ВЫБОР ИНСТРУМЕНТОВ РАЗРАБОТКИ

Для аппаратной части системы нужно выбрать инструменты для программирования МК STM32. Для разработки программ на данный МК в настоящее время существует множество IDE как платных (IAR, CrossWorks, SOMNIUM DRT Cortex-M и т.д), так и бесплатных (Atollic TrueStudio, CooCox, STM32Cube). В целях экономии средств коммерческие IDE не рассматриваются. Бесплатные имеют практически одинаковый функционал, кроме, например, STM32Cube, который имеет встроенный кодогенератор, поэтому выбор той или иной IDE зависит от предпочтений разработчика. Для реализации устройства передатчика Li-Fi была выбрана CooCox IDE (CoIDE) совместно с компилятором C. Основные преимущества CoIDE это [56]:

- наличие отладчика;
- возможно программирование на более высоком уровне без прямого указания регистров;
- поддерживает все устройства STM32, в основе которых лежат ядра ARM Cotex-M0, M0+, M3 и M4;
- не имеет ограничений на размеры кода;
- поддерживаются все необходимые для STM32 конфигурационные файлы и прошивки также, как и интуитивное аппаратное обеспечение, такое как платы STM32 Nucleo или комплекты Discovery Kit, используемые для отладки и проектирования;
- IDE основана на Eclipse, поэтому имеет все ее плюсы.

В качестве инструмента разработки программной части для устройств Android выступают IDE и язык программирования. Есть достаточное количество языков для разработки Android-приложений. В настоящий момент Kotlin признан официальным языком для разработки для Android [57]. В Kotlin можно использовать все существующие фреймворки и библиотеки Java. В отличие от Java (предыдущий официальный язык программирования для Android) он [57]:

- null-безопасен;
- имеет функции-расширения, которые можно использовать вместо наследования;
- имеет специально предназначенные классы для работы с данными;
- во множестве случаев не требует явно приводить типы;
- позволяет использовать лямбда выражения и пр.

В качестве языка программирования для приложения Android был выбран Kotlin по вышеизложенным причинам. Kotlin напрямую поддерживается средой разработки Android Studio, а также IntelliJ IDEA. Android Studio является официальной IDE от Google для разработки Android-приложений, базируется на IntelliJ IDEA, к тому же она направлена только на разработку для Android, тогда как IntelliJ IDEA поддерживает разработку еще на таких языках, как PHP, Scala, Ruby и пр. (набор языков зависит от версии IDE), в следствии этого, IntelliJ IDEA более медленная чем Android Studio. Всё это дает основание выбрать для разработки такую IDE, как Android Studio.

3.9 АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

Перед отправкой данные зашифровываются. В рамках работы был проведен поиск и анализ методов шифрования данных, которые чаще всего применяются для систем связи Li-Fi. Обнаруженные методы, их достоинства и недостатки сведены в таблицу 8.

Таблица 8 – Сравнение методов шифрования для Li-Fi

Название	Достоинства	Недостатки
Семейство Secure Hash Algorithm (SHA) [58]	- надежность тем выше, чем выше версия SHA; - сложность реализации повышается при увеличении версии	- быстродействие понижается с версией SHA; - на первых версиях SHA хэш-функции имеют коллизии, - удлинение передаваемых сообщений
Алгоритм Диффи–Хеллмана и его производные (ECDH) [58]	- надежный; - сложности в реализации	- медленный, так как применяются операции деления
Шифр Цезаря [59]	- быстрый; - простой в реализации	- ненадежный
RC4–алгоритм [60]	- достаточно быстрый; - простой в реализации	- средняя степень надежности
XOR–шифрование [61]	- очень быстрый; - простой в реализации	- ненадежный

Чем более надежен метод шифрования, тем он более медленен, а также сложен в реализации. Световые сигналы, которые применяются для передачи данных в Li-Fi, не могут проходить через стены и прочие преграды, поэтому украсть пользовательские данные, находясь в соседнем помещении не получится. Эта особенность систем связи Li-Fi уже обеспечивает защиту данных, в той или иной степени. Поэтому для обеспечения компромисса между скоростью передачи и защитой данных в системе выбрано для шифрования выбран алгоритм RC4. Блок-схема алгоритма шифрования для дальнейшей реализации приведена на рисунке 25. Шифрование данных происходит на стороне мобильного устройства.



Рисунок 25 – Блок-схема шифрования данных

В силу свойств операции XOR, которая используется для шифрования данных, для дешифровки данных применим такой же алгоритм, как и для шифрования, в таком случае, операция XOR применяется для каждого байта зашифрованной информации и ключа.

Для отправки файлов необходимо формировать посылки данных. В учебных целях был выбран размер посылки равный 256 бит. В соответствии с OOK ZR модуляцией, данные требуется дополнять битами: после 1 вставлять 0, а после 0 -

1. Поэтому число битов без дополнительных, а также без старт- и стоп- битов равно 110. Если число пользовательских бит менее 110, то в таком случае посылка дополняется 0 и соответствующими 1 до требуемого размера. Блок-схема алгоритма формирования посылок представлена на рисунке 26.

Требуется проводить контроль приходящей информации, чтобы выявить ошибки при передаче. Для этого могут использоваться разные методы: посимвольный контроль четности, поблочный контроль четности, вычисление контрольной суммы различного типа. Первые два метода очень просты в реализации, но также ненадежны. Суть их в том, что в конце посылки либо блока посылки добавляется дополнительный бит четности, если проверяется четность, либо нечетности, если наоборот. В посылке при передаче может измениться один бит с 0 на 1, а другой с 1 на 0, и методы контроля четности не смогут обнаружить ошибку. CRC (циклически избыточный код) имеет множество разновидностей. CRC8 – простейшая, контрольная сумма образуется суммированием количеством единичных бит в посылке, и на стороне приема это значение сравнивается с пришедшим количеством единичных бит. Данный вариант имеет те же недостатки, как и методы контроля четности. Но вместо обычного суммирования, может использоваться полином, от которого получают остаток от деления посылки данных, тем самым, он является контрольной суммой, это более надежный способ выявлять ошибки. CRC16, 32, 64 – в таких видах CRC используют только остаток от деления на специальный полином, степень которого соответствует типу CRC. Данные методы более надежны, с возрастанием типа CRC. Крайне важно выбрать правильный полином, от него зависит качество использования данной CRC, существуют стандартные испытанные значения [67]. В данной работе для учебных целей использование CRC16 достаточно отвечает требованиям: посылка данных имеет размер 256 бит, из которых 2 для старт-бита, 2 для стоп-бита, 32 бита – будет отведено под контрольную сумму и остальное 220 – данные.

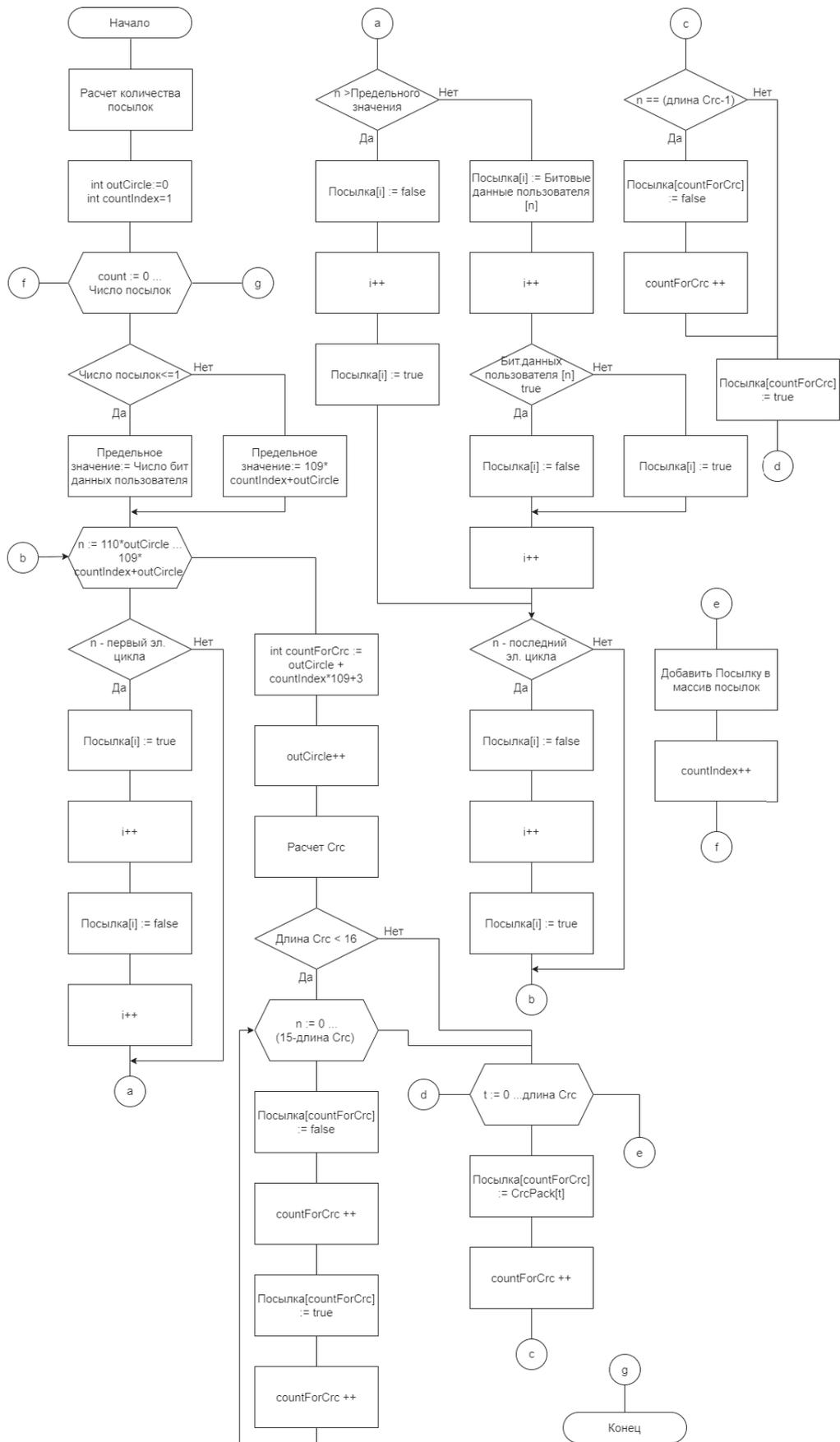


Рисунок 26 – Блок-схема алгоритма формирования посылки

Выбран стандартный полином для CRC16: $x^{16} + x^{15} + x^2 + 1$ [71]. Алгоритм вычисления контрольной суммы и применения ее к посылке данных в данном случае элементарен:

1. Получить остаток от деления посылки данных на 11000000000000101 (98309).
2. Если битовое представление остатка менее 16 бит, то дополнить до 16, вставив в начало нулевые биты.
3. Добавить дополнительные биты для соблюдения требований модуляции к полученному значению.
4. Дополнить посылку значением контрольной суммы из шага 2.

Для разграничения логических единиц данных, в массив посылок между ними вносятся посылки из 256 нулевых бит, что позволяет на стороне приема распознать что именно пришло: ключ безопасности, имя устройства, имя файла, формат файла либо сами данные файла. Алгоритм отправки посылок (которые представляют из себя файлы, выбранные пользователем для передачи) на стороне мобильного приложения представлен на рисунке 27.

Алгоритм работы передатчика аппаратной части представлен на рисунке 28. Суть алгоритма: АЦП передатчика принимает данные по битам с аудиоразъема jack 3.5, затем, в зависимости от значения бита, помещается число 4000 или 2000 (для соответствия ООК RZ модуляции) в буфер для ПДП, из которого ПДП передает данные на ЦАП, ЦАП изменяет состояние светодиода, тем самым модулируя данные.

