СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc479240788)

[**1** ОБЗОР ЛИТЕРАТУРЫ 8](#_Toc479240789)

[**1.1** Цифровые аудио-станции и плагины 8](#_Toc479240790)

[**1.2** Октавер 9](#_Toc479240791)

[**1.2.1** Преобразование Фурье 10](#_Toc479240792)

[**1.2.2** Быстрое преобразование Фурье 11](#_Toc479240793)

[**1.3** Дилэй 12](#_Toc479240794)

[**1.4** Ревёрб 13](#_Toc479240795)

[**2** СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ 16](#_Toc479240796)

[**2.1** Структура программного модуля 16](#_Toc479240797)

[**2.2** Выбор программного средства 16](#_Toc479240798)

[**3** ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 19](#_Toc479240799)

[**4** РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 20](#_Toc479240800)

[**5** ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 21](#_Toc479240801)

[**6** РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 22](#_Toc479240802)

[**7** ЭКОНОМИЧЕСКАЯ ЧАСТЬ 23](#_Toc479240803)

[ЗАКЛЮЧЕНИЕ 24](#_Toc479240804)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc479240805)

[ПРИЛОЖЕНИЕ А 26](#_Toc479240806)

[ПРИЛОЖЕНИЕ Б 27](#_Toc479240807)

[ПРИЛОЖЕНИЕ В 28](#_Toc479240808)

# ВВЕДЕНИЕ

Развитие вычислительной математики и техники привело к тому, что звуки, используемые человечеством для повседневной жизни, стали переводиться из аналоговой формы в цифровую. Цифровой звук имеет перед аналоговым огромные преимущества, хотя нельзя забывать и об определенных его недостатках. Основной ценностью цифрового звука является возможность бесконечно долгого хранения и бесконечного тиражирования материала без потери исходного качества, тогда как у аналогового звука качество теряется при каждой записи-перезаписи. Немаловажно и то, что цифровая техника, в отличие от аналоговой, позволяет добиваться идентичности параметров систем при их массовом производстве и сохраняет эту идентичность при эксплуатации, в то время как характеристики аналоговых изделий обычно отличаются на разных экземплярах и ухудшаются со временем. Кроме того, облегчаются передача звука и его обработка современными цифровыми средствами, в первую очередь, специализированными компьютерами.

Программные модули синтеза и обработки звука широко используются в уже давно популярных жанрах электронной музыки, создавая звуки, непривычные человеческому уху, а потому необычные, интересные. Но это только самое очевидное и, конечно же, не единственное приложение. Прогресс приводит к тому, что становится возможным создание модулей обработки звука, эмулирующих настоящие звукозаписывающие студии с любым оборудованием: педалями эффектов, гитарными усилители и гитарными кабинетами, микрофонами, комнатами. Кроме того, каждый коммерческий и большинство некоммерческих музыкальных произведений проходит ряд обработок на стадии сведения и мастеринга, где исправляются некоторые дефекты записи, корректируются параметры для соответствия стандартам, добавляются эффекты и другое.

Ещё одним достоинством цифрового звука является возможность эмуляции любого аналогового прибора. В настоящее время любой реальный аналоговый прибор можно эмулировать с довольно высокой точностью, которая определяется в большей степени работой по сбору информации о приборе, нежели ограничениями компьютера. Более того, программные модули позволяют получить эффекты, которые невозможно создать с помощью аналоговых приборов. В настоящее время вычислительных ресурсов даже рядовых персональных компьютеров достаточно, чтобы запускать сразу несколько десятков программных модулей обработки звука в реальном времени.

Прогресс электротехники достиг такого уровня, что недостатки цифрового звука становятся ничтожными и меркнут перед достоинствами. Ведь уже несколько лет изготавливаются и широкодоступны АЦП и ЦАП с параметрами достаточными для того, чтобы разница между исходным аналоговым и оцифрованным звуком не была заметна человеческому уху.

Гибкость, удобство, универсальность, дешевизна программного обеспечения для обработки звука и хорошее качество цифрового звука практически не оставляют причин музыкантам, звукорежиссёрам и мастеринг-инженерам не переходить с аналогового оборудования на цифровое.

Таким образом, исходя из вышеперечисленных достоинств цифрового звука и его цифровой обработки, становится понятным, что создание программного обеспечения для обработки звука является актуальной темой. Этот факт, а также моя любовь к музыке обусловили выбор темы дипломного проекта. Цель проекта: разработать программный модуль обработки звуковой дорожки в реальном времени.

