

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерної інженерії

**Пояснювальна записка**

до кваліфікаційної роботи на ступінь вищої освіти магістр

на тему: **«МЕТОДИКА СПРОЩЕННЯ СПОСОБУ  
ЗВУКОЗАПИСУ ТА ОБРОБКИ ЗВУКУ»**

Виконав: студент 6 курсу, групи КСДМ–61  
спеціальності

123 Комп'ютерна інженерія  
(шифр і назва спеціальності)

Сотников А.Д.

(прізвище та ініціали)

Керівник Торошанко Я.І.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

## ВСТУП

Дорога, що веде нас від фонографа Томаса Едісона із олов'яного циліндра та пружної мембрани до сьогоднішніх цифрових аудіосервісів переповнена чудовими винаходами та іноваціями. Наші минулі досягнення сприяють тому, чим ми є сьогодні, і вказують на майбутнє як нескінченний пошук розширення рамок можливого в аудіовсесвіті.

Сучасна студія звукозапису зазнала багато змін від її давніх попередників. Технології, якими ми володіємо в сьогодні, дають нам змогу створювати будь-яку музику, не виходячи навіть з дому. Професійне обладнання, звісно, обійдеться для таких цілей в чималу копійку, проте доволі якісний матеріал можна випускати, навіть не маючи при собі музичних інструментів, використовуючи лише комп'ютерні програми, що відтворюють їх звук.

Величезна кількість передового музичного програмного забезпечення дають кінцевому користувачеві можливість працювати у всіх існуючих жанрах, використовувати необмежену кількість плагінів та звукових ефектів, обробляти та видозмінювати кожен мілісекунду сигналу на аудіодоріжці. Важко повірити, але сучасні світові хіти створюються лише в комп'ютерних програмах, на кшталт Fruitti Loops або Cubase, якими користуються, як і відомі артисти та їх аудіоінженери з великим студіями та дорогим обладнанням, так і музиканти-аматори, у яких наявності лише декілька дешевих аудіоінтерфейсів, MIDI-клавіатура, музичні інструменти та доступ до Windows або MacOS.

Такі гіганти цифрової музичної індустрії як SONY, Fender, Marshall є наразі передовими майстрами у створенні обладнання для якісного запису та відтворення музики. Їхня апаратура присутня майже у кожному професійному залі для звукозапису не один десяток років. А це мікрофони, підсилювачі, аудіо-інтерфейси, навушники, студійні монітори, кабелі та поп-фільтри. Сучасний аудіоінженер звісно прискіпливо ставиться до якості виробленого матеріалу, і в умовах музичної студії, процес звукозапису проходить декілька етапів, перш ніж користувач почує

фінальний продукт. Звісно сьогоднішні технології економлять витрачений час на обробку звукових доріжок, проте ця робота всеодно потребує ретельної уваги.

Донині немає такого апарату, який міг би передати дух живого концерту. Можливо скрупульозно точно записати звук, розкласти його на безліч каналів і передати за секунди на гігантські відстані, але відтворити харизму музиканта, масовий екстаз від виступу популярної групи або відлуння звуків інструментів із підсилювачів, так само важко, як заново створити Землю.

## РОЗДІЛ 1. ОГЛЯД ТЕХНОЛОГІЙ АУДІОЗАПИСУ

### 1.1 Перші технології звукозапису

На відміну від ідеї, що може прийти в голову будь-якому винахіднику, розробка цієї ідеї і подальша модернізація із розвитком паралельних технологій, відбувається занадто повільно і займає велику кількість часу, перш ніж більш спрощені методи або шляхи будуть представлені. Шлях аудіовсесвіту не є виключенням, оскільки занадто багато часу пройшло від моменту представлення першого у світі фонографа до запису мультидоріжок, CD-дисків, та врешті-решт, повної цифровізації аудіо.

На сьогоднішній день технології аудіо розвиваються у стрімкому темпі і щодня виходить велика кількість рішень для тих чи інших задач, пов'язаних із записом, обробкою, накладенням та публікації аудіо-інформації. Деякі із них наслідують минулі розробки, деякі представляють абсолютно новітню технологію. Тому, для будь-кого заохоченого у праці із аудіо, зараз існує неймовірна кількість інструментів та шляхів розвитку, а процеси запису та обробки повністю комп'ютеризовані і потребують лише знань теорії та досвіду роботи із аудіо-програмами на ПК.

150 років тому, у людей, які думали над можливістю записати та відтворити записане, думок про майбутню комп'ютеризацію усього процесу ще не було. Ідея розроблювалась не один десяток років, а стартовою точкою вважається винахід Томаса Едісона під назвою фонограф.

У Європі назва «фонограф» застосовувалася настільки широко, що стала називною. У США, однак, «Фонограф» був торговельною маркою, яка широко не використовувалася після 1901 року. Тому в американському діалекті англійської мови словом англ. phonograph називаються всі пристрої механічного запису та відтворення звуку.



Рисунок 1.1 – Фонограф Томаса Едісона

Хоч, як правило, саме Томасу Едісону приписують повністю цей винахід, він лише став першим, хто представив першу робочу модель і подав заяву на патент, зародивши цим «акустичну еру» в сфері аудіо-інженерії в 1877 році. Перший фонограф складався з обертового циліндра, покритого олов'яною фольгою. Запис здійснювався за допомогою різка, схожого на мегафон, або звукового конуса, прикріпленого до стилуса або «голки». Звуки, вловлені рогом, викликали вібрування стилуса, створюючи відбитки в м'якій фользі під час його обертання. Для відтворення запису у створені канавки було поміщено такий самий стилус, і коли циліндр обертася, вібрації механічно посилювалися рупором.

Запатентувавши фонограф, Едісон більше не займався цією ідеєю, зосередивши свої зусилля на винаході лампочки, поки на ринок не почали з'являтися конкуруючі пристрої. У 1887 році Еміль Берлінер запатентував грамофон. Грамофон використовував плоский вініловий диск, а не циліндр, і таким чином розпочав першу війну форматів аудіозапису.



Рисунок 1.2 – Грамофон Е. Берлінера

Раннє вдосконалення циліндра запису було зроблено Олександром Гремом Беллом та його братом. Вони замінили олов'яну фольгу на циліндрі воском, створивши таким чином «восковий циліндр». Однією з перших проблем з циліндрами була складність масового виробництва. Циліндри були відтворені шляхом перезапису головного циліндра на додаткові циліндри. Однак таким чином можливо було зробити лише близько 25 перезаписів, перш ніж оригінальний циліндр став занадто зношеним, щоб грати. Щоб отримати додаткові копії, виконавцям потрібно було повторити запис кілька разів, що було досить трудомісткою і дорогою процедурою. Нарешті в 1902 році був розроблений метод масового виробництва циліндрів, що значно знизило їх вартість.

Теоретично якість звуку циліндра мала бути набагато кращою, ніж у диска через його постійну лінійну швидкість канавки та більший динамічний діапазон, але насправді різниці було мало. Однак диски можна було б легко і дешево виробляти масово. Для формування копій на тверді гумові диски використовувався майстер-диск. (Спочатку диск записувався лише на одній стороні. Пізніше хтось придумав записувати з обох сторін, створюючи сторону «В» або «зворотню» сторону). Диски також мали перевагу в тому, що їх було легше зберігати, і з них можна було формувати альбоми. Диски зрештою виграли війну форматів, і за двадцять років циліндр майже зник.

Незважаючи на цей досить грубий метод запису, протягом наступних двох десятиліть якість звуку повільно покращувалася, створюючи менше шумів і покращуючи точність. У 1925 році закінчилася акустична ера і почалася ера «електричного звукозапису». Замість рупора для вловлювання звуку використовувався мікрофон, який електрично посилювався і приводився до токарного верстата для вирізання канавок на диску. Цей процес призвів до значного покращення якості звуку. Того року змінилися всі основні лейбли, а за кілька років — майже всі. Проте виступи все одно записувалися безпосередньо на диск.

Електричний процес покращив гнучкість. Хоча спочатку оркестри записували лише за допомогою одного мікрофона, незабаром інженери почали використовувати кілька мікрофонів. З появою мікшера інженер тепер мав більший контроль над звуком і міг змішувати голоси та інструменти. Проте досі не було можливостей редагування. Тож, якщо виконавець зробив помилку або запис мав поганий звук, доводилось переписувати все заново на новому диску.

Робота над магнітними записувачами почалася на початку 1900-х років, спочатку з дротяних самописців. Ці машини використовували дуже тонкий сталевий дріт, який намагнічувався, коли він проходив над головкою запису. Диктофони страждали від поганої якості звуку, важких котушок і важкого дроту. Згодом вони були замінені магнітною стрічкою, смужками пластику, покритими тонко подрібненим залізом (яке пізніше було замінено порошкоподібним оксидом заліза). Лише в середині 1940-х років якість запису через магнітофон зрівнялась і навіть перевершила технологію запису на вініловий диск.

Почувши демонстрацію магнітофона в 1944 році, Бінг Кросбі був вражений якістю звуку. Він також побачив великий потенціал цього нового медіа. У той час майже вся музика на радіо виконувалася наживо. Протягом двох років Кросбі переконав мережу дозволити йому попередньо записати своє радіошоу за допомогою магнітної стрічки. Він також став першою зіркою звукозапису, яка освоїла комерційні записи на плівку.



Рисунок 1.3 - Магнітофон “Амрех 200”

Кросбі був настільки вражений процесом, що інвестував 50 000 доларів в компанію Ampex Electric and Manufacturing Company (яка пізніше стала Ampex Corporation), головну силу в ранньому комерційному записі на касету. Завдяки такому вливанню капіталу, Амрех через декілька років розробив магнітофон для запису на 2, 3 і навіть 4 доріжки.

Магнітофони в студії звукозапису представили інновації, які кардинально змінили спосіб створення музики. Серед іншого вперше можна було відредагувати музичну композицію. Інженер міг вирізати та вставляти — буквально відрізати шматок стрічки з однієї композиції і вставляти в іншу. Стрічка зробила музику набагато гнучкішою. Тепер було легко обрізати сторонні звуки, змішувати звуки з більш ніж одного джерела, регулювати гучність на будь-якому моменті запису. Якщо раніше виконавцю доводилося записувати пісню кілька разів, поки він чи вона не отримали ідеальний дубль, тепер можна було з'єднати два недосконалих дублі, щоб отримати один хороший.

Наступним великим прогресом у магнітофонному записі став «мультитрекінг». Чотириколіїні магнітофонні апарати дозволили інженеру записувати різні речі на різних доріжках. Наприклад, він міг записати вокал на одному треку, барабани на іншому, а інші інструменти на інших треках. Ця



процедура дала йому можливість змішувати різні треки після запису. Раніше все змішування проводилося на одній доріжці. Це також дозволяло перезаписувати одну партію окремо від інших. Наприклад, якщо вокаліст зробив помилку під час дубля, він чи вона могли просто перезаписати вокальний трек. Інструменталісту не потрібно було перезаписувати свою партію.



Рисунок 1.4 – Звукозаписуюче обладнання АМРЕХ для 2-х, 4-х і 16-доріжкового запису

Мультитрекінг також привів до багатьох нових інновацій. На початку 1950-х років Лес Пол (кому також приписують розробку електрогітари) замовив у Амрех першу восьмиколійну машину для власних експериментів. Пол був одним із перших виконавців, які записували додаткові треки після створення оригінальних треків (процес, який став відомий як накладення або overdubbing).

Цифрова ера – цифрові методи запису були запроваджені в 1980-х роках. Протягом наступного десятиліття цифровий запис поступово витісняв аналогову стрічку. Справжні успіхи в цифровому записі відбулися на початку 1990-х років, коли ємність жорстких дисків комп'ютера і швидкість комп'ютерного ЦП значно

зросла. Сьогодні майже всі великі студії звукозапису працюють в цифровому режимі, хоча деякі все ще підтримують магнітні стрічки для певних ситуацій. Цифровий запис повністю усуває проблему із шумом магнітної стрічки.



Рисунок 1.5 - Типова цифрова студія звукозапису

Так само, як магнітна стрічка внесла багато інновацій у сцену запису, цифровий процес також створив нові можливості для творчості. Багато спецефектів, які раніше робилися за допомогою спеціалізованого обладнання, тепер можна виконувати за допомогою програмного забезпечення, що сприяє покращенню якості звуку. Кожного разу, коли сигнал проходить через інше обладнання, збільшується ймовірність появи шуму та спотворень. Програмні ефекти не так імовірно спричинять такі проблеми (хоча погано розроблене програмне забезпечення може негативно вплинути на звук).

Цифровий процес розширив деякі можливості мультитрекингу стрічки. Мультитрековість дозволяла записувати дуети двома виконавцями в різний час. Один артист заходив у студію і записував свою партію. Потім другий виконавець заходив у ту саму студію через кілька днів чи тижнів і записував свій вокальний трек. Завдяки магії змішування ці два голоси будуть змішані в єдиний дует. За

допомогою цифрових файлів можна виконати той самий процес, але замість того, щоб робити це в одній студії, основні файли можна надіслати електронною поштою в іншу студію в іншій частині країни, щоб записати другий вокальний трек. Насправді, багато треків на дуетних альбомах Френка Сінатри та Рея Чарльза були виконані саме таким чином. Деякі з виконавців ніколи не зустрічалися особисто з Сінатрою або Чарльзом.

Процес редагування було значно покращено в цифрову еру. Процедура вирізання та вставлення з епохи стрічки стала комп'ютеризованою версією вирізання та вставки. Будь-яку частину мелодії можна відтепер перенести на будь-яку доріжку у будь-яке місце на мультитреку. Так само, як переміщення слів і абзаців за допомогою текстового редактору. Це та багато іншого можна зробити за допомогою цифрового редагування музичних композицій.

Технологія оптичних дисків розроблялась ще у 70-х роках. До 1976 року компанія Philips мала робочий прототип аудіо-диску з використанням технології лазерних відеодисків. Одночасно і незалежно конкуруюча технологічна компанія Sony створила власний прототип оптичного диска лише для аудіо на основі відеодиска. Philips і Sony працювали разом, створюючи стандарт для компакт-касети в 1960-х роках і вони знову розпочинають співпрацю, результатом якої стане стандарт відомого нам компакт-диска. 30-сантиметровий оптичний диск міг зберегти до шістдесяти хвилин відео та звуку, слугуючи громіздкою версією DVD-дисків 1970-х років. Зчитувач дисків використовував лазери для просвічування прозорого диска, визначав розташування «ямок» і генерував бітовий потік, який потім конвертувався в аудіо- та відео-дані за допомогою спеціального програвача. Однак комерційний провал призвів до того, що компанія Philips вилучила свій відеодиск з ринку незабаром після випуску.

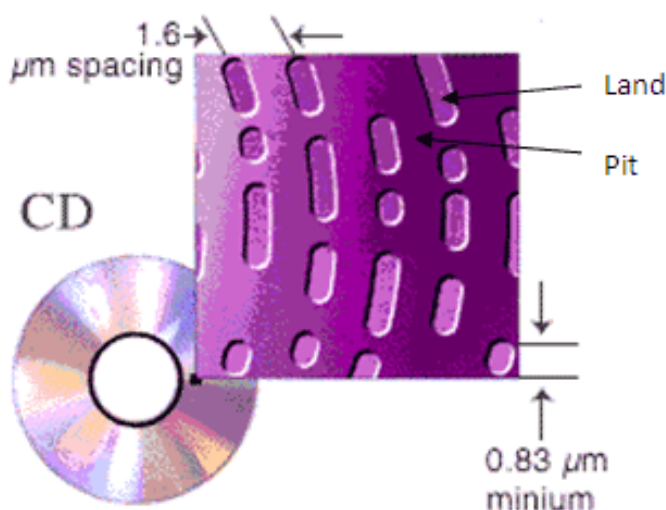


Рисунок 1.5 – Структура компакт-диску

Компакт-диск (CD) був помітним початком переходу від аналогового запису до цифрового. Philips і Sony спочатку не погодилися майже з усіх аспектів того, що має бути стандартним для компакт-дисків, за винятком максимальної години запису. Згодом Philips і Sony зупинилися на частоті дискретизації 44,1 кГц, завдяки специфікаціям цифрового відеозаписуючого обладнання, яке використовувалося для запису звуку в 1970-х роках. Діаметр і стандарти часу відтворення були визначені дружиною віце-президента Sony, яка хотіла, щоб будь-який можливий запис Дев'ятої симфонії Бетховена можна було відтворити на компакт-диску. В результаті виходить диск розміром 120 мм з максимумом 74-хвилинного простору для даних.

Sony випустила компакт-диски локально в 1982 році, після чого компанія Philips випустила на світовий ринок у 1983 році. Спочатку програвач компакт-дисків коштував 2000 доларів, але цільовою аудиторією був шанувальник класичної музики. Оскільки ціни знижувалися протягом наступних кількох років, продажі компакт-дисків обігнали вінілові пластинки наприкінці 1980-х, а невдовзі — касети в 1992 році. В результаті все більше і більше слухачів мали новий носій для насолоди музикою, який пропонував чіткий звук, довший час відтворення, більш висока точність протягом тривалого часу та менший розмір, ніж його аналогові попередники.



Рисунок 1.6 – Перший комерційний CD-програвач від Philips

У 1990-х роках домінували компакт-диски. Однак цифрові аудіофайли почали набувати популярності на початку 2000-х років через невеликий розмір файлів і можливість легкого перенесення. Починаючи з 1988 року, було створено формат, відомий як Motion Picture Experts Group (MPEG), щоб створити стандарти для кодування звуку. MPEG генерував різні режими стиснення звуку. Один з таких режимів, MPEG Audio Layer III, створював високу якість звуку при низьких бітрейтах. До 1992 року Layer III був удосконалений і готовий до використання, але не мав широкого застосування до 1995 року.

14 липня 1995 року стиснення рівня III увійшло в Інтернет з розширенням файлу «.mp3». Процес кодування, запланований у MPEG, буде дорогим і орієнтованим на великі компанії, які хочуть розповсюджувати свої аудіофайли через мережу, тоді як декодування буде дешевим, щоб споживачі могли легко отримати доступ до свого аудіоконтенту. Технічні компанії, як Music Match, використовували MP3, щоб дозволити читати компакт-диски на комп'ютері, а потім відтворювати музику — процес, який незабаром буде відомий як «ripping». До 1997 року кодування та декодування MP3 використовувалися для незаконного розповсюдження музики через Інтернет, проти бажання MPEG. MPEG продовжували працювати над форматами стиснення та кодування, що призвело до створення Advanced Audio Coding, яке виявилось наступником MP3, з більшою кількістю каналів, більш високою частотою вибірки та кращою обробкою

високочастотних аудіосигналів. Було створено більше типів цифрових аудіофайлів, що призвело до набору типів аудіо без втрат, без втрат і без стиснення для різних відтворень.

