

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ  
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ  
ТА БЕЗПЕКИ**

**Кафедра інформаційних технологій**

**РОЗРОБЛЕННЯ ЦИФРОВОГО ПРИСТРОЮ НА  
МІКРОКОНТРОЛЕРІ ARM CORTEX-M4 STM32F407VGT6 ДЛЯ  
АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ ПРАЦІВНИКАМИ ПОЛІЦІЇ**

**Кваліфікаційна робота**  
здобувача вищої освіти  
4 курсу денної форми навчання  
**Михайло ОЛІЙНИК**

**Науковий керівник:**  
доцент, кандидат технічних наук\_  
**Ярема КУЛЕШНИК**

**Рецензент:**  
доцент, кандидат технічних наук\_  
\_\_\_\_\_

***Кваліфікаційна робота допущена до захисту***  
«\_\_\_» \_\_\_\_\_ 2026 р., протокол № \_\_\_\_\_

Завідувач кафедри інформаційних технологій  
\_\_\_\_\_ **Олег ЗАЧЕК**  
(підпис)

Львів  
2026

## **ЗМІСТ**

|                                                                                                    |           |
|----------------------------------------------------------------------------------------------------|-----------|
| <b>ВСТУП.....</b>                                                                                  | <b>6</b>  |
| <b>РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....</b>                                                      | <b>9</b>  |
| 1.1. Аналіз сучасних методів та засобів реєстрації акустичних сигналів.....                        | 9         |
| 1.2. Спеціалізована периферія МК STM32 для роботи з аудіоданими.....                               | 19        |
| <b>РОЗДІЛ 2. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ.....</b>         | <b>39</b> |
| 2.1. Відлагоджувальна плата STM32F407G-DISC1 на МК ARM Cortex-M4 STM32F407VGT6.....                | 39        |
| 2.2. Цифровий MEMS-мікрофон MP45DT02.....                                                          | 42        |
| 2.3. Проектування цифрового пристрою аудіофіксації правопорушень.....                              | 46        |
| <b>РОЗДІЛ 3. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ.....</b> | <b>55</b> |
| 3.1. Алгоритм зчитування PDM-даних з MEMS-мікрофона.....                                           | 55        |
| 3.2. Алгоритм обробки звукових даних у реальному часі.....                                         | 62        |
| <b>РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ.....</b>    | <b>70</b> |
| 4.1. Програмний модуль обробки звукових даних з використанням бібліотеки PDM2PCM 70                |           |
| 4.2. Спеціалізований аудіодрайвер для запису та відтворення звуку.....                             | 72        |
| 4.3. Аналіз роботи цифрового пристрою аудіофіксації правопорушень.....                             | 78        |
| <b>ВИСНОВКИ.....</b>                                                                               | <b>89</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....</b>                                                | <b>90</b> |
| <b>ДОДАТКИ.....</b>                                                                                | <b>92</b> |

## СПИСОК УМОВНИХ СКОРОЧЕНЬ

АЗ – апаратне забезпечення системи.

АЦП (Analog-to-Digital Converter) – аналого-цифровий перетворювач; вузол, що трансформує аналоговий сигнал у дискретну цифрову форму.

МК – мікроконтролер.

ПЗ – програмне забезпечення (сукупність кодів та алгоритмів).

ЦАП (Digital-to-Analog Converter) – цифро-аналоговий перетворювач; пристрій для рекуперації аналогового сигналу з цифрового коду.

DMA (Direct Memory Access) – прямий доступ до пам'яті; технологія, що дозволяє периферійним модулям обмінюватися даними з ОЗП без залучення ресурсів центрального процесора.

I2C (Inter-Integrated Circuit) – послідовна шина керування, що використовує дві лінії (даних SDA та синхронізації SCL) для зв'язку між компонентами на платі.

I2S (Inter-IC Sound) – стандарт послідовної шини, оптимізований для передачі цифрових аудіоданих між мікросхемами.

MSC (Mass Storage Class) – протокол передачі даних через інтерфейс USB, що дозволяє операційній системі розпізнавати пристрій як накопичувач (флеш-карту, плеєр тощо).

PCM (Pulse-Code Modulation) – імпульсно-кодова модуляція; метод представлення аналогового сигналу у вигляді послідовності n-бітних значень.

PDM (Pulse-Density Modulation) – модуляція щільності імпульсів; формат передачі аудіосигналу, де амплітуда кодується густиною одиничних бітів у потоці.

SAI (Serial Audio Interface) – універсальний послідовний аудіоінтерфейс для підключення різноманітної звукової периферії до МК.

SPI (Serial Peripheral Interface) – повнодуплексна послідовна шина для високошвидкісного обміну даними між МК та зовнішніми модулями.

WAV (Waveform Audio File Format) – стандартизований формат контейнера для зберігання нестисненого цифрового аудіо.

## АНОТАЦІЯ

**Олійник М., Кулешник Я.** (керівник). Розроблення цифрового пристрою на мікроконтролері ARM CORTEX-M4 STM32F407VGT6 для аудіофіксації правопорушень працівниками поліції. Бакалаврська кваліфікаційна робота. – Львівський державний університет внутрішніх справ, Львів, 2026.

У роботі спроектовано спеціалізований цифровий пристрій для аудіофіксації правопорушень поліцією. Систему побудовано на 32-бітній платформі STM32F4, де обробку сигналів виконує МК STM32F407VGT6. Технічне рішення передбачає використання вбудованого мікрофона MP45DT02 та стереофонічного ЦАП CS43L22 для якісного опрацювання аудіоданих.

Зчитування аудіосигналу з MEMC-мікрофона MP45DT02 забезпечується модулем I2S2, який працює в режимі напівдуплексного ведучого приймача. Це дозволяє не лише приймати PDM-потік, а й керувати частотою дискретизації через подачу синхросигналу на мікрофон. Використання каналу DMA для трансляції вибірок з інтерфейсу I2S2 до буфера оперативної пам'яті дозволило суттєво знизити навантаженість ядра мікроконтролера під час фіксації звуку.

Для відтворення аудіосигналу задіяно інтерфейс I2S3 (режим ведучого передавача), підключений до 24-бітного стерео-ЦАП CS43L22, налаштування якого виконується по шині I2C1. Робота з зовнішніми носіями (USB-флеш-картами) організована через вбудований USB-модуль у режимі Host за протоколом MSC, що забезпечує підтримку формату WAV. Користувацький інтерфейс управління процесами фіксації та відтворення звуку базується на стандартних кнопках плати STM32F4-DISCOVERY.

Розроблене програмне забезпечення базується на стеку технологій HAL, BSP та бібліотеці перетворення PDM2PCM, що дозволяє ефективно опрацьовувати аудіосигнали на платформі STM32F4. Система підтримує два основні режими: запис аудіопотоку з мікрофона MP45DT02 у форматі WAV на USB-накопичувач та відтворення збережених файлів через 24-бітний аудіо-ЦАП CS43L22.

**Ключові слова:** цифровий пристрій аудіофіксації, правоохоронні органи, МК STM32F407VGT6, ПЛАТА STM32F407G-DISC1, MEMC-мікрофон MP45DT02, аудіо-ЦАП CS43L22, РК-дисплей WH1602B-NYG-CT, модуляція PDM, модуляція PCM, STM32 HAL API, бібліотека PDM2PCM, формат WAV.

## ABSTRACT

**Oliynyk M., Kuleshnyk Ya.** (supervisor). Development of a digital device based on the ARM CORTEX-M4 STM32F407VGT6 microcontroller for audio recording of offenses by police officers. Bachelor's thesis. – Lviv State University of Internal Affairs, Lviv, 2026.

The work designs a specialized digital device for audio recording of offenses by the police. The system is built on a 32-bit STM32F4 platform, where signal processing is performed by the STM32F407VGT6 microcontroller. The technical solution involves the use of a built-in MP45DT02 microphone and a CS43L22 stereo digital-to-analog converter for high-quality processing of audio data.

Reading the audio signal from the MP45DT02 MEMS microphone is provided by the I2S2 module, which operates in half-duplex master receiver mode. This allows not only to receive the PDM stream, but also to control the sampling rate by applying a sync signal to the microphone. Using the DMA channel to transfer samples from the I2S2 interface to the RAM buffer significantly reduced the load on the microcontroller core during sound capture.

For audio signal reproduction, the I2S3 interface (master transmitter mode) is used, connected to the 24-bit stereo DAC CS43L22, the configuration of which is performed via the I2C1 bus. Work with external media (USB flash cards) is organized via the built-in USB module in Host mode using the MSC protocol, which provides support for the WAV format. The user interface for controlling the processes of sound capture and playback is based on the standard buttons of the STM32F4-DISCOVERY board.

The developed software is based on the HAL, BSP technology stack and the PDM2PCM conversion library, which allows for efficient processing of audio signals on the STM32F4 platform. The system supports two main modes: recording the audio stream from the MP45DT02 microphone in WAV format to a USB drive and playing back the saved files via the CS43L22 24-bit audio DAC.

**Keywords:** digital audio recording device, law enforcement agencies, STM32F407VGT6 IC, STM32F407G-DISC1 BOARD, MP45DT02 MEMS microphone, CS43L22 audio DAC, WH1602B-NYG-CT LCD display, PDM modulation, PCM modulation, STM32 HAL API, PDM2PCM library, WAV format.

## ВСТУП

Завдяки появі високопродуктивних мікроконтролерів серії Cortex-M4 та компактних цифрових MEMC-сенсорів, стало можливим проектування малогабаритних пристроїв для професійної роботи зі звуком. Ці компоненти дозволяють реалізувати складні алгоритми обробки аудіоданих безпосередньо в реальному часі, що є критично важливим для систем голосового зв'язку, пристроїв ідентифікації мовця та інтерактивних систем управління, що базуються на звукових командах.

Використання мікрофонів у вбудованих системах має багаторічну історію, проте саме поява технології MEMC спричинила стрімке розширення сфер їх застосування. Інтеграція голосового керування та функцій аудіофіксації стала доступною для широкого спектра пристроїв: від систем “розумного будинку” та автомобільної електроніки до мобільних терміналів, портативних комп'ютерів і медіаплеєрів. Крім того, MEMC-сенсори є невід'ємною частиною сучасних технологій VoIP, систем розпізнавання мовлення, освітніх мультимедійних платформ, ігрових консолей, засобів віртуальної реальності, а також професійної та побутової фото- і відеоапаратури.

Перехід до використання цифрових MEMC-сенсорів зумовлений їхньою високою енергоефективністю, малими розмірами та ефективним придушенням шумів. Такі пристрої ідеально адаптовані для роботи з мікроконтролерною периферією завдяки вбудованим цифровим протоколам передачі даних. Це розширює інженерний інструментарій та дозволяє створювати високопродуктивні вбудовані аудіосистеми з мінімальними апаратними витратами.

Вибір між аналоговим та цифровим виходом MEMC-мікрофона є критичним етапом розробки, оскільки кожен інтерфейс має власний набір переваг і недоліків. Успішна реалізація проекту залежить від того, наскільки точно розробник оцінить параметри цих технологій та обере варіант, що найбільш гармонійно інтегрується в систему обробки звуку.

При проектуванні спецзасобу для реєстрації правопорушень на базі мікроконтролера STM32F407VGT6 (ARM Cortex-M4), найбільш раціональним рішенням є використання цифрового MEMS-мікрофона. Застосування інтерфейсу I2S та передача даних у форматі PDM забезпечують оптимальну стійкість до завад. Такі сенсори є ідеальним вибором для професійних аудіосистем, де критичними параметрами є компактність конструкції, стабільність роботи та висока вірність відтворення звуку за умови економічної доступності компонентів.

В основі роботи обраних мікрофонів лежить перетворення звуку в цифровий потік формату PDM. Завдяки використанню апаратної периферії мікроконтролера (інтерфейсів I2S/SPI), ці високочастотні дані надходять на опрацювання блоками. Це дозволяє в реальному часі конвертувати 1-бітні семпли в багаторозрядні аудіодані стандарту PCM, забезпечуючи сумісність із загальноприйнятими алгоритмами обробки звуку.

Мета роботи полягає у проектуванні та технічній реалізації спеціалізованого засобу аудіофіксації для правоохоронних органів. Пристрій забезпечує безперервне захоплення звукового сигналу з цифрового MEMS-сенсора у форматі PDM, його програмне опрацювання та конвертацію у стандарт PCM з наступним збереженням на зовнішній USB-носій.

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

- провести аналіз сучасних методів та засобів реєстрації акустичних сигналів;
- розглянути схемотехнічні та програмні особливості сполучення PDM-мікрофонів із спеціалізованою периферією МК STM32;
- спроектувати цифровий пристрій для аудіофіксації правопорушень працівниками поліції на базі відлагоджувальної плати STM32F407G-DISC1 з МК ARM Cortex-M4 STM32F407VGT6, цифровим MEMS-мікрофоном MP45DT02, 24-бітним стереофонічним ЦАП CS43L22, символьним РК-дисплеєм (16×2), 4-ма кнопками для керування режимами роботи та USB-накопичувачем;

- розробити алгоритм роботи цифрового пристрою для аудіофіксації правопорушень;
- створити програмне забезпечення цифрового пристрою для аудіофіксації правопорушень згідно алгоритму його роботи.
- створити та дослідити тестовий макет цифрового пристрою для аудіофіксації правопорушень.

**Об’єкт дослідження** – процеси збирання та цифрової обробки аудіосигналів у реальному часовому масштабі.

**Предмет дослідження** – сукупність апаратно-програмних методів та технічних засобів, що забезпечують реєстрацію та обробку звукової інформації в режимі реального часу.

**Методи досліджень.** Для досягнення поставленої мети використано методи теоретичного аналізу та цифрової обробки сигналів (метод цифрової фільтрації, метод системного аналізу часових діаграм), експериментальні (натурні) методи дослідження (метод осцилографічного та логічного аналізу інтерфейсів, метод трасування та внутрішньосхемного налагодження (In-Circuit Debugging)), методи дослідження та оптимізації ПЗ (метод подвійної буферизації (Ping-Pong Buffer), метод аналізу та профілювання файлових систем), методи інженерно-технічної верифікації та тестування (метод спектрального аналізу аудіосигналу, стрес-тестування надійності зберігання даних).

**Структура роботи.** Кваліфікаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Обсяг основного тексту роботи складає 89 сторінок, 76 рисунків, 26 таблиць, 2 додатки і 22 бібліографічних джерела. Загальний обсяг роботи – 96 сторінок.

## РОЗДІЛ 1

### СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

#### 1.1. Аналіз сучасних методів та засобів реєстрації акустичних сигналів

На сьогодні виокремлюють два основні підходи до реєстрації звукових сигналів: класичний електретний (ЕСМ) та сучасний мікроелектромеханічний (MEMS). Конструкція ЕСМ-мікрофонів базується на використанні дискретних елементів, зокрема підсилювального транзистора та конденсатора змінного типу, що функціонує завдяки електретному шару. Вихідним продуктом роботи таких пристроїв є аналоговий сигнал, який для коректного використання потребує додаткового відсіювання шумів та підсилення.

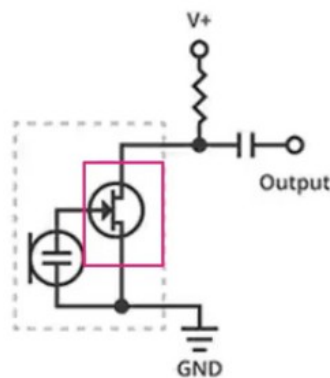


Рис. 1.1. Схематехнічна реалізація конденсаторного мікрофона (ЕСМ-технологія)

Конструктивно MEMS-мікрофон являє собою систему, що інтегрує в собі кремнієву мембрану (чутливий елемент) та спеціалізовану інтегральну схему (ASIC), яка відповідає за обробку сигналів. Залежно від архітектури ASIC, такий пристрій може забезпечувати як аналоговий, так і цифровий формат вихідних акустичних даних (Рис. 1.2).

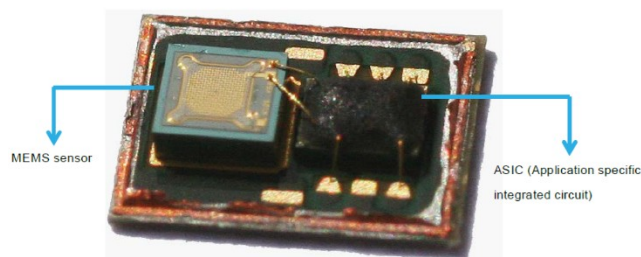


Рис. 1.2. Типова внутрішня архітектура та функціональні вузли MEMS-мікрофона

Функціональна структура MEMS-мікрофонів, що забезпечують формування аналогових та цифрових сигналів, представлена на блок-схемах (Рис. 1.3 та 1.4 відповідно).

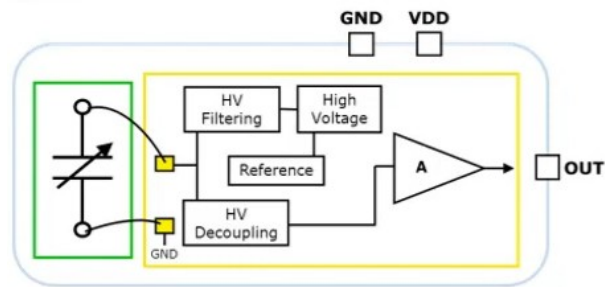


Рис. 1.3. Функціональна схема MEMS-мікрофона з аналоговим інтерфейсом

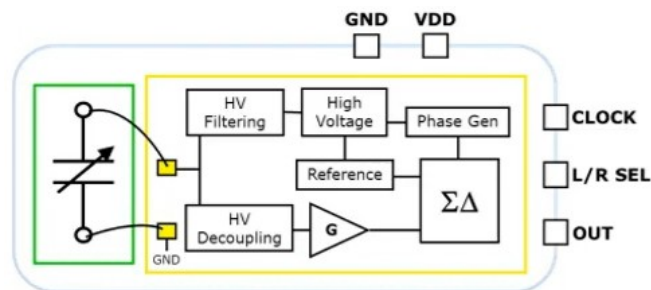


Рис. 1.4. Функціональна схема MEMS-мікрофона з цифровим інтерфейсом

Чутливий елемент, виготовлений за допомогою мікроелектромеханічних технологій, функціонує як кремнієвий конденсатор. Його конструкція передбачає наявність двох кремнієвих обкладок: стаціонарної (виділена зеленим кольором на Рис. 1.5) та еластичної рухомої мембрани (сірого кольору), що реагує на звуковий тиск.

Стаціонарна частина конструкції має електропровідне покриття завдяки нанесеному на неї електроду. Для безперешкодного проходження звукових хвиль у цій поверхні передбачено низку акустичних перфорацій. Своєю чергою, рухома пластина зафіксована лише з одного боку, що забезпечує їй необхідний ступінь вільності для коливань. Наявність спеціального вентиляційного каналу дозволяє компенсувати тиск повітря в тилівій камері, забезпечуючи безперешкодне повернення мембрани у вихідне положення.

Коливання мембрани (діафрагми) всередині камери не лише забезпечують механіко-електричне перетворення, а й у поєднанні з геометрією

самої камери визначають ключові акустичні параметри мікрофона. Зокрема, об'єм камери та рухливість діафрагми безпосередньо впливають на амплітудно-частотну характеристику (АЧХ) пристрою та його чутливість, що відображається на підсумковому співвідношенні сигнал/шум (SNR).

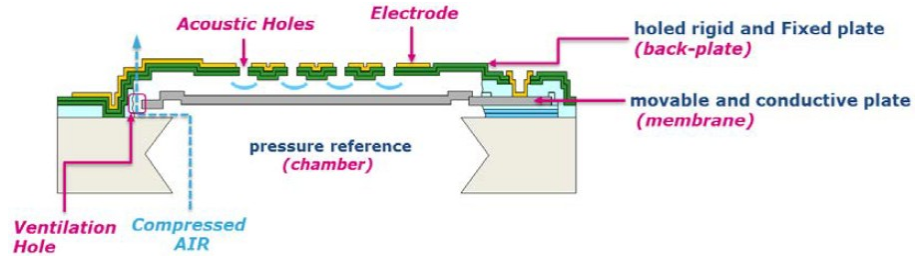


Рис. 1.5. Конструктивні елементи чутливого елемента MEMS-мікрофона

Функціонування чутливого елемента MEMS-мікрофона ґрунтується на роботі змінного конденсатора. Під впливом акустичного тиску рухома мембрана змінює свою відстань відносно стаціонарної задньої пластини, що призводить до відповідної модуляції електричної ємності системи (Рис. 1.6).

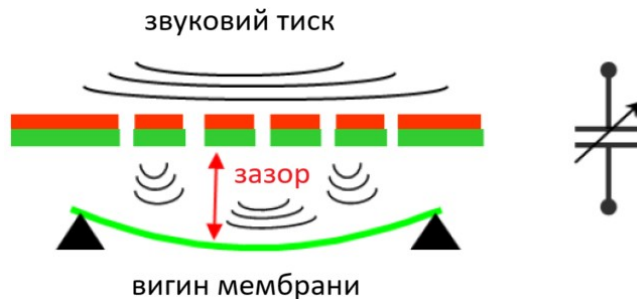


Рис. 1.6. Механізм модуляції електричної ємності чутливого елемента MEMS-перетворювача

Обробка сигналів у MEMS-мікрофоні здійснюється спеціалізованою КМОН-схемою, яка трансформує ємнісні коливання датчика у зручний для подальшої обробки формат: аналогову напругу або цифровий PDM-сигнал. Компонування модуля залежить від розміщення звукового каналу; за цією ознакою розрізняють моделі з верхнім (Upper port) або нижнім (Lower port) розташуванням вхідного отвору в корпусі.

Для покращення експлуатаційних характеристик своїх виробів STMicroelectronics розробила специфічний тип корпусування, що реалізований у цифрових MEMS-мікрофонах (наприклад, серії MP45DT02-M). Конструкція з верхнім розташуванням порту (Top Port) передбачає монтаж

ASIC-контролера під акустичним входом, де він екранується від світла спеціальним купольним покриттям. Чутливий MEMS-елемент встановлюється поруч із ASIC для мінімізації довжини з'єднань та зниження рівня завад.

Архітектура мікрофона включає два кремнієві компоненти на підкладці з контактною групою на нижній стороні. Резонансні камери розрізняють за їх розміщенням відносно напрямку звуку: передня камера формується об'ємом корпусу, а задня – внутрішнім простором за мембраною сенсора. Попри перевагу у вигляді захищеності від пилу, така конфігурація має суттєві недоліки – підвищений рівень шумів та наявність небажаних резонансних піків на частотній характеристиці.

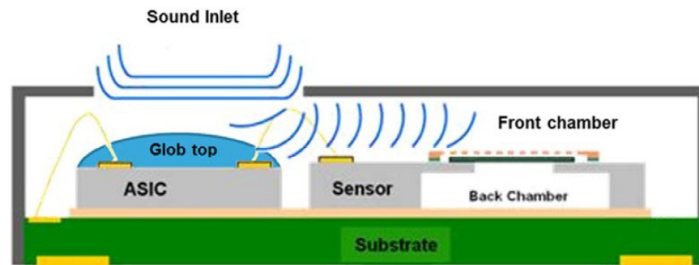


Рис. 1.7. Конструктивне виконання корпусу типу 4x5 для цифрового MEMS-мікрофона моделі MP45DT02

Використання цифрових MEMS-мікрофонів є доцільним у розробках, де критично важливими є компактність, якість звукозапису та довговічність. Можливість групування кількох сенсорів у обмеженому просторі без втрати якості сигналу розширює функціональні можливості аудіосистем у промисловому та побутовому секторах, забезпечуючи високу завадостійкість пристрою.



Рис. 1.8. Зовнішній вигляд популярних модулів I2S MEMS-мікрофонів: від серії SPH0645LM4H до компактних рішень типу SPW2430

У структурі цифрового MEMS-сенсора акустичний сигнал перетворюється на цифрову послідовність безпосередньо на рівні датчика. Взаємодія з мікроконтролером STM32 реалізується шляхом передачі необроблених даних на апаратні периферійні інтерфейси контролера. Кінцевим етапом є цифрова обробка сигналу (DSP) всередині STM32, що дозволяє отримати звуковий потік у загальноприйнятому стандарті.

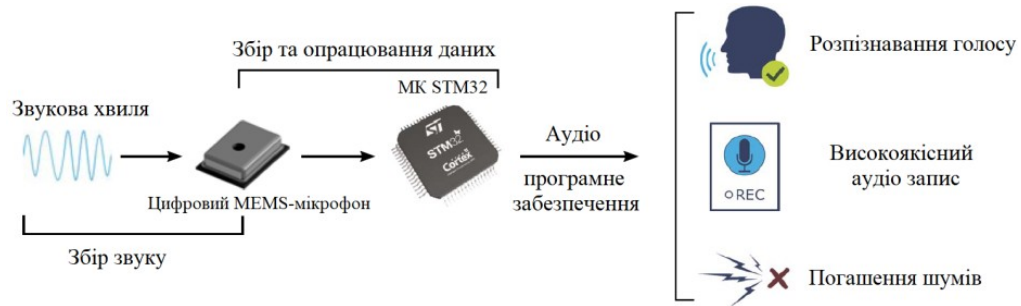


Рис. 1.9. Схема перетворення акустичних коливань у вихідні формати звукових даних

На Рис. 1.10 наведено блок-схему цифрового MEMS-мікрофона з підтримкою інтерфейсу PDM. Основними конструктивними блоками пристрою є чутливий MEMS-елемент, блоки попереднього підсилення та цифровий контур модуляції імпульсної щільності, що забезпечує формування вихідного звукового потоку.

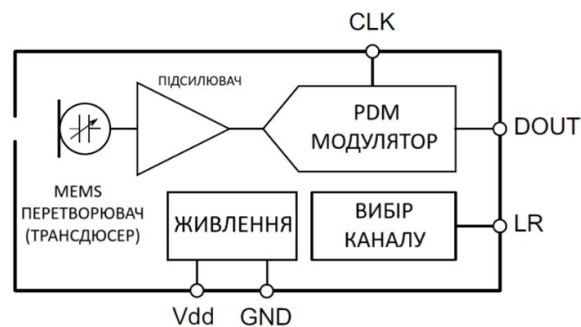


Рис. 1.10. Схема архітектури цифрового MEMS-мікрофона з інтегрованим PDM-модулятором

MEMS-трансдюсер реалізований як змінний конденсатор, що генерує напругу пропорційно до звукового тиску. Інтегрований підсилювач здійснює узгодження імпедансу та буферизацію сигналу, після чого підсилена амплітуда надходить на PDM-модулятор для цифрового кодування.

Процес імпульсно-щільнісної модуляції (PDM) передбачає кодування амплітуди підсиленого сигналу в часову щільність одиничних бітів. Модулятор функціонує під керуванням сигналу CLK, який задає темп дискретизації. Згідно з технічними специфікаціями STMicroelectronics, підтримуваний робочий діапазон тактування для таких пристроїв знаходиться в межах 1...3,25 МГц. Тактова частота CLK безпосередньо встановлює крок дискретизації для аналого-цифрового перетворення всередині мікрофона. Саме з цією швидкістю здійснюється вибірка амплітуди підсиленого сигналу для генерації вихідного PDM-потіку, що забезпечує високу точність часового представлення звуку. Цифровий інтерфейс мікрофона видає дані лише на одному з фронтів тактового сигналу, перемикаючись у високоімпедансний стан на решту періоду. Таке часове розділення дозволяє двом мікрофонам працювати на одній лінії даних. Конкретний активний фронт визначається конфігурацією виводу LR, який замикається на Vdd або

*Таблиця 1.1. Залежність активності вихідного інтерфейсу від фази тактового сигналу*

| Конфігурація LR  | Фаза CLK Low | Фаза CLK High | Призначення |
|------------------|--------------|---------------|-------------|
| Логічний 0 (GND) | Data Out     | High-Z        | Канал А     |
| Логічна 1 (Vdd)  | High-Z       | Data Out      | Канал В     |

Для коректного функціонування внутрішніх компонентів MEMS-сенсора використовується живлення через виводи Vdd (позитивний полюс) та GND (загальний провід). Належна фільтрація ліній живлення є обов'язковою умовою для пригнічення пульсацій, які можуть трансформуватися у фазовий дрижак (джитер) або додатковий шум у вихідному PDM-потіці.

*Таблиця 1.2. Опис інтерфейсних сигналів цифрового MEMS-мікрофона*

| Назва | Напрямок | Призначення  | Опис                                                       |
|-------|----------|--------------|------------------------------------------------------------|
| Vdd   | Input    | Power Supply | Напруга живлення цифрових та аналогових вузлів             |
| GND   | Input    | Ground       | Спільний вивід (земля)                                     |
| LR    | Input    | L/R Select   | Конфігурація фронту видачі даних (GND - Left, Vdd - Right) |
| CLK   | Input    | Clock        | Сигнал синхронізації для роботи модулятора                 |
| DOUT  | Output   | PDM Data     | Вихідний цифровий сигнал з тристабільною логікою           |

*Підключення цифрового MEMS-мікрофона у монорежимі.* Конфігурація мікрофона для роботи в режимі “моно” передбачає подачу потенціалу живлення Vdd на вхід вибору каналу LR, як це продемонстровано на Рис. 1.4.