# **1** ОБЗОР ЛИТЕРАТУРЫ

## **1.1** Цифровые аудио-станции и плагины

Цифровая звуковая рабочая станция – электронная или компьютерная система, предназначенная для записи, хранения, редактирования и воспроизведения цифрового звука. Предусматривает возможность выполнения на ней законченного цикла работ, от первичной записи до получения готового результата. Плагины обычно выполняются в виде разделяемых библиотек [1]. Наиболее часто плагины для обработки звука используются в хост-программе DAW (Digital Audio Workstation), где они применяются к отдельным дорожкам или их группам. Примеры наиболее популярных DAW с кратким описанием в таблице 1.1.

|  |  |
| --- | --- |
| Таблица 1.1 – Примеры и описание наиболее популярных DAW | |
| DAW | Описание |
| FL Studio | Цифровая звуковая рабочая станция (DAW) и секвенсер для написания музыки. Музыка создаётся путём записи и сведения аудио- или MIDI-материала. |
| Cubase | Программное обеспечение для создания, записи и микширования музыки. Обладает мощным аудиодвижком и встроенными профессиональными инструментами |
| Nuendo | Отличается от другой популярной линии программных продуктов — Steinberg Cubase тем, что ориентирована не только на музыкантов, но и на любую профессиональную деятельность по озвучиванию в т.ч. фильмов, тв программ, реклам, радиопередач и прочего. |
| Ableton Live | Программа используется как для студийной работы (аранжировка, сведение), так и для живой игры (импровизация, DJ-инг), и имеет два режима: «Arrangement View» и «Session View». |
| REAPER | Аудиостанция обладает широкими функциональными возможностями и является развитой, профессиональной рабочей системой для создания, записи, редактирования и микширования аудио и MIDI материала, а также мастеринга композиций. При этом программа имеет относительно небольшие размеры. |
| Logic Pro X | Logic Pro X включает огромную коллекцию высококачественных музыкальных сэмплов, инструментов, эффектов и циклов — всё, что нужно для создания композиций профессионального уровня. |

В программах обработки звука плагины выполняют обработку и создание звуковых эффектов, например, мастеринг, применение эквалайзера и сжатие динамического диапазона [2]. Некоторые плагины изменяют технические характеристики звука: глубину, частоту дискретизации и прочее. Практически все аудио-плагины имеют графический пользовательский интерфейс. В GUI присутствуют элементы управления (чаще всего это какие-нибудь ручки), меняющие то, как плагин обрабатывает входящие данные. Часто плагин или standalone приложение уже имеет встроенный набор пресетов, в которых хранятся положения ручек и других параметров. Также имеется возможность сохранять свои собственные пресеты. Наиболее распространённый формат аудио-плагинов – VST.

Обычно виды обработок аудио дорожки классифицируют образом, отражённым в таблице 1.2.

|  |  |
| --- | --- |
| Таблица 1.2 – Типы и виды обработок звука | |
| Тип обработки | Вид обработки |
| Частотная | * фильтр, эквалайзер, питч-шифтер |
| Динамическая | компрессор, лимитер, софтклиппер, эспандер. |
| Модуляционная | * хорус, флэнжер, фэйзер. |
| Частотно-динамическая | многополосный компрессор. |
| Пространственная | дилэй, эхо, реверберация. |
| Искажения | овердрайв, дисторшн, фузз. |

Программные модули для обработки звука могут работать либо с стерео, либо с моно дорожкой, либо и с той и другой.

Virtual Studio Technology (VST) — формат ресурсозависимых (native) плагинов реального времени, которые подключаются к звуковым редакторам и музыкальным редакторам, секвенсорам и т. д. Формат был разработан совместно Propellerhead и Steinberg, впоследствии Propellerhead отказался от дальнейших работ над VST и дальнейшая разработка осуществлялась исключительно Steinberg. В настоящее время в этом формате существуют тысячи плагинов, он стал одним из самых распространённых для звуковых программ. Приложения VST отличаются от плагинов DirectX по нескольким параметрам; в частности, они существуют для Windows, Mac OS X и Linux. Кроме того, в отличие от ранних версий DirectX, плагины VST обладают развитым интерфейсом автоматизации [3].

