

Гай Н. В. Система обмена сообщениями с помощью звуковых сигналов, Выпускная квалификационная работа бакалавра по специальности: «Информационные системы и технологии», Нижегородский государственный технический университет им. Р. Е. Алексеева, кафедра: «Электроника и сети ЭВМ», Нижний Новгород, 2014. Руководитель: доцент кафедры «Вычислительные системы и технологии» Гай В. Е.

Работа посвящена разработке и программной реализации системы обмена сообщениями с помощью звуковых сигналов. Описывается структура системы, приводится выбор методов для реализации модулей, входящих в систему. Подробно рассматривается программная реализация модулей кодирования / манипуляции и деманипуляции / декодирования передаваемых сигналов.

В результате тестирования установлено, что разработанная система корректно решает поставленную задачу обмена сообщениями с помощью звуковых сигналов.

Объём работы 30 страниц. Использовано источников – 6, рисунков – 14, таблиц – 4.

Перв. примен.	Содержание				
	Введение4 1 Техническое задание5 1.1 Назначение разработки и область применения.....5 1.2 Технические требования5 2 Анализ технического задания.6 2.1 Выбор операционной системы6 2.2 Выбор языка программирования7 2.2 Выбор языка программирования8 2.3 Выбор среды разработки.....10 2.4 Выбор способа манипуляции11 3 Разработка структуры системы обмена сообщениями14 4 Разработка алгоритма обмена сообщениями19 5 Разработка программных средств22 6 Тестирование системы25 Список литературы.....30				
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

					ВР-НГТУ-11ИСТ-5-14 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Гай Н.В.						
Провер.	Гай В.Е.						
Реценз.	Ф.И.О.						
Н. Контр.	Кирсанычев А.Н.						
Утверд.	Милов В.Р.						

Система обмена сообщениями с помощью звуковых сигналов			Лит.	Лист	Листов
				3	30
НГТУ им. Р.Е. Алексеева					

Перв. примен.		<h2 style="text-align: center;">Введение</h2> <p>Развитие информационных технологий привело к созданию большого количества технологий обмена данными между различными устройствами: локальные сети, сотовая связь и т. д.</p> <p>С появлением большого количества мобильных устройств возникает проблема обмена информацией между ними. В настоящее время существует несколько технологий обмена информацией между близко расположенными устройствами: Bluetooth, NFC (Near Field Communication – ближняя бесконтактная связь).</p> <p>Однако, не все устройства оснащены модулями на основе подобных технологий. В связи с этим возникла идея создания системы, которая позволит обмениваться устройствам сообщениями с помощью звуковых сигналов. Учитывалось, что большинство современных устройств (ноутбуки, смартфоны, планшеты) имеют в своём составе динамик и микрофон, которые и являются необходимыми для приёма / передачи сообщений.</p> <p>Таким образом, целью данной работы является создание системы обмена сообщениями с помощью звуковых сигналов. Преимущество разрабатываемой системы перед технологией Bluetooth заключается в возможности широковещательной передачи данных.</p> <p>Разрабатываемая система является прототипом аналогичной системы для мобильных устройств: планшетов, смартфонов.</p>					
	Справ. №						
Инов. № подл.	Подпись и дата					<div style="text-align: center;"> ВР-НГТУ-11ИСТ-5-14 ПЗ </div>	Лист
	Инов. № дубл.						<div style="text-align: center;"> 4 </div>
	Взам. инв. №						
	Подпись и дата						
		Изм.	Лист	№ докум.	Подпись		

Перв. примен.		<h1>1 Техническое задание</h1> <h2>1.1 Назначение разработки и область применения</h2> <p>Разрабатываемая система предназначена для передачи сообщений между ЭВМ при помощи звуковых сигналов. Под сообщениями понимаются любые текстовые данные. Для работы с данной системой должна существовать передающая сторона (всегда одна) и принимающая сторона (может быть как одна, так и несколько).</p> <p>Области применения разрабатываемой системы:</p> <ul style="list-style-type: none"> – передача контактов между мобильными телефонами, на которых установлена система обмена сообщениями; – передача ссылок на веб-сайты между мобильными устройствами; – передача текстовых данных (телефон, адрес) или ссылок через теле- радиозфир звуковым сообщением. <p>Разрабатываемая система является прототипом системы обмена сообщениями с помощью звуковых сигналов для мобильных устройств (смартфонов и планшетов).</p> <h2>1.2 Технические требования</h2> <p>Рассмотрим требования, предъявляемые разрабатываемой системой к ЭВМ:</p> <ul style="list-style-type: none"> – операционная система Microsoft Windows не ниже версии XP; – требования к аппаратному обеспечению определяются операционной системой; – клавиатура и мышь; – микрофон и динамики. <p>Разрабатываемая система передачи сообщений звуком, должна обладать следующим функционалом.</p> <p>Передающая сторона:</p> <ul style="list-style-type: none"> – должна предоставлять пользователю возможность ввода текстового сообщения; – позволять выводить на оконечное устройство воспроизведения звука, сгенерированный звуковой сигнал, по введённому текстовому сообщению. <p>Принимающая сторона:</p> <ul style="list-style-type: none"> – должна обладать функционалом приёма звукового сигнала; – извлечения текстового сообщения из звукового сигнала; – вывод полученного текстового сообщения. 				
	Справ. №					
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.	<div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div>
Изм.	Лист	№ докум.	Подпись	Дата		<div>Лист</div> <div>5</div>

Перв. примен.		<h2>2 Анализ технического задания.</h2> <h3>2.1 Выбор операционной системы</h3> <p>Одна из важнейших задач, которую предстоит решить перед началом разработки – это выбор операционной системы. Существует множество операционных систем, и каждая имеет свои особенности в работе, достоинства и недостатки. Широкое распространение имеют операционные системы Windows, Linux и Mac OS.</p> <p>Рассмотрим их более подробно:</p> <ul style="list-style-type: none"> – Windows – это одна из самых известных операционных систем, которая была разработана компанией Microsoft. Данная операционная система получила большое распространение, и используется как для работы на предприятиях, так и для работы на домашнем компьютере. Операционная система Windows первоначально была основана на MS-DOS. Эти серии известны как 9-ые (Windows 95, Windows 98). Все последующие операционные системы Windows основаны на Windows NT. Самые последние из них это Windows 7 и Windows 8. – Mac OS – это операционная система, разработанная компанией Apple. В настоящее время является второй наиболее часто используемой операционной системой после Windows. Mac OS в отличие от Windows, основана на системе Unix. Таким образом, Mac OS считается частью семейства операционных систем Unix, как и Linux. – Linux – общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux создаётся и распространяется в соответствии с моделью разработки свободного и открытого программного обеспечения. Поэтому общее название не подразумевает какой-либо единой «официальной» комплектации Linux; они распространяются в основном бесплатно в виде различных готовых дистрибутивов, имеющих свой набор прикладных программ и уже настроенных под конкретные нужды пользователя. Операционная система Linux не так популярна, как Mac OS или Windows, но в последнее время её популярность возрастает. В отличие от Mac OS или Windows, у Linux свободный и открытый исходный код. <p>Основные различия, преимущества и недостатки операционных систем Windows, Mac OS, Linux:</p> <ol style="list-style-type: none"> 1) Защита <p>По данному критерию операционная система Windows уступает Linux и Mac OS. Причиной этого является популярность операционной системы Windows, и следствие большое количество вредоносных программ разработанных для данной операционной системы.</p>						
	Справ. №							
Подпись и дата								
	Инв. № дубл.							
Взам. инв. №								
Подпись и дата								
Инв. № подл.								
		Изм.	Лист	№ докум.	Подпись	Дата	ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист
								6