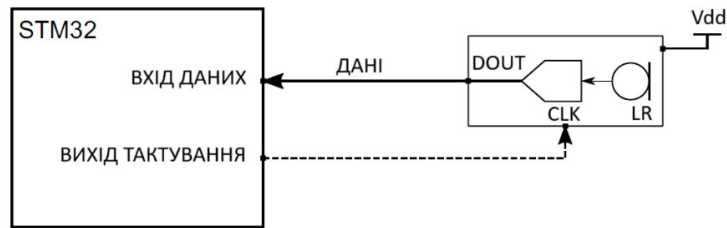


Рис. 1.11. Конфігурація мікрофона для монорежиму: передача даних по каналу Right

Синхронізація видачі даних здійснюється по передньому фронту CLK. Протягом першої півхвилі тактового імпульсу мікрофон транслює інформаційний біт, а впродовж наступної – перемикається в режим високого опору. Це забезпечує часове розділення (TDM) для роботи кількох пристроїв на спільній шині.

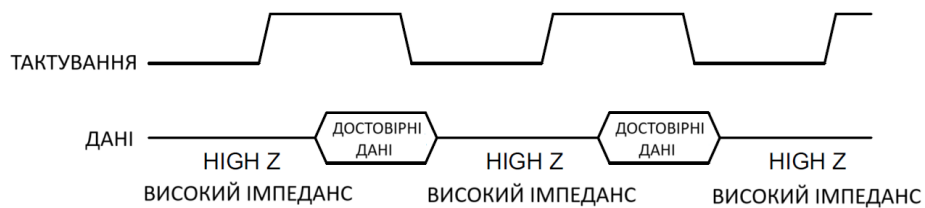


Рис. 1.12. Часова діаграма формування вихідних аудіоданих MEMS-сенсора для правого каналу (Right Channel)

Активація передачі даних по лівому каналу здійснюється шляхом подачі низького логічного рівня на пін LR, що реалізується його підключенням до GND.

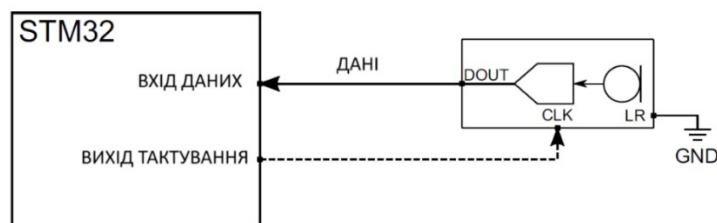


Рис. 1.13. Схема монофонічної трансляції даних із використанням лівого аудіоканалу (Left Channel)

Трансляція вихідного сигналу здійснюється синхронно з заднім фронтом синхросигналу в межах першої половини такту. Далі, протягом наступної півосі періоду, пристрій переводиться у режим високого опору, звільняючи інформаційну шину.



Рис. 1.14. Часова діаграма формування вихідних аудіоданих MEMS-сенсора для лівого каналу (Left Channel)

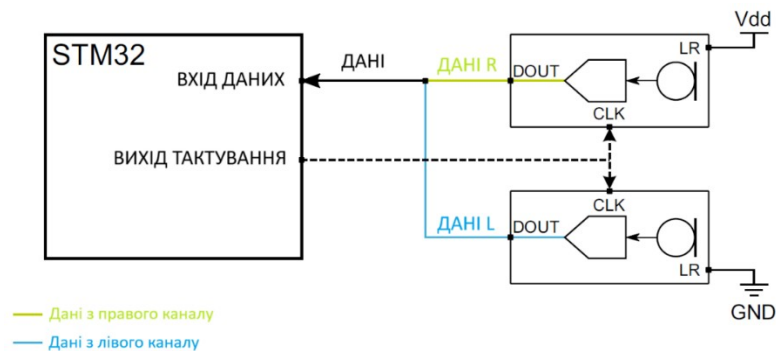


Рис. 1.15. Схема стереофонічного підключення з використанням спільної шини даних

У стереофонічній конфігурації два MEMS-мікрофони використовують спільну шину даних. Поділ у часі досягається за рахунок прив'язки до різних фронтів синхронізації: пристрій із високим потенціалом на піні LR ( $V_{dd}$ ) видає дані на висхідному фронті тактового сигналу. Водночас мікрофон із заземленим піном LR (GND) формує вихідний потік на спадному фронті.

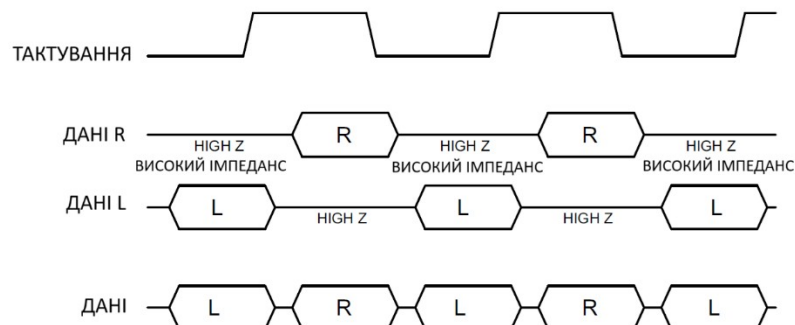


Рис. 1.16. Часова діаграма мультиплексування даних лівого та правого каналів у стереорежимі

*Принцип формування PDM-сигналу.* Модуляція щільності імпульсів (PDM) кодує аналоговий аудіосигнал у швидкісний 1-бітний цифровий потік. Амплітуда вихідної хвилі безпосередньо впливає на частоту появи імпульсів у часі. Так, висока густина одиниць у кадрі відповідає додатній півхвилі сигналу, зосередження нулів – від’ємній, а стан, коли кількість нулів та одиниць є однаковою (еквідистантне чергування), вказує на відсутність сигналу (нульову точку).

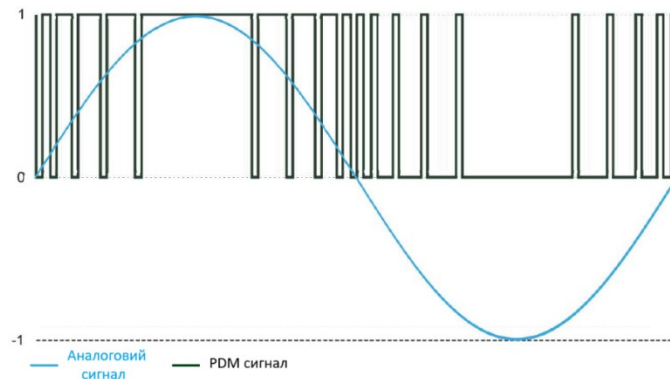


Рис. 1.17. Принцип кодування амплітуди в щільність імпульсів PDM-потіку

*Принцип цифрового кодування PCM.* Метод PCM полягає у представленні аналогової хвилі у вигляді цифрових значень фіксованої довжини. Якість результуючого цифрового сигналу та ступінь його відповідності оригіналу визначаються тактовою частотою взяття відліків і розрядністю квантування, яка задає кількість можливих рівнів амплітуди.

Під частотою дискретизації розуміють інтенсивність взяття дискретних значень із безперервного сигналу за секунду при його переході в цифровий еквівалент. Квантування за рівнем при цьому характеризується розрядністю — параметром, який задає точну кількість бітів, виділених на опис кожного окремого відліку.

Переведення однобітного потоку PDM у PCM-відліки реалізується через алгоритми низькочастотної фільтрації та подальшого зменшення темпу передачі даних. Під час цього етапу (децимації) надвисока частота тактування мікрофона ділиться на відповідний коефіцієнт, щоб досягти кінцевої робочої частоти дискретизації звуку, як-от 8 кГц.

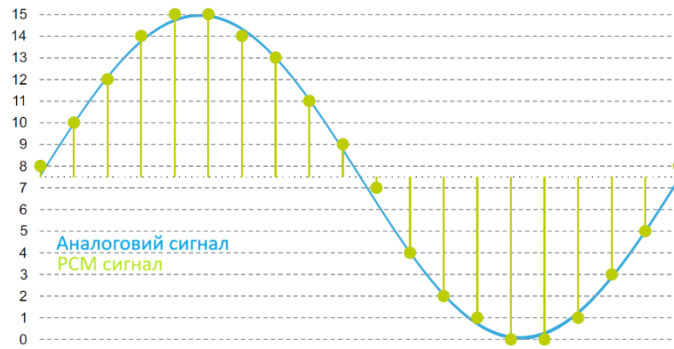


Рис. 1.18. Графічне представлення процесу квантування та дискретизації сигналу в PCM-форматі

Зниження темпу передачі даних шляхом збереження лише кожного  $M$ -го відліку призводить до зменшення частоти дискретизації в  $M$  разів. Це означає, що первинна частота PDM-потoku (яка задається генератором синхросигналу мікрофона) є кратною кінцевій частоті аудіо за коефіцієнтом  $M$ , що в теорії ЦОС (цифрової обробки сигналів) називається фактором проріджування.

Базова частота тактування PDM-інтерфейсу є кратною цільовій частоті дискретизації звуку з фактором масштабування  $M$ . Значення цього коефіцієнта проріджування зазвичай обирається в діапазоні 48...128. Для захисту корисного сигналу від деструктивного впливу спектрального перекриття (спектрального накладання) перед етапом зниження частоти обов'язково застосовується цифровий фільтр нижніх частот (ФНЧ).

*Акустичні параметри MEMS-мікрофонів.* Параметр чутливості відображає відношення вихідного цифрового коду (у шкалі dBFS) до вхідної акустичної дії. Стандартна методика вимірювання передбачає подачу на вхід мікрофона опорного тонального сигналу частотою 1 кГц із рівнем звукового тиску 94 дБ (або абсолютним тиском 1 Па). Відношення сигнал/шум (SNR) відображає різницю (у децибелах) між еталонним тоном із рівнем звукового тиску 94 дБ на частоті 1 кГц та шумовою полицею на виході пристрою за відсутності корисного сигналу. Високе значення цього параметра є критичним для забезпечення високої якості передачі голосу та мінімізації фонових завад. Точка акустичного перевантаження (AOP) відображає максимальну амплітуду вхідного акустичного сигналу, за якої вихідний сигнал мікрофона залишається в межах нормованого спотворення. Зазвичай виробники регламентують рівень

AOP як поріг, на якому сумарні гармонічні спотворення досягають 10%. Коефіцієнт придушення нестабільності живлення (PSRR) характеризує стійкість мікрофона до коливань напруги живлення. Він показує, наскільки ефективно внутрішня схемотехніка пристрою фільтрує зовнішні низькочастотні та високочастотні завади, що надходять від блоку живлення.

*Таблиця 1.2. Переваги використання цифрових MEMS-мікрофонів*

| Особливість                    | Перевага                                                                                                                                                                                                                                                     |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Електромагнітна стійкість      | зниження складності проектування та апаратної інтеграції.                                                                                                                                                                                                    |
| Прямий цифровий вихід          | спрощення розробки цифрових звукових трактів;<br>безпосередній зв'язок із МК без проміжних АЦП;<br>передача стереосигналу по одній лінії зв'язку;<br>економія корисного простору друкованої плати;<br>висока гнучкість при масштабуванні кількості сенсорів. |
| Стабільність цифрової передачі | довільне розміщення мікрофонів у кінцевій системі;<br>застосування уніфікованих методів обробки даних;<br>спрощення інтеграції алгоритмів покращення звуку та ехопоглинання.                                                                                 |

*Таблиця 1.3. Параметри цифрових MEMS-мікрофонів STMicroelectronics*

| Модель      | Розташування порта | SNR, дБ | AOP, дБ SPL | Особливості/Примітки                  |
|-------------|--------------------|---------|-------------|---------------------------------------|
| MP34DB02    | Нижня грань        | 62,6    | 120,0       | Єдина модель у списку з нижнім портом |
| MP34DT01-M  | Верхня грань       | 61,0    | 120,0       | Оптимізовано для мобільних пристроїв  |
| MP34DT02    | Верхня грань       | 60,0    | 120,0       | Базове рішення                        |
| MP34DT04    | Верхня грань       | 64,0    | 120,0       | Покращене відношення сигнал/шум       |
| MP34DT04-C1 | Верхня грань       | 64,0    | 120,0       | Модифікація з підвищеною стабільністю |
| MP34DT05    | Верхня грань       | 64,0    | 122,5       | Найвища стійкість до перевантажень    |
| MP45DT02-M  | Верхня грань       | 61,0    | 120,0       | Збільшений розмір корпусу             |

## **1.2. Спеціалізована периферія МК STM32 для роботи з аудіоданими**

Сімейство 32-розрядних мікроконтролерів STM32 володіє розвиненою апаратною базою для захоплення та обробки звуку в реальному часі. Висока обчислювальна здатність ядра ARM Cortex дозволяє успішно інтегрувати цифрові MEMS-мікрофони. У цьому підрозділі розглянуто схемотехнічні та програмні особливості сполучення PDM-мікрофонів із вбудованими модулями SPI/I2S, SAI та блоком DFSDM для зчитування звуку в режимах

“моно” та “стерео”. Периферія SPI в архітектурі STM32 є універсальною: більшість таких модулів здатна працювати в режимі I2S для трансляції цифрового звуку. Існує два покоління апаратної реалізації даного блоку. Попередній стандарт отримав маркування SPI-V1, тоді як модернізований варіант позначається як SPI-V2. Останній забезпечує розширений функціонал і наразі доступний у флагманських кристалах серії STM32H7x. Апаратний модуль SPI підтримує підключення до двох цифрових мікрофонів у моно- чи стереорежимі за інтерфейсами SPI або I2S: 1. Інтерфейс SPI слугує класичним каналом зв'язку для простої та надійної взаємодії мікроконтролера з периферією. 2. Інтерфейс I2S спеціально оптимізований для роботи зі звуком (як для відтворення через зовнішні цифро-аналогові перетворювачі, так і для оцифрування акустичного сигналу з датчиків).

Монофонічний режим передбачає роботу МК з одним цифровим MEMС-сенсором. Незалежно від обраного протоколу (SPI чи I2S), відповідний апаратний блок МК налаштовується як ініціатор обміну (Master Receiver). Мікроконтролер бере на себе роль джерела тактового сигналу для лінії CLK мікрофона. У відповідь на це мікрофон передає бітовий потік звукових даних зі свого виходу DOUT на вхідні лінії МК (Рис. 1.19).

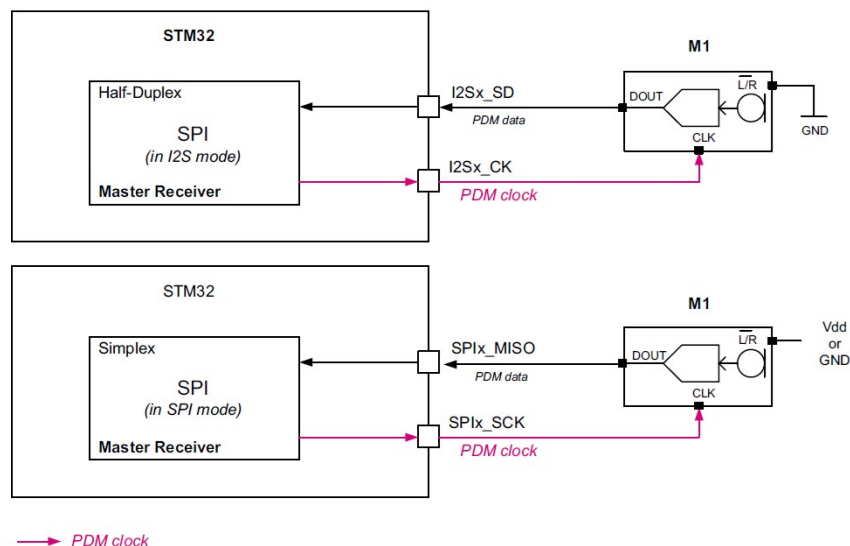


Рис. 1.19. Організація монофонічного аудіоканалу при сполученні цифрового мікрофона з інтерфейсами SPI та I2S у режимі Master Receiver

Апаратне налаштування каналу в інтерфейсі SPI здійснюється шляхом подачі на вивід L/R високого чи низького логічного рівня. Фаза тактування модуля SPI повинна відповідати цій прив'язці:

- Якщо на контакт L/R подано потенціал GND, контролер здійснює захоплення бітів із лінії DOUT по передньому (наростаючому) фронту імпульсу SPIx\_SCK;
- Якщо контакт L/R підтягнутий до  $V_{dd}$ , вибірка даних із виходу DOUT мікрофона виконується по задньому (спадаючому) фронту сигналу тактування SPIx\_SCK.

Для забезпечення стандартного режиму I2S контакт L/R цифрового MEMC-сенсора зазвичай з'єднують із потенціалом GND. Базовий алгоритм шини I2S виконує захоплення інформаційних бітів по передньому (наростаючому) фронту імпульсів I2Sx\_SCK. Важливою архітектурною перевагою оновленого блоку SPI-V2 є можливість програмного керування фазою тактування, що дозволяє перевизначити активний фронт зчитування.

Зчитування прийнятих аудіовідліків із модуля SPI (I2S) та їх подальше збереження в ОЗП реалізується двома шляхами: за допомогою апаратного контролера DMA або шляхом обробки відповідних переривань (IRQ). При цьому в приймальний регістр даних SPIx\_DR безперервно завантажується послідовність бітів із цифрового мікрофона. Приклад структури такого потоку для 16-бітного кадру виглядає наступним чином:

|          |                   |                     |                     |                     |                     |     |                      |
|----------|-------------------|---------------------|---------------------|---------------------|---------------------|-----|----------------------|
| Bit:     | 15                | 14                  | 13                  | 12                  | 11                  | ... | 0                    |
| Content: | M1_b <sub>N</sub> | M1_b <sub>N+1</sub> | M1_b <sub>N+2</sub> | M1_b <sub>N+3</sub> | M1_b <sub>N+4</sub> | ... | M1_b <sub>N+15</sub> |

У наведеному форматі символом M1\_bxx позначено інформаційні біти першого мікрофона, а M1\_bN є старшим (значущим) бітом. Апаратна логіка дозволяє змінювати черговість вибірки бітів у кадрі: за замовчуванням першим транслюється старший біт, проте в налаштуваннях можна вказати пріоритет молодшого біта (LSB). Вбудована периферія також гнучко адаптується під різну довжину інформаційного слова.

*Стереофонічний режим.* Підключення двох цифрових MEMC-мікрофонів до модуля SPI для формування стереосигналу потребує залучення додаткового таймера (Рис. 1.20). Незалежно від обраного протоколу (SPI чи I2S), відповідна периферія STM32 ініціалізується як Master Receiver. Особливістю стереофонічного режиму є генерація інтерфейсом SPI подвоєної тактової частоти. Це необхідно для коректної дискретизації та фіксації бітових потоків з обох мікрофонів по задньому (спадному) фронту імпульсів.

Для оптимізації схемотехніки два мікрофони об'єднуються по одній шині даних. Модуль SPI виступає первинним джерелом тактування для внутрішнього таймера, який зменшує частоту вдвічі (генеруючи вихідні імпульси SPIx\_SCK/I2Sx\_CK). Вже цей поділений сигнал синхронізує роботу самих MEMC-мікрофонів. Захоплення інформаційного потоку здійснюється блоком I2S з виводів DOUT підключених датчиків.

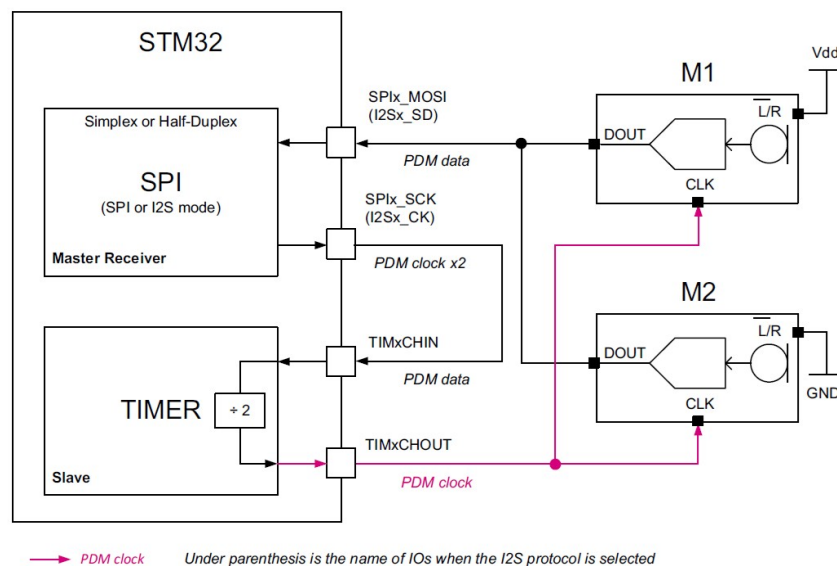


Рис. 1.20. Організація стереофонічного аудіоканалу на базі двох мікрофонів та інтерфейсу SPI (I2S) із діленням тактової частоти

*Генерація синхросигналу за допомогою таймера.* При тактуванні цифрових MEMS-мікрофонів від внутрішнього таймера слід звернути увагу на такі аспекти:

- Алгоритм роботи повинен враховувати фазові зсуви та затримки, що вносяться подільником частоти таймера. Важливо, щоб результуюче вік-

но забезпечувало необхідний часовий запас ( $T_S$ ) для стабілізації та зчитування бітових даних, які транслює сенсор.

- Граничний час затримки ( $T_D$ ), що виникає в тракті таймера між входом ( $TIMx\_CHIN$ ) та виходом ( $TIMx\_CHOUT$ ), дорівнює 5 тактам його опорного сигналу. Як базове тактування для таймера зазвичай виступає шина APB або вихід відповідного помножувача частоти (Рис. 1.13).

Для коректної роботи системи необхідно забезпечити повну синхронізацію опорних сигналів тактування для генератора на вході  $TIMx\_CHIN$  та залученого таймера. У разі неузгодженості цих частот на виході формуватимуться імпульси з нестабільним періодом (динамічні спотворення часової сітки). Це негативно вплине на процес сигма-дельта модуляції всередині цифрового мікрофона та знизить співвідношення сигнал/шум (SNR).

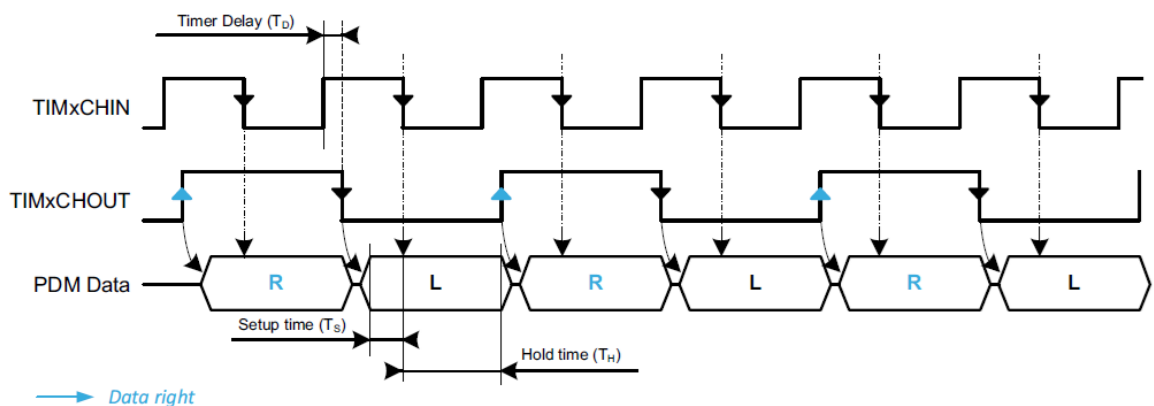


Рис. 1.21. Часова діаграма функціонування інтерфейсу в стереофонічному режимі (з часовим розділенням каналів)

Для збереження прийнятих вибірок (семплів) в ОЗП використовується периферія SPI/I2S у поєднанні з контролером DMA або обробкою переривань. Особливістю стереорежиму є побітове чергування даних, що надходять від лівого та правого каналів. У результаті вміст 16-бітного регістра даних  $SPIx\_DR$  набуває наступного вигляду.

|          |           |           |               |               |               |     |               |
|----------|-----------|-----------|---------------|---------------|---------------|-----|---------------|
| Bit:     | 15        | 14        | 13            | 12            | 11            | ... | 0             |
| Content: | $M1\_b_N$ | $M2\_b_N$ | $M1\_b_{N+1}$ | $M2\_b_{N+1}$ | $M1\_b_{N+2}$ | ... | $M2\_b_{N+7}$ |

Для ідентифікації бітів у потоці використовуються такі позначення:  $M1\_b_{xx}$  – масив бітів аудіоданих першого цифрового мікрофона ( $M1\_b_N$  –

старший розряд).  $M2\_b_{xx}$  – масив бітів даних другого цифрового мікрофона ( $M2\_b_N$  – старший розряд).

Апаратні блоки  $SPI\_V1$  та  $SPI\_V2$  забезпечують гнучкість у виборі формату даних. Наприклад, послідовність збереження бітів для прийнятих семплів інвертується шляхом перемикання інтерфейсу з режиму MSB (Most Significant Bit first) у режим LSB (Least Significant Bit first). Водночас критично важливо враховувати, що чергування бітових потоків від сенсорів  $M1$  та  $M2$  у результуючому слові завжди залишається послідовним.

Подальша обробка прийнятого сигналу, включаючи перетворення PDM-потoku в PCM-формат, вимагає попереднього розділення даних обох каналів. Для цього застосовується програмний алгоритм демультіплексування (деінтерлейсингу), який відновлює послідовність відліків для кожного мікрофона окремо.

*Функціональні особливості інтерфейсу SAI.* Модуль SAI в мікроконтролерах STM32 є універсальним рішенням для підключення широкого спектра аудіопериферії. Він складається з двох самостійних блоків, які мають роздільні джерела тактування та можуть працювати незалежно один від одного. Завдяки наявності розширеного функціонала, деякі ревізії SAI підтримують пряме зчитування PDM-потоків, забезпечуючи підключення мікрофонних масивів місткістю до 8 одиниць на одну шину.

*Монофонічна конфігурація на базі одного блоку SAI.* Для зчитування даних з одного цифрового мікрофона використовується поодинокий підблок інтерфейсу SAI (Рис.1.22). Модуль ініціалізується в режимі Master Receiver, беручи на себе функцію джерела тактування. Мікрофон, своєю чергою, синхронізує видачу PDM-вибірок через вивід DOUT безпосередньо на шину даних  $SAI\_SD\_x$  приймача.

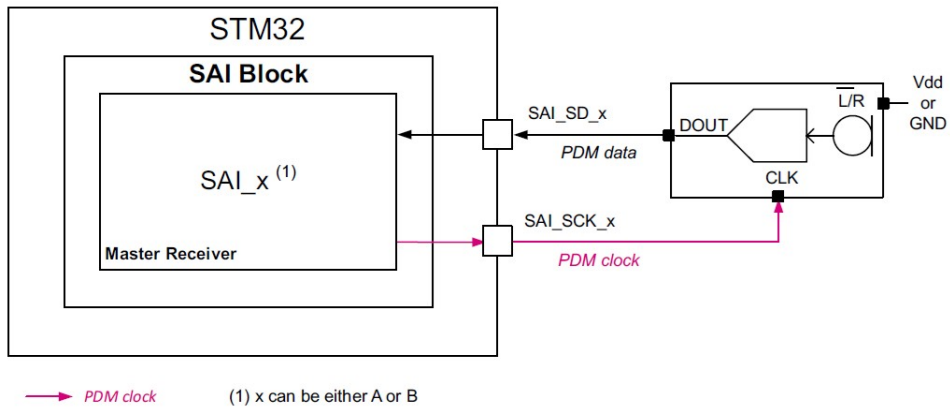


Рис. 1.22. Схема підключення поодинокого цифрового MEMS-мікрофона до підблоку інтерфейсу SAI в монофонічному режимі

Конфігурування виводу L/R шляхом його з'єднання з VDD або GND визначає активний фронт (передній чи задній), за яким мікрофон транслює PDM-потік. Для коректного зчитування параметри синхронізації підблоку SAI мають бути чітко синхронізовані з логікою роботи сенсора. Оскільки підблоки SAI є незалежними, другий канал можна паралельно використовувати для виведення аудіосигналу на зовнішній ЦАП.

Для збереження прийнятих аудіовибірок в ОЗП використовується периферія SAI у поєднанні з каналами DMA або вектором переривань IRQ. Регістр даних інтерфейсу (SAI\_xDR) забезпечує фіксацію неперервного потоку PDM-бітів, що надходять від цифрового мікрофона. Організація даних у регістрі має такий вигляд.

|          |                   |                     |                     |                     |                     |     |                     |
|----------|-------------------|---------------------|---------------------|---------------------|---------------------|-----|---------------------|
| Bit:     | k                 | k-1                 | k-2                 | k-3                 | k-4                 | ... | 0                   |
| Content: | M1_b <sub>N</sub> | M1_b <sub>N+1</sub> | M1_b <sub>N+2</sub> | M1_b <sub>N+3</sub> | M1_b <sub>N+4</sub> | ... | M1_b <sub>N+k</sub> |

Для ідентифікації бітів у буфері використовуються такі позначення: M1\_bxx – біти аудіоданих першого цифрового мікрофона. M1\_bN – старший біт у регістрі даних SAI\_ADR/BDR. Якщо в полі конфігурації розміру даних DS[2:0] встановлено значення 010 (8 біт), то найвищий індекс біта k=7. Підсумковий розмір слова в регістрах даних жорстко прив'язаний до налаштувань поля DS[2:0], а для коректної роботи біти поля SLOTSZ[1:0] мають бути скинуті в 00.

