МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Политехнический институт Факультет приборостроения, информационных технологий и электроники	Кафедра «Информационная безопасность систем и технологий»
Специальность 10.05.03 —	Информационная безопасность автоматизированных систем
Специализация —	Защищённые автоматизированные системы управления

ДИПЛОМНЫЙ ПРОЕКТ

на тему

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МАКЕТА МОБИЛЬНОЙ АППАРАТУРЫ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ РЕЧЕВОЙ ИНФОРМАЦИИ

Студент Тризна Григорий Дмитриевич

Руководитель Клейменов А. А.

Нормоконтролёр Иванов А. П.

Рецензент начальник отдела АО «ПНИЭИ» Оськин Р. С.

Работа допущена к защите (протокол заседания кафедры от 21.06.19 № 16)

Заведующий кафедрой _____ С. Л. Зефиров

Работа защищена с отметкой отлично (протокол заседания ГЭК от 25.06.19 № 26)

Секретарь ГЭК _____ А. П. Иванов

Проверил ВКР Тризна
С. Л. Зефиров

Пенза 2019

АКЦИОНЕРНОЕ ОБЩЕСТВО «Научно-исследовательский электронный институт» АО «ПНИЭИ» ОТДЕЛ КАДРОВ

Подпись Клейменова А.А. Оськина Р.С. удостоверяю

Инженер 1 категории Т.Т. Стальнова

Оглавление

Введение.……4

1 Обзор требований к программному обеспечению макета …….6

2 Анализ существующих алгоритмов кодирования/декодирования речевой информации….……...……..8

3 Анализ существующих механизмов передачи речевой информации 11

4 Анализ существующих протоколов передачи речевой информации в сетях IP (3G/4G) …….…...13

4.1 SIP…….13

4.2 SDP……....19

4.3 RTP……21

4.4 ZRTP ……..……...24

5 Анализ отечественных криптографических алгоритмов …….26

5.1 ГОСТ 218147-89 ...……..26

5.2 ГОСТ Р 34.12-2015 …….27

6 Анализ возможных способов ввода ключевой информации в мобильной аппаратуре……..29

7 Анализ структуры макета с определением трактов прохождения речевой информации ……...32

8 Разработка алгоритма работы программного обеспечения макета…….34

9 Разработка программного обеспечения макета ……44

9.1 Описание условий разработки ……..44

9.2 Разработка модуля считывания и воспроизведения речевой информации ……...44

9.3 Разработка модуля кодирования и раскодирования речевой информации ……...47

9.4 Разработка модуля зашифрования и расшифрования закодированной речевой информации ……49

9.5 Разработка модуля приёма/передачи зашифрованной речевой информации ……...53
9.6 Разработка модуля, реализующего пользовательский интерфейс макета …….57
9.7 Интеграция разработанных модулей ……64
10 Тестирование программного обеспечения макета …….67
10.1 Функциональное тестирование ……...67
10.2 Измерение разборчивости воспроизводимой речи артикуляционным методом ……..68
11 Вредные психофизиологические факторы, влияющие на опорнодвигательный аппарат специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие ……..72
Заключение……...74
Список использованной литературы…….75
Приложение А Промежуточные результаты измерения разборчивости речи артикуляционным методом …….79

Введение

На сегодняшний день идёт активное развитие информационных технологий, в том числе технологий передачи речевой информации, которые широко реализуются в виде мобильной аппаратуры

Эти технологии нашли почти повсеместное применение как в промышленности и бизнесе, так и в личной жизни человека. Они позволили вывести коммуникацию людей на новый уровень, но также и создают новые возможности для совершения несанкционированного доступа к конфиденциальной информации.

Поэтому возникает необходимость во введении дополнительных механизмов защиты информации. Один из самых очевидных вариантов — использование криптографических методов защиты информации.

Также необходимо обеспечить защиту от угроз, которые могут быть реализованы, путём применения недекларированных возможностей программного и аппаратного обеспечения устройств передачи информации. Эта проблема может быть решена путём анализа готового устройства на наличие недекларированных возможностей или же разработкой собственного устройства.

Цель выпускной квалификационной работы в целом — разработка программного обеспечения макета мобильной аппаратуры криптографической защиты речевой информации. Подобный макет — первый этап разработки мобильной аппаратуры передачи речевой информации, защищённой от обозначенных выше угроз.

В данном отчёте приведено обоснование выбора технических решений программного обеспечения макета, таких как выбор кодека,механизма и протоколов передачи речевой информации, криптографического алгоритма и способа ввода ключевой информации, описана структура макета также

приведено описание и результаты основных этапов разработки и тестирования программного обеспечения.

1 Обзор требований к программному обеспечению

Согласно техническому заданию на выпускную квалификационную работу программное обеспечение макета мобильной аппаратуры криптографической защиты речевой информации должно соответствовать следующим требованиям:

— программное обеспечение должно функционировать в среде макета мобильной аппаратуры криптографической защиты речевой информации, реализованного на основе отладочной платы STM3240G-EVAL, к которой подключены аудио гарнитура для ввода/вывода речевой информации и USB-модем для работы в сетях 3G/4G;

— в программном обеспечении должен выполняться ввод ключевой информации с внешнего носителя в форм-факторе micro SD-карты;

— программное обеспечение должно реализовывать считывание с микрофона и воспроизведение оцифрованной речевой информации на подключенной к макету аудио гарнитуре;

— в программном обеспечении должно реализовываться кодирование/декодирование оцифрованной речевой информации по алгоритму G.711 (PCMA) или Speex;

— программное обеспечение должно реализовывать шифрование и расшифрование закодированной речевой информации по алгоритму ГОСТ Р 34.12-2015 («Магма») в режиме гаммирования согласно ГОСТ Р 34.13-2015;

— программное обеспечение должно реализовывать расчет и контроль значения имитовставки на закодированную речевую информацию по алгоритму ГОСТ Р 34.13-2015 в режиме выработки имитовставки;

— программное обеспечение должно реализовывать прием/передачу зашифрованной речевой информации по сетям связи 3G/4G через подключенный к макету USB-модем.

— программное обеспечение должно реализовывать управление соединением (установка, разрыв) и обмен зашифрованной речевой информацией с аналогичным макетом по протоколам технологии IP-телефонии (SIP, RTP);

— в программном обеспечении должна быть предусмотрена возможность набора номера аналогичного макета для его вызова;

— программное обеспечение должно реализовывать прием входящих вызовов на установку соединения от аналогичного макета.

Таким образом в процессе разработки программного обеспечения были разработаны следующие модули:

— модуль считывания звука с аналого-цифрового преобразователя отладочной платы, поступившего с микрофона аудио гарнитуры, и воспроизведения звука на подключенной к плате аудио гарнитуре;

— модуль кодирования и декодирования звуковой информации;

— модуль зашифрования и расшифрования закодированной звуковой информации и вычисления и проверки имитовставки;

— модуль приёма/передачи информации по сетям связи 3G/4G через подключенный к макету USB-модем;

— модуль, реализующий возможности SIP-клиента по управлению соединением и обменом зашифрованной речевой информацией с аналогичным макетом по протоколам технологии IP-телефонии, включающий в себя реализацию протоколов стека TCP/IP и реализацию протоколов IP-телефонии — SIP и RTP;

— модуль, реализующий пользовательский интерфейс средствами отладочной платы.

2 Анализ существующих алгоритмов кодирования/декодирования речевой информации

При разработке системы передачи речевой информации первоначально был определён выполнен анализ алгоритмов сжатия и хранения этой информации и выбран оптимальный, соответствующий условиям работы мобильной аппратуры.

Аудио кодек — компьютерная программа или аппаратное средство, предназначенное для кодирования или декодирования аудиоданных. [1]

Наиболее распространены и активно используются следующие аудио кодеки:

— MPEG-1 Layer III (MP3);
— Advanced Audio Codec (AAC);
— Ogg Vorbis (OGG);
— Free Lossless Audio Codec (FLAC);
— GSM-FR;
— Adaptive multi rate (AMR);
— G.723.1;
— G.729;
— Internet Low Bitrate Codec (iLBC);
— SPEEX;
— G.711.

В таблице 1 приведены основные характеристики рассмотренных кодеков.

Таблица 1 — Основные характеристики аудио кодеков

Кодек	Скорость передачи (кбит/с)	Частоты дискретизации (кГц)	Число аудио каналов	Принцип работы кодека	Открытость формата
MP3 [2]	56-64	32, 44.1, 48	До 2	Психоакустическая модель кодирования [3]	Открытый
AAC [4]	От 8 до 160	От 8 до 96	До 48	Психоакустическая модель кодирования	Лицензируемый
OGG [5]	До 700	От 2 до 192	До 255	Модифицированное дискретное косинус-преобразование	Открытый
FLAC [6]	1411,2	От 0,001 до 655,35	До 8	Сжатие без потерь	Открытый
GSM-FR [7]	13	8	1	Долговременное линейное предсказание с регулярным импульсным возбуждением	Открытый
GSM-EFR [7]	12,2	8	1	Алгебраический код возбуждённого линейного предсказывания	Открытый
AMR [8]	От 1,8 до 12,2	8	1	Алгебраический код возбуждённого линейного предсказывания	Лицензируемый
G.723.1 [9]	5,3 или 6,3 [10]	8	1	Алгебраический код возбуждённого линейного предсказывания	Лицензируемый
G.729 [11]	8	8	1	Алгебраический код возбуждённого линейного предсказывания	Лицензируемый
ILBC [11]	13,33 или 15,2	8 или 16	1	Кодирование с линейным предсказанием	Открытый
SPEEX [12]	От 2 до 44	8, 16 или 32	2	Линейное предсказание с мультикодовым управлением	Открытый
G.711 [13]	64	8	-	Импульсно-кодовая модуляция	Открытый

Кодек SPEEX специально адаптирован для применения в IP-телефонии, позволяет довольно гибко настроить скорость передачи данных и поддерживает основные варианты частоты дискретизации, кроме того его использование не требует лицензирования. Поэтому данный кодек был использован в разработанном программном обеспечении макета.

Кодек G.711 также используется в программном обеспечении макета так как в нём не производится дополнительное сжатие, следовательно его программная реализация не требовательна к возможностям аппаратного обеспечения, по этой же причине обеспечивается высокое качество передаваемой речи.

3 Анализ существующих механизмов передачи речевой информации

После определения способа кодирования речевой информации был выполнен анализ существующих способов передачи речевой информации и выбор оптимального.

В настоящий момент реализация передачи речевой информации возможна с помощью двух механизмов: GSM и IP.

GSM (Global System Mobile) — глобальный стандарт цифровой мобильной связи, с разделением каналов по времени TDMA и частоте FDMA, разработанный под эгидой Европейского института стандартизации электросвязи (ETSI) . [14]

Стандарт поддерживает передачу речи, передачу данных GPRS, передачу коротких сообщений SMS и передачу факсимильных сообщений.

Сеть GSM состоит из 2 подсистем:

— коммутационной подсистемы;

— подсистемы базовых станций.

Коммутационная подсистема выполняет задачи по коммутации вызовов абонентов различных сетей, аутентификации абонентов и шифрования трафика. [21]

Подсистема базовых станций управляет всеми функциями, относящимися к работе радиоканалов в сети GSM. [14]

Для аутентификации абонентов и защиты передаваемого трафика в сетях GSM используются следующие алгоритмы [15]:

— A3 — алгоритм аутентификации;

— A8 — алгоритм генерации ключа шифрования;

— A5 — алгоритм шифрования.

A5 представляет собой семейство алгоритмов шифрования. Наиболее распространённым является A5/1. [15]

На сегодняшний день известно большое количество атак на данный алгоритм шифрования, в том числе с использованием радужных таблиц и корреляционные атаки. [15]

На данный момент применение иных алгоритмов шифрования в сетях GSM не предусмотрено [15].

Для передачи речевой информации с помощью IP используется стек протоколов TCP/IP и ряд, надстраиваемых над ним протоколов таких как SIP, SDP, RTP.

Для реализации сетей IP-телефонии не требуется дополнительной инфраструктуры, что значительно уменьшает стоимость связи. [16]

Возможно обеспечение связи между мобильными устройствами, между персональными компьютерами или между персональным компьютером и мобильным устройством с помощью так называемых IP-телефонов, которые могут быть реализованы как программно, так и аппаратно. [16]

Также возможна интеграция сетей IP-телефонии и сотовой связи, позволяющая обеспечить возможность взаимного обмена информацией между сетями и звонков между абонентами обеих сетей. [16]

В сетях IP-телефонии возможна реализация произвольных криптографических алгоритмов или иных методов защиты информации путём модификации сетевых протоколов.

Таким образом передача речевой информации посредством механизмов IP более предпочтительна в силу меньшей стоимости связи и возможности свободной реализации собственных механизмов защиты. Также это решение более универсально, так как может быть интегрировано в сети GSM.

После того, как был выбран механизм передачи речевой информации, были рассмотрены протоколы, по которым реализуется передача речевой информации с помощью выбранного механизма.

4 Анализ существующих протоколов передачи речевой информации

4.1 SIP

SIP (Session Initiation Protocol — протокол установления сеанса) — протокол передачи данных, описывающий способ установки завершения пользовательского интернет-сеанса, включающего обмен мультимедийным содержимым.

Протокол описывает, каким образом клиентское приложение (например, софтфон) может запросить начало соединения у другого, возможно, физически удалённого клиента, находящегося в той же сети, используя его уникальное имя. Протокол определяет способ согласования между клиентами об открытии каналов обмена на основе других протоколов, которые могут использоваться для непосредственной передачи информации (например, RTP). Допускается добавление или удаление таких каналов в течение установленного сеанса, а также подключение и отключение дополнительных клиентов (то есть допускается участие в обмене более двух сторон — конференц-связь). Протокол также определяет порядок завершения сеанса [17].

В основе разработки протокола лежали следующие принципы:

— простота: включает в себя только шесть методов (функций);

— независимость от транспортного уровня, может использовать UDP, TCP, ATM и т. д.

— персональная мобильность пользователей. Пользователи могут перемещаться в пределах сети без ограничений. Это достигается путём присвоения пользователю уникального идентификатора. При этом набор предоставляемых услуг остается неизменным. О своих перемещениях пользователь сообщает с помощью сообщения REGISTER своему серверу;

— масштабируемость сети. Структура сети на базе протокола SIP позволяет легко её расширять и увеличивать число элементов;

— расширяемость протокола. Протокол характеризуется возможностью дополнять его новыми функциями при появлении новых услуг;

— интеграция в стек существующих протоколов Интернет. Протокол SIP является частью глобальной архитектуры мультимедиа, разработанной комитетом IETF. Кроме SIP, эта архитектура включает в себя протоколы RSVP, RTP, RTSP, SDP.

— взаимодействие с другими протоколами сигнализации. Протокол SIP может быть использован совместно с другими протоколами IP-телефонии, протоколами телефонных сетей общего пользования, и для связи с интеллектуальными сетям.

Для организации взаимодействия с существующими приложениями IP-сетей и для обеспечения мобильности пользователей, SIP использует адрес, подобный адресу электронной почты. В качестве адресов рабочих станций используются универсальные указатели ресурсов URI, так называемые SIP URI. Обычно используется следующий формат [sip «:»] идентификатор [«@» фрагмент], где идентификатор указывает на логин абонента или его номер телефона, а фрагмент определяет хост, который может быть задан доменным именем или IP-адресом. Примеры:

— логин абонента@[Доменное имя];

— доменное имя устройства@[IP-адрес];

— № телефона@[VoIP-шлюз].

Общий стандарт URI определён рекомендацией RFC 3986.

Адрес состоит из двух частей. Первая часть — имя пользователя, зарегистрированного в домене или на рабочей станции. Во второй части адреса указывается имя домена сети, хоста или IP-адрес. Если вторая часть идентифицирует какой-либо шлюз, то в первой указывается телефонный номер абонента.

Протокол SIP имеет клиент-серверную архитектуру.

Клиент выдаёт запросы, с указанием того, что он хочет получить от сервера. Сервер принимает и обрабатывает запросы, выдаёт ответы, содержащие уведомление об успешности выполнения запроса, уведомление об ошибке или информацию, запрошенную клиентом.

Обслуживание вызова распределено между различными элементами сети SIP. Основным функциональным элементом, реализующим функции управления соединением, является абонентский терминал. Остальные элементы сети могут отвечать за маршрутизацию вызовов, а иногда служат для предоставления дополнительных сервисов.

Когда клиент и сервер реализованы в оконечном оборудовании и взаимодействуют непосредственно с пользователем, они называются пользовательским агентским клиентом — User Agent Client (UAC) — и пользовательским агентским сервером — User Agent Server (UAS). Если в устройстве присутствуют и UAC, и UAS, то оно называется пользовательским агентом — User Agent (UA), а по своей сути представляет собой терминальное оборудование SIP.

Сервер UAS и клиент UAC имеют возможность непосредственно взаимодействовать с пользователем. Другие клиенты и серверы SIP этого делать не могут.

Основные элементы сети, использующей SIP:

— абонентский терминал. Устройство, с помощью которого абонент управляет установлением и завершением звонков. Может быть реализован как аппаратно (SIP-телефон), так и программно (Софтфон);

— прокси-сервер. Устройство, которое принимает и обрабатывает запросы от терминалов, выполняя соответствующие этим запросам действия. Прокси-сервер состоит из клиентской и серверной частей, поэтому может принимать вызовы, инициировать запросы и возвращать ответы;

— сервер переадресации. Устройство, хранящее записи о текущем местоположении всех имеющихся в сети терминалах и прокси-серверах. Сервер переадресации не управляет вызовами и не генерирует собственные запросы;

— сервер определения местоположения пользователей. Представляет собой базу данных адресной информации. Необходим для обеспечения персональной мобильности пользователей.

Сообщения протокола SIP (запросы и ответы), представляют собой последовательности текстовых строк, закодированных в соответствии с документом RFC 2279. Структура и синтаксис сообщений SIP идентичны используемым в протоколе HTTP. Структура сообщений протокола SIP приведена на рисунке 1.