В данном дипломном проекте будет разрабатываться программный модуль обработки моно дорожки с использованием следующих основных эффектов: октавер, дилэй и ревёрб. Также вполне вероятно, что будут реализованы и некоторые другие обработки. Они будут упоминаться в разделе 4.

## **1.2** Октавер

Октавер – звуковой эффект или соответствующее устройство, добавляющее к сигналу его копию на октаву или две ниже или выше основного тона. Само обозначение применяется преимущественно к обработке звука электрогитары. Октавер является частным случаем питч-шифтера [4]. При использовании октавера создаётся впечатление, что вместо одного инструмента играют два в разных октавах. В некоторых педалях эффектов, таких, как, например, Boss OC-2, имеется возможность добавлять два дополнительных тона одновременно – один на октаву ниже основного, другой на две октавы ниже основного. При этом каждый из них можно регулировать по громкости, смешивая в желаемой пропорции. Большинство октаверов монофонические, т.е. не могут достраивать октавы к аккордам. Полифонические (цифровые) октаверы могут обрабатывать сигнал, содержащий несколько нот (например, Electro-Harmonix POG).

Для реализации этого эффекта необходимо получить представление звукового сигнала в частотной области, затем обработать и перевести во временную область.

### **1.2.1** Преобразование Фурье

Дискретное преобразование Фурье (ДПФ) – одна из двух наиболее распространённых и мощных процедур цифровой обработки сигналов (другая процедура – цифровая фильтрация). ДПФ позволяет анализировать, преобразовывать и синтезировать сигналы такими способами, которые невозможны при непрерывной (аналоговой) обработке [5].

ДПФ – это математическая процедура, используемая для определения гармонического, или частотного состава дискретных сигналов. Хотя для нас дискретный сигнал представляет набор значений, полученных в результате периодической дискретизации непрерывного сигнала во временной области, мы увидим, что ДПФ полезно для анализа любых дискретных последовательностей, независимо от того, что на самом деле эти последовательности представляют. Истоком ДПФ, конечно же, является непрерывное преобразование Фурье X(f), которое определяется как

|  |  |
| --- | --- |
| , | (1.1) |

где – некоторый непрерывный сигнал во временной области.

В области обработки непрерывных сигналов формула (1.1) используется для преобразования аналитического выражения для непрерывной временной функции в непрерывную функцию X(f) в частотной области. Последующее вычисление значений выражения даёт нам возможность определить частотный состав любого сигнала.

С приходом в нашу жизнь цифровых компьютеров усилия пионеров цифровой обработки привели к разработке ДПФ, которое определяется как дискретная последовательность X(m) в частотной области:

|  |  |
| --- | --- |
| , | (1.2) |

где – дискретная последовательность значений, полученных дискретизацией во временной области непрерывной переменной ;

– основание натурального логарифма;

– константа;

– мнимая единица.

Каждый выходной отсчёт ДПФ представляет собой сумму почленных произведений входной последовательности отсчётов сигнала на последовательность отсчётов комплексной синусоиды (гармоники) вида . Точные значения частоты разных синусоид зависят как от частоты дискретизации , с которой был дискретизирован исходный сигнал, так и от количества отсчётов N. Для АЦП, преобразующих сигнал с источников звука, характерны следующие частоты дискретизации: 44.1 кГц, 48 кГц, 88.2 кГц, 96 кГц. Например, если мы дискретизируем непрерывный сигнал с частотой 48000 отсчётов в секунду (Гц), а затем выполняем 128 точечное ДПФ дискретизированных данных, то основная частота синусоид будет равна Гц. Все частоты, соответствующие , кратны основной частоте: отсчёт соответствует частоте Гц, отсчёт соответствует частоте Гц и так далее до отсчёта . N разных частот анализа ДПФ определяются выражением

|  |  |
| --- | --- |
|  | (1.3) |

Нас интересует как амплитуда, так и фаза каждого отсчёта . Если представить произвольный отсчёт ДПФ как сумму действительной и мнимой частей

|  |  |
| --- | --- |
| , | (1.4) |

то амплитуда вычисляется как

|  |  |
| --- | --- |
| , | (1.5) |

а фазовый угол вычисляется как

|  |  |
| --- | --- |
| . | (1.6) |

Чтобы получить исходный сигнал во временной области необходимо выполнить обратное дискретное преобразование Фурье (ОДПФ). Выражение для ОДПФ имеет вид:

|  |  |
| --- | --- |
| . | (1.7) |

### **1.2.2** Быстрое преобразование Фурье

Хотя ДПФ представляет собой наиболее простую математическую процедуру определения частотного состава временных последовательностей, оно ужасно не эффективно. В 1965 году была опубликована статья Кули и Тьюки, описывающая очень эффективный алгоритм реализации ДПФ. Этот алгоритм сегодня известен как быстрое преобразование Фурье (БПФ). Разработка этого алгоритма изменила цифровую обработку сигналов, сделав доступной всю мощь анализа Фурье.

