

Деркач М.В., Срьоменко А.Ю.

РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ДЛЯ ОБРОБКИ ЦИФРОВОГО ЗВУКУ

У статті розглянуто актуальне питання переходу від аналогового представлення звуку до цифрового, що пов'язане з поширенням комп'ютерних технологій. На сьогодні, персональний комп'ютер зі спеціалізованим аналого-цифровим перетворювачем і програмним забезпеченням здатний виконувати функції студії звукозапису. Більшість програм відрізняються за можливостями і підходом до інтерфейсу користувача. На відміну від вбудованих засобів, програмні модулі, що розроблені в даному проекті, є більш універсальними, сумісними з усіма затребуваними в галузі звуковими робочими станціями. В процесі реалізації модулів було досліджено принципи роботи та елементи керування аналогових реалізацій звукового обладнання. Для реалізації модулів було обрано мову програмування C++, технологію Virtual Studio Technology 3 і платформу для розробки розширень iPlug 2. Платформа містить бібліотеку з основними засобами обробки звуку і створення графічного інтерфейсу плагінів для операційних систем Windows і macOS. Результатом роботи є програмні модулі, що здатні вирішувати основні задачі з обробки звуку: додавання звукових ефектів, таких як перевантаження та відлуння; еквалізацію, у вигляді 7-смугової системи SVF-фільтрів для зменшення шуму, корекції тембру і гучності звукових доріжок. Були створені графічні схеми цифрової обробки сигналів, спроектовано відповідно до реальних пристроїв графічний інтерфейс для програмних модулів. Реалізовані модулі здатні динамічно підключатись до програмного забезпечення для обробки звуку, що сумісне з форматом VST 3, зокрема сучасних цифрових звукових робочих станцій та аудіоредакторів і є сумісними з модулями інших форматів та від сторонніх постачальників. Реалізовані функції дозволяють організувати невелику студію звукозапису з доступним інтерфейсом на одному ПК.

Ключові слова: програмний модуль, звук, графічний інтерфейс, плагін, фільтр, еквалайзер, бібліотека.

Вступ. Звукове супроводження виконує важливі функції в галузях радіомовлення, телебачення, в музичній та кіноіндустрії [1,2]. Окрім запису, збереження та відтворення звуку, виникають задачі його синтезу та обробки, зокрема і в реальному часі – створення звукових ефектів, корекції тембру, позбавлення від шумів. Для вирішення цих задач існує широкий спектр аналогового звукового обладнання як професійного рівня, так і для домашнього застосування.

На сьогодні широкого розповсюдження досягли електронно-обчислювальні машини, зокрема персональні комп'ютери. Отже є актуальним питання застосування цифрових засобів для роботи зі звуком. При переході від аналогової до цифрової обробки виникає необхідність створення відповідного програмного забезпечення, що поєднає традиційні прийоми, особливості роботи зі звуком та звичний інтерфейс з перевагами сучасних цифрових технологій.

Сучасне програмне забезпечення надає широкі можливості з перетворення звуку, але вбудовані засоби звукових редакторів суттєво відрізняються за реалізацією як функцій, алгоритмів обробки, так і інтерфейсу користувача. Розробка незалежних програмних модулів має надати можливість ефективно перетворювати звукові сигнали з одним й тим самим результатом незалежно від програмного забезпечення робочої станції та рівня підготовки користувача.