Слід враховувати, що порядок проходження бітів для отриманих відліків змінюється на протилежний при перемиканні режиму з MSB (старшим розрядом уперед) на LSB (молодшим розрядом уперед). Незалежно від обраного порядку бітів, вміст регістрів даних SAI\_ADR/SAI\_BDR завжди вирівнюється по правому краю.

*Стереоконфігурація на одному блоці SAI.* Для побудови стереосистеми два цифрові мікрофони підключаються до спільного підблоку SAI (рис. 1.16). Модуль ініціалізується як Master Receiver та працює на подвоєній частоті відносно частоти дискретизації мікрофонів. Таке рішення дає змогу апаратному блоку SAI фіксувати по черзі біти від кожного мікрофона на одному й тому самому активному фронті генератора, реалізуючи часове мультиплексування на одній сигнальній лінії.

Для побудови стереосистеми два цифрових мікрофони комутують із поодиноким підблоком SAI. Ініціалізація модуля в режимі Master Receiver та двократне підвищення його частоти тактування відносно PDM-потoku дозволяють апаратно фіксувати біти обох каналів на одному активному фронті генератора. Завдяки цьому досягається безконфліктне часове мультиплексування даних на спільній шині.

Для синхронізації системи тактовий сигнал SAI\_SCK\_x із виходу SAI подається на внутрішній апаратний таймер-подільник частоти на 2. Отриманий сигнал надходить на вивід CLK мікрофона. Оскільки підблоки SAI є незалежними, другий канал можна використовувати паралельно для виведення аудіо на зовнішній ЦАП. Процеси обміну даними у цьому режимі також описуються часовою діаграмою на Рис. 1.21.

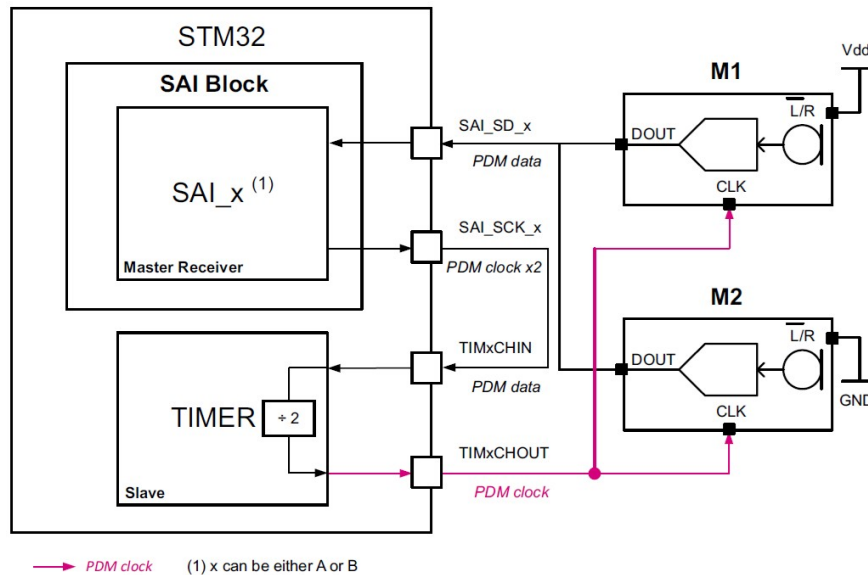


Рис. 1.23. Стереопідключення MEMS-мікрофонів до одного модуля SAI з апаратним подвоєнням частоти опитування через таймер

Для перенесення отриманих семплів в ОЗП використовується периферійний модуль SAI у зв'язці з каналами DMA або вектором переривань. Особливістю стереорежиму є побітове чергування даних, що надходять від лівого та правого каналів. У результаті вміст регістрів даних SAI\_ADR/SAI\_BDR набуває наступного вигляду.

|          |                   |                   |                     |                     |     |                           |                           |
|----------|-------------------|-------------------|---------------------|---------------------|-----|---------------------------|---------------------------|
| Bit:     | k                 | k-1               | k-2                 | k-3                 | ... | 1                         | 0                         |
| Content: | M1_b <sub>N</sub> | M2_b <sub>N</sub> | M1_b <sub>N+1</sub> | M2_b <sub>N+1</sub> | ... | M1_b <sub>N+(K-1)/2</sub> | M2_b <sub>N+(K-1)/2</sub> |

Для ідентифікації бітів у буфері використовуються такі позначення: M1\_b<sub>xx</sub> – біти даних першого PDM-мікрофона (M1\_b<sub>N</sub> – старший біт). M2\_b<sub>xx</sub> – біти даних другого PDM-мікрофона (M2\_b<sub>N</sub> – старший біт). k – порядковий номер розряду в регістрі SAI\_ADR/SAI\_BDR. При конфігурації поля DS[2:0] значенням 100 (розмір слова – 16 бітів), індексація починається з k=15.

Підсумковий розмір слова в регістрах даних жорстко прив'язаний до налаштувань поля DS[2:0]. Додатковою вимогою є те, що керуючі біти поля SLOTSZ[1:0] мають бути скинуті в 00.

Слід враховувати, що порядок проходження бітів для отриманих відліків змінюється на протилежний при перемиканні режиму з MSB на LSB. Важливою умовою є те, що бітові потоки від сенсорів M1 та M2 у результуючому

слові завжди зберігають свою циклічність чергування. Самі ж дані в регістрах SAI\_ADR/SAI\_BDR завжди вирівнюються по правому краю розрядної сітки.

Подальша обробка прийнятого сигналу, включаючи перетворення PDM-потoku в PCM-формат, вимагає попереднього розділення даних обох каналів. Для цього застосовується програмний алгоритм демультимплексування (деінтерлейсингу), який відновлює послідовність відліків для кожного мікрофона окремо.

Якщо апаратний блок SAI не має інтегрованого таймера, зчитування даних з двох PDM-мікрофонів реалізується шляхом синхронізації його підблоків. Мікрофони підключаються індивідуально до кожного з них. Перший підблок конфігурується як провідний приймач (Master Receiver) і бере на себе генерацію тактового сигналу, тоді як другий переводиться в режим веденого приймача (Synchronous Slave Receiver) та використовує готову тактову частоту.

У цьому режимі підблок SAI (Master Receiver) виступає джерелом синхронізації, формуючи опорну частоту для обох мікрофонів та другого підблоку SAI. Завдяки спільній тактовій лінії забезпечується абсолютно синхронне паралельне зчитування PDM-відліків двома апаратними вузлами.

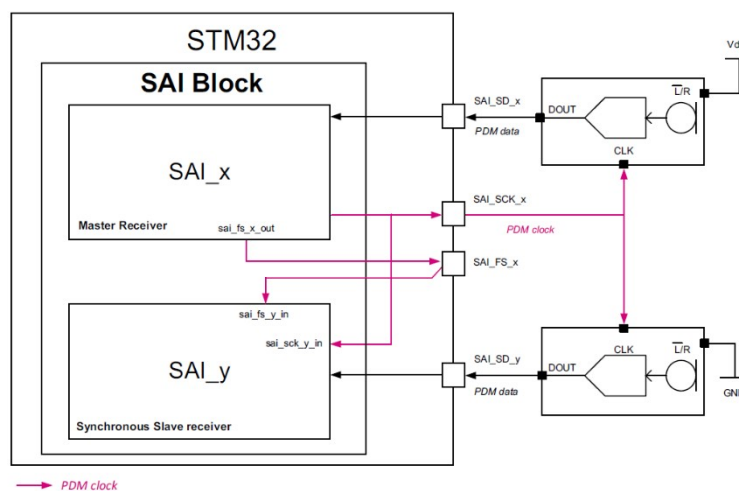


Рис. 1.24. Стереопідключення MEMS-мікрофонів через паралельну синхронізацію підблоків інтерфейсу SAI

Зважаючи на те, що мікрофони працюють на роздільних сигнальних лініях, стан виводів вибору каналу L/R ( $V_{DD}$  або  $GND$ ) не спричиняє конфліктів

на шині. Проте він визначає, за яким саме фронтом тактового сигналу (переднім чи заднім) мікрофон виставлятиме біти даних. Тому для обох підблоків SAI слід індивідуально налаштувати полярність тактування згідно з конфігурацією конкретного мікрофона.

Для перенесення прийнятих аудіовибірок в ОЗП використовуються індивідуальні ресурси кожного підблоку у зв'язці з двома каналами DMA або двома векторами переривань. Оскільки кожен підблок обробляє свій потік незалежно, результуючий формат даних у відповідних регістрах SAI\_ADR та SAI\_BDR відповідає класичному монофонічному режиму роботи інтерфейсу.

Використання виділеного PDM-інтерфейсу забезпечує паралельну роботу з чотирма парами цифрових мікрофонів. Ще однією важливою перевагою цієї периферії є можливість апаратного введення мікрозатримок для кожного незалежного бітового потоку, що надходить від сенсорів.

Кожен інформаційний слот вміщує 8 біт PDM-потoku. Описаний функціонал PDM підтримується лише підблоком SAI\_A, сконфігурованим у режим TDM (мультиплексування з часовим поділом), і не сумісний із підблоком SAI\_B. Для оптимізації споживання пам'яті ОЗП розробник може програмно вказати кількість задіяних мікрофонів: 2, 4, 6 або 8 одиниць.

Наприклад, якщо пристрій використовує 3 мікрофони, у конфігурації необхідно вказати 4 активні канали. Потоки даних PDM транслюватимуться через виводи SAI\_D1 та SAI\_D2. Перша пара сенсорів передаватиме дані по лінії SAI\_D1, а третій мікрофон – по лінії SAI\_D2. Оптимальні варіанти призначення виводів для різної кількості підключених мікрофонів згруповано в Табл. 1.4.

*Таблиця 1.4. Призначення виводів інтерфейсу SAI для роботи з PDM-мікрофонами*

| Кількість мікрофонів | Лінії даних (I/O)              | Лінії тактування (CLK) |
|----------------------|--------------------------------|------------------------|
| 1–2                  | SAI_D1                         |                        |
| 3–4                  | SAI_D1, SAI_D2                 | Будь-яка з ліній:      |
| 5–6                  | SAI_D1, SAI_D2, SAI_D3         | SAI_CK1...SAI_CK4      |
| 7–8                  | SAI_D1, SAI_D2, SAI_D3, SAI_D4 |                        |

Для тактування мікрофонів може використовуватися один загальний вивід SAI\_CKx або декілька роздільних ліній SAI\_CK. Роздільне тактування забезпечує можливість динамічного керування живленням та вибіркового вмкнення зчитування з конкретних мікрофонів. Обов'язковою умовою для повного опитування всієї решітки є встановлення частоти тактування модуля SAI на рівні:  $f_{SAI} = N \times f_{mic}$ , де  $N$  – кількість мікрофонів (2, 4, 6, 8), а  $f_{mic}$  – частота опитування одного сенсора.

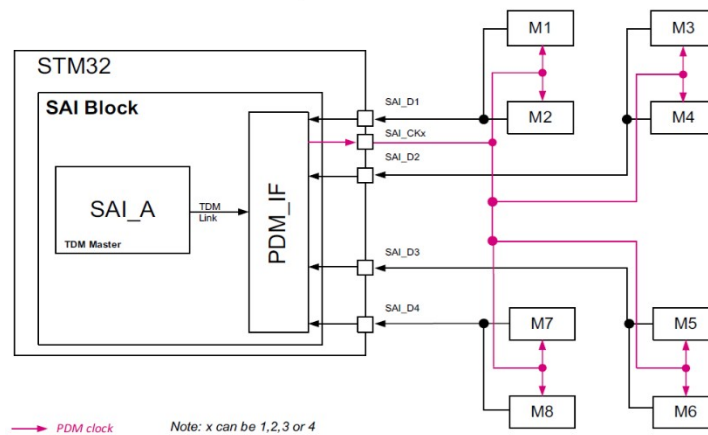


Рис. 1.25. Конфігурація підключення 8 цифрових мікрофонів до інтерфейсу SAI при спільному тактуванні

Використання PDM-режиму в периферії SAI ідеально підходить для масштабування аудіосистем до 4–8 стереопар мікрофонів. Завдяки вбудованій функції апаратного побайтового розділення вхідних потоків, блок SAI самостійно сортує дані від різних сенсорів. Це суттєво розвантажує центральний процесор, позбавляючи його від виконання програмних алгоритмів демультимплексування.

Для перенесення зчитаних підблоком SAI\_A вибірок в ОЗП використовується один канал DMA або відповідний вектор переривань. У цьому режимі регістр даних (SAI\_ADR) акумулює по 8 біт даних від кожного задіяного мікрофона. Структуру заповнення регістра проілюстровано нижче:

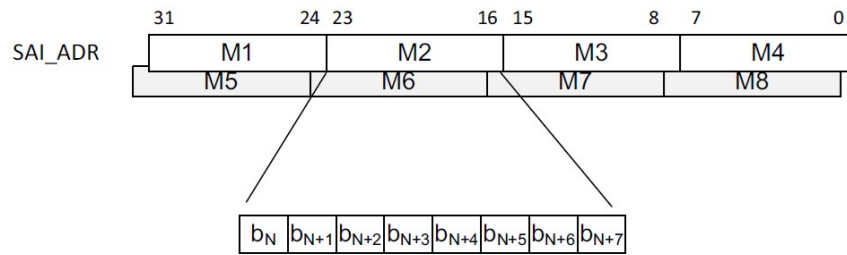


Рис. 1.26. Структура 32-бітного слова даних в PDM-режимі модуля SAI для 8-канальної мікрофонної решітки

Обсяг даних у регістрі SAI\_ADR визначається поточними налаштуваннями периферії SAI. При виборі 8-бітного розміру слота в регістрі зберігається по одному байту з одного мікрофона. Таким чином, щоб зібрати по одному байту вибірки від 8 задіяних PDM-мікрофонів, процесор або контролер DMA має виконати 8 циклів читання регістра SAI\_ADR.

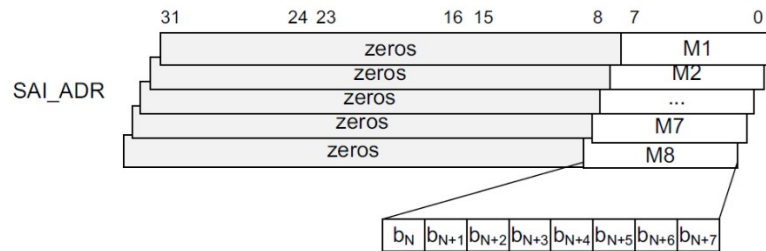


Рис. 1.27. Структура даних у PDM-режимі модуля SAI при послідовному опитуванні 8 каналів (розмір слота – 8 бітів)

Модуль DFSDM в архітектурі STM32 виконує роль цифрового фільтра для обробки бітових потоків сигма-дельта модуляції. Фактично цей блок працює як класичний АЦП, але з можливістю програмного регулювання розрядності та швидкості дискретизації вихідного сигналу.

Пряме підключення MEMS-мікрофонів із PDM-інтерфейсом до блока DFSDM дозволяє апаратно реалізувати операції фільтрації та децимації сигналу. Наявність індивідуального каналу DMA для кожного цифрового фільтра забезпечує автоматичне трасування незалежних аудіопотоків в ОЗП. Такий підхід позбавляє центральний процесор від виконання складних програмних обчислень для фільтрації та демультимплексування каналів.

Деякі версії периферії DFSDM дозволяють вводити мікрозатримки у вхідні цифрові потоки для точного фазового вирівнювання та стабільного зчи-

тування. Величина дискрету затримки відповідає коефіцієнту проріджування сигналу. Максимальна кількість мікрофонів, які можуть працювати з одним блоком DFSDM, безпосередньо обмежена числом його внутрішніх послідовних інтерфейсів та обчислювальних каналів фільтрації.

Зчитування потоку даних з PDM-сенсорів синхронізується опорною частотою DFSDM\_CKOUT, яку формує безпосередньо сам блок DFSDM. Регулювання частоти тактування здійснюється програмно через коефіцієнт ділення. Можливість виведення DFSDM\_CKOUT на роздільні фізичні порти мікроконтролера забезпечує гнучке керування активністю окремих каналів та оптимізацію енергоспоживання системи.

Схема на Рис. 1.28 орієнтована на пристрої з низьким енергоспоживанням. Вона дозволяє залишити активним лише один мікрофон (M1), заблокувавши тактування інших для їхнього переходу в режим сну (Sleep mode). Сенсори M3 та M4 так само можуть бути програмно знеструмлені (вимкнені). Перехід до повнофункціонального багатоканального режиму здійснюється шляхом відновлення та синхронізації тактової частоти на всіх лініях одночасно.

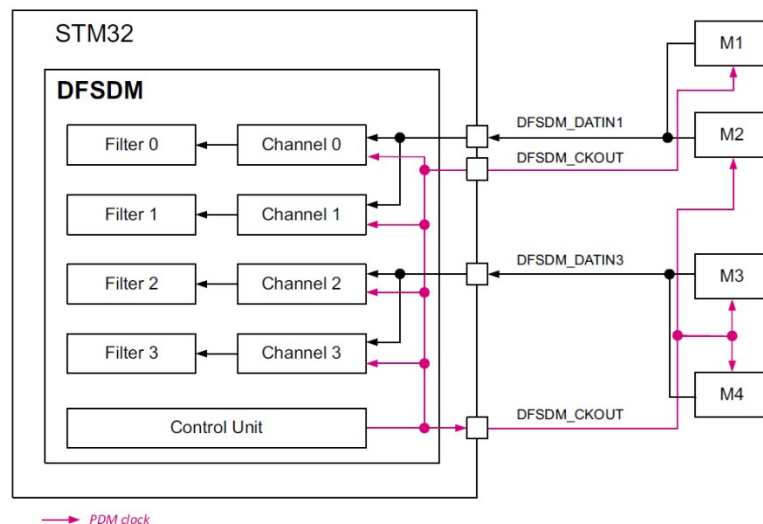


Рис. 1.28. Схема сполучення чотирьох цифрових PDM-мікрофонів із периферійним блоком DFSDM

Для реалізації стереорежиму з двома PDM-мікрофонами в архітектурі DFSDM задіюються два послідовні приймальні канали. Робота цих каналів

повністю координується вбудованим апаратним генератором тактування модуля DFSDM.

Для керування цифровими мікрофонами блок DFSDM формує опорний сигнал DFSDM\_CKOUT. При цьому канали x та x-1 конфігуруються на прийом даних з одного виводу DFSDM\_DATINx. Завдяки зчитуванню відліків по передньому та задньому фронтах синхроімпульсів, забезпечується одночасна робота двох PDM-мікрофонів на спільній шині даних. Отримані вибірки з кожного каналу транслюються на окремі апаратні фільтри для фінальної децимації та обробки.

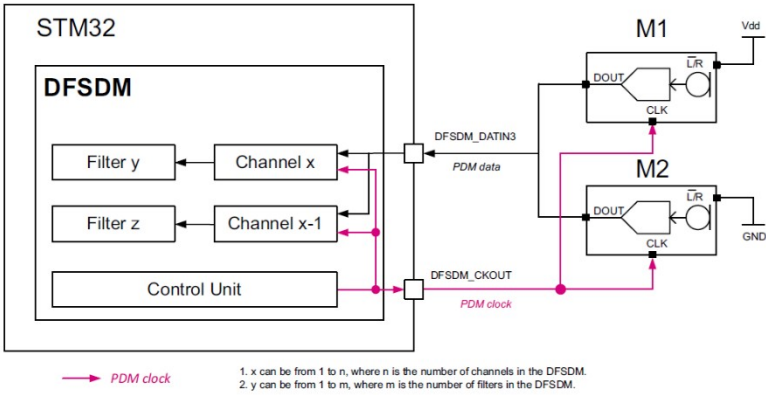


Рис. 1.29. Конфігурація стереопари цифрових мікрофонів в інтерфейсі DFSDM

Для перенесення оброблених цифровими фільтрами вибірок в ОЗП використовується окремий канал DMA або відповідний вектор переривань. Завдяки тому, що кожен фільтр обслуговує виключно свій мікрофон, вихідні дані не потребують додаткового демультимплексування (чергування відсутнє). Число активних каналів DMA у такій конфігурації завжди відповідає кількості працюючих фільтрів. Частота зовнішнього синхросигналу є керуючим фактором для внутрішньої логіки MEMS-мікрофона. У Табл.1.5 наведено залежність режимів роботи MEMS-мікрофона від частоти зовнішнього синхросигналу.

Таблиця 1.5. Залежність режимів роботи MEMC-мікрофона від частоти тактування

| Діапазон частот | Режим функціонування | Особливості та вимоги |
|-----------------|----------------------|-----------------------|
|-----------------|----------------------|-----------------------|

|                                    |                          |                                                                                                                  |
|------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------|
| < 100...200 кГц<br>(або відсутній) | Сон (Sleep mode)         | Мінімальне енергоспоживання, вихід DOUT у високо-імпедансному стані.                                             |
| 400 кГц...800 кГц                  | Знижене енергоспоживання | Струм споживання зменшено за рахунок деякого падіння акустичних характеристик (підтримується не всіма моделями). |
| 1,0 МГц...4,8 МГц                  | Нормальний (Normal mode) | Повна продуктивність, максимальна якість звуку. Вимагає високої стабільності тактового генератора.               |

Активація цифрового мікрофона з режиму сну відбувається автоматично при подачі на вхід CLK стабільного тактового сигналу заданої частоти. Слід враховувати, що пристрою потрібен транзитний час тривалістю в кілька мілісекунд для виходу на номінальний режим роботи та забезпечення очікуваної точності вибірок. Зведені дані щодо робочих частот тактування для різних режимів розробки наведено в Табл.1.6.

*Таблиця 1.6. Значення частоти тактування PDM-мікрофона залежно від цільового призначення системи*

| Цільове призначення проєкту                         | Робочий діапазон частот синхросигналу |
|-----------------------------------------------------|---------------------------------------|
| Базове захоплення звуку (енергоефективний режим)    | 400...800 кГц                         |
| Голосові системи та розпізнавання мови              | 1,0...1,5 МГц                         |
| Високоякісна реєстрація акустичних сигналів (Hi-Fi) | 2,4...4,8 МГц                         |

Конфігурація частоти синхросигналу має спиратися на топологію дерева тактування конкретного мікроконтролера. Необхідно детально вивчити апаратні ресурси блоків SPI, SAI та DFSDM, а також оцінити можливості їхніх внутрішніх генераторів і дільників частоти для забезпечення необхідної точності аудіоінтерфейсів. Архітектура аудіопериферії зазвичай передбачає розділення тактових сигналів на дві групи: тактування шини (Bus Clock): забезпечує роботу цифрової логіки та обмін даними з процесорним ядром (частота APB); тактування ядра (Kernel Clock): слугує базовою опорною частотою, з якої за допомогою дільників формується вихідний синхросигнал для передачі звукового потоку.

Вибір джерела для формування тактових імпульсів визначається задіяною периферією та програмною конфігурацією. Зокрема, архітектура

SPI-V1 у режимі класичного послідовного порту спирається на частоту периферійної шини APB. Проте для реалізації протоколу I2S залучається окремий функціональний блок ядра, оптимізований під стандартні аудіочастоти.

Формування частоти на базі шини APB має значні обмеження в гнучкості налаштування. Справа в тому, що цей тактовий сигнал є груповим: його коригування автоматично змінить режим роботи та швидкість обміну даними для всієї сукупності периферійних пристроїв, які розділяють даний домен тактування.

Водночас при використанні таймера для прийому стереофонічного потоку з мікрофонів, залучення єдиної шини APB (як для таймера, так і для модуля прийому) усуває проблему розсинхронізації, згадану в попередньому розділі. Архітектура деяких МК дозволяє безпосередньо комутувати тактовий сигнал шини APB на вхід опорної частоти (ядра) периферії.

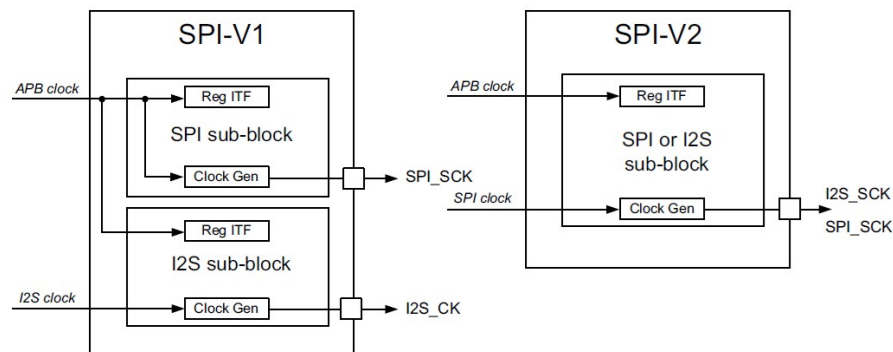


Рис. 1.30. Архітектура тактування ядра та інтерфейсної логіки модуля SPI

У блоках SPI-V2, SAI та DFSDM завжди реалізовано незалежний тракт тактування ядра. Цей підхід забезпечує високу гнучкість конфігурування, дозволяючи генерувати точну сітку частот для роботи з MEMS-мікрофонами без прив'язки до робочої частоти інтерфейсної шини мікроконтролера.

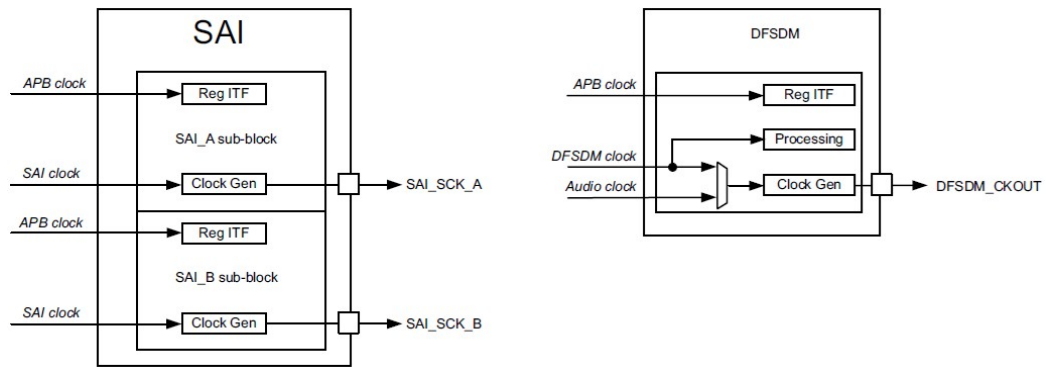


Рис. 1.31. Архітектура тактування та топологія шин для периферійних блоків SAI та DFSDM

Спеціалізована бібліотека для обробки PDM-потoku забезпечує ефективну програмну реалізацію алгоритмів демодуляції та реконструкції аудіосигналу для MEMC-мікрофонів на базі платформи STM32. До складу бібліотеки входять цифрові фільтри децимації, які трансформують однорозрядний високочастотний вхідний сигнал у низькочастотний 16-бітний код PCM з необхідними параметрами семплювання.

Вихідним інтерфейсом цифрового MEMC-сенсора є імпульсно-щільнісна модуляція (PDM). Вона транслюється як високошвидкісний однобітний потік із тактовою частотою  $1 \dots 3,25$  МГц. Прийом та первинна буферизація PDM-даних покладається на апаратний послідовний порт мікроконтролера STM32. Отриманий цифровий потік автоматично пересилається контролером DMA в оперативну пам'ять без безпосередньої участі центрального процесора. Залучення DMA суттєво розвантажує обчислювальні ресурси системи. Сформовані в результаті декодування PCM-відліки готові для подальшого збереження в структурі аудіофайлу на флеш-накопичувачі або для трансляції на зовнішній ЦАП.

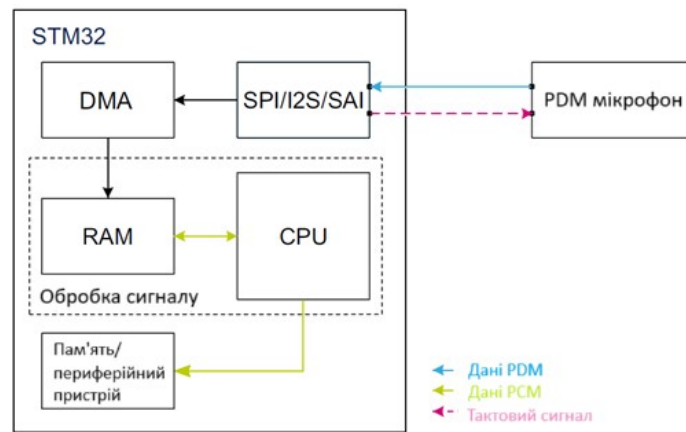


Рис. 1.32. Функціональна схема трансляції, децимації та збереження звукових даних MEMS-мікрофона

*Цифрова обробка звукових даних у форматі PDM.* Спеціалізована бібліотека для декодування PDM-звуку базується на двостадійному алгоритмі обробки. Спочатку виконуються операції цифрової фільтрації та децимації вхідного сигналу, після чого сформований PCM-потік піддається фінальному цифровому опрацюванню згідно з прикладними завданнями проекту.

