Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту
Завідувач кафедри комп’ютерних наук,
доктор технічних наук, професор
_____ Андрій БОНДАРЧУК
«___»_____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»
Спеціальність 122 Комп’ютерні науки
Освітня програма 122.00.01 Інформатика

Тема: ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС РАДІОМОВЛЕННЯ НА RASPBERRY PI

Виконав
студент групи ІНб-2-22-4.0д
Бойко Тарас Мирославович
_____

Науковий керівник
кандидат технічних наук, доцент
Яскевич Владислав Олександрович
_____

Київ – 2026

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»
Завідувач кафедри комп’ютерних наук,
доктор технічних наук, професор
_____ Андрій БОНДАРЧУК
31 жовтня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
«Програмно-апаратний комплекс радіомовлення на raspberry pi»

Виконавець – студент групи ІНб-2-22-4.0д,
спеціальності 122 Комп'ютерні науки,
освітньої програми 122.00.01 Інформатика
Бойко Тарас Мирославович

1. Вихідні дані: Законодавство України у сфері медіа та авторського права. Офіційна документація платформи Raspberry Pi, Raspberry Pi OS, Qt, Python, SQLite, Icecast, MPD та Liquidsoap. Наукові публікації з цифрового радіомовлення, потокової передачі аудіо та edge computing.

2. Основні завдання: Проаналізувати технології та архітектурні рішення систем цифрового радіомовлення. Обґрунтувати використання платформи Raspberry Pi та програмних засобів для реалізації комплексу. Сформувати вимоги до системи та спроєктувати її архітектуру. Розробити діаграми компонентів, розгортання та ER-діаграму бази даних. Реалізувати модулі медіаплеєра, синхронізації, моніторингу, дистанційного управління та збору статистики. Провести тестування комплексу й оформити роботу відповідно до вимог.

3. Пояснювальна записка: Обсяг близько 38 сторінок формату А4 комп’ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.

4. Графічні матеріали: Комп’ютерна презентація доповіді. Діаграми розгортання, компонентів, ER-модель бази даних, схема архітектури комплексу, таблиці порівняння та результатів тестування

5. Додатки: Інтерфейс плеєра BMP, структура плейлиста, фрагменти програмного коду, конфігураційні файли та ілюстративні матеріали.

6. Строк подання роботи на кафедру: 1 червня 2026 р.

Науковий керівник:	Виконавець:
к.п.н., доцент	
Яскевич Владислав Олександрович	Бойко Тарас Мирославович
31 жовтня 2025 р.	31 жовтня 2025 р.

Календарний план

№	Етапи виконання роботи	Термін виконання	Примітка
1	Затвердження теми кваліфікаційної роботи бакалавра	31.10.25	Виконано
2	Аналіз проблеми, вивчення аналогів, формування цілей і завдань роботи	01.11.25 – 11.01.26	Виконано
3	Аналіз технологій цифрового радіомовлення, форматів, протоколів передачі та алгоритмів кодування аудіоданих	26.01.26 – 15.03.26	Виконано
4	Дослідження архітектурних рішень, вибір платформи Raspberry Pi та програмних засобів реалізації комплексу	16.03.26 – 31.03.26	Виконано
5	Розробка архітектури комплексу, UML-діаграм, ER-моделі бази даних і структури програмних модулів	01.04.26 – 15.04.26	Виконано
6	Тестування модулів медіаплеєра, синхронізації, моніторингу та збору статистики відтворення	16.04.26 – 30.04.26	Виконано
7	Впровадження системи дистанційного управління, передавання статистики та перевірка роботи комплексу	01.05.26 – 15.05.26	Виконано
8	Підготовка пояснювальної записки, графічних матеріалів, додатків і проходження перевірки	25.05.26 – 12.06.26	Виконано
9	Захист кваліфікаційної роботи	19.06.26	

Науковий керівник:
к.п.н., доцент
Яскевич Владислав Олександрович
31 жовтня 2025 р.

Виконавець:

Бойко Тарас Мирославович
31 жовтня 2025 р.

РЕФЕРАТ

Кваліфікаційна робота: 38 сторінок, 27 джерел, додаток.

Актуальність теми: зумовлена стрімкою цифровізацією засобів масової комунікації, переходом від традиційних форматів мовлення до мережевих мультимедійних платформ, потребою в автоматизованому обліку використання музичного контенту для нарахування роялті, а також необхідністю інтеграції службових повідомлень, зокрема сигналів повітряної тривоги та хвилини мовчання, у структуру цифрового радіомовлення.

Об'єкт дослідження: інформаційні процеси обробки, управління та автоматизованого відтворення аудіоданих у програмно-апаратних системах мовлення.

Предмет дослідження: програмно-апаратні засоби та методи побудови комплексу радіомовлення на Raspberry Pi з використанням технологій локальної синхронізації медіаконтенту, планового відтворення та збору статистики.

Мета роботи: розроблення та дослідження програмно-апаратного комплексу радіомовлення на базі Raspberry Pi, який забезпечує стабільне та безперервне відтворення аудіоконтенту за локальними плейлистами, можливість централізованого керування контентом та автоматизований облік використання музичних творів.

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

1. Проаналізувати сучасні технології цифрового радіомовлення та існуючі рішення на базі одноплатних комп'ютерів.
2. Спроєктувати архітектуру програмно-апаратного комплексу радіомовлення на Raspberry Pi з урахуванням функціональних і правових вимог.
3. Програмно реалізувати модулі медіаплеєра, синхронізації контенту, моніторингу та підсистему збору статистики відтворення.
4. Провести тестування та оцінку ефективності функціонування розробленого комплексу.

Основні результати: у роботі розроблено програмно-апаратний комплекс автоматизованого управління радіомовленням ВМР на базі Raspberry Pi для мережі торгово-розважальних об’єктів бренду Firebrands. Сформовано функціональні та нефункціональні вимоги до системи, розроблено UML-діаграми компонентів і розгортання, ER-діаграму бази даних із шістьма сутностями. Реалізовано C++/Qt-застосунок-плеєр із бібліотекою libmpg123, Python-модулі синхронізації контенту та моніторингу, Bash-скрипти для передавання статистики на CRM-сервер, а також підсистему дистанційного управління на базі TeamViewer. За результатами тестування середнє завантаження ЦП при відтворенні MP3 склало 7,3 %, час реакції на сигнал пріоритетного переривання – 1,8 с, доступність системи за 30-добовий тест – 99,7 %, точність збору статистики – 100 %. Усі функціональні та нефункціональні вимоги виконані в повному обсязі.

Практичне значення дослідження полягає у можливості впровадження розробленого програмно-апаратного комплексу радіомовлення на базі Raspberry Pi для стабільного автоматизованого відтворення аудіоконтенту, централізованого керування плейлистами, дистанційного моніторингу пристроїв і збору статистики відтворення музичних творів.

Ключові слова: цифрове радіомовлення, Raspberry Pi, BMP, локальна синхронізація контенту, автоматизоване відтворення аудіоданих, збір статистики, Edge Computing, медіаплеєр, MP3, дистанційний моніторинг.

Зміст

ВСТУП 3

1 АНАЛІЗ ТЕХНОЛОГІЙ ТА АРХІТЕКТУРНИХ РІШЕНЬ СИСТЕМ ЦИФРОВОГО РАДІОМОВЛЕННЯ 7

1.1 Інформаційні технології та архітектура систем цифрового радіомовлення 7

1.2 Аналіз форматів, протоколів передачі та алгоритмів кодування аудіоданих 9

1.3 Методи та технології надійного відтворення мультимедійних даних в умовах обмежених ресурсів 11

1.4 Огляд та порівняльний аналіз існуючих програмних рішень для організації автоматизованого медіамовлення 13

Висновки до розділу 1 15

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ РАДІОМОВЛЕННЯМ 16

2.1 Постановка задачі та формування функціональних і нефункціональних вимог до програмної системи 16

2.2 Проєктування архітектури програмно-апаратного комплексу 18

2.3 Розробка інформаційної моделі та проєктування бази даних для збору статистики відтворення 22

Висновки до розділу 2 24

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ КОМПЛЕКСУ УПРАВЛІННЯ РАДІОМОВЛЕННЯМ 25

3.1 Програмна реалізація плеєра, модулів синхронізації контенту та автоматизованого збору статистики 25

3.2 Розробка та налаштування системи моніторингу та дистанційного управління 27

3.3 Тестування системи та оцінка ефективності функціонування комплексу 29

Висновки до розділу 3 31

ВИСНОВКИ 33

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 35

ДОДАТКИ 38

ВСТУП

Сучасний етап розвитку інформаційного суспільства характеризується стрімкою цифровізацією засобів масової комунікації та переходом від традиційних форматів мовлення до мережевих мультимедійних платформ. Радіомовлення, яке історично ґрунтувалося на аналогових технологіях FM-діапазону, у XXI столітті активно трансформується в напрямі цифрових та IP-орієнтованих рішень. Інтернет-радіо, потокова передача аудіо (audio streaming), використання серверів мовлення та хмарної інфраструктури стали стандартом для багатьох комерційних і некомерційних мовників.

Важливою складовою цього процесу є використання відкритих апаратних і програмних платформ, які дають можливість створювати функціональні мультимедійні системи з мінімальними витратами. Одноплатні комп'ютери, зокрема Raspberry Pi, зарекомендували себе як універсальна апаратна основа для побудови інформаційних і мережевих систем різного призначення [1], [4], [5]. Дослідження показують, що Raspberry Pi є технологічною платформою, яка сприяє розвитку інноваційних рішень і забезпечує достатній рівень продуктивності для реалізації мультимедійних сервісів за умови раціонального використання ресурсів [1].

Огляд широкомасштабних застосувань Raspberry Pi підтверджує його придатність до безперервної роботи у складі розподілених систем, а також ефективність з точки зору енергоефективності та вартості [4]. Дослідження, присвячені потоковій передачі мультимедійних даних із використанням Raspberry Pi, демонструють можливість забезпечення прийнятного рівня затримки та стабільності передавання сигналу навіть в умовах обмежених апаратних ресурсів [3]. Крім того, відкриті електронні платформи розглядаються як важливий елемент сучасної цифрової інфраструктури та розвитку сервісів у сфері медіа і смарт-систем [5].