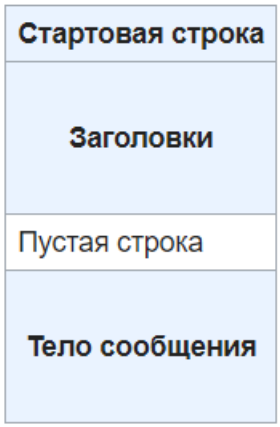

Рисунок 1 — Структура сообщения SIP

— стартовая строка — начальная строка любого SIP-сообщения. Если сообщение является запросом, в ней указывается тип запроса, адресат и номер версии протокола. Если сообщение является ответом на запрос, в ней указывается номер версии протокола, тип ответа и его короткая расшифровка;

— заголовки сообщений содержат информацию, необходимую для обработки сообщения (информация об отправителе, адресате, пути следования и пр.);

— тело сообщения содержит описание сеансов связи. Не все запросы содержат тело сообщения (например запрос BYE). Все ответы могут содержать тело сообщения, но содержимое тела в них бывает разным.

В первоначальной версии протокола SIP (RFC 3261) было определено шесть типов запросов. С помощью запросов клиент сообщает о текущем местоположении, приглашает пользователей принять участие в сеансах связи, модифицирует уже установленные сеансы, завершает их и так далее Тип запроса указывается в стартовой строке. В таблице 2 приведены виды запросов и их описание.

Таблица 2 — Виды и описание запросов SIP

№ п/п	Запрос	Описание
1	INVITE	Приглашение пользователя к сеансу связи. Обычно содержит SDP-описание сеанса
2	ACK	Подтверждение приёма ответа на запрос INVITE
3	BYE	Завершение сеанса связи. Может быть передан любой из сторон, участвующих в сеансе
4	CANCEL	Отмена обработки ранее переданных запросов. Не влияет на запросы, которые уже закончили обрабатываться
5	REGISTER	Перенос адресной информации для регистрации пользователя на сервере определения местоположения
6	OPTIONS	Запрос информации о функциональных возможностях сервера

В процессе развития в протокол был добавлен ряд расширяющих функциональность запросов. Их перечень приведён в таблице 3.

Таблица 3 — Дополнительные запросы SIP

№ п/п	Запрос	Описание
1	PRACK	Временное подтверждение
2	SUBSCRIBE	Подписка на получение уведомлений о событии
3	NOTIFY	Уведомление подписчика о событии
4	PUBLISH	Публикация события на сервере

Продолжение таблицы 3

№ п/п	Запрос	Описание
5	INFO	Передача информации, которая не изменяет состояние сессии
6	REFER	Запрос получателя о передаче запроса SIP
7	MESSAGE	Передача мгновенных сообщений средствами SIP
8	UPDATE	Модификация состояния сессии без изменения состояния

Ответы на запросы сообщают о результате обработки запроса либо передают запрошенную информацию. Структуру ответов и их виды протокол SIP унаследовал от протокола HTTP. Определено шесть типов ответов, несущих разную функциональную нагрузку. Тип ответа кодируется трёхзначным числом, самой важной является первая цифра, которая определяет класс ответа. В таблице 4 перечислены классы ответов и их описания.

Таблица 4 — Классы ответов SIP

Класс ответа	Описание
1XX	Информационные ответы; показывают, что запрос находится в стадии обработки. Наиболее распространённые ответы данного типа — 100 Trying, 180 Ringing, 183 Session Progress
2XX	Финальные ответы, означающие, что запрос был успешно обработан. В настоящее время используется 1 ответ — 200 OK
3XX	Финальные ответы, информирующие оборудование вызывающего пользователя о новом местоположении вызываемого пользователя, например, ответ 302 Moved Temporary
4XX	Финальные ответы, информирующие об отклонении или ошибке при обработке или выполнении запроса, например, 403 Forbidden или 404 Not Found
5XX	Финальные ответы, информирующие о том, что запрос не может быть обработан из-за отказа сервера, 500 Server Internal Error
6XX	Финальные ответы, информирующие о том, что соединение с вызываемым пользователем установить невозможно, например, ответ 603 Decline

На рисунке 2 приведена последовательность сообщений при установлении соединения между двумя абонентам с использованием прокси-сервера и сервера переадресации.

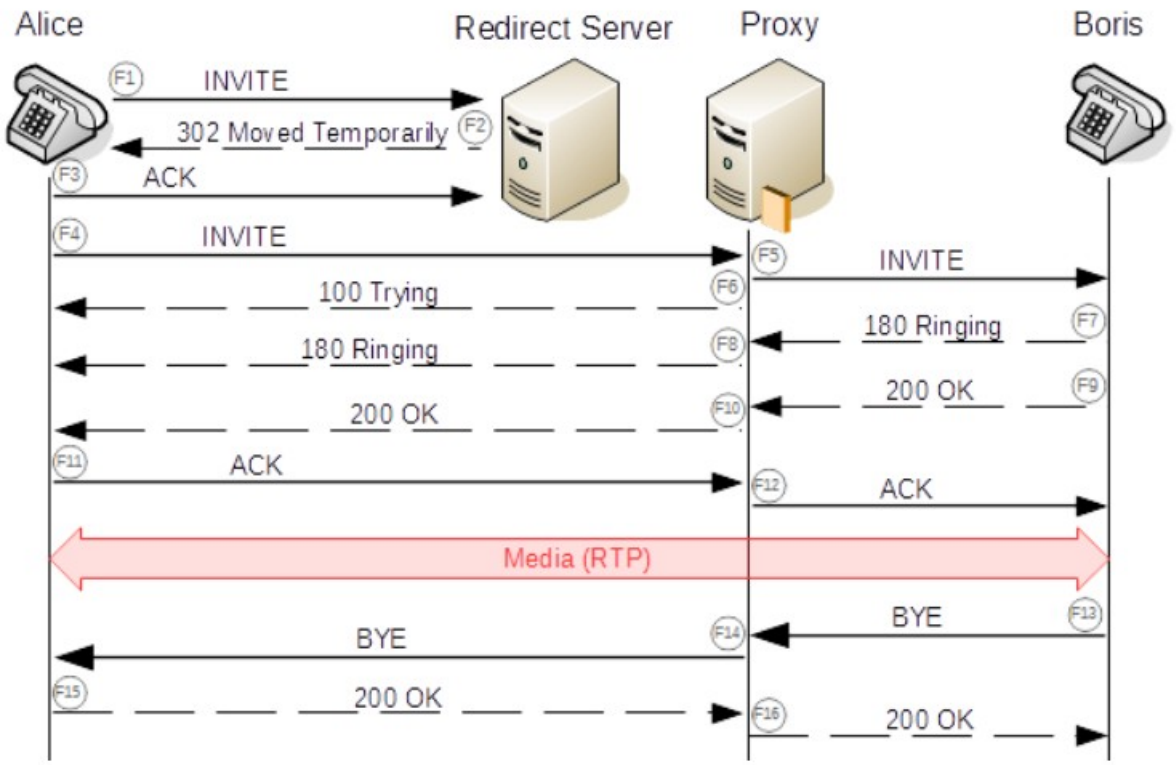

Рисунок 2 — Последовательность установления SIP-соединения

4.2 SDP

SDP (Session Description Protocol) — сетевой протокол прикладного уровня, предназначенный для описания сессии передачи потоковых данных, включая телефонию (телефонные сети общего применения и VoIP), Интернет-радио, приложения мультимедиа [18].

Сессия SDP может реализовывать несколько потоков данных. В протоколе SDP в настоящее время определены аудио, видео, данные, управление и приложения (поточные), сходные с MIME типами электронной почты в Интернет-адресах.

Сообщение SDP, передаваемое от одного узла другому, может указывать:

— адреса места назначения, которые могут быть для медиа-потоков мультикастинг-адресами;

— номера UDP портов для отправителя и получателя;

— медиа-форматы (например кодеки, описываемые профилем), которые могут применяться во время сессии;

— время старта и остановки. Используется в случае широковещательных сессий, например, телевизионных или радиопрограмм. Можно внести время начала, завершения и времена повторов сессии.

В таблице 5 перечислены поля, используемые в протоколе.

Таблица 5 — Поля протокола SDP

№ п/п	Строка (* - необязательная)	Описание
Описание сеанса		
1	v=	Версия протокола, в данный момент версия всегда 0
2	o=	Идентификаторы создателя/владельца и сессии
3	s=	Имя сессии, не может быть пустым
4	i=*	Информация о сессии
5	u=*	URI - адрес, используемый WWW-клиентами, с дополнительной информацией о сессии
6	e=*	E-mail адрес лица, ответственного за конференцию
7	p=*	Номер телефона лица, ответственного за конференцию
8	c=*	Информация для соединения - не требуется, если есть в описании всех медиаданных
9	b=*	Информация о занимаемой полосе пропускания канала связи
10	z=*	Установка для временной зоны
11	k=*	Ключ шифрования
12	a=*	Одна или несколько строк с описанием атрибутов сессии
Описание параметров времени		
1	t=	Время активности сеанса
2	r=*	Число попыток повторов, от нуля и больше
Описание данных передачи мультимедиа		
1	m=	Тип медиаданных и транспортный адрес устройства
2	i=*	Заголовок медиаданных
3	c=*	Информация для соединения - не обязательно, если описана в параметрах сеанса
4	b=*	Информация о занимаемой полосе пропускания канала связи

Продолжение таблицы 5

№ п/п	Строка (* - необязательная)	Описание
5	k=*	Ключ шифрования
6	a=*	От нуля и более строк с описанием атрибутов медиаданных,
Атрибуты медиа сессии		
1	a=rtpmap:PT КОДЕК	Уточнение типа кодека, если это необходимо, где PT - цифровое значение payload type, а кодек — название кодека и частота дискретизации
2	a=fmtp	Дополнительные атрибуты кодека, например, использование VAD или рекомендуемый битрейт
3	a=ptime:FPP	Размер RTP-пакета - или продолжительность (в миллисекундах) передаваемого отрезка медиа-данных в одном пакете RTP. В данном случае значение FPP - это произведение длины одного фрейма оцифрованного звука на количество фреймов в одном RTP-пакете
4	a=РЕЖИМ	Режим приёма и передачи, может принимать значения: sendonly — только отправка данных, recvonly — только приём, sendrecv — режим одновременных приёма и передачи (полнодуплексный режим), inactive — медиа-сессия неактивна

4.3 RTP

RTP (Real Time Transport Protocol) или протокол передачи данных в реальном времени определяет стандартный формат пакета для доставки звуковых и видеоданных по сети Интернет. Протокол определяется стандартом RFC 1889. [19]

Протокол RTP переносит в своём заголовке данные, необходимые для восстановления аудиоданных или видеоизображения в приёмном узле, а также данные о типе кодирования информации (JPEG, MPEG и т. п.). В заголовке данного протокола, в частности, передаются временная метка и номер пакета. Эти параметры позволяют при минимальных задержках определить порядок и момент декодирования каждого пакета, а также интерполировать потерянные пакеты.

RTP не имеет стандартного зарезервированного номера порта. Единственное ограничение состоит в том, что соединение проходит с использованием чётного номера, а следующий нечётный номер используется для связи по протоколу RTCP.

Спецификация RTP описывает два подпротокола:

— протокол передачи данных, RTP, который взаимодействует с передачей данных реального времени. Информация, предоставляемая посредством этого протокола, включает в себя отметку времени(для синхронизации), последовательный номер (для детектирования потери и дублирования пакетов) и формат полезной нагрузки, который определяет формат кодирования данных;

— протокол контроля, RTCP, используемый для определения качества обслуживания, обратной связи и синхронизации между медиа-потоками. Занимаемая полоса пропускания RTCP мала в сравнении с RTP, обычно около 5 процентов.

RTP-сессия устанавливается для каждого потока мультимедиа. Сессия состоит из IP-адреса и пары портов для RTP и RTCP. Например, аудио и видео потоки будут иметь различные RTP-сессии, позволяющие приемнику для этого выделить конкретный поток.

В таблице 6 приведена структура RTP пакета.

Таблица 6 — Структура RTP пакета

Биты	0-1	2	3	4-7	8	9-15	16-31
0	Ver.	P	X	CC	M	PT	Порядковый номер
32	Метка времени						
64	SSRC-идентиикатор						
96, если CC>0	CSRC-идентификаторы						
96+ (CC*32), если X=1	[Заголовок расширения — определённое профилем значение]						[Заголовок расширения — количество блоков данных по 32 бита (EHL)]

Продолжение таблицы 6

Биты	0-1	2	3	4-7	8	9-15	16-31
96+ (CC*32)+ 32	[Заголовок расширения — блоки данных]						
96+ (CC*32)+ X*(32+32 *EHL)	Данные						
Если P=1	Заполнение (Padding data)						L

0-1 — Ver. (2 бита) указывает версию протокола. Текущая версия — 2.

2 — P (один бит) используется в случаях, когда RTP-пакет дополняется пустыми байтами на конце.

3 — X (один бит) используется для указания расширений протокола, задействованных в пакете.

4-7 — CC (4 бита) содержит количество CSRC-идентификаторов, следующих за постоянным заголовком.

8 — M (один бит) используется на уровне приложения и определяется профилем. Если это поле установлено, то данные пакета имеют какое-то особое значение для приложения.

9-15 — PT (7 бит) указывает формат полезной нагрузки и определяет её интерпретацию приложением.

64-95 — SSRC указывает источник синхронизации.

EHL (Extension Header Length) — количество 32-битных слов в блоке данных расширения заголовка.

L — последний байт в пакете, определяющий длину области заполнения в байтах (используется для выравнивания в последнем пакете).

4.4 ZRTP

ZRTP — криптографический протокол согласования ключей шифрования, используемый в системах передачи голоса по IP-сетям (VoIP). ZRTP описывает метод получения ключей по алгоритму Диффи-Хеллмана для организации Secure Real-time Transport Protocol (SRTP). ZRTP осуществляет согласование ключей в том же потоке RTP, по которому установлена аудио/видео связь, то есть не требует отдельного канала связи [20].

Во момент начала звонка создаётся публичный идентификатор, используемый при создании ключей, с помощью которых будет шифроваться медиа поток разговора. Таким образом ключ действителен только в течение одного разговора, образуя сессию Secure RTP (SRTP). При разрыве соединения ключ и весь криптографический контекст уничтожается, что обеспечивает совершенную прямую секретность (PFS). Таким образом существует потенциал встраивания этого механизма в уже существующие программные VoIP продукты, шлюзы и IP-телефоны.

Протокол не требует заранее сгенерированных ключей, или поддержки инфраструктуры обмена ключей (PKI), или центра сертификации (CA). Это избавляет от сложностей создания структуры авторизации, основанной на доверенной поддержке, которая, например, применяется в шифровании SSL. Главной целью организации канала зашифрованной связи — избежать возможности присутствия «Человека посередине» (man in the middle), обеспечивая единую криптографическую защиту между любыми двумя точками.

ZRTP может теоретически применяться совместно с любыми сигнальными протоколами, использующими RTP для передачи медиа потока, включая SIP, H.323, SCCP, MGCP, Unistim и Jingle, так как в теории ZRTP не зависит от сигнализации, осуществляя обмен ключей в медиа сессии RTP.

Сам по себе алгоритм обмена ключами Диффи-Хеллмана не может обеспечить защиту от присутствия человека посередине (man in the middle). Для аутентификации ZRTP использует Short Authentication String (SAS) "короткую строку аутентификации", являющуюся сокращённым представлением криптографического хэша полученных ключей Диффи-Хеллмана. Значения SAS вычисляются на каждой стороне соединения, абоненты передают их друг другу голосом для сверки. Если значения не совпали, то с большой уверенностью можно предположить присутствие Человека посередине. Использование алгоритма Диффи-Хеллмана даёт потенциальному Человеку_посередине всего одну попытку сгенерировать правильную SAS при попытке атаки. Так как SAS получается из старших 32 битов хэша и имеет две формы представления (16-битную B256 в виде пары слов из списка PGP, и 20-битную B32 в виде четырёх символов), она является очень короткой, вероятность необнаружения атаки при использовании SAS в формате B256 равна 1/65536. Применение SAS в формате B32 понижает вероятность необнаружения атаки в 16 раз (по сравнению с B256), до 1/1048576.

Вторым уровнем защиты ZRTP от атак "человек посередине" является непрерывность ключевого материала. Хэш ключевой информации предыдущего вызова подмешивается к параметрам алгоритма Диффи-Хеллмана при следующем вызове (между этими же абонентами), что придаёт протоколу ZRTP сходство с SSH. Если человека посередине не было при первом вызове, он исключается из всех последующих.

Таким образом для реализации вызовов с помощью IP-телефонии достаточно применение протокола SIP для установления соединения и RTP для передачи данных, при условии, что в SIP будет передаваться описание сессии по протоколу SDP. Протокол ZRTP не применялся, так как обмен ключами был реализован без использования сетевых технологий, кроме того это позволило дополнительно замаскировать зашифрованный трафик, так как были использованы только протоколы для передачи открытой информации.

5 Анализ отечественных криптографических алгоритмов

5.1 ГОСТ 28147-89

Для обеспечения криптографической защиты передаваемой с помощью рассмотренных протоколов речевой информации был проведён анализ отечественных криптографических алгоритмов и выбран алгоритм, наиболее соответствующий требованиям протоколов.

ГОСТ 28147-89 — российский стандарт симметричного блочного шифрования, принятый в 1989 году. Полное название — «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».

Стандарт представляет собой блочный шифр с 256-битным ключом и 32 раундами преобразования, оперирующий 64-битными блоками. Алгоритм основан на применении сети Фейстеля. [21]

Алгоритмом предусмотрены 4 режима работы:

— режим простой замены;

— режим гаммирования;

— режим гаммирования с обратной связью;

— режим выработки имитовставки.

В режиме простой замены 64-битные блоки шифруются независимо друг от друга, следовательно в данном режиме могут шифроваться только тексты длиной кратной 64 битам. При этом при шифровании одинаковых блоков открытого текста получаются одинаковые блоки шифротекста, что облегчает криптоанализ.

В режиме гаммирования формируется гамма, которая затем побитово складывается по модулю 2 с исходным открытым текстом для получения шифротекста. Данный режим приспособлен для поточного шифрования и лишён недостатков режима простой замены.