Алгоритм БПФ по основанию 2 – это очень эффективный алгоритм вычисления ДПФ, когда длина ДПФ равна натуральной степени двух.

Из-за специфической формы графа он получил название «бабочка». Данная процедура объединения является основной при построении алгоритмов БПФ по основанию два. На рисунке 1.1 представлен полный граф алгоритма БПФ с прореживанием по частоте для N=8 [6].

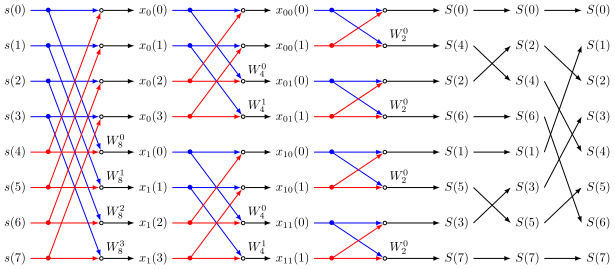


Рисунок 1.1 – Полный граф алгоритма БПФ с прореживанием по частоте для .

Операция бабочки выполняется в соответствии с формулами 1.8 и 1.9.

|  |  |
| --- | --- |
| ; | (1.8) |

|  |  |
| --- | --- |
| . | (1.9) |

## **1.3** Дилэй

Дилэй – звуковой эффект или соответствующее устройство, имитирующие чёткие затухающие повторы (эхо) исходного сигнала. Эффект реализуется добавлением к исходному сигналу его копии или нескольких копий, задержанных по времени. Под дилэем обычно подразумевается однократная задержка сигнала, в то время как эффект «эхо» - многократные повторы. По принципу действия является частным случаем ревербератора. Отличие заключается в том, что дилэй имеет одну линию задержки и больший временной интервал (не менее 50-60 мс), который позволяет отделить оригинальный звук от эффекта на слух. Эффект дилэя не встречается в природе и носит рукотворный характер [7].

Дилэй – довольно простой эффект для реализации. Входной сэмпл сигнала сохраняется в буфер и суммируется с прошлым сэмплом, отстающим в буфере на некоторое число N сэмплов. Обычно прошлый сэмпл ослабляется. И ослабление и длина задержки (N) обычно управляются пользователем [8]. На рисунке 1.2 изображена схема простого эффекта дилэя.

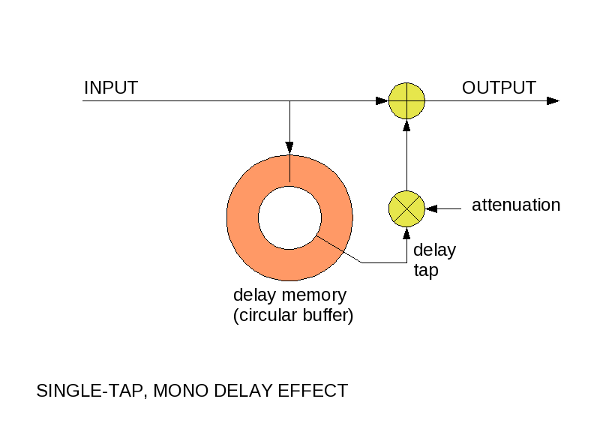


Рисунок 1.2 Схема простого дилэя.

Параметры эффекта:

* обратная связь (feedback, regeneration) – при отсутствии обратной связи на выходе будет одна задержка, при увеличении её значения растёт и количество сигналов на выходе;
* время задержки (delay, time) – промежуток времени между исходным сигналом и его задержкой (задержками);
* баланс (balance, mix) – соотношение исходного и задержанного сигналов.