В Україні цифрова трансформація медіасфери відбувається в межах оновленого правового поля. Закон України «Про медіа» визначає правові засади

функціонування аудіовізуальних і онлайн-медіа, регулює діяльність мовників та встановлює вимоги до прозорості й відповідальності у сфері поширення інформації [25]. Одночасно Закон України «Про авторське право і суміжні права» регламентує порядок використання музичних творів, зокрема у випадках їх публічного відтворення та трансляції в мережі Інтернет [26]. Таким чином, сучасні системи радіомовлення повинні забезпечувати не лише технічну стабільність трансляції, але й механізми коректного обліку використання музичного контенту для подальшого нарахування та виплати роялті правовласникам.

Особливої актуальності проблема цифрового радіомовлення набуває в умовах воєнного стану. Відповідно до положень Кодексу цивільного захисту України, своєчасне інформування населення про загрози є складовою системи державної безпеки [27]. Це зумовлює потребу в інтеграції механізмів автоматичного включення службових повідомлень, зокрема сигналів повітряної тривоги або хвилини мовчання, у структуру цифрових радіотрансляцій. Поєднання функцій комерційного мовлення з можливістю оперативного поширення суспільно важливої інформації підвищує соціальну значущість відповідних технічних рішень.

Попри наявність окремих прикладних реалізацій цифрового радіомовлення на базі Raspberry Pi, відсутній комплексний підхід до створення програмно-апаратної системи, яка поєднує такі функціональні можливості:

1. централізоване та дистанційне керування мовленням;
2. індивідуальне формування плейлистів під конкретного користувача або клієнта;
3. автоматичну інтеграцію рекламних і службових аудіороликів;
4. збір статистики відтворення музичних творів;
5. забезпечення легалізації трансляції шляхом коректного обліку авторських прав.

Інноваційність запропонованого підходу полягає у застосуванні моделі локального відтворення контенту: медіаплеєр Brand Media Player (BMP),

реалізований на мові C++ із використанням бібліотек Qt, щоночі автоматично синхронізує текстові плейлисти та MP3-файли з централізованого файлового сервера. Завдяки цьому відтворення відбувається локально протягом усього дня незалежно від стану мережевого з'єднання. Допоміжні модулі на Python та Shell забезпечують стабільність роботи плеєра, моніторинг його стану та резервне відтворення у разі недоступності сервера. Система підтримує дистанційне налаштування параметрів мовлення, автоматичне вбудовування рекламних блоків і пріоритетних службових повідомлень у загальний потік відтворення [18].

Ключовою ідеєю розроблюваного комплексу є забезпечення прозорого та автоматизованого збору статистичних даних щодо відтворення музичних творів. Це створює технічні передумови для коректного нарахування роялті та сприяє легалізації музичного мовлення. Такий підхід відповідає сучасним вимогам правового регулювання та підтримує принцип справедливої винагороди авторів і виконавців.

Практична значущість роботи полягає у можливості використання розробленого програмно-апаратного комплексу в закладах освіти, торговельних і сервісних підприємствах, офісних приміщеннях, громадських просторах та малих медіапроєктах. Використання недорогого апаратного забезпечення та відкритого програмного забезпечення дозволяє суттєво зменшити витрати порівняно з комерційними радіомовними системами. Модульна архітектура забезпечує масштабованість і гнучкість налаштування відповідно до потреб користувача.

Об’єктом дослідження є інформаційні процеси обробки, управління та автоматизованого відтворення аудіоданих у програмно-апаратних системах мовлення.

Предметом дослідження є програмно-апаратні засоби та методи побудови комплексу радіомовлення на Raspberry Pi з використанням технологій локальної синхронізації медіаконтенту, планового відтворення та збору статистики.

Метою роботи є розроблення та дослідження програмно-апаратного комплексу радіомовлення на базі Raspberry Pi, який забезпечує стабільне та безперервне відтворення аудіоконтенту за локальними плейлистами, можливість централізованого керування контентом та автоматизований облік використання музичних творів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати сучасні технології цифрового радіомовлення та існуючі рішення на базі одноплатних комп’ютерів.
2. Спроєктувати архітектуру програмно-апаратного комплексу радіомовлення на Raspberry Pi з урахуванням функціональних і правових вимог.
3. Програмно реалізувати модулі медіаплеєра, синхронізації контенту, моніторингу та підсистему збору статистики відтворення.
4. Провести тестування та оцінку ефективності функціонування розробленого комплексу.

Методи дослідження. Для вирішення поставлених завдань використовувались: системний аналіз – для дослідження існуючих рішень у сфері цифрового радіомовлення; порівняльний аналіз – для вибору технологій та архітектурних підходів; об'єктно-орієнтоване проєктування та UML-моделювання – для розробки архітектури системи; метод прецедентів – для формування функціональних вимог; експериментальне дослідження – для оцінки технічних характеристик розробленого комплексу.

Результати роботи апробовано шляхом публікації тез доповіді на конференції Інформаційні технології – 2026: зб. тез XII Всеукраїнської науково-практичної конференції молодих учених, 14 трав. 2025 р., м. Київ / Київський столичний університет імені Бориса Грінченка. с. 136–141 [13].

РОЗДІЛ 1

АНАЛІЗ ТЕХНОЛОГІЙ ТА АРХІТЕКТУРНИХ РІШЕНЬ СИСТЕМ ЦИФРОВОГО РАДІОМОВЛЕННЯ

1.1 Інформаційні технології та архітектура систем цифрового радіомовлення

Цифрове радіомовлення є одним із ключових напрямів сучасної мультимедійної індустрії, що забезпечує масовий доступ аудиторії до звукового контенту через телекомунікаційні мережі. Порівняно з аналоговим FM-мовленням, цифрові системи характеризуються вищою якістю відтворення звуку, підтримують ефективне стиснення аудіоданих і забезпечують можливість дистанційного управління та моніторингу мовлення [17]. Перехід від аналогового до цифрового мовлення відбувається поетапно, охоплюючи як технологічну інфраструктуру, так і нормативно-правову базу.

Еволюцію систем радіомовлення доцільно розглядати у трьох основних етапах. Перший (1990–2005 рр.) пов'язаний з появою стандартів цифрового ефірного мовлення DAB (Digital Audio Broadcasting) та DAB+, що здійснюють передачу стиснутих аудіоданих у частотних діапазонах, традиційно відведених для FM-мовлення. Другий (2005–2015 рр.) ознаменувався масовим розгортанням IP-радіомовлення: звуковий контент почав передаватись через мережу Інтернет за протоколами HTTP та RTP із використанням форматів типу SHOUTcast. Третій, поточний етап (з 2015 р.) характеризується переходом до гібридних і децентралізованих моделей, де методи потокової передачі поєднуються з локальним збереженням контенту задля вищої відмовостійкості [1].

З архітектурної точки зору системи цифрового радіомовлення поділяються на три класи залежно від моделі доставки контенту.

Потокове (streaming) мовлення – клієнтський пристрій підключається до медіасервера і отримує безперервний аудіопотік у реальному часі. Перевагою є миттєве оновлення контенту та централізоване управління. Недолік – жорстка

залежність від стабільності мережевого з'єднання: будь-яке переривання каналу безпосередньо впливає на якість мовлення.

Завантажувальне (download-and-play) мовлення – пристрій попередньо завантажує аудіофайли на локальне сховище і відтворює їх автономно. Забезпечує максимальну відмовостійкість, проте обмежує оперативність оновлення контенту та вимагає достатнього обсягу пам'яті.

Гібридна модель поєднує переваги обох підходів: контент завантажується у заплановані проміжки часу (о 00:00), а вдень відтворюється локально з можливістю термінового оновлення за потреби. Такий підхід є доцільним для систем, які мають працювати безперервно, але не можуть повністю залежати від якості мережевого з’єднання. Він дозволяє зберегти централізоване управління контентом і водночас забезпечити автономність кінцевого пристрою.

Система Brand Media Player (BMP) реалізує саме гібридну модель із пріоритетом локального відтворення, що гарантує безперервність мовлення навіть за тривалої відсутності мережевого з'єднання.

Одноплатні комп'ютери набули визнання як ефективна апаратна основа для вбудованих мультимедійних систем. Дослідження Johnston та Cox [1] описують Raspberry Pi як «технологічного дисруптора», що демократизував доступ до обчислювальних ресурсів, поєднавши достатню продуктивність із надзвичайно низькою вартістю. Платформа підтримує Raspberry Pi OS на базі Linux, що забезпечує повноцінне програмне середовище для розробки складних медіааплікацій [18].

Масштабне розгортання Raspberry Pi у розподілених системах досліджено у роботі Jolles et al. [4], де підтверджено прийнятний рівень надійності у промислових умовах. Costa та Duran-Faundez [5] вказують на переваги відкритих апаратних платформ для реалізації IoT-шлюзів і медіапристроїв. Calvo et al. [2] досліджували Raspberry Pi як шлюзовий вузол у розподілених системах збору даних, що підтверджує його придатність до виконання функцій медіавузла.

Порівняльну характеристику моделей Raspberry Pi щодо придатності для задач медіамовлення наведено в таблиці 1.1.

Таблиця 1.1

Порівняльна характеристика моделей Raspberry Pi для медіазастосувань

Модель	ЦП	RAM	USB	Аудіовихід	Придатність для ВМР
RPi 3B	4xA53 1.2 ГГц	1 ГБ	4xUSB 2.0	3.5 мм jack	Висока
RPi 3B+	4xA53 1.4 ГГц	1 ГБ	4xUSB 2.0	3.5 мм jack	Висока
RPi 4B	4xA72 1.5 ГГц	2-8 ГБ	2xUSB 3.0	3.5 мм jack	Дуже висока
RPi Zero 2W	4xA53 1.0 ГГц	512 МБ	micro-USB	Відсутній	Низька
RPi Zero W	1xA7 1.0 ГГц	512 МБ	micro-USB	Відсутній	Дуже низька

Примітка. Оцінка придатності враховує: обчислювальну потужність, підтримку 24/7, наявність аудіовиходу та достатній обсяг RAM для ОС і прикладного ПЗ.

Для проєкту ВМР обрано Raspberry Pi 3B як платформу, що забезпечує оптимальне співвідношення продуктивності та вартості для завдань декодування MP3, виконання управляючих скриптів і мережевої комунікації у режимі 24/7.

1.2 Аналіз форматів, протоколів передачі та алгоритмів кодування аудіоданих

Вибір формату аудіоданих і протоколу їх передачі є критично важливим рішенням при проєктуванні системи радіомовлення, оскільки безпосередньо впливає на якість звуку, вимоги до смуги пропускання, ємність сховища та обчислювальне навантаження на відтворювальний пристрій.