Перв. примен.															
Справ. №															
Подпись и дата															
Инв. № дубл.															
Взам. инв. №															
Подпись и дата															
Инв. № подл.															
<div> <div>2) Стабильность работы</div> <div> <p>Наиболее стабильной является операционная система Linux по сравнению с Mac OS и Windows. У данной операционной системы бывают сбои, но происходят они гораздо реже, чем на Windows или OS X.</p> </div> <div>3) Необходимое оборудование</div> <div> <p>Для Windows и Linux, можно выбрать практически любое оборудование, для установки операционной системы. Имеется большой ассортимент оборудования от таких производителей как Dell, HP, Acer, Gateway, Lenovo, Asus, и т.д. Для Mac OS, есть только оборудование от Apple.</p> </div> <div>4) Программное обеспечение</div> <div> <p>У операционной системы Windows самая большая библиотека программного обеспечения, по сравнению с Linux и Mac OS, следовательно, большинство программ и сделаны для Windows. Такие популярные программы, как Microsoft Office доступны только для операционной системы Windows.</p> </div> <div>5) Применение и разработка программного обеспечения</div> <div> <p>Операционная система Windows является основной платформой для применения и разработки программного обеспечения, в отличие от Linux и Mac OS. Большинство программ для операционной системы Windows написаны на таких языках программирования как C++, C#, Java и Visual Basic. Некоторые из этих языков программирования работают на операционных системах Mac OS и Linux, но новые и наиболее распространенные из них работают только в операционной системе Windows, например Visual Basic и C #.</p> <p>Для разработки прототипа системы передачи сообщений с помощью звука была выбрана операционная система Windows. Она соответствует необходимым требованиям для разработки данного проекта.</p> </div> </div>															
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<div> <div>Лист</div> <div>7</div> </div>
Изм.	Лист	№ докум.	Подпись	Дата											

Перв. примен.	<h2>2.2 Выбор языка программирования</h2> <p>Выбор языка программирования немаловажный этап разработки программного приложения. Учитывая то, что разработка ведётся под операционную систему Windows, можно выделить такие распространенные языки программирования как: C++, C#, и Java.</p> <p>Рассмотрим их более подробно:</p> <ul style="list-style-type: none"> – C++ – это компилируемый строго типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования. – C# – это объектно-ориентированный язык программирования, для разработки приложений для платформы Microsoft .NET Framework. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Переняв многое от своих предшественников – языков C++, Pascal, Модула, Smalltalk и в особенности Java – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов) – Java – это объектно-ориентированный язык программирования. Изначально язык программирования разрабатывался для бытовой электроники, но впоследствии стал использоваться для написания апплетов, приложений и серверного программного обеспечения. Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) – программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее. В Java добавлены все хорошие черты C++. Решающим факторам выбора языка стало наличие библиотеки для работы со звуковыми файлами – NAudio, которая была разработана для использования с языком программирования C#. Основные функции, которые включает библиотека Naudio, и которые необходимы для реализации проекта это: 															
	Справ. №	Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.										
<div style="display: flex; justify-content: space-between; align-items: center;"> <table border="1" style="width: 40%;"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table> <div style="text-align: center; width: 50%;"> <p>ВР-НГТУ-11ИСТ-5-14 ПЗ</p> </div> <div style="width: 10%; text-align: center;"> <p>Лист</p> <p>8</p> </div> </div>							Изм.	Лист	№ докум.	Подпись	Дата					
Изм.	Лист	№ докум.	Подпись	Дата												

Перв. примен.		Справ. №		<div>– воспроизводить и записывать звуковой сигнал, – генерировать звуковой сигнал – считывать информацию из звуковых файлов</div>								
Подпись и дата		Инв. № дубл.		Взам. инв. №		Подпись и дата		<div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div>				
Инв. № подл.												
Изм.	Лист	№ докум.	Подпись	Дата						Лист		
										9		

Перв. примен.		<h2 style="text-align: center;">2.3 Выбор среды разработки</h2> <p>В настоящее время существует широкий выбор средств разработки с различным инструментарием, работающие на разных платформах и наиболее подходящие для разных операционных систем. При выборе среды разработки, требовалось, чтобы она поддерживала разработку на языке программирования C#, следовательно, поддерживала разработку для платформы .NET, и подходила по техническим требованиям для установки для операционной системы Windows. Исходя из перечисленных требований, была выбрана среда разработки Microsoft Visual Studio 2010, которая полностью подходит для решения поставленных задач.</p> <p>Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования, или инструментов для прочих аспектов процесса разработки программного обеспечения.</p>				
	Справ. №					
Подпись и дата		<div style="text-align: right; font-weight: bold; font-size: 1.2em;">ВР-НГТУ-11ИСТ-5-14 ПЗ</div>				
Инов. № дубл.		<div style="text-align: right; font-weight: bold; font-size: 1.2em;">10</div>				
Взам. инв. №						
Подпись и дата						
Инов. № подл.						
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

2.4 Выбор способа манипуляции

Модуляция – процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала (сообщения).

Манипуляция (цифровая модуляция) – в теории передачи дискретных сообщений процесс преобразования последовательности кодовых символов в последовательность элементов сигнала (частный случай модуляции – при дискретных уровнях модулирующего сигнала).

При передаче дискретной информации посредством модуляции единицы и нули кодируются изменением амплитуды, частоты или фазы несущего синусоидального сигнала. В случае, когда модулированные сигналы передают дискретную информацию, вместо термина «модуляция» иногда используется термин «манипуляция»: амплитудная манипуляция (Amplitude Shift Keying, ASK), частотная манипуляция (Frequency Shift Keying, FSK), фазовая манипуляция (Phase Shift Keying, PSK).

Рассмотрим их более подробно:

- при амплитудной манипуляции каждому цифровому символу сопоставляется своя амплитуда несущего сигнала. Частота и фаза манипулированного сигнала остаются неизменными. Амплитудная манипуляция редко используется на практике, т.к. из всех видов манипуляции наименее помехоустойчива. Амплитудная манипуляция обычно применяется в сочетании с другими видами манипуляции.

Рисунок 1 – Амплитудная манипуляция

- при частотной манипуляции каждому цифровому символу сопоставляется своя частота несущего сигнала. Амплитуда и фаза манипулированного сигнала не меняются.

					ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист 11
Изм.	Лист	№ докум.	Подпись	Дата		

Частотная манипуляция весьма помехоустойчива, поскольку искажениям при помехах подвергается в основном амплитуда сигнала, а не частота.

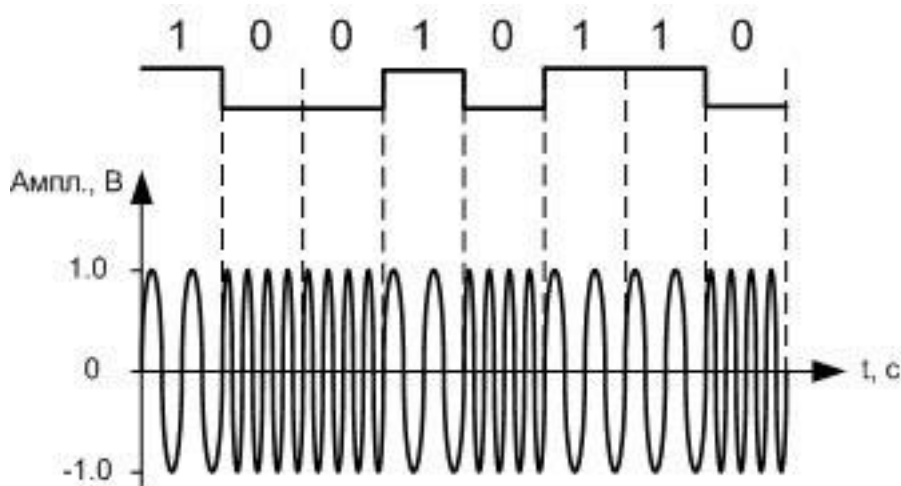


Рисунок 2 – Частотная манипуляция

– при фазовой манипуляции каждому цифровому символу сопоставляется своя начальная фаза несущего сигнала при неизменной амплитуде. Данный вид манипуляции наиболее сложен в реализации, но и наиболее помехоустойчив по сравнению с двумя другими видами манипуляции.