Цифрові аудіофайли перемістилися з робочого столу в реальний світ у 1998 році з винаходом першого MP3-плеєра, що дозволило користувачам брати свої файли .mp3 із собою в дорогу. Ранні програвачі, такі як MPMan F10 Saehan, могли зберігати 32 МБ даних, приблизно 8 пісень і проклали шлях для Apple iPod.

У міру вдосконалення технологій і збільшення обсягу пам'яті MP3-плеєри розвивалися на початку 2000-х років. Слухачі купували компакт-диски, зчитували їх на комп'ютері і зберігали улюблені пісні на своїх MP3-плеєрах. Одночасно популярні програми для обміну музикою, такі як Napster і Limewire, дозволяли користувачам платформу ділитися та завантажувати піратську музику.

На додаток до зростання популярності обміну музикою, такі програми, як iTunes і Myspace, почали продавати музику у вигляді цифрових завантажень у 2003 і 2004 роках. Популярність цього засобу швидко зросла, оскільки потреба у фізичних копіях музики зменшилася. Оскільки перехід до цифрового завантаження набрав обертів, сервіси потокової передачі музики, які забезпечували чисте цифрове аудіо, такі як Spotify і Pandora, почали набирати популярність серед слухачів музики. Незалежно від цих досягнень у сфері завантаження та хмарної музики, лише у 2014 році фізичні продажі музики обігнали цифрові доходи.

## **1.2 Цифро-аналоговий конвертер**

Перш ніж поглибитись в надри цифрової аудіоінженерії, що захопила світ з появою перших CD-дисків та MP3-плеєрів варто наголосити про сам процес перетворення цифрових файлів (будь то .mp3 або .wav) в аналоговий сигнал, який передається на вашу стерео-систему або навушники, або підсилювач, або будь-який інший пристрій, що має динамік.

Це цифро-аналоговий конвертер або Digital-Analog Converter (скорочено DAC або D2A). DAC приймає цифрові дані і перетворює їх в аналоговий звуковий

сигнал. Після цього він посиляє аналоговий сигнал на підсилювач. Коли ви чуєте цифрові записи, ви насправді слухаєте аналоговий сигнал, який був перетворений з цифрового DAC. Здебільшого DAC являє собою невеликий мікрочіп усередині будь-якого аудіо-пристрою, але є й цілі DAC-пристрої, що відрізняються підвищеною якістю відтворення цифрових файлів.

Як і у більшості речей, пов'язаних з аудіо, один DAC не обов'язково такий хороший, як інший DAC. Наприклад, ваш смартфон містить лише базовий DAC чіп. Він виробляє звук, який «достатньо хороший», щоб ви могли вести розмову по телефону, але він не є оптимальним для того, щоб отримати максимум від ваших улюблених музичних записів.

П'ятдесят років тому нам не потрібен був DAC для створення аналогового сигналу. Мікрофони в студії звукозапису вловлювали і зберігали звук у вигляді аналогових сигналів, як правило, у формі стрічки з барабана. Потім аналоговий сигнал вдавлювався в канавки для запису. Щоразу, коли ви хотіли послухати пісню, металева голка на вашому програвачі «відчувала» ці канавки і створювала електричний аналоговий сигнал. Він передавав сигнал через передпідсилювач і, зрештою, на динаміки.

Сьогодні аудіоінженери перетворюють аналогові сигнали в бітовий потік чисел (одиниць і нулів). Ця серія чисел є цифровим звуковим сигналом. Щоб його прослухати, потрібно перетворити його назад в аналоговий сигнал. Ось чому нам потрібні DAC. Без них ми б не змогли насолоджуватися портативністю та зручністю цифрового аудіо.

Ось як відбувається процес:

Під час запису виконавець прокладає трек. Мікрофони сприймають звуки голосів та інструментів як аналогові звукові сигнали.

Інженер зберігає аналогові сигнали як цифрові. Записувальне обладнання використовує аналого-цифрові перетворювачі для перетворення аналогових

сигналів в цифрові для зберігання. Зараз це зазвичай означає збереження їх у вигляді цифрових сигналів у вигляді цифрового аудіофайлу.

Під час відтворення DAC декодує збережені цифрові сигнали. При цьому DAC перетворює ці сигнали назад в аналоговий звук.

DAC посиляє перетворені аналогові сигнали на підсилювач. Підсилювач, у свою чергу, надсилає музику через навушники або стереодинаміки.

Як би магічно не звучав цей процес – і ми вважаємо, що це неймовірно технологічно, – є більш серйозне питання: чи достатньо хороший DAC на наших пристроях? Для тих із нас, хто цінує чудовий звук, це часто не так.

Придбавши зовнішній DAC, можна насолоджуватися якістю звуку, яка набагато перевершує DAC на вашому смартфоні або комп'ютері. Також зовнішній DAC можливо використовувати практично в будь-якому сет-апі для прослуховування. Тим не менш, вибраний вами DAC повинен «відповідати» типу пристрою, з якого ви відтворюєте свою музику:

Настільні та портативні комп'ютери: для більшості сучасних комп'ютерів слід використовувати USB DAC. Просто підключіть USB-кабель до доступного порту на вашому пристрої. Потім підключіть цей кабель до відповідного входу вашого DAC. Якщо ви використовуєте навушники, багато USB DAC мають вбудований підсилювач для навушників. Щоб підключити комп'ютер до аудіосистеми, вам потрібно підключити аудіокабель від DAC до системи. На деяких комп'ютерах вам, можливо, доведеться ввійти і призначити аудіовихід на DAC.

Смартфони та планшети: для невеликих пристроїв, якими ви користуєтеся в дорозі, все, що вам потрібно зробити, це придбати адаптер, який забезпечує підключення USB через цифровий вихід вашого пристрою. На смартфоні це, як правило, той самий вихід, який ви використовуєте для підключення зарядного пристрою. До USB-порту адаптера підключіть невеликий DAC, створений для портативних пристроїв і ви готові прослуховувати музику у високій якості.



Домашні стереосистеми: саме тут у вас, ймовірно, буде найбільший вибір підключень. Ви можете використовувати USB DAC для підключення ноутбука або комп'ютера до стереосистеми. Якщо ваш програвач потокової передачі музики має цифровий вихід, ви можете додати DAC, щоб покращити його звук у вашій системі.



Рисунок 1.7 – Компактний DAC для ноутбуку

### 1.3 Цифрові аудіо-формати

Цифрові аудіофайли, які є насправді колекцією бітів, яку можна зберігати й редагувати, у стандартній формі, можна зберігати на дисках комп'ютера, передавати на інші комп'ютери чи семплери, ділитися в Інтернеті для завантаження, додавання до відеофайлів або відтворення в режимі реального часу. Вони відрізняються від доріжок аудіо компакт-дисків або касет, які здебільшого містять лише вихідні дані зразка або елементи, непотрібні для комп'ютера, такі як дані для виправлення помилок і субкод, які допомагають направляти CD-лазер. Ось чому доріжка компакт-диска має бути перетворена у читаний формат аудіофайлу для використання комп'ютерною програмою. Стандартний 16-розрядний

стереофайл розміром 44,1 КБ споживає приблизно 10 мегабайт дискового простору за хвилину звуку.

Аудіофайли бувають різних типів, які можуть вплинути на їх бітову глибину, багатоканальну організацію, схему стиснення, частоту дискретизації, організацію байтів від високого до низького або навпаки (називний порядковість) та кількість збереженої інформації без вибірки. в області, яка називається заголовком, в одиницях, які називаються шматками. Деякі формати аудіофайлів також можуть містити дані MIDI. Деякі можуть містити дані із сеансу редагування, який їх створив, наприклад, межі та назви регіонів або точки циклу та кореневі ключі для семплерів. Багато аудіопрограм здатні відкривати та конвертувати декілька форматів файлів у певних межах. А деякі звукові формати працюють тільки на певній комп'ютерній платформі, наприклад, PC або Mac. Деякі додаткові терміни, які ви можете побачити, дивлячись на формат звукового файлу, пов'язані з бітовою глибиною, часто відноситься до того, як комп'ютери зберігають числа різного розміру. Звичайні розміри зразків аудіофайлів часто називають 8-розрядними символами, 16-розрядними короткими цілими числами, 24-розрядними знаковими та беззнаковими цілими числами, 32-розрядними довгими чи 32-бітними числами з плаваючою комою (float).

Роздільна здатність звуку визначається частотою дискретизації та бітовою глибиною цифрового аудіозапису. Частота дискретизації – це кількість вибірок аудіо за секунду, яка контролює частотну характеристику запису. Під час запису цифрового звуку ваш інтерфейс робить тисячі знімків кожної форми хвилі та відтворює їх, щоб звучати як єдиний запис. Схоже на те, як рухається відео, хоча це просто серія нерухомих зображень, як фліпбук. Тобто частота дискретизації представляє собою окремі малюнки у вашому аудіо-фліпбуку. Кожен зразок являє собою один маленький кадр великої картини. Використання більшої частоти дискретизації призведе до більшої кількості вибірок.

Теорема вибірки Найквіста-Шеннона стверджує, що максимальна частотна характеристика цифрового запису становить рівно половину використовуваної

частоти дискретизації. Найнижча частота дискретизації, яку ви повинні використовувати для професійно випущеної музики, становить 44 100 кГц/с, часто скорочену до простого 44,1 Кб. При цій частоті дискретизації ваш трек матиме максимальну частотну характеристику 22050 Гц, що виходить за межі діапазону людського слуху та гарантує, що ви чуєте кожну деталь запису.

Багато професійних інженерів не намагаються використовувати частоту дискретизації 48 кГц і вище, тому що немає помітного поліпшення якості, а файли займають більше місця на жорсткому диску.

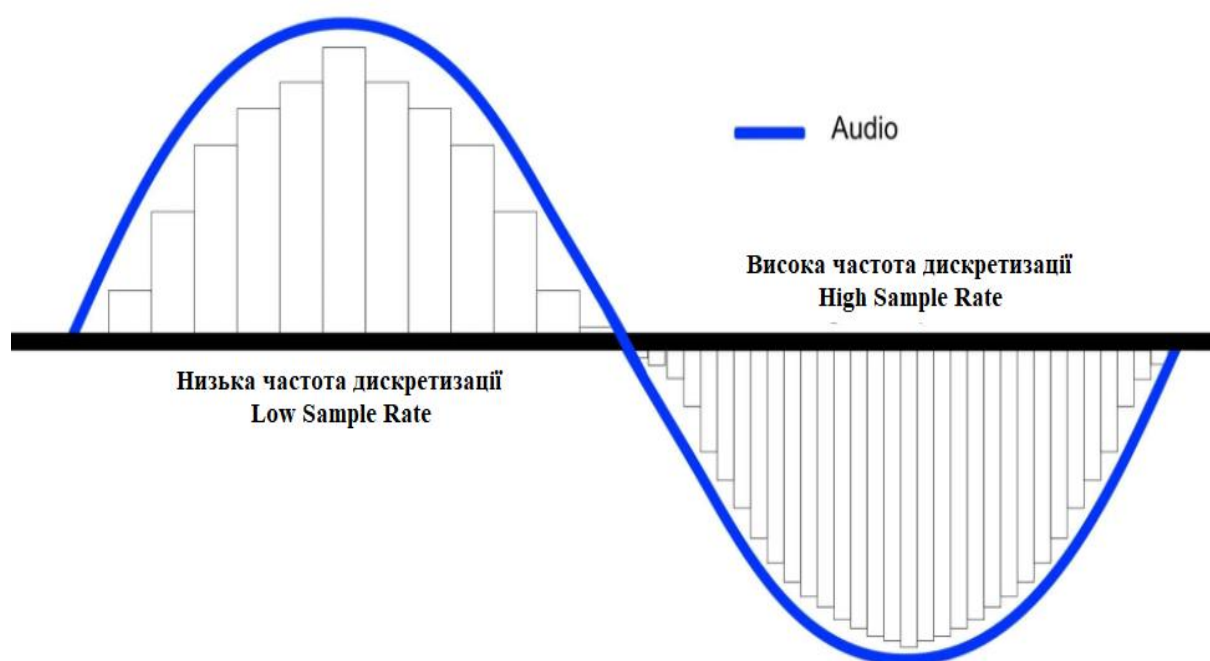


Рисунок 1.8 – Графік частот дискретизації

Розрядність або бітова глибина — це вимірювання того, скільки даних можна зберегти в кожному зразку, також вона контролює динамічний діапазон запису. Найнижча бітова глибина, яку ви повинні використовувати для професійних записів, — 16 біт, якщо ви не збираєтеся досягти ретро 8-бітового звуку Chiptune.

Багато DAW та цифрових аудіозаписувачів пропонують налаштування до 24 або навіть 32-бітної розрядності, що забезпечує нижчий рівень шуму та більший динамічний діапазон.

Якщо частота дискретизації — це кількість нерухомих кадрів або зображень у вашому фліпбуці, то подумайте про бітову глибину як розмір блокнота, в якому ви малюєте. Вона визначає, скільки місця вам потрібно для запису.

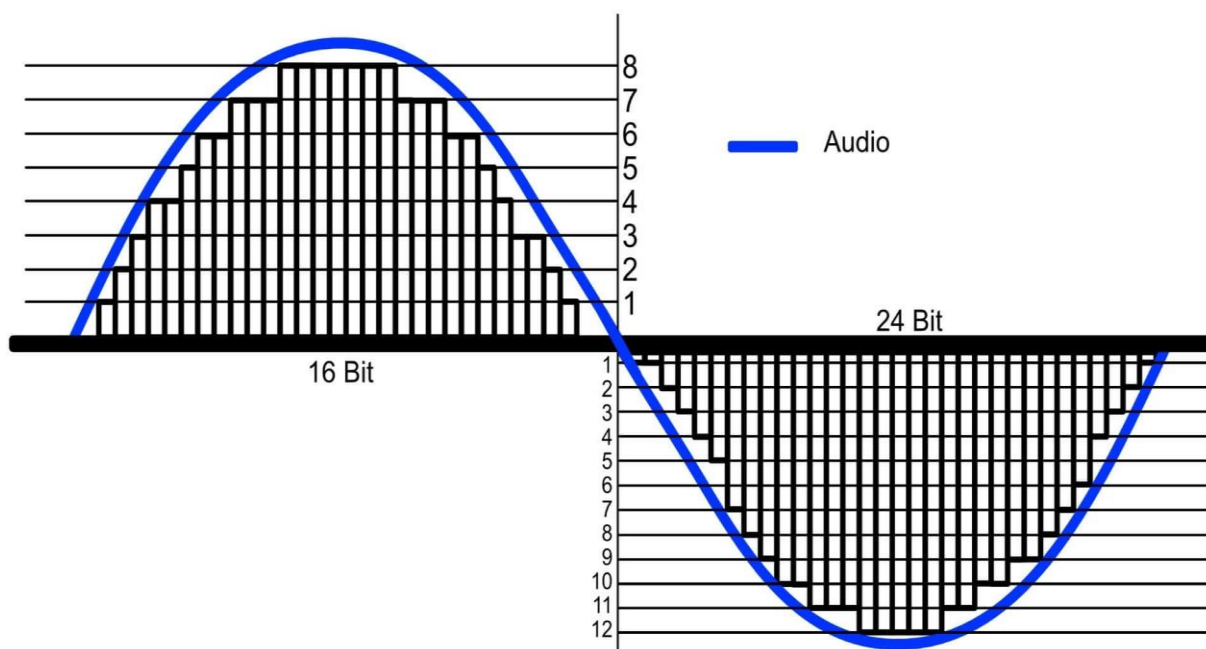
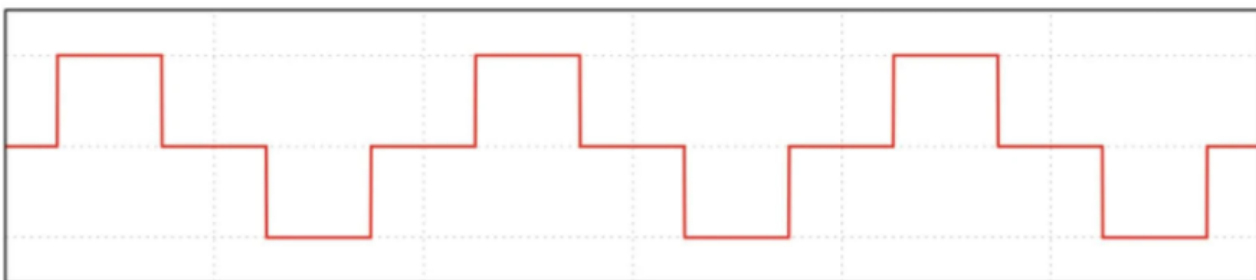
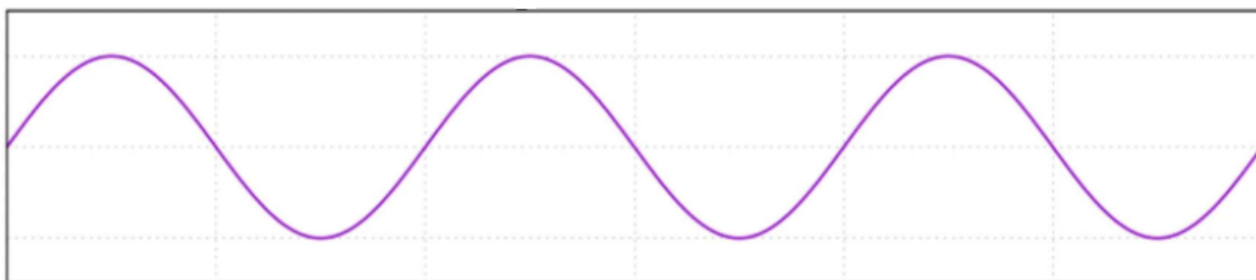


Рисунок 1.9 – Графік розрядності бітів

Кожен цифровий аудіофайл займає певну кількість місця на жорсткому диску. Щоб зменшити розмір цифрових аудіофайлів, деякі формати використовують стиснення даних. Однак це спричиняє деяку втрату якості, оскільки ви не отримуєте повного «зображення» вихідної форми сигналу.