Аналіз останніх досліджень і публікацій. Автор статті [3] пропонує фокусуватися на характеристиках звуку, і передбачає, що існує нерозривний зв'язок між вібруючим об'єктом і середовищем, в яке поширюється вібрація і знаходиться суб'єкт, що слухає. Наприклад, у кіновиробництві саундтрек допомагає глядачам оживити фільм, тобто для покращення враження слухача необхідно вирішити, який штучний звук передає суть звуку і дії, зображеного в сцені. У статті [4] автори представляють автоматизований інструмент глибокого навчання, який використовується для синтезу репрезентативною звуковою доріжкою для відео, який використовується для зв'язування аудіофайлів з беззвучним відео або для визначення критичних сценаріїв і забезпечення синтезованої, посиленої і синхронізованою за часом звуковою доріжкою. Для цього використано надійну багатомасштабну рекурентну нейронну мережу (RNN) і згорнуту нейронну мережу (CNN) для кращого розуміння складних асоціацій введення-виведення. Іншим прикладом можна вважати роботу [5], де автори зосереджені на застосуванні звукової обробки в художньому контексті, а саме для поліпшення візуальних вражень від поїздки на поїзді. У той час як ультразвукова обробка і слухові дисплеї зазвичай використовуються як засоби для представлення даних та інформування користувача, запропонований проект «Звук/треки» спрямований на поліпшення візуального сприйняття погляду з вікна потяга на проїжджаючий пейзаж шляхом додавання звукового вимірювання. Це дозволяє міркувати над візуальними враженнями і поглиблювати стан споглядання. Додаток можна запускати на мобільних телефонах з вбудованою камерою і на ноутбуках з вебкамою. У роботі пропонуються і обговорюються різні підходи до ультразвукової обробки і представлені різні сценарії застосування.

Мета статті. Розробити програмне забезпечення для обробки цифрового звуку в реальному часі у вигляді незалежних програмних модулів, що динамічно підключаються та є заміною аналогового звукового обладнання.

Основний зміст роботи. Серед проблем, що стоять в галузях звукозапису і звукорежисури є створення звукових ефектів, боротьба з шумом, корекція тембру і гучності звукових доріжок.

Одним з найпоширеніших звукових ефектів є перевантаження – додавання до звуку додаткових тонів, які виникають при обмеженні амплітуди звукової хвилі. Розроблений програмний модуль обробки звуку передбачає емуляцію звукового ефекту перевантаження (Distortion). Цей ефект з'являється при нелінійному спотворенні сигналу, тобто в результаті обробки виникають такі складові, що були відсутні у вхідному сигналі – гармоніки. Гармоніки є додатковими тонами, що за частотою завжди вище основного, утворюючи в поєднанні з ним натуральний звукоряд. Через це перевантаження широко застосовується в багатьох музичних жанрах, найчастіше в складі електрогітарних трактів з ламповими підсилювачами як окремий модуль, або вбудовують в підсилювачі як окремий звуковий канал. Основним засобом досягнення ефекту є кліппінг сигналу – обмеження амплітуди через відсікання форми хвилі.

Нижче наведено найпростішу реалізацію такої звукової обробки, що відсікає складові сигналу, які перевищують вказаний рівень (завдається змінною Threshold):

```
double Threshold;
for (int s = 0; s < nFrames; ++s, ++input, ++output)
{
    if(*input >= 0)
    {
        // Обмеження верхньої складової сигналу:
        *output = fmin(*input, Threshold);
    }
    else
    {
        // Обмеження нижньої складової сигналу:
        *output = fmax(*input, -Threshold);
    }
}
```

Даний алгоритм здійснює перетворення сигналу та вже входить до складу бібліотеки iPlug 2 як процедура clip(), але має такі недоліки як низька якість звуку на виході та маленький діапазон налаштування сили ефекту – результат з'являється лише при значному підвищенні гучності звуку.

В аналогових пристроях обробка здійснюється більш плавно, що можна побачити, проаналізувавши перевантажений сигнал з осцилографом. На рисунку 1 показано форми сигналів після так званого «м'якого» та «жорсткого» кліппінгу. Незалежно від сили прикладання ефекту, на відміну від простої цифрової реалізації кліппінгу, звукова хвиля має більш плавну форму.

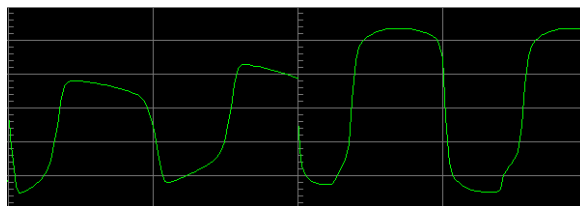


Рисунок 1 - Форма хвилі, обробленої аналоговим пристроєм

Застосування до кожного семплу нелінійної функції (1) в алгоритмі обробки сигналу дозволяє отримати подібне до аналогового обмеження амплітуди звукової хвилі, яке є більш плавним та впливає на силу ефекту і якість отриманого звуку.

$$f(x) = \operatorname{sgn}(x)(1 - e^{-|x|}), \quad (1)$$

де x – вхідний сигнал.