Основные типы дилэя:

* slapback – одиночная задержка длительностью до 120 мс;
* echo – более длительная задержка с обратной связью;
* reverse – в цифровых эффектах возможно воспроизведение записанного в буфер сигнала в обратном порядке;
* ping pong delay – поочерёдное панорамирование задержанного сигнала в левый/правый канал.

Планируется реализовать дилэй, с задержкой от 20 до 1000 мс и синхронизацией с темпом хост-программы, с возможностью выбора режима «reverse».

## **1.4** Ревёрб

Реверберация – это процесс постепенного уменьшения интенсивности звука при его многократных отражениях. Иногда под реверберацией понимается имитация данного эффекта с помощью ревербераторов [9].

Явление реверберации состоит в суперпозиции различных эхосигналов от одного источника звука. Эффект реверберации можно наблюдать в закрытых помещениях после выключения источника звука. Обычно избыточная длительность реверберации приводит к неприятной гулкости, «пустоте» помещения, а недостаточная – к резкому отрывистому звучанию, лишённому музыкальной полноты. Искусственно создаваемая реверберация в определённых пределах способствует улучшению качества звучания, создавая ощущение приятного «резонанса» помещения.

Реверберацию можно разделить на два компонента, которые видны на представлении импульсного отклика в комнате:

1. Ранние отражения – первые отражения, которые мы слышим в течение приблизительно 100 мс вместе с прямым звуком от источника.
2. Поздняя реверберация – слышимый звук после 100 мс, до момента его полного затухания. Поздняя реверберация характеризуется плотной текстурой рассеянных отражений, которые достигают наших ушей несколькими путями. Эти рассеянные отражения находятся не в фазе относительно друг друга, что вызывает эффект гребенчатой фильтрации. Мы воспринимаем этот эффект как «атмосферу», характер места [10].

Структурная схема обработки сигнала ревербератором показана на рисунке 1.3. Такое решение предложил звуковой инженер, музыкант и доктор наук Стэнфордского университета Джеймс Энди Мурер, усовершенствовав алгоритм Шрёдера [11]. Исходя из Алгоритма Мурера, к исходному сигналу добавляется сигнал, прошедший через сеть ранних отражений, состоящую из множества дилэй-линий. Далее к нему добавляется сигнал обработанный сетью поздних отражений, состоящей из параллельных гребенчатых фильтров и фазового фильтра с дилэй-линией.

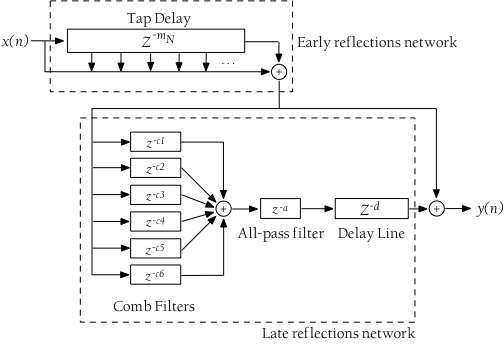


Рисунок 1.3 – Структурная схема ревербератора Мурера.

Гребенчатый фильтр — в обработке сигналов электронный фильтр, при прохождении сигнала через который к нему добавляется он сам с некоторой задержкой. В результате получается фазовая компенсация. АЧХ гребенчатого фильтра состоит из ряда равномерно распределённых пиков, так что она выглядит как гребёнка.

В цифровых системах, фильтр задаётся формулой 1.10:

|  |  |
| --- | --- |
| , | (1.10) |

где – коэффициенты;

– запаздывание сигнала;

– входной сигнал;

– выходной сигнал.

В обзоре литературы обычно содержится краткий анализ литературных

источников различных типов, использованных в процессе работы над дипломным проектом. Здесь приводятся основные сведения, почерпнутые из литературы. Возможен анализ патентной чистоты.

# **2** СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Разрабатываемый программный модуль следует разбить на отдельные логически взаимосвязанные блоки, что является необходимым условием для обеспечения гибкости его структуры. При данном подходе становится возможной выборочная модернизация отдельных частей программного кода, с минимальным влиянием на остальные части проекта, либо, в идеальном случае, вовсе без их изменения.

## **2.1** Структура программного модуля

Для определения логических модулей необходимо в первую очередь определить основные стадии обработки звука программным модулем:

* обработка эффектом «октавер»;
* обработка эффектом «дилэй»;
* обработка эффектом «ревёрб».