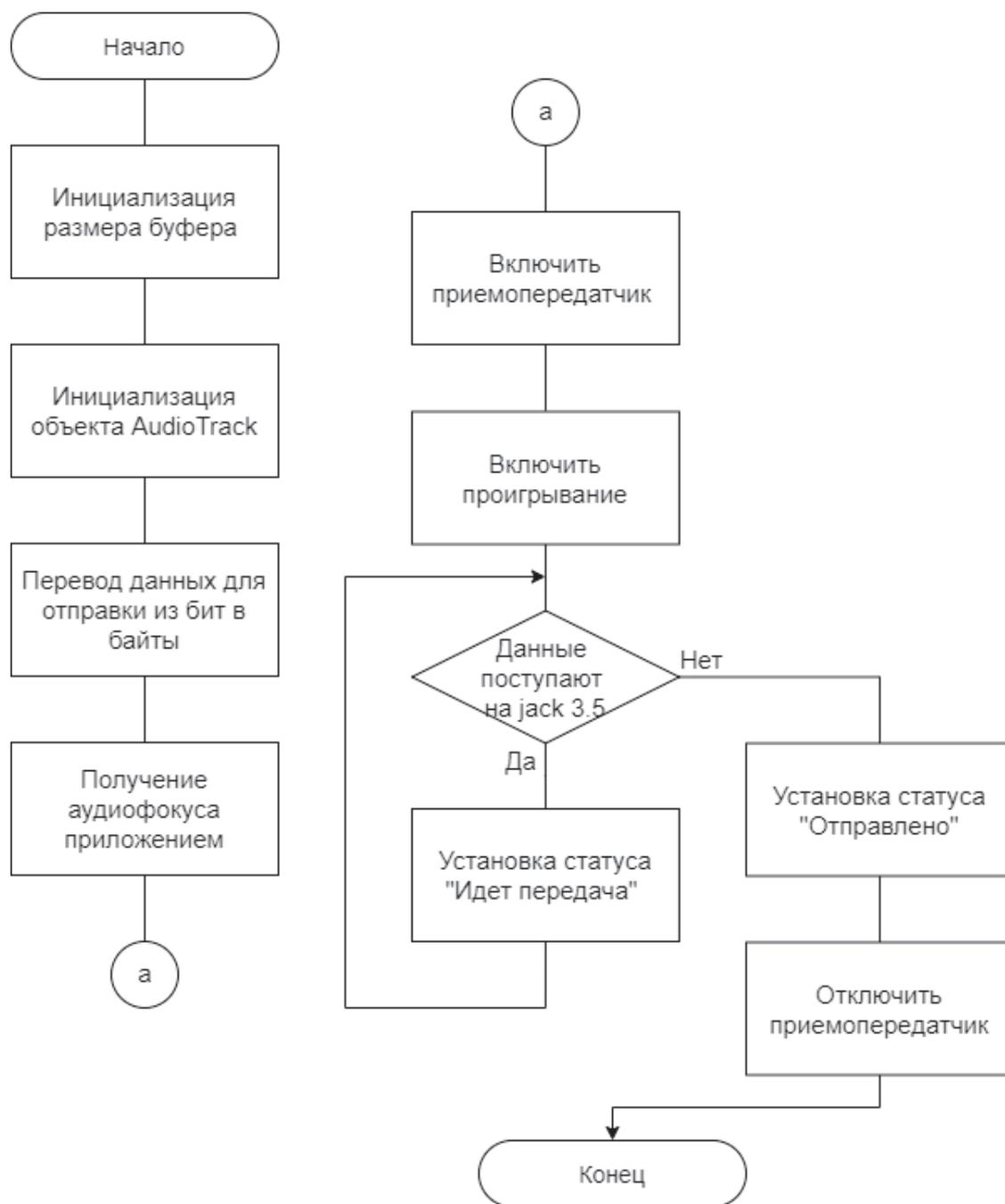


Рисунок 27 – Блок-схема алгоритма передачи на стороне мобильного приложения

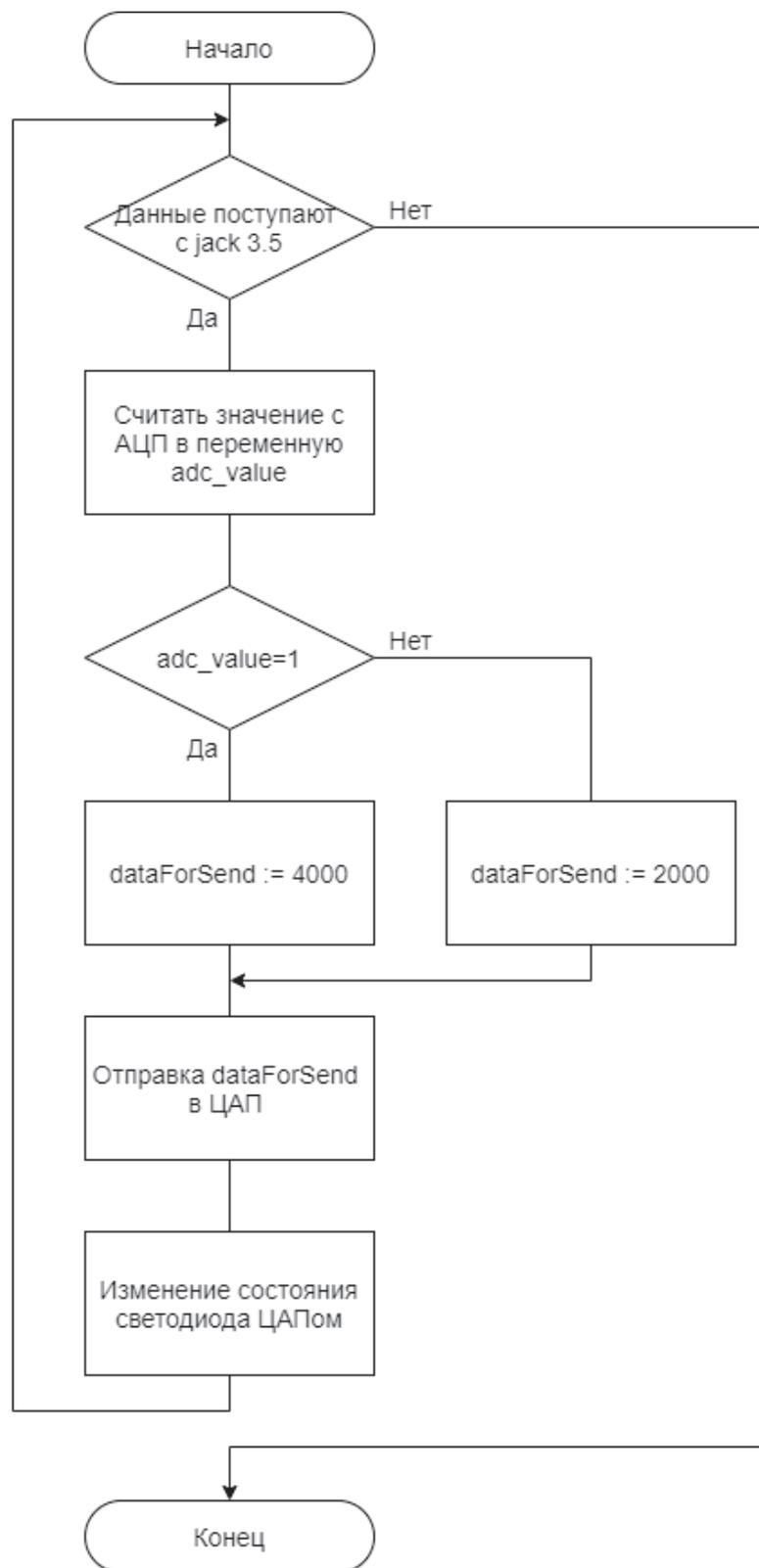


Рисунок 28 – Алгоритм работы передатчика (аппаратная часть)

3.10 ОПИСАНИЕ ДАННЫХ

Прототип системы нацелен на передачу данных любых форматов с одного мобильного устройства Android на другое. Входные данные в мобильное приложение представляют из себя файлы, который выбирает пользователь на своем устройстве через проводник, после чего:

1. Имена файлов преобразуются в бинарный формат;
2. Данные зашифровываются;
3. Для каждого имени файла формируется посылка из 256 бит, состоящая из старт-бита, битов данных, контрольной суммы и стоп-бита, а также дополнительных нулевых битов после 1 и единичных после 0 в соответствии с ООК ZR модуляцией;
4. Шаги 2 и 3 повторяются для названия форматов файлов;
5. Шаги 2 и 3 повторяются для данных файла.

После перечисленных шагов данные в промежуточном бинарном, зашифрованном состоянии подаются на передатчик аппаратной части системы в виде аналогового сигнала. АЦП передатчика преобразовывает данные в цифровой вид. Далее происходит обработка данных в соответствии с требованиями к модуляции. Выходные данные с передатчика представляют из себя световой сигнал (аналоговый) различной интенсивности, который излучает светодиод: 0 соответствует 2000 значению с ЦАП, 1 – 4000, если с ЦАП не приходят данные, то светодиод постоянно горит с максимальной яркостью, что равно значению 4096 с ЦАП МК.

4. РЕАЛИЗАЦИЯ

В рамках выполнения ВКР, были реализованы: приложение на устройства Android, прошивка на МК STM32F100RBT6, собрано устройство приемопередатчика на макетной плате. Далее приведены скриншоты и фотографии, демонстрирующие основные результаты работы.

4.1 ПРИЛОЖЕНИЕ ANDROID

Для приложения Android было разработано 4 активити: меню - MainActivity, режим передачи – MainActivitySend, режим настроек – MainActivitySetting и справка - MainActivitySpravka.

MainActivity – главное меню, из которого пользователь выбирает режим приложения. MainActivitySpravka содержит справочную информацию о приложении.

MainActivitySetting – экран для задания таких пользовательских настроек, как ключ безопасности и имя устройства. Настройки хранятся в файле dataUser.txt. Для файла используется режим MODE_PRIVATE, что дает гарантию, что другие приложения не будут иметь доступ к этому файлу. По умолчанию имя устройства пользователя «User1», пароль «User123». Заданы следующие ограничения: для имени устройства – более 3 и менее 11 символов, для ключа безопасности – более 4 и менее 11. При несоответствии вводимых данных заданным параметрам, пользователь увидит соответствующее всплывающее сообщение – объект класса Toast. Для работы с файлами использовались классы FileInputStream и InputStreamReader для чтения из файла и FileOutputStream и OutputStreamWriter для записи в файл. Если файл с данными будет удален, то при вводе новых данных, он будет создан.

MainActivitySend – активити, в котором находится основной функционал. Через него происходит управление питанием устройства приемопередатчика,

отправка имени устройства и ключа безопасности, добавление/удаление файлов и их отправка, также в процессе передачи меняется статус отправляемых данных.

Интерфейс пользователя описанных экранов, а также экрана для режима приема в различных ориентациях представлены на рисунках 29, 30, 31, 32.