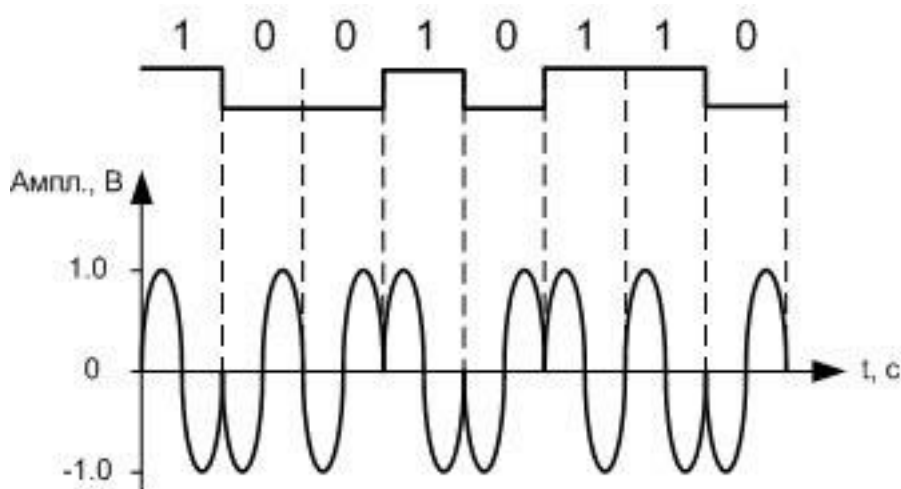


Рисунок 3 – Фазовая манипуляция

Учитывая особенности и возможности рассмотренных выше видов манипуляции, была выбрана фазовая манипуляция, но учитывая её основной недостаток, когда ошибка в одном символе может привести к ошибочному детектированию всех последующих символов, была выбрана дифференциальная фазовая манипуляция (относительная фазовая

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> <div> <div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div> <div>Лист</div> <div>13</div> </div> </div>					

манипуляция или фазоразностная манипуляция), которая свободна от этого нежелательного эффекта.

При дифференциальной фазовой манипуляции каждому информационному элементу ставится в соответствие не абсолютное значение начальной фазы, а ее изменение относительно предыдущего значения. Если информационный элемент есть двойной бит (дибит), то в зависимости от его значения (00,01,10,11) начальная фаза сигнала может изменяться на 90, 180, 270 градусов или не изменяться вовсе.

Перв. примен.	<h3 style="margin: 0;">3 Разработка структуры системы обмена сообщениями</h3>															
	<p>Разрабатываемую систему обмена сообщениями можно представить в виде передающей и принимающей стороны, которые включают в себя основные модули работы системы (см. рис. 4).</p>															
Справ. №	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 10px; width: 45%;"> <p style="text-align: center; margin: 0;">Передающая сторона</p> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center;">Модуль кодирования сообщения в двоичный код</div> <div style="text-align: center; margin: 5px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center;">Модуль формирования фазоразностного сигнала</div> <div style="text-align: center; margin: 5px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center;">Модуль воспроизведения сигнала</div> </div> <div style="border: 1px solid black; padding: 10px; width: 45%;"> <p style="text-align: center; margin: 0;">Принимающая сторона</p> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center;">Модуль записи сигнала</div> <div style="text-align: center; margin: 5px;">↓</div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center; flex: 1;">Модуль демодуляции сигнала</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center; flex: 1;">Модуль вычисления дискретного преобразования Фурье</div> </div> <div style="text-align: center; margin: 5px;">←</div> <div style="text-align: center; margin: 5px;">↓</div> <div style="border: 1px solid black; padding: 5px; margin: 5px; text-align: center;">Модуль декодирования сообщения</div> </div> </div>															
	<p>Рисунок 4 – Структура системы обмена сообщениями</p>															
<p>Рассмотрим модули, находящиеся на передающей стороне:</p>																
Подпись и дата	<p>1) Модуль кодирования текстового сообщения в двоичный код – служит для приёма текстового сообщения введённого с клавиатуры, и перевод данного сообщения в двоичный код. При переводе текстового сообщения в двоичный код используется расширенная таблица ASCII (ср866).</p>															
	<p>2) Модуль формирования дифференциального фазоманипулированного сигнала (фазоразностного сигнала) – формирует сигнал, на основе двоичного кода полученного из предыдущего модуля. В зависимости от бита (0 или 1), генерируется сигнал одинаковой частоты, но с разной начальной фазой.</p>															
Инв. № дубл.	<p>Модуляция – процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала (сообщения).</p>															
	<p>Манипуляция (цифровая модуляция) – в теории передачи дискретных сообщений процесс преобразования последовательности кодовых символов в последовательность элементов сигнала (частный случай модуляции – при дискретных уровнях модулирующего сигнала).</p>															
Взам. инв. №	<p>Дифференциальная фазовая манипуляция (Differential Phase-shift keying (DPSK)) – один из видов фазовой модуляции, при которой фаза несущего колебания меняется не при каждом изменении значения модулирующего импульса, а при изменении разности.</p>															
Подпись и дата																
Инв. № подл.																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; height: 20px;"></td> <td style="width: 15%; height: 20px;"></td> <td style="width: 15%; height: 20px;"></td> <td style="width: 15%; height: 20px;"></td> <td style="width: 15%; height: 20px;"></td> <td style="width: 15%; height: 20px;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подпись</td> <td style="text-align: center;">Дата</td> <td></td> </tr> </table>											Изм.	Лист	№ докум.	Подпись	Дата	
Изм.	Лист	№ докум.	Подпись	Дата												
<div style="display: flex; justify-content: space-between; align-items: center;"> ВР-НГТУ-11ИСТ-5-14 ПЗ <div style="border: 1px solid black; padding: 2px 5px; font-size: 0.8em;">Лист 14</div> </div>																

Перв. примен.	<p>Фазоманипулированный сигнал имеет следующий вид:</p> $x_m(t) = g(t) \sin(2\pi f_c t + \varphi_m(t)) \quad (1)$ <p>где $g(t)$ определяет огибающую сигнала;</p> <p>$\varphi_m(t)$ является модулирующим сигналом и может принимать m дискретных значений;</p> <p>f_c – частота несущей;</p> <p>t – время.</p> <p>Если $m = 2$, то фазовая манипуляция называется двоичной фазовой манипуляцией (DBPSK, B-Binary – 1 бит на 1 смену фазы), если $m = 4$ – квадратурной фазовой манипуляцией (DQPSK, Q-Quadro – 2 бита на 1 смену фазы), $m = 8$ (8-PSK – 3 бита на 1 смену фазы) и т.д. Таким образом, количество бит n, передаваемых одним перескоком фазы, является степенью, в которую возводится двойка при определении числа фаз, требующихся для передачи n – порядкового двоичного числа</p> <p>В качестве несущего сигнала в фазовой манипуляции используется гармоническое колебание.</p> <p>Гармоническое колебание – колебание, при котором физическая (или любая другая) величина изменяется с течением времени по синусоидальному или косинусоидальному закону:</p> $x(t) = A \cdot \sin(\omega t + \varphi) \quad (2)$ <p>где A, ω, φ – постоянные величины, которые могут исполнять роль информационных параметров сигнала;</p> <p>x – смещение (отклонение) колеблющейся точки от положения равновесия в момент времени t;</p> <p>A – амплитуда колебаний, это величина, определяющая максимальное отклонение колеблющейся точки от положения равновесия;</p> <p>ω – циклическая частота, величина, показывающая число полных колебаний происходящих в течение 2π секунд;</p> <p>$(\omega t + \varphi)$ – полная фаза колебаний;</p> <p>φ – начальная фаза колебаний.</p>																	
	Справ. №																	
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.													
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> <td rowspan="2"> <div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div> </td> <td>Лист</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>15</td> </tr> </table>						Изм.	Лист	№ докум.	Подпись	Дата	<div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div>	Лист						15
Изм.	Лист	№ докум.	Подпись	Дата	<div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div>	Лист												
						15												

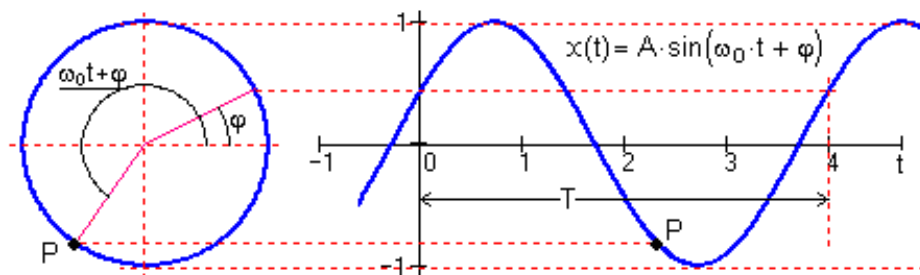


Рисунок 5 – Синусоидальный гармонический сигнал

Формулу синусоидального гармонического колебания можно также представить как:

$$x(t) = A \cdot \sin(2\pi f_0 t + \varphi) \quad (3)$$

период одного колебания

$$T = \frac{1}{f_0} = \frac{2\pi}{\omega_0} \quad (4)$$

где f_0 – циклическая частота в герцах;

ω_0 – угловая частота в радианах.

3) Модуль воспроизведения звукового сигнала – данный модуль отвечает за вывод сигнала, полученного в результате его манипуляции, на окончательное устройство воспроизведения звука, т.е. на динамики, звуковые колонки и т.д.