На рисунку 1.10 можемо побачити що собою графічно представляють формати .wav і .mp3. Коли .wav передає стиснені біти музики в повному обсязі, .mp3 файл обрізає ці біти і надає нам "урізану" версії композиції, якість якої буде гірша за .wav, проте незброєним вухом її буде важко відрізнити.

## Нестиснений WAV файл (uncompressed)



## Стиснений із втратами MP3 файл (lossy)

Рисунок 1.10 – Різниця між нестисненим та стисненим файлом

Існує три типи цифрових аудіофайлів:

- нестиснений (uncompressed);
- стиснений із втратами (lossy);
- без втрат (lossless);

Нестиснені файли не використовують стиснення даних. Вони пропонують найвищу якість звуку, але також є найбільшими файлами. Використовуючи імпульсно-кодову модуляцію, нестиснені файли пропонують 100% точне відтворення оригінального запису. Нестиснені файли зазвичай використовуються для фізичних випусків, таких як компакт-диски та DVD-диски.

Файли з втратами використовують стиснення даних, щоб значно зменшити розмір аудіофайлів. Кодування з втратами за допомогою алгоритмів для виявлення та видалення нечутних частин запису, щоб заощадити місце. Після стиснення музичних даних таким чином, можна отримати до 1/10 розміру вихідного файлу. Однак це може негативно вплинути на якість вашої музики, і немає способу відновити втрачені дані після стиснення файлу.

Файли без втрат пропонують найкраще з обох сторін. Вони використовують особливий тип стиснення даних, який називається кодуванням без втрат, що зменшує розмір файлу, не впливаючи на якість звуку. Подібно до стиснення архіву ZIP, дані стискаються під час зберігання та передачі файлів, але під час відтворення файл декодується та відтворюється в оригінальній якості.

Перелік основних аудіо-форматів та форм стиснення, які вони використовують:

- WAV — без стиснення;
- AIFF—нестиснутий;
- ALAC—без втрат;
- FLAC—без втрат;
- MP3—з втратами;
- AAC—з втратами;
- WMA—з втратами;
- OOG?;

WAV означає Waveform Audio. Цей нестиснений тип файлу спочатку був розроблений IBM і Microsoft у 1991 році і досі є типом файлу за замовчуванням на машинах Windows. Він підтримує частоту дискретизації до 192 кГц, розрядність до 32 біт. Файли WAV дуже поширені і сумісні з усіма апаратними та програмними програвачами, які підтримують цифрові файли. Однак файли WAV, як правило, дуже великі, і їх завантаження або буферизація під час потокової передачі може зайняти багато часу.

AIFF означає формат файлу обміну аудіо. Розроблений Apple у 1988 році, цей нестиснутий аудіофайл, по суті, є версією файлу WAV від Apple. Він також підтримує частоту дискретизації до 192 кГц, бітову глибину до 32 біт. Файли AIFF підтримують теги метаданих, яких файли WAV не мають. Хоча AIFF спочатку був розроблений виключно для машин Apple, тепер його можна використовувати практично на будь-якому пристрої.

FLAC розшифровується як Free Lossless Audio Codec і є цифровим аудіоформатом з відкритим вихідним кодом без втрат. Файли FLAC можна передавати та декодувати з блискавичною швидкістю, що робить їх популярним вибором для потокової передачі HD. Крім того, FLAC пропонує дев'ять різних рівнів стиснення даних на основі швидкості кодування (швидкості декодування відносно однакові). Незважаючи на те, що він є одним з найпопулярніших форматів без втрат, в Apple Music або iTunes досі немає підтримки FLAC.

ALAC розшифровується як Apple Lossless Audio Codec і є поглядом Apple на аудіокодек без втрат. На відміну від більшості інших типів файлів, файли ALAC використовують розширення, відмінне від їх імені. Можливо, ви більше знайомі з файлами ALAC з розширеннями ".m4a". При стисненні файли ALAC мають приблизно 60% розміру вихідного файлу. Як і FLAC, вони не стиснуті під час відтворення і звучать ідентично оригінальному запису. Хоча ALAC підтримується Mac і ПК, апаратна підтримка обмежена переважно пристроями iOS. Файли ALAC можна завантажувати на сайти цифрової потокової передачі, але вони конвертуються в MP3 для відтворення.

Мабуть, найпопулярніший цифровий аудіоформат із втратами, MPEG-1 Audio Layer 3, або MP3, був представлений у 1993 році. У той час пропускна здатність була обмежена, а компактний розмір MP3 дозволяв легко ділитися та розповсюджувати музичні дані в Інтернеті. Однак цей формат шкодить якості звуку. Під час кодування MP3 бітрейт або швидкість перетворення визначає розмір і якість аудіофайлу. Найвищий бітрейт становить 320 кбіт/с, що займає найдовше кодування, але забезпечує найякісніші результати. Найнижчий рекомендований бітрейт – 128 кбіт/с. Навіть ті, у кого немає навичок слуху, зазвичай можуть сказати, що з MP3-файлами з бітрейтом 128 Кбіт/с звучить дуже замкнено та неякісно. Тому якщо ви все-таки вирішите використовувати цей формат, краще зупинитися на 320 Кбіт/с.

AAC означає Advanced Audio Coding і спочатку був розроблений командою технологічних компаній у 1997 році, включаючи Bell, Fraunhofer, Dolby, Sony,

Nokia, LG Electronics, NEC, NTT Docomo та Panasonic. AAC забезпечує кращу якість звуку, ніж MP3-файли, хоча вони все ще звучать не так добре, як файли без втрат. Однак він пропонує частоту дискретизації до 96 кГц, на відміну від обмеження в MP3 в 48 кГц. Завдяки цьому файли AAC можуть захоплювати більш широкий діапазон частот. Завдяки компактному розміру та покращеній якості звуку, AAC швидко став популярним типом файлів для мобільних сервісів та потокового програмного забезпечення, такого як iTunes, YouTube та Tidal.

WMA означає Windows Media Audio. Як ви вже здогадалися з назви, це версія MP3 від Microsoft. Подібно до AAC, файли WMA забезпечують покращену якість звуку порівняно з MP3 при тому ж бітрейті. Однак тип файлу WMA ніколи не був настільки поширений, як AAC і MP3, і все ще не підтримується пристроями Android.

OGG Vorbis — це ще один новий тип файлу, розроблений для покращення формату MP3. OOG є форматом з відкритим вихідним кодом, що означає, що він безкоштовний для комерційного та некомерційного використання, що робить його популярним вибором для тих, хто не хоче платити за програмне забезпечення високоякісного кодера. Spotify фактично використовує файли OOG як аудіоформат за замовчуванням, оскільки це найменший формат аудіофайлу. Однак більшість апаратних пристроїв все ще несумісні з OOG, що ускладнює його використання деякими пристроями.

Звичайно, це не єдині доступні типи файлів — лише найпопулярніші та широко використовувані формати. Який же формат найкращий для використання? Це буде залежати від цілей аудіо, з яким ви працюєте. Для запису, мікшування та мастерингу завжди слід використовувати нестиснені аудіофайли, такі як WAV та AIFF. Якщо ви хочете носити з собою всю музичну бібліотеку, куди б ви не були, або миттєво транслювати улюблені пісні без буферизації, використовуйте формат із втратами, як-от AAC або MP3. Якщо для вас важлива висока якість аудіо-файлу та її компактний розмір, використовуйте формат файлу без втрат, наприклад FLAC або ALAC.



## 1.4 Компоненти цифрової студії

Створення музичної студії в офіційному приміщенні або вдома не потребує великих зусиль, проте може обійтись у велику копійку. Незалежно від того, чи ви записуєте для інших виконавців чи створюєте особисту студію для власної музики чи розмовних проєктів, створення творчого простору для створення музики та інших видів аудіо потребує більш-менш професійного підходу з точки зору обладнання, для поліпшення якості фінального матеріалу.

Усі студії використовують подібні комплекти обладнання, хоча деякі без наворотів, які складають деякі більші студії. Незважаючи на те, що для початківця аудіопродюсера може бути трохи заплутано зібрати студію, є кілька основних елементів обладнання, які забезпечать плавний старт, навіть при обмеженому бюджеті. Ось вичерпний список основного обладнання, яке можна знайти в студії звукозапису. Окрім комп'ютера, що виконує основну роль при записуванні та зведенні музичних треків, також необхідно звернути увагу на вибір інших компонентів:

Цифрова робоча станція (DAW). Хороша DAW є суттєва для будь-якого аудіоінженера. Вибираючи DAW, важливо враховувати, які ресурси вона використовуватиме на комп'ютері. Виробники аудіо, які можуть мати менш надійну машину, повинні ретельно обміркувати, яку DAW вони використовуватимуть, щоб уникнути перевантаження ресурсів свого комп'ютера та прискорити день, коли його потрібно замінити.

Мікрофони та підставки. Великі професійні студії мають величезний асортимент мікрофонів для різних інструментів і типів голосу. Продюсерам аудіо, які тільки починають роботу, не обов'язково потрібно так багато, особливо тим, хто має бюджет, але вони повинні знати, які мікрофони вони вибирають.

Мікшерний пульти. Якщо DAW має вбудовану мікшерну плату, фізична мікшерна плата може не бути абсолютно необхідною. Однак наявність базової мікшерної плати під рукою є важливою частиною базового обладнання для

аудіоінженера. Змішувачі мають різноманітні канали, від чотирьох до двадцяти. Вони використовуються для встановлення рівнів для запису, керування гучністю та вихідним сигналом, а також для додавання ефектів до записів. Виробники аудіо, можливо, захочуть розглянути деякі стандартні торгові плати, такі як Yamaha, Soundcraft, SSL, API, Neve.

Акустичний екран. Підготовлені та вироблені на замовлення ізольовані кімнати для запису можуть коштувати велику кількість грошей. Для невеликих студій інвестування або створення базових звукових акустичних екранів є життєво важливими для отримання якісних записів, оскільки вони не дають непотрібним звукам та шумами попасти на музичну доріжку.

Наушники для мікшування. Зрештою, аудіо продюсер повинен почути все, що записував з великою точністю, що робить вибір навушників для мікшування абсолютно незамінними. Не всі навушники для мікшування створені однаково, але гідну пару не повинно бути важко знайти. Аудіоінженерам варто розглянути стандартні мікшерні навушники від таких брендів, як Yamaha, Korg і Beyerdynamic. Менш дорогі бренди, такі як Audio Technica, також пропонують надійні варіанти навушників для мікшування та моніторингу в студії.

Студійні колонки. Вони необхідні для перевірки якості звуку після запису дубля. Як аудіо продюсери, так і виконавці повинні слухати запис через навушники та студійні колонки, щоб забезпечити якісний звук для обох і якомога краще усунути будь-які проблеми, які вони знайдуть. Студійні колонки бувають різних розмірів і спектру якості. Серед стандартних можна виділити Yamaha NS-10, KRK Rokit, M-Audio Studiophile і Behringer Truth. Початківці повинні знати, що студійні колонки будуть коштувати чимало грошей, можливо, стільки ж або більше, ніж комп'ютер, на якому вони записують музику. Покупка б/у варіантів може зменшити фінансовий тягар, хоча ці варіанти все одно будуть дещо дорогими. Для тих, у кого є невеликий бюджет, колонки та комп'ютер мають стояти першими у черзі, коли йде мова про пріоритетність купівлі звукозаписуючого обладнання.

Інструменти. У більшості студій під рукою буде принаймні невеликий вибір інструментів. Діапазон дуже залежить від сфери фокусування студії, хоча навіть у деяких невеликих студіях буде вражаючий вибір. У багатьох студіях буде вибір стандартних рок-інструментів, включаючи електро- та акустичні гітари, клавішні та синтезатори, бас-гітари та барабанну установку. Вони також матимуть якісні власні підсилювачі для гітар та басів. Деякі студії також можуть мати класичні інструменти, такі як струнні, дерев'яні духові та ударні. Студії, які хочуть створювати інструментальні колекції для виконавців, витратять багато грошей і часу на їх придбання. Багато студій обирають власні інструменти високої якості, тому, якщо виконавець не володіє першокласним інструментом, він все одно може прийти і записати на відмінному інструменті, який забезпечить кращий звук.

### 1.5 Процес звукозапису через комп'ютерні програми

Існує багато програм різного рівня звукозапису для різних потреб. Деякі з них прості у використанні навіть для людини, яка не розуміє половини її функцій. У той час як більшість мають складний інтерфейс і щоб повністю розібратися в ньому, пишуться цілі книги. На прикладі FL Studio, яка користується величезною популярністю, як і серед новичків, таких і матових аудіоінженерів, можна розглянути сам процес звукозапису.



Рис. 1.7 – Інтерфейс FL Studio

Дуже потужна цифрова аудіо робоча станція, FL Studio дозволяє записувати живу музику за допомогою акустичних інструментів і створювати електронну музику. Це робить створення музики простим, завдяки кнопкам покрокового секвенсора та легкої функції запису. Користувачі можуть записувати з MIDI-клавіатури і FL Studio з легкістю перефразує всі ноти. Після завершення шаблонів ви зможете зберегти їх у списку відтворення, де до них можна легко отримати доступ, щоб використовувати їх у разі потреби в треку.

FL Studio постачається з набіром інструментів, які ви можете налаштовувати за допомогою функцій швидкої автоматизації в програмному забезпеченні. Він використовує інтерфейс, сумісний з мультисенсорними моніторами, легко читається і масштабується. Якщо ви використовуєте FL Studio на пристрої з сенсорним дисплеєм, ви можете працювати з ним так, ніби це була фізична мікшерна дошка, яка імітує досвід професійної студії звукозапису.

У панелі View можна налаштувати вікна із інтерфейсами, які будуть відображатись у програмі під час роботи. Із стандартних вікон можна виділити:

Цифровий мікшер (DAW). Коли ви запускаєте FL Studio вперше, завантажиться демонстраційна пісня. Це хороша відправна точка, яка дозволяє вам побачити, що ви можете робити з DAW. Image-Line створила багато різноманітних шаблонів, які допоможуть вам почати роботу. Наприклад, ви можете завантажити шаблон EDM-House, який включає всі необхідні інструменти, маршрути мікшерів та розташування інструментів. Якщо хочете, ви також можете налаштувати його з нуля.

Channel Rack містить інструменти, які створюють музичні біти, і внутрішні генератори, які керують автоматизацією. Кожен шаблон має доступ до всіх інструментів у стійці. Іншими словами, всі біти грають з одного набору інструментів. Патерни не обмежуються одним інструментом, як у більшості інших секвенсорів.

Плейлист. На ньому розташовуються музичні доріжки, які ви записуєте та редагуєте.

Також у меню можна поставити панель швидкого доступу до плагінів та проектів, тачпад, налаштовувач темпу та ін.

Окрім відомого усім FL Studio, існує також Cubase від Steinberg. За своїм зовнішнім виглядом вони не надто відрізняються, про те є великі відмінності в технічному плані і вибір програми для ваших потреб буде скоріше базуватись на обладнанні, яке ви використовуєте.

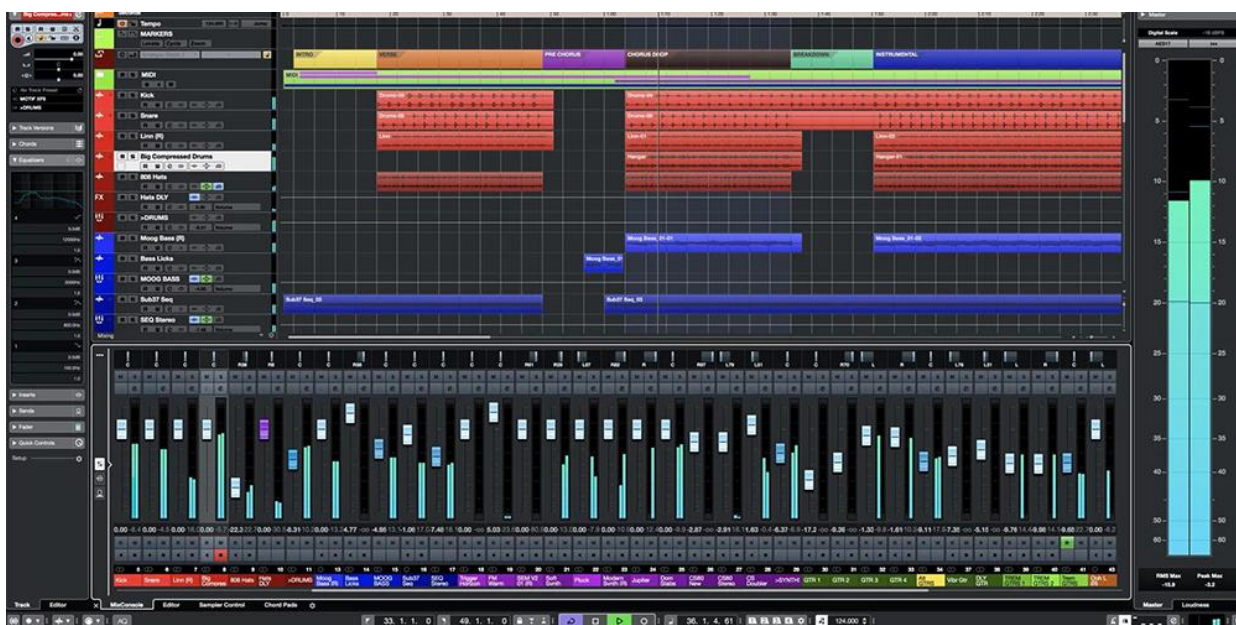


Рис 1.8 – Зовнішній вигляд Cubase

Cubase надає інструменти, необхідні для редагування, змішування та запису. Аудіо студійної якості обіцяють протягом усього виробництва. Програмне забезпечення регулярно оновлюється, щоб покращити роботу користувача, з новими функціями або оновленими оригінальними функціями для постійної модернізації. При купівлі Cubase, користувач також стає частиною товариства Steinberg, яке надає багато інших аудіо-програм, які буде можливо придбати з невеликою знижкою.