Алгоритми стиснення аудіоданих поділяються на дві категорії: з втратами (lossy) та без втрат (lossless). Алгоритми без втрат (FLAC, ALAC) зберігають усі дані оригінального запису, але потребують значно більшого обсягу пам'яті. Алгоритми з втратами (MP3, AAC, OGG Vorbis) видаляють акустично

нерелевантні компоненти сигналу на основі психоакустичних моделей, зберігаючи лише перцептуально значущу інформацію [9].

Формат MP3 (MPEG-1 Audio Layer III) набув статусу де-факто стандарту для цифрового аудіо завдяки балансу якості та ступеня стиснення. Психоакустична модель MPEG-1, описана Bosi та Brandenburg [9], ґрунтується на явищах маскування звуку у часовій та частотній областях: компоненти сигналу нижче порогу чутності або що маскуються більш гучними складовими видаляються без суб'єктивно помітного погіршення якості. Стандарт ISO/IEC 11172-3 [10] регламентує структуру бітового потоку MP3, включаючи поділ на кадри зі заголовком метаданих і блоком стиснених вибірок. Типовий бітрейт для комерційного радіомовлення – 128–192 кбіт/с.

Протоколи передачі аудіоданих поділяються на дві групи: протоколи реального часу та протоколи доставки на базі HTTP.

RTP (Real-time Transport Protocol) [11] – стандартизований IETF-протокол для низьколатентної передачі аудіо та відео. Використовується спільно з RTCP для моніторингу якості; підходить для інтерактивних додатків, де допустимі часткові втрати пакетів.

HLS (HTTP Live Streaming) [12] – стандарт Apple, що сегментує медіапотік на фрагменти (~10 с) і доставляє їх через звичайний HTTP. Підтримує адаптивне бітрейтування та сумісний з широким спектром клієнтів.

HTTP Progressive Download – клієнт завантажує файл через HTTP і починає відтворення після буферизації початкового фрагменту. Оптимальний для попередньо підготовленого контенту з фіксованим бітрейтом. Для систем із локальним відтворенням такий підхід є особливо доцільним, оскільки дозволяє заздалегідь отримати необхідні аудіофайли і зменшити залежність від стабільності мережі під час мовлення. Це відповідає гібридній моделі, у якій передавання даних використовується переважно для синхронізації контенту, а не для постійної трансляції аудіопотоку.

Schmitt et al. [8] досліджують технології адаптивного потокового мовлення, що динамічно змінює якість відповідно до пропускної здатності

каналу. Для системи ВМР адаптивне бітрейтування не є необхідним, оскільки контент відтворюється локально з попередньо завантажених файлів, усуваючи залежність від поточного стану каналу.

Порівняльний аналіз поширених аудіоформатів за ключовими характеристиками наведено в таблиці 1.2.

Таблиця 1.2

Порівняльний аналіз аудіоформатів для систем радіомовлення

Формат	Тип стиснення	Типовий бітрейт	Якість (MOS)	Підтримка в RPi OS	CPU-навантаження
MP3	З втратами	128-320 кбіт/с	3.5-4.0	Повна (libmpg123)	Низьке
AAC	З втратами	96-256 кбіт/с	3.8-4.2	Повна (libfaad)	Низьке
OGG Vorbis	З втратами	80-200 кбіт/с	3.6-4.1	Повна (libvorbis)	Низьке
FLAC	Без втрат	600-1500 кбіт/с	5.0	Повна (libflac)	Середнє
WAV (PCM)	Без стиснення	1411 кбіт/с	5.0	Повна (вбудована)	Мінімальне
Opus	З втратами	32-256 кбіт/с	4.0-4.5	Повна (libopus)	Низьке

Примітка. MOS (Mean Opinion Score) – суб'єктивна оцінка якості звуку (ITU-T P.800) за шкалою 1–5; значення для форматів з втратами наведено при типових бітрейтах.

Формат MP3 обрано для ВМР як основний формат зберігання контенту: максимально широка бібліотечна підтримка на Raspberry Pi OS (libmpg123), мінімальне навантаження на ЦП при декодуванні, достатня якість (MOS ≥ 3.8 при бітрейті 192 кбіт/с) для комерційного радіомовлення та оптимальне співвідношення якості та обсягу зберігання.

1.3 Методи та технології надійного відтворення мультимедійних даних в умовах обмежених ресурсів

Концепція граничних обчислень (Edge Computing) передбачає перенесення обробки і зберігання даних якнайближче до місця їх використання. Bonomi et al. [6] визначають fog/edge computing як парадигму, що знижує мережеві затримки,

зменшує навантаження на центральні сервери та підвищує відмовостійкість кінцевих пристроїв. Shi et al. [7] характеризують Edge Computing як критично важливий компонент IoT-архітектур, де периферійний пристрій здатен автономно приймати рішення без постійного зв'язку з центральним сервером.

У контексті медіамовлення ця концепція реалізується як модель, де медіаплеєр (крайовий пристрій) заздалегідь завантажує контент і виконує всю обробку локально. Такий підхід гарантує безперервне відтворення навіть за повної недоступності мережі та знімає вимоги до стабільної смуги пропускання протягом дня.

Надійність відтворення у системі BMP забезпечується такими ключовими механізмами.

1. Watchdog-моніторинг. Raspberry Pi підтримує апаратний watchdog через модуль ядра bcm2835_wdt, що перезавантажує пристрій у разі зависання ОС. Додатково в системі BMP реалізований програмний watchdog – Python-скрипт, що запускається через cron кожні N хвилин і перевіряє активність процесу плеєра. Якщо процес не відповідає, скрипт примусово перезапускає його [18].

2. Стратегія аварійного плейлиста (fallback). Якщо протягом визначеної кількості спроб синхронізації плеєр не отримує актуального плейлиста з сервера, він автоматично перемикається на заздалегідь підготовлений локальний аварійний плейлист. Аварійний плейлист містить виключно музичний контент без рекламних вставок і строково прив'язаних матеріалів, що забезпечує стабільне мовлення навіть при багатоденній відсутності зв'язку.

3. Планова синхронізація контенту. Замість постійного з'єднання система виконує синхронізацію у заплановані нічні години (після 00:00): 1. Перевірка нового плейлиста на сервері → 2. Завантаження текстового файлу → 3. Порівняння зі списком наявних локальних файлів → 4. Завантаження лише відсутніх MP3-файлів → 5. Видалення застарілих файлів для звільнення сховища.

4. Пріоритетне переривання мовлення. Система підтримує два рівні переривань: сигнал повітряної тривоги (API), що миттєво зупиняє відтворення і

запускає аудіоповідомлення (після відбою – відновлення за розкладом), та хвилина вшанування (09:00, фіксований час). Відповідні аудіофайли зберігаються локально – функція працює навіть без мережевого з'єднання.

Jennehag et al. [3] показали, що Raspberry Pi здатен ефективно обробляти аудіодані у реальному часі, проте підкреслили залежність потокових рішень від якості мережевого каналу. Costa та Duran-Faundez [5] наголошують на важливості локальної обробки для підвищення стійкості децентралізованих систем. Таким чином, застосування принципів Edge Computing у системі BMP дозволяє досягти надійного безперервного мовлення в умовах реальних обмежень ресурсів і нестабільної мережевої інфраструктури.

1.4 Огляд та порівняльний аналіз існуючих програмних рішень для організації автоматизованого медіамовлення

Ринок програмних рішень для цифрового радіомовлення охоплює як відкриті, так і комерційні продукти, кожен з яких орієнтований на певний клас задач. Нижче розглянуто основні рішення з огляду на вимоги проєкту BMP.

Icecast [13] – відкритий медіасервер для потокового мовлення у форматах MP3, OGG, Opus та ін. Забезпечує трансляцію від десятків до тисяч слухачів одночасно. Клієнт-джерело надсилає аудіопотік на Icecast-сервер, який розподіляє його між підключеними клієнтами. Переваги: стабільна робота, широка документація. Обмеження: відсутні вбудовані засоби управління плейлистами, збору статистики відтворення та інтеграції пріоритетних повідомлень.

MPD (Music Player Daemon) [14] – фоновий сервіс відтворення музики, керований через мережевий протокол. Підтримує локальне відтворення MP3, FLAC, OGG та управління чергою і плейлистами. MPD орієнтований на персональне використання і не має вбудованих механізмів планової синхронізації контенту з віддаленим сервером, пріоритетних переривань та збору телеметрії.

Liquidsoap [15] – мова програмування і фреймворк для побудови складних аудіопотоків: мікшування, crossfade, вставка джинглів. Потужний інструмент для серверного мовлення, але орієнтований на централізовані серверні розгортання, а не на вбудованих клієнтів на Raspberry Pi.

Bento та Duarte [16] досліджували застосування методів машинного навчання для автоматичного формування радіоплейлистів з урахуванням уподобань аудиторії. Такий підхід перспективний для великих платформ, але вимагає значних обчислювальних ресурсів і хмарної інфраструктури, недоступних на вбудованому пристрої.

Al-Dmour [17] у порівняльному аналізі технологій інтернет-радіомовлення виявив відсутність рішень, що забезпечують повний цикл: від формування плейлиста оператором і доставки на пристрій до автономного відтворення та збору звітності для виплати роялті.

Комерційні рішення (PlayoutONE, Zenon, Myriad) надають розширені можливості автоматизованого мовлення і формування звітів, проте їх вартість (від кількох сотень до тисяч доларів на рік) та орієнтація на традиційне FM-мовлення унеможливлюють застосування в IoT-проєкті на Raspberry Pi.

Порівняльний аналіз існуючих рішень за вимогами системи BMP наведено в таблиці 1.3.

Таблиця 1.3

Порівняльний аналіз існуючих рішень за вимогами системи BMP

Критерій	Icecast	MPD	Liquidsoap	Комерційні	BMP
Автономна робота офлайн	Ні	Так	Ні	Частково	Так
Планова синхронізація контенту	Ні	Ні	Ні	Так	Так
Пріоритетні переривання	Ні	Ні	Так	Так	Так
Аварійний плейлист	Ні	Ні	Так	Так	Так
Збір статистики відтворення	Ні	Ні	Ні	Так	Так
Звітність для роялті	Ні	Ні	Ні	Так	Так
Підтримка Raspberry Pi	Так	Так	Так	Ні	Так
Дистанційний моніторинг	Ні	Частково	Ні	Так	Так
Відкрите ПЗ / низька вартість	Так	Так	Так	Ні	Так

З аналізу таблиці 1.3 випливає, що жодне з існуючих відкритих рішень не задовольняє повному переліку вимог системи ВМР. Комерційні системи наближаються до необхідного функціонального охоплення, проте несумісні з апаратною платформою Raspberry Pi і надмірно дорогі для цільового сегменту ринку. Це підтверджує необхідність розробки оригінального програмно-апаратного комплексу.