Сигнал будет проходить эти три стадии от одной к другой в указанной выше последовательности, претерпевая обработку, соответствующую текущей стадии. Для каждой стадии на структурной схеме необходимо выделить соответствующий блок. Также необходимо последовательно соединить эти блоки.

Пользователь должен иметь возможность вносить изменения в параметры эффектов обработки, которые будут влиять на работу трёх вышеуказанных блоков эффектов. Для этих целей необходимо выделить блок интерфейса пользователя и блок параметров обработки. В соответствии с методологией системного подхода в разработке архитектуры, программный модуль разбивается на совокупность сущностей, представленных на структурной схеме (см. чертеж ГУИР.400201.161 С1).

Таким образом, структурная схема будет состоять из следующих блоков:

* обработка эффектом «октавер»;
* обработка эффектом «дилэй»;
* обработка эффектом «ревёрб»;
* блок параметров обработки;
* блок интерфейса пользователя.

*Блок эффекта «октавер»* первым принимает на обработку исходный звуковой сигнал. Он предназначен для добавления гармоник частотой степени двойки к исходному звуковому сигналу. Внутри него с помощью дискретного преобразования Фурье получается сигнал в частотном домене. Далее сигнал обрабатывается и с помощью обратного дискретного преобразования Фурье переводится во временной домен. Затем обработанный сигнал складывается с исходным сигналом в задаваемой пользователем пропорции и подаётся на следующий логический блок.

*Блок эффекта «дилэй»* создаёт копию поступившего на вход сигнала и добавляет её к исходному через некоторое задаваемое пользователем время. Сигнал с выхода снова подаётся на вход этого блока уже с уменьшенным уровнем. Также внутри этого блока может происходить обработка сигнала, подаваемого с выхода на вход. Например, может быть реализована фильтрация низких частот. С выхода блока эффекта «дилэй» сигнал поступает на блок эффекта «ревёрб».

*Блок эффекта «ревёрб»* отвечает за эмуляцию постепенного уменьшения интенсивности звука при его многократных отражениях. Он будет состоять из двух компонентов: блока ранних отражений и блока поздних отражений.

*Блок интерфейса пользователя* представляет собой совокупность средств, при помощи которых пользователь взаимодействует с программным модулем обработки звуковой дорожки в реальном времени. Для построения интерфейса будет использованы компоненты, встроенные в фреймворк Juce (о фреймворке Juce написано в подразделе 2.2). Данный блок отвечает за получения ввода пользователя и связан лишь с блоком параметров обработки двусторонней связью.

*Блок параметров обработки* предназначен для хранения параметров, влияющих на работу блоков эффектов. Среди таких параметров обязательно будут присутствовать следующие:

* соотношение обработанного сигнала к необработанному (для всех эффектов);
* время задержки (для эффектов дилэй и ревёрб);
* громкость сигнала подаваемого с выход на вход (для эффекта дилэй).

Данный блок кроме двусторонней связи с блоком интерфейса пользователя имеет исходящие односторонние связи с каждым из блоков эффектов обработки звуковой дорожки.

## **2.2** Выбор программных средств

В задании в качестве языка программирования задан язык C++, а в качестве среды разработки – Visual Studio 2012 Express. Для выполнения поставленной задачи необходимо выбрать фреймворк, который бы соблюдал требований задания, избавил от низкоуровневого программирования, включающий компоненты графического интерфейса пользователя, а также компоненты для работы со звуком. Фреймворк Juce удовлетворяет всем вышеперечисленным требованиям.

Juce – это открытый кроссплатформенный инструментарий разработки ПО (фреймворк) для языка C++, используемый для разработки GUI приложений и плагинов.

Цель Juce – позволить компилировать один и тот же исходный текст в программы, одинаково работающие на Windows, Mac OS X и Linux (последние версии – также iPhone и Android) платформах. Он поддерживает различные среды разработки и компиляторы, такие как GCC, Xcode и Visual Studio.

Juce впервые опубликован в 2004, держателем его кода является британская компания Raw Material Software. Имеет двойную GPL/коммерческую лицензию.

Подобно многим другим фреймвокам (напр. Qt, wxWidgets, FLTK и другим), Juce содержит классы, позволяющие программе работать с графикой и звуком, разбирать XML, работать с сетью и криптографией. За счёт этого нуждающиеся в дополнительных библиотеках программисты могут использовать только библиотеку Juce, или хотя бы сократить количество сторонних библиотек, которые они используют. На это разработчиков Juce вдохновил JDK языка Java. По их заявлению, они собирались из Juce сделать «что-то подобное для С++».