Режим гаммирования наиболее удобен для реализации в мобильной аппаратуре передачи речевой информации.

В режиме гаммирования с обратной связью гамма формируется на основе предыдущего блока зашифрованных данных. Стойкость шифрования повышается, но происходит размножение ошибок из-за взаимосвязи блоков.

В режиме выработки имитовставки не происходит шифрования как такового. Для 2 и более блоков открытого текста размером 64 бита на основе ключевых данных вырабатывается дополнительный блок, который используется для подтверждения того, что текст не был изменён или подменён в процессе передачи по каналу связи. Проверка осуществляется путём вычисления этого блока принимающей стороной на основе ключа и расшифрованного текста.

К достоинствам алгоритма можно отнести высокое быстродействие программной реализации алгоритма и наличие защиты от навязывания ложных данных путём выработки имитовставки.

5.2 ГОСТ Р 34.12-2015

Стандарт, пришедший на смену ГОСТ 28147-89, включает в себя следующие два алгоритма шифрования, работающие с блоками в 64 и 128 бит соответственно [22]:

— «Магма» - полный аналог ГОСТ 28147-89, перенесённый в данный стандарт;

— «Кузнечик» — симметричный алгоритм блочного шифрования с размером блока 128 бит и длиной ключа 256 бит и использующий для генерации раундовых ключей сеть Фейстеля.

В отличие от алгоритма «Магма» в основе «Кузнечика» лежит не сеть Фейстеля, а так называемая SP-сеть: преобразование, состоящее из нескольких одинаковых раундов, при этом каждый раунд состоит из нелинейного и линейного преобразований, а также операции наложения ключа. В отличие от

сети Фейстеля, при использовании SP-сети преобразуется весь входной блок, а не его половина.[21]

Данные алгоритмы могут работать в тех же режимах, что и ГОСТ 28147-89.

Таким образом для использования в мобильной аппаратуре передачи речевой информации был выбран алгоритм ГОСТ 28147-89/ ГОСТ Р 34.12-2015 «Магма» в режиме гаммирования в силу его меньшей вычислительной сложности и возможности генерации гаммы для имитации потокового режима работы.

6 Анализ возможных способов ввода ключевой информации в мобильной аппаратуре

Кроме выбора алгоритма шифрования речевой информации для обеспечения её защищённости был выполнен анализ существующих методов ввода ключевой информации в мобильной аппаратуре и выбран оптимальный, обеспечивающий своевременное и безопасное обновление ключа шифрования с учётом условий функционирования мобильной аппаратуры.

Для ввода ключевой информации в мобильную аппаратуру могут использоваться следующие методы:

— ручной ввод ключевой информации;

— ввод ключевой информации с отчуждаемых электронных носителей информации (SD-карт);

— передача ключевой информации с помощью беспроводных сетей связи;

— передача ключевой информации с помощью ближней бесконтактной связи (NFC);

— ввод ключевой информации с бумажных носителей информации посредством штрих-кодов.

Ручной ввод ключевой информации требует посимвольного ввода ключевой информации в мобильную аппаратуру, вследствие чего скорость ввода информации будет более низкой чем в рассмотренных далее методах. При этом необходим способ распространения ключевой информации в понятном пользователю виде. Исходя из этого данный метод не является оптимальным в силу наличия вероятности неверного ввода информации пользователем или непреднамеренной ошибке при передаче пользователю ключевой информации, что приведёт к невозможности информационного обмена.

Хранение ключевой информации на SD-карте влечёт за собой необходимость дополнительно хранения этой SD-карты и централизованного

своевременного обновления ключевой информации. Однако для ввода ключевой информации не требуется никаких дополнительных условий кроме подключения SD-карты к мобильной аппаратуре перед началом работы с ней.

Передача ключевой информации с помощью беспроводных сетей связи позволяет выполнять ввод ключевой информации без какого-либо участия пользователя и подключения внешних устройств. При использовании данного метода требуется специализированная процедура распространения ключей, снижающая вероятность перехвата и компрометации ключа в канале связи до приемлемых значений, что однако не исключает возможности реализации угроз нарушения конфиденциальности ключевой информации посредством использования уязвимостей протоколов обмена информацией в беспроводных сетях связи.

NFC – «ближняя бесконтактная связь» — технология беспроводной высокочастотной связи малого радиуса действия, с помощью которой можно обмениваться данными между устройствами в пределах 10 сантиметров. Передача ключевой информации с использованием данной технологии выполняется с высокой скоростью и не требует специальных навыков пользователя. Однако использование NFC требует соответствующего оснащения как на мобильной аппаратуре так и на устройстве, с которого осуществляется ввод ключевой информации. Также передача ключевой информации с помощью NFC создаёт дополнительную угрозу нарушения конфиденциальности ключевой информации за счёт её утечки посредством ПЭМИН.

Использование штрихкодов требует наличия в мобильной аппаратуры камеры, характеристики которой позволяют выполнять распознавание штрих-кодов, а устройство генерации ключей должно быть способно генерировать штрих-коды требуемого формата. Для ввода ключевой информации посредством штрих-кодов могут использовать как минимум двухмерные штрих-коды (QR-коды, AZTEC, PDF417), так как существующие стандарты линейных

одномерных штрих-кодов (EAN, UPC, Code39, Code128, Codabar, Interleaved 2 of 5) ограничивают максимальный объём содержащейся информации 30 байтами, в то время как один ключ шифрования в алгоритме ГОСТ Р 34.12-2015 «Магма» имеет размер 32 байта, что накладывает дополнительные требования по сложности распознаваемых штрих-кодов. Также перед началом работы с аппаратурой будет необходимо считать штрих-код, содержащий ключевую информацию, что возможно далеко не всегда в силу повреждения носителя штрих-кода, недостаточной освещённости или недостатков камеры.

Таким образом хранение ключевой информации на SD-карте — наиболее универсальный метод из представленных, так как при вводе ключевой информации с SD-карты исключена возможность неверного ввода и обеспечивается высокая скорость ввода в отличие от ручного метогда, исключена возможность перехвата как при передаче с помощью беспроводных сетей, не требуется специальных технических средств как при использовании NFC или штрих-кодов, а так же не требуется совершать дополнительных действий кроме подключения SD-карты как при ручном вводе, использовании штрих-кодов или NFC. Единственное ограничение — необходимость возможности подключения SD-карты к мобильной аппаратуре, однако в настоящее время подавляющее большинство образцов мобильной аппаратуры оснащено соответствующими разъёмами.

7 Анализ структуры макета с определением трактов прохождения речевой информации

После выбора конкретных решений в области передачи речевой информации и особенностей используемого криптографического алгоритма был проведён анализ структуры макета с определением трактов прохождения речевой информации.

На рисунке 3 приведена структура макета.

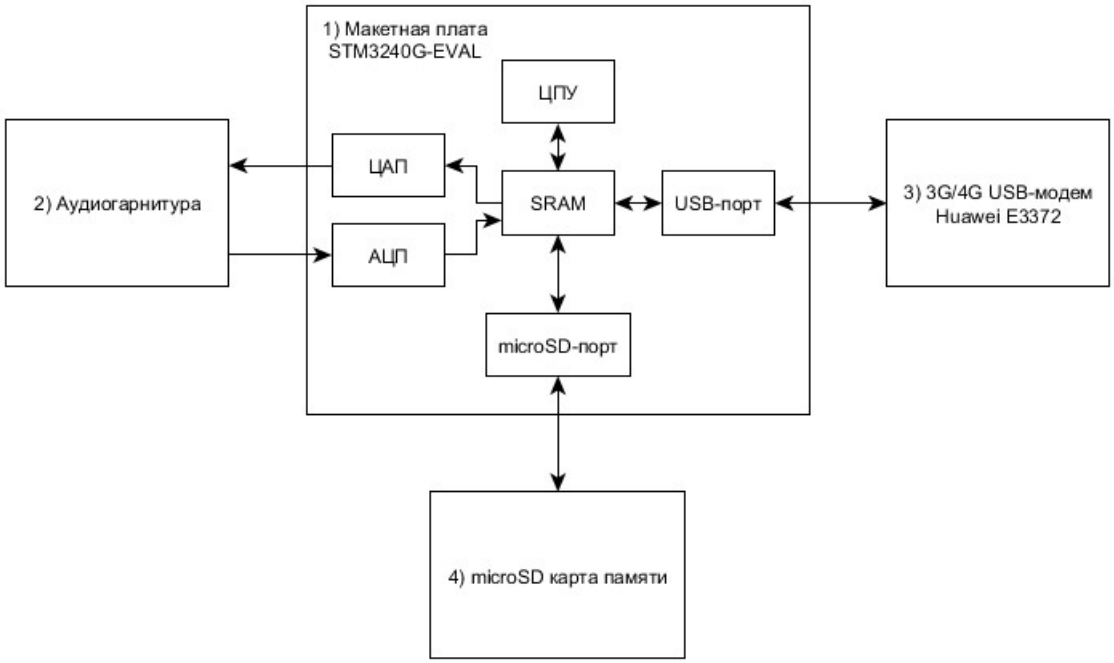

Рисунок 3 — Структура макета

Цифрой 1 в структуре отмечена макетная плата STM3240G-EVAL.

Данная плата оснащена следующими компонентами, используемыми в работе макета [21]:

— микропроцессор STM32F407IGH6, оснащённый контроллером прямого доступа к памяти;

— 16 Мбит SRAM;

— разъём для установки microSD карты памяти;

— USB-OTG (HS и FS) порты с Micro-AB коннекторами;

— интерфейс I^2C-шины;

— аудио-ЦАП I^2S;

— аудио АЦП;

— разъём для подключения стерео аудио гарнитуры;

— LCD сенсорный экран.

Цифрой 2 в структуре отмечена аудиогарнитура, включающая в себя микрофон и наушники.

Цифрой 3 в структуре обозначен 3G/4G USB-модем Huawei E3372.

Цифрой 4 в структуре обозначена microSD карта памяти, используемая для хранения ключевой информации.

При реализации тракта передачи речевой информации целесообразно использовать механизм DMA при записи и воспроизведении звука для выполнения данных операций без непосредственного участия процессора макета.

Таким образом для записи и воспроизведения речевой информации, её кодирования, шифрования и передачи будут использоваться АЦП, I^2S-ЦАП, механизм DMA, аудио гарнитура, 3G/4G USB-модем, SRAM, микропроцессор, на рисунке 4 представлены тракты прохождения речевой информации в макете при её передаче и приёме.

Рисунок 4 — Тракты прохождения речевой информации в макете

8 Разработка алгоритма работы программного обеспечения макета

На основе результатов анализа структуры макета был разработан алгоритм работы программного обеспечения макета.

Первоначально была определена схема организации связи между макетами. Для установки соединения между макетами и передачи данных в рамках вызова будет использоваться общедоступный SIP-сервер. На рисунке 5 приведена схема организации связи между макетами.

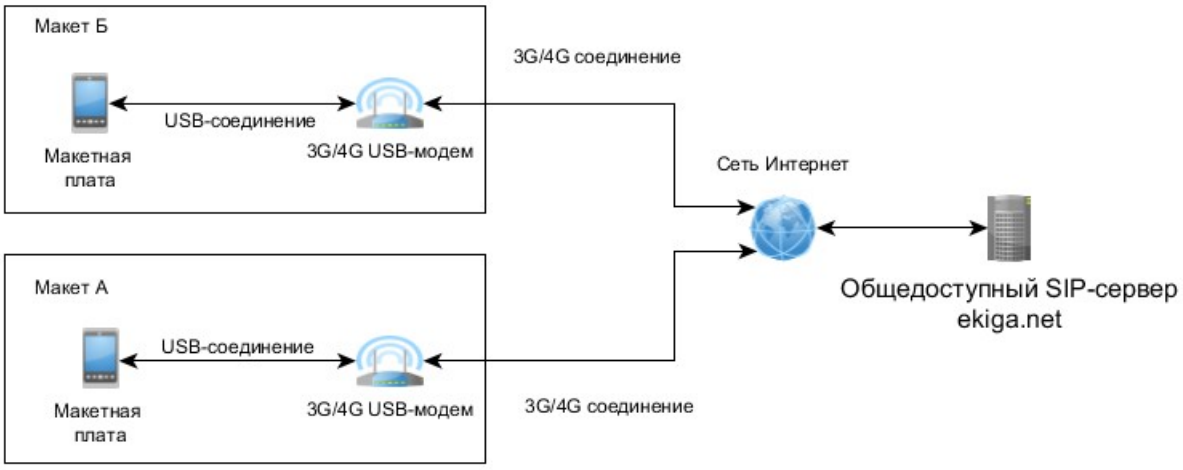

Рисунок 5 — Схема организации связи между макетами

Далее был разработан алгоритм работы программного обеспечения макета. Первоначально была построена UML-диаграмма последовательности, на которой отражён порядок информационного взаимодействия между макетами и общедоступным SIP-сервером при инициализации макетов и в ходе совершения вызова. На диаграмме отражён порядок обмена сообщениями между макетами и SIP-сервером при регистрации макетов на сервере, при установлении соединения во время совершения вызова, при обмене аудио-данными и при разрыве соединения во время завершения вызова по протоколам SIP и RTP, как описано в подразделах 4.1 и 4.3. Диаграмма информационного взаимодействия представлена на рисунке 6.

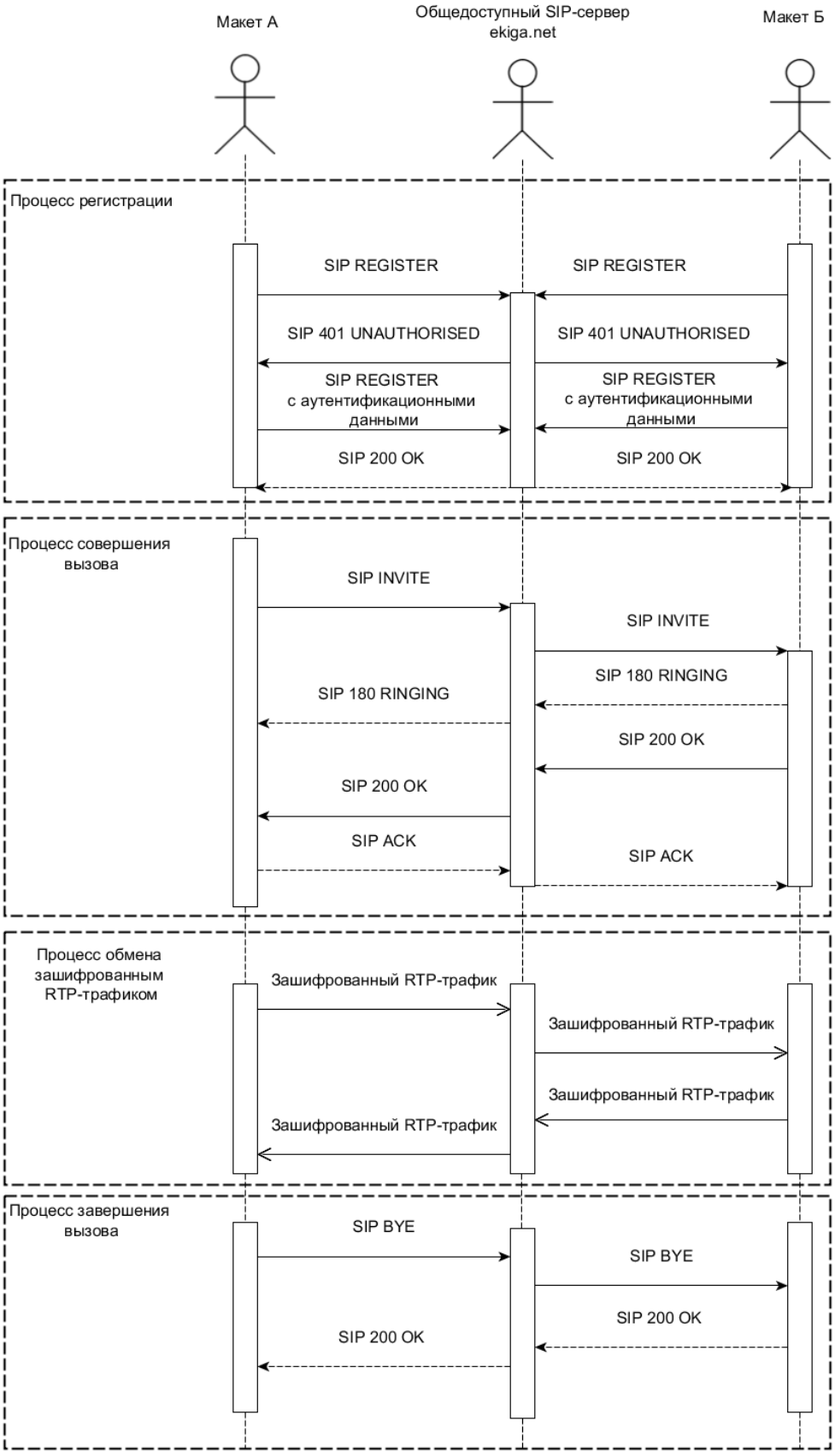

Рисунок 6 — Диаграмма информационного взаимодействия макетов

На диаграмме представлен процесс информационного обмена между общедоступным SIP-сервером ekiga.net и 2 макетами. Обмен выполняется в ходе следующих процессов:

— регистрация макетов — оба макета отправляют на SIP-сервер сообщения SIP после чего SIP-сервер запрашивает их аутентификацию, макеты отправляют ответ на запрос аутентификации и SIP-сервер подтверждает регистрацию отправкой на макеты сообщения SIP 200 OK;

— вызов макетом А макета Б — макет А через SIP-сервер отправляет макету Б запрос на соединение SIP INVITE, макет Б отвечает на запрос сообщениями SIP 180 RINGING и SIP 200 OK, макет А завершает процесс установления соединения отправкой сообщения SIP ACK;

— обмен зашифрованным RTP-трафиком между макетом А и макетом Б в ходе совершения вызова — макет А отправляет считанную, закодированную и зашифрованную речевую информацию макету Б в RTP-пакетах и принимает аналогичные пакеты от макета Б, макет Б отправляет считанную, закодированную и зашифрованную речевую информацию макету А в RTP-пакетах и принимает аналогичные пакеты от макета А;

— завершение вызова — макет А отправляет макету Б сообщение SIP BYE, макет Б подтверждает завершение вызова сообщением SIP 200 OK.