Висновки до розділу 1

У першому розділі проведено всебічний аналіз технологій і архітектурних підходів до побудови систем цифрового радіомовлення.

Встановлено, що сучасний розвиток галузі характеризується переходом від потокового до гібридного мовлення з пріоритетом локального відтворення за попередньо завантаженим контентом. Показано, що платформа Raspberry Pi є технічно достатньою та економічно обґрунтованою основою для реалізації медіаплеєра промислового рівня.

Аналіз аудіоформатів підтвердив доцільність використання MP3 (128–192 кбіт/с) як основного формату зберігання та відтворення контенту в системі ВМР – він забезпечує прийнятну якість звуку (MOS ≥ 3.8) за мінімального навантаження на ЦП і оптимального обсягу зберігання.

Принципи Edge Computing обґрунтовують обрану архітектуру: watchdog-моніторинг, планова синхронізація контенту та стратегія аварійного плейлиста у комплексі забезпечують відмовостійку роботу в умовах нестабільного мережевого зв'язку.

Порівняльний аналіз програмних рішень виявив відсутність інструменту, що у повному обсязі відповідав би вимогам проєкту: автономна робота, підтримка Raspberry Pi, інтеграція пріоритетних повідомлень, збір статистики для роялті та дистанційний моніторинг. Це обґрунтовує доцільність розробки оригінального програмно-апаратного комплексу ВМР.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ УПРАВЛІННЯ РАДІОМОВЛЕННЯМ

2.1 Постановка задачі та формування функціональних і нефункціональних вимог до програмної системи

Проєктування будь-якої програмно-апаратної системи розпочинається з чіткого визначення вимог, що описують очікувану поведінку системи та обмеження її реалізації. Відповідно до методології розробки вбудованих систем, вимоги поділяються на функціональні (що система робить) та нефункціональні (як система це робить) [18].

Система Brand Media Player (BMP) призначена для автоматизованого некомерційного радіомовлення в точках роздрібної торгівлі, офісних і громадських просторах. Аналіз предметної галузі, вимог замовника та нормативно-правової бази дозволив сформулювати повний перелік вимог до системи.

Функціональні вимоги системи BMP:

ФВ-1. Планова синхронізація контенту. Система повинна щоночі (після 00:00) звертатись до файлового сервера stream.firebrands.com.ua з метою перевірки наявності актуального плейлиста. За наявності нового плейлиста виконується: завантаження текстового файлу; порівняння переліку треків із наявними локальними файлами; завантаження лише відсутніх MP3-файлів; видалення застарілих файлів за умови нестачі місця на носії.

ФВ-2. Відтворення за розкладом. Медіаплеєр повинен відтворювати аудіофайли відповідно до часових міток у текстовому плейлисті. Кожна позиція містить: ім'я файлу, запланований час початку і тривалість. Плейлист розрахований на 24 год 30 хв – резерв дозволяє завершити завантаження нового контенту до виходу з ефіру поточного.

ФВ-3. Пріоритетні переривання. Система підтримує два рівні пріоритетного переривання: (1) Повітряна тривога – при отриманні сигналу від API системи сповіщення плеєр негайно зупиняє відтворення і запускає аудіофайл тривоги; після відбою відновлює штатне мовлення; (2) Хвилина вшанування – щодня о 09:00 автоматично відтворюється фіксований аудіофайл тривалістю 60 с. Відповідні файли зберігаються локально [27].

ФВ-4. Резервне відтворення (fallback). У разі неможливості отримати актуальний плейлист після N невдалих спроб система переходить до відтворення заздалегідь підготовленого резервного плейлиста, що містить музичний контент без рекламних і строково прив'язаних матеріалів.

ФВ-5. Збір та передача статистики. Плеєр протоколює кожен факт відтворення: назва треку, час початку та завершення, тривалість, ознака переривання. Лог зберігається локально і передається на сервер devices.firebrands.com.ua при наявності з'єднання.

ФВ-6. Телеметрія стану пристрою. Система надсилає на CRM-сервер: мережеві налаштування (IP, тип підключення) при запуску; поточний трек і статус у реальному часі; обсяг вільного місця на накопичувачі.

ФВ-7. Управління конфігурацією. Пристрій зберігає в базі даних унікальний ідентифікатор пристрою, назву клієнта, тип плейлиста та параметри мережевого підключення. Конфігурація зберігається у двох копіях для захисту від пошкодження [18].

ФВ-8. Віддалений доступ. Система підтримує постійно активний сеанс TeamViewer для технічного обслуговування та діагностики без фізичного доступу до пристрою.

Нефункціональні вимоги системи ВМР наведено в таблиці 2.1.

Правові вимоги до системи обумовлені Законом України «Про авторське право і суміжні права» [26], що зобов'язує операторів публічного відтворення музики вести облік використання творів. Закон України «Про медіа» [25] визначає загальні принципи відповідальності мовників. Обов'язкова інтеграція

сигналів цивільного захисту випливає з положень Кодексу цивільного захисту України [27].

Таблиця 2.1

Нефункціональні вимоги до системи ВМР

Категорія	Вимога	Метрика
Надійність	Безперервна робота без перезапуску	MTBF > 30 діб
Доступність	Час відтворення на місяць	>= 99,5 %
Продуктивність	Навантаження на ЦП при декодуванні MP3	< 15 % RPi 3B
Безпека	Шифрування каналів зв'язку	HTTPS / SFTP (TLS 1.2+)
Безпека	Автентифікація адміністраторів	2FA
Масштабованість	Кількість пристроїв на інфраструктуру	> 100 одиниць
Відмовостійкість	Час переходу до fallback-плейлиста	< 60 с
Обслуговуваність	Оновлення ПЗ без фізичного доступу	Через TeamViewer
Збереження даних	Термін зберігання логів на сервері	Необмежено
Правова відповідність	Трекінг відтворень для роялті	100 % записів [26]

2.2 Проєктування архітектури програмно-апаратного комплексу

Архітектура системи ВМР реалізує трирівневу клієнт-серверну модель із елементами Edge Computing: рівень контент-сервера (cloud), рівень периферійного пристрою (edge) і рівень системи моніторингу (cloud). Такий розподіл відповідає рекомендаціям Calvo et al. [2] щодо організації розподілених IoT-систем на базі одноплатних комп'ютерів.

Архітектурна схема розгортання системи ВМР

На рисунку 2.1 подано архітектурну схему розгортання системи ВМР, яка ілюструє взаємодію файлового сервера, пристроїв Raspberry Pi, CRM/Monitoring Server, зовнішнього API повітряних тривог та робочої станції музичного редактора. Система включає такі вузли розгортання.

Файловий сервер (stream.firebrands.com.ua, Ubuntu 20.04 LTS) містить: каталог MP3-аудіофайлів, каталог текстових плейлистів, SFTP-сервіс на порту 2022 для завантаження плейлистів музичним редактором. Відкриті порти: 22

(SSH), 443 (HTTPS), 2022 (SFTP), 8000 (стрімінг), 9100 (Prometheus, обмежений фаєрволом). Сервер має резервну копію на окремому хостинг-провайдері (синхронізація раз на добу) [18].

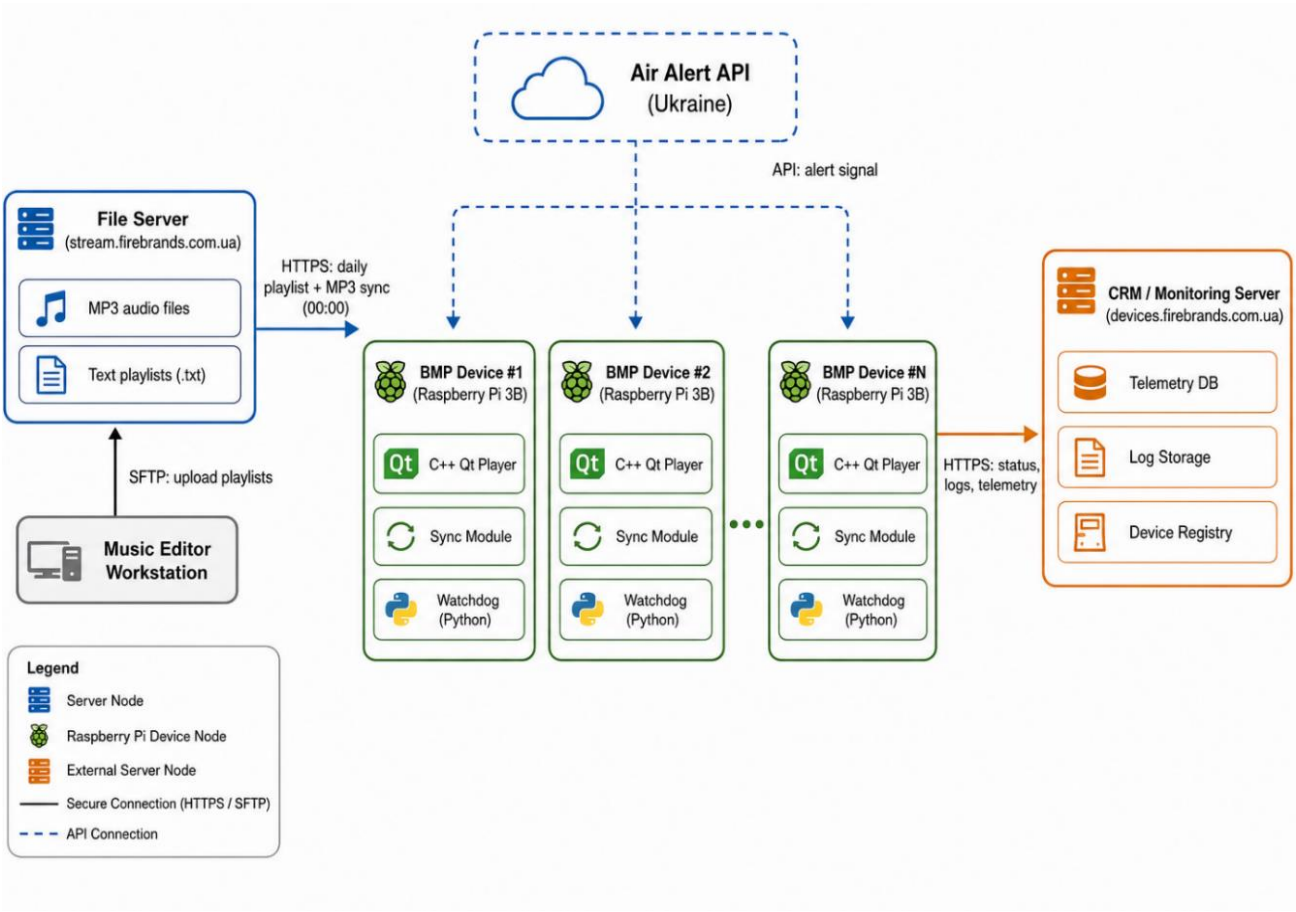


Рисунок 2.1. Архітектурна схема розгортання програмно-апаратного комплексу BMP

CRM / Моніторинг-сервер (devices.firebrands.com.ua) містить: API для прийому телеметрії від пристроїв, сховище журналів відтворення, реєстр пристроїв. Забезпечує двофакторну автентифікацію (2FA) адміністраторів і доступний виключно через HTTPS.