### Основні відмінності між Cubase і FL Studio:

Cubase має ємність від 24 до необмеженої кількості MIDI – залежно від того, яка у вас версія, в той час як FL Studio пропонує до 500 доріжок, що можуть бути використані в один час, без залежності від версії програми.

Cubase містить 14 гігабайт звуків, тоді як звуки, включені в FL Studio, більш обмежені.

Cubase вимагає оплати за оновлення, тоді як оновлення FL Studio безкоштовні при одноразовій покупці програмного забезпечення

Cubase найкраще підходить для створення пісень на основі живих інструментів, тоді як FL Studio більше для MIDI-композицій та створення електронної музики.

Cubase доступний 9 різними мовами, тоді як FL Studio доступний лише 3 різними мовами – англійською, голландською та китайською

Більш простою для невимогливого користувача буде програма Audacity, яка наповнена великою кількістю функцій та можливостей, проте більшість із них лімітовані до найпростіших версій. Вона має ненавантажений інтерфейс і основні завдання при побаченні музичних доріжок виконуються в кілька кліків, без особливих зусиль. Вона також є повністю безкоштовним, кросс-платформним open-сурс проектом, яка доступна для більшості наявних ОС. Недоліком є, що праця в такій програмі зводиться максимально лише до запису або обробки готової композиції, оскільки програма не підтримує більшість популярних плагінів та електронної емуляції інструментів та ефектів, записувати навіть живі інструменти через неї – не буде достатньо цікавим процесом. Повноцінної DAW цю програму назвати не можна, проте вона ідеально підійде для швидкого та неприскіпливого редагування музичних доріжок.

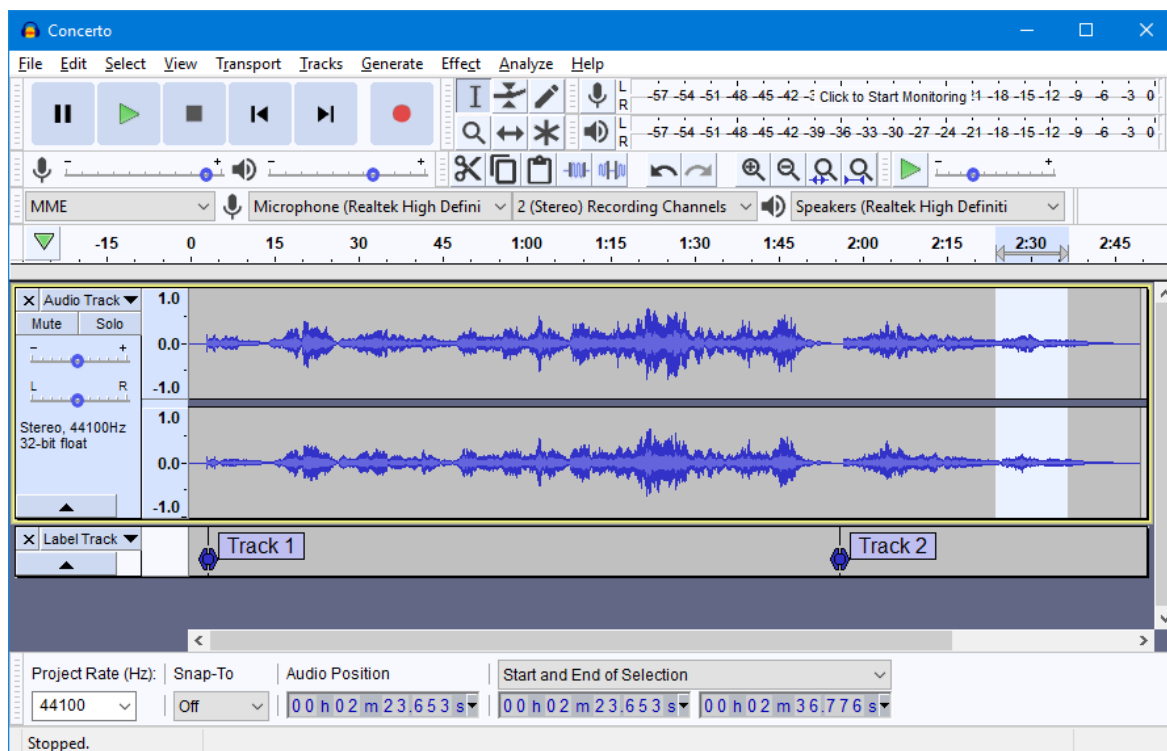


Рисунок 1.9 – Зовнішній вигляд Audacity

Таблиця 1.1 – Технічні відмінності між Cubase, FL Studio, Audacity

Мінімальні системні вимоги	Cubase	FL Studio	Audacity
Працює із ОС	Windows, MacOS	Windows, MacOS	Windows, MacOS, GNU/Linux
RAM	1024 MB	4 GB	100 MB
Рекомендований дисплей	1280x800	3840x2160	1024x768
Частота дискретизації	44.1 Hz	192 Hz	44.1 Hz

Після того, як трек був записаний розпочинається процес зведення та мастерингу доріжок, Зведення - процес створення з окремих записаних треків кінцевого запису, наступний після аранжування етап створення звукозапису.

Зведення - не суто технічний процес з'єднання різних треків в єдине ціле, це скоріше творча діяльність, від якої залежать особливості звучання результату.

Процес зведення ставить перед нами два завдання — по-перше, потрібно поєднати в єдине ціле безліч аудіо-доріжок, а по-друге постаратися позбавитися «комп'ютерного» звучання, щоб музика грала максимально динамічно і красиво.

Мастеринг – процес створення еталонного носія, копією якого є споживчий носій аудіозапису (компакт-диск, вінілова платівка, магнітофонна касета). Згодом мастерингом стали називати процес підготовки фінального міксу.



## РОЗДІЛ 2 СТВОРЕННЯ АУДІОПЛАГІНУ

### 2.1 VST-плагіни

Плагіни Virtual Studio Technology (VST) – це те, що використовують, як і новачки та професійні інженери та виконавці звукозапису для покращення своїх аудіопроектів. Плагін — це тип програмного забезпечення, яке працює всередині іншого програмного забезпечення. Без спеціального програмного забезпечення VST-плагін самостійно, звісно, не запуститься, тому для запуску таких плагінів використовуються вищеописані Cubase та FL Studio. Сам стандарт VST був вперше запроваджений компанією Steinberg для створення звукових ефектів та синтезаторів.

Хоча плагіни VST призначені для використання з додатками DAW, різні типи плагінів VST мають різні цілі. Плагіни інструментів VST дозволяють вставляти різноманітні віртуальні інструменти в аудіозапис, не маючи доступу до фізичних версій цих інструментів, а плагіни ефектів VST дозволяють змінювати звуки новими та захоплюючими способами. У сукупності різні типи плагінів VST можуть допомогти вам створити більш привабливий кінцевий продукт.

Оригінальний метод запису та мікшування вимагав фізичних інструментів та фізичного обладнання для досягнення таких ефектів, як стиснення та реверберація. Обладнання, необхідне для цього виду робіт, непомірно дороге і займає багато місця. Технологія віртуальної студії була розроблена для заміни або доповнення фізичного обладнання. Замість того, щоб інвестувати в інструменти, синтезатори, обладнання для ефектів та інше дороге обладнання, ви можете використовувати плагіни VST, які виконують ті ж функції. Насправді, ви можете створити цілу пісню з нуля, використовуючи плагіни VST і DAW на ваш вибір, ніколи не торкаючись справжнього музичного інструменту.

Плагіни VST зазвичай поділяються на три основні категорії:

Інструменти VST: ці плагіни генерують аудіо, яке звучить так, ніби воно було створене синтезатором або традиційним інструментом, таким як фортепіано чи

гітара. Багато плагінів VSTi створені так, щоб звучати та виглядати, як популярні апаратні синтезатори, багато з яких надзвичайно дорогі або більше не доступні.

Ефекти VST: ці плагіни беруть аудіо та змінюють його різними способами. Вони не можуть створювати новий звук, але можуть додавати реверберацію та інші ефекти. Ця категорія також включає плагіни, які забезпечують різні типи візуального зворотного зв'язку.

Ефекти VST MIDI: ці плагіни працюють з даними цифрового інтерфейсу музичних інструментів (MIDI) самостійно або спільно з іншими плагінами VST. Якщо ви музикант і зацікавлені в підключенні такого інструменту, як клавіатура, до вашої DAW через MIDI-з'єднання, це стане в нагоді.



Рисунок 2.1 – Типовий інструментальний VST-плагін

Інструментальний VST плагіни, або плагіни VSTi, емулюють різні інструменти на цифрових аудіо робочих станціях. Ви можете емулювати майже будь-який інструмент з VSTi за меншу вартість, ніж вартість фактичного інструменту. Перевага плагінів VST полягає в тому, що вони дозволяють

створювати музику з нуля, яка звучить так, ніби вона була зіграна на інструменті, як-от фортепіано чи саксофон, або створювали за допомогою синтезатора.

Такі плагіни можуть з легкістю емулювати навіть такі синтезатори, як Moog Voyager, Yamaha CS-80, Hammond В3 та ін. Головною особливістю таких синтезаторів було їх культове звучання та наймовірно висока ціна, яку не може собі дозволити більшість людей. Існують, як і безкоштовні, так і преміум версії VST-плагінів таких синтезаторів, що суттєво відрізняються власним інтерфейсом та кількістю функцій, що вони пропонують.

Ще одна важлива перевага використання VSTi плагінів в тому – це простір. Адже синтезатори з високою кількістю емульованих інструментів та ефектів потребують неабияку кількість місця в приміщенні. Якщо це спеціалізована студія, то таке місце може знайтись, про те організовуючи студію вдома варто подумати про наявність зайвих квадратних метрів для такого обладнання.

Якщо плагіни VSTi створюють аудіо з нуля, плагіни ефектів VST беруть цей звук і змінюють його різними способами. Ці плагіни часто моделюються за моделлю реальних фізичних ефектів так само, як багато плагінів VSTi створені так, щоб звучати як справжні педалі ефектів та синтезатори.



Рисунок 2.2 – Типовий VST-плагін ефектів

Більшість цифрових аудіо-станцій мають вбудовані ефекти та інструменти тонкого налаштування вашого міксу, проте плагіни ефектів можуть вивести це на новий рівень. Будь-якого ефекту, який ви бажаєте почути на аудіо-доріжці, можна досягти за допомогою такого плагіну, будь-то реверб, панорамування або імітація синт-вейву.



Рисунок 2.3 – Типовий VST MIDI-плагін

Останніми представниками VST плагінів є VST MIDI-плагіни. MIDI (Musical Instrument Digital Interface) - це протокол, що містить набір стандартних команд для взаємодії між звуковими пристроями. Він був розроблений у 1983 році та дає можливість електронним інструментам взаємодіяти між собою, комп'ютером чи іншим MIDI-обладнанням. MIDI не створює або передає звук. Цей протокол передає повідомлення (такі як нота-вкл/нота викл, висота (pitch), контрольні сигнали, тощо), а вже музичні інструменти або програма на комп'ютері приймає ці повідомлення та генерує звук. VST-плагіни для MIDI оброблюють MIDI-повідомлення власноруч або передають їх на інші плагіни для. Вони можуть виконувати відносно базові завдання, такі як транспонування або арпеджування.

MIDI-композиції без будь-якої обробки звучать штучно для досвідченого вуха і змінювати ці звуки для більш природнього та живого звучання можливо за підключення інших VST-плагінів ефектів.

## 2.2 Фреймворк JUCE

JUCE (Jules' Utility Class Extensions) це комплексна бібліотека класів C++ для розробки кроссплатформенного програмного забезпечення. Він містить практично все, що вам може знадобитися для створення більшості додатків, особливо добре підходить для побудови складних GUI, обробки графіки та аудіо.

JUCE підтримує наступні платформи та можливості:

- Mac OS X програми та VST/AudioUnit/RTAS/NPAPI плагіни, що компілюються за допомогою Xcode;
- Windows Програми та VST/AudioUnit/RTAS/NPAPI/ActiveX плагіни, що збираються за допомогою MS Visual Studio. Результати повністю сумісні з Windows XP, Vista та Win7/8;
- Linux програми та плагіни, що збираються для будь-якого ядра версії 2.6 та старше;
- додатки на iOS для iPhone/iPad, що збираються на Xcode;
- Android-додатки, які збираються за допомогою Ant або Eclipse за допомогою Android NDK.

JUCE розроблявся силами однієї людини та з набору класів для особистого використання переріс у повноцінний фреймворк. OpenSource ліцензія з'явилася в 2003 році, а з 2005 фреймворк отримав платну її версію для закритого ПЗ.

На більшості модулів фреймворку стоїть ліцензія GPLv2, v3 та AGPLv3, а на модулі Core - ISC. Існує також комерційна версія програми, яка позбавить розробника від необхідності публікації власного коду:

- ліцензія на один продукт - 399 \$;
- ліцензія на необмежену кількість продуктів — 699 \$;

- апгрейд нових версій – 349 \$;
- апгрейд старих версій – 349 \$

Сам JUCE досить легкий, трохи більше 28 Мб. Папка docs містить докладну документацію щодо встановлення та підключення фреймворку у різних ОС та середовищах розробки.

Имя	Дата изменения	Тип	Размер
.github	06.10.2021 18:44	Папка с файлами	
docs	06.10.2021 18:44	Папка с файлами	
examples	06.10.2021 18:44	Папка с файлами	
extras	06.10.2021 18:44	Папка с файлами	
modules	06.10.2021 18:44	Папка с файлами	
.gitignore	06.10.2021 18:44	Файл "GITIGNORE"	2 КБ
.gitlab-ci.yml	06.10.2021 18:44	Файл "YML"	1 КБ
BREAKING-CHANGES	06.10.2021 18:44	Текстовый докум...	62 КБ
ChangeList	06.10.2021 18:44	Текстовый докум...	26 КБ
CMakeLists	06.10.2021 18:44	Текстовый докум...	7 КБ
LICENSE.md	06.10.2021 18:44	Файл "MD"	2 КБ
README.md	06.10.2021 18:44	Файл "MD"	5 КБ

Рисунок 2.4 – Компоненти JUCE

Стосовно установки варто помітити що JUCE - це не набір бібліотек у звичному сенсі або набором вихідних джерел з яких ви повинні зібрати бібліотеки для статичної або динамічної лінківки, а набір заголовних файлів і файлів вихідного коду, що містять усі класи, що надаються фреймворком. Цей підхід не тільки спрощує міграцію з однієї платформи на іншу, позбавляючи вас необхідності включати до складу дистрибутивів бібліотеки фреймворку або вимагати їх наявності на комп'ютері користувача, але й вирішує проблему розрядності бібліотек від яких залежить фреймворк. Так як стандартні бібліотеки або стандартне оточення сучасних ОС, яке використовується в проектах JUCE (всі вони описуються в документації, що поставляється в комплекті з фреймворком), як правило, зібрані з підтримкою як 32 так і 64 bit. Так що вам варто стежити за бітністю тільки тих бібліотек, які ви додатково використовуєте у своєму проекті.

Усі модулі фреймворку знаходяться в папці `modules`. Весь вихідний код добре структурований і коментований, так що проблем з розумінням виникнути не потрібно.

У папці `extras` можна знайти приклади використання фреймворку. Серед іншого там ви знайдете дві утиліти, які допоможуть вам у розробці. Для цих утиліт, та й для всіх прикладів, вже є готові проекти для різних платформ та IDE, все що вам залишається це відкрити проект та скомпілювати приклад чи утиліту.

Перша з них це Projucer. JUCE Projucer — це генератор коду та IDE-проектів. Ви вказуєте, який тип проекту ви хочете (наприклад, програма з графічним інтерфейсом, аудіо плагін тощо), і цільові IDE, з якими ви збираєтеся працювати (наприклад, Visual Studio, XCode), а також деякі інші деталі, такі як назва проекту. Після цього Projucer створить повний проект/рішення для кожної IDE, включаючи декілька файлів «початкового» коду. Потім ви можете відкрити кожен проект/рішення у відповідній IDE, натиснути «Побудувати» і за кілька хвилин отримати робочий додаток або плагін для цільової платформи.

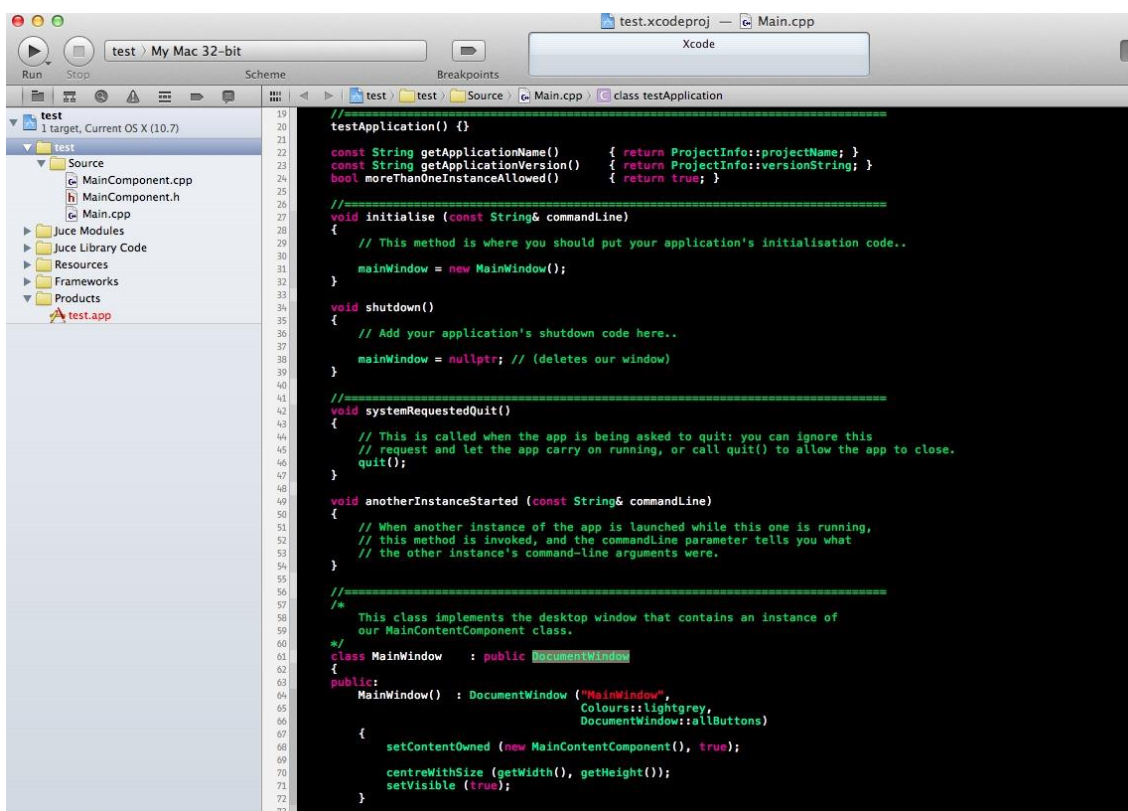


Рисунок 2.5 – Зовнішній вигляд стандартного проекту Projucer

Кажучи іншими словами, Projucer налаштований під усі IDE та платформи, які підтримуються фреймворком. Для перенесення коду від розробника не потрібно нічого, крім наявності на цільовій системі утиліти Projucer. Кожен скомпільований проект можна відкрити в цільовій IDE, яку ви використовуєте для своєї ОС.