Рассмотрим модули, находящиеся на принимающей стороне:

1) Модуль записи звукового сигнала – при помощи микрофона, записывается передаваемый звуковой сигнал. Переданный сигнал сохраняется на жесткий диск в формате wav, для его дальнейшего анализа следующими модулями.

2) Модуль демодуляции сигнала – в данном модуле полученный сигнал разбивается на сегменты, и при помощи анализа полученных данных из модуля вычисления дискретного преобразования Фурье, сравнивает начальные фазы соседних сегментов, с целью определения принятого бита.

3) Модуль вычисления дискретного преобразования Фурье – служит для вычисления дискретного преобразования Фурье по отдельным сегментам принятого сигнала.

В основе преобразования Фурье лежит очень простая идея – почти любую периодическую функцию можно представить суммой гармонических составляющих. Преобразование Фурье предназначено для перевода сигнала, заданного во временном пространстве, в частотное пространство. Во временном пространстве функция времени задаётся привычным для всех образом (например, звуковой сигнал), т.к. по оси абсцисс

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

откладывается время. В частотном пространстве функция времени отображается несколько иначе за счёт того, что по оси абсцисс откладывается частота, а по оси ординат – амплитуда гармоник, составляющие функцию. Представление функции в частотной области называют спектром функции.

Главное достоинство преобразования Фурье заключается в том, что оно может применяться как для непрерывных функций, так и для дискретных. В последнем случае оно называется дискретным преобразованием Фурье.

Дискретное преобразование Фурье – это преобразование конечных последовательностей (комплексных) чисел, которое, как и в непрерывном случае, превращает свёртку в поточечное умножение. Используется в цифровой обработке сигналов и в других ситуациях, где необходимо быстро выполнять свёртку, например, при умножении больших чисел.

Прямое преобразование:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi \cdot i}{N} k \cdot n} \quad \text{при } k = 0, \dots, N-1 \quad (5)$$

где N – количество значений сигнала, измеренных за период, а также количество компонент разложения;

$x_n, \quad n = 0, \dots, N-1$, – измеренные значения сигнала в дискретных временных точках с номерами, $n = 0, \dots, N-1$ которые являются входными данными для прямого преобразования;

$X_k, \quad k = 0, \dots, N-1$, – N комплексных амплитуд синусоидальных сигналов, слагающих исходный сигнал, являются выходными данными для прямого, поскольку амплитуды комплексные, то по ним можно вычислить одновременно и амплитуду, и фазу;

k – индекс частоты. Частота k – го сигнала равна $\frac{k}{T}$, где T – период времени, в течение которого брались входные данные.

При вычислении дискретного преобразования Фурье будет использована формула Эйлера для представления комплексной экспоненты с помощью тригонометрических функций.

Формула Эйлера:

$$e^{i\varphi} = \cos \varphi + i \sin \varphi \quad (6)$$

где e – основание натурального логарифма,

i – мнимая единица.

Изм.	Лист	№ докум.	Подпись	Дата	ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист
						17

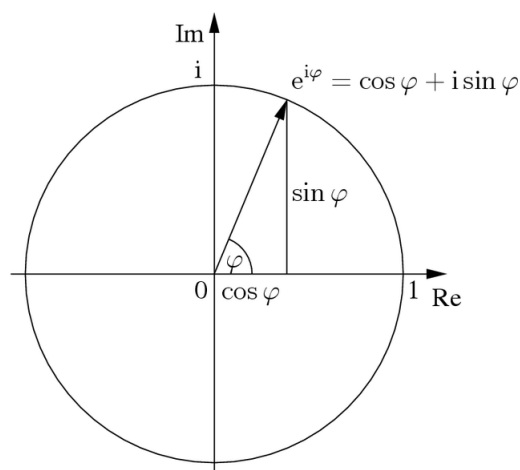


Рисунок 6 – Геометрический смысл формулы Эйлера

В результате дискретное преобразование Фурье можно будет рассчитать по следующей формуле:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot (\cos \varphi - i \cdot \sin \varphi) \quad (7)$$

где $\varphi = \frac{2\pi}{N} \cdot k \cdot n$

4) Модуль декодирования сообщения – выполняет декодирование сообщения, по принятому двоичному коду, в текстовое сообщения. Результат декодирования выводится на экран.

Перв. примен.	<h2 style="margin: 0;">4 Разработка алгоритма обмена сообщениями</h2>				
	<p>Рассмотрим блок-схемы работы принимающей и передающей стороны (см. рис. 7).</p>				
Справ. №	<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> <pre> graph TD Start([Начало]) --> Input[Ввод текстового сообщения] Input --> Convert[Перевод строки в десятичный код] Convert --> Decision{Сообщение полностью переведено в звуковой формат?} Decision -- да --> Playback[Воспроизведение сгенерированного звукового сообщения] Playback --> End([Конец]) Decision -- нет --> Convert2[Перевод сообщения из десятичного кода в двоичный код] Convert2 --> Generate[Генерация звукового сообщения по двоичному коду] Generate --> Decision </pre> </div> <div style="width: 45%;"> <pre> graph TD Start([Начало]) --> Record[Запись звукового сигнала] Record --> Divide[Деление сигнала на сегменты] Divide --> Decision{Проанализирован весь сигнал?} Decision -- да --> Convert2[Перевод сообщения из двоичного кода в десятичный код] Convert2 --> FormText[Формирование текстового сообщения по десятичному коду] FormText --> End([Конец]) Decision -- нет --> Compare[Сравнение начальных фаз соседних сегментов] Compare --> FormBin[Формирование двоичного кода по принятому сигналу] FormBin --> Decision </pre> </div> </div>				
	<div style="display: flex; justify-content: space-around;"> <div style="width: 45%;"> <p>Рисунок 7.а – Блок-схема работы передающей стороны</p> </div> <div style="width: 45%;"> <p>Рисунок 7.б – Блок-схема работы принимающей стороны</p> </div> </div>				
Подпись и дата	<p>Принцип работы передающей стороны (см. рис.7.а):</p> <p>Введённое пользователем сообщение кодируется в десятичный код по расширенной таблице ASCII (ср866). Далее, каждое десятичное число переводится в двоичный код. Операция перевода десятичного кода в двоичный код и генерация сигнала выполняется несколько раз, в зависимости от количества символов введённого сообщения. Сгенерированный сигнал выводится на оконечное устройство воспроизведения звука.</p> <p>Рассмотрим более подробно функцию генерации звукового сообщения по двоичному коду (см. рис. 8):</p>				
	<div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Изм.</p> <p>Лист</p> <p>№ докум.</p> <p>Подпись</p> <p>Дата</p> </div> <div style="width: 50%; text-align: center;"> <p style="font-size: 1.2em; font-weight: bold;">ВР-НГТУ-11ИСТ-5-14 ПЗ</p> </div> <div style="width: 10%; text-align: right;"> <p>Лист</p> <p style="font-size: 1.2em; font-weight: bold;">19</p> </div> </div>				
Инв. № подл.	<div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Изм.</p> <p>Лист</p> <p>№ докум.</p> <p>Подпись</p> <p>Дата</p> </div> <div style="width: 50%; text-align: center;"> <p style="font-size: 1.2em; font-weight: bold;">ВР-НГТУ-11ИСТ-5-14 ПЗ</p> </div> <div style="width: 10%; text-align: right;"> <p>Лист</p> <p style="font-size: 1.2em; font-weight: bold;">19</p> </div> </div>				
	<div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Изм.</p> <p>Лист</p> <p>№ докум.</p> <p>Подпись</p> <p>Дата</p> </div> <div style="width: 50%; text-align: center;"> <p style="font-size: 1.2em; font-weight: bold;">ВР-НГТУ-11ИСТ-5-14 ПЗ</p> </div> <div style="width: 10%; text-align: right;"> <p>Лист</p> <p style="font-size: 1.2em; font-weight: bold;">19</p> </div> </div>				

Перв. примен.				
Справ. №				
Подпись и дата				
Инв. № дубл.				
Взам. инв. №				
Подпись и дата				
Инв. № подл.				

```

graph TD
    Start([Начало]) --> Dec1{Сгенерирован сигнал для всех передаваемых байтов}
    Dec1 -- да --> End([Конец])
    Dec1 -- нет --> Dec2{Сгенерирован сигнал для всего байта}
    Dec2 -- да --> Gen1[Генерация сигнала  
y = sin(ωt + φ)  
φ = 3π/2]
    Gen1 --> Rec1[Запись сгенерированного сигнала в wav файл]
    Rec1 --> Dec1
    Dec2 -- нет --> Dec3{Передаваемый бит == 1}
    Dec3 -- да --> Gen2[Генерация сигнала  
y = sin(ωt + φ)  
φ = π/2]
    Dec3 -- нет --> Gen3[Генерация сигнала  
y = sin(ωt + φ)  
φ = π]
    Gen2 --> Rec2[Запись сгенерированного сигнала в wav файл]
    Gen3 --> Rec2
    Rec2 --> Dec2
  
```

Рисунок 8 – Блок-схема функции генерации звукового сообщения по двоичному коду

Данная функция генерирует звуковой сигнал по формуле гармонического колебания и записывает его в wav файл. Значение начальной фазы изменяется в зависимости от принятого бита.