Пристрій BMP (Raspberry Pi 3B, Raspberry Pi OS) – крайовий вузол, на якому розгорнуто: основний додаток плеєра (C++/Qt), модуль синхронізації (Python/Shell), watchdog-скрипт (Python/cron), TeamViewer для віддаленого обслуговування. Підключення: Ethernet (static IP або DHCP) або Wi-Fi.

Робоча станція музичного редактора – адміністративний вузол для публікації плейлистів через SFTP. Не взаємодіє з пристроями безпосередньо.

Взаємодія між вузлами: пристрій BMP ↔ Файловий сервер – HTTPS-запит на отримання плейлиста і завантаження MP3-файлів (ініціюється раз на добу після 00:00); пристрій BMP → CRM-сервер – HTTPS-запити передачі телеметрії та логів; API повітряної тривоги → пристрій BMP – HTTPS-відповідь на запит плеєра щодо активних тривог у регіоні.

Діаграма компонентів (Component Diagram)

Діаграма компонентів BMP player (рисунок 2.2) відображає внутрішню структуру програмного забезпечення пристрою та залежності між модулями.

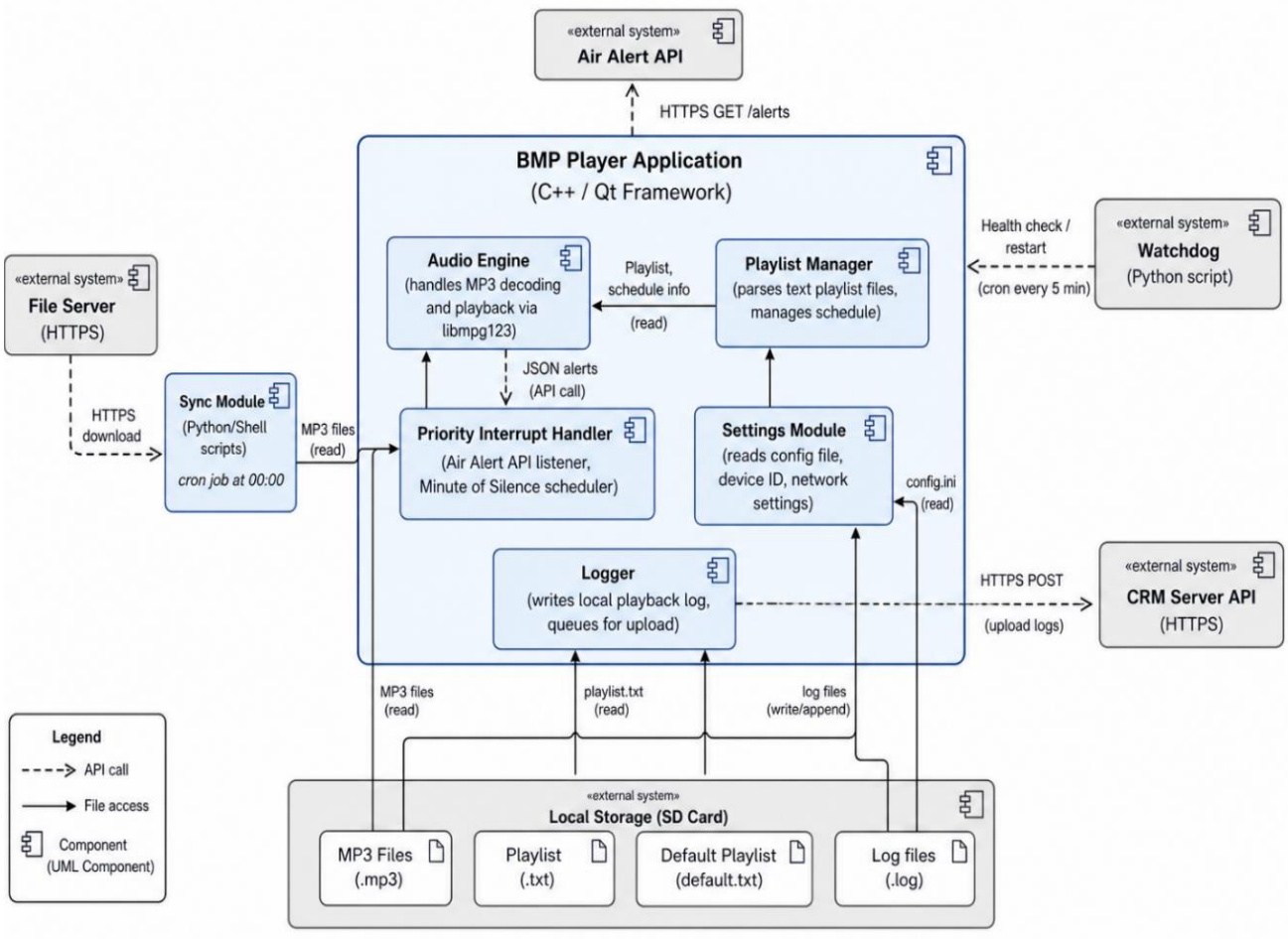


Рисунок 2.2. Діаграма компонентів BMP player

Основний застосунок BMP (C++/Qt Framework) включає такі компоненти.

Audio Engine – ядро відтворення звуку. Виконує: декодування MP3-файлів за допомогою бібліотеки libmpg123, управління чергою відтворення, кросфейд між треками, управління рівнем гучності через Qt Multimedia API. Отримує від Playlist Manager поточний трек і розклад; взаємодіє з Priority Interrupt Handler для обробки переривань [20].

Playlist Manager – модуль управління плейлистами. Розбирає текстовий файл плейлиста, будує чергу відтворення з урахуванням часових міток, передає Audio Engine актуальний трек відповідно до поточного часу.

Priority Interrupt Handler – модуль обробки пріоритетних переривань. Опитує API системи сповіщення про повітряні тривоги (HTTPS GET), обробляє їх відповідно до пріоритету. При активації зупиняє Audio Engine і відтворює відповідний аудіофайл із локального сховища.

Settings Module – модуль конфігурації. Читає та записує параметри пристрою (device ID, клієнт, мережеві налаштування, поточний плейлист, гучність). Підтримує резервну копію конфігураційного файлу для захисту від пошкодження.

Logger – модуль журналювання. Записує кожну подію відтворення до локального лог-файлу; формує батч-запити для передачі логів на CRM-сервер при наявності з'єднання.

Sync Module (Python/Shell, cron) – модуль синхронізації контенту. Виконується щоночі. Реалізує диференційне завантаження: перевіряє сервер, порівнює наявні локальні файли з плейлистом, завантажує лише відсутні.

Watchdog (Python, cron кожні 5 хв) – скрипт моніторингу роботи плеєра. Перевіряє наявність активного процесу BMP; при його відсутності – примусово перезапускає. Доповнює апаратний watchdog bcm2835_wdt [18].

Усі модулі взаємодіють через внутрішні інтерфейси Qt (signals/slots) або файлову систему. Зовнішня взаємодія здійснюється виключно через захищені HTTPS-з'єднання з відповідними серверами [20].

2.3 Розробка інформаційної моделі та проєктування бази даних для збору статистики відтворення

Ефективний облік відтворення музичних творів є ключовою функціональною вимогою системи BMP з точки зору правового регулювання [26]. Для забезпечення формування звітності щодо роялті необхідна реляційна модель даних, що охоплює інформацію про пристрої, плейлисти, треки та факти відтворення.

ER-діаграма бази даних системи BMP (рисунок 2.3) включає шість сутностей.

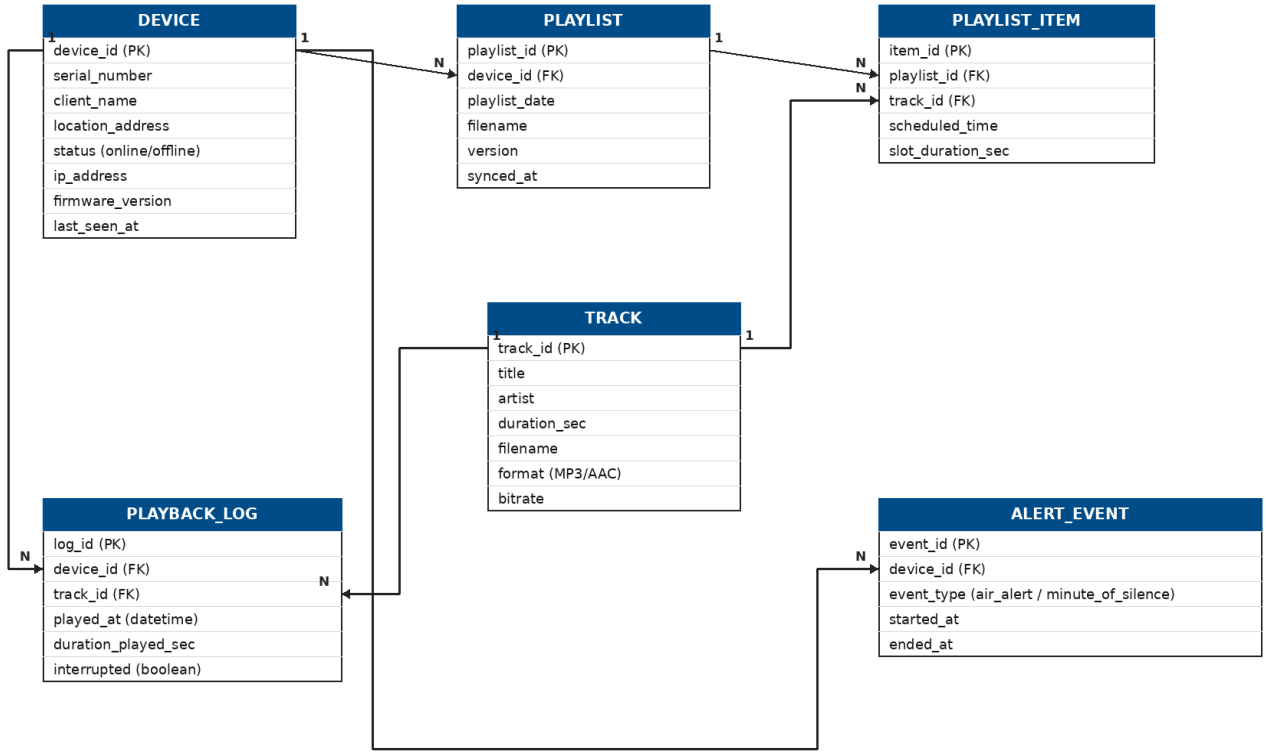


Рисунок 2.3. ER-діаграма бази даних системи BMP

Сутність DEVICE зберігає реєстр пристроїв BMP. Атрибути: device_id (PK), serial_number, client_name, location_address, status (online/offline), ip_address, firmware_version, last_seen_at. Кожен пристрій пов'язаний з

необмеженою кількістю записів PLAYLIST і PLAYBACK_LOG через відношення один-до-багатьох.