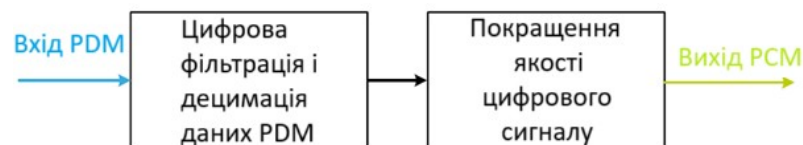


Рис. 1.33. Цифрове опрацювання звукового сигналу

- Етап 1: Вхідний PDM-сигнал проходить крізь блок фільтрації та децимації. Результатом цього процесу є перетворення однобітового потоку на стандартний аудіосигнал із необхідною частотою дискретизації та роздільною здатністю.
- Етап 2: Цифровий звук із виходу попередньої ланки фільтрації піддається фінішній обробці. Використання комбінації височастотного та низькочастотного фільтрів дозволяє відсікти небажані шуми й сформувати належну якість вихідного сигналу.

Передбачена функція ініціалізації дозволяє програмно вмикати або вимикати ФВЧ та ФНЧ, а також задавати необхідні значення частоти зрізу для кожного з них. Якщо пара PDM-мікрофонів працює на одній шині з чергуванням бітів, на рівні програмного забезпечення необхідно реалізувати алгоритм

деінтерлівінгу. Тільки після повного виокремлення даних кожного з мікрофонів можна переходити до етапу цифрової фільтрації та формування PCM-відліків. Вихідний PDM-потік із цифрового мікрофона є послідовністю 1-бітних значень із частотою проходження від 1,0 до 3,25 МГц. Дані захоплюються приймальним вузлом інтерфейсу DFSDM, що виконує роль мосту між ядром мікроконтролера та сигма-дельта модулятором MEMS-сенсора. Цифрова фільтрація виконується у фоновому режимі та не потребує постійної уваги CPU. Вона забезпечує перетворення 1-бітного потоку на вибірки з вищою роздільною здатністю та меншою частотою. Для цього первинні (необроблені) аудіодані спочатку транслуються з мікрофона в буфер RAM за допомогою DMA, після чого програмний цифровий фільтр здійснює їх фінальну обробку. Вихідні PCM-дані з цифрового фільтра передаються для подальшої обробки, залежно від конфігурації системи, аудіопотік може архівуватися у форматі WAV на картах пам'яті чи USB-накопичувачах або транслюватися на зовнішній ЦАП для відтворення.

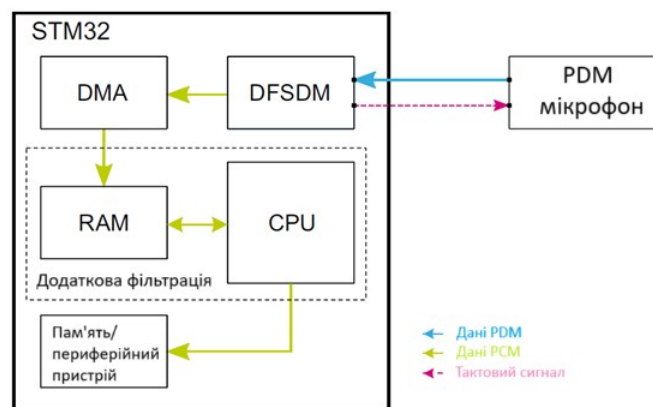


Рис. 1.34. Алгоритм збору та перетворення цифрового звукового сигналу на базі інтерфейсу DFSDM

## РОЗДІЛ 2

### АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ

Для проєктування апаратного забезпечення цифрового пристрою аудіофіксації правопорушень для працівників поліції було сформовано таку елементну базу:

- відлагоджувальна плата STM32F407G-DISC1 на базі високо-продуктивного МК STM32F407VGT6 (архітектура ARM Cortex-M4);
- цифровий MEMS-мікрофон MP45DT02 з PDM-інтерфейсом;
- 24-бітний стереофонічний цифро-аналоговий перетворювач (DAC) CS43L22 з інтегрованим підсилювачем класу D;
- алфавітно-цифровий рідкокристалічний дисплей WH1602B-NYG-CT (інформаційна ємність: 16 символів на 2 рядки);
- портативний флеш-накопичувач з інтерфейсом USB для збереження аудіофайлів.

#### **2.1. Відлагоджувальна плата STM32F407G-DISC1 на МК ARM Cortex-M4 STM32F407VGT6**

Відлагоджувальна плата STM32F407G-DISC1 базується на потужному мікроконтролері STM32F407VGT6 (ядро Cortex-M4 з математичним співпроцесором FPU). Його процесорне ядро забезпечує продуктивність на тактовій частоті до 168 МГц. Внутрішні ресурси пам'яті включають 192 Кбайт ОЗП та 1 Мбайт Flash-пам'яті. Апаратний інтерфейс USB підтримує архітектуру OTG (On-The-Go), що дає змогу гнучко перемикатися між ролями ведучого та веденого пристроїв. Для систем безпеки вбудовано апаратний блок RNG. Потужний периферійний комплекс, окрім базових інтерфейсів (ADC, DAC, SPI, I2C, PWM, RTC), містить швидкісні шини: SDIO (для SD-карт), DCMI (для паралельного інтерфейсу відеокамер) та інтерфейс Ethernet MAC (Рис. 2.1).

Інтегровані на платі цифровий MEMS-мікрофон MP45DT02 та 24-бітний стерео-ЦАП CS43L22 (з підсилювачем класу D та блоком цифрового регулю-

вання параметрів звуку) формують повноцінний аудіотракт. Це відкриває можливість проектування на базі мікроконтролера високоякісних систем цифрової обробки сигналів (DSP) та комплексів реєстрації аудіоінформації.

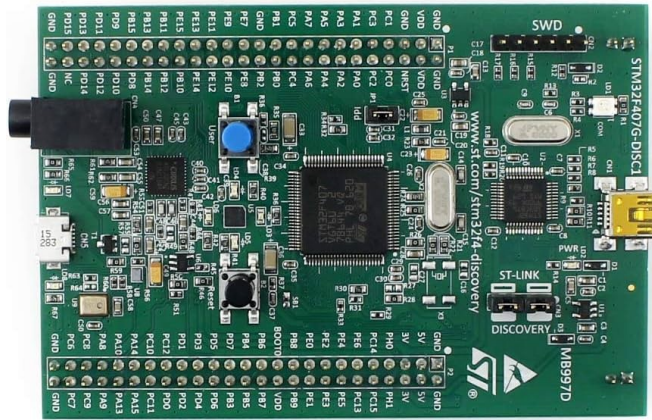


Рис. 2.1. Зовнішній вигляд відлагоджувальної плати STM32F407G-DISC1

Апаратна частина платформи також містить інтегрований 3-осьовий цифровий акселерометр LIS302DL. Серед його розширених функціональних можливостей слід виділити динамічне масштабування діапазону вимірювань, детектування імпульсних впливів (одиначних та подвійних поштовхів) і формування апаратних переривань при настанні заданих подій.

Зв'язок із датчиком прискорень може здійснюватися через послідовні шини SPI або I2C. Для сповіщення процесора передбачені лінії переривань. Блок світлодіодної індикації містить 4 різнокольорові індикатори (LD3–LD6), згруповані навколо сенсора. Органи ручного керування представлені двома тактовими кнопками: вільно програмованою клавішею та кнопкою перезавантаження системи.

Стабілізацію напруги на платі STM32F407G-DISC1 забезпечує мікросхема LD3985M33R. Даний регулятор розрахований на максимальний вихідний струм до 150 мА, що повністю покриває потреби базової архітектури МК та малопотужної периферії. Проте для розгалужених систем із високим енергоспоживанням (великою кількістю зовнішніх блоків) обов'язковою умовою є використання додаткового потужного стабілізатора з вихідною напругою 3,3 В.

Для завантаження коду та трасування програм передбачено вбудований апаратний налагоджувач ST-Link. Візуалізація його станів (налагодження чи запис пам'яті) забезпечується двоколірним світлодіодом. Подача напруги живлення на плату реалізована через стандартний порт mini-USB. У Табл.2.1 наведено технічні параметри платформи STM32F4DISCOVERY.

*Таблиця 2.1. Технічні параметри платформи STM32F4DISCOVERY*

| Категорія           | Характеристика                                                       |
|---------------------|----------------------------------------------------------------------|
| Мікроконтролер      | STM32F407VGT6 (ARM Cortex-M4F, 1 Мбайт Flash, 192 Кбайт RAM)         |
| Емуляція та зв'язок | Вбудований ST-LINK/V2-A, Virtual COM-port, Mass storage              |
| Живлення            | Вхід: USB або зовнішнє 5 В. Вихід для периферії: 3 В та 5 В          |
| Вбудовані сенсори   | Акселерометр LIS302DL/LIS3DSH, MEMS-мікрофон MP45DT02                |
| Аудіотракт          | 24-бітний ЦАП CS43L22 з інтегрованим підсилювачем класу D            |
| Керування           | Кнопка скидання (Reset), кнопка користувача (User)                   |
| Індикація           | 8 світлодіодів (живлення, USB-трафік, USB-помилки, 4 користувацьких) |
| Інтерфейси          | USB OTG FS (micro-AB), повний доступ до всіх пінів LQFP100           |

У Табл.2.2 наведено порівняння характеристик інструментальних засобів розробки ПЗ для МК STM32.

*Таблиця 2.2. Порівняльна характеристика інструментальних засобів розробки ПЗ*

| Назва IDE              | Компілятор | Тип ліцензії        | Особливості/Статус                                    |
|------------------------|------------|---------------------|-------------------------------------------------------|
| STM32CubeIDE           | GCC ARM    | Безкоштовна         | Офіційне середовище від ST, повна інтеграція з CubeMX |
| Keil MDK-ARM           | ARM C/C++  | Комерційна (e Lite) | Промисловий стандарт, потужний симулятор              |
| IAR Embedded Workbench | IAR C/C++  | Комерційна          | Найвища швидкість та оптимізація коду                 |
| Atollic TrueSTUDIO     | GCC        | Безкоштовна         | Покладена в основу сучасної STM32CubeIDE              |
| CooCox CoIDE           | GCC        | Безкоштовна         | Проект архівовано, підтримку припинено                |

Для розширення апаратних можливостей до плати STM32F407G-DISC1 можуть бути інтегровані додаткові модулі через наявні гребінки розширення. Програмна складова комплекту STM32F407G-DISC1 Discovery Kit базується на безкоштовних інструментах і драйверах з пакету STM32CubeF4. Детальна схема розпіновки (pinout) із зазначенням

функціоналу кожного виводу плати представлена на Рис. 2.2.

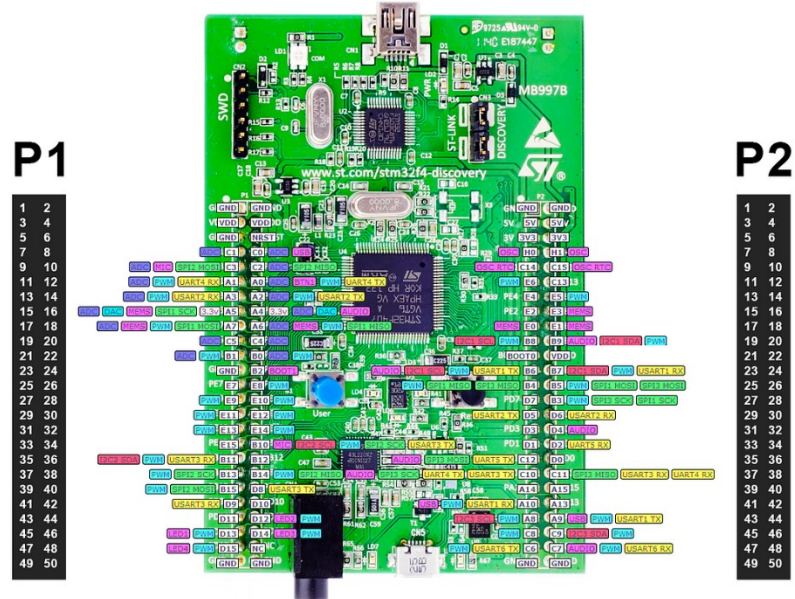


Рис. 2.2. Топологія та специфікація контактів інтерфейсних роз'ємів плати STM32F407G-DISC1

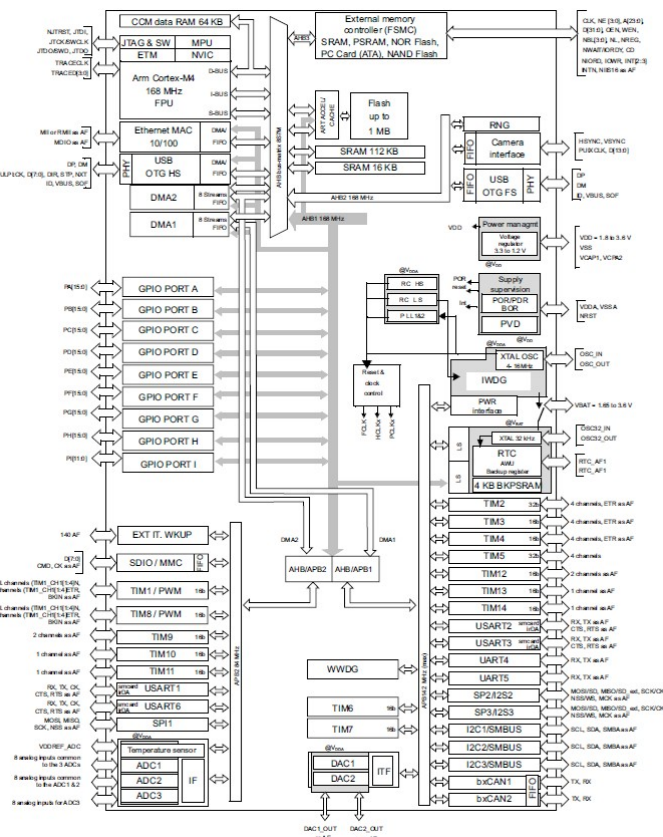


Рис. 2.3. Блок-схема внутрішньої архітектури периферії МК STM32F407VGT6

## 2.2. Цифровий MEMS-мікрофон MP45DT02

Мікрофон MP45DT02 – це мініатюрний всеспрямований цифровий сенсор із верхнім розташуванням звукового отвору. Конструктивно пристрій об’єднує в одному корпусі MEMS-капсуль та інтегральну схему зчитування з можливістю об’єднання двох мікрофонів у стереопару. Чутлива мембрана виявляє коливання звукового тиску і виготовляється методом прецизійної мікрообробки кремнію, що є стандартом для сучасних акустичних датчиків. Вбудована електронна схема обробки сигналу базується на CMOS-технологічному процесі. Це забезпечило апаратну реалізацію сигма-дельта модулятора, який безпосередньо генерує високочастотний цифровий потік у форматі PDM.

Параметри MP45DT02 охоплюють точку акустичного переваантаження 120 dBSPL, відношення сигнал/шум 61 дБ та чутливість -26 дБ. Датчик постачається в SMD-сумісному корпусі та розрахований на роботу в промисловому температурному діапазоні (-30...+85°C). Цифровий вихідний потік у поєднанні з малою товщиною корпусу (1,25 мм) зумовлюють доцільність його використання в малогабаритній портативній техніці та вбудованих системах.

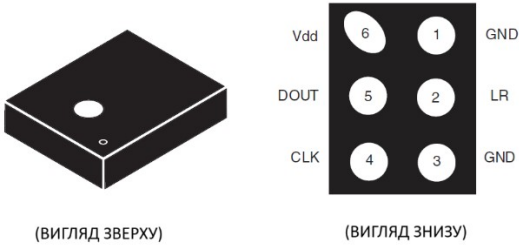


Рис. 2.4. Схема розташування та призначення контактних площадок МЕМС-мікрофона MP45DT02

Таблиця 2.1. Призначення виводів мікросхеми мікрофона MP45DT02

| № виводу | Назва | Функціональне призначення                                                                                                                                 |
|----------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | GND   | Загальний провід (0 В, “земля”)                                                                                                                           |
| 2        | LR    | Вибір лівого/правого аудіоканалу (Left/Right). Для активації стереорежиму вивід першого мікрофона з’єднують із GND (або VDD), а другого – з VDD (або GND) |
| 3        | GND   | Загальний провід (0 В, “земля”)                                                                                                                           |
| 4        | CLK   | Вхід сигналу тактування (Clock)                                                                                                                           |
| 5        | DOUT  | Вихід цифрового PDM-поток даних                                                                                                                           |
| 6        | VDD   | Вхід напруги живлення                                                                                                                                     |

Таблиця 2.2. Динамічні параметри передачі даних MP45DT02

| Символ | Характеристика | Min | Max | Одиниця |
|--------|----------------|-----|-----|---------|
|--------|----------------|-----|-----|---------|

|             |                                                   |     |      |     |
|-------------|---------------------------------------------------|-----|------|-----|
| $f_{CLK}$   | Робоча частота синхронізації                      | 1,0 | 3,25 | МГц |
| $f_{PD}$    | Частота переходу в режим сну (Power Down)         | -   | 0,23 | МГц |
| $T_{CLK}$   | Період сигналу CLK                                | 308 | 1000 | нс  |
| $T_{R,EN}$  | Затримка ввімкнення шини даних для правого каналу | 30  | -    | нс  |
| $T_{R,DIS}$ | Затримка вимкнення шини даних для правого каналу  | -   | 16   | нс  |

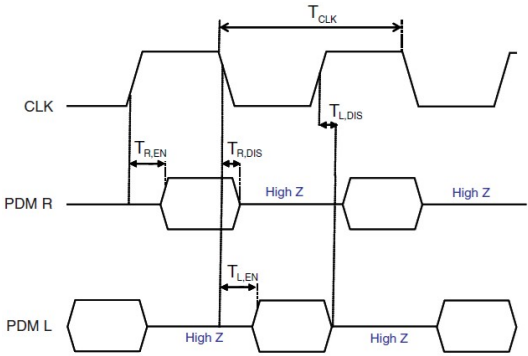


Рис. 2.5. Часові діаграми PDM-інтерфейсу мікросхеми MP45DT02

Графік амплітудно-частотної характеристики мікрофона MP45DT02 зображено на Рис. 2.6.

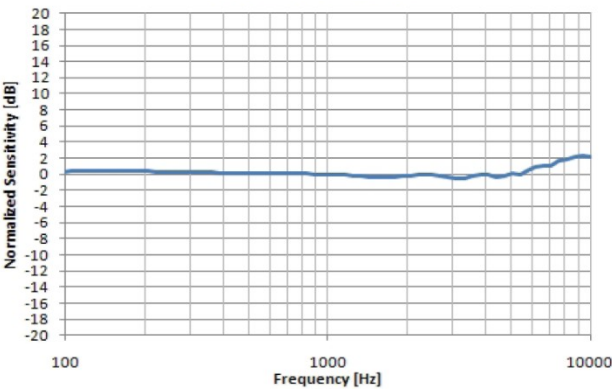


Рис. 2.6. Типова амплітудно-частотна характеристика мікрофона MP45DT02, нормована на частоті 1 кГц

Таблиця 2.3. Допустимі відхилення АЧХ цифрових мікрофонів

| Частотний діапазон, Гц | Нижня межа | Верхня межа | Одиниця виміру  |
|------------------------|------------|-------------|-----------------|
| 20...100               | -5         | +5          | dBr (при 1 кГц) |
| 100...8000             | -2         | +2          | dBr (при 1 кГц) |
| 8000...10000           | -5         | +5          | dBr (при 1 кГц) |

Акустичний сенсор мікрофона MP45DT02 реалізований на базі кремнієвого чіпа і працює за ємнісним принципом. Чутливий елемент утворений парою провідних пластин (рухомою мембраною та нерухомою основою).

Звуковий тиск викликає механічні коливання рухомої пластини, модулюючи міжелектродну ємність, яка згодом аналізується вбудованою інтегральною схемою.

Таблиця 2.4. Формування вихідних даних залежно від тактування

| Підключення L/R          | При CLK=0 (Низький рівень)     | При CLK=1 (Високий рівень)     |
|--------------------------|--------------------------------|--------------------------------|
| До шини заземлення (GND) | Активний стан (передача даних) | Високоімпедансний стан (Hi-Z)  |
| До шини живлення (VDD)   | Високоімпедансний стан (Hi-Z)  | Активний стан (передача даних) |

Конфігурація виводу L/R задає режим видачі даних на шину Dout. Залежно від необхідного каналу (лівого чи правого), цей вивід комутують із лінією додатного потенціалу Vdd або з шиною заземлення GND.

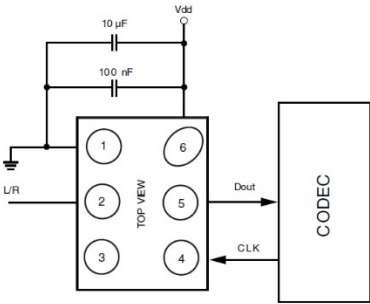


Рис. 2.7. Схема підключення мікрофона MP45DT02 до МК STM32F407VGT6

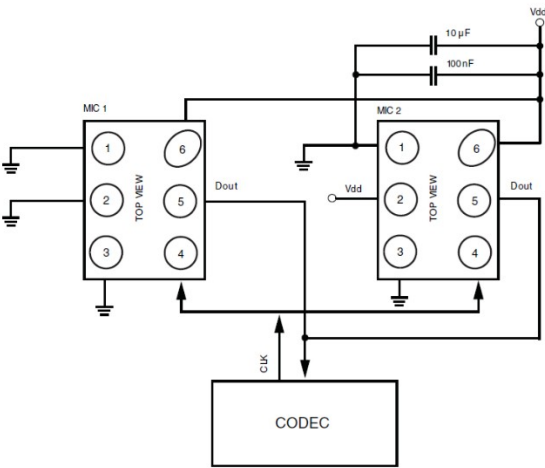


Рис. 2.8. Схема підключення стереопари мікрофонів MP45DT02 до спільної шини даних МК

Блокувальні керамічні конденсатори номіналом 100 нФ та 10 мкФ мають бути розташовані в безпосередній близькості до виводу живлення (Vdd) сенсора. Вивід вибору каналу L/R комутується з лінією позитивного потенціалу Vdd або із шиною заземлення GND.

### 2.3. Проєктування цифрового пристрою аудіофіксації правопорушень

Розробка поліцейського спецприладу для звукозапису доказової бази вимагає точного налаштування апаратної периферії та створення первинного С-коду в середовищі STM32CubeMX. Головне завдання системи полягає у зчитуванні PDM-потoku з цифрових мікрофонів (як в один, так і у два канали) з його подальшою транскодацією у звичний PCM-формат.

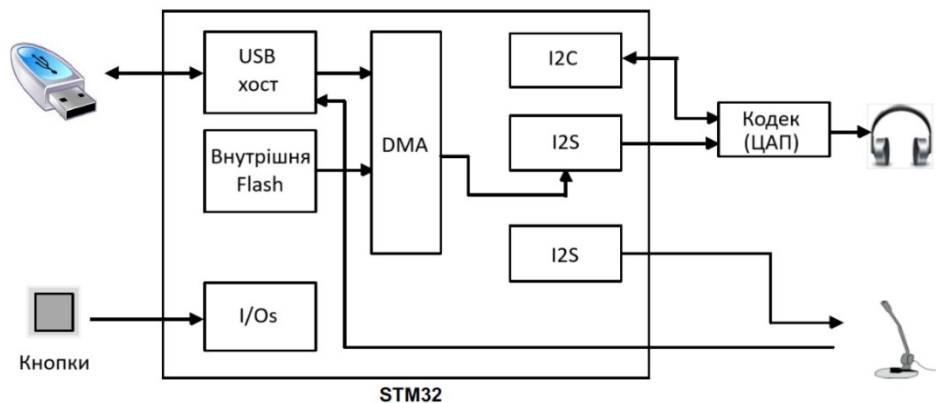


Рис.2.9. Архітектура цифрового засобу для звукозапису правопорушень

Створений проєкт на стадії розробки наповнюється відповідним програмним забезпеченням. До розроблюваної системи висуваються такі вимоги:

- Отримання аудіосигналу має здійснюватися у форматі PCM (з частотами дискретизації 16 кГц чи 48 кГц).
- Для цифрового PDM-мікрофона передбачено тактування із частотою, що перевищує частоту дискретизації PCM-потoku в 64 рази (застосовується 64-кратна надлишкова дискретизація).

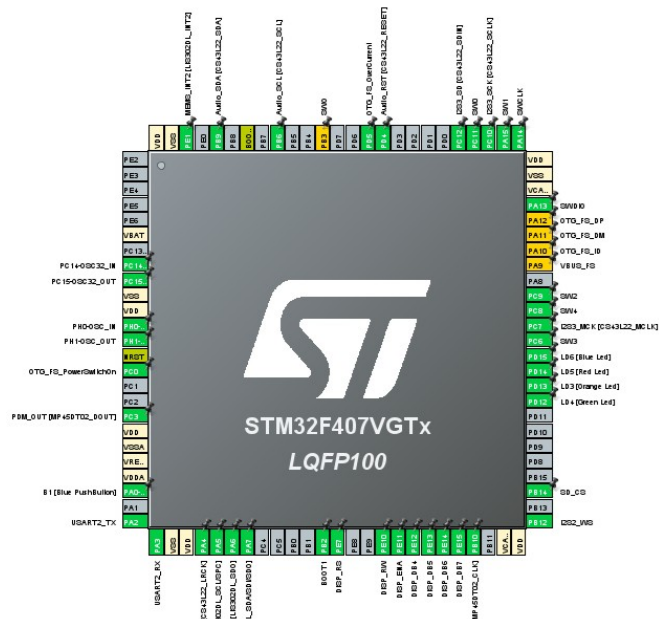


Рис. 2.10. Параметри налаштування МК STM32F407VGT6 в утиліті STM32CubeMX

Практична реалізація проекту базується на налагоджувальній платі STM32F407G-DISC1 та цифровому MEMS-мікрофоні MP45DT02, сполученому через шину I2S. Для формування вихідного PCM-потoku з частотою дискретизації 16 кГц, модуль I2S мікроконтролера має генерувати тактовий сигнал із частотою 1,024 МГц для одного каналу (моно) та 2,048 МГц для двох каналів (стерео). Відповідно, при переході на частоту дискретизації 48 кГц, необхідна частота тактування мікрофона зростає до 3,072 МГц у монорежимі та до 6,144 МГц у стереорежимі.

*Конфігурація модуля I2S2 з допомогою утиліти STM32CubeMX.* Процес активації та налаштування I2S виконується в утиліті STM32CubeMX. На вкладці конфігурації виводів у розділі мультимедійних інтерфейсів вибирається модуль I2S2. Для взаємодії з мікрофоном його роботу потрібно перевести в режим ведучого з напівдуплексним обміном даними (Half-Duplex Master). Наочна ілюстрація вибору цих параметрів зображена на рисунку.

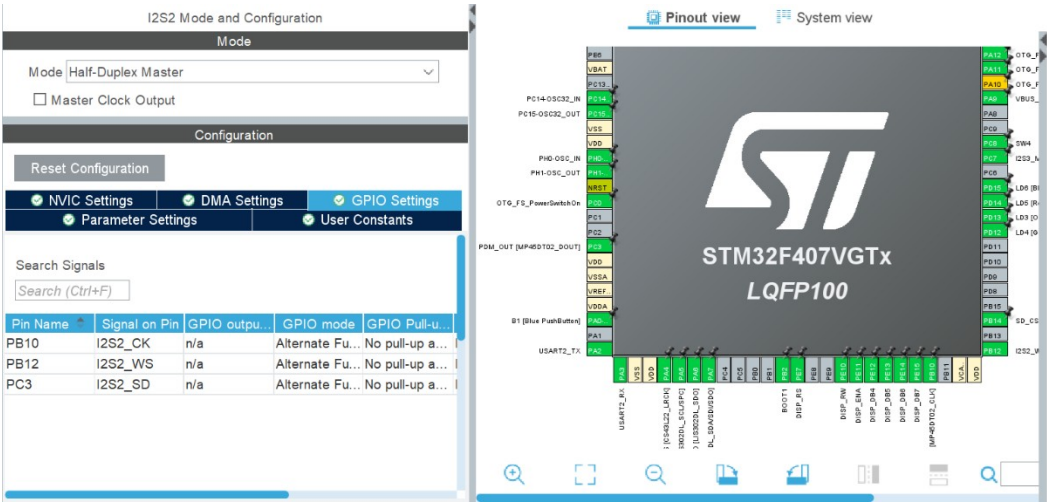


Рис. 2.11. Конфігурація периферійного модуля I2S2 в режимі ведучого (Half-Duplex Master)

Коли налаштування завершено, лінії I2S2\_SD, I2S2\_CK та I2S2\_WS змінюють колір на зелений, що свідчить про їхню готовність до роботи. Проте специфіка цього проєкту дозволяє не використовувати вивід I2S2\_WS. Щоб раціонально використати ресурси мікроконтролера, цей пін можна перевести в режим загального призначення (GPIO). Така операція вимагає редагування згенерованого коду безпосередньо у файлі `stm32f4xx_hal_msp.c` у блоці ініціалізації периферії (MSP).