Принцип работы принимающей стороны (см. рис. 7.б):

Принятый сигнал делится на сегменты, количество сегментов зависит от характеристик сигнала. Далее по каждому сегменту вычисляется преобразование Фурье, в результате вычисления полученная начальная фаза сравнивается с начальной фазой соседнего сегмента, для определения принятого бита и формирования двоичного кода. Операция сравнение начальных фаз соседних сегментов и формирование двоичного кода выполняется несколько раз, в зависимости от количества сегментов в принятом сигнале. Сгенерированный десятичный код переводится в двоичный код, на основе которого формируется текстовое сообщение по расширенной таблице ASCII (cp866).

					ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист 20
Изм.	Лист	№ докум.	Подпись	Дата		

Рассмотрим более подробно функцию сравнения начальных фаз соседних сегментов (см. рис. 9):

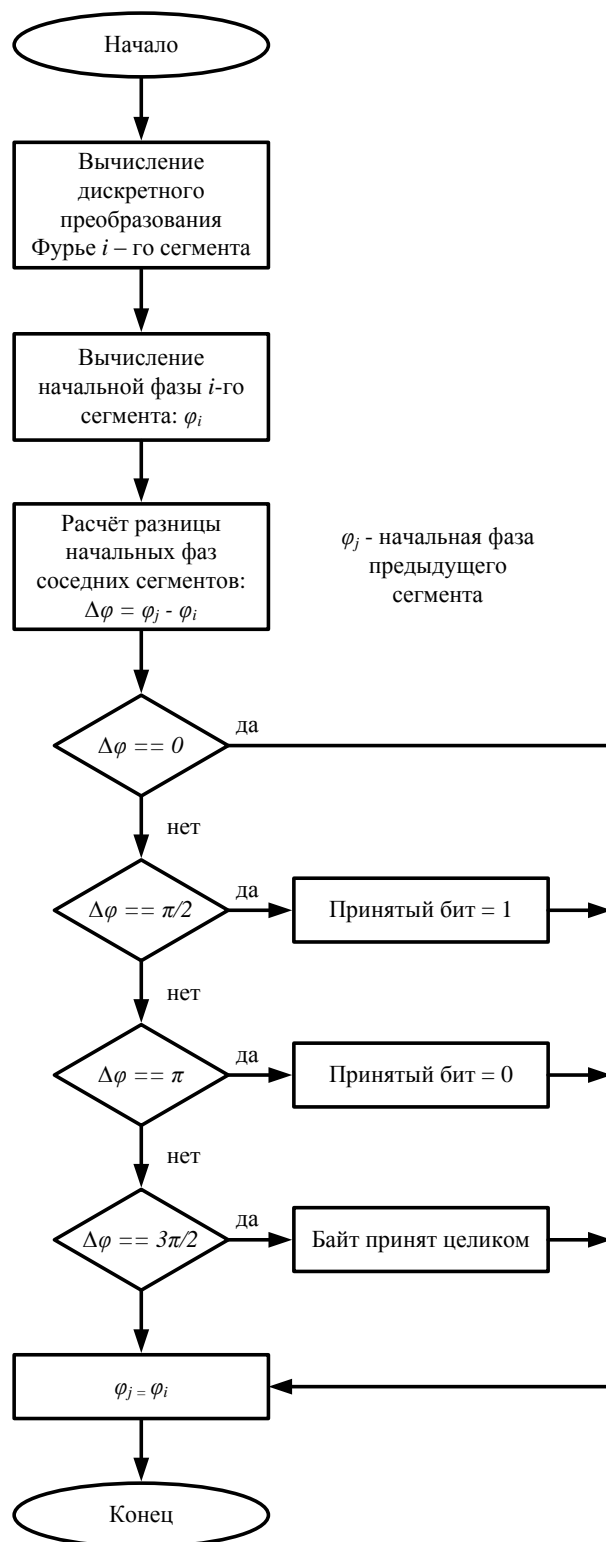


Рисунок 9 – Блок-схема функции сравнения начальных фаз соседних сегментов

Данная функция выполняет вычисление дискретного преобразования Фурье по принятому сегменту, для дальнейшего расчёта начальных фаз соседних сегментов, и анализа принятого сигнала.

Перв. примен.	<h2 style="margin: 0;">5 Разработка программных средств</h2>											
	<p>Рассмотрим основные переменные, используемые в программе и необходимые для её реализации:</p> <ol style="list-style-type: none"> 1) int rate = 48000 – количество отсчётов в секунду; 2) int bps = 100 – количество передаваемых бит в секунду; 3) double carrier = 8000.0 – частота периодов в секунду; 4) double car_phase = 0 – начальная фаза. <p>Метод <code>generate()</code> предназначен для генерации сигнала по формуле гармонического сигнала</p> <p>Таблица 1 – Метод <code>generate()</code>.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <tr> <th style="width: 15%;">№Номер</th> <th style="width: 20%;">Имя параметра</th> <th style="width: 20%;">Тип параметра</th> <th style="width: 45%;">Комментарий</th> </tr> <tr> <td style="text-align: center;">1</td> <td>length</td> <td>int, входной параметр</td> <td>Принимает значение равное количеству отсчётов для генерации одного бита</td> </tr> </table> <pre> void generate(int length) // Генерация гармонического сигнала { float[] data = new float[length]; // Размер массива зависит от количества отсчётов int i; for (i = 0; i < length; i++) { data[i] += (float) (64.0 * Math.Sin((2 * Math.PI * carrier * (i) / (float)rate) + car_phase)); waveFile.WriteSample((float)(data[i]) / 64.0f); // Записывает один отсчёт в файл } car_phase = car_phase + 2.0f * Math.PI * carrier * (double)i / (double)rate; car_phase = car_phase - 2.0 * Math.PI * (double) (Int32)(float)(car_phase / (2.0 * Math.PI)); } </pre> <p>Метод <code>generate_nextbit()</code> предназначен для формирования признак сгенерированного байта, изменив значения начальной фазы – $3\pi/2$. Данный метод не имеет входных и выходных параметров.</p> <pre> void generate_nextbit() { car_phase += 3 * Math.PI / 2.0; generate(rate / bps); } </pre> <p>Метод <code>generate_bit()</code> предназначен для анализа принятого бита и изменения значения начальной фазы: $\pi/2$ если значение бита равно 1, и π если значение бита равно 0.</p>					№Номер	Имя параметра	Тип параметра	Комментарий	1	length	int, входной параметр
№Номер	Имя параметра	Тип параметра	Комментарий									
1	length	int, входной параметр	Принимает значение равное количеству отсчётов для генерации одного бита									
Справ. №												

Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

					ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист 22
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.	Таблица 2 – Метод <i>generate_bit()</i>																												
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 15%;">№Номер</th> <th style="width: 25%;">Имя параметра</th> <th style="width: 25%;">Тип параметра</th> <th style="width: 35%;">Комментарий</th> </tr> <tr> <td style="text-align: center;">1</td> <td>c</td> <td>char, входной параметр</td> <td>Принимает значение бита</td> </tr> </table>					№Номер	Имя параметра	Тип параметра	Комментарий	1	c	char, входной параметр	Принимает значение бита																
№Номер	Имя параметра	Тип параметра	Комментарий																										
1	c	char, входной параметр	Принимает значение бита																										
Справ. №	<pre> void generate_bit(char c) // Генерация бита { if (c == '1') // Если бит равен 1 { car_phase += Math.PI / 2.0; // то car_phase = car_phase + (Pi/2) } else // Если бит не равен 0 { car_phase += Math.PI; // то car_phase = car_phase + Pi } generate(rate / bps); // Вызов функции generate } </pre> <p>Метод <i>fourier()</i> предназначен для вычисления дискретного преобразования Фурье.</p> <p>Таблица 3 – Метод <i>fourier()</i></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 10%;">№</th> <th style="width: 25%;">Имя параметра</th> <th style="width: 25%;">Тип параметра</th> <th style="width: 40%;">Комментарий</th> </tr> <tr> <td style="text-align: center;">1</td> <td>x_signal</td> <td>массив float, входной параметр</td> <td>Массив значений отсчётов сигнала</td> </tr> <tr> <td style="text-align: center;">2</td> <td>n</td> <td>double, входной параметр</td> <td>Модифицированная несущая частота</td> </tr> <tr> <td style="text-align: center;">3</td> <td>length</td> <td>int, входной параметр</td> <td>Длина сегмента</td> </tr> <tr> <td style="text-align: center;">4</td> <td>re</td> <td>double, выходной параметр</td> <td>Вещественная часть, полученная в результате вычисления дискретного преобразования Фурье</td> </tr> <tr> <td style="text-align: center;">5</td> <td>im</td> <td>double, выходной параметр</td> <td>Мнимая часть, полученная в результате вычисления дискретного преобразования Фурье</td> </tr> </table> <pre> double fourier(float[] x_signal, double n, int length, out double re, out double im) { double[] x_complex = {0, 0}; int i; for (i = 0; i < length; i++) { x_complex[0] += x_signal[i] * Math.Cos(Math.PI * 2 * i * n / (double)length); x_complex[1] += x_signal[i] * Math.Sin(Math.PI * 2 * i * n / (double)length); } double norm = Math.Sqrt(x_complex[0] * x_complex[0] + x_complex[1] * x_complex[1]); im = x_complex[1] / norm; // Im re = x_complex[0] / norm; // Re return norm / length; } </pre>					№	Имя параметра	Тип параметра	Комментарий	1	x_signal	массив float, входной параметр	Массив значений отсчётов сигнала	2	n	double, входной параметр	Модифицированная несущая частота	3	length	int, входной параметр	Длина сегмента	4	re	double, выходной параметр	Вещественная часть, полученная в результате вычисления дискретного преобразования Фурье	5	im	double, выходной параметр	Мнимая часть, полученная в результате вычисления дискретного преобразования Фурье
	№	Имя параметра	Тип параметра	Комментарий																									
1	x_signal	массив float, входной параметр	Массив значений отсчётов сигнала																										
2	n	double, входной параметр	Модифицированная несущая частота																										
3	length	int, входной параметр	Длина сегмента																										
4	re	double, выходной параметр	Вещественная часть, полученная в результате вычисления дискретного преобразования Фурье																										
5	im	double, выходной параметр	Мнимая часть, полученная в результате вычисления дискретного преобразования Фурье																										
Подпись и дата																													
Инв. № дубл.																													
Взам. инв. №																													
Подпись и дата																													
Инв. № подл.																													

					ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист 23
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>					Лист
<div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div>					24

Метод *receive()* разбивает сигнал на сегменты, и вычисляет по каждому сегменту преобразование Фурье, анализирует полученные результаты, на основе которых формирует текстовое сообщение.

Метод *generate_signal()* генерирует сигнал по двоичному коду, и записывает его в wav файл.

6 Тестирование системы

Пользовательский интерфейс разработанной системы представлен на рисунке 10.

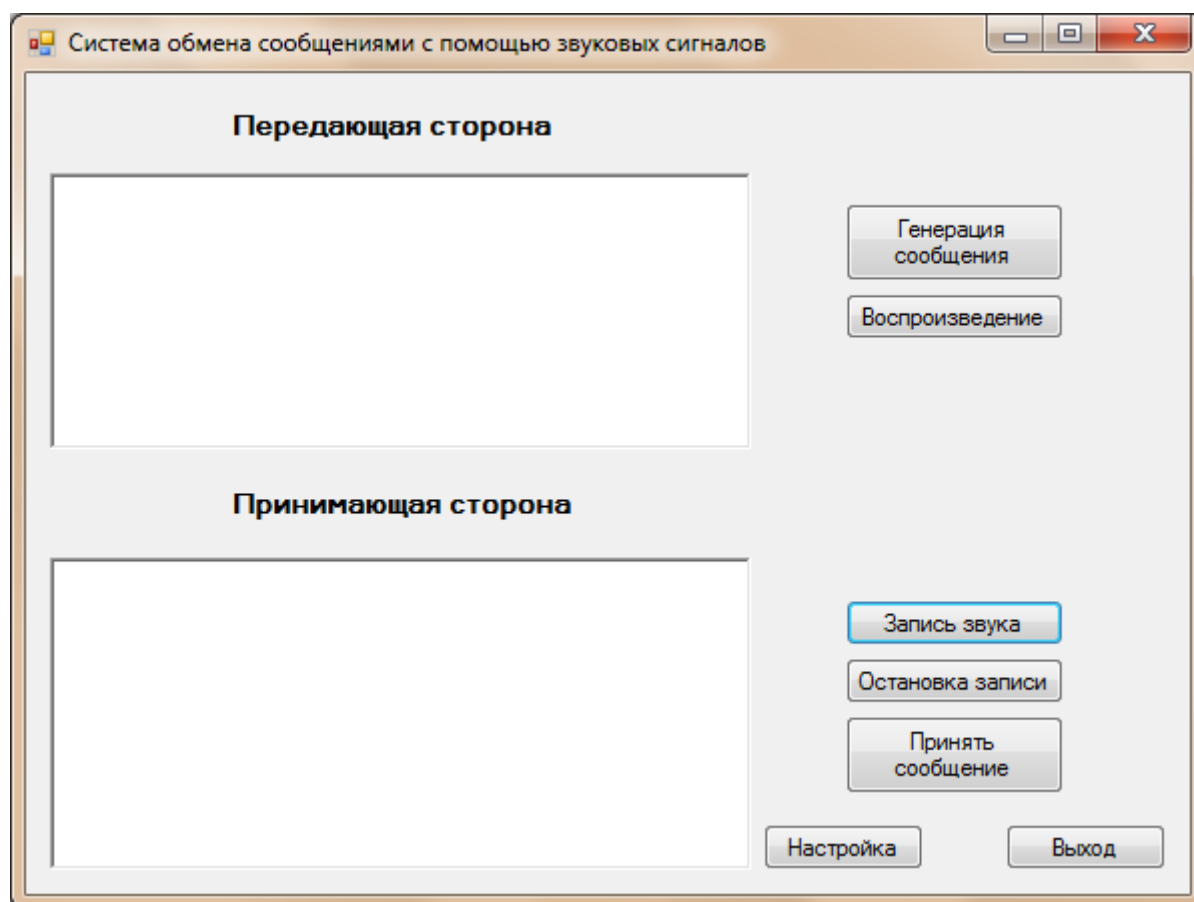


Рисунок 10 – Интерфейс системы

Разработанное программное обеспечение реализует в одном приложении функции передающей и приёмной стороны.

Рассмотрим порядок работы с системой. Для передачи сообщения необходимо:

- 1) ввести передаваемое сообщение в текстовое поле;
- 2) нажать кнопку «Генерация сообщения»;
- 3) воспроизвести сгенерированный сигнал, нажав кнопку «Воспроизведение».

Для приёма сообщения необходимо:

- 1) нажать кнопку «Запись звука» и записать передаваемый сигнал.
- 2) нажать кнопку «Принять сообщение», в результате переданное сообщение будет отображено в текстовом поле.