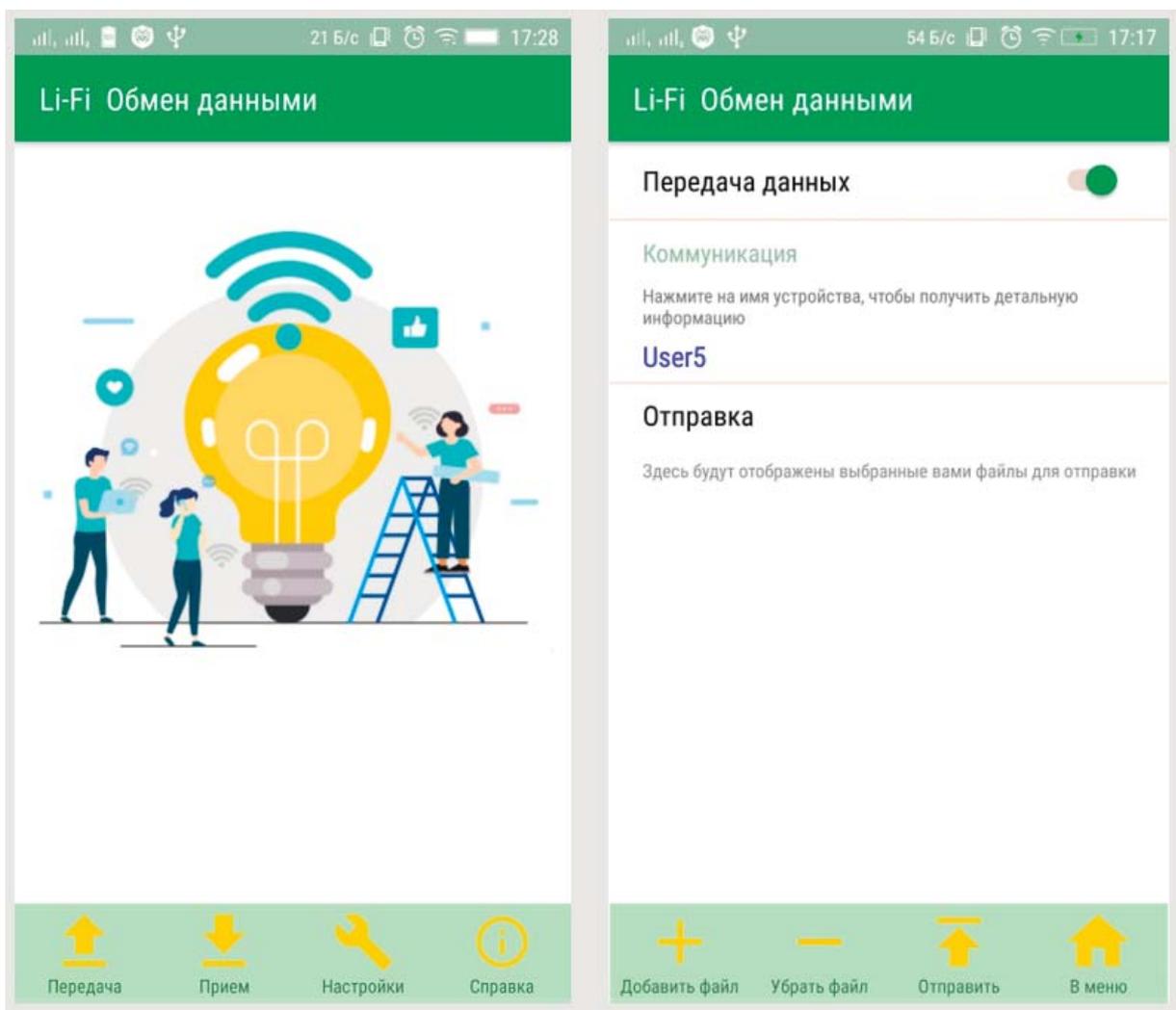


Рисунок 29 – Интерфейс пользователя: меню и режим передачи (вертикальная ориентация)

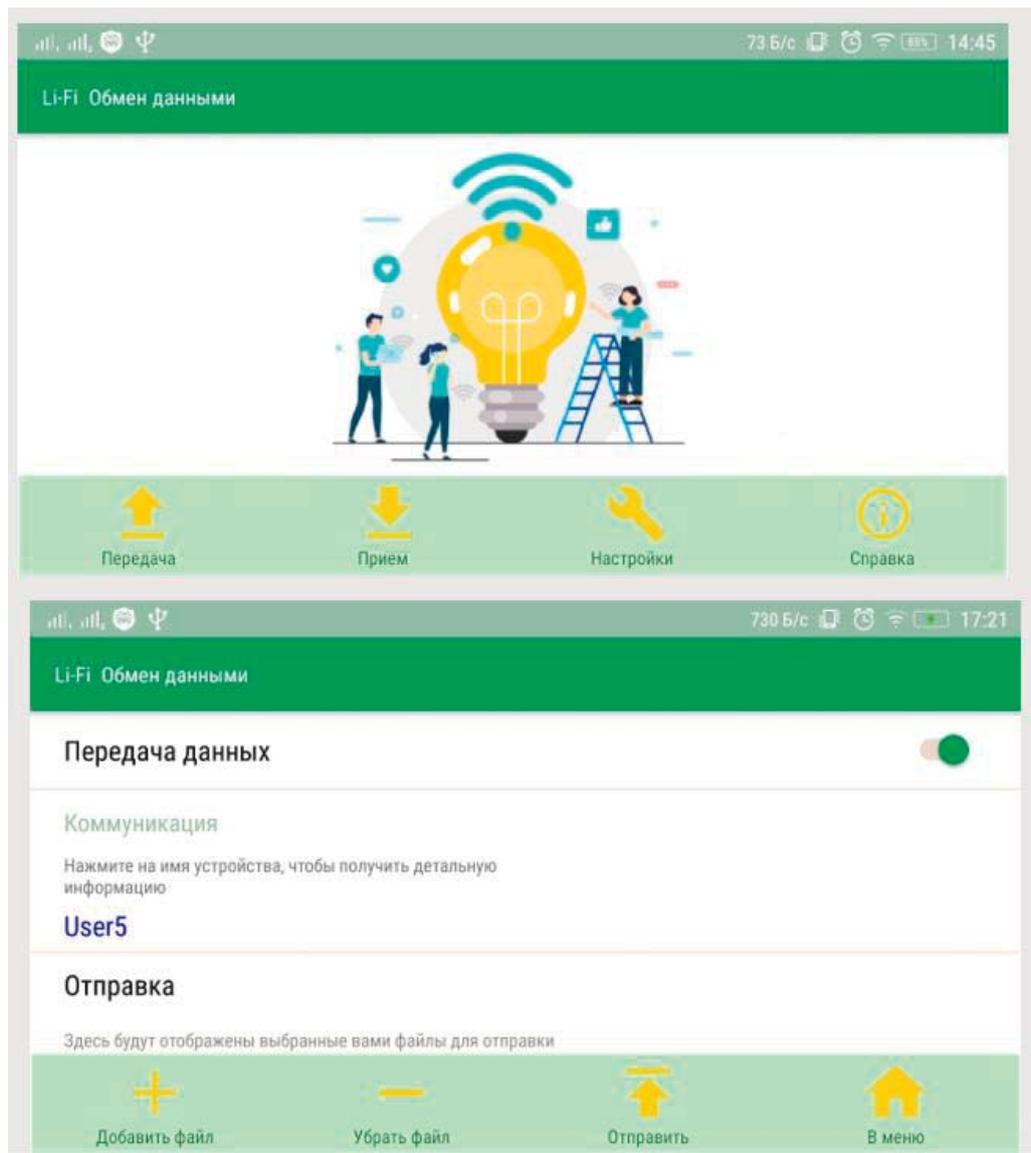


Рисунок 30 – Интерфейс пользователя: меню и режим передачи (горизонтальная ориентация)

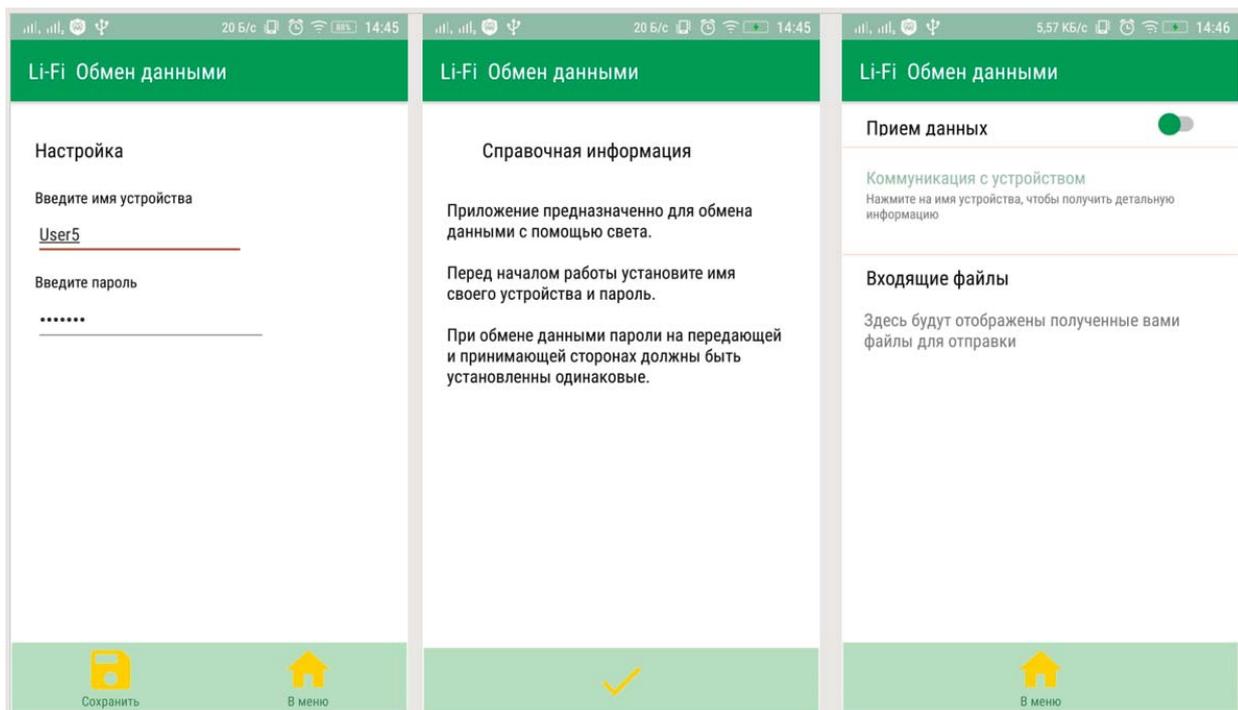


Рисунок 31 – Интерфейс пользователя: настройки, справка и режим приема (вертикальная ориентация)