*Конфігурація джерела синхросигналу для модуля I2S2.* Конфігурація генератора синхросигналу для блоку I2S2 забезпечує роботу пристрою із частотами звуку 16 та 48 кГц в одноканальному або двоканальному режимах. Базовим джерелом стабільного сигналу є зовнішній генератор HSE на 8 МГц. Для досягнення повної синхронізації між таймером та інтерфейсом I2S (що критично для стереорежиму), тактування останнього здійснюється через дільник PLLR. Критерієм точності такого налаштування є різниця між реальною частотою дискретизації та розрахунковим показником. У Табл.2.5 наведено параметри конфігурації тактування I2S2 та похибка частоти дискретизації.

Таблиця 2.5. Параметри конфігурації тактування I2S2 та похибка частоти дискретизації

|         |       |         |      |      |      |      |         |            |
|---------|-------|---------|------|------|------|------|---------|------------|
| Цільова | Режим | Джерело | DivM | PLLN | DivR | DivP | Частота | Відхилення |
|---------|-------|---------|------|------|------|------|---------|------------|

| частота (fs) | роботи                          | сигналу (Mux) |   |     |   |   | I2S (МГц) | (ppm) |
|--------------|---------------------------------|---------------|---|-----|---|---|-----------|-------|
| 16 кГц       | Моно                            | PLLI2SR       | 5 | 192 | 2 | - | 153,60    | 0     |
|              | Моно                            | PLLI2SR       | 5 | 192 | 5 | - | 61,44     | 0     |
|              | Сtereo таймером) <sup>(3)</sup> | PLLР          | 7 | 344 | 6 | 4 | 65,52     | -186  |
| 48 кГц       | Моно                            | PLLI2SR       | 5 | 192 | 2 | - | 153,60    | 0     |
|              | Моно                            | PLLI2SR       | 5 | 192 | 5 | - | 61,44     | 0     |
|              | Сtereo таймером) <sup>(3)</sup> | PLLР          | 7 | 344 | 4 | 4 | 98,29     | -186  |

Ключовою вимогою для коректного функціонування I2S є забезпечення вищої тактової частоти порівняно з частотою периферійної шини APB. Особливості налаштування частоти тактування інтерфейсу I2S для монофонічного режиму проілюстровані на Рис. 2.12.

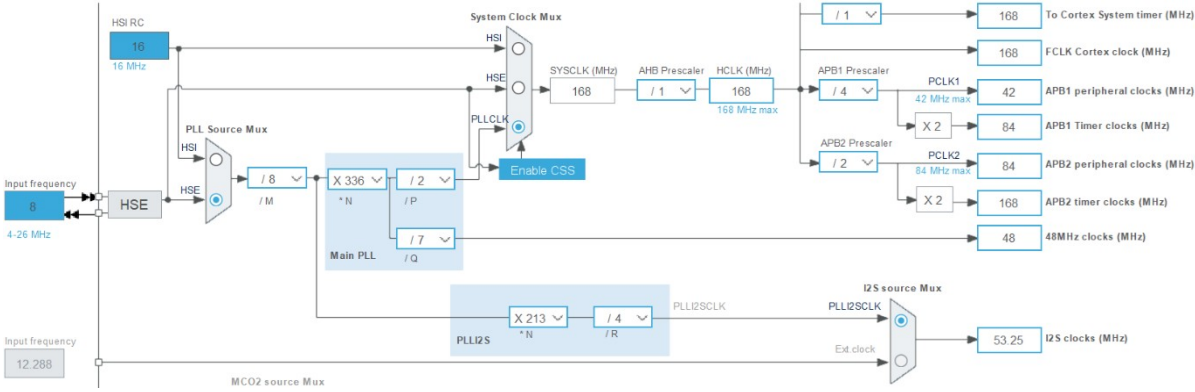


Рис. 2.12. Параметри тактування шини I2S в утиліті STM32CubeMX

Процес вибору інтерфейсу I2S2 в робочому середовищі показано на Рис. 2.13. Конфігурування здійснюється у вкладці Pinout & Configuration шляхом переходу до блоку Multimedia.

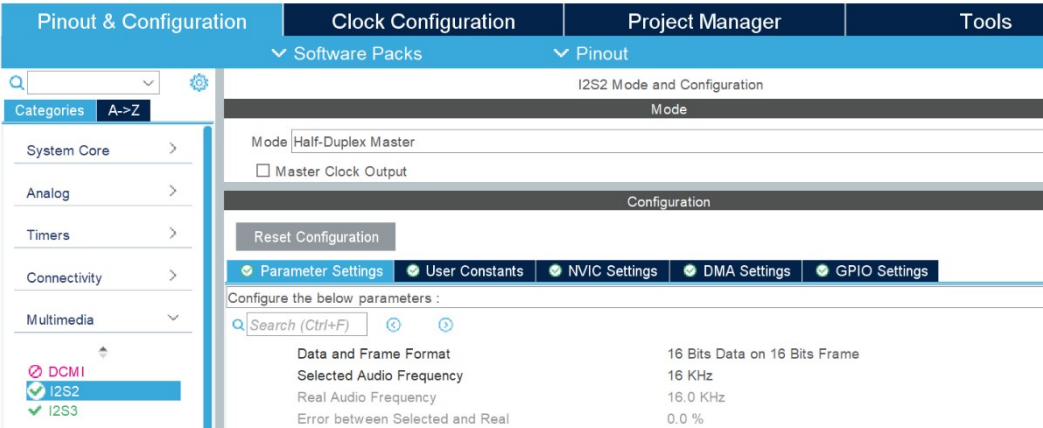


Рис. 2.13. Вікно ініціалізації периферійного інтерфейсу I2S2

а) *Встановлення робочих параметрів I2S2.* Налаштування виконується в меню I2S2 Mode and Configuration на вкладці Parameter Settings, де задаються такі параметри: для частоти дискретизації (Selected Audio Frequency) вказується показник 16 кГц; як стандарт зв'язку (Communication Standard) застосовується протокол MSB First (Left Justified). Особливістю є те, що I2S2 фіксує дані з мікрофона під час спаду синхроімпульсу. Також врахуйте, що для монофонічного режиму ніжку мікрофона L/R слід підключити до “землі” (GND). Графічне відображення цих налаштувань для частоти 16 кГц (моно) наведено на Рис.2.14.

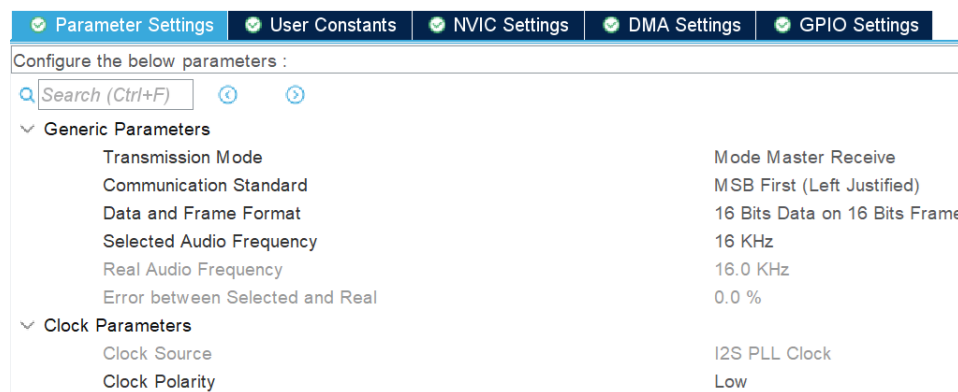


Рис.2.14. Налаштовані параметри I2S2 для оцифрування звуку (16 кГц, моно)

б) *Налаштування режиму DMA для периферії I2S2.* Для запису PDM-потoku, що надходить через інтерфейс I2S2, у буфер ОЗП задіяно апаратний контролер DMA. Процедура налаштування передбачає перехід до розділу DMA Settings у загальному меню I2S2 Mode and Configuration та додавання нового запиту DMA під назвою SPI2\_RX. Графічна ілюстрація додавання цього запиту для шини I2S2 наведена на Рис.2.15.

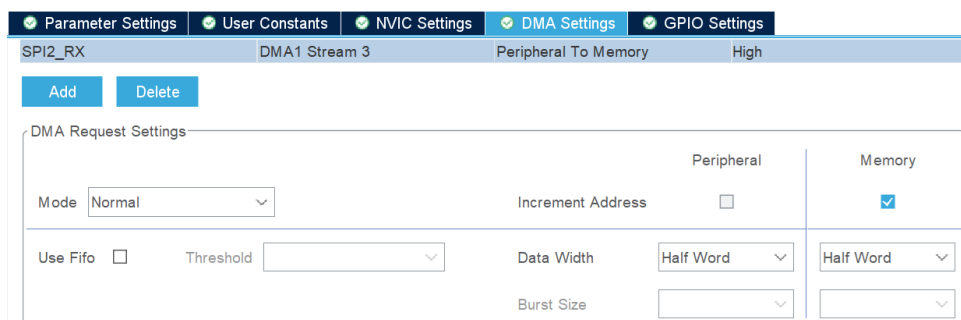


Рис. 2.15. Конфігурація каналу DMA для прийому даних по інтерфейсу I2S2

Щоб завершити конфігурацію модуля DMA, необхідно клікнути на раніше доданий запит SPI2\_RX та заповнити параметри у нижньому вікні

(DMA Request Settings). Наочний приклад виставлених значень цих параметрів зображено на рисунку.

*Параметризація інтерфейсів I2S3 та I2C1 для роботи з аудіокодеком CS43L22.* Для відтворення PCM-аудіо через зовнішній 24-бітний стерео-ЦАП CS43L22 задіюється інтерфейс I2S3. Налаштування здійснюється в утиліті CubeMX шляхом вибору модуля I2S3 у розділі мультимедійних пристроїв. Для нього встановлюється режим напівдуплексного ведучого пристрою на передачу (Half-Duplex Master/Master Transmit). Після активації відповідні ніжки мікроконтролера, що йдуть до ЦАП (а саме PA4, PC7, PC10 та PC12), змінять свій колір на зелений.

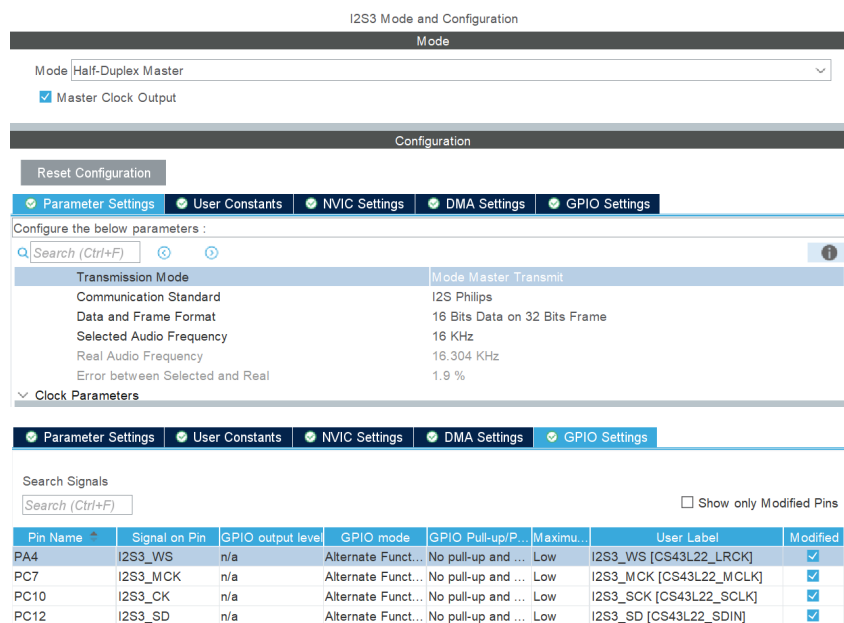


Рис. 2.16. Конфігурація модуля I2S3 в режимі ведучого передавача (Master Transmit)

*Налаштування режиму DMA для периферії I2S3.* Для трансляції записаних у пам'яті МК звукових даних у форматі PCM на аудіочип CS43L22 задіюється апаратний контролер DMA. У діалоговому вікні I2S3 Mode and Configuration необхідно відкрити розділ DMA Settings і згенерувати запит DMA – SPI3\_TX. Візуалізація процесу створення цього запиту для шини I2S3 наведена на Рис.2.17.

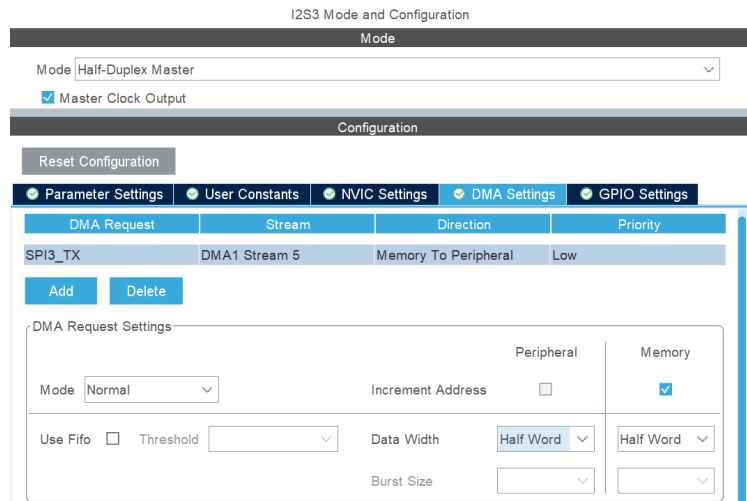


Рис. 2.17. Додавання запиту DMA SPI3\_TX для виведення даних через I2S3

Фіналізація налаштувань модуля DMA передбачає вибір створеного запиту SPI3\_TX та заповнення відповідних полів у вікні DMA Request Settings. Конкретні параметри, які потрібно встановити для стабільної трансляції даних, детально зображені на рисунку.

*Конфігурація шини I2C1 для управління аудіочіпом.* Для надсилання керуючих сигналів на цифро-аналоговий перетворювач CS43L22 використовується периферія I2C1. Щоб виставити необхідні параметри зв'язку, в утиліті CubeMX потрібно відкрити вкладку Pinout & Configuration, перейти до категорії Connectivity та натиснути на I2C1. Вікно з конфігураційними даними показано на Рис.2.18.

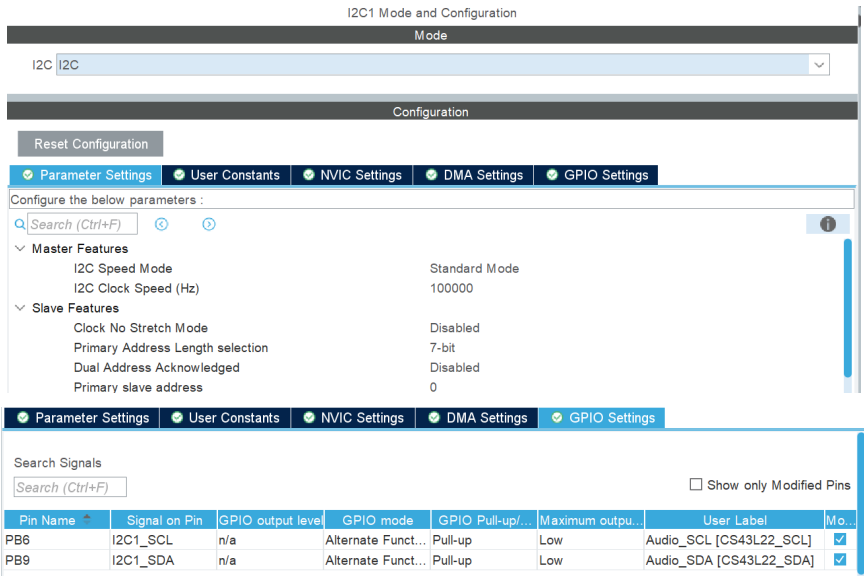


Рис. 2.18. Конфігурація шини I2C1 для зв'язку з аудіокодеком

*Додавання програмного пакета PDM2PCM.* За допомогою генератора коду STM32CubeMX можна легко додати до проєкту бібліотеку PDM2PCM та виставити її параметри ще на етапі попереднього конфігурування системи. Наочна ілюстрація активації цієї бібліотеки представлена на Рис.2.20.

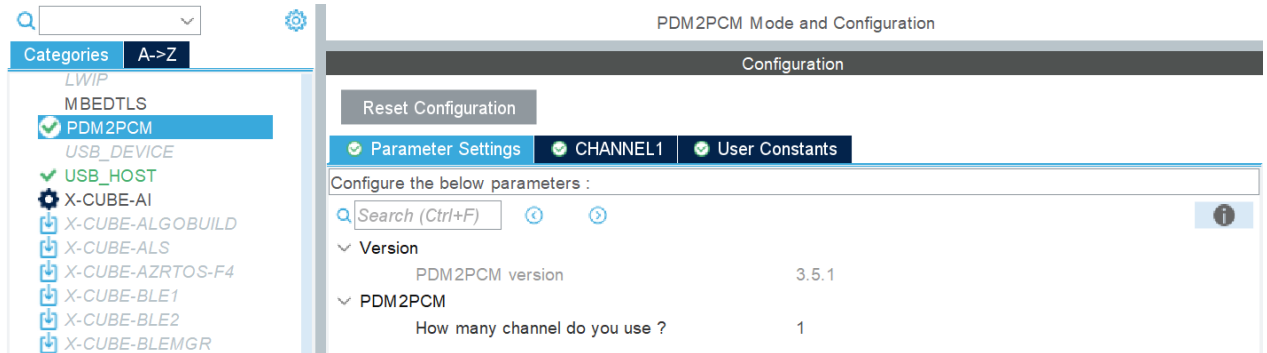


Рис. 2.20. Конфігурація пакета PDM2PCM для обробки аудіосигналу

Альтернативним підходом до розгортання бібліотеки PDM2PCM є її пряма інтеграція у готовий проєкт. Після первинної генерації файлової структури за допомогою CubeMX, подальша робота переноситься в IDE (наприклад, STM32CubeIDE). У дерево проєкту додається заголовковий файл та відповідна статична бібліотека. Для цільової платформи STM32F407xx використовується файл дефініцій `pdm_filter.h`. Дистрибутив містить оптимізовані збірки бінарних файлів для різних середовищ компіляції: `libPDMFilter_CM4F_IAR.a` – для IAR; `libPDMFilter_CM4F_Keil.lib` – для ARMCC (Keil); `libPDMFilter_CM4F_GCC.a` – для крос-компіляторів GNU.

Дистрибутив бібліотеки PDM2PCM поставляється у складі фірмового пакета STM32Cube\_FW\_F4 (коренева папка `Middlewares\ST\STM32_Audio\Addons\PDM`). Обов'язковою умовою успішної лінковки проєкту є додавання директорії із заголовковим файлом `pdm_filter.h` та статичним бінарним файлом до списку інклюдів (Include Paths) в налаштуваннях компілятора IDE.

*Налаштування режиму USB для підключення зовнішніх носіїв.* Щоб підключити до STM32 звичайну флешку чи карту пам'яті, модуль USB має функціонувати як хост із підтримкою класу Mass Storage Class (MSC) та файлової системи FAT для зчитування папок і файлів. У середовищі CubeMX у

розділі USB\_OTG\_FS Mode and Configuration ми обираємо опцію Host Only. Також активується параметр Activate\_VBUS, що дозволяє подавати живлення на підключений накопичувач, який не має власного джерела енергії.

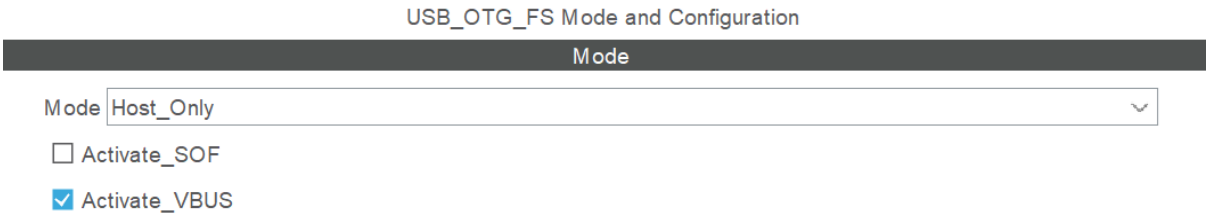


Рис. 2.21. Конфігурація модуля USB\_OTG\_FS у режимі Host Only

На наступному етапі в полі Class for FS необхідно встановити значення Mass Storage Host Class (як зображено на рисунку нижче).

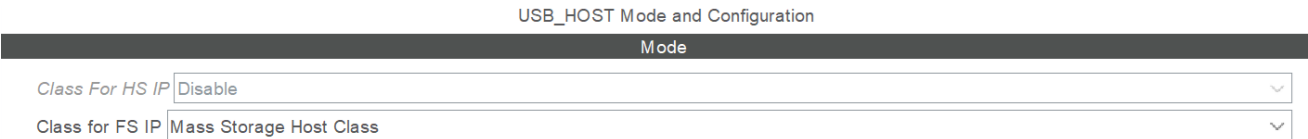


Рис. 2.22. Вибір класу Mass Storage Host для файлової системи

На завершення виконується конфігурування пакета FatFS. Для забезпечення сумісності із сучасними ємними накопичувачами вмикається підтримка формату exFAT, а також дозволяється використання довгих назв для файлів. Разом із цим верхню межу розміру сектора встановлено на рівні 4096.

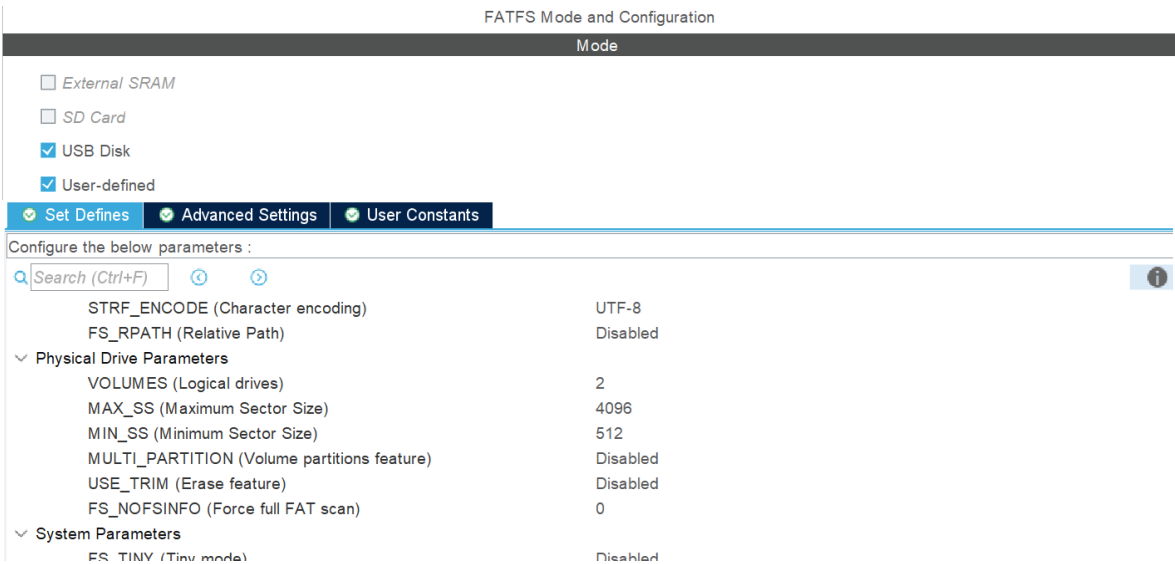


Рис. 2.23. Конфігурація параметрів файлової системи FatFS для підтримки сучасних носіїв

## РОЗДІЛ 3. АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ

### 3.1. Алгоритм зчитування PDM-даних з MEMS-мікрофона

Вихідний сигнал з MEMS-мікрофона формується у вигляді PDM-потoku — послідовності високочастотних 1-бітних цифрових значень. Бібліотека обробки PDM2PCM орієнтована на прийом даних, згрупованих по 8 блоків (по одному байту), які надходять від мікрофона через синхронну шину I2S мікроконтролера STM32F407VGTx. Передача даних з лінії DOUT мікрофона здійснюється синхронно з тактовими імпульсами на вході CLK. Генерацію цього опорного тактового сигналу забезпечує апаратний модуль I2S мікроконтролера (Рис. 3.1).

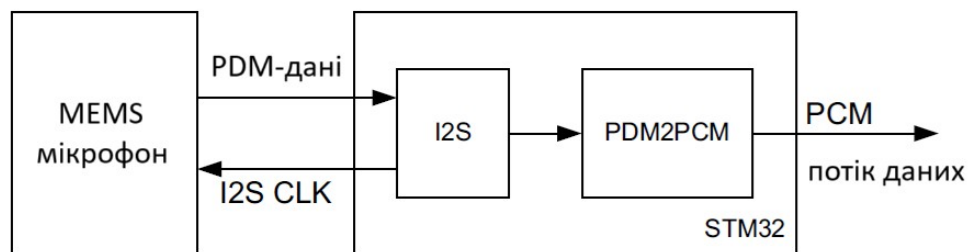


Рис. 3.1. Структурна організація процесу перетворення PDM-сигналу в PCM за допомогою програмного пакета PDM2PCM

Однобітні аудіодані у форматі PDM групуються в байти (по 8 біт), а потім проходять етапи фільтрації та зменшення частоти дискретизації. Графічне представлення цього процесу наведено на Рис.3.2.



Рис. 3.2. Графічне представлення етапів фільтрації та децимації PDM-сигналу

За допомогою формату PDM здійснюється перетворення звукового аналогового сигналу на цифровий вигляд. У цьому потоці амплітуда хвилі не квантується на конкретні рівні, як це реалізовано в імпульсно-кодовій модуляції (PCM). Замість цього інформацію про гучність чи форму сигналу

несе густина розміщення імпульсів. Щоб перетворити такий бітовий потік на звичні кадри даних, використовують алгоритми проріджування. Стартовим кроком такої обробки виступає процедура усереднення за методом ковзного середнього (Рис.3.3).

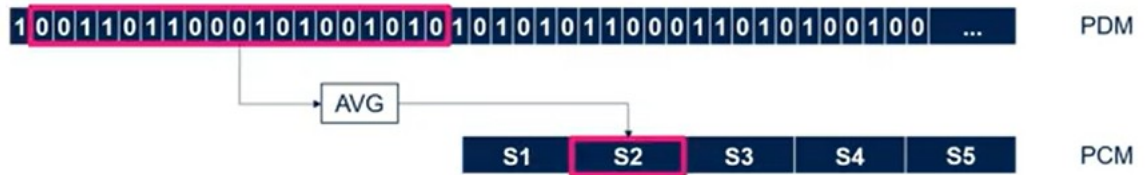


Рис. 3.3. Реалізація першого етапу децимації: обчислення ковзного середнього

На наступному, другому етапі сигнал проходить крізь низькочастотний фільтр. Графічне відображення цього процесу представлено на Рис.3.4.

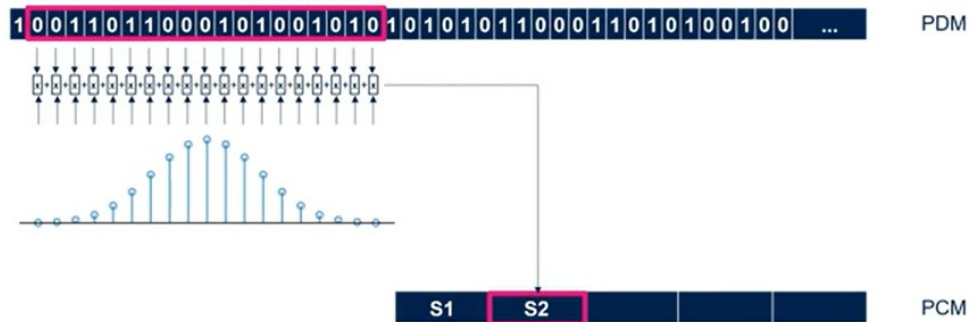


Рис. 3.4. Очищення звукового сигналу від залишків високочастотного шуму за допомогою ФНЧ

Далі виконується операція проріджування, яка дозволяє зменшити частоту дискретизації сигналу. Потім за допомогою фільтра високих частот із сигналу видаляється постійна складова (Рис. 3.5). На цьому ж рисунку наведено приклад децимації з коефіцієнтом  $M=3$ : система залишає лише одну вибірку з кожних  $M$  штук, а інші просто видаляє. Таким чином, нова частота дискретизації стає рівною початковій частоті, поділеній на коефіцієнт  $M$ .

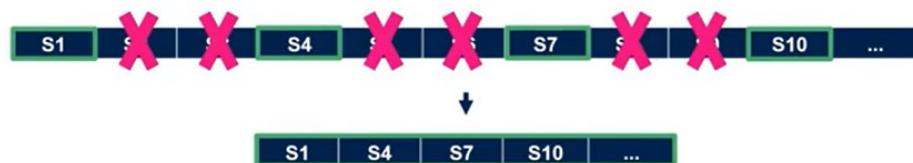


Рис. 3.5. Алгоритм зменшення частоти дискретизації та фільтрації зміщення

Частотні параметри отриманого PCM-сигналу залежать від обраного коефіцієнта проріджування, який вказується перед початком роботи бібліотеки. Конкретні значення тактових частот для PDM-мікрофона та необхідні коефіцієнти для отримання бажаної частоти дискретизації вихідного звуку зведені в Табл. 3.1.