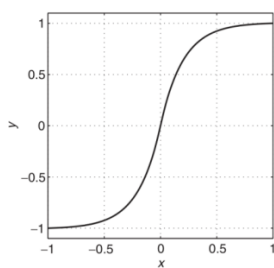


Рисунок 2 - Графік функції (згідно формули 1)

На рис. 2 зображено графік функції, з якого видно нелінійну зміну вхідного сигналу.

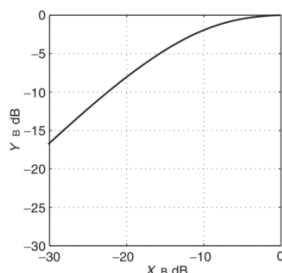


Рисунок 3 - Залежність амплітуд сигналів

Як видно з графіку функції, сигнал на виході буде перетворено в симетричне посилення в середині амплітуди, та плавним обмеженням при високих рівнях звуку. Рисунок 3 відображає залежність рівня вихідного сигналу від рівня вхідного сигналу в дБ.

Отриманий ефект робить звук більш гучнішим та жорстким, найчастіше його застосовують до звучання електрогітар. Наявно перевірити перетворення сигналу можна за допомогою осцилографів (рис. 4) на вході та виході модулю.



Рисунок 4 - Форма тестового сигналу на вході та виході плагіну

Форма хвилі тестового синусоїдального сигналу перетворюється на більш трапецієподібну, що характерно для перевантаженого звуку. На рис. 5 зображено слабкий сигнал електрогітари, отриманий з АЦП, і оброблений розробленим модулем сигнал, придатний для подальшої передачі до емулятору підсилювача.

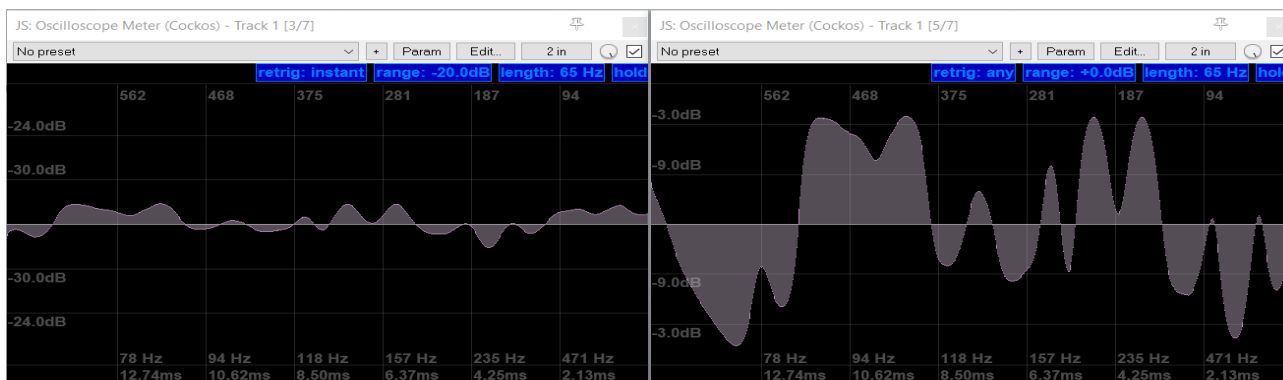


Рисунок 5 - Форма сигналу з входу АЦП і на виході плагіну

Графічний інтерфейс VST-плагінів традиційно створюється подібним до реальних аналогічних пристроїв. Інтерфейс плагіну спроектовано враховуючи типовий зовнішній вигляд аналогових блоків ефектів, що використовуються в галузі професійними звукорежисерами та музикантами.

Іншим затребуваним художнім ефектом є відлуння. Його основою є використання додаткового звукового буферу, в якому звуковий сигнал затримується на визначений час і додається до основного. В залежності від часу затримки і співвідношення сигналів, отриманий звук набуває об'ємності, додається відлуння звуку в просторі.

Лінія затримки в цифровій реалізації ефекту створюється за допомогою циклічного буферу – структури даних фіксованого розміру зі способом організації за принципом FIFO. Засобами мови програмування C++ цю структуру організовано як масив з двома вказівниками: на елемент, що зчитується, та на елемент, який необхідно перезаписати. Вказівники зсуваються при кожній ітерації, і при досягненні кінця структури повертаються на початок. Така структура дозволяє організувати потік даних, без необхідності переміщувати елементи в пам'яті.