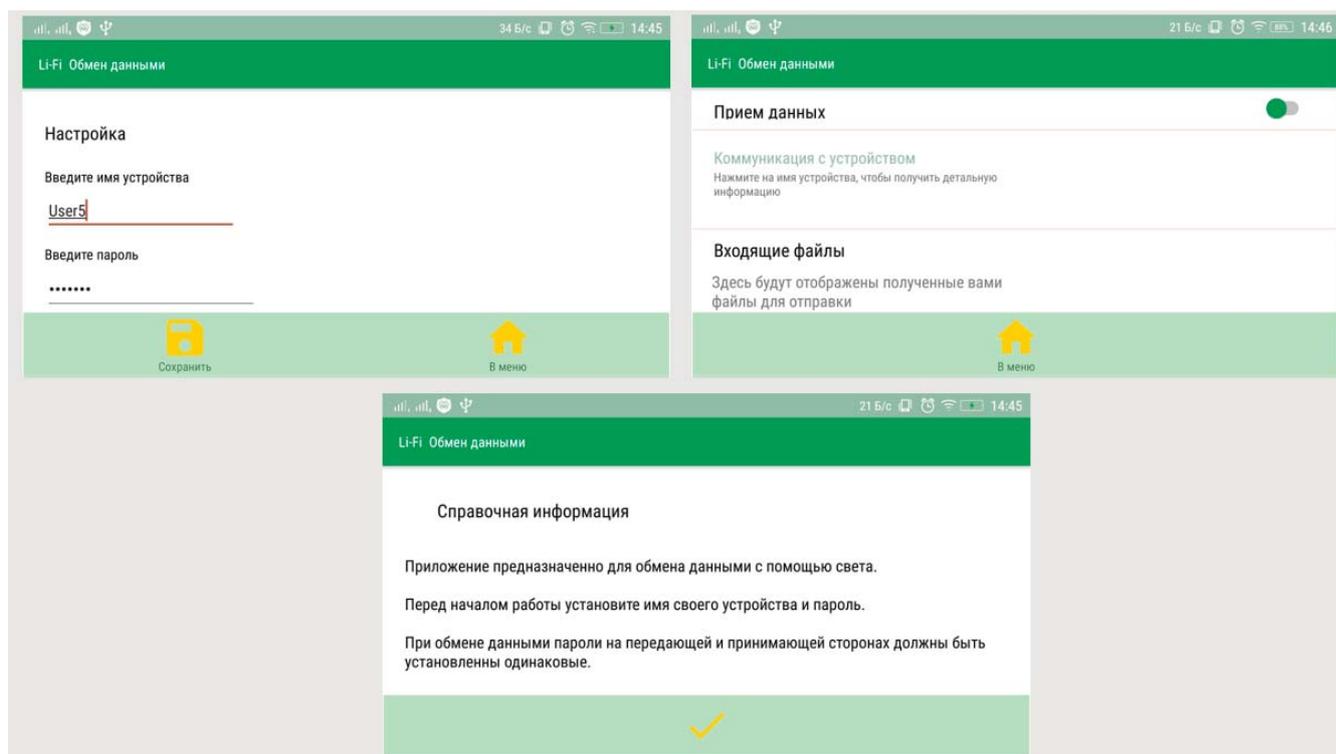


Рисунок 32 – Интерфейс пользователя: настройки, справка и режим приема (горизонтальная ориентация)

После нажатия в меню кнопки «Передача данных» вызывается переопределенный метод `onCreate(savedInstanceState: Bundle?)` из `MainActivitySend`, он служит для создания активити для передачи данных, кроме того, в нем реализовано чтение из файла «dataUser.txt» пользовательских данных, дальнейшее их шифрование и формирование посылок для пароля и имени устройства, также в методе реализованы слушатели: на событие изменения компонента `switchSend` и на событие клика по `nameDeviceText` - надписи имени устройства. Вид слушателя на изменение состояния переключателя приведен на листинге 1.

Листинг 1 – Слушатель события переключения `switchSend`

```
switchSend.setOnCheckedChangeListener{buttonView, isChecked->
    if (isChecked) flagSend=true
    else{
        flagSend = false
        audioManager.setMicrophoneMute(true)
        minusFileTxt.setEnabled(true)
        buttonMinusFile.setEnabled(true)
        infoForUser("Вы отключили режим передачи",
Toast.LENGTH_SHORT) } }
```

Если файла не существует, либо пользователь очистил его вручную, то переключатель `switchSend` для изменений будет заблокирован, а на экран выведется всплывающее Toast-сообщение о том, что перед работой необходимо задать имя устройства и пароль безопасности в настройках приложения. Для отображения всех всплывающих сообщений реализован метод `fun infoForUser(messageForUser:String, long:Int)`, у которого 1-ый аргумент – текст сообщения, 2-ой – время отображения сообщения. При изменении положения `switchSend` меняется флаг `flagSend`, который служит для определения, включен ли режим передачи или нет. В зависимости от состояния `flagSend` метод отправки данных определяет отправлять ли данные, отменить передачу или же вовсе ее не начинать, пока пользователь не переведет `switchSend` в On. При переключении `switchSend` в положение Off `flagSend` будет установлен в false, выведется всплывающее сообщение с соответствующей информацией, если во время этого, происходила передача файлов, то она будет прервана, и рядом с отправляемыми

файлами изменится статус на «Прервано», устройство приемопередатчика будет отключено.

Управление питанием приемопередатчика происходит с помощью работы с каналом для микрофона через метод класса AudioManager `setMicrophoneMute()`: при отключении микрофона – приемопередатчик будет также выключен (в метод передается значение `true`), при включении – приемопередатчик включен (в метод передается `false`). По умолчанию, `switchSend` установлен в положение `On`, `flagSend` имеет состояние `true`, приемопередатчик выключен до тех пор, пока пользователь не выберет файлы и не нажмет кнопку «Отправить». После отправки файлов либо сбоя при отправке устройство приемопередатчика будет автоматически отключено.

При клике по `nameDeviceText` - надписи имени устройства, полученного из файла «`dataUser.txt`», пользователь увидит всплывающее сообщение с ключом безопасности, который он задал, что сделано для удобства при согласовании ключей для передачи или приема файлов.

На экране для передачи данных имеется 4 кнопки: «Добавить файл», «Убрать файл», «Отправить», «В меню». При нажатии кнопки «В меню» вызывается метод возврата в меню - `sendMain (view: View)`.

При нажатии на кнопку «Добавить файл» будет вызван метод `addFile(view: View)`. Он открывает файловый менеджер устройства, чтобы пользователь мог добавить файлы для передачи. Если файловый менеджер не установлен, то будет выведено всплывающее сообщение, информирующее об этом. Результат выборки придет в переопределенный метод `onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)` в параметр `data`; `resultCode` указывает успешно ли прошел вызов, `requestCode` – идентификатор для указания с какого активити пришел результат.

Метод `onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)` служит для подготовки выбранных файлов к отправке:

- 1 шаг - получение uri выбранного файла и его запись в глобальный массив uri arrayFiles;
- 2 шаг - с помощью доступа к данным из поставщика контента через клиентский объект contentResolver получение байтового представления файла, формата файла и его имени;
- 3 шаг – формирование ключей и шифрование этими ключами имени файла, его формата и содержимого файла применением методов реализованного класса шифрования RC4;
- 4 шаг – формирование посылок имени, формата и содержимого файла для передачи методом packageCreate(dataUsers: ByteArray), в котором параметр dataUsers – данные для отправки в формате байтового массива.

Шаги 3 и 4 отражены во фрагменте кода в листинге 2.

Листинг 2 – Шифрование и отправка содержимого файла

```

Val keyForUnitsFile="LightFile".toByteArray()
val encoderFile:RC4=RC4(keyForUnitsFile)
val encoderResultFile= encoderFile.encode(fileUser, fileUser.size)
...
GlobalScope.launch() {
...
    packageCreate(encoderResultFile)
    splitPackage()
... }

```

Кроме того, если файл был отправлен успешно, то после выбора нового, информация об отправленном стирается, если же файл еще не отправлялся или имел проблемы при передаче – его uri и сформированные посылки данных остаются в глобальных массиве и списке для дальнейшей обработки и отправки.

RC4 шифрование передаваемых данных было реализовано по алгоритму из раздела 3.9. Для этого был создан класс RC4, с полями итераторов $x = 0$ и $y = 0$, и полем S ключевого потока, он имеет конструктор, который при создании объекта, проводит инициализацию ключевого потока S предназначенным для этого методом. В классе реализованы методы:

- initParam(key: ByteArray) – метод, который вызывается в конструкторе, данный метод сначала инициализирует ключевое поле значениями от 0 до

255, а затем, используя входной ключ `ByteArray` (различный для видов объектов отправки), выполняет перестановки в ключевом поле; применяемые начальные ключи для последующей генерации ключей шифрования: `"LightNamePassw"` – для пароля устройства, `"LightNameDevice"` – для имени устройства, `"LightName"` – для имени файла, `"LightFormat"` – для формата файла, `"LightFile"` – для содержимого файла);

- `keyRandom(): Int` - метод, который возвращает псевдослучайное значение из ключевого поля `S` для дальнейшего применения операции XOR к нему и значению из массива пользовательских данных;
- `encode(dataBinaryUsers: ByteArray, size: Int): ByteArray` – метод для шифрования данных, применяет к каждому байту данных пользователя и псевдослучайному значению ключа `S`, преобразованного в байты, операцию XOR, возвращает массив зашифрованных данных, код метода приведен на листинге 3.

Листинг 3 – Метод шифрования данных

```
fun encode(dataBinaryUsers: ByteArray, size: Int): ByteArray {
    var dataUsers = dataBinaryUsers
    var chiperData = ByteArray (dataUsers.size, {0})
    for (n in 0 until size) {
        chiperData[n] = (dataUsers[n] xor keyRandom().toByte())
    }
    return chiperData }
}
```

После шифрования файлов, следует этап формирования посылок. Посылки формируются в методе `packageCreate(dataUsers: ByteArray)`, который был реализован по алгоритму из раздела 3.9, представленному на рисунке 26. Чтобы сформировать посылку, необходимо работать с битами. Для работы с битами был использован класс `BitSet`. После преобразования байтового массива пользовательских данных `dataUsers` в битовый формат, происходит формирование посылок для дальнейшей передачи: добавление старт-, стоп-битов, CRC и дополнительных битов. Каждая посылка имеет 256 бит. После составления очередной посылки данных, она записывается в глобальный список битовых

массивов ListArraySend. Фрагмент кода для формирования посылок приведен на листинге 4.

Листинг 4 – Фрагмент формирования посылок данных (без CRC)

```
private suspend fun packageCreate(dataUsers: ByteArray) {
...
do { if (size <= 1) predel = dataUsers.size * 8
    else predel = 109 * countIndex+outCircle
    for (n in (110 * outCircle)..(109 * countIndex+outCircle)) {
        if (n == 110 * outCircle) {
            sendPackTemp[count] = true
            count++
            sendPackTemp[count] = false
            count++ }
        if (n > predel) {
            sendPackTemp[count] = false
            count++
            sendPackTemp[count] = true
            count++
        } else {
            sendPackTemp[count] = sendPack[n]
            count++
            if (sendPack[n] == true) sendPackTemp[count] = false
            else sendPackTemp[count] = true
            count++ }
        if (n == 109 * countIndex+outCircle) {
            sendPackTemp[count] = false
            count++
            sendPackTemp[count] = true } }
... } while (outCircle <= size-1) }
```

Формирование CRC реализовано в методе `crcPack (packData:BitSet): BitSet`, который возвращает в битовом формате рассчитанную контрольную сумму. Аргумент `packData` – посылка данных со старт-, стоп-, основными и дополнительными битами. Метод базируется на алгоритме, описанном в разделе 3.9. Рассчитанная CRC вставляется перед стоп-битом и также дополняется битами для ООК RZ.

Для разграничения объектов передачи используются специальные нулевые посылки, создаваемые методом `splitPackage()`. Одна такая посылка представляет из себя битовую посылку из 256 0, которая заносится в глобальный список битовых массивов ListArraySend. Таким образом, на стороне приема можно, например, идентифицировать, когда закончится передача ключа безопасности, и начнется передача имени устройства.

Все продолжительные операции в приложении такие, как формирование посылок, подготовка файлов к отправке и их отправка, реализованы в отдельных потоках для того, чтобы избежать блокирования пользовательского интерфейса во время их выполнения. Для организации многопоточности был использован механизм сопрограмм (корутин). Основное их преимущество - легковесность по сравнению с потоками Thread, кроме того, логика программы визуально выражена последовательно [72]. Методы в проекте, которые работают в фоновом режиме, обозначены модификатором `suspend`: `packageCreate(dataUsers: ByteArray)`, `sendInformation()`, `splitPackage()`. Создание очередной сопрограммы происходит с помощью билдера `launch()` с диспетчером по умолчанию (см. листинг 2). Когда возникает необходимость изменить элемент экрана из корутины, например, статус передачи, происходит кратковременное переключение контекста, в котором работает сопрограмма, на главный контекст (Main) через диспетчер.