Исходя из схемы организации связи, диаграммы информационного взаимодействия макетов и требований технического задания был разработан алгоритм работы приложения макета. Алгоритм представлен на рисунке 7.

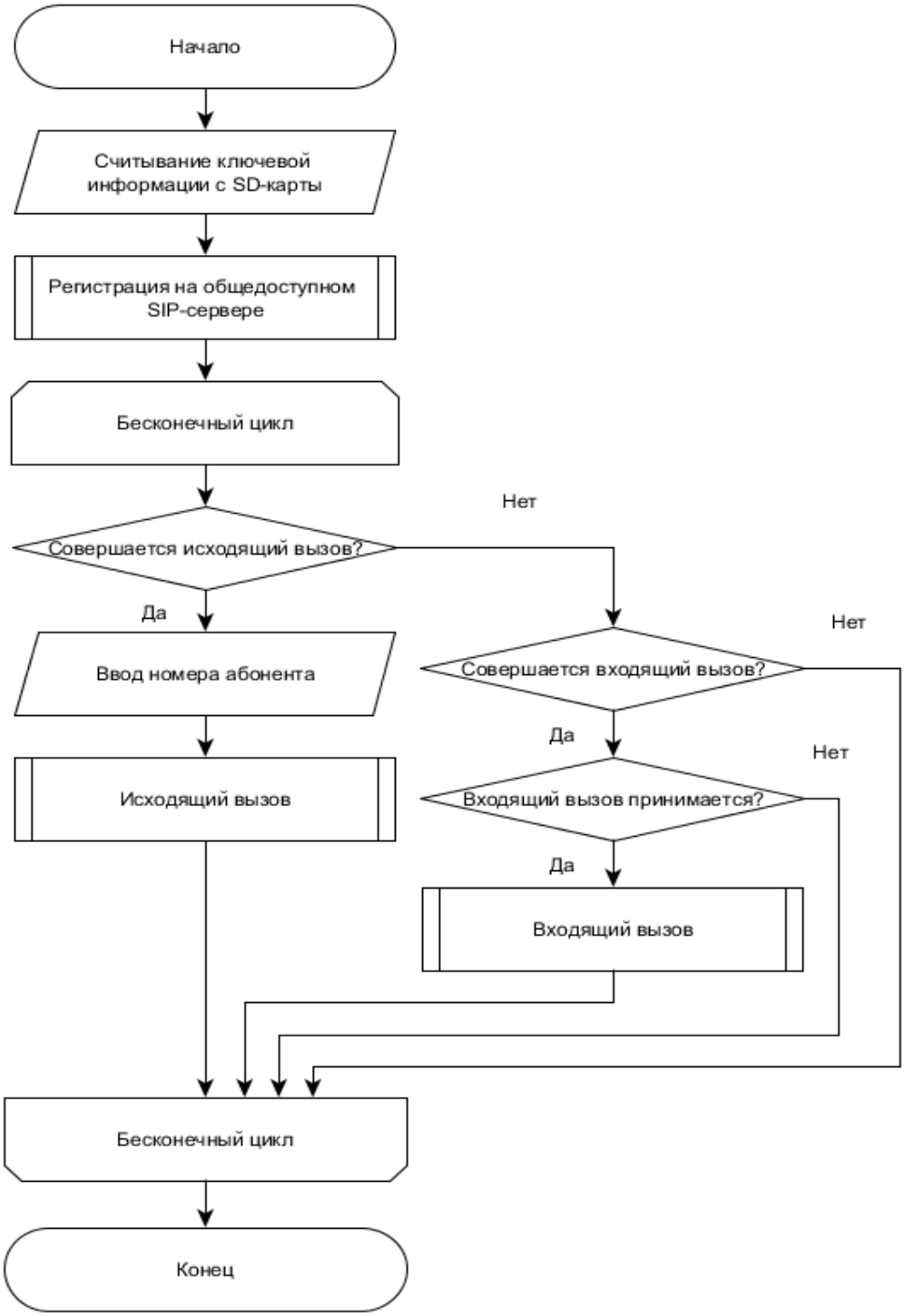

Рисунок 7 — Алгоритм работы программного обеспечения макета

Первоначально выполняется подготовка программного обеспечения к работе: считывается ключевая информация с SD-карты и выполняется регистрация на общедоступном SIP-сервере. Алгоритм процедуры регистрации предоставлен на рисунке 8.

Далее постоянно выполняется проверка наличия входящего вызова от аналогичного макета, а также поддерживается готовность к вводу номера и вызову аналогичного макета. В случае поступления входящего вызова начинается процедура приёма вызова, в случае совершения исходящего номера начинается процедура совершения исходящего вызова. Алгоритмы указанных процедур представлены на рисунках 9 и 10.

Рисунок 8 — Алгоритм регистрации макета

Процесс регистрации заключается в отправке запроса на регистрацию, получении ответного запроса на аутентификацию, вычисления необходимого ответа, повторного запроса на регистрацию с полученными данными и ожидании сообщения об успешном завершении регистрации.

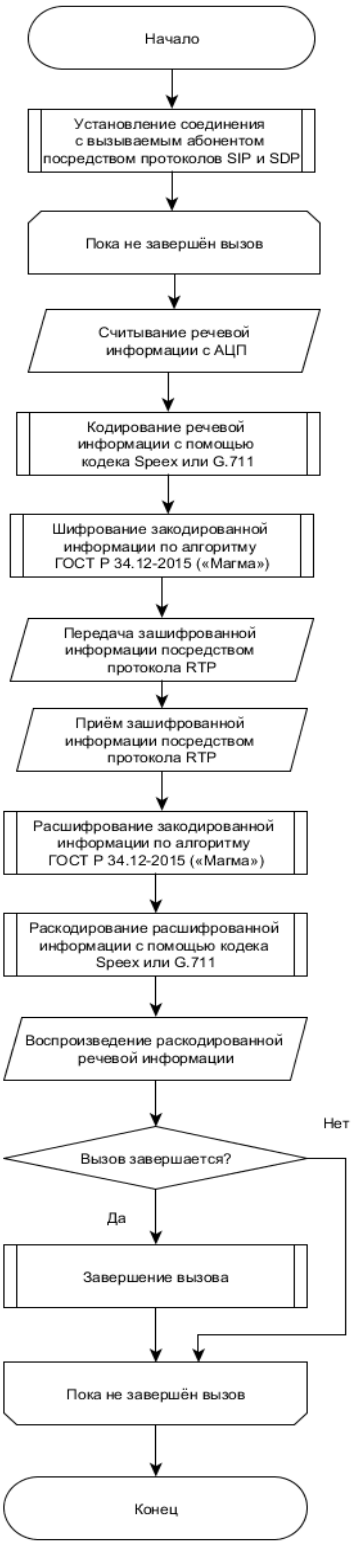

Рисунок 9 — Алгоритм совершения исходящего вызова

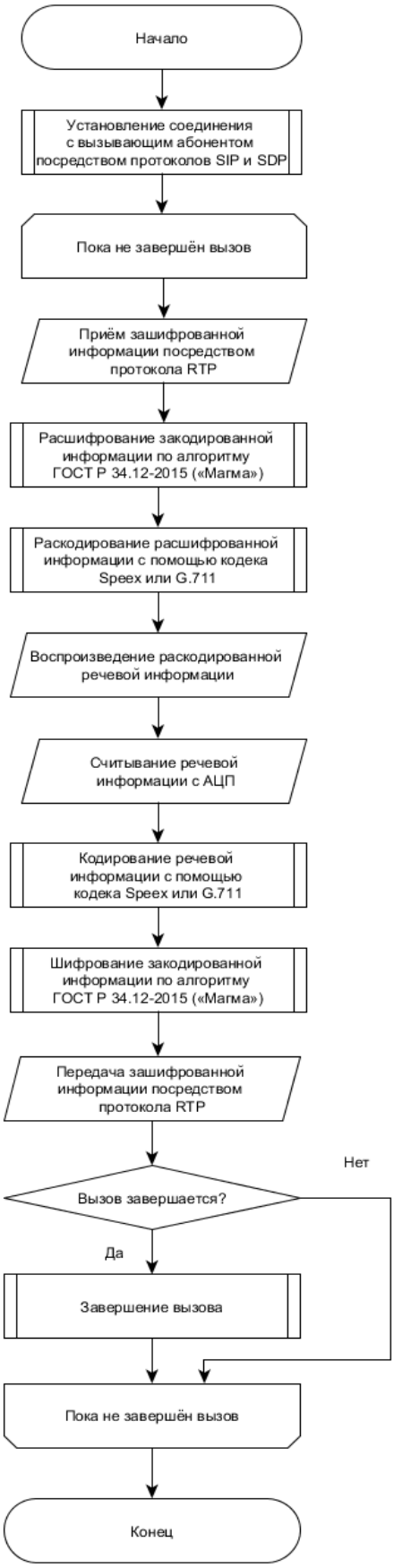

Рисунок 10 — Алгоритм приёма входящего вызова

Первоначально в исходящем и входящем вызовах происходит установление соединения с собеседником посредством протоколов SIP и SDP, алгоритм данной процедуры для входящего и исходящего вызовов представлен на рисунках 11 и 12.

Далее до завершения вызова одним из собеседников выполняется приём зашифрованной речевой информации, заключающийся в приёме RTP-пакета, расшифровании, раскодировании и воспроизведении его содержимого и передача зашифрованной речевой информации, заключающаяся в считывании речевой информации, её кодировании, зашифровании и отправке в RTP-пакете.

Как входящий так и сходящий вызов оканчиваются процедурой завершения вызова, алгоритм которой приведён на рисунке 13.

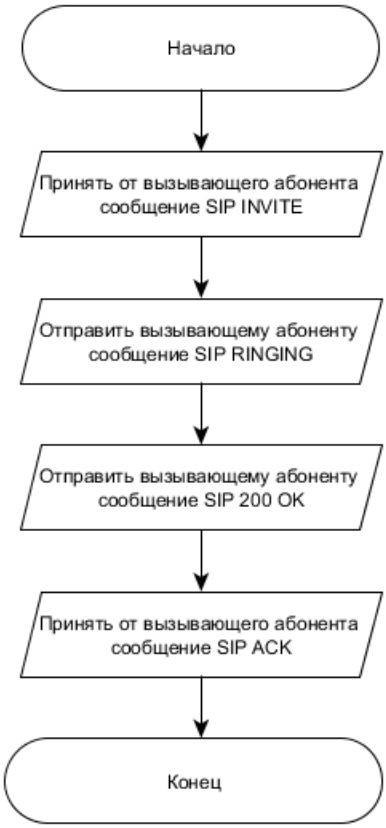

Рисунок 11 — Алгоритм установления соединения с вызываемым абонентом

Процедура установления соединения с вызываемым абонентом начинается с отправки сообщения SIP INVITE, далее ожидается приём

сообщений SIP RINGING и SIP 200 OK от вызываемого абонента, после чего вызываемому абоненту отправляется сообщение SIP ACK.

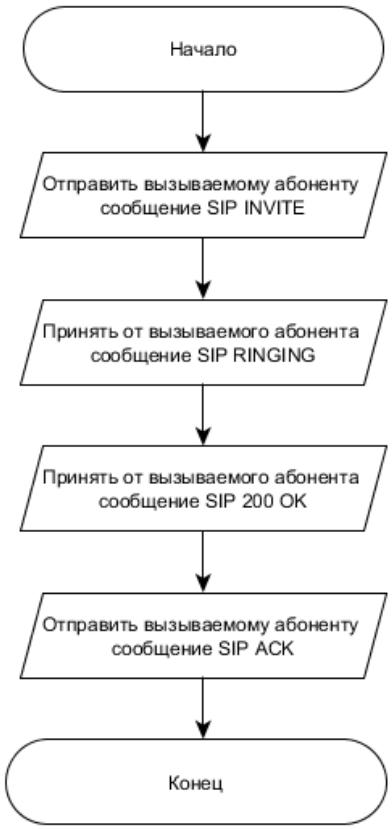

Рисунок 12 — Алгоритм установления соединения с вызывающим абонентом

Процедура установления соединения с вызывающим абонентом начинается с приёма сообщения SIP INVITE от вызывающего абонента, далее выполняется отправка сообщений SIP RINGING и SIP 200 OK, после чего от вызывающего абонента принимается SIP ACK.

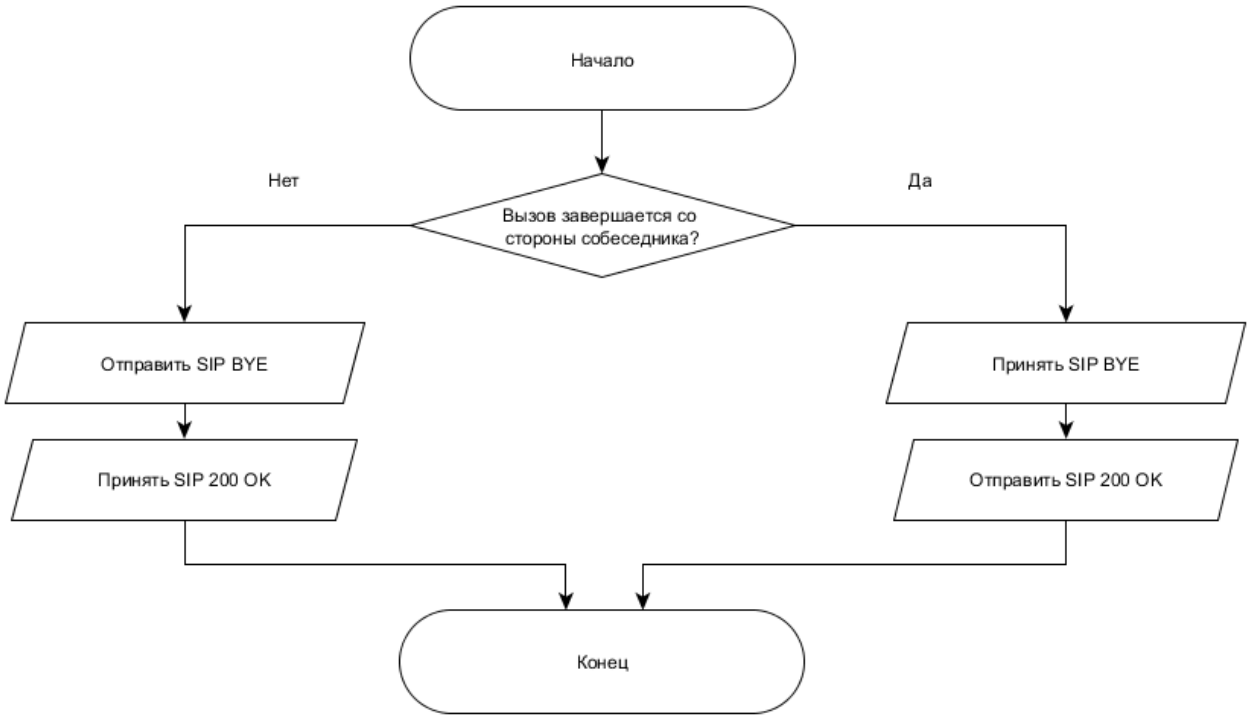

Рисунок 13 — Алгоритм завершения вызова

Если вызов совершается со стороны собеседника, то сперва выполняется приём сообщения SIP BYE, после чего отправляется сообщение SIP 200 OK, иначе сперва отправляется сообщение SIP BYE, после чего ожидается приём сообщения SIP 200 OK.

9 Разработка программного обеспечения макета

9.1 Описание условий разработки

Разработка программного обеспечения макета выполнялась на языке C/C++ в среде разработки IAR Embedded Workbench for ARM 7.40.3.8938.

Отладка программного обеспечения макета выполнялась с помощью отладчика J-Link-ARM.

В целях упрощения настройки аппаратных компонентов макетной платы подключение библиотек устройств и их настройка проводились с помощью программы STM32CubeMX.

9.2 Разработка модуля считывания и воспроизведения речевой информации

Основные функции данного модуля — считывание речевой информации с микрофона аудио-гарнитуры и сохранение её в памяти макета и передача речевой информации из памяти макета на наушники аудио-гарнитуры для её воспроизведения. На рисунке 14 приведены алгоритмы считывания и воспроизведения речевой информации.

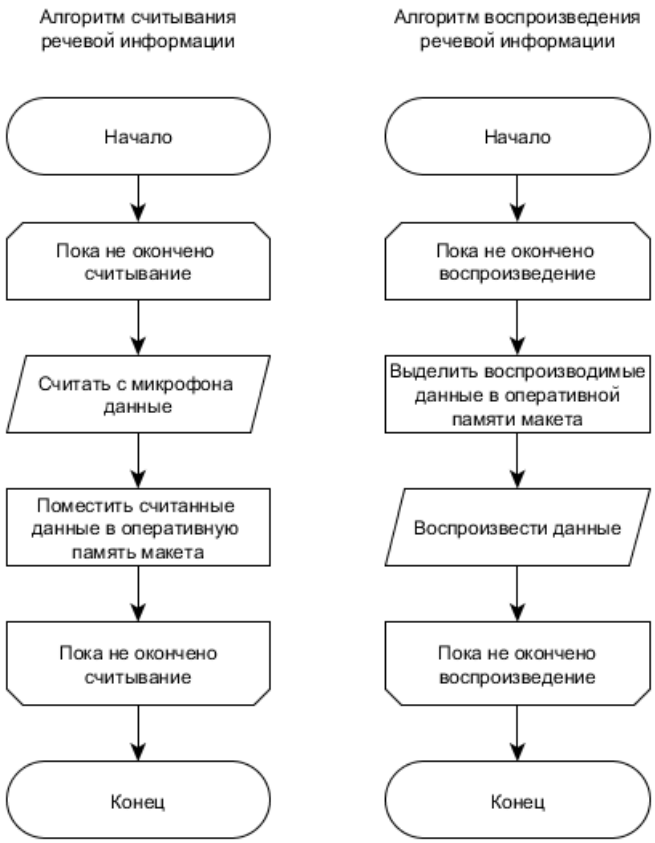

Рисунок 14 — Алгоритмы считывания и воспроизведения речевой информации

Таким образом в процессе работы модуля постоянно выполняются операции чтения и записи в оперативную память. В макетной плате STM3240G-EVAL реализован механизм прямого доступа к памяти (DMA), что позволяет выполнять запись и чтение из оперативной памяти без непосредственного участия процессора. То есть модуля выполняется параллельно с работой остальной программы.