Записавши до буферу необхідну кількість семплів, яка залежить від часу затримки та частоти дискретизації, поєднати чистий звуковий сигнал з накладеним ефектом можна за формулою (2).

$$y(n) = dry * x(n) + wet * [x(n - D) + fb * y(n - D)], \quad (2)$$

де $x(n)$ – семпл вхідного сигналу; dry – рівень основного сигналу; wet – рівень затриманого сигналу; D – кількість семплів, що залежить від часу затримки; fb – рівень зворотного зв'язку.

На основі змішування чистого сигналу з лінією затримки також створюються інші звукові ефекти:

- фленжер – змішування оригінального сигналу з затриманим від 5 до 15 мс. При переході звуку з одного каналу до іншого, створюється ефект, схожий на звук реактивного літака;
- хорус – змішування оригінального сигналу з затриманим до 5 мс. Дозволяє отримати ефект «хорового» звучання;
- фейзер – додавання до оригінального сигналу його затриманої копії зі зміненою фільтром фазою;
- реверберація – відбиття звуку в закритих приміщеннях для імітування об'ємного звучання. На відміну від ефекту луни з затримкою від 50 мс, цей тип ефекту відтворює більшу кількість сигналів з меншою змінною затримкою (від 0.01 до 1 мс).

Роботу плагіну продемонстровано двома осцилографами на вході та виході модулю (рис. 6).

При заданих параметрах 42 мс і 100% рівні затримки, отриманий на виході сигнал зсунуто відносно «чистого» на 42 мс. Різниця в амплітудах сигналів виникає через вплив зворотного зв'язку.

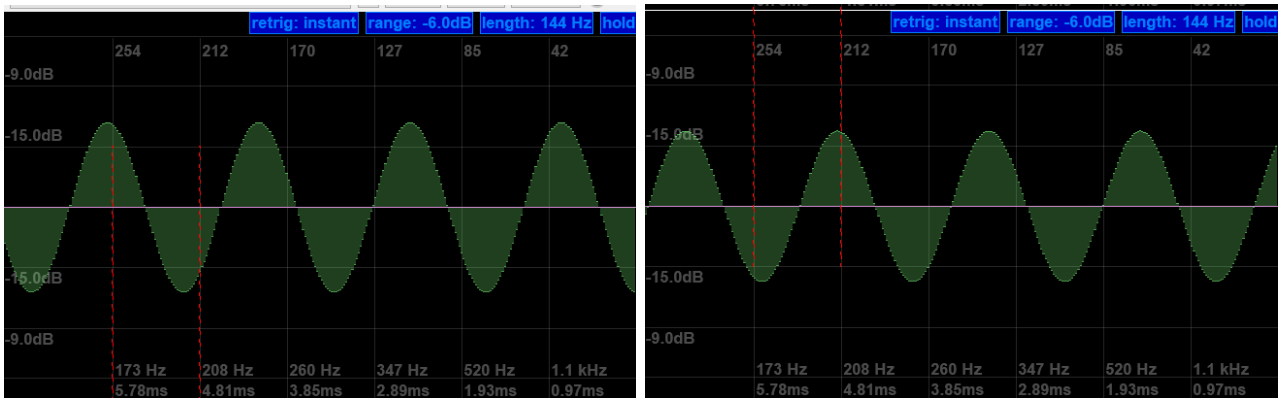


Рисунок 6 - Сигнал на вході та виході плагіну

Для корекції тембру необхідно створити систему електронних фільтрів – еквайзер. Одним з них є фільтр змінної стану, який поєднує в собі фільтри діапазонів низьких, середніх та високих частот. Він отримав широке поширення в аналоговому обладнанні, але має і програмну реалізацію.