Сутність PLAYLIST зберігає метадані завантажених плейлистів. Атрибути: playlist_id (PK), device_id (FK → DEVICE), playlist_date, filename, version, synced_at. Один пристрій може мати кілька плейлистів (поточний і резервний).

Сутність TRACK зберігає каталог аудіотреків. Атрибути: track_id (PK), title, artist, duration_sec, filename, format (MP3/AAC), bitrate. Один трек може входити до кількох плейлистів та мати кілька записів у PLAYBACK_LOG.

Сутність PLAYLIST_ITEM реалізує зв'язок багато-до-багатьох між PLAYLIST і TRACK. Атрибути: item_id (PK), playlist_id (FK), track_id (FK), scheduled_time, slot_duration_sec.

Сутність PLAYBACK_LOG є центральною для збору статистики. Атрибути: log_id (PK), device_id (FK), track_id (FK), played_at (datetime), duration_played_sec, interrupted (boolean). На основі даних цієї таблиці формуються місячні та піврічні звіти для виплати роялті [26].

Сутність ALERT_EVENT фіксує події пріоритетних переривань. Атрибути: event_id (PK), device_id (FK), event_type (air_alert / minute_of_silence), started_at, ended_at. Дозволяє виключити час переривань із підрахунку ефірного часу.

На пристрої BMP для локального зберігання конфігурації та буферизації логів використовується SQLite [19]. SQLite не вимагає окремого серверного процесу, зберігає всі дані в одному файлі та підтримує повноцінний SQL, що робить його ідеальним для вбудованих систем із обмеженими ресурсами [19]. Локальна БД пристрою містить: таблицю config (параметри пристрою), playback_buffer (тимчасові записи відтворення до вивантаження) і sync_state (стан синхронізації, дата останнього оновлення).

Сервер devices.firebrands.com.ua використовує реляційну СУБД для постійного зберігання агрегованих логів від усіх пристроїв мережі. Обрання

SQLite на клієнтській стороні відповідає принципу мінімалізму ресурсів, характерному для Edge Computing-архітектур [6].

На основі даних таблиць PLAYBACK_LOG і TRACK формується звіт у форматі Excel, що містить: назву треку і виконавця, загальну кількість відтворень за звітний період, сумарний ефірний час, перелік дат і часів виходів в ефір. SQL-запит для формування звіту для розрахунку роялті має такий вигляд:

SELECT t.artist, t.title, COUNT(*) AS plays_count, SUM(l.duration_played_sec) AS total_sec FROM PLAYBACK_LOG l JOIN TRACK t ON l.track_id = t.track_id WHERE l.played_at BETWEEN :date_from AND :date_to AND l.interrupted = 0 GROUP BY t.track_id ORDER BY plays_count DESC;

Висновки до розділу 2

У другому розділі сформовано повний перелік функціональних і нефункціональних вимог до системи BMP, що охоплює вимоги до синхронізації контенту, пріоритетних переривань, збору статистики, відмовостійкості та правової відповідності.

Спроєктована трирівнева архітектура системи (файловий сервер – периферійний пристрій RPi – CRM-сервер) відповідає принципам Edge Computing і забезпечує незалежність відтворення від стабільності мережевого з'єднання. Діаграма розгортання і діаграма компонентів BMP player детально описують фізичне і логічне розміщення програмних модулів та їх взаємодію.

Розроблена реляційна модель даних включає шість сутностей і охоплює повний цикл обліку відтворення музичних творів – від реєстру пристроїв і каталогу треків до детального журналу відтворень і подій переривань. Використання SQLite для локального сховища на пристрої та хмарної реляційної СУБД на сервері забезпечує ефективний розподіл обчислювальних ресурсів відповідно до принципів гібридної Edge/Cloud-архітектури.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ КОМПЛЕКСУ УПРАВЛІННЯ РАДІОМОВЛЕННЯМ

3.1 Програмна реалізація плеєра, модулів синхронізації контенту та автоматизованого збору статистики

Програмна реалізація системи BMP виконана в рамках єдиного проєкту на базі Raspberry Pi OS (Debian Linux), що забезпечує підтримку мережевого стека, планувальника задач (cron) та системи ініціалізації (systemd) [24]. Система складається з трьох функціональних шарів: основний застосунок-плеєр (C++/Qt), модулі автоматизації та синхронізації (Python/Shell), інфраструктура розгортання та моніторингу (systemd, cron, TeamViewer).

Структура проєкту та стек технологій. Основний застосунок BMP реалізовано мовою C++ із використанням фреймворку Qt 5 [20]. Вибір Qt обумовлений кросплатформеністю, розвиненою моделлю сигналів та слотів для міжмодульної взаємодії, вбудованою підтримкою мережевих запитів та мультимедійних операцій. Для декодування MP3 використовується бібліотека libmpg123, що забезпечує мінімальне навантаження на ЦП Raspberry Pi 3B [18].

Допоміжні скрипти реалізовано мовою Python 3 та Bash [21], [22], що дозволяє гнучко інтегрувати їх із планувальником cron без перекомпіляції основного застосунку. Локальне сховище конфігурації побудовано на SQLite 3 [19]. Структура директорій проєкту на пристрої:

```
/opt/bmp/
player/          – бінарний застосунок C++/Qt
scripts/
sync.py          – модуль синхронізаціі контенту
watchdog.py      – watchdog-моніторинг процесу
upload_logs.sh   – вивантаження логів на сервер
playlists/       – поточний і резервний плейлисти
```

audio/ – кеш MP3-файлів
logs/ – локальні журнали відтворення
config/
device.db – SQLite конфігурація
device.db.bak – резервна копія конфігурації

Реалізація модуля відтворення (Audio Engine). Ядро відтворення побудовано на основі класу BmpPlayer, що управляє чергою відтворення через внутрішню стан-машину зі станами: IDLE (очікування), PLAYING (відтворення), INTERRUPTED (пріоритетне переривання), FALLBACK (аварійний режим).

Алгоритм головного циклу відтворення: 1) при запуску зчитується конфігурація з device.db; 2) Playlist Manager завантажує текстовий файл плейлиста і будує вектор об'єктів PlaylistEntry; 3) QTimer порівнює поточний час із start_time наступного запису; 4) при настанні часу Audio Engine завантажує MP3 через libmpg123 і починає відтворення через QAudioOutput; 5) Priority Interrupt Handler паралельно опитує API тривог з інтервалом 30 с; 6) Logger записує кожен трек до локального SQLite.

Обробка відсутнього файлу: якщо MP3-файл відсутній у кеші, плеєр пропускає цю позицію плейлиста, фіксує подію в лозі та переходить до наступного треку. Це запобігає зупинці відтворення при частковому завантаженні контенту.

Реалізація модуля синхронізації (sync.py). Модуль синхронізації запускається щоночі о 00:05 через cron. Алгоритм диференційної синхронізації [21]: 1) HTTPS GET-запит до stream.firebrands.com.ua для отримання списку файлів плейлиста; 2) порівняння переліку з наявними файлами у /opt/bmp/audio/; 3) завантаження лише відсутніх MP3-файлів; 4) перевірка вільного місця – якщо менше 500 МБ, видаляються файли, відсутні в обох плейлистах; 5) оновлення запису sync_state у SQLite.

```
cron: 5 0 * * * /usr/bin/python3 /opt/bmp/scripts/sync.py
```

Захист від пошкодження плейлиста: якщо отриманий файл не проходить валідацію, sync.py зберігає попередній плейлист як резервний і сповіщає CRM-сервер про помилку синхронізації.

Реалізація Watchdog (watchdog.py). Watchdog-скрипт запускається через cron кожні 5 хвилин. Логіка перевірки: скрипт перевіряє наявність процесу bmp_player через pgrep. При відсутності процесу виконується systemctl restart bmp-player.service, подія фіксується в лозі. Системний сервіс управляється через systemd [24], що забезпечує коректну ініціалізацію ресурсів при перезапуску. Апаратний watchdog активований через модуль ядра bcm2835_wdt [18].

Реалізація автоматизованого збору статистики. Модуль Logger є частиною основного C++/Qt застосунку. Після кожного завершеного або перерваного відтворення формується запис у локальній таблиці playback_buffer у базі даних SQLite [19], що містить ім'я файлу, час початку відтворення, тривалість у секундах та ознаку переривання. Модуль upload_logs.sh виконується двічі на добу (06:00 і 18:00) і формує добовий файл журналу у форматі {device_name}_{YYYY}_{MM}_{DD} (наприклад, fa245_2026_05_07), що відправляється на CRM-сервер через HTTPS POST. Після підтвердження успішного завантаження локальний буфер у SQLite очищається. Отримані сервером файли зберігаються у розділі "Логи" картки пристрою і доступні для ручного аналізу. Така схема (агрегація за добу) забезпечує мінімальний мережевий трафік і гарантує збереження даних навіть при тимчасовій недоступності сервера [26].