Друга програма - це `binarybuilder`, яка дозволяє перенести бінарні файли безпосередньо в код ваших вихідних файлів. Наприклад картинки та логотипи перетворюються на набір бінарних даних, що можуть бути використані у власному коді. Ця утиліта також є частиною `Introjucer` і використовується під час редагування GUI у менеджері ресурсів. Її також можна зібрати, як окрему консольну програму.

### **2.3 Запуск Projucer та створення проекту**

Оскільки всі існуючі програми використовують стандарт VST від Steinberg. Для того, щоб мати всі наявні функції та класи для програмування власного аудіо-плагіну, нам знадобиться завантажити останню версію `Virtual Studio Technology Software Development Kit 3.4.7` на офіційному сайті Steinberg.

Оскільки розробка плагіну буде відбуватись на ОС Windows, нам знадобиться Visual Studio як середовище розробки та збірки аудіо-плагіну. Першим чином, потрібно буде зібрати Projucer із open-source файлів програми в директорії JUCE. Ми отримаємо `.exe` файл програми Projucer, який вже готовий для запуску на Windows.

Після запуску Projucer першої відобразиться сторінка, де користувач зможе створити новий проект. Під час наступних запусків найчастіше ви побачите останній відкритий проект з форматом `.jucer`, який запам'ятовує Projucer. Щоб створити новий проект, потрібно перейти до меню `File Projucer`, закрити будь-який відкритий проект та створити новий за допомогою «New project».



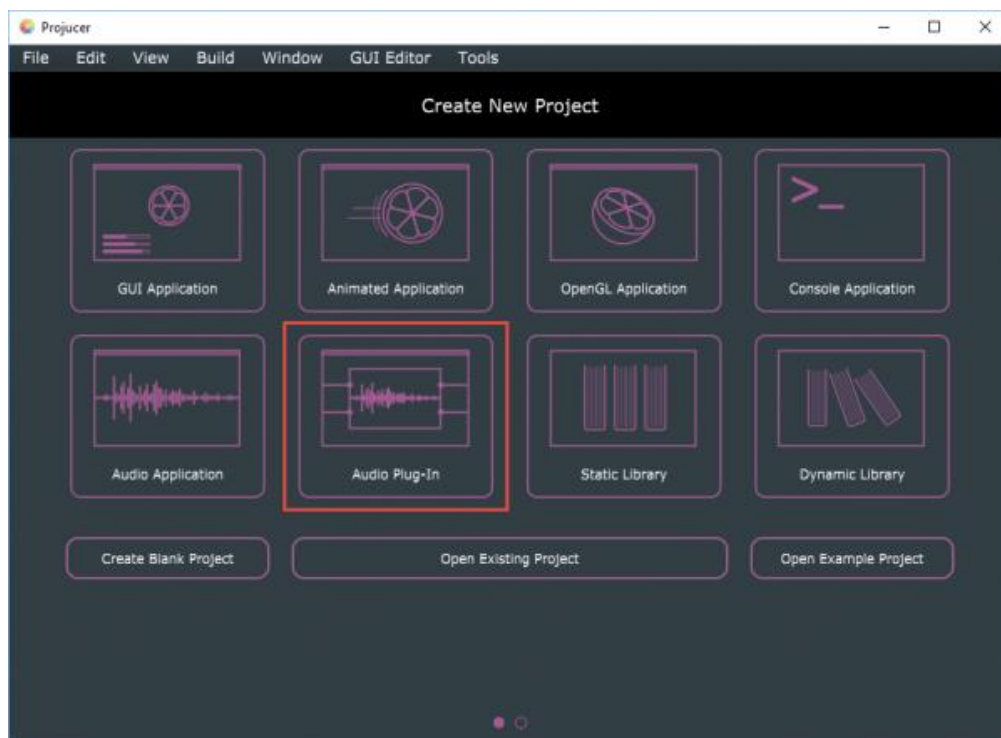


Рисунок 2.6 – Стартовий інтерфейс Projucer

Сторінка «Створити новий проект» містить вісім великих графічних кнопок для восьми найбільш поширених типів проектів. Оскільки ми створюємо аудіо-плагін, нам потрібна опція «Audio Plug-in», що позначена на знімку екрана вище. Це перенесе вас у «другу половину» сторінки створення проекту.

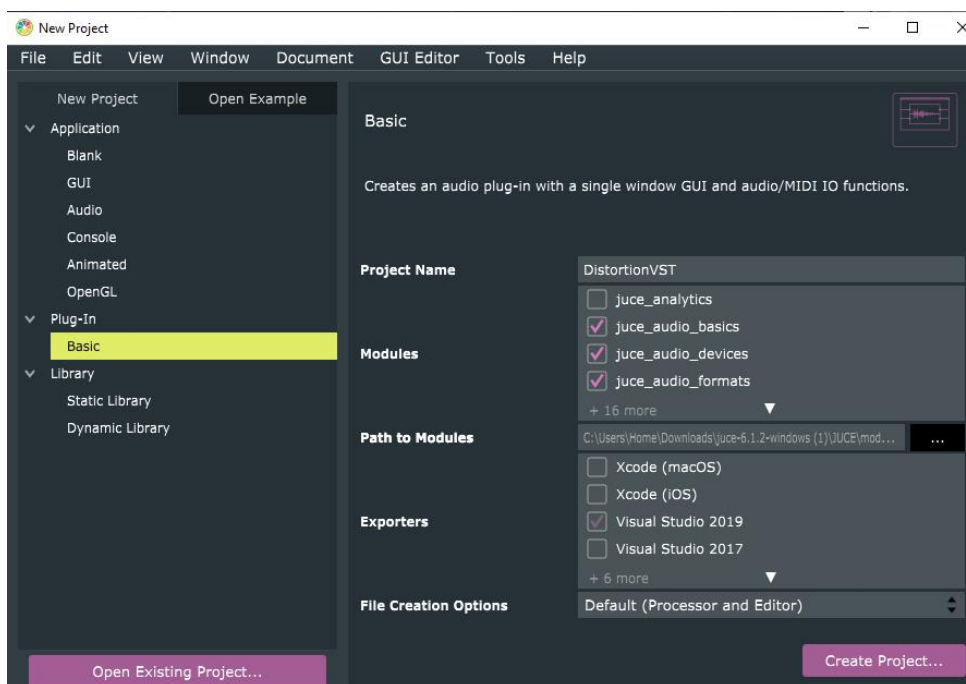


Рисунок 2.7 – Створення проекту в Projucer

У правій половині Projucer, ми маємо вказати ім'я нашого майбутнього проекту, які модулі JUCE ми будемо використовувати (в нашому випадку: `juce_audio_basics`, `juce_audio_devices`, `juce_audio_formats`, `juce_plugin_client`, `juce_processors`, `juce_core`, `juce_data_structures`, `juce_events`, `juce_graphics`, `juce_gui_basics`).

Вказавши де буде розташований наш проект, ми переходимо до поля «Назва проекту». Ми будемо створювати простий аудіо-плагін для електрогітари і його назва буде `DistortionVST`, після застосування налаштувань та створення проекту, на місці розташування буде створена папка `DistortionVST`, де будуть зберігатись файли проекту.

У блоці Exporters ми обираємо середовище розробки. В нашому випадку це буде Visual Studio, для користувачів Mac – це Xcode, для Android та Linux – Android та Linux Makefile відповідно.

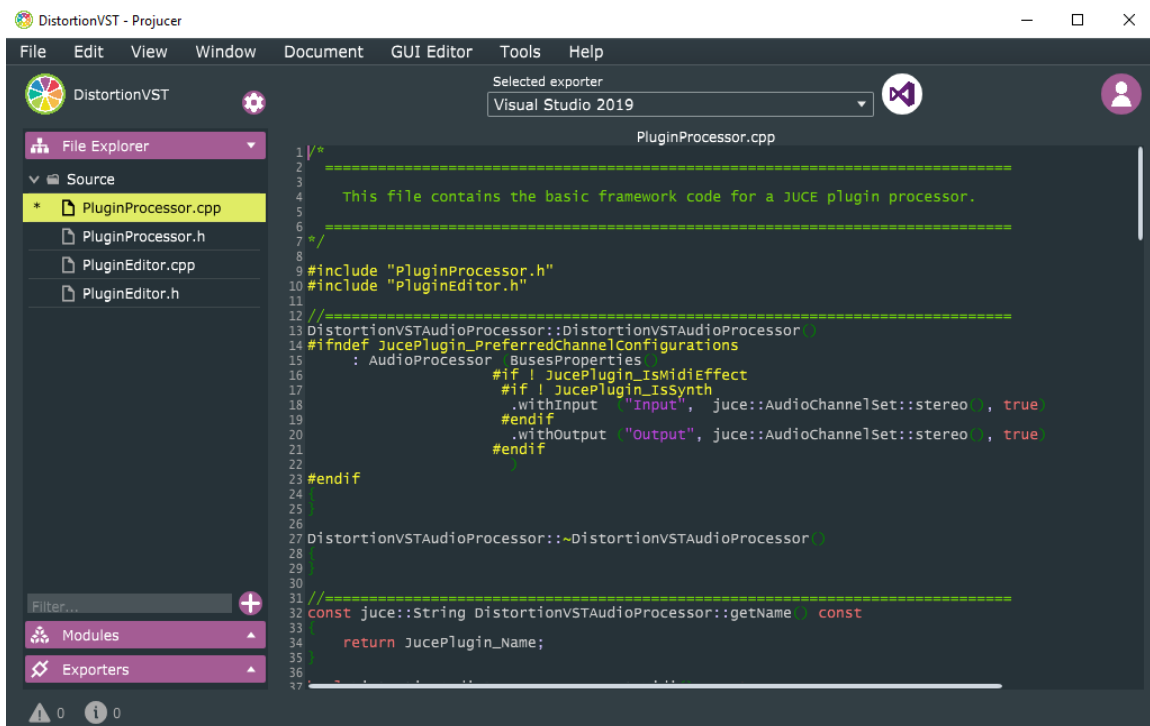


Рисунок 2.8 – Створений проект в Projucer

Після того, як ми створимо проект в програмі, вона автоматично нам згенерує 4 файли, з якими ми зможемо працювати у Visual Studio. В цьому і полягає основна

зручність Projucer. Нам не потрібно самостійно створювати та прописувати основні класи та функції для роботи із аудіо. У новому проекті вже буде написана вся необхідна інформація і вся наша робота при написанні плагіну буде лише обмежуватись викликом класів та функцій, а також написанням GUI для плагіну.

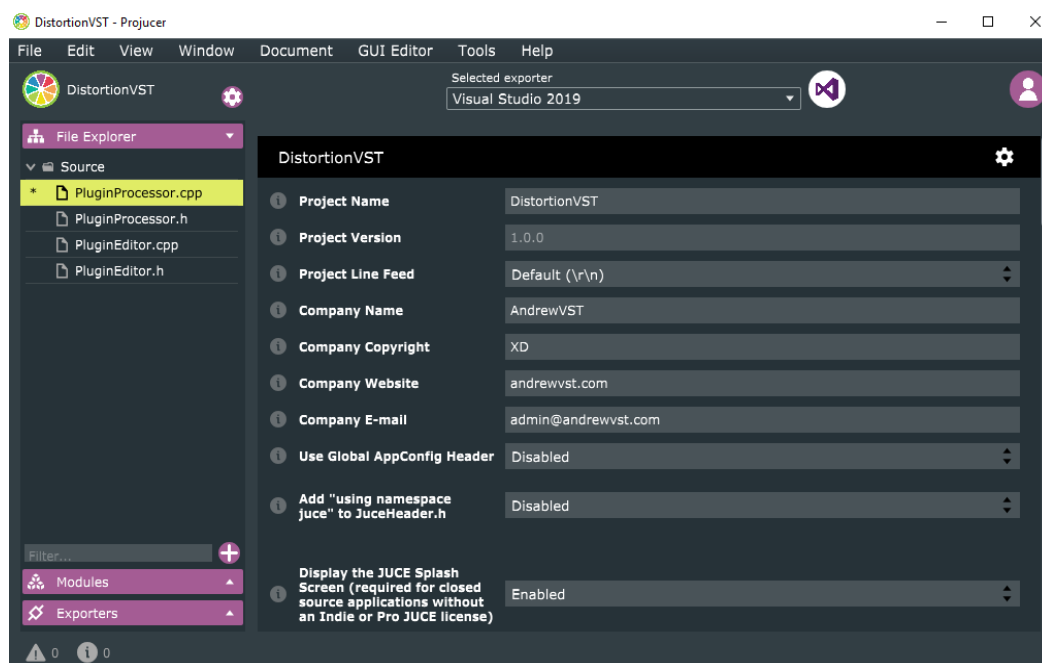


Рисунок 2.9 – Меню налаштувань створеного проекту

Одразу після створення проекту, ми можемо провести кілька базових налаштувань для подальшої комфортної роботи. У полях зверху ми можемо вказати ім'я компанії, автора, електронну пошту, вебсайт, тощо. Поля нижче торкатимуться безпосередньо самого плагіна, В нашому випадку ми помічаємо чек-бокси VST та AU у блоці Plugin Formats, а також додаємо налаштування {1,1},{2,2} до поля Plugin Channel Configurations, що означає, що наш плагін можна буде використати з одним або двома каналами входу та виходу.

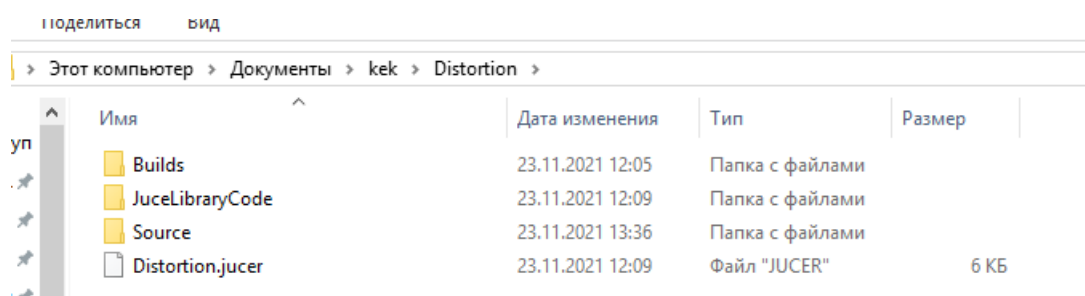


Рисунок 2.10 – Файловий каталог створеного проекту

Projucer створить нову папку для вашого нового проекту, що містить три підпапки та файл формату. `.jucer`:

- файл `.jucer` — це файл проекту Projucer, який містить усі деталі, які було вказано вище;
- папка `Source` містить файли вихідного коду, які є унікальними для цього проекту. Для проекту `Audio Plugin` спочатку буде чотири файли, які будуть редагуватись у `Visual Studio`;
- папка `JuceLibraryCode`, де ми вказали модулі, що будуть використовуватись при написанні плагіни;
- папка `Builds`, в якій розташовані файли проекту `Visual Studio`;

Папка із проектом `VisualStudio` має один файл `.rc` (бінарні ресурси), один файл `.sln` (рішення) і три пари файлів `.vcproj/.vcproj.filters` для трьох окремих проектів `Visual Studio` (`SharedCode`, `StandalonePlugin`, `VST3`). `VST3` – для написання плагіна `VST` (`DistortionVST`), `StandalonePlugin` – для створення окремої програми, `SharedCode` – для збірки файлу `.lib` з усього коду.

З них файл «спільного коду» (`SharedCode`) є найбільшим. Він містить посилання на всі файли коду в папках `Source` і `JuceLibraryCode`, а також на безліч файлів у папці «`JUCE modules`» (саме чому Projucer потрібно було знати, де його знайти). Інші два проекти містять лише пару файлів коду обгортки.

## 2.4 Запуск створеного проекту через Visual Studio

Звісно, код проекту можна редагувати і в самому Projucer, проте компілювання буде відбуватись лише через `Visual Studio`. Після того, як всі налаштування будуть збереженні, ми зможемо запусити проект у `Visual Studio`.

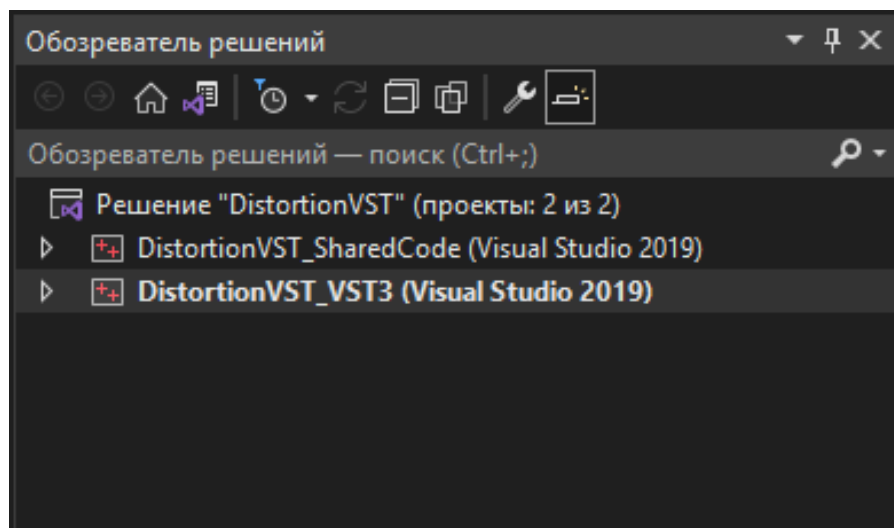


Рисунок 2.11 – Оглядач рішень створеного проекту у Visual Studio

У «оглядачі рішень» ми побачимо 2 проекти, і до кожного з них потрібно під'єднати раніше завантажений Steinberg SDK. Це можна виконати за допомогою опції «властивості», клікнувши лівою кнопкою мишки на проект, та перейшовши до меню «VC++ каталоги» у вікні властивостей, де ми вкажемо шлях до каталогу VST3 всередині архіву VST-SDK.