Програмну реалізацію фільтру можна описати системою рівнянь (3).

$$\begin{aligned} y_{\text{ФНЧ}}(n) &= F_1 y_{\text{СПФ}}(n) + y_{\text{ФНЧ}}(n - 1) \\ y_{\text{СПФ}}(n) &= F_1 y_{\text{ФВЧ}}(n) + y_{\text{СПФ}}(n - 1) \\ y_{\text{ФВЧ}}(n) &= x(n) - y_{\text{ФНЧ}}(n - 1) - 1/Q y_{\text{ФНЧ}}(n - 1), \end{aligned} \quad (3)$$

де $y_{\text{ФНЧ}}(n)$ – вихід фільтру низьких частот; $y_{\text{СПФ}}(n)$ – вихід смугово-проникного фільтру; $y_{\text{ФВЧ}}(n)$ – вихід фільтру високих частот; Q – добротність; F_1 – коефіцієнт підстроювання (4)

$$F_1 = 2 \sin(\pi f_c / f_{\text{дискретизації}}) \quad (4)$$

Перевагами цього фільтру є те, що основні параметри фільтру (чутливість, добротність та центральна частота) можуть бути задані незалежно один від одного. SVF отримав широке поширення в аналоговій аудіоапаратурі, зокрема еквайзерах та синтезаторах, а програмна реалізація цього фільтру входить до складу бібліотеки iPlug 2.

Графічний еквайзер реалізовано як 7-смугову систему SVF-фільтрів. В скинутому до нульового значення стані повзунків, фільтри не змінюють амплітуду сигналу. Для прикладу, повзунки графічного еквайзера розташовано в порядку збільшення частот, і їх положення відтворює форму лінії показань на екрані аналізатору спектру частот (рис. 7). Діапазон частот, на які впливають фільтри, підібрано з урахуванням частот, на яких можна змінити тембр звучання музичних інструментів та голосу. Послаблення частот надає можливість використовувати модуль як фільтр шуму.

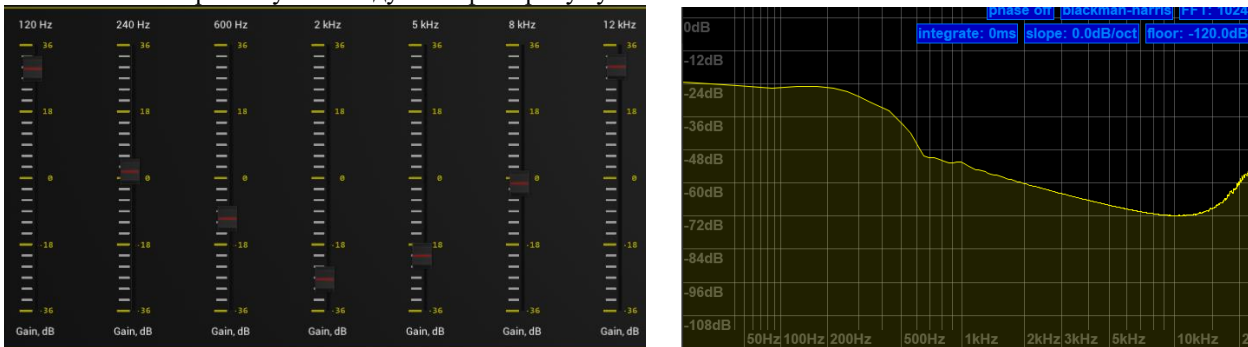


Рисунок 7 - Відповідність налаштувань еквайзера графіку АЧХ

Пресети до еквайзера створено з урахуванням типового застосування пристроїв при роботі зі звуком – змінення тембру: для посилення низьких частот, посилення високих частот або послаблення середніх частот. Такі режими роботи найчастіше використовують при технічній еквалізації окремих записаних звукових доріжок. Параметри передумовок наведено в таблиці 1.

Таблиця 1 Параметри пресетів еквайзера

Пресет	Підсилення частот, дБ						
	120 Гц	240 Гц	600 Гц	2 кГц	5 кГц	8 кГц	12 кГц
Скинутий стан	0	0	0	0	0	0	0
V-Shape	+9	+6,3	+1,8	-2	+0,7	+2,6	+4
Bass boost	+6,5	+7	+3,3	-1,9	-1,7	-2	-2
Treble boost	-1	-1,5	-1,5	-1,3	+2,3	+5,2	+5,5