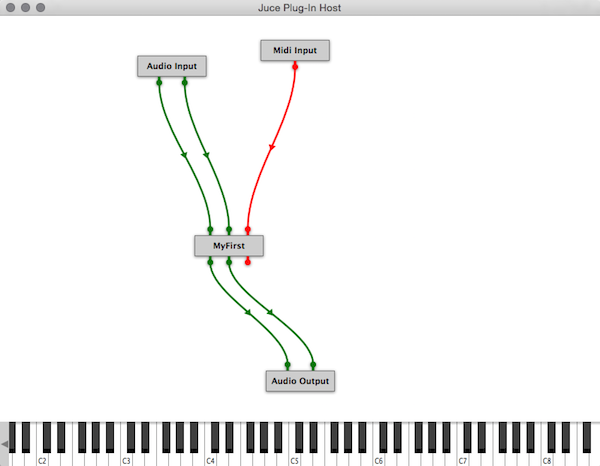


Рисунок 2.1 – Интерфейс хост-программы Juce.

Примечательная особенность Juce по сравнению с другими аналогичными фреймворками — большой набор аудиофункций. Дело в том, что библиотека Juce сначала была разработана как часть аудиосеквенсора Tracktion, и лишь затем стала самостоятельным продуктом. Juce включает в себя поддержку воспроизведения звука через аудио и MIDI интерфейсы, полифонические синтезаторы, понимает файлы распространённых аудиоформатов (таких как WAV, AIFF, FLAC, и Vorbis). Он также содержит интерфейсы-оболочки для построения различных аудио плагинов, таких как эффекты и инструменты VST. Это привело к его широкому распространению в сообществе разработчиков аудио-ПО.

В поставку Juce входят классы-обёртки для создания как аудиоплагинов, так и браузерных плагинов. При сборке аудиоплагина, получается единый бинарный файл, который поддерживает несколько форматов плагинов (VST, RTAS, AU). Поскольку весь платформо- и форматозависимый код содержится в классах-обёртках, то пользователь может собирать плагины в формате VST/RTAS/AU для макинтошей и Windows из одного и того же исходного кода.

Имеется также неофициальное ответвление библиотеки, расширенное дополнительными возможностями, поддерживаемое сообществом, которое называется Juced.

На сайте этого варианта фреймворка можно найти также дополнительную документацию по Juce, которая поможет освоить библиотеку. Последняя версия фреймворка на данный момент – 4.3.

Фреймворк Juce имеет хост-программу в которой можно проверять работу плагина, а также производить отладку, что очень важно. Её интерфейс изображён на рисунке 2.1.

Для разработки VST плагинов необходимо скачать пакет VST SDK с официального сайта Steinberg – разработчика этого формата. Текущая версия этого средства разработки – 3.6.7.

Также при разработке программного модуля обработки звуковой дорожки в реальном времени будет использоваться система управления версиями – Git. Выбор сделан в пользу именно этой системы управления версиями из-за ряда причин:

1. Git использует распределённую модель вместо традиционной клиент серверной. Она не нуждается в централизованном хранилище: вся история изменения документов хранится на каждом компьютере в локальном хранилище. При необходимости локальный репозиторий может быть передан командой push на удалённый репозиторий.
2. Постоянное подключение к сети не является необходимым.
3. Git обладает высокой производительностью.
4. Продуманная и уже знакомая мне система команд.

А в качестве сервиса предоставляющего хостинг для размещения git-репозиториев выбран веб-сайт GitHub.

Системное проектирование – аналог разработки структурной схемы для аппаратных дипломных проектов. В этом разделе на основе системного подхода определяется обобщенная структура пакета программ, программы или модуля, описывается назначение выделенных блоков, а также связи между ними. Если программные средства разработки не заданы, то производится их выбор. Данный раздел должен сопровождать чертеж схемы структурной либо схемы работы системы и содержать ссылки на него.