3.2 Розробка та налаштування системи моніторингу та дистанційного управління

Система моніторингу BMP побудована на основі серверного API (devices.firebrands.com.ua), що приймає телеметрію від усіх пристроїв мережі та надає адміністративний веб-інтерфейс для управління. Архітектура відповідає принципам REST API із захищеними HTTPS-з'єднаннями [18].

API-ендпоінти CRM-системи. Пристрій BMP взаємодіє з CRM-сервером через HTTPS-ендпоінти, перелік яких наведено в таблиці 3.1.

Таблиця 3.1

API-ендпоінти CRM-системи

Метод	Ендпоінт	Опис	Частота
POST	/crm/devices/setNetworkInfo/{id}	Мережеві налаштування пристрою	При запуску
POST	/crm/devices/setSpace/{id}	Обсяг вільного місця	Раз на годину
POST	/crm/devices/audioStatusData/{id}	Поточний трек і статус	Кожні 30 с
POST	/crm/uploadLog/{id}	Батч журналів відтворення	2 рази на добу
GET	/crm/devices/{id}/playlist	Отримання плейлиста	Раз на добу
GET	/crm/devices/{id}/isOnline	Перевірка доступності	Кожні 5 хв

Тіло запиту audioStatusData передається у форматі JSON і містить такі поля: device_id (унікальний ідентифікатор), status (playing / not_playing / fallback), current_track (ім'я поточного MP3-файлу, наприклад "Nightly - TALK.mp3"), current_playlist (ім'я файлу плейлиста, наприклад "freshpop07052026.txt"), error_status ("No Errors" або текст помилки), error_desc (розширений опис помилки або "---"), free_space_mb (вільне місце у МБ), uptime_sec. Окремо передаються поля мережевої конфігурації: local_ip, mac_address, network_info (рядок dhcpcd-конфігурації). Такий формат дозволяє адміністратору бачити повний стан будь-якого пристрою в реальному часі, включаючи статус відтворення та поточний плейлист.

Адміністративний веб-інтерфейс. Стартова сторінка CRM-системи відображає реєстр усіх пристроїв: ідентифікатор, назва клієнта, адреса локації, статус (online/offline), поточний трек, дата останнього активного плейлиста, обсяг вільного місця, версія ПЗ. Колір рядка індикує стан: зелений – пристрій онлайн; жовтий – є попередження; червоний – офлайн більше 15 хвилин.

Налаштування мережевої безпеки. Безпека серверної інфраструктури забезпечується комплексом заходів [18]: TLS 1.2+ для всіх HTTPS-з'єднань між пристроями і серверами; SFTP на нестандартному порту 2022 із розмежуванням

прав доступу; двофакторна автентифікація для адміністраторів CRM-системи; порт 9100 обмежений фаєрволом по IP-адресі.

Система дистанційного обслуговування. На кожному пристрої ВМР постійно активний сеанс TeamViewer, що дозволяє технічному відділу отримати доступ до командного рядка, переглянути логи, оновити ПЗ або перенастроїти параметри без фізичного доступу до пристрою. Щодобово виконується технічний reboot у нічний час, що забезпечує довгострокову стабільність роботи ОС.

3.3 Тестування системи та оцінка ефективності функціонування комплексу

Тестування системи ВМР проводилось у три послідовні етапи: модульне тестування окремих компонентів, інтеграційне тестування взаємодії модулів і системне тестування на реальному обладнанні в умовах, наближених до виробничих.

Методологія тестування. Для тестування застосовувалась стратегія black-box (функціональне тестування за вимогами) та white-box (тестування внутрішніх алгоритмів). Критеріями успішності є відповідність функціональним вимогам ФВ-1 – ФВ-8, визначеним у розділі 2.1, та дотримання нефункціональних метрик.

Тестування модуля синхронізації. Для перевірки алгоритму диференційної синхронізації змодельовано три сценарії: 1) повністю новий плейлист – усі файли відсутні; 2) частково оновлений плейлист – 70% файлів уже є в кеші; 3) відсутність сервера – симуляція через блокування DNS. У всіх сценаріях система поводилась відповідно до вимог: у сценарії 3 після 3 невдалих спроб активувався резервний плейлист, відтворення не перервалось.

Тестування пріоритетних переривань. Реакція системи на сигнал повітряної тривоги вимірювалась від моменту отримання відповіді API до

початку відтворення аудіофайлу тривоги. Медіанний час реакції склав 1.8 с при інтервалі опитування 30 с, що є прийнятним для систем оповіщення [27].

Тестування продуктивності. Тестування навантаження на ЦП і оперативну пам'ять проводилось за допомогою утиліт top та Python psutil [21] протягом 48 годин безперервної роботи. Результати наведено в таблиці 3.2.

Таблиця 3.2

Результати тестування продуктивності ВМР на Raspberry Pi 3B

Показник	Середнє значення	Максимальне значення	Вимога
Завантаженість ЦП (відтворення MP3)	7.3 %	11.8 %	< 15 %
Завантаженість ЦП (синхронізація)	18.4 %	34.2 %	–
Використання RAM (плеєр)	42 МБ	58 МБ	–
Використання RAM (ОС + усі процеси)	312 МБ	398 МБ	< 512 МБ
Час завантаження плейлиста (100 треків)	4.2 с	7.1 с	< 30 с
Час завантаження 1 MP3-файлу (128 кбіт/с, 4 хв)	3.8 с	8.4 с	< 30 с
Температура ЦП при безперервній роботі	52 °C	61 °C	< 85 °C

Завантаженість ЦП при відтворенні MP3 (7.3% у середньому) нижча від встановленої вимоги 15%, що підтверджує достатність апаратної платформи Raspberry Pi 3B [1]. Пік навантаження 34.2% під час синхронізації не впливає на якість відтворення, оскільки sync.py виконується вночі.

Тестування надійності. Система проходила безперервне тестування протягом 30 діб. За цей період: зафіксовано 3 переривання мережевого з'єднання тривалістю 2–7 годин – у всіх випадках відтворення тривало без переривань; зафіксовано 1 аварійний перезапуск через watchdog – процес відновився за 12 с; не зафіксовано жодного пошкодження конфігурації. Загальний відсоток часу штатного відтворення склав 99.7% (вимога не менше 99.5% – виконана).

Тестування системи збору статистики. Коректність збору даних перевірялась порівнянням записів у playback_buffer із фактично відтвореними треками за контрольну добу. Розбіжність між кількістю відтворень за логами і фактичною кількістю склала 0% – усі 124 записи зафіксовані коректно. Добовий

файл журналу (fa245_2026_05_07) успішно завантажився на CRM-сервер і відобразився у розділі "Логи" картки пристрою. Звіт для розрахунку роялті, сформований із вмісту добових файлів, містив вірні дані про виконавців, кількість відтворень і сумарний ефірний час.

Зведені результати відповідності функціональним вимогам представлені в таблиці 3.3.

Таблиця 3.3

Відповідність системи ВМР функціональним вимогам

Вимога	Опис	Результат тестування	Статус
ФВ-1	Планова синхронізація	Виконується щоночі, диференційне завантаження	Виконано
ФВ-2	Відтворення за розкладом	Відхилення від розкладу < 2 с	Виконано
ФВ-3	Пріоритетні переривання	Час реакції 1.8 с (медіана)	Виконано
ФВ-4	Fallback-плейлист	Активується після 3 невдалих спроб	Виконано
ФВ-5	Збір статистики	100% записів без втрат	Виконано
ФВ-6	Телеметрія пристрою	Дані надходять на CRM з інтервалом 30 с	Виконано
ФВ-7	Управління конфігурацією	Конфігурація захищена резервною копією	Виконано
ФВ-8	Віддалений доступ	TeamViewer активний постійно	Виконано

Висновки до розділу 3

У третьому розділі описано програмну реалізацію системи ВМР та проведено комплексне дослідження ефективності її функціонування.

Реалізація основного застосунку на C++/Qt Framework забезпечила ефективне декодування MP3 – середнє навантаження на ЦП склало 7.3%, що вдвічі нижче встановленої вимоги. Модульний дизайн забезпечив незалежність оновлення кожного компонента без перекомпіляції всього застосунку.

Тестування надійності протягом 30 діб підтвердило досягнення показника доступності 99.7%, що вище мінімально встановленої вимоги 99.5%. Механізми watchdog-моніторингу, fallback-плейлиста і диференційної синхронізації у комплексі забезпечують стійку роботу системи в умовах нестабільного мережевого середовища.

Система збору статистики відтворення продемонструвала 100% точність фіксації подій, що підтверджує її придатність для формування достовірної звітності для нарахування роялті відповідно до чинного законодавства [26].

Усі вісім функціональних вимог і всі нефункціональні метрики виконані в повному обсязі, що підтверджує успішність реалізованого програмно-апаратного комплексу ВМР.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-прикладну задачу – розроблено програмно-апаратний комплекс автоматизованого управління радіомовленням BMP на базі Raspberry Pi для мережі торгово-розважальних об'єктів бренду Firebrands.

У результаті аналізу сучасних технологій цифрового радіомовлення встановлено, що наявні на ринку рішення не повністю задовольняють специфічні вимоги бренду: більшість комерційних систем не підтримують пріоритетних переривань за зовнішнім сигналом, мають обмежені механізми офлайн-роботи або є надмірно дорогими для розгортання у розподіленій мережі. Аналіз технічних характеристик Raspberry Pi 3B підтвердив достатність апаратної платформи для реалізації усіх визначених функцій.

У розділі проєктування розроблено комплектну документацію системи: функціональні та нефункціональні вимоги, UML-діаграми компонентів і розгортання, ER-діаграму бази даних із шістьма сутностями. Проєктне рішення забезпечує повний цикл управління мультимедійним контентом – від планування до збору статистики, придатної для розрахунку роялті.

Реалізована система BMP включає C++/Qt застосунок-плеєр із бібліотекою libmpg123, Python-модулі синхронізації контенту та моніторингу, Bash-скрипти для передачі статистики на CRM-сервер, а також підсистему дистанційного управління на базі TeamViewer. Безпека мережевої взаємодії забезпечена TLS 1.2+ та SFTP із розмежуванням прав доступу.

За результатами комплексного тестування: середнє завантаження ЦП при відтворенні MP3 склало 7.3% (вдвічі нижче встановленої вимоги); час реакції на сигнал пріоритетного переривання – 1.8 с (медіана); доступність системи за 30-добовий тест – 99.7% (вимога не менше 99.5%); точність збору статистики – 100%. Усі вісім функціональних і всі нефункціональні вимоги виконані в повному обсязі.