При нажатии на кнопку «Убрать файл» можно удалить из списка для отправки последний добавленный файл. Для этого будет вызван метод `removeFile(view: View?)`. С его помощью происходит актуализация информации об отправляемых файлах на экране, очищаются данные, относящиеся к удаляемому файлу, из `ListArraySend` - глобального массива пользовательских посылок, используя вспомогательный метод `removeListSend()`.

После нажатия на кнопку «Отправить» начинается передача выбранных файлов. Реализация данного функционала базируется на алгоритме, приведенном в разделе 3.9 на рисунке 27. Для передачи вызывается метод `sendData(view: View?)`, в котором запускается в сопрограмме метод `sendInformation()`, он начинает преобразование битовых данных в единый байтовый массив, который после подготовки данных, получения аудиофокуса и включения приемопередатчика, поступит на ЦАП мобильного устройства, откуда аналоговый сигнал будет подан на аудио-разъем jack 3.5. Поступление данных на ЦАП и после на аудиоразъем реализованы с помощью AudioTrack API, так как средства

разработки для Android не позволяют получить доступ к данной периферии напрямую, AudioTrack был выбран в учебных целях. Для использования ЦАП мобильного устройства используются: объект AudioTrack, настроенный для потоковой работы, метод проигрывания play() и метод записи в буфер write (audioData: ByteArray, offsetInBytes: Int, sizeInBytes: Int), из которого play() берет данные для передачи в ЦАП [73]. Описание параметров: audioData – данные для передачи в виде массива байт, offsetInBytes – количество байт, на которое нужно выполнить смещение данных для передачи, sizeInBytes – размер буфера для записи. Размер буфера был определен методом AudioTrack getMinBufferSize на основании указанных настроек. В учебных целях указана частота 8000 Гц. Настройка буфера и объекта AudioTrack для дальнейшей передачи данных приведена на листинге 5.

Листинг 5 – Настройка размера буфера и объекта AudioTrack

```
var buffersize = AudioTrack.getMinBufferSize(
    8000, AudioFormat.CHANNEL_OUT_MONO,
    AudioFormat.ENCODING_PCM_16BIT)
var trackplayer = AudioTrack(
    AudioAttributes.Builder().setUsage(USAGE_MEDIA)
    .setContentTypes(AudioAttributes.CONTENT_TYPE_UNKNOWN)
    .setLegacyStreamType(STREAM_MUSIC)
    .build(),
    AudioFormat.Builder().setEncoding(ENCODING_PCM_16BIT)
    .setChannelMask(CHANNEL_OUT_MONO)
    .setSampleRate(8000)
    .build(),
    buffersize,
    AudioTrack.MODE_STREAM,
    AUDIO_SESSION_ID_GENERATE)
```

На каждом этапе работы с файлами отображается информация об их состоянии на экране с помощью метода statusPack(source:CharSequence, messageForUser:String, colorText:Int), аргументы метода: source – источник данных, который необходимо изменить, messageForUser – сообщение статуса, colorText – цвет текста сообщения. Для смены статуса происходит изменение свойства text у элемента TextView InfoAddFiles. Реализованы следующие статусы: «Пожалуйста, подождите» - при формировании посылок файлов, «Подготовка данных» - при преобразовании посылок в байтовый массив, «Идет передача» - при передаче,

«Отправлено» - после успешной отправки, «Прервано» - пользователь отменил передачу.

Чтобы отследить, что данные отправлены, на стороне мобильного устройства был реализован слушатель обновления позиции маркера, указывающего на буфер данных. Информация о длине буфера для данного слушателя указывается с помощью метода `AudioTrack setNotificationMarkerPosition(bufferSize)`. Как только достигнут маркер конца буфера, статус отправляемых файлов меняется на «Отправлено», подача питания на устройство приемопередатчика прекращается.

Разработанные методы приложения и их назначение сведены в таблицу 9.

Таблица 9 – Методы приложения

Класс	Метод	Назначение
MainActivity (исходный код и код разметки см. в приложении Г в листингах Г.3 и Г.4)	<code>override fun onCreate (savedInstanceState: Bundle?)</code>	Создание экрана меню
	<code>fun sendActivity (view: View)</code>	Открытие экрана режима передачи
	<code>fun settingActivity (view: View)</code>	Открытие настроек
	<code>fun spravkaActivity (view: View)</code>	Открытие справки
MainActivitySend (исходный код и код разметки экрана см. в приложении Г в листингах Г.1 и Г.2)	<code>override fun onCreate (savedInstanceState: Bundle?)</code>	Создание экрана режима передачи
	<code>private fun infoForUser (messageForUser:String, long:Int)</code>	Создание сообщений для пользователя
	<code>private suspend fun splitPackage ()</code>	Создание разделителей объектов передачи
	<code>private fun statusPack (source:CharSequence, messageForUser:String, colorText:Int)</code>	Создание статуса посылки

Продолжение таблицы 9

Класс	Метод	Назначение
MainActivitySend (исходный код и код разметки экрана см. в приложении Г в листингах Г.1 и Г.2)	fun addFile (view: View)	Добавление файла для передачи
	fun sendData (view: View?)	Запуск метода отправки в отдельном потоке
	private suspend fun packageCreate (dataUsers: ByteArray)	Формирование посылок данных
	private suspend fun sendInformation()	Отправка файлов
	override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)	Запуск шифрования выбранного файла и его элементов
	fun removeFile(view: View?)	Удаление из списка отправки последнего добавленного файла
	private fun removeListSend()	Удаление посылок удаляемого файла из списка пользовательских посылок для передачи
	override fun onMarkerReached(trackplayer: AudioTrack)	Оповещение о том, что файлы были отправлены, отключение питания приемопередатчика
	fun sendMain (view: View)	Возврат в меню

Продолжение таблицы 9

Класс	Метод	Назначение
RC4 (исходный код см. в приложении Г в листинге Г.3)	private fun initParam(key: ByteArray)	Генерация ключа шифрования
	private fun keyRandom(): Int	Получение случайного байта ключа шифрования
	fun encode(dataBinaryUsers: ByteArray, size: Int): ByteArray	Шифрование данных
- (исходный код см. в приложении Г в листинге Г.3)	fun crcPack (packData:BitSet): BitSet	Расчет CRC16
MainActivitySetting (исходный код и код разметки экрана см. в приложении Г в листингах Г.5 и Г.6)	override fun onCreate(savedInstanceState: Bundle?)	Создание экрана настроек
	fun mainWindowActivity (view: View)	Выход в меню
	fun saveDeviceSettings(view: View)	Сохранение настроек
MainActivitySpravka (исходный код и код разметки экрана см. в приложении Г в листингах Г.7 и Г.8)	override fun onCreate(savedInstanceState: Bundle?)	Создание экрана справки

Процесс работы мобильного приложения приведен на скриншотах, представленных на рисунках 33, 34. На рисунке 33 скриншоты обозначают следующие действия:

1. получение информации об устройстве по нажатию на имя устройства «User5»;
2. добавление файла для отправки через нажатие на «Добавить файл»;
3. прерывание передачи файла (нажатие кнопки «Отправить» и следом – отключение передачи данных и приемопередатчика).

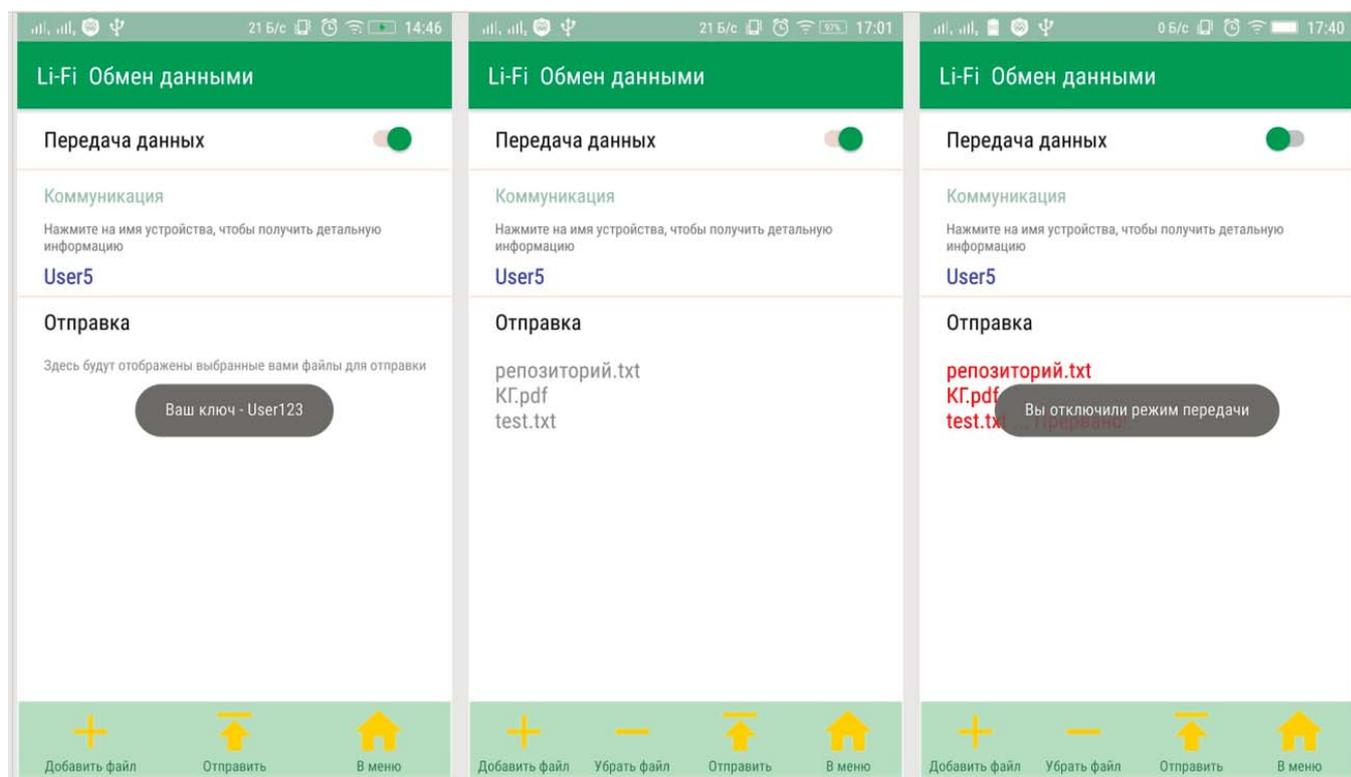


Рисунок 33 – Демонстрация мобильного приложения в работе (1)

На рисунке 34 скриншоты демонстрируют следующие действия:

1. отправка файла по нажатию «Отправить» - этап подготовки файла;
2. отправка файлов по нажатию «Отправить» - выбранные файлы успешно отправлены;
3. настройка имени и пароля – контроль приложением правильного ввода данных.