Изм.	Лист	№ докум.	Подпись	Дата

ВР-НГТУ-11ИСТ-5-14 ПЗ

Лист

25

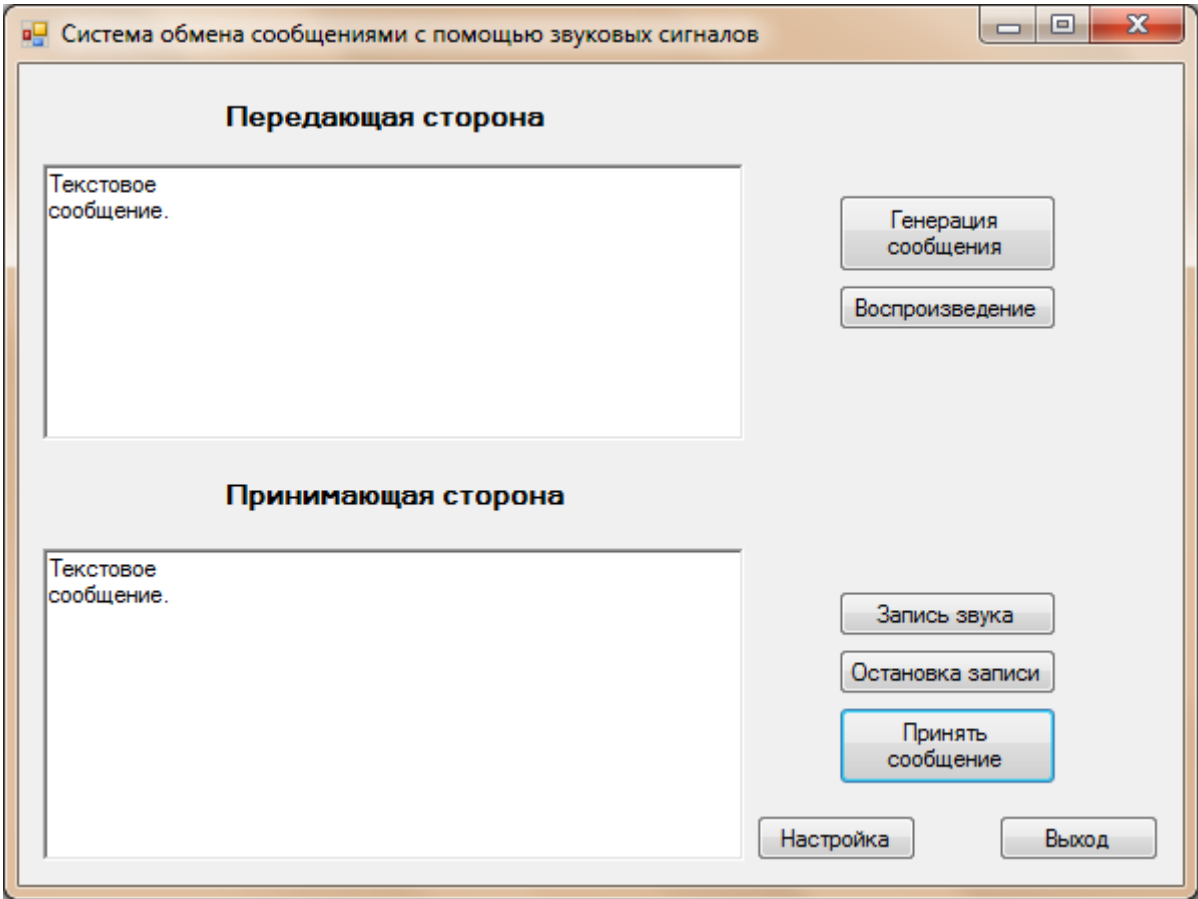


Рисунок 11 – Результат работы системы

Осциллограмма сгенерированного файла показана на рисунке 12. Неравномерности по амплитуде объясняются несовершенством приёмо / передающего оборудования.

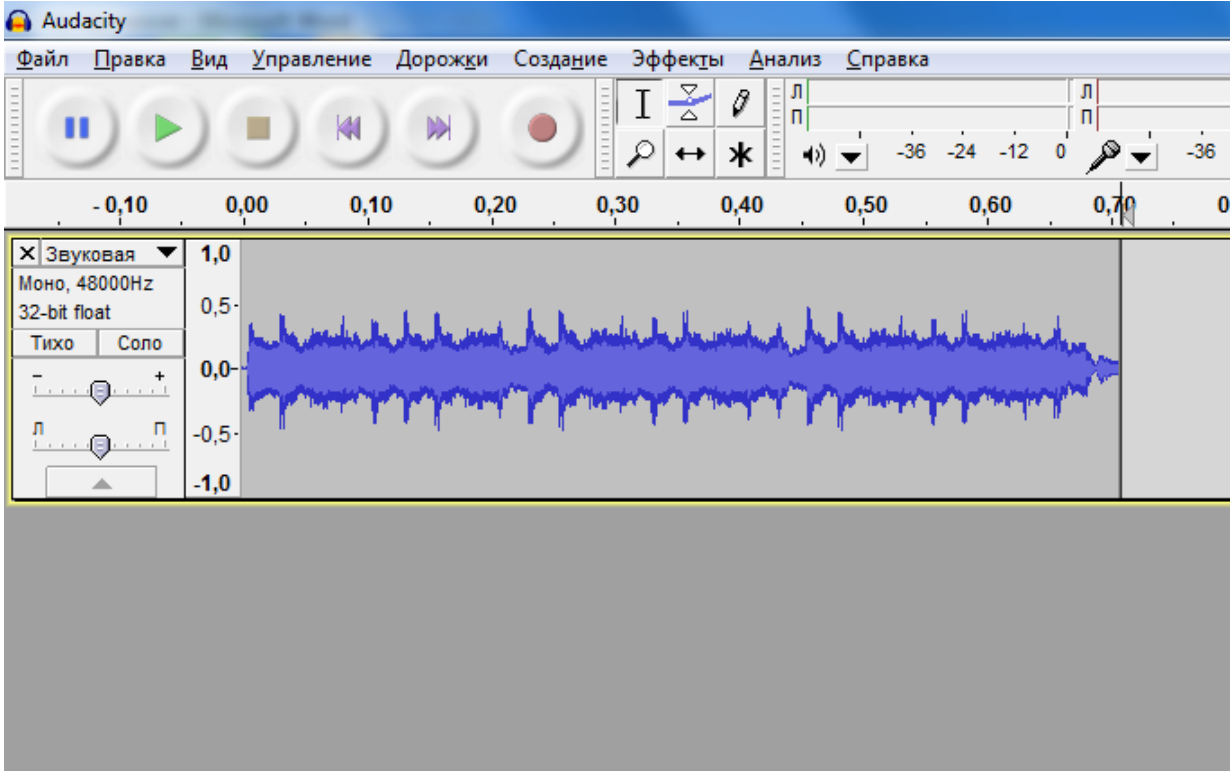


Рисунок 12 – Сгенерированный файл

Перв. примен.	<p>Тестирование системы выполнялось следующим образом:</p> <p>1) тестирование правильности работы системы на одном файле: в этом случае на передающей стороне генерировался звуковой файл, который далее анализировался принимающей стороной (для передачи файла не использовался звуковой канал); результаты этого тестирования подтвердили корректность методов кодирования / манипуляции и деманипуляции / декодирования;</p> <p>2) тестирование при обмене сообщениями между двумя компьютерами, использовалось два ноутбука: Asus K17 и Asus 50ID (ноутбуки были расположены друг напротив друга); на данном этапе тестирования выполнялся анализ допустимых значений параметров системы. Система обмена сообщениями имеет два настраиваемых параметра:</p> <ul style="list-style-type: none"> – частота несущей; – количество бит, передаваемых в секунду. <p>Рассмотрим устойчивость работы системы при различных значениях параметров.</p> <p>Учитывая, что в качестве результата работы системы требуется достоверная передача сообщения, будем указывать: есть ошибка или нет (см. табл. 4).</p> <p style="text-align: center;">Таблица 4 – Результаты тестирования</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <th>№Номер</th> <th>Частота дискретизации (в герцах)</th> <th>Число бит в секунду</th> <th>Результат</th> </tr> <tr><td>1</td><td>4000</td><td>4</td><td>Верная передача</td></tr> <tr><td>2</td><td>4000</td><td>8</td><td>Ошибка передачи</td></tr> <tr><td>3</td><td>4000</td><td>16</td><td>Ошибка передачи</td></tr> <tr><td>4</td><td>8000</td><td>10</td><td>Верная передача</td></tr> <tr><td>5</td><td>8000</td><td>20</td><td>Верная передача</td></tr> <tr><td>6</td><td>8000</td><td>30</td><td>Верная передача</td></tr> <tr><td>7</td><td>8000</td><td>40</td><td>Верная передача</td></tr> <tr><td>8</td><td>8000</td><td>50</td><td>Ошибка передачи</td></tr> <tr><td>9</td><td>10000</td><td>10</td><td>Верная передача</td></tr> <tr><td>10</td><td>10000</td><td>20</td><td>Верная передача</td></tr> <tr><td>11</td><td>10000</td><td>30</td><td>Ошибка передачи</td></tr> <tr><td>12</td><td>12000</td><td>10</td><td>Верная передача</td></tr> <tr><td>13</td><td>12000</td><td>20</td><td>Верная передача</td></tr> <tr><td>14</td><td>12000</td><td>30</td><td>Ошибка передачи</td></tr> </table>					№Номер	Частота дискретизации (в герцах)	Число бит в секунду	Результат	1	4000	4	Верная передача	2	4000	8	Ошибка передачи	3	4000	16	Ошибка передачи	4	8000	10	Верная передача	5	8000	20	Верная передача	6	8000	30	Верная передача	7	8000	40	Верная передача	8	8000	50	Ошибка передачи	9	10000	10	Верная передача	10	10000	20	Верная передача	11	10000	30	Ошибка передачи	12	12000	10	Верная передача	13	12000	20	Верная передача	14	12000	30	Ошибка передачи
	№Номер	Частота дискретизации (в герцах)	Число бит в секунду	Результат																																																													
1	4000	4	Верная передача																																																														
2	4000	8	Ошибка передачи																																																														
3	4000	16	Ошибка передачи																																																														
4	8000	10	Верная передача																																																														
5	8000	20	Верная передача																																																														
6	8000	30	Верная передача																																																														
7	8000	40	Верная передача																																																														
8	8000	50	Ошибка передачи																																																														
9	10000	10	Верная передача																																																														
10	10000	20	Верная передача																																																														
11	10000	30	Ошибка передачи																																																														
12	12000	10	Верная передача																																																														
13	12000	20	Верная передача																																																														
14	12000	30	Ошибка передачи																																																														
Справ. №																																																																	