Звукові частотні фільтри з бібліотеки iPlug 2 реалізовано як об'єкти класу SVF. Програмний код налаштування фільтрів і обробки сигналу їх системою:

```
SVF<sample, 2> eqLPShelf = SVF<sample, 2>::EMode::kLowPassShelf;
eqLPShelf.SetFreqCPS(120.);
eqLPShelf.SetGain(LowShelfGain);
SVF<sample, 2> eqMid = SVF<sample, 2>::EMode::kBell;
eqMid.SetFreqCPS(600.);
eqMid.SetGain(MidGain);
SVF<sample, 2> eqHPShelf = SVF<sample, 2>::EMode::kHighPassShelf;
eqHPShelf.SetFreqCPS(12000.);
eqHPShelf.SetGain(HighShelfGain);
if (pswitch)
{
    eqLPShelf.ProcessBlock(inputs, lshelfpassed, 2, nFrames);
    eqMid.ProcessBlock(lpassed, midpassed, 2, nFrames);
    eqHPShelf.ProcessBlock(midpassed, outputs, 2, nFrames);
}
```

Висновок. В результаті було розроблено програмних модулів, що здатні вирішувати основні задачі з обробки звуку. Програмний модуль для реалізації ефекту перевантаження звуку виконує емуляцію аналогового пристрою – блоку ефектів. Найчастіше такі блоки застосовують в звукових трактах перед входом гітарних підсилювачів, тому для сумісності з гітарним обладнанням, плагін окрім кліппінгу має виконувати підсилення рівню сигналу. В алгоритмі перетворення звуку враховується два параметри – гучність вихідного сигналу

(volume) та сила прикладання ефекту (drive). В залежності від комутації програми-хосту, плагін одночасно може здійснювати обробку декількох звукових каналів.

Реалізований модуль шумового затвору має два параметри з відповідними елементами керування:

- Threshold (порог відкриття);
- Hold (час утримання затвору).

Кнопка в нижній частині переводить плагін в режим пропуску. Порог встановлюється в залежності від рівня шуму у вхідному сигналі. Для запобігання хибного закриття та більш плавної роботи без втрат звуку, затвор утримується в відкритому стані на деякий час, що задається параметром Hold від 1 мс до 1 с.

Програмний модуль ефекту луни (Delay) має 3 регулятори, що відповідають за наступні параметри:

- Delay (час затримки) – регулює час, на який буде затримано сигнал від 0 до 500 мс;
- Feedback (рівень зворотного зв'язку) – регулює рівень впливу зворотного зв'язку на вихідний сигнал від 0 до 100%;
- Level (рівень ефекту) – відповідає за співвідношення «чистого» сигналу до затриманого. При 0% ефект відсутній, при 100% – чути лише затриманий звук.

Інтерфейс модулю еквалайзера спроектовано на основі апаратних еквалайзерів. Він складається з 7 повзунків, що відповідають за посилення та послаблення відповідних частот від +36 дБ до -36 дБ, кнопки та індикатору живлення.

Література

1. Yeoh J. Sounds Like Chicken: Sensory Marketing and Sound Effects / J. Yeoh, D. Allan // Indian Journal of Marketing 50(8-9). – 2020. – pp. 19–31.
2. Cardany A. A Coincidental Sound Track for Time Flies / A. Cardany // General Music Today 28(1). – 2014. – pp. 23–26.
3. Solomos M. From Sound to Sound Space, Sound Environment, Soundscape, Sound Milieu or Ambiance ... / M. Solomos // Paragraph. – 2018. – Vol. 41, Is. 1. – pp. 95–109.
4. Ghose S. AutoFoley: Artificial Synthesis of Synchronized Sound Tracks for Silent Videos With Deep Learning / S. Ghose, J. J. Prevost // IEEE Transactions on Multimedia. – 2021. – Vol. 23. – pp. 1895-1907.
5. Knees P. Sound/tracks: artistic real-time sonification of train journeys / P. Knees, T. Pohle, G. Widmer // Journal on Multimodal User Interfaces. – 2012. – Vol. 6(1-2). – pp. 87–93.