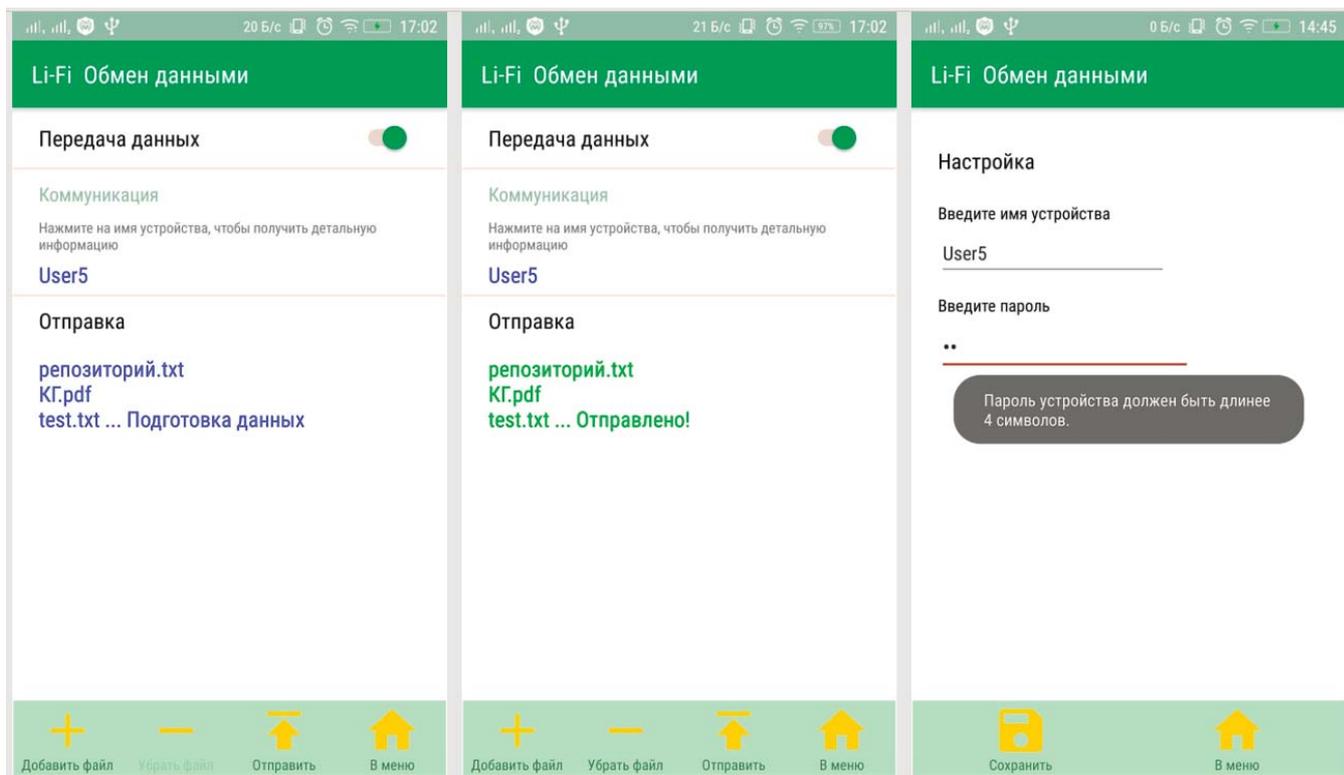


Рисунок 34 – Демонстрация мобильного приложения в работе (2)

Демонстрация работы шифрования данных приведена на рисунке 35. На нем приложение запущено в режиме отладки для того, чтобы отразить данные после шифрования имени файла «test2», а также после использования метода Encode к зашифрованному тексту для дешифрования, чтобы показать, что алгоритмы работают корректно.

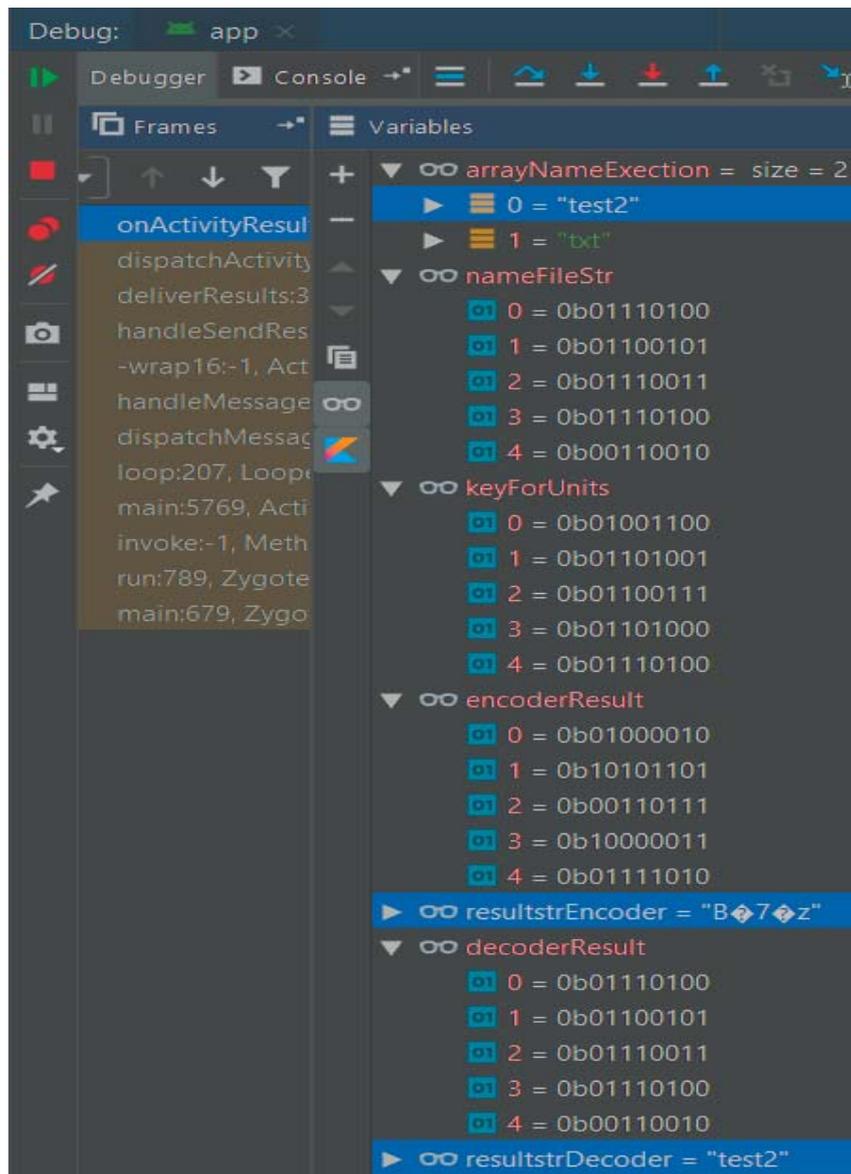


Рисунок 35 – Шифрование и дешифрование имени файла

4.2 УСТРОЙСТВО ПРИЕМОПЕРЕДАТЧИКА

Для программы управления приемопередатчиком были реализованы функции приема данных с аудиоразъема jack 3.5 и их модуляции, опираясь на алгоритм из раздела 3.9, приведенный на рисунке 28. Для приема создана функция void ADCMobileInit(), прием данных происходит через 2-ой канал АЦП. В функции приема происходит инициализация и настройка необходимой периферии: порта A2 для 2 канала АЦП, АЦП с использованием инжектированного канала, таймера 4 для считывания значения с АЦП с частотой,

аналогичной частоте поступления данных с мобильного устройства на аудиоразъем – 8 КГц. В обработчике прерывания таймера 4 происходит анализ полученного значения с АЦП и запись значения для передачи в буфер dataForSend.

Использование инжектированных каналов связано с тем, что требуется использовать АЦП не только в передающем модуле приемопередатчика, но и в принимающем, что позволяет сохранять данные каждого модуля в разных регистрах, в отличие от регулярных каналов, у которых лишь один регистр данных [75].

Для реализации модуляции написана функция void DAC_DMAInit(). В ней происходит инициализация и настройка необходимой периферии, необходимой для задачи модуляции. Основная периферия: ЦАП (1-ый канал), ПДП (13 канал), таймер 6, порт A4 – выход ЦАПа (к данному порту подключен светодиод). При обновлении счетного регистра таймера происходит запрос к ПДП, откуда берутся данные с указанного адреса буфера и передаются в ЦАП.

В основной функции программы void main() происходит вызов перечисленных выше функций. Список разработанных функций и обработчиков приведен в таблице 10.

Таблица 10 – Функции программы для МК

Файл	Функция/Обработчик прерывания	Назначение
adc_mobile.c (исходный код представлен в листинге Д.2 в приложении Д)	void ADCMobileInit ()	Инициализация и настройка периферии для приема данных с мобильного устройства
	void TIM4_IRQHandler ()	Запись значения для передачи в ЦАП в соответствии с полученными данными

Продолжение таблицы 10

Файл	Функция/Обработчик прерывания	Назначение
dac_dma.c (исходный код представлен в листинге Д.1 в приложении Д)	void DAC_DMAInit ()	Инициализация и настройка периферии для модуляции данных
main.c	void main ()	Вызов перечисленных выше функций, бесконечная работа программы

Для проверки работоспособности модуляции был передан символ «Л» на частоте 8 кГц, его бинарный код: 10100111. Результат того, что символ передается, демонстрирует снимок осциллографа, приведенный на рисунке 36. Реализация программы для передатчика аппаратной части приведена в листинге Д.

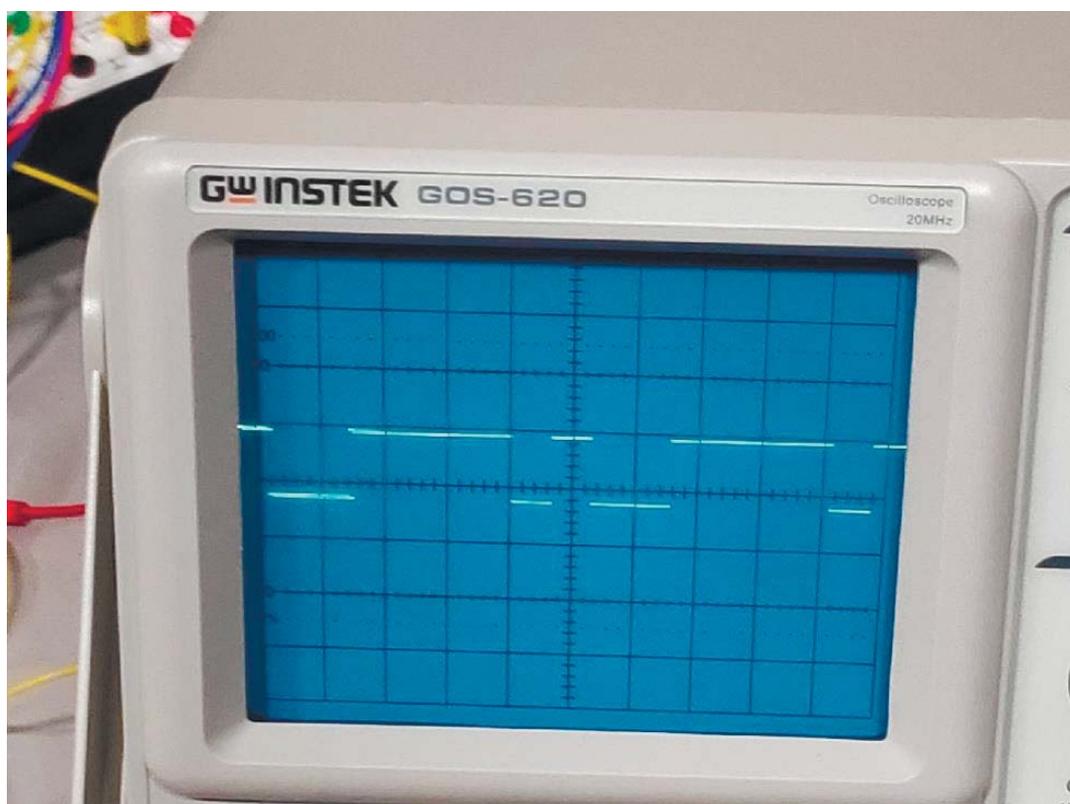


Рисунок 36 – Передача символа

В рамках работы были собраны учебные образцы устройств приемопередатчика на макетных платах. На рисунке 37 приведена фотография одного такого учебного образца.