Было выполнено написание исходного кода модуля и проведена его отладка.

В результате был получен модуль, в котором реализованы функции считывания и воспроизведения звука параллельно с работой главного модуля проекта посредством использования механизма DMA.

Для реализации модуля были сделаны следующие изменения в проекте:

— для реализации записи звука с помощью программы STM32CubeMX был подключен аналого-цифровой преобразователь макетной платы и канал механизма прямого доступа к памяти номер 2 поток 0. Также был подключен таймер номер 3 с частотой срабатывания 50 Герц для тактирования работы аналого-цифрового преобразователя;

— для воспроизведения звука было решено использовать интерфейс I^2S, для этого была подключена библиотека драйвера микросхемы cs43l22 [23], с помощью программы STM32CubeMX был подключен канал прямого доступа к памяти номер 1 поток 4 [23].

Также был разработаны функции, обеспечивающие работу модуля. Разработанные функции перечислены в таблице 7.

Таблица 7 — Функции модуля считывания и воспроизведения звука

Имя функции	Файл, в котором функция определена	Описание функции
`init_audio`	`broker.c`	инициализация драйвера воспроизведения звука
`start_audio_in`	`broker.c`	начать запись звука (запуск таймера АЦП и работы DMA)
`start_audio_out`	`broker.c`	начать воспроизведение звука (запуск с помощью драйвера BSP работы DMA)
`stop_audio_in`	`broker.c`	остановить запись звука (остановка таймера и работы DMA)
`stop_audio_out`	`broker.c`	остановить воспроизведение звука (остановка драйвером BSP работы DMA)
`in_irq_handler`	`broker.c`	обработка записанного звука, принятого от DMA
`out_irq_handler`	`broker.c`	обработка звука перед воспроизведением и передача его DMA

Для реализации частоты дискретизации воспроизводимого и записываемого звука равной 8 килогерц при приёме и воспроизведении звука обрабатываются фрагменты данных размером 160 байт 50 раз в секунду.

Во избежание конфликта между обработчиками записанного и воспроизводимого звука и DMA для хранения записанного звука используется

буфер размера вдвое большего, чем размер обрабатываемого фрагмента — 320 байт.

При воспроизведении звука каждый фрагмент удваивается в размерах для воспроизведения в обоих каналах аудио-гарнитуры, то есть буфер имеет размер в 4 раза больше одного фрагмента — 640 байт.

Для передачи сохранённого с помощью DMA звука обработчику была дополнена функция `DMA2_Stream0_IRQHandler` в файле stm`32f4xx_it.c`.

Для воспроизведения звука были переопределены функции драйвера BSP `BSP_AUDIO_OUT_TransferComplete_CallBack` и `BSP_AUDIO_OUT_HalfTransfer_CallBack` в файле `main.c`.

В данных функциях предусмотрен вызов обработка при записи/чтении половины буфера и всего буфера.

9.3 Разработка модуля кодирования и раскодирования речевой информации

Цель работы модуля кодирования и раскодирования речевой информации — кодирование считанной и раскодирование воспроизводимой речевой информации с помощью кодеков G.711 или SPEEX. На рисунке 15 приведён алгоритм работы модуля кодирования и раскодирования в сочетании с модулем считывания и воспроизведения речевой информации независимо от выбранного кодека.

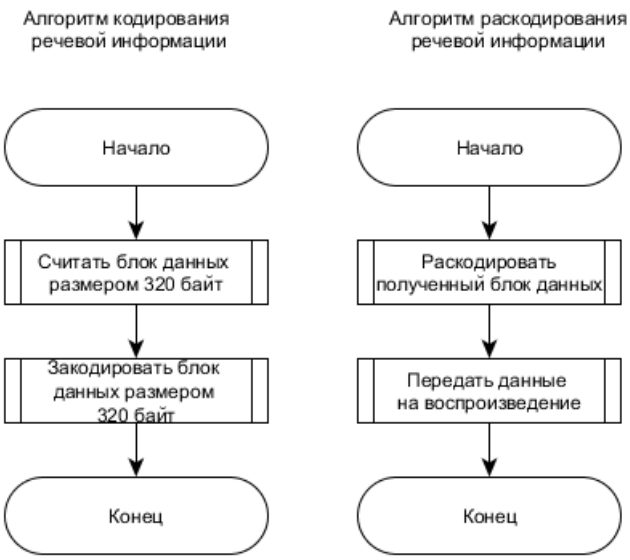

Рисунок 15 — Алгоритм работы модуля кодирования и раскодирования в сочетании с модулем считывания и воспроизведения речевой информации

То есть работа модуля заключается вызове функций кодирования речевой информации по окончании считывания блока данных размером 320 байт и вызове функций раскодирования после приёма блока закодированных данных перед их воспроизведением.

За основу модуля были взяты библиотеки кодеков SPEEX [24] и G.711 [25], взятые из открытых источников.

SPEEX инициализируется со следующими параметрами:

— размер кодируемого блока речевой информации — 320 байт;

— скорость передачи данных — 18200 бит в секунду.

Кодек G.711 не требует дополнительной настройки для работы, так как сразу предназначен для обработки звука с частотой дискретизации 8 килогерц со скоростью передачи данных 64000 бит в секунду. Для использования в программном обеспечении макета было выбрано кодирование по закону `ulaw`.

Было выполнено написание исходного кода модуля и проведена его отладка.

В результате был реализован модуль, в котором реализованы функции кодирования и раскодирования речевой информации с помощью кодека SPEEX

или G.711 с возможностью выбора кодека. Также с помощью функций разработанного модуля был дополнен модуль считывания и воспроизведения речевой информации для выполнения кодирования считанной и раскодирования воспроизводимой речевой информации.

В таблице 8 перечислены разработанные функции, предназначенные для работы с кодеком SPEEX.

Таблица 8 — Функции для работы с кодеком SPEEX

Имя функции	Описание функции
`open`	инициализация кодека с установкой степени сжатия звука
`close`	инициализация кодека с установкой степени сжатия звука
`simple_encode`	закодировать 1 блок речевой информации
`simple_decode`	раскодировать один блок речевой информации

Функция `open` была добавлена в функцию `init_audio` для подключения кодека SPEEX при инициализации аудио.

В функции `in_irq_handler` и `out_irq_handler` было добавлено соответственно кодирование и раскодирование звука посредством обоих кодеков. Переключение между кодеками производится определением констант в файле `broker.h`: `CODEC` – SPEEX, `G711` – G.711.

9.4 Разработка модуля зашифрования и расшифрования закодированной речевой информации

Задача данного модуля — обеспечение конфиденциальности и подтверждение подлинности передаваемой речевой информации, что достигается путём шифрования передаваемой речевой информации и вычисления имитовставки от закодированной речевой информации на передающей стороне и проверки правильности переданной имитовставки на принимающей стороне.

За основу модуля была взята реализация алгоритма ГОСТ Р 34.12-2015 («Магма») [26], находящаяся в свободном доступе.

Ключевая информация перед стартом работы макета хранится на SD-карте, в процессе инициализации макета ключевая информация считывается в оперативную память макета для дальнейшего использования.

Для работы модуля зашифрования и расшифрования речи используются два ключа шифрования: один для зашифрования и расшифрования, второй для выработки имитовставки.

На рисунках 16 и 17 представлены алгоритмы работы модуля при передаче и приёме речевой информации.

Рисунок 16 — Алгоритм работы модуля зашифрования и расшифрования закодированной речевой информации при передаче речевой информации

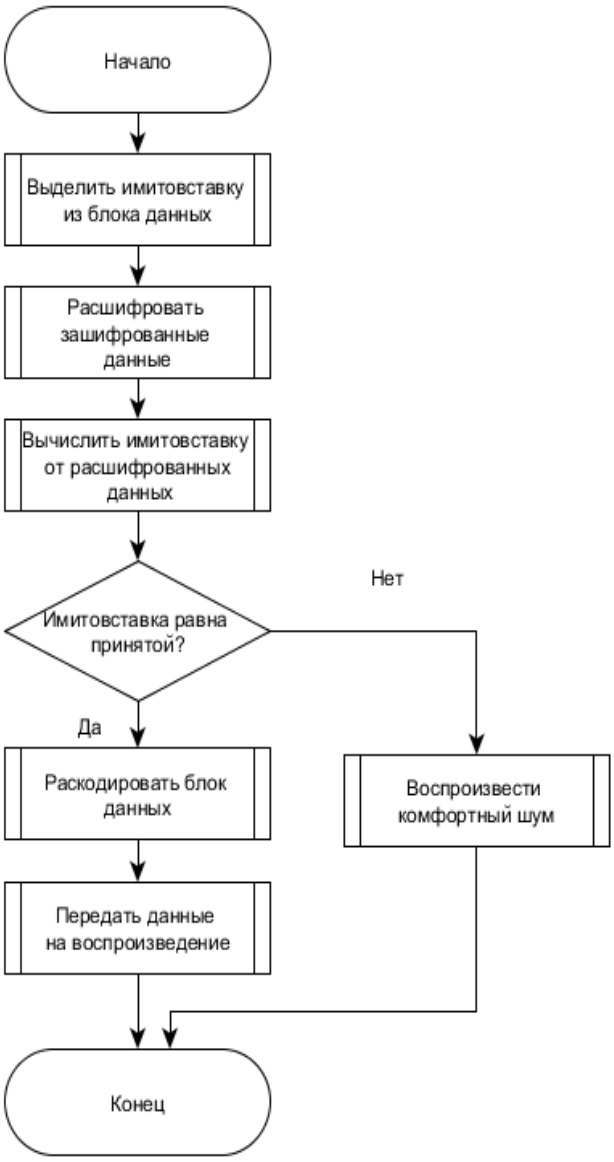

Рисунок 17 — Алгоритм работы модуля зашифрования и расшифрования закодированной речевой информации при приёме речевой информации

При включённом кодеке SPEEX зашифрованию подвергается один блок данных размером 46 байт, далее для него вычисляется имитовставка размером 32 бита и выполняется передача зашифрованного блока и имитовставки в канал связи.

При включённом кодеке G.711 шифруется один блок данных размером 160 байт, далее для него вычисляется имитовставка размером 32 бита и выполняется передача зашифрованного блока и имитовставки в канал связи.

При приёме от принятых данных отделяется имитовставка, остаток расшифровывается, от результата расшифрования вычисляется имитовставка и сравнивается с принятой, если имитовставки совпадают, то расшифрованные данные передаются на воспроизведение, иначе воспроизводится комфортный шум.

Было выполнено написание исходного кода модуля и проведена его отладка.

В результате был реализован модуль, в котором выполняется зашифрование и расшифрование закодированной речевой информации по алгоритму ГОСТ Р 34.12-2015 («Магма»). Также было реализовано считывание ключей шифрования с подключенной к макету SD-карты. С помощью функций данного модуля был дополнен модуль считывания и воспроизведения речевой информации для обеспечения зашифрования считываемой и расшифрования воспроизводимой речевой информации.

Для обеспечения возможности взаимодействия макета с SD-картой с помощью программы STM32CubeMX в проект были добавлены функции для работы с SD-картой в режиме SDIO 1 bit.

Для реализации модуля были разработаны функции, выполняющие подготовку модуля к работе. Разработанные функции перечислены в таблице 9.

Таблица 9 — Функции модуля зашифрования и расшифрования

Имя функции	Описание функции
`read_key`	Считывание ключей шифрования с SD-карты
`write_key`	Запись ключей шифрования на SD-карту
`init_crypto`	Инициализация контекста шифрования

Далее была выполнена модификация функций `init_audio`, `in_irq_handler` и `out_irq_handler` в файле `broker.c`.

В функцию `init_audio` была добавлена функция `init_crypto`.

В функции `in_irq_handler` и `out_irq_handler` было добавлено соответственно зашифрование вводимой информации и расшифрование выводимой.

Предусмотрена возможность включения и отключения шифрования путём определения константы `ENCRYPTION` в файле `broker.h`.

9.5 Разработка модуля приёма/передачи зашифрованной речевой информации

Задача модуля приёма/передачи зашифрованной речевой информации состоит в реализации подключения макета к сети Интернет посредством формирования, отправки, приёма и анализа пакетов стека протоколов TCP/IP и реализации возможностей SIP-клиента по приёму и отправке пакетов протоколов RTP и SIP для выполнения функций макета.

В макете используется USB-модем Huawei E3372 в режиме Hi-Link, который выполняет обмен данными с USB-хостом по протоколу RNDIS. Следовательно для реализации модуля необходима реализация протоколов стека TCP/IP, Ethernet и RNDIS, также должен быть реализован обмен данными с 3G/4G USB-модемом.

На рисунке 18 представлены алгоритмы работы модуля приёма/передачи зашифрованной речевой информации при приёме и передаче пакетов.

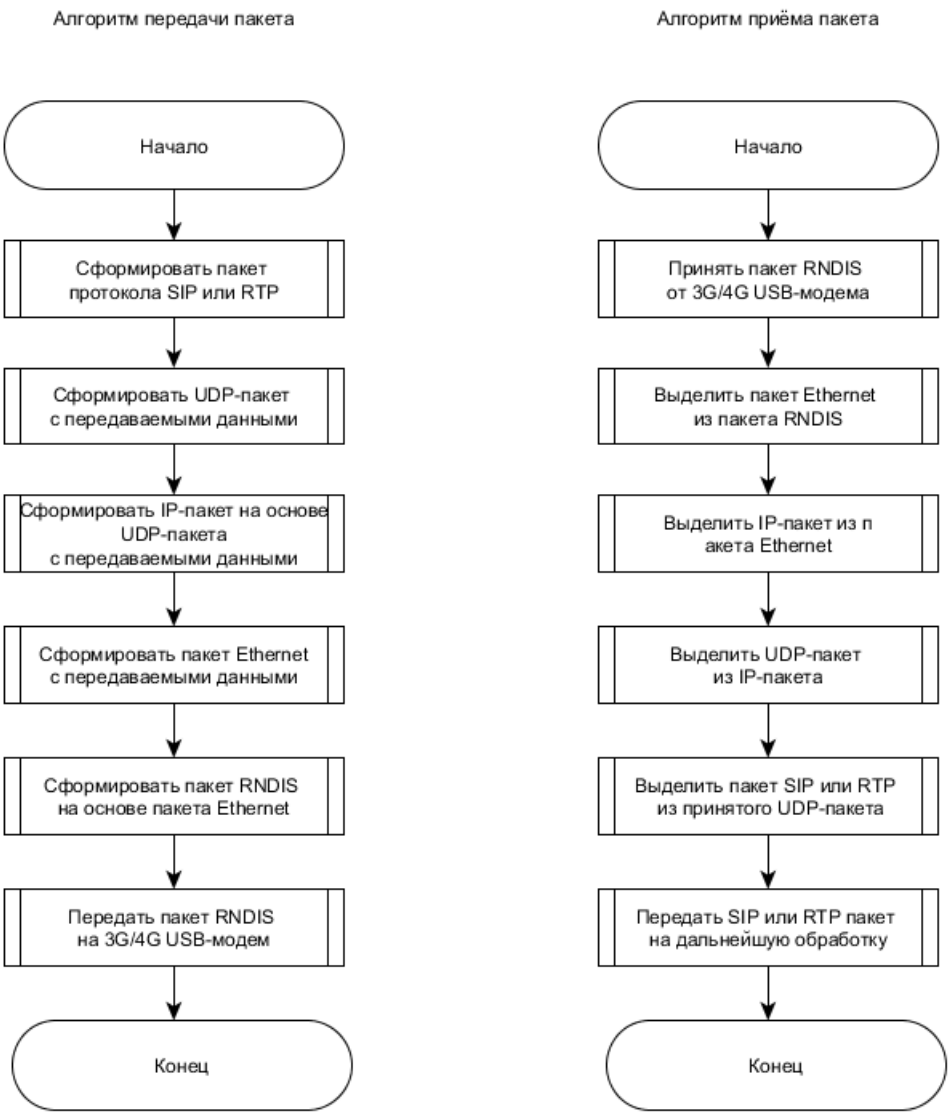

Рисунок 18 — Алгоритмы работы модуля приёма/передачи зашифрованной речевой информации при приёме и передаче пакетов.

Было выполнено написание исходного кода модуля и проведена его отладка.

В результате был реализован модуль, с помощью которого возможен приём и отправка SIP и RTP-пакетов с помощью 3G/4G USB-модема Huawei E3372. Также реализованы возможности SIP-клиента по по регистрации на общедоступном SIP-сервере,совершению и приёму вызовов, приёму и передаче RTP-трафика в рамках вызова, для чего также был дополнен модуль считывания и воспроизведения речевой информации.

Первоначально были реализованы возможности USB-хоста по взаимодействию с 3G/4G USB-модемом. В качестве основы для разработки была взята билблиотека `STM32_USB_Host_Library`[23], которая была скорректирована в соответствии с особенностями работы 3G/4G USB-модема. В результате был выделен специальный класс USB-устройств для модема и были разработаны функции, необходимые для обмена данными с устройством.

Далее была разработана реализация протокола RNDIS в соответствии со спецификацией протокола [27], за основу был взят проект с открытым исходным кодом `lrndis` [28], в котором был реализован протокол RNDIS для работы на USB устройстве.

В рамках реализации протокола RNDIS были разработаны функции, представленные в таблице 10.