*Таблиця 3.1 Співвідношення коефіцієнтів децимації та частотних параметрів інтерфейсу*

| Коефіцієнт проріджування | Тактова частота PDM-мікрофона, кГц | Частота дискретизації PCM, кГц |
|--------------------------|------------------------------------|--------------------------------|
| 128                      | 1024                               | 8                              |
|                          | 2048                               | 16                             |
|                          | 3072                               | 24                             |
| 80                       | 1280                               | 16                             |
| 64                       | 1024                               | 16                             |
|                          | 2048                               | 32                             |
|                          | 3072                               | 48                             |
| 48                       | 768                                | 16                             |
| 32                       | 512                                | 16                             |
| 24                       | 384                                | 16                             |
| 16                       | 256                                | 16                             |

Далі цифровий сигнал, що пройшов етапи фільтрації та проріджування, надходить на фільтр високих частот, який прибирає зміщення постійної складової. Після цього він обробляється цифровим підсилювачем, який дозволяє гнучко регулювати рівень гучності (ослаблювати чи підсилювати) результуючого PCM-сигналу. Функцію високочастотного фільтра виконує однополосний рекурсивний фільтр. Щоб задати частоту зрізу, використовується спеціальний параметр `high_pass_tap` дескриптора `PDM_Filter_Handler_t`. При конфігуруванні системи важливо враховувати, що це значення має лежати в межах інтервалу  $[0:1]$ .

Формат представлення даних Q0.31 передбачає, що одиниця є граничним максимумом, а вся шкала кодується 31 бітом. Наприклад, обравши коефіцієнт `high_pass_tap` рівним 0,98, ми отримаємо його машинне значення як  $0,98 \times (2^{31} - 1) = 2104533974$ . Графіки амплітудно-частотних характеристик фільтра для трьох відмінних значень цього параметра при стандартній частоті дискретизації вихідного PCM-звуку 16 кГц відображено на Рис.3.6.

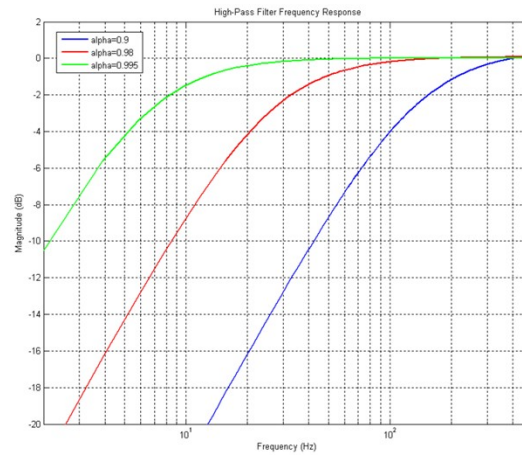


Рис. 3.6. Амплітудно-частотна характеристика ФВЧ при різних значеннях параметра `high_pass_tap`

Якщо вказати нульове значення коефіцієнта `high_pass_tap`, фільтр високих частот буде вимкнено. Далі блок цифрової гучності зменшує або збільшує значення семплів, після чого вони приводяться до 16-бітного знакового типу. Для керування рівнем підсилення всього PCM-поток використовується параметр `mic_gain`, який задається в децибелах. Користувач може змінювати його в межах від -12 дБ до 51 дБ із дискретністю в 1 дБ.

Якщо встановити коефіцієнт `high_pass_tap` рівним нулю, функція високочастотної фільтрації вимикається. Далі алгоритм цифрової гучності зменшує або збільшує значення аудіосемплів перед тим, як обмежити їх до допустимого діапазону 16-бітного знакового числа. Для керування рівнем гучності всього PCM-поток використовується параметр `mic_gain`, що вимірюється в dB. Користувачеві доступне налаштування в межах від -12 dB до 51 dB із кроком в 1 dB.

Формат вихідних даних з MEMS-мікрофона MP45DT02 представляє собою високочастотний PDM-потік однорозрядних семплів із частотою від 1 МГц до 3,25 МГц. Зняття цих даних блоками по 8 біт здійснюється через синхронний послідовний порт SPI або I2S, інтегрований у процесор STM32F407VGTx. Видача інформації з лінії даних мікрофона прив'язана до фронтів тактового сигналу CLK. У свою чергу, генерування цього керуючого сигналу покладається на внутрішню периферію SPI/I2S мікроконтролера.

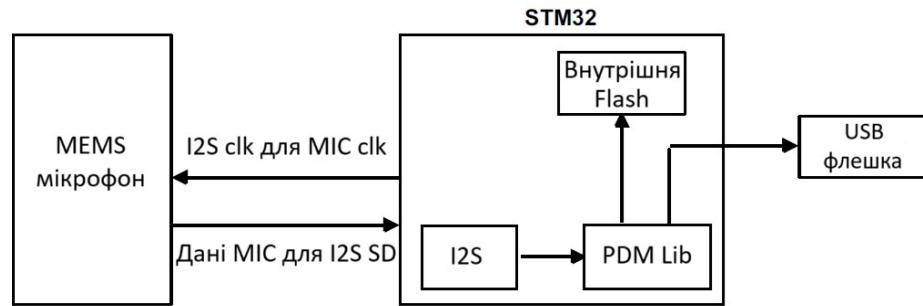


Рис. 3.7. Схема структурна сполучення MEMS-мікрофона з МК STM32

Потік із мікрофона надходить на систему проріджування. Вона складається зі спеціального децимуючого фільтра, який переводить однобітні PDM-дані у формат PCM, та двох індивідуально налаштованих IIR-фільтрів (ФНЧ та ФВЧ). На виході ми отримуємо реконструйований звук у 16-бітному PCM-форматі. Такі дані готові до подальшого використання за алгоритмом розробника: їх можна зберегти на флеш-пам'ять як .WAV аудіофайл або передати на аудіокодек (ЦАП) через шину I2S.

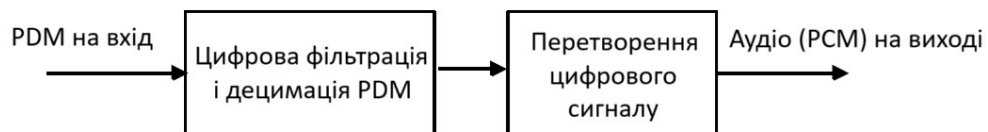


Рис. 3.8. Структурна схема конвеєра цифрової обробки PDM-сигналу

Для отримання звуку з потрібними параметрами частоти й точності, вихідний PDM-сигнал підлягає фільтрації та проріджуванню. При цьому частота зчитування даних із мікрофона (що керується сигналом на лінії CLK) повинна перебувати в кратному співвідношенні з фінальною частотою звуку. Наприклад, якщо коефіцієнт децимації дорівнює 80, а цільова частота становить 30 кГц, то генератор повинен тактувати мікрофон на частоті 2,4 МГц. Після завершення всієї обробки ми отримуємо 16-бітне число в діапазоні від -32768 до 32767 (якщо встановлено нульове підсилення 0 dB).

Цифровий сигнал, отриманий після попередньої фільтрації, проходить фінальне узгодження. На першому етапі задіюється фільтр високих частот, головна задача якого – прибрати з сигналу постійну складову. Він реалізований як IIR-фільтр із частотою зрізу, що знаходиться за межами

чутного людиною звукового діапазону. Таке рішення гарантує збереження вихідної якості аудіопотоку.

На другому етапі застосовується фільтр низьких частот, який також базується на IIR-фільтрі. Програма дозволяє користувачеві вимикати чи активувати обидва фільтри та виставляти потрібні частоти зрізу в параметрах ініціалізації. Величину підсилення можна змінювати через спеціальну змінну `MicGain` за наступною формулою:

$$G = \frac{MicGain}{64} \quad (3.1)$$

До складу пакета PDM2PCM входить керуюча структура `PDMFilter_InitStruct` та 4 спеціалізовані функції фільтрації. Робота бібліотеки спирається на два масиви, які створюються в основному коді: один призначений для вхідного PDM-сигналу, а інший – для вихідного PCM-звуку. масив `data` містить вхідні байти (тип `unsigned char/uint8_t`). Його довжина відповідає об'єму даних за 1 мс і визначається множенням тисячної частини вихідної частоти на кратність проріджування та на число підключених мікрофонів (із подальшим діленням на 8 для переведення бітів у байти); масив `dataOut` зберігає готові відліки (тип `unsigned short/uint16_t`). Його розмір дорівнює тисячній частині вихідної частоти, помноженій на кількість каналів.

Оголошена у файлі `pdm_filter.h` структура `PDMFilter_InitStruct` служить для базового налаштування цифрового фільтра. Її програмна реалізація виглядає так:

```
typedef struct{
    uint16_t Fs;
    float LP_HZ;
    float HP_HZ;
    uint16_t Out_MicChannels;
    char InternalFilter[34];
} PDMFilter_InitStruct;
```

У цьому блоці даних змінна `Fs` відповідає за частоту дискретизації вихідного PCM-потoku. Параметри `LP_HZ` та `HP_HZ` типу `float` визначають частоти зрізу для низько- та високочастотних алгоритмів відповідно. Поле

`Out_MicChannels` вказує на число каналів запису, а масив `InternalFilter` виділяє 34 байти для службових потреб обчислювального ядра бібліотеки.

Однобітні PDM-семпли збираються по 8 штук і представляються у виді звичайного 8-бітного байта. Змінна `Out_MicChannels` вказує на кількість мікрофонів у системі та потрібна для того, щоб правильно чергувати дані від різних джерел у фінальному буфері. Готовий вихідний звуковий семпл займає 16 біт пам'яті. Для внутрішніх розрахунків під час проріджування даних бібліотека самостійно використовує зарезервовану область пам'яті `InternalFilter` розміром 34 байти.

До складу пакета PDM2PCM входить єдиний заголовний файл `pdm_filter.h` та попередньо скомпільовані бінарні файли, адаптовані під різні інструментальні середовища: для компілятора IAR передбачено файл статичної бібліотеки `libPDMFilter_IAR.a`; для компілятора ARM (Keil) постачається бібліотечний файл `libPDMFilter_Keil.lib`; для вільного компілятора GNU (GCC) використовується бінарний файл `libPDMFilter_GCC.a`.

Початкове налаштування фільтра для одиночного мікрофона покладається на функцію `PDM_Filter_Init`. Вона зчитує вихідні дані зі структури `PDMFilter_InitStruct` та проводить внутрішні розрахунки. Типовий лістинг коду для ініціалізації системи на частоту 16 кГц має такий вигляд:

```
PDMFilter_InitStruct Filter; Filter.LP_HZ = 8000.0f; Filter.HP_HZ = 10.0f; Filter.Fs = 16000;
Filter.Out_MicChannels = 1; PDM_Filter_Init(&Filter);
```

Перш ніж викликати ініціалізацію фільтра `PDM_Filter_Init`, необхідно увімкнути тактування вбудованого апаратного блоку CRC в мікроконтролері. Нехтування цим правилом призведе до того, що функція поверне помилку, а мікроконтролер перейде до виконання нескінченного циклу, повністю зупинивши роботу пристрою.

Для перетворення PDM-даних з одного мікрофона використовуються функції виду `PDM_Filter_xx_xx`. Вони повертають готові PCM-семпли. Їхня кількість на виході відповідає тисячній частині від частоти дискретизації, за-

даної при ініціалізації. Якщо обрана частота не ділиться націло на 1000 (як у випадку з 44100 Гц або 22050 Гц), то кількість відліків просто округлюється вниз – до 44 та 22 відповідно. Базовий прототип цієї функції записується так: `uint32_t PDM_Filter_XX_XX(uint8_t* data, uint16_t* dataOut, uint16_t MicGain, PDMFilter_InitStruct * Filter)`. Аргументи функції обробки звуку мають наступне призначення: `data` – буфер із сирими PDM-даними (передається адреса першого семпла для обробки); `dataOut` – буфер для збереження вихідних даних, прорахованих фільтром `PDM_Filter`; `MicGain` – коефіцієнт гучності (від 0 до 64), що керує підсиленням звукової хвилі; `Filter` – дескриптор структури з усіма параметрами, які раніше були передані у функцію `PDM_Filter_Init`.

Бібліотека містить 4 різні версії цієї функції, що дозволяє обрати оптимальний коефіцієнт проріджування (64 або 80) та порядок розміщення бітів у байті (LSB чи MSB).

### 3.2. Алгоритм обробки звукових даних у реальному часі

Реалізація алгоритму обробки звуку в реальному часі базується на використанні бібліотеки `PDM2PCM`, яка відповідає за конвертацію однобітного PDM-потoku з мікрофона в багаторозрядний PCM-звук. Цей інструмент входить до стандартного середовища розробки `STM32CubeIDE`. Бібліотека є універсальною та підтримує роботу з PDM-давачами на базі потужних лінійок мікроконтролерів, зокрема `STM32F4x`, `STM32F7x` та `STM32H7x`.

Бібліотека `PDM2PCM` містить спеціальну функцію фільтрації та проріджування звукового потоку, яка переводить PDM-дані з цифрового MEMS-мікрофона у PCM-формат. На виході ми отримуємо звук із роздільною здатністю 16 біт та частотою дискретизації 16 кГц. Програма дозволяє легко підлаштовуватися під різні частоти зчитування PDM-даних завдяки можливості тонкого налаштування коефіцієнтів проріджування, програмного підсилення та високочастотної фільтрації.

Бібліотека `PDM2PCM` розрахована на обробку однобітних PDM-даних із частотою проходження від 768 кГц до 2048 кГц. Окремі однобітні семпли

об'єднуються в байти (по 8 штук) завдяки синхронним послідовним портам SPI або I2S на борту STM32. Існують спеціальні версії цієї бібліотеки для різних ядер та середовищ розробки. Дані про те, скільки пам'яті вимагає бібліотека при різних тактових частотах мікрофона, зведені в Табл.3.2.

*Таблиця 3.2. Вимоги бібліотеки PDM2PCM до пам'яті залежно від частоти мікрофона*

| Частота мікрофона    | Flash (код)<br>(байт) | Flash (дані)<br>(байт) | Stack<br>(байт) | RAM<br>(байт) | Частота<br>(МГц) |
|----------------------|-----------------------|------------------------|-----------------|---------------|------------------|
| 2,048 МГц (Кдец=128) | 7020                  | 789                    | 50              | 1028          | 4,9              |
| 1,280 МГц (Кдец=80)  | 6350                  | 620                    | 50              | 850           | 3,4              |
| 1,024 МГц (Кдец=64)  | 5900                  | 510                    | 50              | 680           | 2,7              |
| 768 кГц (Кдец=48)    | 5100                  | 430                    | 50              | 520           | 2,3              |
| 512 кГц (Кдец=32)    | 4400                  | 350                    | 50              | 390           | 1,8              |

Щоб підключити бібліотеку PDM2PCM, знадобляться два обов'язкові файли: інтерфейсний `pdm2pcm_glo.h` та скомпільований бінарний файл, обраний під вашу платформу та середовище. Вони містять готові структури для експорту в проект. Наприклад, як загальний заголовний файл для аудіомодулів у складі фреймворку часто використовується `audio_fw_glo.h`. Сама бібліотека надає розробнику 5 API-функцій: `PDM_FilterInit`, `PDM_Filter_setConfig`, `PDM_Filter_getConfig`, `PDM_Filter_deInterleave` та головну функцію `PDM_Filter`.

Завданням процедури `uint32_t PDM_Filter_Init(PDM_Filter_Handler_t *pHandler)` є первинна підготовка пам'яті для модуля PDM2PCM, генерація таблиць пошуку та виставлення дефолтних налаштувань. Опис аргументів цієї функції представлений у Табл. 3.3.

*Таблиця 3.3. Специфікація виклику PDM\_Filter\_Init*

| I/O                | Назва                 | Тип                                 | Опис                                   |
|--------------------|-----------------------|-------------------------------------|----------------------------------------|
| Вхід               | <code>pHandler</code> | <code>PDM_Filter_Handler_t *</code> | Адреса дескриптора у статичній пам'яті |
| Повернене значення | -                     | <code>uint32_t</code>               | Статус виконання (код помилки)         |

Для налаштування параметрів роботи фільтра у реальному часі використовується функція `PDM_Filter_setConfig`. Вона зчитує дані зі структури

PDM\_Filter\_Config\_t і застосовує їх до поточного сеансу обробки. Важливо, що розробник може викликати цю функцію в будь-який момент, що дозволяє реалізувати, наприклад, адаптивне підсилення звуку.

*Таблиця 3.4. Опис аргументів PDM\_Filter\_setConfig*

| I/O                | Назва    | Тип                    | Опис                                          |
|--------------------|----------|------------------------|-----------------------------------------------|
| Вхід               | pHandler | PDM_Filter_Handler_t * | Адреса об'єкта обробника у статичній пам'яті  |
| Вхід               | pConfig  | PDM_Filter_Config_t *  | Адреса структури з актуальними налаштуваннями |
| Повернене значення | -        | uint32_t               | Результат виконання (числовий код помилки)    |

Для отримання актуальних налаштувань фільтра використовується функція PDM\_Filter\_getConfig. Вона витягує діючі параметри із захищеної області пам'яті обробника та копіює їх у призначену для користувача структуру PDM\_Filter\_Config\_t. Опитувати стан системи за допомогою цієї функції можна прямо під час запису звуку. Параметри та результати роботи процедури подано в Табл.3.5.

*Таблиця 3.5. Параметри функції PDM\_Filter\_getConfig*

| I/O                | Назва    | Тип                    | Опис                                       |
|--------------------|----------|------------------------|--------------------------------------------|
| Вхід               | pHandler | PDM_Filter_Handler_t * | Адресний покажчик на об'єкт обробника      |
| Вихід              | pConfig  | PDM_Filter_Config_t *  | Буферна структура для приймання параметрів |
| Повернене значення | -        | uint32_t               | Статус виконання операції                  |

Процедура PDM\_Filter\_deInterleave призначена для сортування звукових відліків за каналами. Вона приймає масив, де дані записані впереміш (наприклад, стереосигнал), і розділяє їх на самостійні потоки для подальшої індивідуальної обробки. Параметри виклику функції подано в Табл.3.6.

*Таблиця 3.6. Специфікація виклику PDM\_Filter\_deInterleave*

| I/O   | Назва    | Тип                    | Опис                                           |
|-------|----------|------------------------|------------------------------------------------|
| Вхід  | pDataIn  | void *                 | Адреса вхідного змішаного буфера               |
| Вихід | pDataOut | void *                 | Адреса вихідного буфера для розділених каналів |
| Вхід  | pHandler | PDM_Filter_Handler_t * | Адреса об'єкта обробника у пам'яті             |

|                    |   |          |                                 |
|--------------------|---|----------|---------------------------------|
| Повернене значення | — | uint32_t | Статус завершення (код помилки) |
|--------------------|---|----------|---------------------------------|

Функція `PDM_Filter` є основним обчислювальним ядром, яке трансформує потік PDM-даних на PCM-відліки. Програма повинна звертатися до неї циклічно для обробки кожного нового кадру звукової інформації. Специфікацію вхідних та вихідних параметрів описано в Табл.3.7.

*Таблиця 3.7. Параметри функціонального блоку PDM\_Filter*

| I/O                | Назва    | Тип                    | Опис                               |
|--------------------|----------|------------------------|------------------------------------|
| Вхід               | pDataIn  | void *                 | Адреса вхідного буфера PDM         |
| Вихід              | pDataOut | void *                 | Адреса вихідного буфера PCM        |
| Вхід               | pHandler | PDM_Filter_Handler_t * | Адреса об'єкта обробника у пам'яті |
| Повернене значення | -        | uint32_t               | Статус завершення (код помилки)    |

Перелік кодів, які може повернути модуль `PDM2PCM` у разі виникнення помилок, представлено в Табл.3.8. Оскільки кожному типу збою відповідає окремий біт, у підсумковому значенні може накопичитися інформація про кілька помилок одночасно.

*Таблиця 3.8. Значення та інтерпретація помилок модуля PDM2PCM*

| Назва помилки                   | Значення | Функціональний зміст                                     |
|---------------------------------|----------|----------------------------------------------------------|
| PDM_FILTER_NO_ERROR             | 0x0000   | Робота завершилася без помилок                           |
| PDM_FILTER_ENDIANNESS_ERROR     | 0x0001   | Формат порядку байтів не підтримується                   |
| PDM_FILTER_BIT_ORDER_ERROR      | 0x0002   | Задано непідтримуваний порядок бітів                     |
| PDM_FILTER_CRC_LOCK_ERROR       | 0x0004   | Захист: цільовий процесор не належить до сімейства STM32 |
| PDM_FILTER_DECIMATION_ERROR     | 0x0008   | Помилковий або непідтримуваний коефіцієнт децимації      |
| PDM_FILTER_GAIN_ERROR           | 0x0040   | Недопустиме значення підсилення сигналу                  |
| PDM_FILTER_SAMPLES_NUMBER_ERROR | 0x0080   | Некоректно вказана кількість аудіосемплів                |

*Структура статичних параметрів бібліотеки PDM2PCM.* Для збереження внутрішнього стану, налаштувань фільтрів та виділення буферної пам'яті в бібліотеці `PDM2PCM` використовується структура

`PDM_Filter_Handler_t`. Її конфігурують перед початком роботи системи.

Програмний код оголошення структури:

```
typedef struct {
    uint16_t bit_order;      // Порядок бітів
    uint16_t endianness;    // Порядок байтів
    uint32_t high_pass_tap;  // Коефіцієнт ФВЧ
    uint16_t in_ptr_channels; // Вхідні канали
    uint16_t out_ptr_channels; // Вихідні канали
    // Буфер статичної пам'яті для розрахунків
    uint32_t pInternalMemory[PDM2PCM_INTERNAL_MEMORY_SIZE];
} PDM_Filter_Handler_t;
```

Функціональне призначення елементів структури `PDM_Filter_Handler_t` систематизовано в Табл.3.9. Параметризація даних полів є критичною для коректної детермінації вхідного сигналу та розподілу пам'яті.

*Таблиця 3.9. Опис елементів статичної конфігурації PDM2PCM*

| Параметр                      | Функціональне призначення              | Специфікація значення                                |
|-------------------------------|----------------------------------------|------------------------------------------------------|
| <code>bit_order</code>        | Бітова інверсія вхідного потоку        | Визначає пріоритетність бітів (за замовчуванням LSB) |
| <code>endianness</code>       | Схема адресації байтів у пам'яті       | Визначає порядок слідування байтів (Little Endian)   |
| <code>high_pass_tap</code>    | Характеристика рекурсивного ФВЧ        | Коефіцієнт у форматі числа з фіксованою комою Q31    |
| <code>in_ptr_channels</code>  | Розмірність вхідного вектору каналів   | Ціле число (наприклад, 1 для монофонічного запису)   |
| <code>out_ptr_channels</code> | Розмірність вихідного вектору каналів  | Ціле число (наприклад, 1 для монофонічного запису)   |
| <code>pInternalMemory</code>  | Адресний простір для внутрішніх потреб | Показчик на виділений буферний масив                 |

*Структура динамічних параметрів `PDM_Filter_Config_t`.*

Налаштування модуля `PDM2PCM` можна змінювати “на льоту”, оновлюючи дані в структурі динамічних параметрів. Після внесення змін необхідно викликати функцію `PDM_Filter_setConfig`, щоб нові конфігурації вступили в дію. Програмна реалізація структури:

```
typedef struct {
    uint16_t decimation_factor; // Фактор децимації
    uint16_t output_samples_number; // Число семплів на виході
    int16_t mic_gain; // Цифрове підсилення
} PDM_Filter_Config_t;
```

У Табл.3.10 систематизовано параметри динамічної конфігурації обчислювального модуля. Регулювання даних полів забезпечує гнучкість системи при зміні акустичних умов або режимів енергоспоживання.

Таблиця 3.10. Характеристики динамічних налаштувань фільтрації

| Параметр              | Функціональне призначення              | Специфікація та межі значень                                  |
|-----------------------|----------------------------------------|---------------------------------------------------------------|
| decimation_factor     | Коефіцієнт децимації вхідного потоку   | Набір констант: від 16 до 64 (визначає частоту вихідного PCM) |
| output_samples_number | Потужність вихідного вектора семплів   | Натуральне число (кількість відліків на один фрейм обробки)   |
| mic_gain              | Масштабування амплітуди вихідного коду | Діапазон [-12 дБ: +51 дБ] з дискретністю 1 дБ                 |

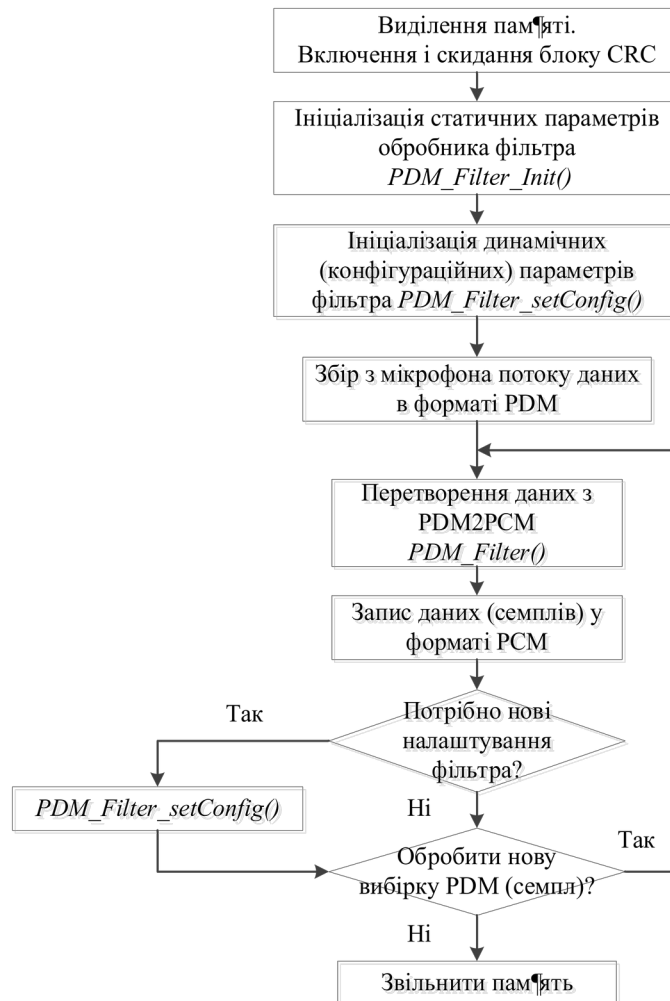


Рис. 3.9. Алгоритм функціонування програмного модуля PDM2PCM для обробки звукових даних

Вхідним інтерфейсом для програмного пакета PDM2PCM слугує послідовність PDM-даних, сформована цифровим мікрофоном відповідно до частоти його тактування. Дані передаються у вигляді байтових пакетів (одинарним або подвійним потоком). На виході алгоритм формує аудіодані у

PCM-форматі, які надалі можуть бути збережені у стандартному звуковому контейнері WAV.

WAV – це поширений формат аудіофайлів, що забезпечує зберігання нестиснутого звукового потоку. Висока якість даних, ідентична оригінальним цифровим записам, зумовила його широке використання в професійному ПЗ для аудіоредагування. Протокол було розроблено компаніями IBM та Microsoft на основі контейнера RIFF, де дані організовані у вигляді структурованих фрагментів. Хоча формат офіційно іменується WAVE, частіше вживається аббревіатура за його розширенням – .wav. Вперше представлений у 1991 році, він пройшов ряд оновлень (останнє – у 2007 р.) і залишається базовим для систем, що потребують найвищої точності відтворення. Будову такого файлу наочно зображено на Рис. 3.10.

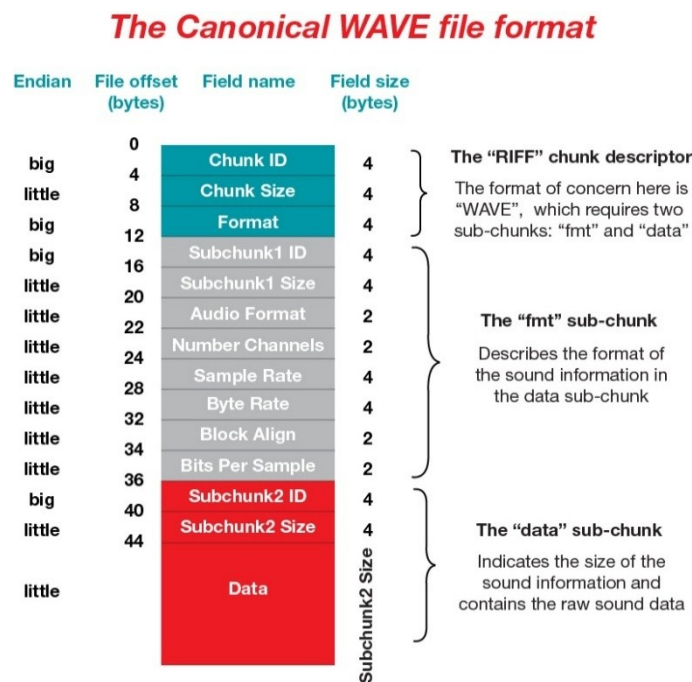


Рис. 3.10. Організація заголовків та блоків даних у структурі файлу формату WAV

Оскільки файл стандарту WAVE є окремим випадком реалізації контейнера RIFF, він будується за аналогічними принципами організації даних. На початку документа розташовується заголовок, за яким ідуть два критично важливі фрагменти. Перший із них – інформаційний блок `fmt`, що описує технічні характеристики сигналу, а другий – блок `data`, який об'єднує

ідентифікатор сегмента, значення його довжини в байтах та фактичний звуковий потік у нестиснутому вигляді.