Тепер коли SDK під'єднано, ми можемо повернутись до файлів нашого проекту та почати написання плагіну. 4 основних файла для редагування будуть зберігатись у нашому проекті, у папці Source: PluginEditor.cpp, PluginEditor.h, PluginProcessor.cpp, PluginProcessor.h.

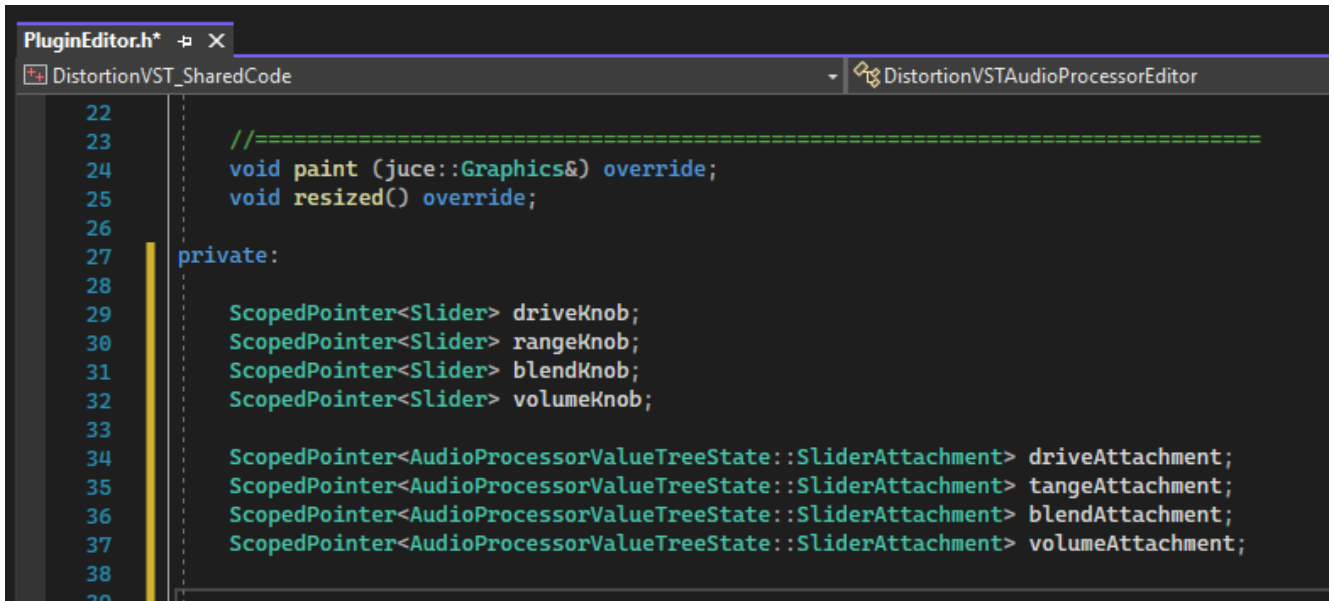
Плагіни, створені за допомогою JUCE, складаються із PluginEditor та PluginProcessor:

Об'єкт PluginEditor відповідає за елементи графічного інтерфейсу користувача (GUI). В ньому ми створимо зовнішній вигляд нашого плагіну.

Об'єкт PluginProcessor займається обробкою звуку. Точніше, процесор пов'язує наш код обробки з платформою, на якій він працює. Оскільки ми створюємо плагін VST, процесор з'єднає всі необхідні входи та виходи, щоб ми могли використовувати плагін у DAW.

Плагіни за допомогою JUCE пишуться на мові C++.

## 2.5 Написання графічного інтерфейсу плагіну



```

22
23 //=====
24 void paint (juce::Graphics&) override;
25 void resized() override;
26
27 private:
28
29     ScopedPointer<Slider> driveKnob;
30     ScopedPointer<Slider> rangeKnob;
31     ScopedPointer<Slider> blendKnob;
32     ScopedPointer<Slider> volumeKnob;
33
34     ScopedPointer<AudioProcessorValueTreeState::SliderAttachment> driveAttachment;
35     ScopedPointer<AudioProcessorValueTreeState::SliderAttachment> rangeAttachment;
36     ScopedPointer<AudioProcessorValueTreeState::SliderAttachment> blendAttachment;
37     ScopedPointer<AudioProcessorValueTreeState::SliderAttachment> volumeAttachment;
38
39

```

Рисунок 2.12 – Створення GUI плагіну

Графічний дизайн аудіо-плагіну складатиметься з 4 регуляторів ефектів (knobs): регулятор перевантаження (driveKnob), регулятор дальності (rangeKnob), регулятор міксу (blendKnob) та регулятор гучності (volumeKnob). Кожен із регуляторів буде посиланням на об'єкт класу ScopedPointer, який створить нам повзунок (Slider) для налаштування рівня навантаження звукового ефекту.

Далі ми маємо задати розмір вікна для плагіну та відобразити створені об'єкти у файлі PluginEditor.cpp, в конструкторі DistortionVSTAudioProcessorEditor. Ми можемо використати розширення 500x200, відредагувавши функцію setSize(), та додавши об'єкти за допомогою addAndMakeVisible(). Параметри відображення об'єктів ми будемо задавати за допомогою параметрів setSliderStyle, та setTextBoxStyle для текстового позначення ефектів. Щоб наш повзунок рухався та налаштовував рівень навантаженості ефекту ми також створили об'єкти «Attachments», які будуть виступати у ролі регулювальників ефектів. Attachments, в свою чергу, зчитуватимуть «стан» навантаженості ефекту за допомогою функції getState(), яку ми пропишемо пізніше.

```

//=====
DistortionVSTAudioProcessorEditor::DistortionVSTAudioProcessorEditor (DistortionVSTAudioProcessor& p)
: AudioProcessorEditor (&p), audioProcessor (p)
{
    addAndMakeVisible(driveKnob = new Slider("Drive"));
    driveKnob->setSliderStyle(Slider::Rotary);
    driveKnob->setTextBoxStyle(Slider::NoTextBox, false, 100, 100);

    addAndMakeVisible(rangeKnob = new Slider("Range"));
    rangeKnob->setSliderStyle(Slider::Rotary);
    rangeKnob->setTextBoxStyle(Slider::NoTextBox, false, 100, 100);

    addAndMakeVisible(blendKnob = new Slider("Blend"));
    blendKnob->setSliderStyle(Slider::Rotary);
    blendKnob->setTextBoxStyle(Slider::NoTextBox, false, 100, 100);

    addAndMakeVisible(volumeKnob = new Slider("Volume"));
    volumeKnob->setSliderStyle(Slider::Rotary);
    volumeKnob->setTextBoxStyle(Slider::NoTextBox, false, 100, 100);

    driveAttachment = new AudioProcessorValueTreeState::SliderAttachment(p.getState(), "drive", *driveKnob);
    rangeAttachment = new AudioProcessorValueTreeState::SliderAttachment(p.getState(), "range", *rangeKnob);
    blendAttachment = new AudioProcessorValueTreeState::SliderAttachment(p.getState(), "blend", *blendKnob);
    volumeAttachment = new AudioProcessorValueTreeState::SliderAttachment(p.getState(), "volume", *volumeKnob);

    // Make sure that before the constructor has finished, you've set the
    // editor's size to whatever you need it to be.
    setSize (500, 200);
}

```

Рисунок 2.13 – Налаштування розміру вікна та об'єктів

Також у конструкторі `resized()` ми прописуємо розміри регуляторів і їх відображення по середині вікна. Такі маніпуляції можна зробити за допомогою `setBounds()`.

```

void DistortionVSTAudioProcessorEditor::resized()
{
    driveKnob->setBounds(((getWidth() / 5) * 1) - (100 / 2), (getHeight() / 2) - (100 / 2), 100, 100);
    rangeKnob->setBounds(((getWidth() / 5) * 1) - (100 / 2), (getHeight() / 2) - (100 / 2), 100, 100);
    blendKnob->setBounds(((getWidth() / 5) * 1) - (100 / 2), (getHeight() / 2) - (100 / 2), 100, 100);
    volumeKnob->setBounds(((getWidth() / 5) * 1) - (100 / 2), (getHeight() / 2) - (100 / 2), 100, 100);
    // This is generally where you'll want to lay out the positions of any
    // subcomponents in your editor..
}

```

Рисунок 2.14 – Налаштування розміру об'єктів

Тепер нам необхідно прописати функцію `getState()`, щоб Attachments змогли отримувати інформацію про стан навантаженості ефекту і регулювати його. Усі дії із таким функціоналом проводяться у файлах `PluginProcessor`. Для початку в `.h` файлі, в приватній частині класу `DistortionVSTAudioProcessor` ми створимо об'єкт типу `AudioProcessorValueTreeState` з назвою `state`, а в публічній частині створимо геттер, який повідомлятиме про стан того чи іншого ефекту.

```

    AudioProcessorValueTreeState& getState();

private:

    ScopedPointer<AudioProcessorValueTreeState> state;
    //=====================================================================
    JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR (DistortionVSTAudioProcessor)
};

```

Рисунок 2.15 – Написання функціоналу у PluginProcessor

Після цього ми маємо оголосити саму функцію `getState()` у `.cpp` файлі, в цьому випадку вона буде посиланням на приватний об'єкт `state`.

```

#endif
{
    state = new AudioProcessorValueTreeState(*this, nullptr);
}

AudioProcessorValueTreeState& DistortionVSTAudioProcessor::getState() {
    return *state;
}

```

Рисунок 2.16 – Оголошення функції `getState()` та об'єкту `state` у `cpp` файлі

Тепер коли GUI нашого плагіну повністю прописане, ми можемо зібрати його та запустити через FL Studio. Після збірки у файлах проекту з'явиться файл із форматом `.dll`, це і є макет того, що буде із себе представляти цей плагін.

Этот компьютер > Документы > kek > DistortionVST > Builds > VisualStudio2019 > x64 > Debug > VST3 >

Имя	Дата изменения	Тип	Размер
Distorti.1424C173.tlog	30.11.2021 14:54	Папка с файлами	
<b>DistortionVST.dll</b>	30.11.2021 14:04	Расширение при...	9 878 КБ
DistortionVST.exp	30.11.2021 0:48	Exports Library File	1 КБ
DistortionVST.ilc	30.11.2021 14:04	Incremental Linke...	40 817 КБ
DistortionVST.lib	30.11.2021 0:48	Object File Library	3 КБ
DistortionVST.pdb	30.11.2021 14:04	Program Debug D...	54 676 КБ
DistortionVST.vst3.recipe	30.11.2021 14:04	Файл "RECIPE"	1 КБ
DistortionVST_VST3	30.11.2021 14:04	Текстовый докум...	1 КБ
DistortionVST_VST3.vcxproj.FileListAbsol...	30.11.2021 0:48	Текстовый докум...	1 КБ
include_juce_audio_plugin_client_VST3.obj	30.11.2021 0:48	Object File	7 464 КБ
resources.res	30.11.2021 0:48	Compiled Resourc...	1 КБ

Рисунок 2.17 – Зібраний `.dll` файл макету плагіну



FL Studio імпортує VST-плагіни користувача з однієї папки, шлях до якої можна вказати в налаштуваннях. Якщо не змінювати шлях вручну, то за замовчуванням усі VST-плагіни імпортуються до програми із C:\Program Files (x86)\VstPlugins, куди ми скопіюємо DistortionVST.dll.

Після того, як ми зайдемо у FL Studio, у вікні Mixer ми маємо натиснути лівою кнопкою миші на Slot 1 (або будь-який інший канал, в залежності від того, куди буде підключений інструмент, щоб застосувати ефекти), далі Select і More Plugins. У новому вікні у нас буде відображений список усіх наявних плагінів (стандартні плагіни, що йдуть із комплектом FL Studio). Ми пролистаємо список зверху вниз, але не бачимо нашого плагіна, оскільки список плагінів ще не оновлено, і тому нам знадобиться додатково натиснути кнопку Manage в кінці списку, яка, в свою чергу, відкриває вікно плагін-менеджеру. У вікні FL Studio Plugin Manager, що з'явиться, ми натискаємо на "Scan", після чого чекаємо, поки програма відсканує всі присутні плагіни і оновить список. Після цього закриваємо вікно та у списку плагінів зможемо побачити наш DistortionVST.

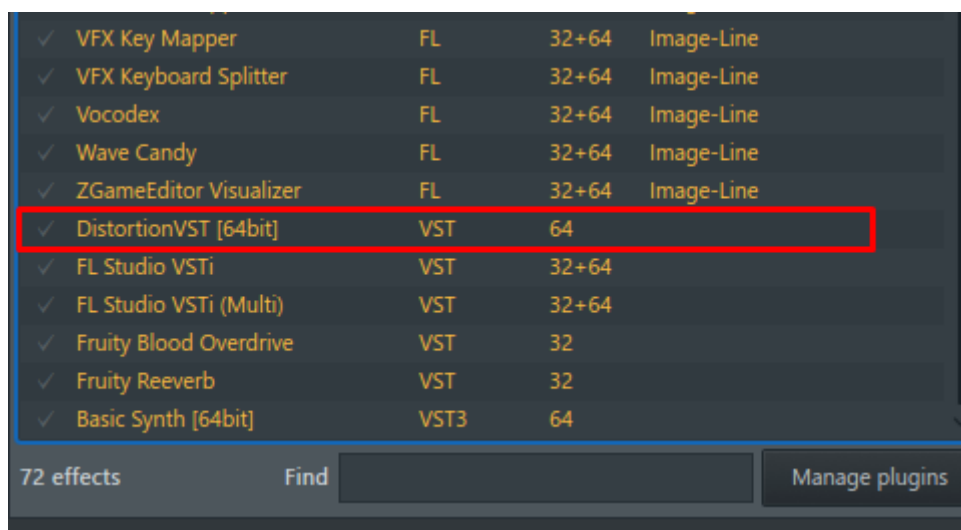


Рисунок 2.18 – Плагін успішно завантажився в FL Studio

Двічі натискаємо на DistortionVST і відкривається вікно створеного плагіна (а точніше його макету поки), де вже повністю функціональні регулятори ефектів і їх можна покрутити. Для того щоб програма працювала як це замислюється і

накладала ефекти на звук, нам звісно необхідно прописати додатковий функціонал. Бо наразі у нас працездатні лише регулятори.

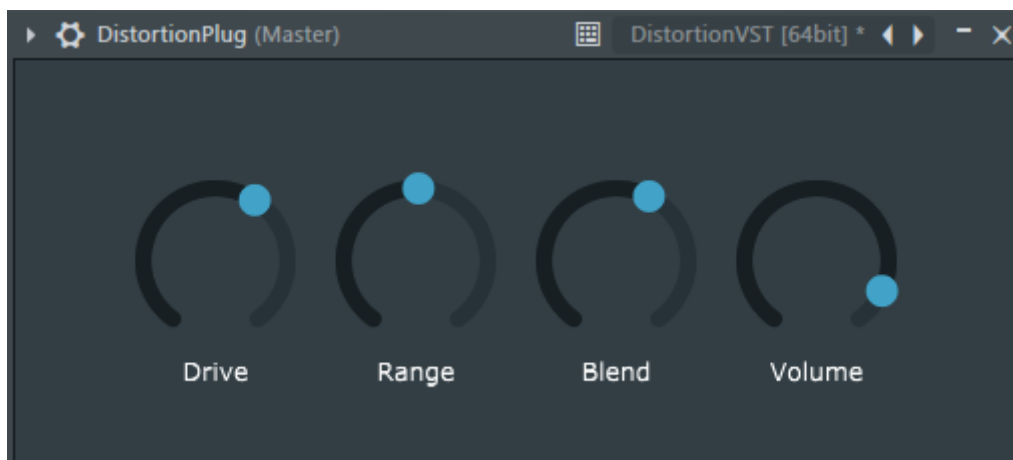


Рисунок 2.19 – Зовнішній вигляд плагіну

У роботі буде написано лише 4 найпростіших ефектів дисторшну, оскільки для введення додаткових функцій і можливостей потребує чимало часу, і розробкою професійних плагінів, звісно, займається ціла команда інженерів. Однак для демонстрації працездатності фреймворку JUCE і як його можна застосовувати для написання плагінів, цього буде цілком достатньо, оскільки без такого фреймворка важко уявити яким би важким і довгим виглядав наш написаний код. JUCE містить у собі вже всі необхідні написані бібліотеки для спрощення роботи програміста до написання всього лише кількох рядків, щоб забезпечити функціонал того чи іншого елемента для роботи із аудіо.

## 2.6 Ефект дисторшн

Дисторшн (також «дистошн», від англ. *distortion* — спотворення) — різновид звукового ефекту чи використовуваного його отримання приладу. Досягається *distortion* за рахунок «жорсткої» обрізки амплітуди аудіосигналу. Дисторшн є широко-розповсюдженим у таких музичних жанрах як хард-рок, метал і панк-рок, де він є основним гітарним ефектом. Крім того, за його допомогою обробляють звучання драм-машини в хардкор-техно (переважно це спідкор та брейккор). Нерідко під цим визначенням ховається ціла категорія принципово однакових ефектів (типу овердрайву, фуззи та інших), принцип роботи яких заснований на

виникненні нелінійного спотворення аудіосигналу. Вони отримали назву ефектів «перевантаження», а дозволяють домогтися їх пристрою — «спотворювачі», а музичному середовищі їх називають «примочки».

Поміж дисторшну також виділяють наступні ефекти «спотворення»:

Overdrive - спотворення досягаються шляхом "м'якого кліпування", надаючи "теплоти" звуку, найчастіше досягається за допомогою перевантаження ламп.

Fuzz - спотворення досягаються шляхом "жорсткого кліпування", створює сильне, брудне спотворення.

Warp - додає грубі спотворення і дуже довгий сустейн, перетворюючи підсилювач на "звіра", що гарчить.