Таблица 10 — Фнукции реализации протокола RNDIS

Имя функции	Описание функции
`RNDIS_Init`	отправка сообщения RNDIS Init для начала процесса инициализации
`RNDIS_Query`	запрос определённого параметра устройства (OID)
`RNDIS_Set`	отправка команды установки определённого параметра устройства
`RNDIS_SendControl`	отправка управляющего сообщения RNDIS
`RNDIS_GetResponce`	запрос ответа на отправленное ранее управляющее сообщение
`RNDIS_ResponceAvailable`	запрос сообщения о готовности ответа к отправке
`RNDIS_Initialization_Sequence`	функция, выполняющая последовательность запросов к устройству для его инициализации
`RNDIS_AwaitResponce`	вспомогательная функция, объединяющая запрос готовности ответа и запрос самого ответа
`RNDIS_Process`	основная функция, которая должна вызываться в основном цикле программы для выполнения инициализации модема, отправки сообщений из очереди отправки и приёма сообщений
`RNDIS_TransmitPacket`	отправка пакета RNDIS Data Packet (передача пакета функции отправки USB)
`RNDIS_ReceivePacket`	приём пакета RNDIS Data Packet (получение пакета от функции приёма USB)

Продолжение таблицы 10

Имя функции	Описание функции
`RNDIS_PutInQueue`	постановка пакета в очередь на отправку — эта функция вызывается при необходимости отправить пакет
`RNDIS_OutFromQueue`	удаление отправленного пакета из очереди
`RNDIS_GetReadyToReceive`	вспомогательная функция, готовящая программу к приёму очередного пакета данных

Очередь отправки пакетов была реализована в целях защиты от потери пакетов при различной скорости отправки и формирования пакетов. Также в рамках очереди отправки пакетов была реализована возможность фрагментации пакетов, чья длина превышает пропускную способность High Speed USB.

Далее было выполнено подключение стека реализации стека протоколов TCP/IP и Ethernet. За основу была взята реализация стека TCP/IP для мобильных платформ LWIP[25] и библиотека интерфейса Ethernet[23].

Интерфейсу макетной платы был назначен IPv4 адрес 192.168.8.2/24.

Далее была выполнена коррекция функций, реализующих передачу пакетов Ethernet в канал связи и их приём `ethernetif_init` и `ethernetif_output`. Для этого были разработаны представленные в таблице 11 функции.

Таблица 11 — Функции, обеспечивающие передачу Ethernet пакетов RNDIS

Имя функции	Описание функции
`usb_low_level_input`	Приём пакета RNDIS и выделение из него пакета Ethernet
`usb_low_level_output`	Передача пакета Ethernet RNDIS

Заключительный шаг разработки модуля — реализация SIP-клиента. Для работы макета было решено использовать общедоступный SIP-сервер ekiga.net, на котором было создано две тестовых учётных записи sip:841848@ekiga.net и sip:841848@ekiga.net.

За основу был взят и доработан SIP-клиент с открытым исходным кодом[29].

В результате в клиенте была реализована возможность авторизации на SIP-сервере путём разработки функции `send_sip_authorized`, в которой выполняется расчёт ответа на запрос HTTP Digest аутентификации с помощью реализации хеш-функции MD5[30] (функция `create_responce`).

Также была реализована передача пакетов от SIP-клиента стеку TCP/IP и их обратный приём. Для этого были разработаны функции, приведённые в таблице 12.

Таблица 12 -Функции взаимодействия между SIP-клиентом и стеком TCP/IP

Имя функции	Описание функции
`init_pcb`	инициализация структур, необходимых для отправки и приёма UDP-пакетов
`send_rtp`	сформировать и отправить UDP-пакет на основе RTP-пакета
`send_sip`	сформировать и отправить UDP-пакет на основе SIPпакета
`sip_receive_callback`	передать на обработку принятый SIP-пакет
`rtp_receive_callback`	передать на обработку принятый RTP-пакет
`RTP_PutInQueue`	поместить принятый RTP-пакет в очередь обработки
`RTP_OutFromQueue`	удалить обработанный пакет из очереди обработки

Очередь обработки принятых RTP-пакетов была разработана для обеспечения последовательной обработки принятых пакетов и избежания возможных потерь пакетов.

Также для формирования связи между считыванием звука и формированием RTP-пакетов были дополнены функции `in_irq_handler` и `out_irq_handler` файла `broker.c`.

9.6 Разработка модуля, реализующего пользовательский интерфейс макета

Задача модуля — обеспечить пользователю возможность управления макетом, а именно совершения и приёма вызовов, отклонения входящих

вызовов, отмены исходящий, сброса текущих вызовов и набора номера вызываемого абонента.

Для вывода сообщений пользователю было решено использовать LCD-экран макетной платы, а для ввода команд пользователей — сенсорный экран макетной платы.

В соответствии с указанными возможностями были определены состояния интерфейса, перечисленные в таблице 13.

Таблица 13 — Перечень состояний пользовательского интерфейса

Состояние	Описание состояния
«ожидание вызова»	В данном состоянии возможно начать набирать номер вызываемого абонента и перейти в состояние «набор номера» и вывести краткую информацию о макете
«набор номера»	В данном состоянии возможен набор номера, вызов набранного номера — переход в состояние «установка соединения», возврат в состояние «ожидание вызова», а также переход в состояние «входящий вызов» при поступлении входящего вызова
«установка соединения»	В данном состоянии происходит ожидание приёма вызова вызываемым абонентом, далее произойдёт переход в состояние «соединение установлено», также возможен сброс вызова и переход в состояние «ожидание вызова», также в случае, если введён неверный номер или SIP-сервер по тем или иным причинам не может установить соединение, происходит переход в состояние «сообщение о невозможности установки соединения»
«соединение установлено»	В данном состоянии происходит обмен аудио-данными с вызываемым абонентом, возможен сброс вызова и переход в состояние «вызов завершён»
«входящий вызов»	В данном состоянии пользователю сообщается о поступившем входящем вызове и предоставляется возможность принять вызов и совершить переход в состояние «соединение установлено» или сбросить вызов и перейти в состояние «ожидание вызова»
«сообщение о невозможности установки соединения»	В данном состоянии пользователю сообщается о невозможности установки соединения, после чего выполняется переход в состояние «ожидание вызова»
«вызов завершён»	В данном состоянии пользователю сообщается о завершении вызова, после чего выполняется переход в состояние «ожидание вызова»

Также была разработана диаграмма состояний интерфейса. Диаграмма представлена на рисунке 19.

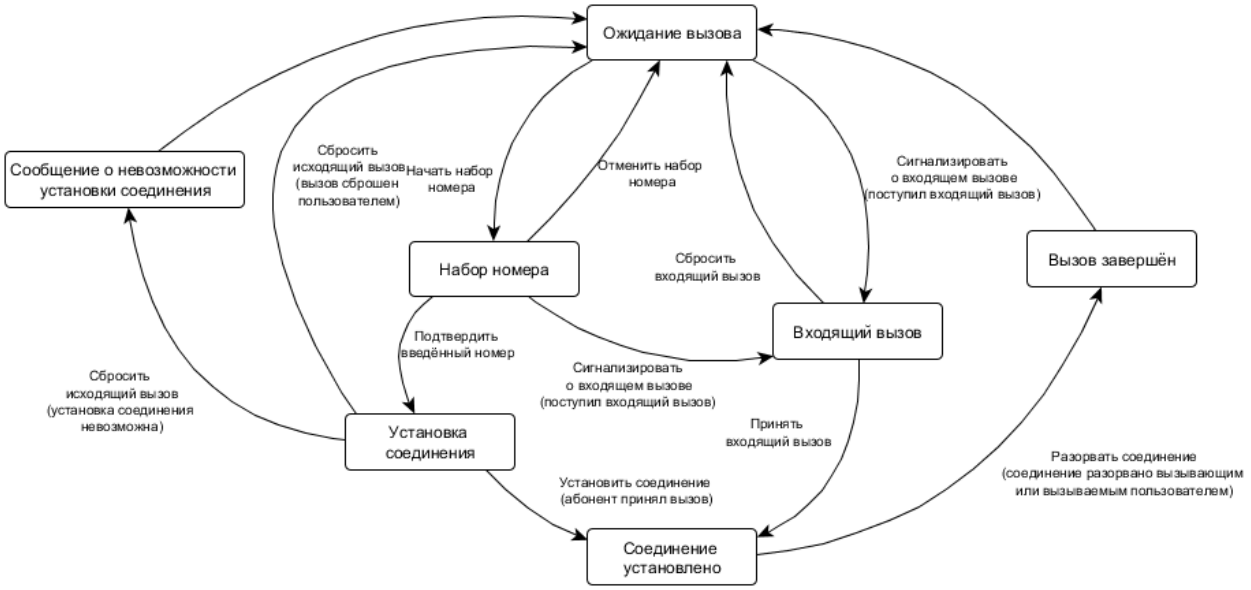

Рисунок 19 — Диаграмма состояний интерфейса

Было выполнено написание исходного кода модуля и проведена его отладка.

В результате был реализован модуль, выполняющий функции пользовательского интерфейса макета, с помощью которого возможно управление совершаемыми и принимаемыми вызовами и набор номера вызываемого абонента.

В ходе разработки модуля были разработаны функции, приведённые в таблице 14.

Таблица 14 — Функции пользовательского интерфейса

Имя функции	Описание функции
`UI_Init`	Инициализация LCD, калибровка сенсорного экрана
`MENU_controller`	Основная функция, определяющая предоставляемые интерфейсом возможности в зависимости от состояния интерфейса
`MAIN_menu`	Реализует состояние интерфейса «ожидание вызова»
`ADDRESS_INPUT_menu`	Реализует состояние инерфейса «набор номера»
`CALLING_menu`	Реализует состояние интерфейса «установка соединения»
`ESTABLISHED_menu`	Реализует состояние интерфейса «соединение установлено»
`INCOMING_CALL_menu`	Реализует состояние интерфейса «входящий вызов»

Продолжение таблицы 14

Имя функции	Описание функции
ABOUT_menu	Реализует показ краткой информации о макете в состоянии «ожидание вызова»
ERROR_menu	Реализует состояние интерфейса «сообщение о невозможности установки соединения»
TERMINATE_menu	Реализует состояние интерфейса «вызов завершён»
set_INCOMING_CALL	Устанавливает состояние интерфейса «входящий вызов»
set_ESTABLISHED	Устанавливает состояние интерфейса «соединение установлено»
set_ERROR	Устанавливает состояние интерфейса «сообщение о невозможности установки соединения»
set_TERMINATE	Устанавливает состояние интерфейса «вызов завершён»
draw_telephone	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «набор номера»
draw_main	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «ожидание вызова»
draw_about	Отображает на LCD-экране краткое описание макета
draw_calling	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «установка соединения»
draw_established	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «соединение установлено»
draw_incoming_call	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «входящий вызов»
draw_error	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «сообщение о невозможности установки соединения»
draw_terminate	Отображает на LCD-экране элементы, необходимые для реализации состояния интерфейса «вызов завершён»
digit_highlight	Вспомогательная функция, выполняющая подсветку набираемых цифр на LCD-экране

Также для реализации интерфейса была разработана реализация шрифта, содержащего символы кириллического алфавита.

Ниже приведены схематические изображения интерфейса пользователя в различных состояниях.

На рисунке 20 приведено состояние интерфейса в момент калибровки сенсорного экрана.

Рисунок 20 — Калибровка сенсорного экрана

На рисунке 21 приведён вид пользовательского интерфейса в состоянии «ожидание вызова».

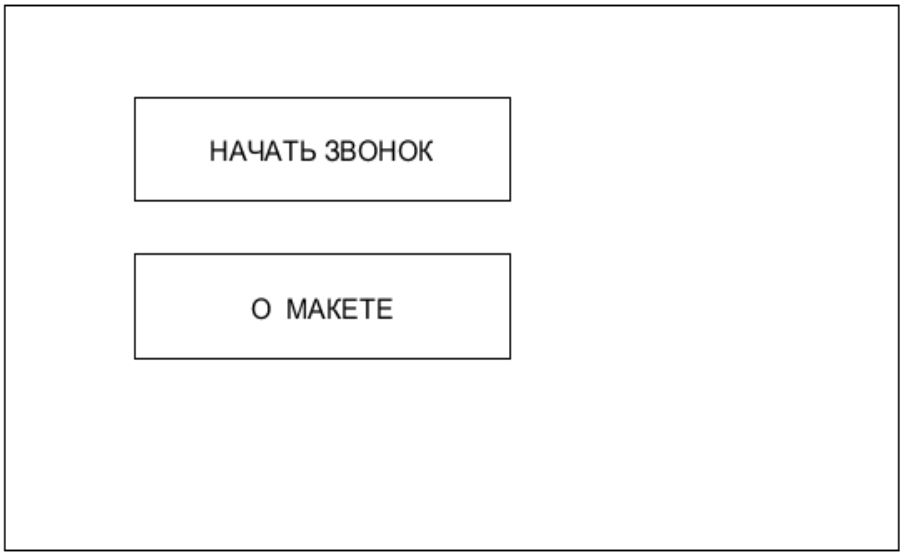

Рисунок 21 — Пользовательский интерфейс в состоянии «ожидание вызова»

На рисунке 22 приведён вид пользовательского интерфейса в состоянии «набор номера».

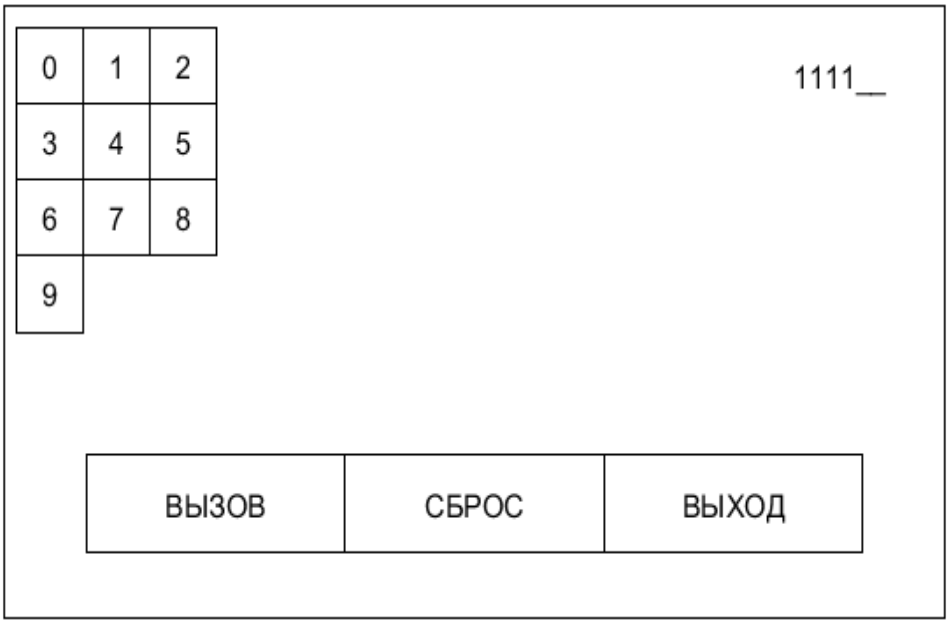

Рисунок 22 — Пользовательский интерфейс в состоянии «набор номера»

На рисунке 23 приведён вид пользовательского интерфейса в состоянии «установка соединения».

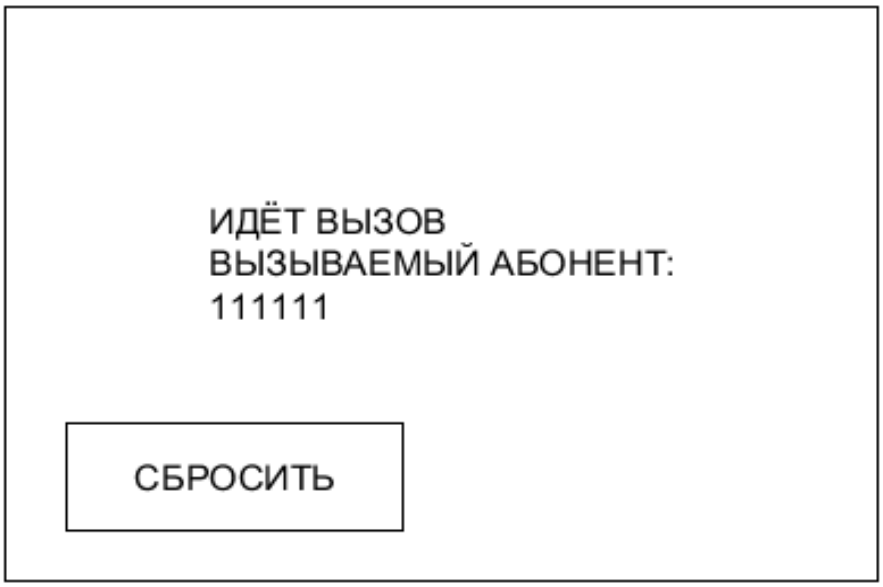

Рисунок 23 — Пользовательский интерфейс в состоянии «установка соединения»

На рисунке 24 изображён вид пользовательского интерфейса в состоянии «входящий вызов».

Рисунок 24 — Пользовательский интерфейс в состоянии «входящий вызов»

На рисунке 25 изображён вид пользовательского интерфейса в состоянии «соединение установлено».

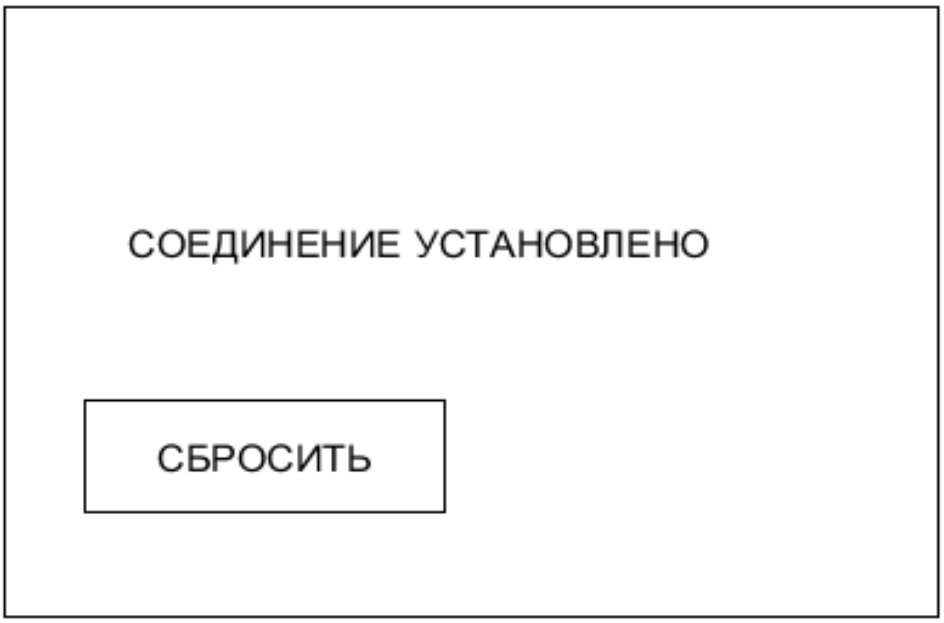

Рисунок 25 — Пользовательский интерфейс в состоянии «соединение установлено»

На рисунке 26 изображён вид пользовательского интерфейса в состоянии «сообщение о невозможности установки соединения».