Стандартна побудова файлу типу .wav, що містить нестиснутий PCM-звук, базується на таких принципах організації даних: 1. Зміщення (Offset): вказує на точну позицію байта, з якого починається опис певного поля (зокрема, поле *ChunkSize* займає позиції 04–07). 2. Порядок слідування байтів (Endianness): визначає ієрархію зберігання байтів у пам'яті (Little-endian або Big-endian) для значень, розмір яких перевищує 1 байт. 3. Фіксований розмір: стандартний заголовок такого файлу має довжину 44 байти.

## РОЗДІЛ 4

### ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ АУДІОФІКСАЦІЇ ПРАВОПОРУШЕНЬ

#### 4.1. Програмний модуль обробки звукових даних з використанням бібліотеки PDM2PCM

Для коректної роботи бібліотеки PDM2PCM у програмному модулі реалізовано алгоритм ініціалізації, що складається з таких кроків:

1. Встановлення конфігураційних параметрів за допомогою функції `PDM_Filter_setConfig()`. Це обов'язкова операція, яка проводиться перед початком обробки даних.
2. Безпосередня ініціалізація обробника через виклик `PDM_Filter_Init()`. Дана функція викликається при кожному запуску системи після зупинки, забезпечуючи готовність бібліотеки до роботи з новими даними.

Додатковою апаратною умовою використання пакета PDM2PCM є підключення блока обчислення контрольної суми (CRC). Даний метод широко застосовується в мікропроцесорних системах для контролю за відсутністю помилок у цифрових масивах. Завдяки використанню CRC стає можливим ефективно підтвердження цілісності пакетів даних у процесі їхнього транзиту чи довготривалого зберігання. Взаємодія з пакетом PDM2PCM здійснюється через спеціалізовані структури даних, визначені у файлі `pdm2pcm_glo.h`. На етапі підготовки системи до роботи виконується послідовна конфігурація обробника фільтрів. Зокрема, через структуру `PDM_Filter_Handler_t` задаються параметри вхідного потоку: порядок бітів (*bit order*), формат відліків (*endianness*), кількість каналів та коефіцієнт фільтра високих частот. Наступним кроком через структуру `PDM_Filter_Config_t` визначаються динамічні характеристики, такі як коефіцієнт децимації (у даному випадку 64), кількість вихідних семплів та рівень програмного підсилення мікрофона. Обов'язковою умовою є попередня активація модуля CRC через службу HAL.

Функціонування програмного блоку збору даних розпочинається лише після повної підготовки апаратних інтерфейсів та налаштування бібліотеки

перетворення. Алгоритм побудований на обробці переривань: як тільки в оперативній пам'яті формується масив бітів звукової інформації достатньої довжини, ініціюється виклик обробника PDM2PCM. Архітектура рішення дозволяє оперативно змінювати динамічні конфігураційні параметри між двома черговими циклами обробки сигналу.

```

/* Виконання фільтрації: перетворення PDM у PCM для поточного кадру */
PDM_Filter(&pdm_buffer[0], &pcm_buffer[0], &PDM1_filter_handler);
/* Оперативна корекція коефіцієнта підсилення мікрофона */
PDM1_filter_config.mic_gain = 12; // Встановлення рівня підсилення
/* Оновлення конфігурації фільтра для наступної ітерації обробки */
PDM_Filter_setConfig((PDM_Filter_Handler_t*) &PDM1_filter_handler,
&PDM1_filter_config);

} while (1); // Безперервний цикл обробки сигналу

```

Наведений алгоритм демонструє процес ітераційної обробки звукових кадрів. Основна процедура фільтрації (`PDM_Filter`) забезпечує трансформацію PDM-потoku в PCM-формат. Гнучкість системи підтверджується можливістю програмної модифікації параметрів «на льоту»: зміна коефіцієнта підсилення мікрофона виконується без зупинки основного циклу збору даних, що критично важливо для систем моніторингу в реальному часі.

*Опис алгоритму взаємодії з API-функціями бібліотеки PDM2PCM.* Процес обробки аудіоінформації в системі можна розділити на три фази: підготовчу, операційну та завершальну.

- Підготовча фаза (Кроки 1–3) охоплює програмно-апаратне розгортання: від активації апаратного блоку CRC та виділення буферів у RAM до встановлення параметрів децимації та фільтрації.
- Операційна фаза (Кроки 4–8) реалізує конвеєрну обробку звуку. Вона базується на безперервному прийомі даних через шину I2S2 та їхній трансформації за допомогою бібліотечних функцій PDM2PCM. Алгоритм передбачає гнучкість налаштувань (зміна гучності чи фільтрів) безпосередньо під час роботи.

- Завершальна фаза (Крок 9) гарантує коректне вивільнення обчислювальних ресурсів мікроконтролера після припинення сеансу аудіофіксації.

#### 4.2. Спеціалізований аудіодрайвер для запису та відтворення звуку

Для реалізації процесів фіксації та відтворення звукової інформації було використано спеціалізований аудіодрайвер BSP\_AUDIO. Інтерфейси керування драйвером, зокрема функції BSP\_AUDIO\_OUT\_Init() та BSP\_AUDIO\_OUT\_Play(), визначені у заголовному файлі stm32f4\_discovery\_audio.h, який інтегровано до складу програмного проєкту. Архітектура драйвера базується на рівні абстракції медіадоступу (MAL — Media Access Layer). Даний рівень об'єднує низькорівневі функції взаємодії з апаратними пристроями, що забезпечують трансляцію або програвання аудіопотоку. Ці функції реалізовані як локальні процедури (наприклад, I2S3\_Init()) безпосередньо у файлі реалізації stm32f4\_discovery\_audio.c.

*Недоліки та обмеження аудіодрайвера BSP\_AUDIO.* Експлуатація драйвера BSP\_AUDIO виявила певні недоліки, пов'язані з акустичними завадами. Зокрема, при недостатньо високій якості вхідних аудіоданих на виході динаміка може виникати значний рівень високочастотного шуму, що негативно впливає на розбірливість запису. Для нівелювання цього ефекту та підвищення чистоти вихідного сигналу доцільно використовувати частоти дискретизації, що перевищують 48 кГц. Проте слід враховувати компромісний характер такого рішення: зростання частоти дискретизації прямо пропорційно збільшує обсяг вихідного аудіофайлу, що підвищує вимоги до місткості накопичувача та швидкодії інтерфейсу запису.

*Особливості керування аудіокодеком CS43L22.* Процес конфігурування ЦАП CS43L22 через інтерфейс I2C1 вимагає особливої уваги до обробки переривань. Оскільки обмін даними по шині I2C є критичною операцією, виклик процедур переривань під час комунікації може призвести до її некоректного

завершення. Для запобігання конфліктам реалізовано механізм тимчасового блокування переривань перед початком сеансу зв'язку з ЦАП та їх розблокування після завершення транзакції. Такий захищений режим керування застосовується виключно на етапах ініціалізації (BSP\_AUDIO\_OUT\_Init), зупинки відтворення (BSP\_AUDIO\_OUT\_Stop) або під час оперативної зміни параметрів звуку (функції SetVolume, SetMute та SetOutputMode). Варто зауважити, що під час безпосередньої трансляції PCM-даних через інтерфейс I2S додаткова комунікація по шині I2C не проводиться, що мінімізує навантаження на процесор у режимі реального часу.

*Обмеження обробки заголовків аудіофайлів.* У поточній реалізації програмного забезпечення не впроваджено механізм динамічного аналізу (парсингу) структури аудіофайлів. Це означає, що система не проводить автоматичне визначення метаданих файлу, таких як режим відтворення (моно/стерео), загальний розмір даних, частота дискретизації чи довжина заголовка. Натомість використовується фіксована конфігурація параметрів, що відповідає специфікаціям розробленого пристрою. Такий підхід забезпечує стабільність роботи прототипу, проте вимагає суворої відповідності вхідних даних попередньо заданим установкам. До архітектурних особливостей реалізованого пристрою належать такі параметри:

1. Підтримка багатоканальності: Програмний модуль підтримує лише двоканальне (стерео) відтворення. У разі використання монофонічних джерел звуку алгоритм передбачає дублювання аудіоданих для обох каналів ЦАП. Це реалізується або подвійною відправкою даних на шину I2S, або попереднім перетворенням структури вихідного буфера.
2. Фіксована розрядність: Система працює виключно з 16-бітним представленням амплітуди сигналу. Даний формат оцифрування є оптимальним для обраної архітектури STM32F4 та забезпечує високу сумісність із бібліотекою PDM2PCM.

Архітектура драйвера BSP\_AUDIO передбачає використання функцій I2S2\_MspInit та I2S2\_Init для підготовки тракту зчитування звуку.

Функція `I2S2_MspInit` відповідає за ініціалізацію апаратно-залежних вузлів (MCU Support Package), тоді як `I2S2_Init` задає логічні параметри роботи шини I2S2, адаптуючи її до заданої частоти дискретизації `AudioFreq`. Розрахунок тактового сигналу для MEMS-сенсора виконується за формулою подвоєної частоти вибірки, що забезпечує коректну передачу PDM-даних.

Обробка помилок інтерфейсу I2S2, зокрема тих, що пов'язані з переповненням або спустошенням FIFO-буфера DMA, реалізована через `void HAL_I2S_ErrorCallback`.

*Обробка виняткових ситуацій інтерфейсу I2S2.* Для моніторингу та усунення апаратних збоїв у роботі інтерфейсу I2S2 використовується функція зворотного виклику `HAL_I2S_ErrorCallback`. Вона автоматично ініціюється при виникненні помилок у FIFO-буферах або при критичних збоях у роботі контролера прямого доступу до пам'яті (DMA). Реалізація цієї функції дозволяє програмно відстежувати некоректні стани прийому звукових даних та ініціювати процедури відновлення зв'язку з MEMS-мікрофоном без перезавантаження всього пристрою.

Конфігурація інтерфейсу I2S3 для роботи з ЦАП здійснюється за допомогою функцій `I2S3_MspInit` та `I2S3_Init`. Головним параметром тут є `AudioFreq` — він встановлює необхідну частоту дискретизації звуку.

Налаштування бібліотеки PDM2PCM здійснюється за допомогою функції `PDMDecoder_Init`. Її аргументи дозволяють встановити частоту дискретизації (`AudioFreq`) та визначити кількість каналів обробки (`Chnlnbr`): значення 1 відповідає монофонічному режиму, а 2 — стереофонічному.

*Функції `BSP_AUDIO` для відтворення звукових PCM-даних.* Для взаємодії МК STM32 із ЦАП CS43L22 застосовується функція `BSP_AUDIO_OUT_Init`. Вона здійснює конфігурацію апаратної периферії, зокрема інтерфейсів I2C, I2S, GPIO, а також налаштування DMA та переривань (IRQ).

Для ініціалізації первинного відтворення аудіосигналу використовується функція `BSP_AUDIO_OUT_Play`. Аргумент `pBuffer` передає адресу розміщення даних у пам'яті (вказівник на початок аудіофайлу або потоку), а параметр `Size` визначає загальний обсяг звукового буфера, що підлягає обробці. Для тимчасової зупинки відтворення аудіопотоку використовується функція `BSP_AUDIO_OUT_Pause`. Вона переводить процес виведення звуку в режим паузи без розриву з'єднання. Для продовження відтворення після використання паузи призначена функція `BSP_AUDIO_OUT_Resume`. Вона відновлює трансляцію аудіосигналу, який було раніше призупинено через `BSP_AUDIO_OUT_Pause`.

Керування гучністю здійснюється через функцію `BSP_AUDIO_OUT_SetVolume(uint8_t Volume)`. Діапазон: параметр задається від 0% до 100%.

Для встановлення частоти дискретизації ЦАП CS43L22 використовується функція `BSP_AUDIO_OUT_SetFrequency`. Її виклик проводиться після попередньої ініціалізації через `BSP_AUDIO_OUT_Init`. Параметр `AudioFreq` безпосередньо визначає частоту, з якою буде відтворюватися звуковий потік.

Для швидкого керування вимкненням звуку призначена функція `BSP_AUDIO_OUT_SetMute`. Вона приймає параметр `Cmd`, який визначає стан аудіовиходу: повне приглушення сигналу або повернення до попереднього рівня гучності.

Функціонування кожного режиму спирається на використання відповідних функцій зворотного виклику (Callbacks). Прототипи цих функцій, що мають загальний формат іменування `BSP_AUDIO_OUT_XXXCallback()`, містяться у заголовковому файлі `stm32f4_discovery_audio.h`.

Система сповіщень про стан буферів реалізована через механізм Callback-функцій. Зокрема, `BSP_AUDIO_OUT_HalfTransfer_Callback`

сигналізує про те, що перша частина звукових даних була успішно опрацьована. Коли завершується робота з другою частиною масиву, викликається `BSP_AUDIO_OUT_TransferComplete_Callback`. Для обробки критичних ситуацій, пов'язаних із помилками передачі або апаратної буферизації, передбачена функція `BSP_AUDIO_OUT_Error_Callback`.

Для керування поточними параметрами відтворення та станом периферії передбачено набір функцій драйвера `BSP_AUDIO_OUT`: `BSP_AUDIO_OUT_Stop (uint32_t Option)` повністю припиняє передачу аудіоданих. `BSP_AUDIO_OUT_SetVolume(uint8_t Volume)` та `BSP_AUDIO_OUT_SetMute()` відповідають за регулювання рівня звуку та його миттєве приглушення. `BSP_AUDIO_OUT_SetFrequency(uint32_t AudioFreq)` дозволяє змінити частоту дискретизації, а `BSP_AUDIO_OUT_SetOutputMode` – обрати режим виводу. Заголовковий файл `stm32f4_discovery_audio.h` містить у своїй фінальній частині описи API та функцій зворотного виклику драйвера `BSP_AUDIO`. Механізм взаємодії з буфером реалізований через два основні колбеки: `BSP_AUDIO_OUT_HalfTransfer_Callback`, що спрацьовує після завершення трансляції першої частини даних, та `BSP_AUDIO_OUT_TransferComplete_Callback`, який активується по закінченню передачі другої половини. У разі критичних збоїв на рівні периферії або помилок шини передачі даних, система ініціює виклик `BSP_AUDIO_OUT_Error_Callback`.

*Функції аудіодрайвера BSP\_AUDIO для запису звуку.* Для підготовки апаратних засобів до процесу захоплення звуку застосовується функція `BSP_AUDIO_IN_Init`. Вона здійснює базове налаштування модулів I2S, GPIO, DMA та обробку переривань (IRQ). Параметри функції: `AudioFreq`: визначає частоту вибірки звукового сигналу в Гц (наприклад, 16000 або 44100); `BitRes`: встановлює розрядність даних (16, 32 біти тощо); `ChnlNbr`: вказує на кількість використовуваних каналів.

Для запуску процесу захоплення звуку з мікрофона використовується функція `BSP_AUDIO_IN_Record`. Вона приймає вказівник на цільовий

масив (`pData`) та визначає обсяг пам'яті (`Size`), виділений під аудіодані. Опрацювання накопиченої інформації здійснюється через механізм переривань, для чого розробнику необхідно імплементувати функції зворотного виклику: `AUDIO_IN_RxHalfCpltCallback` (для першої частини буфера) та `BSP_AUDIO_IN_ReceiveComplete_Callback` (для повного заповнення).

Для завершення процесу захоплення звукового сигналу з мікрофона використовується функція `BSP_AUDIO_IN_Stop`. Вона забезпечує повну зупинку роботи периферії, відповідальної за вхідний аудіопотік.

Для тимчасового припинення процесу захоплення аудіосигналу використовується функція `BSP_AUDIO_IN_Pause`. Вона дозволяє поставити запис на паузу, зберігаючи поточні налаштування сесії.

Відновлення призупиненого аудіопотоку реалізується функцією `BSP_AUDIO_IN_Resume`. Змінити гучність вхідного потоку можна функцією `BSP_AUDIO_IN_SetVolume`. Вона встановлює програмний рівень гучності у межах від 0 до 100%. Перетворення аудіопотоку з PDM на PCM здійснюється за допомогою функції `BSP_AUDIO_IN_PDMPtoPCM`. Вона опрацьовує вхідний буфер `PDMBuf` і записує результат у `PCMBuf`. Більшість цифрових мікрофонів (як ті, що встановлені на платах STM32 Discovery) видають сигнал у форматі PDM, який неможливо відтворити чи обробити безпосередньо. Ця функція перетворює складний потік імпульсів у стандартні 16-бітні значення (PCM), з якими працює більшість аудіосистем.

Для реалізації механізму подвійної буферизації драйвер використовує колбек-функції `HalfTransfer` та `TransferComplete`. Ці функції викликаються МК у міру того, як DMA наповнює буфери звуковою інформацією. Це дозволяє програмісту почати обробку (наприклад, фільтрацію або конвертацію з PDM у PCM) однієї частини даних, поки інша частина ще записується. Функція `void BSP_AUDIO_IN_Error_Callback(void)` слугує обробником виняткових

ситуацій. Система ініціює її виклик автоматично, якщо під час роботи з мікрофоном виникає апаратна помилка або збій у процесі передачі даних.

На Рис. 4.1 представлена блок-схема, що ілюструє послідовність етапів збору та подальшої обробки звукової інформації.

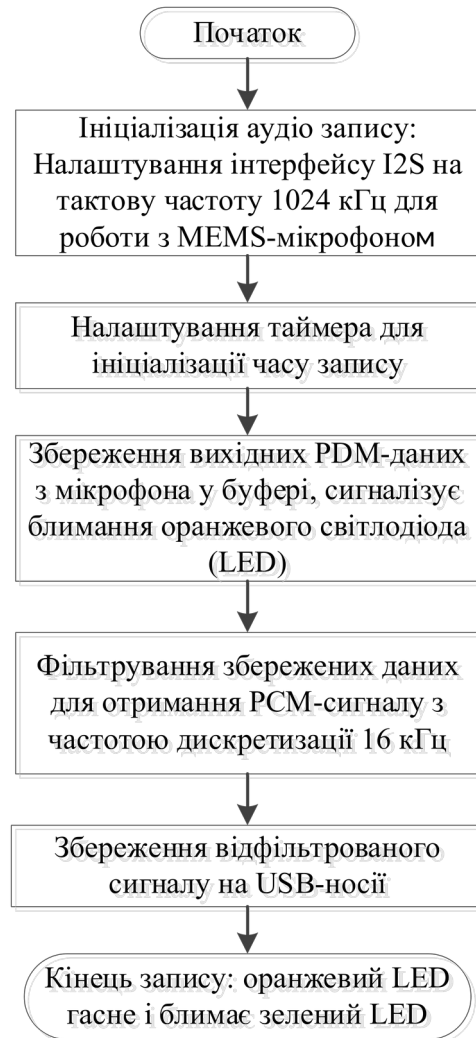


Рис. 4.1. Схема алгоритму збору та цифрової обробки аудіосигналу

#### 4.3. Аналіз роботи тестового макету цифрового пристрою аудіофіксації правопорушень

У досліджуваному макеті застосовано мікрофон MP45DT02, який генерує вихідний аудіосигнал у форматі щільностно-імпульсної модуляції (PDM). У даному потоці кожен біт інформації виступає як окремий відлік (семпл). Фізичний зміст амплітуди звукового сигналу в PDM-кодуванні полягає у щільності розподілу одиниць: що вищим є рівень амплітуди

вихідного аналогового сигналу, то більшою стає концентрація одиничних бітів у цифровій послідовності.

Зважаючи на те, що в PDM-сигналі рівень амплітуди звуку корелює зі щільністю одиниць, первинна обробка полягає в обчисленні ковзного середнього за певною вибіркою семплів. Для ілюстрації розглянемо процес усереднення в різних акустичних умовах: у стані спокою: для послідовності [0, 0, 0, 1, 0] середнє значення становить 0,2; за наявності інтенсивного шуму: для ряду [1, 0, 1, 1, 1] середній показник зростає до 0,8. Отримані в результаті усереднення цифрові значення дозволяють набагато точніше відтворити форму вихідного аналогового сигналу. Процес фільтрації PDM-потoku методом ковзного середнього базується на обчисленні середнього значення послідовності одnobітних вибірок. Для максимізації інформативності кожної отриманої величини необхідно, щоб кожна ітерація усереднення охоплювала унікальний сегмент PDM-даних. З цією метою вихідний сигнал з мікрофона піддається передискретизації (децимації) з відповідним коефіцієнтом кратності.

Алгоритм ковзного середнього для PDM-сигналу базується на знаходженні середнього арифметичного одnobітних значень. З метою максимізації інформативності кожної обробленої порції даних, процедуру усереднення слід застосовувати до унікальних сегментів бітового потоку. Реалізація такого підходу вимагає передискретизації (децимації) вхідного сигналу з мікрофона з відповідним коефіцієнтом, що забезпечує коректну обробку кожного набору семплів.

Розглянемо процес на прикладі: якщо вибірка PDM-семплів здійснюється на частоті, що вп'ятеро перевищує цільову частоту дискретизації, після обчислення ковзного середнього виконується децимація – збереження лише кожного п'ятого отриманого значення. Реалізація ковзного середнього фактично представляє собою роботу FIR-фільтра. При децимації з кроком 5 застосовується нормувальний множник 1/5. Для мінімізації спектральних спотворень, що виникають під час дискретизації, необхідно інтегрувати низькоча-

стотну фільтрацію. Оскільки ФНЧ може бути побудований як FIR-фільтр, він здатен повністю замінити алгоритм ковзного середнього. Ключовим моментом є розрахунок оптимальних вагових коефіцієнтів, які забезпечують необхідну смугу пропускання та відсікають завади у високочастотній області. У даному дослідженні замість базового осереднення застосовується класичне рівняння FIR-фільтрації для п'яти семплів з аналогічним коефіцієнтом проріджування.

Слід враховувати, що порядок FIR-фільтра не обов'язково має збігатися з коефіцієнтом пониження частоти дискретизації; він може бути значно більшим. Ключовим аргументом на користь розширення кількості семплів є необхідність підвищення крутизни зрізу фільтра низьких частот, що дозволяє мінімізувати вплив шумів та високочастотних завад на корисний сигнал.

Підсумовуючи алгоритм обробки, виділяють три базові етапи конвертації аудіопотоку з PDM у PCM. На практиці ці етапи часто скорочують до двох, поєднуючи фільтрацію низьких частот безпосередньо з децимацією (проріджуванням). Такий підхід базується на вибірковому обчисленні: система генерує лише ті значення, що відповідають цільовій частоті дискретизації, ігноруючи проміжні результати. Процедура децимації поєднує в собі антиаліасингову фільтрацію та зниження частоти оцифрування. Алгоритмічно це виглядає так: Етап 1: Зчитування сигналу з MEMS-мікрофона з надлишковою частотою (коефіцієнт  $x$ ). Етап 2: Обробка масиву даних за допомогою низькочастотного фільтра, де глибина аналізу охоплює не менше  $x$  відліків. Етап 3: Квантування за часом, що передбачає вилучення надлишкових даних і фіксацію лише кожного  $x$ -го результату фільтрації.

*Застосування модуля I2S2 для збору аудіоданих з мікрофона MP45DT02.* Організація збору аудіоданих через шину I2S2 дозволяє реалізувати стереофонічний режим роботи. Для цього пара мікрофонів MP45DT02 підключається до загальної лінії передачі. Якщо контакт вибору каналу (L/R) підключений до спільного проводу, мікрофон ідентифікується системою як лівий. Процес вибірки даних у цьому випадку синхронізований зі спадним

фронтом сигналу CLK, а під час зростаючого фронту мікрофонний вихід переходить у високоімпедансний стан, звільняючи шину.

Режим правого каналу активується шляхом подачі високого потенціалу (Vdd) на вхід вибору L/R. У цьому стані мікроконтролер фіксує інформаційні біти PDM-потоків в момент наростання тактового сигналу. Під час спаду тактового імпульсу мікрофон деактивує свою вихідну лінію, переводячи її у високоімпедансний стан. Детальна часова діаграма взаємодії компонентів ілюструється на відповідному графіку (Рис. 4.2). Для забезпечення коректного зчитування PDM-потоків мікроконтролером STM32 необхідно враховувати часові затримки, зазначені в технічній документації виробника (Табл. 4.1).

Таблиця 4.1. Електричні характеристики таймінгів MP45DT02

| Параметр | Визначення                              | Значення (min) | Значення (max) | Одиниці |
|----------|-----------------------------------------|----------------|----------------|---------|
| f_clk    | Тактування в основному робочому режимі  | 1,0            | 3,25           | MHz     |
| f_pd     | Поріг переходу в Power-down режим       | -              | 0,23           | MHz     |
| T_clk    | Період одного такту CLK                 | 308            | 1000           | ns      |
| T_R,en   | Час валідності даних для правого каналу | 30             | -              | ns      |
| T_R,dis  | Час деактивації виходу (правого)        | -              | 16             | ns      |
| T_L,en   | Час валідності даних для лівого каналу  | 30             | -              | ns      |
| T_L,dis  | Час деактивації виходу (лівого)         | -              | 16             | ns      |

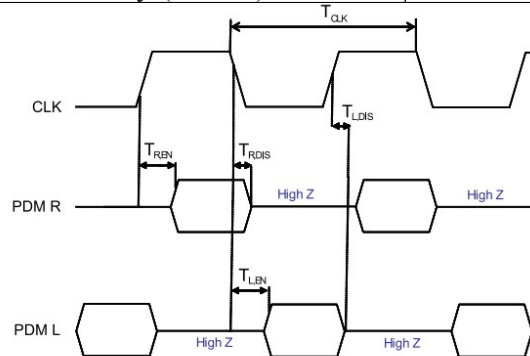


Рис. 4.2. Графічне представлення циклів зчитування PDM-даних

Рис. 4.2 ілюструє процеси синхронізації, характерні для інтерфейсу мікрофона MP45DT02. Діаграма демонструє критичні часові інтервали: затримку встановлення валідних даних після фронту CLK та час переходу виходу в неактивний стан. Таке візуальне представлення підтверджує можливість паралельного підключення двох датчиків для формування стереосигналу без використання додаткових ліній зв'язку.

Конструктивно на платі STM32F407G-DISC1 мікрофон MP45DT02 налаштований на роботу в режимі лівого каналу (вивід L/R підключений до GND). У цьому режимі вибірка даних мікроконтролером проводиться, коли тактовий сигнал має низький рівень. Така логіка роботи, де лінія даних залишається стабільною протягом напівперіоду CLK, є специфічною реалізацією, що не повністю збігається зі стандартним протоколом I2S.

Згідно зі специфікацією протоколу I2S, зміна інформаційного стану на лінії даних відбувається за спадним фронтом тактового сигналу, тоді як їх зчитування (фіксація) здійснюється за наростаючим фронтом. Такий механізм синхронізації забезпечує стабільність бітів у момент їх вибірки МК. Типова часова діаграма стандартного інтерфейсу I2S наведена на Рис. 4.3.

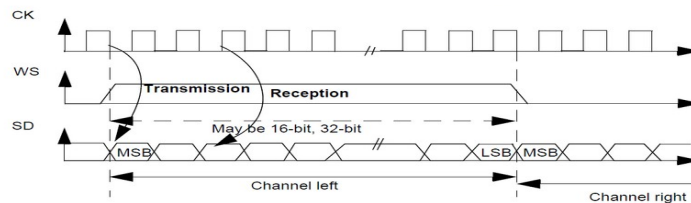


Рис. 4.3. Часова діаграма інтерфейсу I2S (MSB-вирівнювання, формат 16/32 біт, режим CPOL=0)

На часовій діаграмі Рис.4.3 зображено структуру передачі даних у стандарті I2S з активним низьким рівнем тактового сигналу в стані спокою (CPOL=0). Дані передаються, починаючи з найбільш значущого біта (MSB), при цьому формат кадру може бути адаптований для 16-бітної або 32-бітної розрядності. Таке налаштування забезпечує сумісність МК STM32 з типовими аудіокодеками та цифровими мікрофонами.

Застосування протоколу I2S для роботи з MP45DT02 вимагає ретельного врахування часових затримок. Згідно з документацією, мікрофону потрібно  $T_{DIS}=16$  нс для переходу лінії в стан високого імпедансу після позитивного перепаду тактування. У свою чергу, швидкість наростання сигналу з боку МК STM32F4 детермінована параметрами навантаження; зокрема, при ємності шини 30 пФ час наростання становить близько 4 нс. Ці часові співвідношення є критичними для запобігання конфліктам на шині даних.

Успішна вибірка даних через периферієльний модуль I2S можлива завдяки тому, що час реакції мікрофона MP45DT02 перевищує тривалість фронту тактування МК. За період, що становить понад 4 нс, мікроконтролер STM32F407VGT6 завершує цикл читання поточного стану шини даних. Це відбувається безпосередньо перед тим, як мікрофон почне очищення лінії, що запобігає втраті або спотворенню інформації в межах одного такту.

Для забезпечення необхідної якості звукозапису при аудіофіксації правопорушень було обрано наступні параметри цифрової обробки:

Таблиця 4.2. Параметри конфігурації CMSIS-фільтрації

| Параметр                 | Значення | Примітка                          |
|--------------------------|----------|-----------------------------------|
| Частота I2S CLK          | 1024 кГц | Тактування MP45DT02               |
| Частота зрізу ФНЧ        | 6 кГц    | Обмеження спектра мовного сигналу |
| Кількість вибірок (Taps) | 256      | Глибина FIR-фільтрації            |
| Коефіцієнт децимації     | 64       | Ступінь проріджування даних       |
| Вихідна частота PCM      | 16 кГц   | Цільова частота дискретизації     |

На Рис. 4.4 представлено структурну схему проекту. Графічно компоненти розділені за функціональним призначенням: блоки помаранчевого кольору у верхній частині ілюструють послідовні стадії обробки аудіосигналу, тоді як жовті прямокутники відображають конфігурацію системи тактування.