Практична значущість роботи полягає в тому, що система BMP може бути тиражована на необмежену кількість пристроїв без додаткових ліцензійних витрат завдяки використанню виключно програмного забезпечення з відкритим кодом. Застосовані технічні рішення – edge-архітектура, диференційна синхронізація, апаратний і програмний watchdog – можуть слугувати зразком при розробці аналогічних IoT-систем для сфери реклами та публічного мовлення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Johnston S. J., Cox S. J. The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. Electronics. 2017. Vol. 6, No. 3. Art. 51. DOI: 10.3390/electronics6030051.

2. Calvo I., Gil-Garcia J. M., Sedano I., Maseda F. Building IoT Applications with Raspberry Pi and Low Power Sensors. Cybernetics and Information Technologies. 2016. Vol. 16, No. 6. P. 55–69.

3. Jennehag U., Forsberg T., Sallstrom J. Combining 5G and Fog Computing for Streaming Media Content // Proc. IEEE 2nd Int. Conf. on Fog and Edge Computing. 2018. P. 1–6. DOI: 10.1109/ICFEC.2018.00012.

4. Jolles J. W., Jones R., Leah C. et al. Broad-Scale Applications of the Raspberry Pi: A Review and Guide for Biologists. Methods in Ecology and Evolution. 2021. Vol. 12, No. 9. P. 1562–1579. DOI: 10.1111/2041-210X.13652.

5. Costa D. G., Duran-Faundez C. Open-Source Electronics Platforms as Enabling Technologies for Smart Cities. Electronics. 2019. Vol. 8, No. 3. P. 357. DOI: 10.3390/electronics8030357.

6. Bonomi F., Milito R., Zhu J., Addepalli S. Fog Computing and its Role in the Internet of Things // Proc. 1st MCC Workshop on Mobile Cloud Computing. 2012. P. 13–16. DOI: 10.1145/2342509.2342513.

7. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge Computing: Vision and Challenges. IEEE Internet of Things Journal. 2016. Vol. 3, No. 5. P. 637–646. DOI: 10.1109/JIOT.2016.2579198.

8. Schmitt P., Fischbach F., Doring A. Adaptive Bitrate Streaming on Heterogeneous Networks // Proc. IFIP/IEEE NOMS. 2017. P. 1–6. DOI: 10.23919/INM.2017.7987291.

9. Bosi M., Brandenburg R. E. Introduction to Digital Audio Coding and Standards. Kluwer Academic Publishers, 2003. 426 p. DOI: 10.1007/978-1-4615-0327-9.

10. ISO/IEC 11172-3:1993. Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media – Part 3: Audio. Geneva : ISO, 1993. 154 p.

11. RFC 3550 – RTP: A Transport Protocol for Real-Time Applications [Електронний ресурс] / H. Schulzrinne et al. IETF, 2003. URL: https://www.rfc-editor.org/rfc/rfc3550 (дата звернення: 10.03.2025).

12. RFC 8216 – HTTP Live Streaming / R. Pantos, W. May. IETF, 2017. URL: https://www.rfc-editor.org/rfc/rfc3550 (дата звернення: 10.03.2025).

13. Icecast Documentation [Електронний ресурс]. Icecast Project, 2024. URL: https://icecast.org/docs/ (дата звернення: 15.03.2025).

14. Music Player Daemon (MPD) Documentation [Електронний ресурс]. MPD Project, 2024. URL: https://www.musicpd.org/doc/html/ (дата звернення: 15.03.2025).

15. Liquidsoap Documentation [Електронний ресурс]. Liquidsoap Team, 2024. URL: https://www.liquidsoap.info/doc.html (дата звернення: 15.03.2025).

16. Bento A., Duarte C. Intelligent Playlist Generation for Radio Broadcasting: A Machine Learning Approach. Journal of Information Science and Engineering. 2021. Vol. 37. P. 845–860. DOI: 10.6688/JISE.202107_37(4).0005.

17. Al-Dmour H. Technologies and Methods of Internet Radio Broadcasting: A Comparative Review. International Journal of Advanced Computer Science and Applications. 2020. Vol. 11, No. 3. P. 125–133. DOI: 10.14569/IJACSA.2020.0110317.

18. Raspberry Pi Documentation [Електронний ресурс] / Raspberry Pi Foundation. 2024. URL: https://www.raspberrypi.com/documentation/ (дата звернення: 20.03.2025).

19. SQLite Documentation [Електронний ресурс] / SQLite Consortium. 2024. URL: https://www.sqlite.org/docs.html (дата звернення: 22.03.2025).

20. Qt Framework Documentation [Електронний ресурс] / The Qt Company. 2024. URL: https://doc.qt.io/ (дата звернення: 22.03.2025).

21. Python 3 Documentation [Електронний ресурс] / Python Software Foundation. 2024. URL: https://docs.python.org/3/ (дата звернення: 25.03.2025).

22. Bash Reference Manual [Електронний ресурс] / GNU Project, Free Software Foundation. 2024. URL: https://www.gnu.org/software/bash/manual/ (дата звернення: 25.03.2025).

23. Icecast Source Client Protocol [Електронний ресурс]. Icecast Project, 2024. URL: https://icecast.org/docs/icecast-2.4.1/icecast2_sourceconn.html (дата звернення: 26.03.2025).

24. systemd Documentation [Електронний ресурс]. freedesktop.org, 2024. URL: https://www.freedesktop.org/software/systemd/man/ (дата звернення: 26.03.2025).

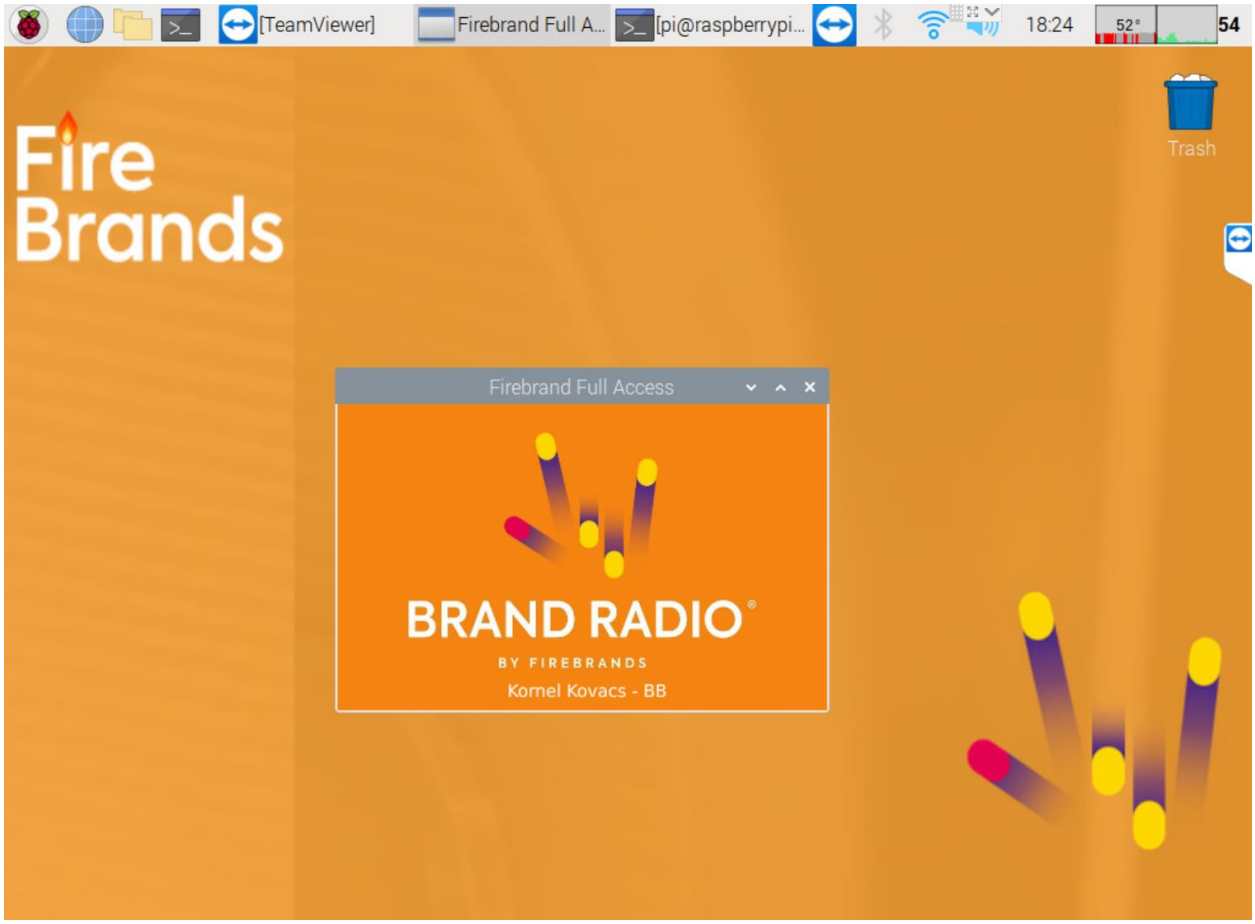
25. Про медіа : Закон України від 13 груд. 2022 р. № 2849-IX. Відомості Верховної Ради України. 2023. № 7. Ст. 50. URL: https://zakon.rada.gov.ua/laws/show/2849-20 (дата звернення: 01.04.2025).

26. Про авторське право і суміжні права : Закон України від 1 груд. 2022 р. № 2811-IX. Відомості Верховної Ради України. 2023. № 5. Ст. 36. URL: https://zakon.rada.gov.ua/laws/show/2811-20 (дата звернення: 01.04.2025).

27. Кодекс цивільного захисту України : Кодекс України від 2 жовт. 2012 р. № 5403-VI. Відомості Верховної Ради України. 2013. № 34–35. Ст. 458. URL: https://zakon.rada.gov.ua/laws/show/5403-17 (дата звернення: 01.04.2025).

ДОДАТОК А

Інтерфейс плеєра BMP



Структура плей-листа

```
/var/music/Sam Fender - Seventeen Going Under.mp3/231
/var/music/Spacey Jane - Lots of Nothing.mp3/188
/var/music/Vaccines - Back in Love City.mp3/212
/var/music/Lightning Seeds - The Life of Riley.mp3/236
/var/music/Foo Fighters, Mark Ronson - Making a Fire (Rx).mp3/262
/var/music/Inhaler - A Night on the Floor.mp3/233
/var/music/Academic - Not Your Summer.mp3/219
```