Рисунок 26 — Пользовательский интерфейс в состоянии «сообщение о невозможности установки соединения»

На рисунке 27 изображён вид пользовательского интерфейса в состоянии «вызов завершён»

Рисунок 27 — Пользовательский интерфейс в состоянии «Вызов завершён»

9.7 Интеграция разработанных модулей

Последний шаг формирования работоспособного программного обеспечения макета — объединение разработанных ранее модулей.

Главный модуль — в нём определена функция main, где вызываются функции инициализации аппаратного обеспечения макетной платы. К нему подключен модуль считывания и воспроизведения речевой информации для активации АЦП и ЦАП I^2S. Модуль, реализующий пользовательский интерфейс макета также подключен к главному модулю для активации пользовательского интерфейса и периодического вызова функции, контролирующей текущее состояние пользовательского интерфейса. Также к главному модулю подключен SIP-клиент для выполненя инициализаци, модуль, реализующий USB-хост и реализация RNDIS для организации приёма и передачи сообений путём периодического вызова функций контроля состояния приёма и передачи сообщений.

Модули кодирования и раскодирования и зашифрования и расшифрования речевой информации подключены к модулю считывания и воспроизведения речевой информации для организации соответственно кодирования и зашифрования во время считывания и расшифрования и раскодирования во время воспроизведения речевой информации.

К модулю приёма/передачи зашифрованной речевой информации, а конкретно SIP-клиенту подключен модуль, реализующий считывание и воспроизведение речевой информации, так как считывание и воспроизведение запускается и прекращается в зависимости от состояния вызова, изменяемого SIP-клиентом. Также подключен модуль, реализующий пользовательский интерфейс макета так как изменения состояния интерфейса могут вызвать изменение состояния вызова, так же как и изменение состояния вызова может вызвать изменение состояния пользовательского интерфейса.

На рисунке 28 приведена схема объединения модулей в проекте.

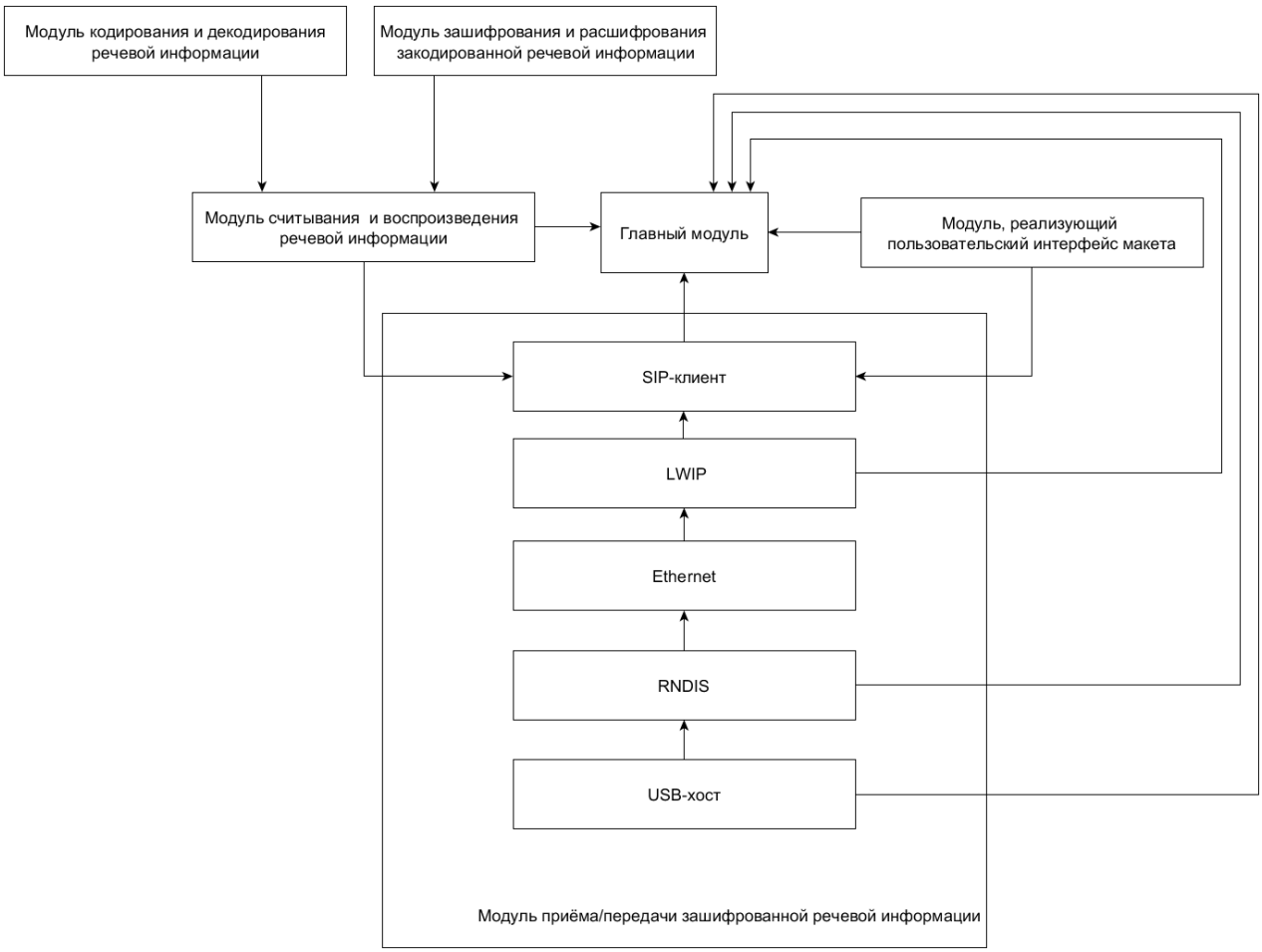

Рисунок 28 — Схема объединения модулей

10 Тестирование программного обеспечения макета

10.1 Функциональное тестирование

После разработки и отладки было проведено функциональное тестирование программного обеспечения макета с целью подтверждения выполнения им заявленных в техническом задании на выпускную квалификационную работу функций.

Были выполнены следующие тесты:

— совершение вызова устройства, не являющегося макетом, при отключенном шифровании;

— приём вызова от устройства, не являющегося макетом, при отключенном шифровании;

— совершение вызова аналогичного макета с приёмом вызова;

— совершение вызова аналогичного макета с отменой вызова с вызывающей стороны;

— совершение вызова аналогичного макета со сбросом вызова с вызываемой стороны;

— совершение вызова абонента в состоянии оффлайн;

— совершение вызова несуществующего абонента.

Каждый тест был выполнен как при активном кодеке SPEEX, так и G.711

В таблице 15 приведены результаты функционального тестирования.

Таблица 15 — Результаты функционального тестирования

Тест	Кодек	Вызов	Передача речи	Завершение вызова	Тест пройден?
Совершение вызова устройства, не являющегося макетом, при отключенном шифровании	SPEEX	Произошёл	Произошла	Успешно	Да
	G.711	Произошёл	Произошла	Успешно	Да
Приём вызова от устройства, не являющегося макетом, при отключенном шифровании	SPEEX	Произошёл	Произошла	Успешно	Да
	G.711	Произошёл	Произошла	Успешно	Да

Продолжение таблицы 15

Тест	Кодек	Вызов	Передача речи	Завершение вызова	Тест пройден?
Совершение вызова аналогичного макета с приёмом вызова	SPEEX	Произошёл	Произошла	-	Да
	G.711	Произошёл	Произошла	-	Да
Совершение вызова аналогичного макета с отменой вызова с вызывающей стороны	SPEEX	Произошёл	Произошла	Успешно	Да
	G.711	Произошёл	Произошла	Успешно	Да
Совершение вызова аналогичного макета со сбросом вызова с вызываемой стороны	SPEEX	Произошёл	Произошла	Успешно	Да
	G.711	Произошёл	Произошла	Успешно	Да
Совершение вызова абонента в состоянии оффлайн	SPEEX	Не произошёл	-	-	Да
	G.711	Не произошёл	-	-	Да
Совершение вызова несуществующего абонента	SPEEX	Не произошёл	-	-	Да
	G.711	Не произошёл	-	-	Да

Все заявленные тесты были успешно пройдены, что подтверждает способность программного обеспечения выполнять свои функции, а именно: принимать вызовы от аналогичных макетов, совершать вызовы аналогичных макетов, сбрасывать вызовы и предоставлять возможность набора номера вызываемого устройства.

10.2 Измерение разборчивости воспроизводимой речи артикуляционным методом

Также было проведено измерение разборчивости передаваемой макетом речи с помощью методики, приведённой в ГОСТ Р 50840-95 [31].

Для этого было проведено зачитывание пяти артикуляционных слоговых таблиц [31] с передачей речи в канал связи и прослушивание принятой в результате речи с записью принятых слогов.

После чего была оценена средняя разборчивость речи для каждой таблицы по следующей формуле:

$$S_k = \frac{1}{N} \sum_{i=1}^{N} S_i \quad , \tag{1}$$

где S_k – результат единичного измерения в процентах, N – число единичных измерений, S_i – результат разбора одного слога таблицы — 1 — разобран верно, 0 — разобран неверно.

Далее было вычислено среднее значение разборчивости для всех пяти таблиц по следующей формуле:

$$S = \frac{1}{5} \sum_{k=1}^{5} S_k \quad , \tag{2}$$

где S – итоговый результат измерений в процентах, S_k – результат единичного измерения в процентах.

Измерения проводились для кодека SPEEX и G.711 по отдельности.

В таблицах А.1, А.2, А.3, А.4 и А.5 приведены результаты прослушивания артикуляционных слоговых таблиц 1, 2, 3, 4 и 5 при активном кодеке SPEEX.

В таблице 16 приведены результаты единичных измерений, рассчитанные по формуле 1 и итоговый результат, рассчитанный по формуле 2.

Таблица 16 — Результаты единичных измерений для кодека SPEEX

Измерение	S_1 %	S_2 %	S_3 %	S_4 %	S_5 %
Результат	72	82	74	66	74
S %	73,6				

Таким образом итоговое значение разборчивости речи при активном кодеке SPEEX равняется 73,6%, что относится к 1 классу качества — понимание передаваемой речи без затруднений [31].

Аналогичные измерения были проведены при активном кодеке G.711.

В таблицах А.6, А.7, А.8, А.9 и А.10 приведены результаты прослушивания артикуляционных слоговых таблиц 1, 2, 3, 4 и 5 при активном кодеке G.711.

В таблице 17 приведены результаты единичных измерений, рассчитанные по формуле 1 и итоговый результат, рассчитанный по формуле 2.

Таблица 17 — Результаты единичных измерений для кодека G.711

Измерение	S_1 %	S_2 %	S_3 %	S_4 %	S_5 %
Результат	90	96	90	92	94
S %	92,4				

Таким образом итоговое значение разборчивости речи при активном кодеке G.711 равняется 92,4%, что относится к высшему классу качества — понимание передаваемой речи без малейшего напряжения внимания [31].

На рисунке 29 приведено диаграмма сравнения результатов единичных измерений всех пяти таблиц для кодеков SPEEX и G.711.

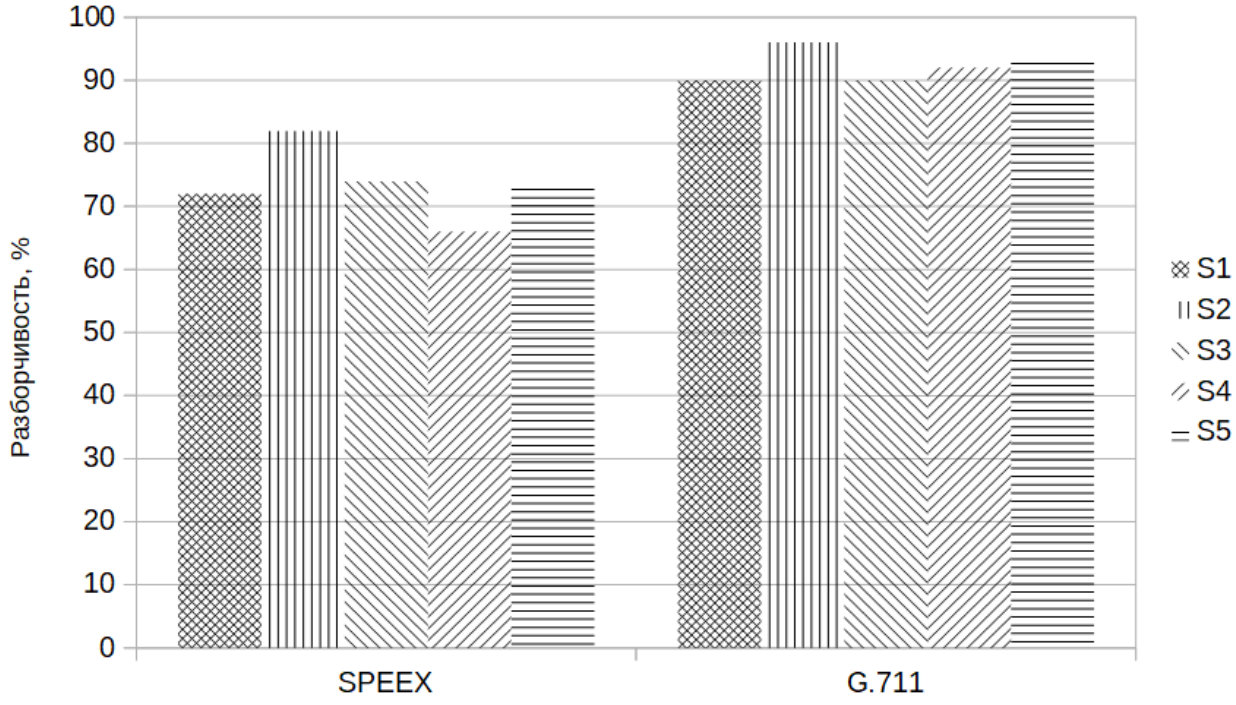

Рисунок 29 — Диаграмма сравнения результатов единичных измерений всех пяти таблиц для кодеков SPEEX и G.711

В результате тестирования было выяснено, что кодек G.711 обеспечивает лучшее качество передаваемой речи чем SPEEX, однако его реализация требует

большей полосы пропускания канала связи, что приводит к задержкам воспроизведения, которые однако не оказывают значительного влияния на качество воспроизведения.

11 Вредные психофизиологические факторы, влияющие на опорно-двигательный аппарат специалиста по защите информации. Методы и средства физической культуры, снижающие их воздействие

Согласно ГОСТ 12.0.003-2015 опасные и вредные производственные факторы, обладающие свойствами психофизиологического воздействия на организм человека, подразделяются на физические перегрузки, связанные с тяжестью трудового процесса, и нервно-психические перегрузки, связанные с напряжённостью трудового процесса. [32]

Вредные психологические факторы, влияющие на опорно-двигательный аппарат специалиста по защите информации относятся к физическим нагрузкам, связанным с тяжестью трудового процесса.

Физические перегрузки, связанные с тяжестью трудового процесса подразделяются на [32]:

— статические, связанные с рабочей позой;

— динамические нагрузки, связанные с массой поднимаемого и перемещаемого вручную груза;

— динамические нагрузки, связанные с повторением стереотипных рабочих движений.

Работа специалиста по защите информации в большинстве случаев не связана с чрезмерными динамическими нагрузками. Зачастую выполнение трудовых обязанностей специалиста по защите информации связано с нахождением на сидячем рабочем месте в постоянной позе, что приводит к малоподвижному образу жизни.

Малоподвижный образ жизни — причина развития у специалиста по защите информации гиподинамии.

Гиподинамия — нарушение функций организма (опорно-двигательного аппарата, кровообращения, дыхания, пищеварения) при ограничении двигательной активности, снижении силы сокращения мышц. [33]

Гиподинамия оказывает влияние на все системы организма. Для опорно-двигательного аппарата последствия гиподинамии проявляются в виде ослабевания и последующей атрофии мышц, искривлении позвоночника, заболеваниях суставов, что в дальнейшем может нанести ещё больший урон всему организму человека.

Для профилактики гиподинамии необходима регулярная ежедневная физическая активность. В качестве такой активности могут выступать любые физические упражнения, наиболее просто реализуемыми среди них являются следующие методы и средства физической культуры:

— пешие прогулки;
— скандинаская ходьба;
— бег;
— плавание;
— занятия фитнесом;
— езда на велосипеде.

Заключение

В ходе выполнения выпускной квалификационной работы был выполнены следующие этапы:

— обзор существующих аудио-кодеков, механизмов передачи речевой информации, протоколов передачи речевой информации, отечественных криптографических алгоритмов, способов ввода ключевой информации в мобильной аппаратуре и выбор соответствующих требованиям технического задания на выпускную квалификационную работу;

— анализ структуры макета с определением трактов прохождения речевой информации;

— разработка алгоритма работы программного обеспечения макета;

— разработка модулей программного обеспечения макета и их интеграция;

— функциональное тестирование программного обеспечения макета;

— измерение разборчивости передаваемой макетом речи артикуляционным методом.

По результатам тестирования было определено, что разработанное программное обеспечение макета соответствует требованиям, предъявляемым техническим заданием на выпускную квалификационную работу. По результатам измерения разборчивости воспроизводимой макетом речи было определено, что качество работы макета также соответствует требуемому.

На основании чего можно сделать вывод, что задание на выпускную квалификационную работу выполнено в полном объёме.