# **3** ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Функциональное проектирование – аналог соответствующего раздела аппаратного дипломного проекта. Это основной раздел ПЗ, дающий ключ к пониманию функционирования разрабатываемой программы и исчерпывающую информацию о ее структуре с точки зрения описания данных и обрабатывающих их подпрограмм – функций и процедур. Обработка данных, как известно, является основной целью работы любой программы. Поэтому здесь описываются заданные константы, пользовательские переменные, внутренние и внешние массивы и так далее. При определении необходимости в какой-либо обработке данных вводится соответствующая подпрограмма. При использовании объектно- ориентированного подхода могут описываться структура и взаимоотношения между классами со ссылками на чертеж диаграммы классов. Если при программировании (особенно в системах визуального программирования) используются стандартные библиотеки, то упор должен делаться на описание самостоятельно разработанных фрагментов кода. Стандартные функции лишь упоминаются. При использовании БД описывается структура таблиц и связи между ними со ссылками на чертеж модели данных. Здесь же раскрываются и потоки данных между подпрограммами, то есть входные и выходные аргументы процедур и функций. Рекомендуется структурировать раздел в соответствии с блоками, выделенными на этапе системного проектирования. Данный раздел должен сопровождать чертеж диаграммы последовательности и содержать ссылки на него. Могут быть ссылки на чертеж схемы данных.

# **4** РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Разработка программных модулей – аналог разработки принципиальной схемы аппаратного дипломного проекта. В этом разделе подробно описываются уже внутренние алгоритмы ключевых процедур и функций с разбиением на отдельные подразделы. Здесь же описывается реализация наиболее интересных алгоритмов, например, алгоритмов шифрования. Данный раздел может сопровождать чертежи схем программ и

содержать ссылки на них.

# **5** ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

В разделе, посвященном программе и методике испытаний,

описываются внутренние (если самотестирование заложено в программу) и внешние средства тестирования. Могут использоваться как оригинальные, так и стандартные тесты. Рассматриваются способы проверки надежности (устойчивости, стабильности и так далее) разработанной программы в различных режимах, включая многопользовательский и многозадачный режимы, а также корректность обработки входных, промежуточных и выходных данных, в том числе: в области граничных значений допустимых диапазонов, заведомого неправильных данных, файлов большого размера и так далее. Для каждого из тестов приводятся исходные данные, параметры и результаты.

# **6** РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В руководстве пользователя дается описание работы с программой. Указываются требования к аппаратному (процессор, объем ОЗУ, объем дискового пространства и так далее) и программному (ОС, необходимое ПО, драйверы и так далее) обеспечению. Описывается процесс инсталляции с указанием каталогов, ключей реестра, конфигурационных файлов и так далее. Также описывается пользовательский интерфейс с указанием элементов управления (пунктов меню, кнопок, закладок и так далее), режимов работы и последовательности действий. Здесь могут приводиться скриншоты работы программы.

# **7** ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Экономика

# ЗАКЛЮЧЕНИЕ

В Результате

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wikipedia [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Цифровая звуковая рабочая станция.
2. Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Плагин>.
3. Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Virtual_Studio_Technology>.
4. Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Октавер>.
5. Лайонс, Р. Цифровая обработка сигналов: Второе издание/ Лайонс Р. – Пер с англ. – М. : ООО «Бином-Пресс», 2006 г. – 656 с.
6. DSPLIB [Электронный ресурс]. – Режим доступа: <http://ru.dsplib.org/content/fft_dec_in_time.html>.
7. Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Дилэй>.
8. Audio Artillery [Электронный ресурс]. – Режим доступа: <http://audioartillery.com/projects/tonecore_dsp_dev_kit_guide>.
9. Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Реверберация>.
10. Open Source Acoustic Design [Электронный ресурс] : Algorithmic Reverberation. – Режим доступа: http://arqen.com/wp-content/docs/Hybrid-Convolution-Algorithmic-Reverb.pdf.
11. Christianfloisand wordpress [Электронный ресурс]. – Режим доступа: <https://christianfloisand.wordpress.com/2012/10/18/algorithmic-reverbs-the-moorer-design>/.

* Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Juce>.
* Juce [Электронный ресурс]. – Режим доступа: <https://www.juce.com/doc/tutorial_create_projucer_basic_plugin>.

# ПРИЛОЖЕНИЕ А

(обязательное)

Исходный код некоторых алгоритмов программного модуля

# ПРИЛОЖЕНИЕ Б

(обязательное)

Спецификация программного дипломного проекта

# ПРИЛОЖЕНИЕ В

(обязательное)

Ведомость документов