Подпись и дата					
	Инв. № дубл.				
Взам. инв. №					
	Подпись и дата				
Инв. № подл.					
	Изм.	Лист	№ докум.	Подпись	Дата

ВР-НГТУ-11ИСТ-5-14 ПЗ

Лист

27

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

					ВР-НГТУ-11ИСТ-5-14 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

Значение частоты дискретизации не имеет смысла делать менее 48000 герц. Указанное значение частоты поддерживает большинство звуковых карт, а её уменьшение приведёт к снижению точности приёма / передачи сигнала.

Описанное тестирование системы выполнялось в условиях малого шума (шума работы ноутбуков). При включённой громкой фоновой музыке и громких разговорах передача сообщений выполнялась с ошибками.

Разработанная система поддерживает возможность указания двух параметров: частоты несущей и числа передаваемых бит в секунду. По умолчанию, в системе сохранены следующие параметры: частота дискретизации – 48 кГц, число бит в секунду – 30, частота несущей – 8000 Гц.

Вызов формы настройки выполняется по нажатию на кнопку «Настройка» (см. рис. 13).

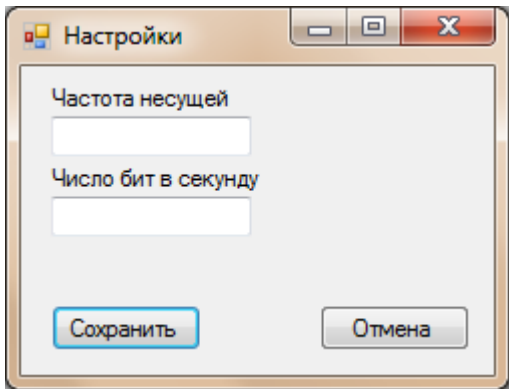


Рисунок 13 – Настройка системы

В случае если в поле, содержащее отправляемое сообщение, не указан текст, будет выведено пояснение пользователю (см. рис. 14).

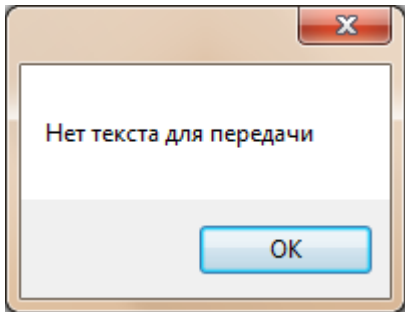


Рисунок 14 – Сообщение пользователю

Значение частоты дискретизации не имеет смысла делать менее 48000 герц. Указанное значение частоты поддерживает большинство звуковых карт, а её уменьшение приведёт к снижению точности приёма / передачи сигнала.

Описанное тестирование системы выполнялось в условиях малого шума (шума работы ноутбуков). При включённой громкой фоновой музыке и громких разговорах передача сообщений выполнялась с ошибками.

Разработанная система поддерживает возможность указания двух параметров: частоты несущей и числа передаваемых бит в секунду. По умолчанию, в системе сохранены следующие параметры: частота дискретизации – 48 кГц, число бит в секунду – 30, частота несущей – 8000 Гц.

Вызов формы настройки выполняется по нажатию на кнопку «Настройка» (см. рис. 13).

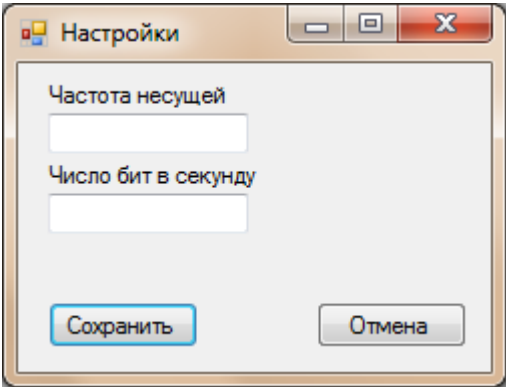


Рисунок 13 – Настройка системы

В случае если в поле, содержащее отправляемое сообщение, не указан текст, будет выведено пояснение пользователю (см. рис. 14).

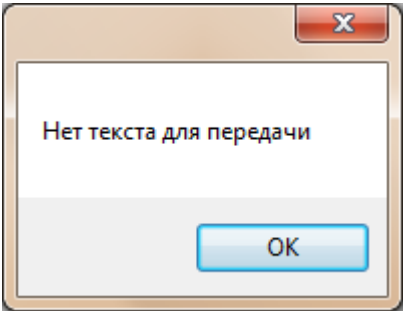


Рисунок 14 – Сообщение пользователю

Перв. примен.										
Справ. №										
<div> <div> <div>Закключение</div> <div> <p>В результате выполнения выпускной квалификационной работы спроектирована и программно реализована система обмена сообщениями с помощью звуковых сигналов. Для передачи сообщений с помощью звуковых сигналов использовалась дифференциальная фазовая манипуляция.</p> <p>Созданная система предназначена для обмена коротким текстовыми сообщениями: ссылками, номерами телефонов и т.д. Тестирование системы подтвердило её работоспособность и возможность использования для решения поставленной задачи.</p> <p>В дальнейшем на основе созданной системы планируется разработать её аналог для мобильных устройств: смартфонов, планшетов.</p> </div> </div> </div>										
Инв. № подл.	Подпись и дата				Инв. № дубл.					
	Взам. инв. №									
	Подпись и дата									
	Инв. № подл.									
Изм.	Лист	№ докум.	Подпись	Дата	<div> <div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div> <div>Лист</div> <div>29</div> </div>					

Перв. примен.	<p align="center">Список литературы</p> <p>1. Прокис Дж. Цифровая связь. – Пер. с англ. // Под ред. Д. Д. Кловского. – М.: Радио и связь, 2000. – 800 с. – ISBN 5-256-01434-X.</p> <p>2. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. – Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1104 с. – ISBN 5-8459-0497-8.</p> <p>3. Феер К. Беспроводная цифровая связь: методы модуляции. – Пер. с англ. // Под ред. В. И. Журавлёва. – М.: Радио и связь, 2000. – 520 с. – ISBN 5-256-01444-7.</p> <p>4. Джон Скит С# для профессионалов: тонкости программирования, 3-е издание, новый перевод = С# in Depth, 3rd ed.. – М.: «Вильямс», 2014. – 608 с. – ISBN 978-5-8459-1909-0</p> <p>5. Кристиан Нейгел и др. С# 5.0 и платформа .NET 4.5 для профессионалов = Professional С# 5.0 and .NET 4.5. – М.: «Диалектика», 2013. – 1440 с. – ISBN 978-5-8459-1850-5</p> <p>6. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд Язык программирования С#. Классика Computers Science. 4-е издание = С# Programming Language (Covering С# 4.0), 4th Ed. – СПб.:«Питер», 2012. – 784 с. – ISBN 978-5-459-00283-6</p>														
	Справ. №														
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.										
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<div>ВР-НГТУ-11ИСТ-5-14 ПЗ</div> <div>Лист</div> <div>30</div>
Изм.	Лист	№ докум.	Подпись	Дата											