Список использованной литературы

1 Словарь «Академик» [электронный ресурс] - https://dic.academic.ru/dic.nsf/ruwiki/793945#.D0.A1.D0.BC._.D1.82.D0.B0.D0.BA.D0.B6.D0.B5 – режим доступа — свободный — дата последнего обращения 23.02.2019

2 Википедия Свободная энциклопедия [электронный ресурс] - https://ru.wikipedia.org/wiki/MP3 – режим доступа — свободный — дата последнего обращения — 23.02.2019

3 MPEG для чайников [электронный ресурс] http://www.codenet.ru/progr/formt/mpeg4all.php – режим доступа — свободный — дата последнего обращения — 23.02.2019

4 Формат AAC [электронный ресурс] - https://audiocoding.ru/formats/aac/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

5 Формат OGG Vorbis Audio [электронный ресурс] - https://audiocoding.ru/formats/ogg-vorbis-audio/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

6 Формат FLAC [электронный ресурс] - https://audiocoding.ru/formats/flac/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

7 Как работает радиоинтерфейс в GSM-сетях [электронный ресурс] - https://habr.com/ru/post/268127/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

8 Формат AMR [электронный ресурс] - https://audiocoding.ru/formats/amr/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

9 Компоненты H.323. Рекомендация ITU-T G.723.1 [электронный ресурс] - http://iptop.net/tech/g7231a.html – режим доступа — свободный — дата последнего обращения — 23.02.2019

10 Рекомендация МСЭ-Т G.723.1 [электронный ресурс] - https://www.itu.int/rec/T-REC-G.723.1-200605-I/en – режим доступа — свободный — дата последнего обращения — 23.02.2019

11 VOIP кодеки — подробное описание и характерисики [электронный ресурс] - https://wiki.merionet.ru/ip-telephoniya/5/voip-codecs/ – режим доступа — свободный — дата последнего обращения — 23.02.2019

12 Формат Speex [электронный ресурс] - https://audiocoding.ru/formats/speex/ — режим доступа — свободный — дата последнего обращения — 23.02.2019

13 База знаний Asterisk [электронный ресурс] - http://asterisk.ru/knowledgebase/ITU+G.711 – режим доступа — свободный — дата последнего обращения — 23.02.2019

14 Портал о современных технологиях мобильной и беспроводной связи [электронный ресурс] - http://1234g.ru — режим доступа — свободный — дата последнего обращения — 26.02.2019

15 Безопасность GSM сетей: шифрование данных [электронный ресурс] - https://habr.com/ru/post/186838/ — режим доступа — свободный — дата последнего обращения — 26.02.2019

16 Основы глобальной сети Internet. VoIP или IP-телефония -это основной инструмент бизнес — коммуникаций [электронный ресурс] - https://www.lessons-tva.info/edu/trainbus/1_1.html — режим доступа — свободный — дата последнего обращения — 26.02.2019

17 Описание RFC пртокола SIP [электронный ресурс] - https://wiki.dieg.info/sip — режим доступа — свободный — дата последнего обращения — 23.02.2019

18 Протокол SDP [электронный ресурс] - https://wiki.dieg.info/sdp — режим доступа — свободный — дата последнего обращения — 23.02.2019

19 RTP и RTCP протоколы для IP-телефонии [электронный ресурс] - https://compress.ru/article.aspx?id=11931#n_3_1 — режим доступа — свободный — дата последнего обращения — 23.02.2019

20 М. Ковцур Протоколы обеспечения безопасности IP-телефонии [электронный ресурс] - http://www.lastmile.su /files/article_pdf/3/article_3445_57.pdf — режим доступа — свободный — дата последнего обращения — 23.02.2019

21 ГОСТ 28147-89 Системы обработки информации. Защита криптографическая Алгоритм криптографического преобразования [электронный ресурс] - http://docs.cntd.ru/document/gost-28147-89 — режим доступа — свободный — дата последнего обращения — 23.02.2019

22 В ГОСТе сидел «Кузнечик» [электронный ресурс] - https://habr.com/ru/post/266359/ — режим доступа — свободный — дата последнего обращения — 23.02.2019

23 Библиотека STM Cube F4 STM32Cube_FW_F4 [электронный ресурс] - https://github.com/fboris/STM32Cube_FW_F4 - режим доступа свободный — дата последнего обращения — 02.06.2019

24 Библиотека кодека SPEEX [электронный ресурс] - https://www.speex.org/downloads/ - режим доступа свободный — дата последнего обращения — 02.06.2019

25 Библиотека кодека G.711 [электронный ресурс] - http://web.mit.edu/audio/src/build/i386_linux2/sox-11gamma-cb/g711.c - режим доступа свободный — дата последнего обращения — 02.06.2019

26 Реализация алгоритма шифрования ГОСТ Р 34.12-2015 («Магма») [электронный ресурс] - https://github.com/ygrec/gost_28147 - режим доступа свободный — дата последнего обращения — 02.06.2019

27 Спецификация RNDIS [электронный ресурс] - https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/WinArchive/%5BMS-RNDIS%5D.pdf - режим доступа свободный — дата последнего обращения — 02.06.2019

28 LRNDIS stm32 ethernet over usb (rndis + lwip) [электронный ресурс] - https://github.com/fetisov/lrndis - режим доступа свободный — дата последнего обращения — 02.06.2019

29 Paulo R. P.VoIP Application: trabalho final de qualificação — Imprensa da Universidade Técnica de Lisboa 2007

30 Реализация хеш-функции MD5 [электронный ресурс] - https://gist.github.com/creationix/4710780 - режим доступа свободный — дата последнего обращения — 02.06.2019

31 ГОСТ Р 50840-95 Передача речи по трактам связи. Методы оценки качества разборчивости и узнаваемости — принят и введён в действие Постановлением Госстандарта России от 21.11.95 №579 — М.: Госстандарт России: Издательство стандартов, 1995

32 ГОСТ 12.0.003-2015 Межгосударственный стандарт. Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация — введён в действие Приказом Росстандарта от 09.06. 2016 № 602-ст — М.: Госстандарт России: Издательство стандартов, 2017

33 Гиподинамия. Википедия Свободная энциклопедия [электронный ресурс] - https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%BE%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%8F - режим доступа — свободный — дата последнего обращения — 15.06.2019

Приложение А

(обязательное)

Промежуточные результаты измерения разборчивости речи артикуляционным методом

Таблица А.1 — Результаты прослушивания таблицы 1 для кодека SPEEX

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	няк	няк	11	мюф	мюф	21	фсен	фсен	31	ёк	ёк	41	вох	вох
2	пуль	пуль	12	зош	зош	22	ряй	ряй	32	стял	стал	42	жоф	жоф
3	бущ	буш	13	плись	плись	23	бёр	бёр	33	гыпь	гыпь	43	тэф	тэф
4	мят	мят	14	рыц	рыц	24	чать	шать	34	геф	еф	44	трух	трух
5	свум	свун	15	дум	дум	25	выс	быс	35	бац	бас	45	фек	фек
6	сось	сос	16	нысь	нысь	26	гум	гум	36	трит	трит	46	лефь	леф
7	пиц	пиц	17	лут	лут	27	прюх	рюх	37	шись	шись	47	вик	вик
8	чтал	чтал	18	сич	сищ	28	шеть	шеть	38	деп	деп	48	щёх	щёх
9	дюрь	дюрь	19	вость	ость	29	тют	тют	39	здес	здес	49	кась	кась
10	хач	хащ	20	дыс	тыс	30	ефь	фь	40	торь	торь	50	зех	зех
S_1	72													

Таблица А.2 — Результаты прослушивания таблицы 2 для кодека SPEEX

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	лир	лир	11	дуч	дуч	21	чтел	ште	31	жась	жась	41	лёнь	лёнь
2	нёт	нёт	12	свар	свар	22	лась	лась	32	кеш	кеш	42	маф	маф
3	трен	рен	13	порь	порь	23	хуф	уф	33	пыц	пыц	43	зюс	зюс
4	рась	рась	14	нар	ар	24	песть	песть	34	гуть	хуть	44	чёф	щёф
5	скум	кум	15	зунь	ун	25	вый	вый	35	меш	меш	45	буф	буф
6	цать	цать	16	сул	сул	26	касть	асть	36	бяс	бяс	46	шеп	шеп
7	дян	дян	17	пяф	пяф	27	реть	реть	37	тущ	тущ	47	длаф	лаф
8	стоц	стоц	18	золь	золь	28	ряс	ряс	38	дес	дес	48	тёр	тёр
9	прыл	прыл	19	сель	сель	29	фыл	фыл	39	сых	сых	49	непь	непь
10	кром	кром	20	хил	хил	30	рут	рут	40	яс	яс	50	шоль	шоль
S_2	82													

Таблица А.3 — Результаты прослушивания таблицы 3 для кодека SPEEX

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	рон	рон	11	фсес	сес	21	юц	ютс	31	люч	люч	41	куй	куй
2	няш	няш	12	чумь	чумь	22	пел	пел	32	дыст	ыст	42	зич	зич
3	цый	сый	13	щал	щал	23	тряпь	ряпь	33	пуч	пуч	43	жик	вжик
4	репь	репь	14	крыс	крыс	24	тёщ	тёщ	34	хас	хас	44	гом	гом
5	шох	шох	15	вяф	вяф	25	мыл	мыл	35	нип	нип	45	прас	прас
6	гипь	гипь	16	фай	фай	26	зах	зах	36	штись	тис	46	боть	боть
7	сей	сей	17	рир	рир	27	ность	ость	37	тум	тум	47	шарь	шарь
8	фсоф	соф	18	самь	самь	28	дёч	дёч	38	вяй	вяй	48	нунь	унь
9	тас	тас	19	руфь	руфь	29	тац	тац	39	ныф	ныф	49	длин	лин
10	лер	лер	20	дас	тас	30	прит	прит	40	мям	мям	50	вафь	вавь
S_3	74													

Таблица А.4 — Результаты прослушивания таблицы 4 для кодека SPEEX

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	зуфь	суфь	11	бюн	бюн	21	цуч	суч	31	плык	лык	41	сёф	сёф
2	стыч	тыч	12	холь	холь	22	кик	кик	32	фуч	фуч	42	кать	кать
3	жен	жен	13	дяф	тяф	23	моть	оть	33	чтэт	штет	43	дор	дор
4	ер	ер	14	нень	ень	24	дреть	дреть	34	вак	вак	44	пон	пон
5	нярь	нарь	15	штоть	тофь	25	рюх	рюх	35	лен	лен	45	тяч	сяч
6	знун	нун	16	люпь	люпь	26	чесь	чесь	36	тух	тух	46	сусь	сусь
7	гань	гань	17	проц	роц	27	мифь	мифь	37	вур	ур	47	бан	бан
8	кос	кос	18	шич	шич	28	трар	трар	38	зос	зос	48	вень	вень
9	сым	ым	19	пещ	пеш	29	зяк	сяк	39	хет	хет	49	тисть	тисть
10	лоц	лоц	20	щеть	щеть	30	ноть	ноть	40	здап	зда	50	рул	рул
S_4	66													

Таблица А.5 — Результаты прослушивания таблицы 5 для кодека SPEEX

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	рёк	рёк	11	зырь	сырь	21	фок	фок	31	дамь	дамь	41	стар	стар
2	ниш	ниш	12	чтуй	чтуй	22	шусь	шусь	32	пюм	пюм	42	бань	бань
3	зном	ном	13	быс	быс	23	хул	кул	33	кац	кац	43	вюр	вюр
4	рыс	рыс	14	тет	тет	24	штай	штай	34	бёк	бёк	44	нах	нах
5	гек	гек	15	скес	кес	25	мис	мис	35	чинь	шинь	45	бем	бем
6	тань	тань	16	мофь	моф	26	дват	ват	36	щурь	щурь	46	сах	сах
7	вуф	вуф	17	лям	лям	27	трюль	трюль	37	луф	луф	47	десь	тесь
8	трёп	рёп	18	гоф	хоф	28	нуц	нуц	38	фиш	фиш	48	жус	жус
9	мёх	мёх	19	сях	сях	29	рель	рель	39	сузнь	сузнь	49	цой	сой
10	пась	пась	20	роц	роц	30	ший	ший	40	прень	рень	50	лорь	лорь
S_5	74													

Таблица А.6 — Результаты прослушивания таблицы 1 для кодека G.711

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	няк	няк	11	мюф	мюф	21	фсен	фсен	31	ёк	ёк	41	вох	вох
2	пуль	пуль	12	зош	зош	22	ряй	ряй	32	стял	стал	42	жоф	жоф
3	бущ	буш	13	плись	плись	23	бёр	бёр	33	гыпь	гыпь	43	тэф	тэф
4	мят	мят	14	рыц	рыц	24	чать	чать	34	геф	геф	44	трух	трух
5	свум	свум	15	дум	дум	25	выс	быс	35	бац	бас	45	фек	фек
6	сось	сос	16	нысь	нысь	26	гум	гум	36	трит	трит	46	лефь	лефь
7	пиц	пиц	17	лут	лут	27	прюх	рюх	37	шись	шись	47	вик	вик
8	чтал	чтал	18	сич	сищ	28	шеть	шеть	38	деп	деп	48	щёх	щёх
9	дюрь	дюрь	19	вость	вость	29	тют	тют	39	здес	здес	49	кась	кась
10	хач	хач	20	дыс	тыс	30	ефь	ефь	40	торь	торь	50	зех	зех
S_1	90													

Таблица А.7 — Результаты прослушивания таблицы 2 для кодека G.711

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	лир	лир	11	дуч	дуч	21	чтел	чтел	31	жась	жась	41	лёнь	лёнь
2	нёт	нёт	12	свар	свар	22	лась	лась	32	кеш	кеш	42	маф	маф
3	трен	рен	13	порь	порь	23	хуф	хуф	33	пыц	пыц	43	зюс	зюс
4	рась	рась	14	нар	нар	24	песть	песть	34	гуть	хуть	44	чёф	щёф
5	скум	скум	15	зунь	зунь	25	вый	вый	35	меш	меш	45	буф	буф
6	цать	цать	16	сул	сул	26	касть	касть	36	бяс	бяс	46	шеп	шеп
7	дян	дян	17	пяф	пяф	27	реть	реть	37	тущ	тущ	47	длаф	лаф
8	стоц	стоц	18	золь	золь	28	ряс	ряс	38	дес	дес	48	тёр	тёр
9	прыл	прыл	19	сель	сель	29	фыл	фыл	39	сых	сых	49	непь	непь
10	кром	кром	20	хил	хил	30	рут	рут	40	яс	яс	50	шоль	шоль
S_2	96													

Таблица А.8 — Результаты прослушивания таблицы 3 для кодека G.711

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	рон	рон	11	фсес	фсес	21	юц	ютс	31	люч	люч	41	куй	куй
2	няш	няш	12	чумь	чумь	22	пел	пел	32	дыст	дыст	42	зич	зич
3	цый	сый	13	щал	щал	23	тряпь	ряпь	33	пуч	пуч	43	жик	жик
4	репь	репь	14	крыс	крыс	24	тёщ	тёщ	34	хас	хас	44	гом	гом
5	шох	шох	15	вяф	вяф	25	мыл	мыл	35	нип	нип	45	прас	прас
6	гипь	гипь	16	фай	фай	26	зах	зах	36	штись	тис	46	боть	боть
7	сей	сей	17	рир	рир	27	ность	ность	37	тум	тум	47	шарь	шарь
8	фсоф	соф	18	самь	самь	28	дёч	дёч	38	вяй	вяй	48	нунь	нунь
9	тас	тас	19	руфь	руфь	29	тац	тац	39	ныф	ныф	49	длин	длин
10	лер	лер	20	дас	дас	30	прит	прит	40	мям	мям	50	вафь	вавь
S_3	90													

Таблица А.9 — Результаты прослушивания таблицы 4 для кодека G.711

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	зуфь	зуфь	11	бюн	бюн	21	цуч	цуч	31	плык	плык	41	сёф	сёф
2	стыч	тыч	12	холь	холь	22	кик	кик	32	фуч	фуч	42	кать	кать
3	жен	жен	13	дяф	дяф	23	моть	оть	33	чтэт	чтэт	43	дор	дор
4	ер	ер	14	нень	нень	24	дреть	дреть	34	вак	вак	44	пон	пон
5	нярь	нярь	15	штоть	тофь	25	рюх	рюх	35	лен	лен	45	тяч	сяч
6	знун	знун	16	люпь	люпь	26	чесь	чесь	36	тух	тух	46	сусь	сусь
7	гань	гань	17	проц	роц	27	мифь	мифь	37	вур	вур	47	бан	бан
8	кос	кос	18	шич	шич	28	трар	трар	38	зос	зос	48	вень	вень
9	сым	сым	19	пещ	пещ	29	зяк	зяк	39	хет	хет	49	тисть	тисть
10	лоц	лоц	20	щеть	щеть	30	ноть	ноть	40	здап	здап	50	рул	рул
S_4	92													

Таблица А.10 — Результаты прослушивания таблицы 5 для кодека G.711

№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог	№	Слог	Принятый слог
1	рёк	рёк	11	зырь	зырь	21	фок	фок	31	дамь	дамь	41	стар	стар
2	ниш	ниш	12	чтуй	чтуй	22	шусь	шусь	32	пюм	пюм	42	бань	бань
3	зном	зном	13	быс	быс	23	хул	хул	33	кац	кац	43	вюр	вюр
4	рыс	рыс	14	тет	тет	24	штай	штай	34	бёк	бёк	44	нах	нах
5	гек	гек	15	скес	кес	25	мис	мис	35	чинь	шинь	45	бем	бем
6	тань	тань	16	мофь	мофь	26	дват	дват	36	щурь	щурь	46	сах	сах
7	вуф	вуф	17	лям	лям	27	трюль	трюль	37	луф	луф	47	десь	тесь
8	трёп	трёп	18	гоф	гоф	28	нуц	нуц	38	фиш	фиш	48	жус	жус
9	мёх	мёх	19	сях	сях	29	рель	рель	39	сузнь	сузнь	49	цой	цой
10	пась	пась	20	роц	роц	30	ший	ший	40	прень	прень	50	лорь	лорь
S_5	94													