Waveshaper - дуже гнучкий ефект, користувачеві пропонується створити передавальну характеристику самому, що дозволяє використовувати цей ефект хоч як жорсткий обмежувач, хоч як софтбокс, овердрайв та багато іншого.

Tube Saturation/Valve - "ужирнювач", "утеплювач" ефект лампової сатурації. Принцип дії схожий на софтбокс, сатурація плавно "розплющує" верхівки хвиль високих сигналів, піднімаючи ділянки низького рівня і збільшуючи відчуття щільності звуку. Valve і tube — англійська та американська назви лампи.

Bitcrusher - принцип дії полягає у зниженні бітності та/або частоти дискретизації сигналу.

Ефект використовується не тільки в поєднанні з електрогітарою — він підходить і до інших інструментів (наприклад, бас-гітарі). Але для обробки звучання бас-гітари використовується кілька змінених ефектів, оскільки звичайний гітарний дисторшн не здатний передати весь спектр низьких частот. А їхнє обрізання негативно відбивається на якості звучання. Як альтернативу бас-гітарю обробляють звичайним дисторшном з наступним рівним змішуванням початкового та спотвореного сигналу. Крім того, дисторшн використовується для обробки звучання смичкових інструментів та вокальних партій.

Синтезатори, гітарні ефект-процесори, а також ПЗ для комп'ютерів, що застосовується для звукової обробки, мають у своєму складі дисторшн як один з компонентів.

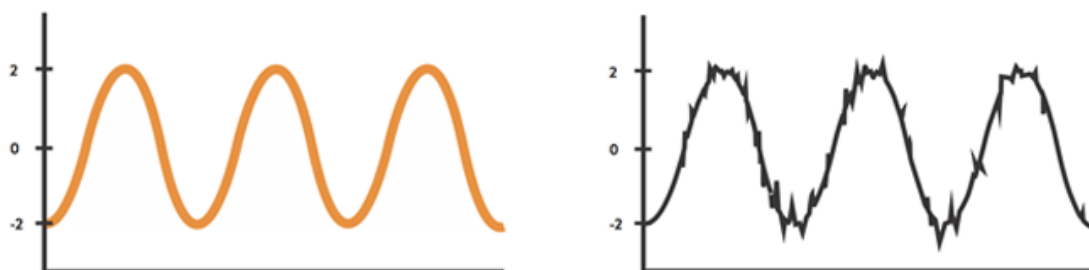


Рисунок 2.20 – Схема спотворення звукової амплітуди чистого сигналу

У його основі лежить звуковий ефект, що виникає під час перевантаження підсилювачів. Він схожий з кліпінгом, що є, по суті, різновидом дисторшну. Суть дисторшна у цьому, що у побудованих на лампах чи транзисторах підсилювачах з'являються нелінійні спотворення аудіосигналу, особливо у пікових значеннях гучності. Порівняно з традиційним кліпінгом і схемами, в яких використовується елементарний обмежувач, «перевантаження» (особливо якщо перевантажувати ламповий пристрій) характеризується складним взаємозв'язком між спектральними компонентами з амплітудою та спектральним складом аудіосигналу. Отриманий за допомогою лампового пристрою перевантаження зазвичай цінується набагато вище «транзисторного» дисторшн-ефекту.

Враховуючи передатні властивості будь-якого підсилювача, слід зазначити, що вихідний аудіосигнал змінюється у них за зміни вхідного. Зазвичай посилення слабкого вхідного аудіосигналу здійснюється без спотворень (або з мінімальними), але зі зростанням його амплітуди збільшуються значення нелінійних спотворень. Їх величина залежить від безлічі нюансів: типу підсилювача, його схемотехніки, глибини та значення зворотного зв'язку, та ін. Спотворення варіюється у великому діапазоні. Найчастіше лінійна характеристика пристроїв відносна в широкому спектрі амплітуд аудіосигналу, що виходить, але у разі перевищення певного порогового значення відбувається вихід вихідного каскаду з лінійного режиму, що

призводить до зростання рівня нелінійних спотворень. За допомогою ручки "Посилення" ("Gain") можна збільшити значення посилення пристрою, що є аналогом збільшення амплітуди вхідного аудіосигналу і призводить до зростання спотворень. Вони називаються нелінійними через виникнення нових спектральних складових у звуковому спектрі.

Виходить, що при подачі на вхідний каскад підсилювача аудіосигналу у формі чистого синуса на виході утворюється спотворена синусоїда з присутністю гармонік. Але подібний метод дозволяє отримати ефект дисторшн тільки за великих гучностей. Для отримання тихого спотвореного сигналу потрібно використання спеціальних каскадів, що спотворюють, здатних надавати звуку нелінійність в широкому спектрі амплітуд.

Типова конфігурація підсилювача складається з передпідсилювача («преампа») та підсилювача потужності («потужника»). Тому досягти ефекту дисторшн можна лише за допомогою перевантаження або самого підсилювача, або пред-підсилювача.

Широко поширені як аналогові, і цифрові прилади, що емулюють множинні варіанти ефекту дисторшн. Окремі схеми навіть дають змогу імітувати звучання перевантаження підсилювачів від популярних брендів.

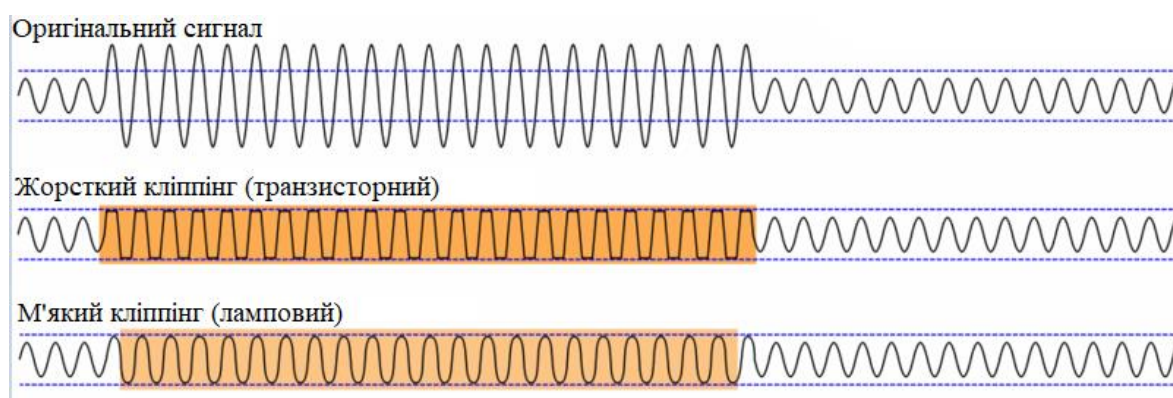


Рисунок 2.21 – Різниця між транзисторним та ламповим кліппінгом

Традиційно розрізняють два види навантажень звукового сигналу – ламповий та транзисторний. Ламповий овердрайв є формою м'якого кліпування, тоді як

транзисторне кліпування нагадують жорстке кліпування за рахунок надто перевантажених ламп.

До широкого застосування транзисторів, підсилювачі та сповіщувачі традиційно робили на основі вакуумних ламп. Вони мають максимальну вхідну напругу, вище якої починається зміна сигналу, що посилюється, а також мінімальне, при якому сигнал також змінюється. Коли якась частина вхідного сигналу підходить до цієї межі, посилення сигналу лампами стає менш лінійним таким чином, що частини сигналу, що мають меншу напругу, посилюються більше ніж ті, що мали більшу напругу. Це викликає стиск піків вихідного сигналу, внаслідок чого сигнал виглядає сплющеним. Такий ефект називається "м'яким кліпуванням", це також генерує нові гармоніки, які додають "тепла" та збагачують тембр інструменту. Якщо лампа підсилює звук ще сильніше, стиснення стає екстремальнішим і піки сигналів обрізаються. Це додає додаткові гармоніки непарного порядку, створюючи "брудний" або "пухкий" тембр.

Лампові сповіщувачі зазвичай називають "Overdrive", спотворення досягається шляхом перевантаження ламп у підсилювачі або за допомогою спеціальних пристроїв. При посиленні/кліпуванні сигналу може використовуватися кілька каскадів ламп, це створює більш "товсте" і складніше спотворення звуку. У деяких сучасних лампових ефектах, "брудний" тембр насправді досягається не за рахунок високої напруги, а за допомогою запуску в ланцюг напруги, яка надто низько для компонентів схеми, в результаті чого виникає велика нелінійність і спотворення. Ці схеми називають "starved plate", в результаті "смерті" амплітуди звуку.

Транзистори поведуться набагато лінійніше і таким чином точніше посилюють сигнал, поки вхідна напруга не виходить за межі робочої області. При перевищенні межі сигнал буде кліпувати без стиснення, це відомо як "жорстке кліпування" або "жорстке лімітування". Цей тип спотворення робить більше непарних гармонік. В електронному вигляді, це зазвичай досягається або

посиленням сигналу в точці де він повинен бути кліповані, або кліпування відбувається на діодах.

### 2.6.1 Реалізація дисторшну

Педальні overdrive/distortion. Оскільки вони часто призначені для роботи з низькою напругою, таким як 9-вольтовою батареєю, перевантаження та спотворення досягаються за допомогою використання транзисторів. Класичні приклади Ibanez Tube Screamer та Electro-Harmonix Big Muff. Деякі сучасні педальні ефекти використовують лампи, зазвичай вони також працюють при напругах, які знаходяться нижче лінійної поведінки ламп, в результаті чого виходять різкі та buzzy спотворення. Зазвичай педальні дисторшні посилюють сигнал, який може бути використаний для вхідного каскаду передусилювача, що призводить до подальшого спотворення, і в деяких випадках збільшення рівня.

Спотворення підсилювача. Гітарний підсилювач служить посилення слабкого сигналу інструменту до рівня який може керуватися підсилювачем потужності. Підсилювач часто містить схеми формування тембру інструменту, зокрема еквайзер і параметр посилення рівня (gain). Для отримання спотворень часто використовують кілька каскадних етапів посилення/кліпування. Оскільки перший компонент лампового підсилювача - це лампи, вихідний рівень попередніх елементів сигнального ланцюга має сильний вплив на спотворення на етапі підсилювача.

У проміжку 1980 - 1990 років, багато підсилювачів мали ручку "master volume", яка по суті була атенюатором між секціями передусилювача і підсилювача, що дозволяло зручно отримувати спотворення високого рівня в секції передусилювача гітарного підсилювача, не навантажуючи лампи. Однак, це також призводить до того, що лампи підсилювача працюють в межах лінійної області посилення, що зменшує спотворення, що додаються до вихідного сигналу.

## 2.7 Додавання компонентів ефекту до плагіну

Усі компоненти ефекту DistortionVST ми будемо прописувати безпосередньо у конструкторі processBlock у файлі PluginProcessor.cpp. Змінна totalNumInputChannels означає кількість каналів, для яких буде використовуватись наш плагін. Якщо ви використовуєте моно-пристрій для вводу/виводу аудіо, то буде використовуватись лише один канал, якщо стерео-пристрій – то 2 канали. В нашому випадку використовуються 2 канали.

Для кожного каналу створюється вказівник до буферу, що є насправді масивом значень типу float або звукові дані, що будуть надходити до каналу та конвертуватись за допомогою цифро-аналогового перетворювача, що працює на кожній звуковій карті.

Ми створимо змінні з типом float для кожного компоненту ефекту, щоб налаштувати їх значення через повзунок.

```
float drive = *state->getRawParameterValue("drive");
float range = *state->getRawParameterValue("range");
float blend = *state->getRawParameterValue("blend");
float volume = *state->getRawParameterValue("volume");
```

Рисунок 2.22 – Змінні майбутніх ефектів типу float

Після цього нам необхідно створити цикл for, що «спотворюватиме» за допомогою нашого ефекту кожен звуковий біт, що потрапить до нашого каналу в масив channelData.

```
for (int channel = 0; channel < totalNumInputChannels; ++channel)
{
    auto* channelData = buffer.getWritePointer (channel);

    for (int sample = 0; sample < buffer.getNumSamples(); sample++) {
        channelData++;
    }
}
```

Рисунок 2.23 – Базовий вигляд обробки звуку без ефектів



Тепер нам необхідно створити математичну функцію ефекту, який ми збираємось застосовувати при обробці звукових бітів.

Щоб візуалізувати те, що ми збираємося зробити зі звуком на нашому плагіні, ми можемо скористатися безкоштовним додатком Geogebra, який можна як завантажити окремо на комп'ютер, так і запустити в браузері. Це дуже корисний інструмент для того, щоб візуально представити математичні графіки, що дуже корисно при роботі з обробкою аудіо та створенні плагінів.

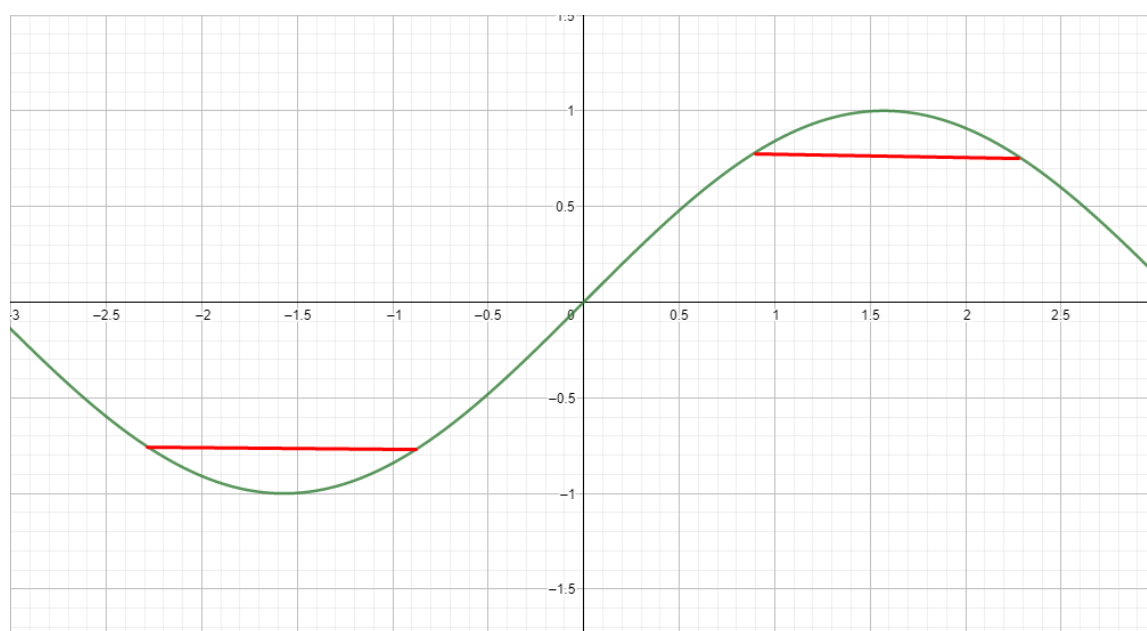


Рисунок 2.24 – Візуалізація простого дисторшну на графіку синусоїдної функції

Звук, що подається на канал буде сприйматись у формі звичайної синусоїди. Ефект «дисторшну» має спотворювати цю синусоїду кліппінгом, створивши спотворення сигналу, якщо звукова амплітуда досягне значення 0,8 або -0,8 (наприклад), як показано на рисунку 2.20. Тобто, коли хвиля досягає цих значень, вона не буде створювати дугу, а замість цього звук піде по червоній ліній, саме це і створить ефект «спотворення» або «дисторшну».

Проте це звісно дуже жорстка форма ефекту і на виході ми отримуємо забагато шуму та сирого, неякісного звуку, замість цього ми можемо використати іншу функцію для того, щоб досягти непоганого результату.

Замість цього ми можемо використати функцію  $f(x) = \frac{2}{\pi} * \operatorname{tg}^{-1}(x)$

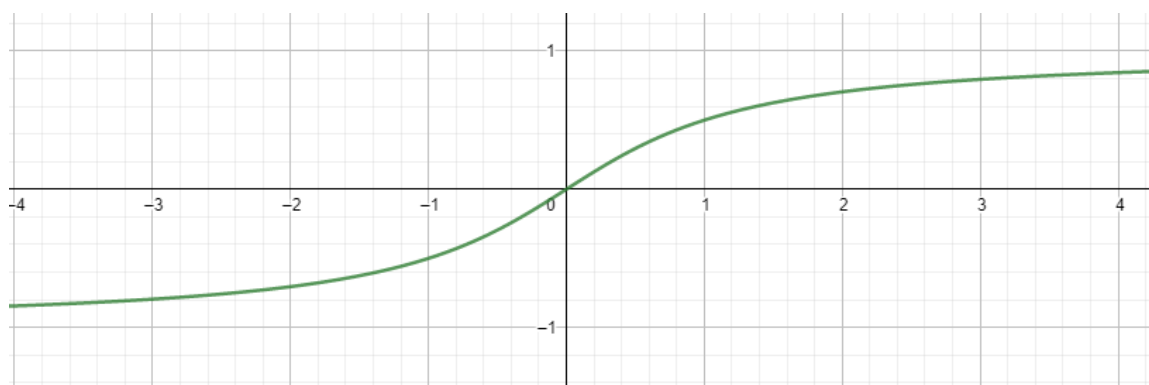


Рисунок 2.25 – Графік функції

На виході ми отримаємо плавну функцію, без жорсткого обрізання звукової хвилі. Тепер ми можемо записати цю формулу в код та додати ефект дисторшну, який буде залежити від навантаженості повзунків «drive» та «range», оскільки вони взаємопов'язані.

```
*channelData *= drive * range;
*channelData = (2.f / float_Pi) * atan(*channelData);
*channelData++;
```

Рисунок 2.26 – Імплементация функції в кодi

Далі нам потрібно прописати «blend» для нашого звукового ефекту, який буде виступати в ролі «міксу» чистого та спотвореного звуку. Для цього нам необхідно також створити змінну типу float, що буде зберігати чистий звук до того, як його буде спотворено, ми її назвемо cleanSignal. До channelData нам необхідно додати змінну blend та змінну cleanSignal та поділимо її на 2, щоб отримати середній мікс чистого та спотвореного звуку. І також додаємо до channelData змінну volume, яка відповідатиме за рівень гучності.

```
float cleanSignal = *channelData;

*channelData *= drive * range;

*channelData = ((2.f / float_Pi) * atan(*channelData) * blend) + ((cleanSignal * (1.f / blend))/2)*volume;

*channelData++;
```

Рисунок 2.27 – Імплементация «blend» в кодї

Таким чином ми створили базовий ефект Distortion для обробки звуку через змінну channelData. Тепер нам необхідно задати параметри вхідних значень для drive, range, blend та volume у тому ж PluginProcessor.cpp в конструкторі DistortionVSTAudioProcessor. Для range ми поставимо значення параметру 3000, щоб створити тяжкий звук «дисторшну», для volume ми можемо поставити значення 3.

```
#endif
{
    state = new AudioProcessorValueTreeState(*this, nullptr);

    state->createAndAddParameter("drive", "Drive", "Drive", NormalisableRange<float>(0.f, 1.f, 0.0001), 1.0, nullptr, nullptr);
    state->createAndAddParameter("range", "Range", "Range", NormalisableRange<float>(0.f, 3000.f, 0.0001), 1.0, nullptr, nullptr);
    state->createAndAddParameter("blend", "Blend", "Blend", NormalisableRange<float>(0.f, 1.f, 0.0001), 1.0, nullptr, nullptr);
    state->createAndAddParameter("volume", "Volume", "Volume", NormalisableRange<float>(0.f, 3.f, 0.0001), 1.0, nullptr, nullptr);

    state->state = ValueTree("drive");
    state->state = ValueTree("range");
    state->state = ValueTree("blend");
    state->state = ValueTree("volume");
}
```

Рисунок 2.28 – Параметри початкових значень

Збираємо рішення, отримуємо готовий файл .dll в каталогах DistortionVST\Builds\VisualStudio2019\x64\Debug\Shared Code і копіюємо його в папку з плагінами для FL Studio, потім переходимо в FL Studio і запускаємо готовий плагін.