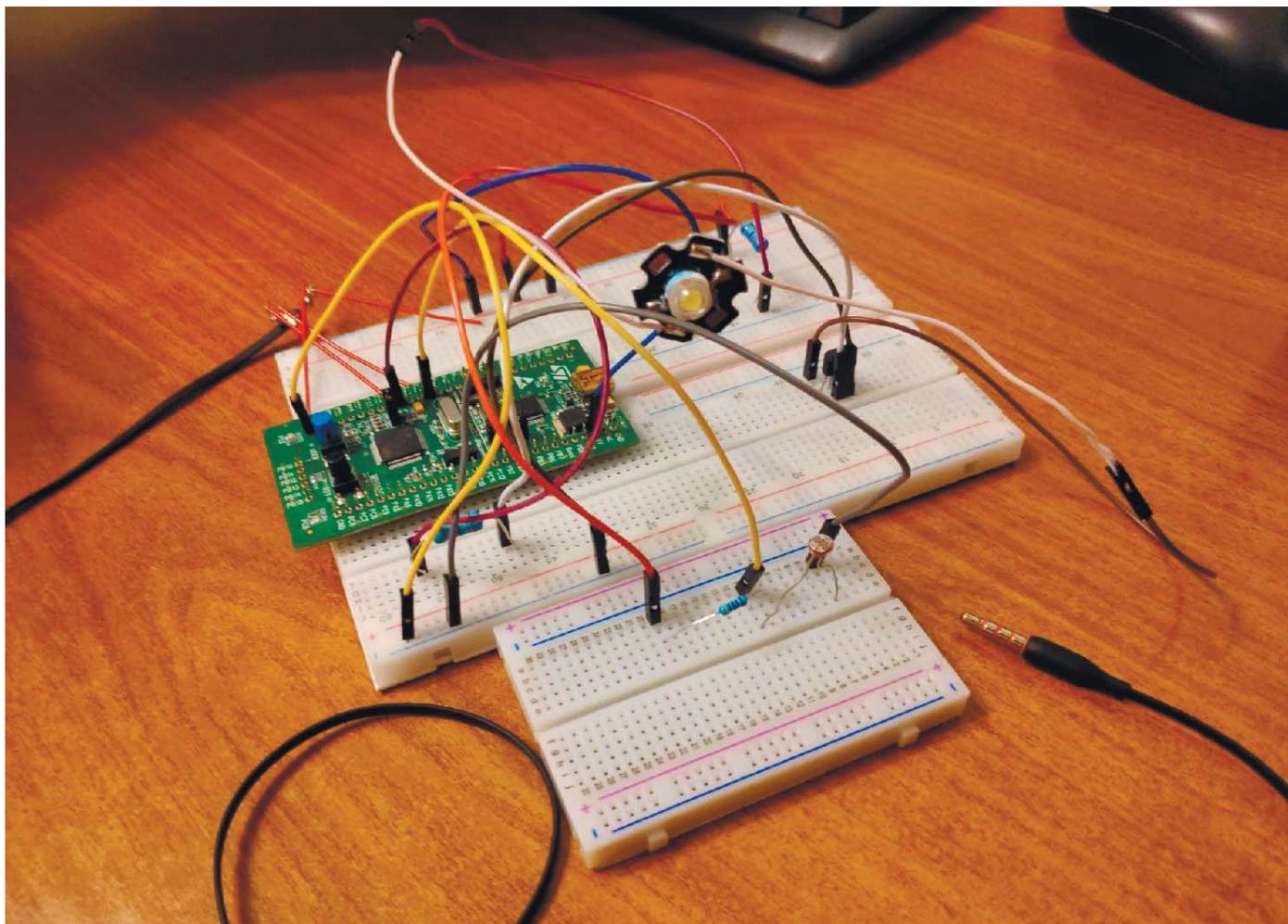


Рисунок 37 – Учебный образец приемопередатчика

5. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

Для проверки работоспособности прототипа передающего модуля программно-аппаратного комплекса системы связи Li-Fi было проведено функциональное тестирование. Тестирование – это один из способов контроля качества разрабатываемой ПО, проверка соответствия между реальным и ожидаемым поведением программы. Функциональное тестирование позволяет проверить способность ПО решать задачи, необходимые пользователям (функциональные требования) [74]. Результаты тестирования передающего модуля мобильного приложения и прошивки для STM32 сведены в таблицу 10.

Таблица 11 – Результаты функционального тестирования

Название теста	Входные данные	Ожидаемый результат	Полученный результат	Результат теста
Добавление файлов	Пользователь нажал кнопку «Добавить файл» в экране передачи	Открытие файлового менеджера	Открытие файлового менеджера	Тест пройден
	Пользователь выбрал файл в файловом менеджере	Появление имени и формата выбранного файла на экране режима передачи	Появление имени и формата выбранного файла на экране режима передачи	Тест пройден

Продолжение таблицы 11

Название теста	Входные данные	Ожидаемый результат	Полученный результат	Результат теста
Отправка файлов	Пользователь нажал кнопку «Отправить» в экране передачи, не выбрав файлы	Вывод сообщения, что нет файлов для отправки, отправка не начата	Всплывающее сообщение, что нет файлов для отправки, отправка не начата	Тест пройден
	Пользователь нажал кнопку «Отправить» в экране передачи, предварительно выбрав файлы	Отправка файлов	Отправка файлов, вывод сообщения о статусе передачи	Тест пройден
Получение данных об устройстве	Пользователь нажал на имя устройства в экране передачи	Получить информацию о ключе безопасности устройства	Всплывающее сообщение, показывающее ключ безопасности	Тест пройден
Прервать передачу	Пользователь перевел переключатель в состояние Off при передаче в экране передачи	Отмена передачи и информирование об этом пользователя	Отмена передачи, сообщение пользователю об отмене операции	Тест пройден

Продолжение таблицы 11

Название теста	Входные данные	Ожидаемый результат	Полученный результат	Результат теста
Задание имени и ключа	Пользователь в экране настроек вводит имя устройства и пароль и нажимает «Сохранить»	Введенные данные сохраняются	Введенные данные сохраняются с помощью записи в файл, всплывающее сообщение о сохранении	Тест пройден
	Пользователь в экране настроек вводит некорректное имя устройства или пароль и нажимает «Сохранить»	Вывод сообщения о неправильном вводе, новые данные не сохранены	Всплывающее сообщение о неправильном вводе без сохранения новых данных	Тест пройден
Получение справки	Пользователь нажал кнопку «Справка» в меню	Открытие экрана со справкой	Открытие экрана со справкой	Тест пройден

Как видно из таблицы 11 и скриншотов, приведенных в предыдущем разделе, передающий модуль прототипа системы связи Li-Fi для передачи данных между мобильными устройствами Android работает корректно.

6. ЗАКЛЮЧЕНИЕ

В представленной работе были разобраны основные темы, связанные с проектированием и разработкой системы связи Li-Fi для мобильных устройств Android. Обозначены задачи, которые необходимо выполнить для достижения цели. В рамках выполнения ВКР проанализированы: актуальность темы; похожие существующие решения, выявлены их недостатки; возможные варианты реализации, среди которых выбраны подходящие.

На основании недостатков найденных аналогов определены требования к прототипу системы. Кроме того, спроектированы: архитектура системы; схема электрическая принципиальная; выполнена трассировка печатной платы.

Подобраны компоненты для аппаратной части с учётом их стоимости. Разработаны основные алгоритмы, такие как: шифрование передаваемых файлов (алгоритм мобильного приложения); формирование посылок для отправки из выбранных файлов (алгоритм мобильного приложения); расчет CRC16 (алгоритм мобильного приложения); передача данных с мобильного устройства на приемопередатчик; прием данных с аудиоразъема jack 3.5 мобильного устройства приемопередатчиком; модуляция данных (для приемопередатчика).

Реализовано:

- мобильное приложение системы Li-Fi с основным функционалом для устройств на ОС Android 6.0 и выше (передающий модуль и общая часть);
- программа для МК приемопередатчика (для передающего модуля);
- собраны учебные образцы устройств приемопередатчика на макетных платах.

В будущем система может быть доработана до двунаправленной системы Li-Fi и для использования выхода в Интернет, что понизит расходы по сравнению с другими технологиями организации сети, а также обеспечит высокую скорость обмена данными.

Библиографический список

- 1 Идеи, достойные распространения. -
https://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb/.
Дата обращения: 03.02.2020.
- 2 Новости Университета ИТМО. -
<http://news.ifmo.ru/ru/science/photonics/news/6805/>. Дата обращения:
03.02.2020.
- 3 Elec.ru - Электротехнический интернет-портал. -
<https://www.elec.ru/articles/li-fi-svetodiodnye-kommunikacii-v-pogone-za-potreb/>. Дата обращения: 03.02.2020.
- 4 ЭМИЛИНК - проектирование, производство и дистрибуция телекоммуникационных и электротехнических продуктов и решений для ИТ и инженерной инфраструктуры зданий. -
https://emilink.ru/preimuschestva_i_nedostatki_volokonno_opticheskiy_kabelya/.
Дата обращения: 04.02.2020.
- 5 О витой паре. - http://netcab.narod.ru/vitpar_cab.html. Дата обращения:
04.02.2020.
- 6 Robohunter - сообщество робототехников. - <https://robohunter.com/news/perspektivnaya-slojnaya-i-dorogaya-vse-o-5g-samoi-progressivnoi-mobilnoi-svyazi16133/>. Дата обращения: 04.02.2020.
- 7 Khan, L. U. Visible light communication: Applications, architecture, standartization and research challenges / L. U. Khan // Digital Communications and Networks. - 2017. - V. 3, №2. P. 78-88.
- 8 Шнайдер, Г. Е. Использование технологии Li-Fi для передачи данных / Г. Е. Шнайдер // Материалы Международной научно-технической конференции "INTERMATIC". - М.: МИРЭА. - 2015. - С. 1-4.
- 9 О Bluetooth. - <https://sites.google.com/site/zarabay/cover-page/preimusestva-i-nedostatki-setej-bluetooth/>. Дата обращения: 06.02.2020.