References

1. Yeoh J. Sounds Like Chicken: Sensory Marketing and Sound Effects / J. Yeoh, D. Allan // Indian Journal of Marketing 50(8-9). – 2020. – pp. 19–31.
2. Cardany A. A Coincidental Sound Track for Time Flies / A. Cardany // General Music Today 28(1). – 2014. – pp. 23–26.
3. Solomos M. From Sound to Sound Space, Sound Environment, Soundscape, Sound Milieu or Ambiance ... / M. Solomos // Paragraph. – 2018. – Vol. 41, Is. 1. – pp. 95–109.
4. Ghose S. AutoFoley: Artificial Synthesis of Synchronized Sound Tracks for Silent Videos With Deep Learning / S. Ghose, J. J. Prevost // IEEE Transactions on Multimedia. – 2021. – Vol. 23. – pp. 1895-1907.
5. Knees P. Sound/tracks: artistic real-time sonification of train journeys / P. Knees, T. Pohle, G. Widmer // Journal on Multimodal User Interfaces. – 2012. – Vol. 6(1-2). – pp. 87–93.

The article considered the actual issue of the transition from analogue to digital representation of sound. This is due to the proliferation of computer technology. Today, a personal computer with a specialized analog-to-digital converter and software is capable of performing the functions of a recording studio. Most programs differ in their capabilities and approach to the user interface. Unlike built-in tools, the software modules developed in this project are more universal, compatible with all sound workstations in demand in the field. In the process of implementing the modules, the principles of operation and control elements of analog implementations of audio equipment were investigated. To implement the modules, the programming language C ++, Virtual Studio Technology 3 and the platform for developing iPlug 2 extensions were chosen. The platform contains a library with the main tools for sound processing and creating a graphical interface for plug-ins for Windows and macOS operating systems. The result of the work is software modules that can solve the main tasks of sound processing: adding sound effects, such as overload and echo; equalization, in the form of a 7-band SVF filter system for noise reduction, tone correction and volume of soundtracks. Graphic circuits for digital signal processing were created, and a graphical interface for software modules was designed in accordance with real devices. The implemented modules are able to dynamically connect to audio processing software, are compatible with the VST 3 format, in particular with modern digital sound workstations and audio editors, and are also compatible with modules of other formats and from third-party suppliers. The implemented functions allow organizing a small recording studio with an accessible interface on one PC.

Keywords: software module, sound, graphical interface, plugin, filter, equalizer, library.

В статтєе рассмотрен актуальный вопрос перехода от аналогового представления звука к цифровому, что связано с распространением компьютерных технологий. На сегодня, персональный компьютер со

специализированным аналого-цифровым преобразователем и программным обеспечением способен выполнять функции студии звукозаписи. Большинство программ отличаются по возможностям и подходом к интерфейсу пользователя. В отличие от встроенных средств, программные модули, разработанные в данном проекте, являются более универсальными, совместимыми со всеми востребованными в области звуковыми рабочими станциями. В процессе реализации модулей были исследованы принципы работы и элементы управления аналоговых реализаций звукового оборудования. Для реализации модулей был выбран язык программирования C ++, технологию Virtual Studio Technology 3 и платформу для разработки расширений iPlug 2. Платформа содержит библиотеку с основными средствами обработки звука и создания графического интерфейса плагинов для операционных систем Windows и macOS. Результатом работы является программные модули, способные решать основные задачи по обработке звука: добавление звуковых эффектов, такие как перегрузка и эхо; эквализацию, в виде 7-полосной системы SVF-фильтров для уменьшения шума, коррекции тембра и громкости звуковых дорожек. Были созданы графические схемы цифровой обработки сигналов, спроектировано в соответствии с реальными устройствами графический интерфейс для программных модулей. Реализованные модули способны динамически подключаться к программному обеспечению для обработки звука, совместимы с форматом VST 3, в частности с современными цифровыми звуковыми рабочими станциями и аудиоредакторами, а также совместимы с модулями других форматов и от сторонних поставщиков. Реализованные функции позволяют организовать небольшую студию звукозаписи с доступным интерфейсом на одном ПК.

Ключевые слова: программный модуль, звук, графический интерфейс, плагин, фильтр, эквалайзер, библиотека.

Деркач М.В. – к.т.н., доцент кафедры комп'ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: glN459@gmail.com

Єрьоменко А.Ю. – здобувач вищої освіти Східноукраїнського національного університету імені Володимира Даля, e-mail: andrey.eriomenko@gmail.com