Рисунок 2.29 – Запуск готового плагіну в FL Studio

Тепер підключаємо плагін на перший канал, а також включаємо гітарний підсилювач та електрогітару, щоб перевірити працездатність плагіна. Як показано на звукових шкалах зліву у нижньому кутку та вгорі біля секундоміра на рисунку 2.25, плагін успішно приймає та обробляє вхідний звуковий сигнал з підсилювача. Звук не дуже «спотворюється» при таких налаштуваннях ефекту, оскільки повзунки Drive та Range викручені на мінімальних значеннях, а Blend на високому значенні, що дає нам звучання переважно чистого звуку.



Рисунок 2.30 – Плагін із Drive та Range на високих налаштуваннях

Тепер накрутимо Drive та Range, а Blend закрутимо назад, щоб побачити та почути ефект дисторшну, а також графічно порівняємо «чистий» звук з підсилювача та «спотворений» звук.



Рисунок 2.31 – Порівняння чистого звуку та спотвореного звуку

VST-плагін працює за будь-яких умов і його можна використовувати при роботі з різним музичним обладнанням, будь то гітарний підсилювач, синтезатор або мікрофон. Звичайно, ефекту, що ідеально звучить, важко добитись шляхом, який був описаний у цьому розділі, проте цей проект демонструє потенціал і можливості фреймворку JUCE, за допомогою якого професійні команди програмістів також створюють і продають свої музичні плагіни

## РОЗДІЛ 3 НАЛАШТУВАННЯ ДОМАШНЬОЇ МУЗИЧНОЇ СТУДІЇ

### 3.1 Збірка ПК для комфортної роботи

При збірці ПК виключно для роботи із музикою варто подумати про ресурсоємність. Дорогі карти графічного процесора чудово підійдуть, якщо ваша машина буде використовуватися як робоча станція для ігор або відео, але для створення виключно музики це не так важливо. Що важливо для музики, це можливість швидкого доступу до даних і певним чином їх обробляти.

Особливо актуальна ресурсоємність при роботі із «семплами», коли буде життєво-необхідно отримати доступ до великої кількості файлів і відтворити їх без збоїв. Або якщо, можливо, ви продюсуєте для групи і типовий проект чи аранжування охоплює багато треків, кожен наповнений віртуальними інструментами чи плагінами. Тут вам потрібна машина, яка може швидко й ефективно виконувати всі процеси.

Хоча більшість брендів цифрових аудіо-програм (DAW) пропонують «мінімальні» технічні характеристики свого софту і, кожен, хто коли-небудь користувався ними, знає, що цей умовний мінімум необхідний лише для того, щоб програма запустилась, не гарантуючи комфортну роботу в ній. Як і в будь-якому випадку, чим більше грошей ви витрачаєте, тим краще ваше очікування від компоненту, що ви купуєте.

Заходячи здалеку, варто відразу сказати, що роблячи збірку під аудіообробку варто вибирати процесор з лінійки Intel i5 і вище, а також думати про мінімальний об'єм оперативної пам'яті у 8 ГБ. Достатній обсяг оперативної пам'яті – це основа для створення музики на ПК. Знову ж таки, мінімальні характеристики DAW скажуть, що потрібно 4 ГБ, проте навантаження, що можуть відбуватись у програмі при такому об'ємі можуть шокувати. Повільний час завантаження, глючне аудіо та жорсткі обмеження на розмір багатодоріжкових композицій чекають, якщо об'єм вашої ОП нижче мінімальних 8 ГБ.

Інша ключова область - пам'ять ваших накопичувачів. Для роботи з музикою пріоритетніше мати на увазі вибір SSD замість жорстких дисків, оскільки швидкість передачі даних набагато вища, тому витягувати великі аудіофайли з пам'яті набагато швидше та надійніше. Великі SSD, однак, є відносно дорогими, тому поєднання SSD і HDD є одним з можливих рішень.

Крім цього, потрібно підібрати систему охолодження, графіку, материнську плату та корпус, щоб все вмістити. Охолодження є життєво важливим компонентом будь-якого комп'ютера, і є кілька варіантів, які слід розглянути. З іншого боку, можна наповнити корпус вентиляторами та регулювати температуру за допомогою повітря. Це доступніше, але може створити проблеми, якщо ви намагаєтеся записувати акустичні інструменти в одній кімнаті через білий шум. Крім того, рідинне охолоджуюче рішення використовує подібні інженерні методи, що й автомобіль, коли комбінація охолодженої рідини – зазвичай дистильованої води – проходить через трубку для охолодження процесора, перш ніж тепер тепла рідина охолоджується через радіатор і вентилятори. Перевага цієї системи полягає в значно зниженому рівні шуму, хоча вони можуть бути дорожчими, ніж традиційні системи охолодження ПК.

При виборі материнської плати, важливо звернути увагу на кількість USB-роз'ємів (включаючи USB-C) і на те, наскільки легко вона підключається до інших компонентів. БП можна буде визначити після того, як будуть підібрані усі основні компоненти, а корпус взяти розміру ATX, оскільки ми не розраховуємо на занадто габаритну робочу станцію.

Для тих, хто балансує між продуктивністю та ціною, AMD Ryzen 5 3600 — чудове місце для початку. Вентилятор, що входить у комплект, також сприяє охолодженню вашої системи під тиском. Для більш широких студій або тих, хто шукає максимальної швидкості, Intel i7-9700K — чудовий варіант. Він миттєво впорається з будь-якими задачами рендеру та змішування, та дозволить вам додавати плагіни та ефекти до душі, не турбуючись про втрату продуктивності.

Існує думка, що 32 ГБ оперативної пам'яті ніколи і нікому не знадобиться. Проте, беручи до уваги, що це доволі дешевий компонент, ніякої шкоди не буде в тому, щоб взяти собі таку кількість ОП. Можна спробувати будь-які інші марки-виробники ОП, одним із шанованих стандартів є Viper Steel Series 32 ГБ.

Якщо ви збираєтесь використовувати комбінацію SSD та HDD при збірці ПК, то можна зупинитись на комбінації Samsung 860 EVO 500 ГБ та будь-яким HDD на ваш вибір та розподілити файли на вашому ПК так, щоб ваші музичні проекти та DAW займали простір на SSD, а готові результати проектів та інше можуть зайняти місце на HDD. Якщо ви бажаєте досягти максимальної швидкості та продуктивності, розділ пам'яті, що повністю заповнює SSD, — саме для вас. Диск Crucial MX500 пропонує величезні 2 ТБ місця, що означає, що вам більше ніколи не доведеться покладатися на старі жорсткі диски.

Будучи центром вашої збірки, материнська плата є однією з областей, які потрібно вибрати правильно. На щастя, сучасні материнські плати досить прості для розуміння і пропонують усі необхідні підключення та сумісності. MSI B450 — це чудовий варіант для тих, хто дорожить витратами, і включає підключення Wi-Fi і Bluetooth, тому вам не знадобляться додаткові ключі чи адаптери. При збільшенні цінового діапазону можна покласти око на ASUS ROG Maximus XI може похвалитися додатковими можливостями підключення, що означає, що зовнішні пристрої даних або монітори можуть працювати якнайкраще.

Передбачається, що комп'ютер для виробництва музики буде також служити місцем для записування інструментів. Щоб ПК залишався максимально тихим, знадобиться система рідинного охолодження. Cooler Master LC240E — пропонується як один з бюджетних варіантів. Для збірок, які використовують більш потужні компоненти, є ймовірність того, що процесор буде працювати сильніше, і тому потрібна більша потужність охолодження. NZXT Kraken Z73 — це вибір вищого класу, який забезпечує кращу продуктивність охолодження та тихішу роботу при низьких навантаженнях.



### 3.2 Периферійні пристрої

Незважаючи на те, що вбудованих звукових компонентів материнської плати може вистачити для базових потреб при роботі із аудіо, завдяки чому інвестиції в звукову карту мають меншу віддачу, ніж будь-коли, жодне інше рішення для внутрішніх компонентів не перевершить наявність спеціальної карти для обробки вашого звуку, особливо під час керування високоякісним обладнанням, особливо коли точність і якість мають неабияке значення.

Якщо ви хочете створити серйозну систему без зовнішнього ЦАП, то така високопродуктивна/дорожча звукова карта може бути обов'язковою витратою. Asus випустила плату професійного рівня вищого класу – Asus Essence STX II, яка включає підсилювач для навушників 600 Ом, приголомшливо низьке (для внутрішнього компонента ПК!) співвідношення сигнал/шум 123 дБ, а також змінні гнізда підсилювача, щоб ви могли налаштувати картку самостійно, щоб краще відповідати вашим налаштуванням після його встановлення. Крім того, ви також отримуєте підтримку для всього, від Dolby Home Theater до високочутливих навушників IEM.



Рисунок 3.1 – Звукова карта ASUS Essence

Хоча внутрішня звукова карта є чудовим вибором з точки зору зручності, зменшення безладу кабелів і близькості до інших входних портів, якість будь-якого внутрішнього аудіопроцесора буде пом'якшена електричними перешкодами всередині корпусу вашого ПК. З цієї причини багато серйозних користувачів виробництва звуку звернулися до зовнішніх ЦАП і підсилювачів, щоб задовольнити свої потреби. В цій категорії вони не занадто відрізняються від звукових карт, проте такі девайси вважаються більш незалежними та якісними, щоб використовувати з ними мікрофони, динаміки та навушники. Для задоволення потреб у якості звуку можна звернути увагу на Creative Sound Blaster E1, яка є дешевим представником ЦАП, але її обмежені можливості цілком достатньо при роботі з аудіо. Вона має двоканальний інтерфейс, можливість підключення по Bluetooth, а також може використовуватись для прийому дзвінків, оскільки має вбудований мікрофон та динамік.



Рисунок 3.2 – Цифро-аналоговий перетворювач Creative Sound Blaster E1

Також необхідно приділити належну увагу вибору аудіоінтерфейсу. Особливо якщо ви збираєтесь використовувати ваш ПК для запису музики, голосу і так далі. Ціна такого девайсу знову варіюватиметься від ваших потреб та очікувань. Чим більше каналів для запису, входів для інструментів та мікрофонів,

тим більша і ціна. Якщо ви збираєтесь записувати до двох доріжок одночасно або ведете невелику особисту студію, зазвичай достатньо 2-4 вхідних каналів. Якщо ви збираєтесь записувати широкий діапазон інструментів або багато входів одночасно, вам слід розглянути більш потужний виділений аудіоінтерфейс, який встановлюється в стійку і має 8-24 або більше каналів введення-виведення або вхідних та вихідних каналів. Також важливо звернути увагу на наявність MIDI-інтерфейсу для підключення MIDI-клавіатури.

Якщо ви регулярно записуєте вокал, вам потрібно переконатися, то вам буде життєво-необхідний високоякісний інтерфейс без затримки. Затримка — це різниця між живим і відтвореним звуком. При виборі інтерфейсу для ПК середньої продуктивності, потрібно звернути увагу на моделі аудіоінтерфейсу із швидшим процесором, щоб він міг зняти додаткове навантаження з вашого ПК або MAC і забезпечити якість записаного та відтвореного звуку.

Загалом, можна виділити 2 види аудіо-інтерфейсів – десктопні та ті, що монтуються в стійку. Десктопні інтерфейси – це компактні бокси, які наповнені всім необхідним для домашнього запису або запису в невеликій студії з обмеженою кількістю інструментів. Ті, що монтуються в стійку, або rackmount-інтерфейси – це більш спеціалізоване обладнання, призначені для серйозних студій звукозапису, де одночасно можуть записуватись декілька інструментів, з великою кількістю входів, що вимагає набагато більше потужності обробки, і можуть записувати з більш високими налаштуваннями якості звуку або швидкістю передачі даних з більшим динамічним діапазоном.



Рисунок 3.3 – Rackmount-аудіоінтерфейс

Для наших цілей ми беремо простий десктопний аудіо-інтерфейс - Behringer Xenyx Q502 USB 5 Input USB Mixer Audio Interface. Це аудіо-інтерфейс, який також виконує роль мікшера.

Великою перевагою роботи з апаратним мікшером є негайне практичне керування. Потрібно збільшити канал у міксі для навушників? Просто простягніть руку та ввімкніть відповідний допоміжний елемент керування — не потрібно відкривати програмну панель керування. Хочете застосувати еквалайзер і стиснення, щоб відшліфувати свій звук на шляху? Знову ж таки, у вас є спеціальні, миттєво доступні елементи керування для них. Немає завантаження плагінів або доступу до окремого меню ефектів. Це робить мікшери з інтерфейсами особливо корисними також, якщо ви маєте відношення до подкастингу та стримінгу.

Тобто ми матимемо ширші можливості для роботи із аудіо, якщо виберемо аудіоінтерфейс в якості USB-мікшера.

Багато новачків у світі музичного виробництва або редагування аудіо, ймовірно, виберуть популярний USB-мікрофон, як-от Blue Yeti Snowball. Звісно, у цьому виборі немає нічого поганого, оскільки USB-мікрофон не може перевершити з точки зору зручності та простоти використання (зазвичай це пристрої plug-and-play). Проте з такими мікрофонами складніше працювати, особливо коли потрібно записувати декілька мікрофонів водночас і вони починають конфліктувати один з одним, оскільки ПК не може визначити, який мікрофон є головним. З іншої сторони, нам пропонується XLR-мікрофон, який використовується на професійних студіях і є довготривалим стандартом у сфері музики.



Рисунок 3.4 – Роз’єми XLR та USB

Серед основних переваг XLR варто виділити те, що він усуває фоновий шум на відміну від USB, що робить звук чистішим. Також XLR сумісний із будь-яким музичним обладнанням, включаючи аудіоінтерфейси, через які звук мікрофона буде записаний на комп'ютер. USB-мікрофони в цьому плані можуть похвалитись лише своєю мобільністю і тим що їх підключення до ПК або Mac не вимагає додаткових витрат на обладнання, оскільки вони підключаються напряму до USB роз'єму вашого комп'ютера.

Для більш поверхневого використання мікрофона ми можемо взяти самий простий XLR-мікрофон Shure SM48-LC. Цей динамічний мікрофон оснащений внутрішнім ударним картриджем, решіткою зі сталеві сітки та внутрішнім фільтром для пом'якшення звуків вітру та дихання. SM48 спроектований так, щоб працювати як на відкритому повітрі, так і в приміщенні, а також при використанні для співу чи мови. Мікрофон включає мікрофонний тримач і чохол на блискавці для перенесення.

Загалом матимемо наступну таблицю із компонентів, які будуть обрані до нашої студії.

Таблиця 3.1 – Збірка цифрової домашньої студії

Назва	Ціна
ПК на базі Intel i7-9700K з ОЗУ 32 ГБ, SSD 1ТБ та рідинним охолодженням Cooler Master LC240E	~900\$
Звукова карта ASUS Xonar Essence STX II	305\$
Цифро-аналоговий перетворювач Creative Sound Blaster E1	84\$
Behringer Xenyx Q502 USB 5 Input USB Mixer Audio Interface.	82\$

## Продовження таблиці 3.1

XLR-мікрофон Shure SM48-LC	80\$
Усього:	~1451\$

Наша збірка обійдеться користувачу у 1451\$, при умові, якщо будуть вибрані дешеві компоненти ПК на яких не акцентувалась увага (корпус, материнська плата, відеокарта). Альтернативно багато користувачів купують ноутбуки та ПК на базі Mac OS, що широко використовується при роботі із мультимедіа. Також потрібно урахувати, що за цю суму будуть доступні пристрої та обробки та запису аудіо, самі інструменти та MIDI-клавіатури невраховані в суму та остаточна ціна залежатиме від персональних смаків користувача.

## ВИСНОВКИ

У роботі були розглянуті минулі та сучасні методи звукозапису. Були описані різні програми для якісної роботи з аудіо та VST-плагінами, невід'ємна роль яких була продемонстрована у роботі. Було розглянуто популярний ефект "дисторшн".

Дисторшн — різновид звукового ефекту чи використовуваного його отримання приладу. Досягається *distortion* за рахунок «жорсткої» обрізки амплітуди аудіосигналу. Плагін якого був також відтворений на C++ за допомогою фреймворка JUCE. JUCE - це комплексна бібліотека класів C++ для розробки кроссплатформенного програмного забезпечення. Він містить практично все для створення більшості музичних додатків, побудови складних GUI, обробки графіки та аудіо.

Його робота та запис, який зроблено за допомогою плагіну буде продемонстровано у презентації.

Також була підібрана збірка компонентів типової студії в домашніх умовах для оптимальної роботи з аудіо.