- 10 Социальный интернет-сервис Pinterest. - <https://www.pinterest.ru/pin/598204762984707260/>. Дата обращения: 07.02.2020.
- 11 Сайт компании OledComm. - <https://www.oledcomm.net/>. Дата обращения: 07.02.2020.
- 12 О mesh-сети на основе света от компании Velmenni. - <https://glazdik.ru/novosti/unikalnaya-texnologiya-peredachi-dannyx-cherez-svet-li-fi-est-li-u-etoj-texnologii-budushhee.html>. Дата обращения: 07.02.2020.
- 13 Сайт компании PureLiFi. - <https://purelifi.com/>. Дата обращения: 07.02.2020.
- 14 Сайт компании Signify. - <https://www.signify.com/ru-ru/our-company/news/press-release-archive/2018/20180430-philips-lighting-introduces-lifi>. Дата обращения: 08.02.2020.
- 15 Сайт компании Lucibel. - <http://www.lucibel.io/web/guest/communication>. Дата обращения: 08.02.2020.
- 16 Решение LiFiMAX от OledCom. - <https://www.oledcomm.net/lifi-max/>. Дата обращения: 08.02.2020.
- 17 Сайт о LiFi. - <https://lifi.co/news/oledcomm-launches-lifimax/>. Дата обращения: 08.02.2020.
- 18 Интернет-журнал «Системный администратор». - <http://samag.ru/archive/article/1731/>. Дата обращения: 09.02.2020.
- 19 Сайт компании Velmenni. - <https://www.velmenni.com/about-lifi.php>. Дата обращения: 09.02.2020.
- 20 Университет ИТМО - факультет лазерной фотоники и оптоэлектроники. - <http://lfo.itmo.ru/li-fi>. Дата обращения: 09.02.2020.
- 21 Новости о технике. - <https://ww.9to5mac.com/2019/06/20/truelifi/#>. Дата обращения: 09.02.2020.
- 22 Хабр - сообщество IT-специалистов. - <https://habr.com/ru/post/196562/>. Дата обращения: 11.02.2020.

- 23 ELECTRIC INFO - онлайн журнал про электричество. - <http://electrik.info/main/praktika/1441-shim-kontroller-vidy-i-shemy.html>. Дата обращения: 11.02.2020.
- 24 Рынок Микроэлектроники. - <http://www.gaw.ru/html.cgi/txt/doc/dac/dacsindex.htm>. Дата обращения: 11.02.2020.
- 25 РобоКраф - сайт о робототехнике. - <http://robocraft.ru/blog/arduino/34.html>. Дата обращения: 12.02.2020.
- 26 Haas, H. LiFi is a paradigm-shifting 5G technology / H. Haas // Reviews in Physics. - 2018. V. 3. P. 26-31.
- 27 Использование лазеров на рабочем месте: практическое руководство / МОТ, Международная Ассоциация по радиационной защите, Международный Комитет по неионизирующему излучению. – Женева: Международное бюро труда, 1993. - 68 с.
- 28 ALLED.RU - светодиодный магазин. - <http://alled.ru/led-svetodiod.html?sl=RU>. Дата обращения: 13.02.2020.
- 29 ULTRAHD.SU - информационно-аналитический проект. - <http://ultrahd.su/video/microled-oled-otlichiya.html>. Дата обращения: 13.02.2020.
- 30 INCINE - сайт компаний по техподдержке и продаже оборудования из научно – исследовательского кинофотоинститута. - <http://wr.incine.ru/?p=1453>. Дата обращения: 13.02.2020.
- 31 Ходоков, А. А. Методы модуляции технологии LI-FI. / А. А. Ходоков. - Минск, Республика Беларусь: Белорусская государственная академия связи, 2014. - 7 с.
- 32 Авдеев, В. А. Периферийные устройства: интерфейсы, схемотехника, программирование. / В. А. Авдеев. - М.: ДМК Пресс, 2009. - 848 с.

- 33 Jain, Adarsh. Demonstration of RZ-OOK modulation scheme for high speed optical data transmission. / Adarsh Jain, K. R. Bahl, Alak Banik // Eleventh International Conference on Wireless and Optical Communications Networks. - Vijayawada, India. - 2014. – P. 1-6.
- 34 ResearchGate - профессиональная сеть для ученых и исследователей. - https://www.researchgate.net/figure/Waveform-of-OOK-NRZ-and-OOK-RZ_fig1_271417252. Дата обращения: 13.02.2020.
- 35 Developers - для разработчиков Android. - <https://developer.android.com/about/dashboards/index.html>. Дата обращения: 13.02.2020.
- 36 Кочегаров, И. И. Микроконтроллеры AVR. Лабораторный практикум: уч. пособие. / И. И. Кочегаров, В. А. Трусов. - Пенза: Изд-во ПГУ, 2012. - 122 с.
- 37 EASY ELECTRONICS - электроника для всех. - <http://we.easyelectronics.ru/STM32/dac-v-stm32.html>. Дата обращения: 14.02.2020.
- 38 8-bit AVR Microcontroller with 1K Bytes In-System Programmable Flash: ATtiny13. (Datasheet). - <https://pdf1.alldatasheet.com/datasheet-pdf/view/80274/ATMEL/ATTINY13.html>. Дата обращения: 15.02.2020.
- 39 8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash: ATmega48/V, ATmega88/V, ATmega168/V. (Datasheet). - http://radio-point.narod.ru/datasheet/a1/atmega48_88_168.pdf. Дата обращения: 15.02.2020.
- 40 SAM E70 Atmel. (Datasheet). - http://catalog.gaw.ru/index.php?page=component_detail&id=78844. Дата обращения: 15.02.2020.
- 41 Microchip PIC16C71X 8-Bit CMOS Microcontrollers with A/D. (Datasheet). - <http://www.microchip.ru/cdrom/ww1.microchip.com/downloads/en/DeviceDoc/30272a.pdf>. Дата обращения: 15.02.2020.

- 42 Infineon XMC1100 AB-Step Microcontroller Series for Industrial Application. (Datasheet). - https://www.infineon.com/dgdl/Infineon-xmc1100_AB-DS-v01_08_EN.pdf?fileId=5546d4624a0bf290014a4bdaff9325bd. Дата обращения: 15.02.2020.
- 43 Infineon XMC4100/4200 Series for Industrial Application. (Datasheet). - <https://www.compel.ru/item-pdf/d1fb9679fc8205a37b24c35b6afe3363/ps/infin~xmc4100.pdf>. Дата обращения: 15.02.2020.
- 44 STM32F100x4, STM32F100x6, STM32F100x8, STM32F100xB. (Datasheet). - <https://static.chipdip.ru/lib/820/DOC000820500.pdf>. Дата обращения: 15.02.2020.
- 45 EFM32 Giant Gecko Family EFM32GG. (Datasheet). - <https://www.silabs.com/documents/public/data-sheets/efm32gg-datasheet.pdf>. Дата обращения: 15.02.2020.
- 46 ЧИП и ДИП - магазин электронных компонентов. - <https://www.chipdip.ru>. Дата обращения: 16.02.2020.
- 47 Бэйкер, Б. Что нужно знать цифровому инженеру об аналоговой электронике. / Б. Бэйкер. - М.: Додэка-XXI, 2010. - 360 с.
- 48 Mouser Electronics - дистрибьютором полупроводников и электронных компонентов. - <https://ru.mouser.com/ProductDetail/STMicroelectronics/STLQ50C50R?qs=CUsY4352MFhoWHVppG9%2F8w%3D%3D>. Дата обращения: 17.02.2020.
- 49 Кузнецов, М. И. Основы электротехники. Уч. пособие. / М. И. Кузнецов. - М.: Изд-во «Высшая школа», 1970. - 368 с.
- 50 Терра электроника - web-портал для разработчиков электроники. - <https://www.terraelectronica.ru/news/5188>. Дата обращения: 18.02.2020.

- 51 КОМПЭЛ - поставка компонентов и модулей для производителей электронной аппаратуры. - <https://www.compel.ru/lib/53954>. Дата обращения: 18.02.2020.
- 52 Radio-sale - Интернет-магазин радиодеталей и комплектующих. - <https://radio-sale.ru>. Дата обращения: 18.02.2020.
- 53 CHIPINFO - справочник по микросхемам и ежедневная лента новостей электроники. - https://chipinfo.pro/elements/acoustics/jack_pinout.shtml. Дата обращения: 19.02.2020.
- 54 Чернышев, А. Классы точности печатных плат и современные технологии / А. Чернышев // Печатный монтаж. - 2009. - Вып. 6. - С. 19-21.
- 55 Rexant - поставщик кабельной продукции и расходных материалов и для монтажа слаботочных систем. - <https://rexant-shop.ru/>. Дата обращения: 19.02.2020.
- 56 Блог, посвященный робототехнике. - <http://robotosha.ru/stm32/free-ide-stm32.html>. Дата обращения: 20.02.2020.
- 57 Tproger - издание о разработке и обо всём, что с ней связано. - <https://tproger.ru/translations/kotlin-vs-java-android/>. Дата обращения: 20.02.2020.
- 58 Фергюсон, Н. Практическая криптография. / Нильс Фергюсон, Брюс Шнайер; пер. с англ. - М.: Издательский дом «Вильямс», 2005. - 432 с.
- 59 Anusuya, S. Jeya. Data Transmission by Ceaser Cipher Wheel encryption using Lifi. / S. Jeya Anusuya // International Journal of Advance Research, Ideas and Innovations in Technology. - 2018. - V. 4, № 2. - P. 512-517.
- 60 Klein, A. Attacks on the RC4 stream cipher. / A. Klein // Designs, codes and cryptography. - 2008. - V. 48, № 3. - P. 269-286.
- 61 Сайт о криптографии. - <http://kriptografea.narod.ru/XOR.html>. Дата обращения: 21.02.2020.

- 62 Оптоволоконные линии связи. - <http://www.on-lan.ru/sety/C/2.1.2.htm>. Дата обращения: 22.02.2020.
- 63 Li-Fi: Будущее интернета. - <https://habr.com/ru/post/435262/>. Дата обращения: 22.02.2020.
- 64 Наумкина, Л. Г. Электроника: уч. пособие. / Л. Г. Наумкина. – М.: Горная книга, 2007. - 326 с.
- 65 АЗИМУТ ФОТОНИКС - поставка оптоэлектронных компонентов ведущих мировых производителей. - http://www.azimp.ru/catalogue/mini_lasers/12071/. Дата обращения: 22.02.2020.
- 66 Петрусь, И. П. Современные методы построения коммуникационных линий в видимой части спектра электромагнитных волн / И. П. Петрусь // Международная конференция студентов, аспирантов и молодых ученых "Информационные технологии, телекоммуникации и системы управления": сборник докладов. – Екатеринбург: [УрФУ]. - 2015. - С. 212-216.
- 67 CRC-алгоритмы обнаружения ошибок. - http://embedded.ifmo.ru/embedded_old/ETC/REFERAT/crc/crc.htm. Дата обращения: 23.02.2020.
- 68 Лопанов, А.Н. Мониторинг и экспертиза безопасности жизнедеятельности: уч. пособие / А. Н. Лопанов, Е.В. Климова. - Белгород: Изд-во БГТУ, 2009. - 201 с.
- 69 Мощность. - <https://ru.wikipedia.org/wiki/Мощность>. Дата обращения: 27.02.2020.
- 70 Электрический конденсатор. - https://ru.wikipedia.org/wiki/Электрический_конденсатор. Дата обращения: 27.02.2020.
- 71 Блог об Ардуино электронике и ИТ. - <https://soltau.ru/index.php/themes/dev/item/461-kak-poschitat-kontrolnuyu-summu-crc32-crc16-crc8>. Дата обращения: 22.04.2020.

- 72 Сопрограммы. Kotlin. документация. -
<https://kotlinlang.ru/docs/reference/coroutines.html>. Дата обращения:
22.04.2020.
- 73 О передаче данных через аудиоразъем. - <https://software.intel.com/ru-ru/android/articles/using-the-audio-jack-as-data-interface-on-android-systems>.
Дата обращения: 22.04.2020.
- 74 Unetway - сайт для развития сотрудничества между компаниями и IT-специалистами. - <http://unetway.com/tutorial/funkcionalnoe-testirovanie/>. Дата обращения: 27.04.2020.
- 75 My Controller - изучение микроконтроллеров STM32. -
http://mycontroller.ru/old_site/stm32-adc-primeryi-ispolzovaniya-inzhektirovannyie-kanalyi/default.htm. Дата обращения: 29.04.2020.