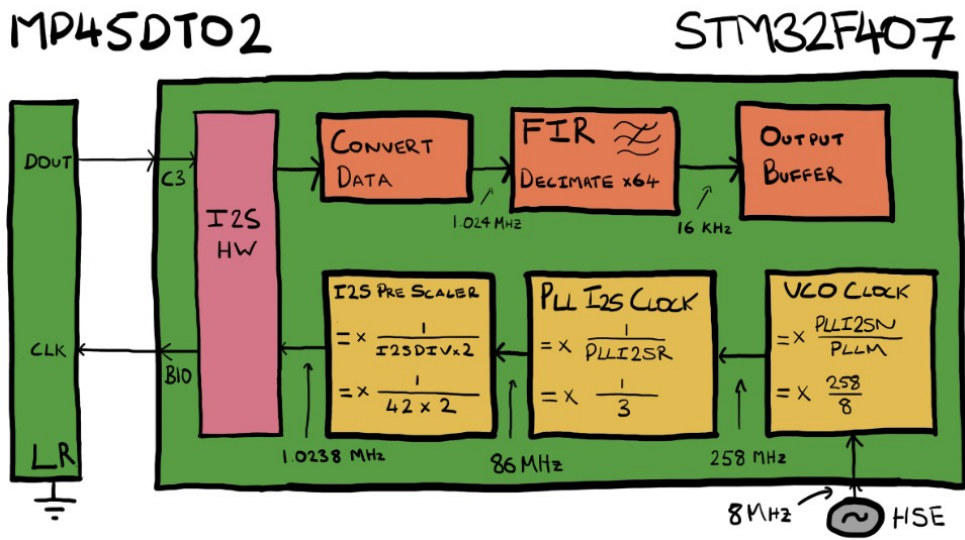


Рис. 4.4. Структурна схема конфігурації МК STM32F407 та алгоритму обробки сигналу мікрофона MP45DT02

Фізичне з'єднання MEMS-мікрофона MP45DT02 з МК STM32F407VGT6 реалізовано згідно зі схемотехнічними рішеннями, наведеними у Табл.4.3. Для забезпечення прийому аудіоданих задіяно периферійний модуль I2S2, конфігурація виводів якого відповідає альтернативним функціям обраних GPIO.

Таблиця 4.3. Схема інтерфейсного підключення мікрофона до МК

| Порт мікроконтролера | Сигнал інтерфейсу | Вивід мікрофона |
|----------------------|-------------------|-----------------|
| GPIOB, Pin 10        | I2S2_CK (Clock)   | CLK             |
| GPIOC, Pin 3         | I2S2_SD (Data)    | DOUT            |

Для забезпечення коректного збору даних з MEMS-мікрофона необхідно точно розрахувати частоту тактування інтерфейсу I2S. Робоча частота синхронізації ( $f_{CLK}$ ) визначається як добуток цільової частоти дискретизації вихідного PCM-сигналу ( $f_s$ ) та обраного коефіцієнта децимації ( $M$ ):

$$f_{CLK}=f_s \cdot M=16 \text{ кГц} \cdot 64=1024 \text{ кГц}$$

Таким чином, для отримання якісного аудіопотоку з частотою 16 кГц при заданому рівні проріджування, системна частота тактування мікрофона має становити 1,024 МГц. Для отримання цільової частоти тактування інтерфейсу I2S (1,024 МГц) було налаштовано внутрішню систему множників та дільників мікроконтролера STM32F407VGT6. Основні етапи формування сигналу синхронізації на основі зовнішнього резонатора (HSE) наведено в Табл. 4.4.

Таблиця 4.4. Алгоритм формування тактової частоти I2S

| Етап/Сигнал              | Формула розрахунку                    | Значення параметрів                 | Результат  |
|--------------------------|---------------------------------------|-------------------------------------|------------|
| HSE (Вхідний сигнал)     | -                                     | Вхідна частота кварцу               | 8 МГц      |
| VCO Clock (Вихід PLLI2S) | $f_{in} \cdot \frac{N}{M}$            | $8 \text{ МГц} \cdot \frac{258}{8}$ | 258 МГц    |
| PLL I2S Clock            | $\frac{VCO}{R}$                       | 258 МГц/3                           | 86 МГц     |
| I2S Divider              | $\frac{f_{PLL\_I2S}}{I2SDIV \cdot 2}$ | 86 МГц/(42·2)                       | 1,0238 МГц |

Після визначення апаратної конфігурації та параметрів синхронізації мікрофона MP45DT02 розглянемо програмні механізми захоплення та перетворення вхідного потоку. Процес захоплення даних реалізовано через

інтерфейс I2S2, який налаштований на безперервний збір відліків із мікрофона. Система генерує переривання з періодичністю в 1 мс, що свідчить про заповнення буфера даних. На рівні апаратного зчитування отримані значення зберігаються в масиві типу `uint16_t`. Оскільки кожен біт у форматі PDM є самостійним семплом, кожне 16-бітне слово в масиві фактично містить послідовність із 16 окремих звукових вибірок.

Вибір кількості відліків та частоти зрізу на даному етапі тестування мав ітераційний характер. Ключовим технічним обмеженням виступала теорема Котельникова (Найквіста), згідно з якою частота зрізу фільтра повинна бути нижчою за половину частоти дискретизації вихідного сигналу. Для цільового PCM-потoku із частотою 1 кГц межа Найквіста становить 8 кГц, що робить значення у 6 кГц коректним для запобігання ефекту накладання спектрів.

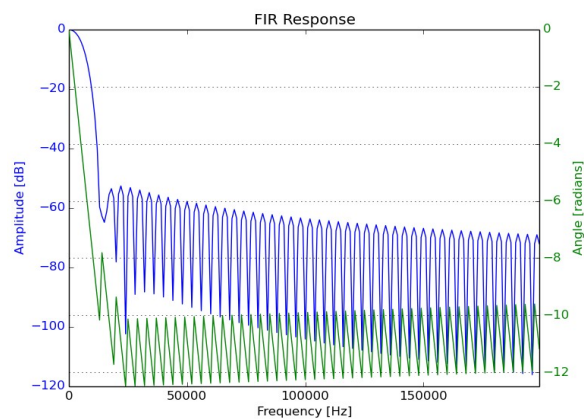


Рис. 4.5. Амплітудно-частотна характеристика FIR-фільтра децимації бібліотеки CMSIS-DSP

На графіку зображено АЧХ розробленого ФНЧ 255-го порядку. Розрахована крива демонструє високу рівномірність у смузі пропускання (до 6 кГц) та крутий спад у перехідній зоні. Така характеристика є критично важливою для ефективного придушення PDM-шумів та запобігання аліасингу при зниженні частоти дискретизації до 16 кГц.

Аналіз частотної характеристики підтверджує, що фільтр демонструє стабільний відгук у найбільш релевантному для людського мовлення діапазоні частот. Це забезпечує високу лінійність передачі аудіосигналу та ефективне придушення позасмугових завад.

Для верифікації роботи алгоритму було реалізовано механізм буферизації з часовим вікном у 1,6 с. Такий інтервал обрано як максимально допустимий, виходячи з ресурсів динамічної пам'яті МК. Алгоритм працює за принципом лінійного заповнення: досягнення кінцевої адреси буфера сигналізує про завершення тестового циклу. У цей момент система переходить у стан готовності до зчитування та верифікації отриманого аудіосигналу.

Процес тестування пройшов шлях від виявлення випадкових шумів до отримання стабільного аудіопотоку. З метою кількісної верифікації працездатності коду було застосовано метод порівняння сигналів. Для цього згенеровано тестовий звуковий файл, що складався з накладених чистих тонів (суперпозиція синусоїд). Запис проводився в безпосередній близькості від джерела звуку за допомогою налагоджувальної плати STM32F407G-DISC1. Кінцевий етап включав зіставлення часових діаграм і спектральних характеристик вихідного (еталонного) та результуючого (записаного) файлів для підтвердження лінійності тракту обробки.

На Рис.4.6 представлено фрагмент сигналу після проходження через FIR-фільтр та етап децимації. Часова діаграма демонструє відновлену форму хвилі (PCM), що відповідає тестовій суперпозиції тонів. Відсутність видимих високочастотних спотворень та артефактів квантування свідчить про ефективність застосованого алгоритму фільтрації низьких частот.

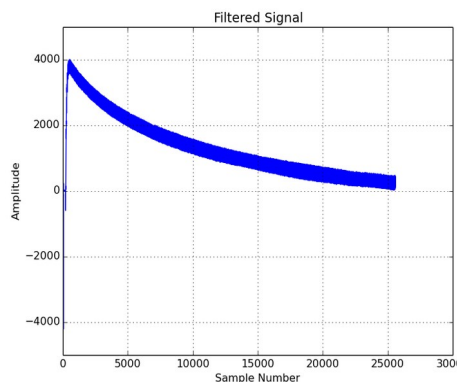


Рис. 4.6. Осцилограма відфільтрованого аудіосигналу в часовій області

Верифікація роботи цифрового тракту здійснювалася шляхом зіставлення спектрів сигналів. За допомогою спеціалізованого програмного забезпечення було виконано обчислення FFT для накопиченого масиву даних.

Спектральний аналіз (Рис. 4.7) дозволив ідентифікувати ключові гармоніки записаного сигналу та порівняти їх із частотним профілем джерела. Збіг пікових значень на графіках FFT свідчить про високу лінійність системи та відсутність значних амплітудних спотворень після етапів фільтрації та децимації.

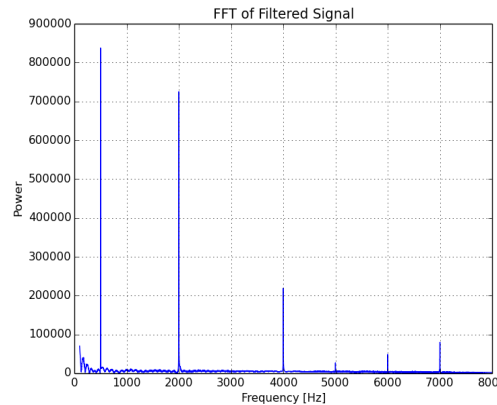


Рис. 4.7. Спектральний склад записаного сигналу після FFT-обробки

Рис. 4.7 ілюструє частотний спектр сигналу, отриманого з мікрофона MP45DT02. Положення основних спектральних максимумів повністю збігається з частотами еталонних тонів, що генерувалися під час тесту. Це є кількісним доказом того, що цифровий тракт МК STM32F407 коректно відновлює аудіоінформацію з PDM-поток, зберігаючи цілісність фазових та амплітудних характеристик вихідного сигналу. Для об'єктивного порівняння вхідних та вихідних даних використано багаточастотний сигнал (Табл. 4.5).

Таблиця 4.5. Параметри еталонних гармонік тестового сигналу

| Номер гармоніки | Частота, Гц | Цільове призначення в тесті              | Результат фіксації (ШПФ) |
|-----------------|-------------|------------------------------------------|--------------------------|
| 1               | 500         | Перевірка НЧ-діапазону                   | Присутня                 |
| 2               | 2000        | Перевірка СЧ-діапазону                   | Присутня                 |
| 3               | 4000        | Перевірка лінійності в смузі пропускання | Присутня                 |

Для верифікації селективності FIR-фільтра було проведено стрес-тест із використанням частот, що виходять за межі смуги пропускання. Аналіз результуючого спектра (Рис. 4.7) показав: 1. Компонента 7 кГц: Залишається в спектрі через те, що перехідна зона фільтра (від 6 до 8 кГц) має скінченну крутизну. 2. Дзеркальні складові (Aliasing): Тони 10 кГц та 11 кГц через недостатнє пригнічення до моменту децимації “віддзеркалилися” всередину

робочого діапазону. Зокрема, сигнал 10 кГц з'явився на позначці 6 кГц, а 11 кГц – на рівні 5 кГц. Це спостереження дозволяє кількісно визначити реальну здатність фільтра до придушення шумів у смузі загородження.

На Рис.4.8 представлено тестовий макет пристрою для аудіофіксації правопорушень, що реалізує алгоритми цифрової обробки сигналів. Макет побудовано на налагоджувальній платі STM32F407G-DISC1, яка об'єднує ядро ARM Cortex-M4, модуль FPU та необхідні периферійні інтерфейси (I2S, DMA). Таке компонування дозволяє проводити реальну перевірку працездатності розробленого програмного забезпечення та оцінювати якість фіксації аудіоінформації в умовах, наближених до експлуатаційних.

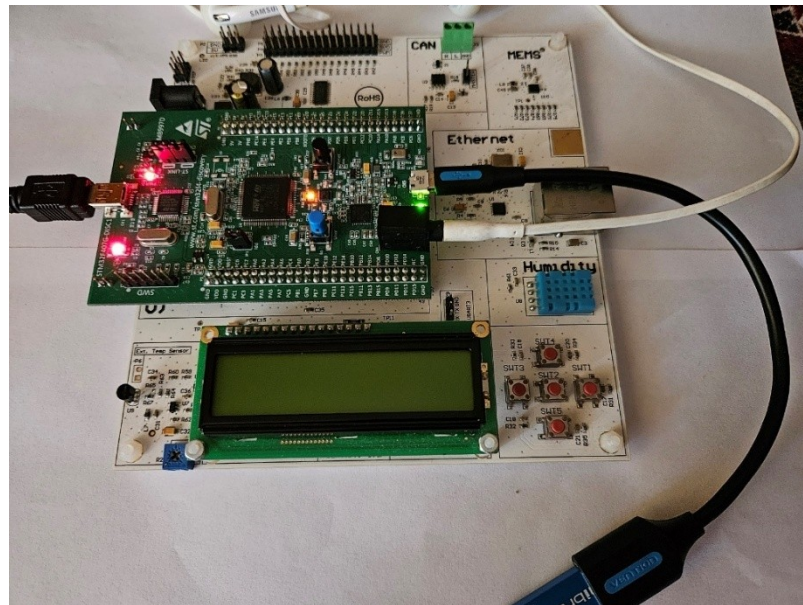


Рис. 4.8. Тестовий макет цифрового пристрою аудіофіксації на базі МК STM32F407VGT6

## ВИСНОВКИ

У кваліфікаційній роботі виконано комплексне дослідження сучасних технологій оцифрування звукових сигналів, а також алгоритмів їхньої обробки в реальному часі на базі МК родини STM32. Розглянуто конструктивні особливості та принципи функціонування цифрових MEMS-мікрофонів, наведено практичні схеми їхньої інтеграції для запису моно- та стереозвуку. Окрему увагу приділено методам спряження сенсорів із модулями SPI/I2S, SAI та DFSDM. Результатом роботи стала розробка спеціалізованого диктофона на МК STM32F407VGT6, призначеного для використання співробітниками поліції під час фіксації правопорушень. Технічна реалізація пристрою для аудіофіксації правопорушень представлена макетом на базі відлагоджувальної системи STM32F407G-DISC1. Основним обчислювальним вузлом є МК STM32F407VGT6, який координує роботу цифрового MEMS-сенсора MP45DT02 для захоплення звуку та високоточного стерео-ЦАП CS43L22 для виведення аудіосигналу. Взаємодія мікрофона MP45DT02 з обчислювальним ядром здійснюється через інтерфейс I2S2. Для оптимізації роботи процесора під час отримання PDM-потoku задіяно контролер DMA, який забезпечує пряме перенесення даних у буфер RAM для їх подальшої програмної фільтрації. Збереження результатів аудіофіксації реалізовано на USB-носії за допомогою модуля USB Host та протоколу MSC. Відтворення звуку та керування аудіокодеком CS43L22 виконується через модулі I2S3 (трансляція PCM-даних) та I2C1 (передача команд керування). Користувачський інтерфейс для перемикання режимів роботи пристрою побудований на базі штатних кнопок налагоджувальної плати. Програмний комплекс системи аудіофіксації реалізовано мовами C/C++. В основі алгоритму збору звукових даних лежить використання драйвера BSP\_AUDIO, адаптованого для МК STM32F407VGT6, та спеціалізованої бібліотеки PDM2PCM. Процес обробки сигналів включає етапи цифрової фільтрації, децимації та конвертації вхідного PDM-потoku в формат PCM. Кінцеві результати записуються на зовнішній USB-накопичувач у контейнері формату WAV.

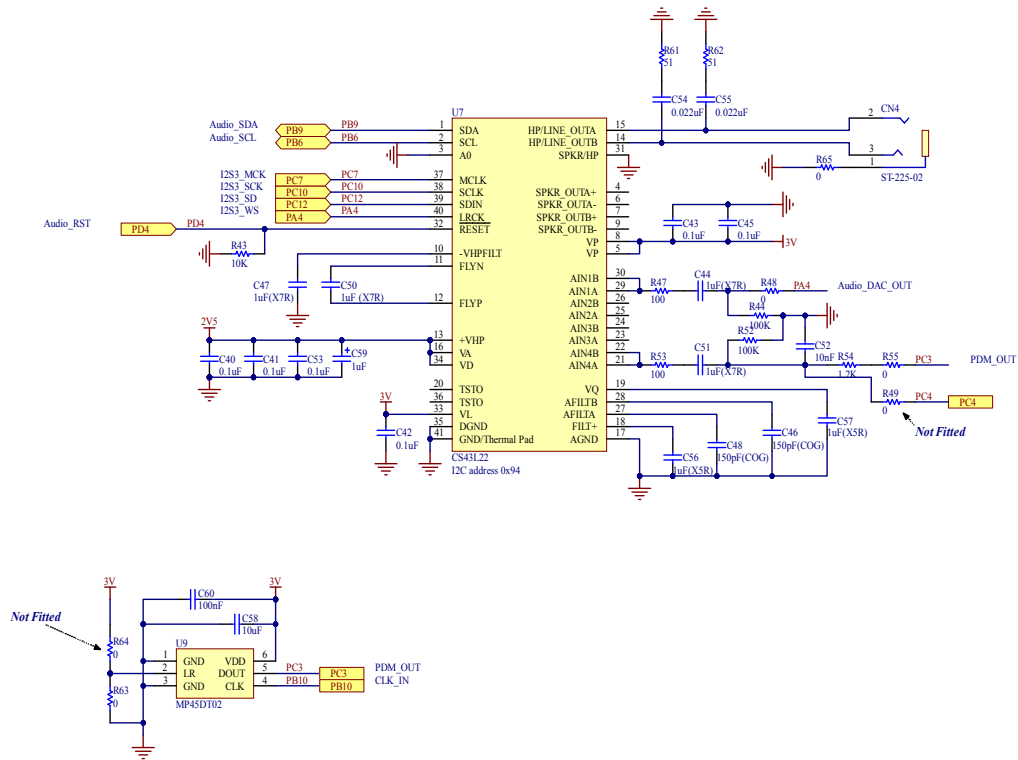
## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Norris D. Programming with STM32: Getting Started with the Nucleo Board and C/C++. 1st ed. McGraw Hill TAB, 2018. 304 p.
2. Noviello C. Mastering STM32. A step-by-step guide to the most complete ARM Cortex-M platform. 2nd ed. Lean Publishing, 2018. 852 p.
3. Pakdel M. Advanced Programming with STM32 Microcontrollers: Master the software tools behind the STM32 microcontroller. Elektor Verlag, 2020. 215p.
4. Geoffrey B. Discovering the STM32 Microcontroller. 2016. 244 p.
5. STM32 Programming Tutorials & Projects. *STMicroelectronics Education*. URL: [https://www.st.com/content/st\\_com/en/support/learning/stm32-education.html](https://www.st.com/content/st_com/en/support/learning/stm32-education.html) (дата звернення: 08.04.2026).
6. DeepBlue Embedded. STM32 ARM Programming Tutorials. URL: <https://deepbluembedded.com/stm32-arm-programming-tutorials/> (дата звернення: 08.04.2026).
7. Mazidi M. A., Chen S., Ghaemi E. STM32 Arm Programming for Embedded Systems. MicroDigitalEd, 2018. 378 p.
8. Amos B. Hands-On RTOS with Microcontrollers: Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools. Packt Publishing, 2020. 496 p.
9. Ibrahim D. Nucleo Boards Programming with the STM32CubeIDE: Hands-on in more than 50 projects. Elektor Verlag, 2021. 498 p.
10. Discovery kit with STM32F407VG MCU. Data brief. *STMicroelectronics*. URL: [https://www.st.com/resource/en/data\\_brief/stm32f4discovery.pdf](https://www.st.com/resource/en/data_brief/stm32f4discovery.pdf).
11. STM32F407VG Microcontrollers. *STMicroelectronics Official Website*. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html>.
12. RM0090 Reference Manual. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm-based 32-bit MCUs. Rev. 19. STMicroelectronics, 2021. 1751 p.
13. PM0214 Programming Manual. STM32 Cortex-M4 MCUs and MPUs programming manual. STMicroelectronics, 2020. 262 p.

14. UM1725 User manual. Description of STM32F4 HAL and low-layer drivers. STMicroelectronics, 2020. 2227 p.
15. AN10658 Application note. Sending I2C-bus signals via long communications cables. NXP Semiconductors, 2008. 18 p.
16. What is the I2S Communication Protocol? *DigiKey Maker Tutorials*. URL: <https://www.digikey.com/en/maker/tutorials/2023/what-is-the-i2s-communication-protocol>.
17. UM1472 User manual. Discovery kit with STM32F407VG MCU. STMicroelectronics, 2020. 32 p.
18. AN3997 Application note. Audio playback and recording using the STM32F4DISCOVERY. STMicroelectronics, 2011. 15 p.
19. AN3998 Application note. PDM audio software decoding on STM32 microcontrollers. STMicroelectronics, 2011. 10 p.
20. AN5027 Application note. Interfacing PDM digital microphones using STM32 MCUs and MPUs. STMicroelectronics, 2019. 66 p.
21. STM32CubeMX code configurator. *STMicroelectronics*. URL: <https://www.st.com/en/development-tools/stm32cubemx.html>.
22. STM32CubeIDE integrated development environment. *STMicroelectronics*. URL: <https://www.st.com/en/development-tools/stm32cubeide.html>.

## **ДОДАТКИ**

## Додаток 1



Принципова схема тракту обробки звуку (МЕМС-мікрофон MP45DT02 та аудіокодек CS43L22)

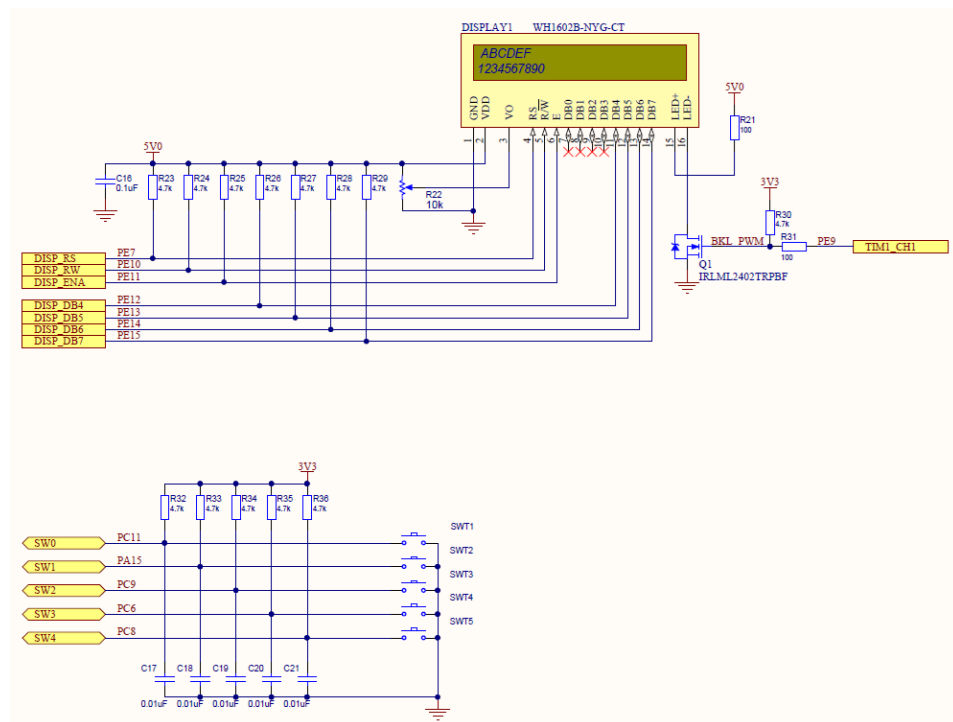


Схема електрична принципова модуля інтерфейсу користувача (LCD WH1602B та клавіатурна матриця)

## Додаток 2

### Фрагмент лістинга програмного забезпечення цифрового пристрою на МК STM32F407VGT6 для аудіофіксації правопорушень працівниками поліції

```
#include "main.h"
/* Структури керування периферією */
TIM_HandleTypeDef htim_led_indicator;
TIM_OC_InitTypeDef soc_led_config;
USBH_HandleTypeDef hUsbHost;
FATFS usb_fatfs_handle;
/* Зовнішні прапори та змінні стану */
extern uint32_t AudioPlaybackActive;
extern __IO uint32_t GlobalLedsStatus;
__IO uint16_t timer_compare_val = 16826;
/* Системні змінні пристрою */
uint8_t sensor_chip_id = 0;
__IO uint32_t operation_mode = CMD_PLAY;
__IO uint32_t is_button_pressed = 0;
char usb_logic_path[4];
/* Перерахування станів додатку для зручності */
typedef enum {
    STATE_FS_INIT = 0,
    STATE_AUDIO_PROCESS,
    STATE_IDLE
} AppSubState_t;
MSC_ApplicationTypeDef current_app_state = APPLICATION_IDLE;
static AppSubState_t internal_logic_state = STATE_FS_INIT;
/* Прототипи локальних функцій */
static void LedTimer_Peripheral_Init(void);
static void SystemClock_Setup(void);
static void USB_Host_EventCallback(USBH_HandleTypeDef *pHost, uint8_t vId);
static void Process_Storage_Application(void);
static void Execute_Audio_Task(void);

int main(void)
{
    /* Базова ініціалізація HAL та системної частоти */
    HAL_Init();
    SystemClock_Setup();
    /* Конфігурація індикаторів плати Discovery */
    BSP_LED_Init(LED3); // Error
    BSP_LED_Init(LED4); // Status
    BSP_LED_Init(LED5); // Activity
    BSP_LED_Init(LED6); // Ready
    /* Ініціалізація акселерометра та зчитування ID */
    if (BSP_ACCELERO_Init() != ACCELERO_OK) {
        Error_Handler();
    }
    sensor_chip_id = BSP_ACCELERO_ReadID();
    /* Налаштування інтерфейсу користувача */
```

```

BSP_LED_On(LED4);
LedTimer_Peripheral_Init();
RepeatState = REPEAT_ON;
GlobalLedsStatus = LEDS_OFF;
/* Ініціалізація вбудованої та зовнішніх кнопок */
BSP_PB_Init(BUTTON_KEY, BUTTON_MODE_EXTI);
BSP_PB_Init(SWT1_KEY, BUTTON_MODE_GPIO);
/* Запуск стека USB Host */
if (FATFS_LinkDriver(&USBH_Driver, usb_logic_path) == 0) {
    USBH_Init(&hUsbHost, USB_Host_EventCallback, 0);
    USBH_RegisterClass(&hUsbHost, USBH_MSC_CLASS);
    USBH_Start(&hUsbHost);
    while (1) {
        switch (current_app_state) {
            case APPLICATION_START:
                Process_Storage_Application();
                break;
            case APPLICATION_IDLE:
            default:
                break;
        }
        USBH_Process(&hUsbHost);
    }
}
while (1);
}
/* @brief Обробник подій USB Host */
static void USB_Host_EventCallback(USBH_HandleTypeDef *pHost, uint8_t vId){
    switch (vId) {
        case HOST_USER_CLASS_ACTIVE:
            current_app_state = APPLICATION_START;
            break;
        case HOST_USER_DISCONNECTION:
            WavePlayer_CallBack();
            current_app_state = APPLICATION_IDLE;
            f_mount(NULL, (TCHAR const*)"", 0);
            break;
        default:
            break;
    }
}

/* @brief Керування файловою системою на USB носії */
static void Process_Storage_Application(void){
    switch (internal_logic_state) {
        case STATE_FS_INIT:
            if (f_mount(&usb_fatfs_handle, (TCHAR const*)usb_logic_path, 0) != FR_OK){
                Error_Handler();
            }
            internal_logic_state = STATE_AUDIO_PROCESS;
            break;
        case STATE_AUDIO_PROCESS:
            Execute_Audio_Task();
            internal_logic_state = STATE_FS_INIT;
            break;
    }
}

```

```

        default:
            break;
    } }
/* @brief Виконання вибраної аудіо-команди */
static void Execute_Audio_Task(void){
    if (operation_mode == CMD_PLAY) {
        if (RepeatState == REPEAT_ON) {
            WavePlayerStart();
        }
    } else if (operation_mode == CMD_RECORD) {
        RepeatState = REPEAT_ON;
        WaveRecorderProcess();
    } }
/* @brief Обробка зовнішніх переривань від кнопок */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
    if (GPIO_Pin == GPIO_PIN_0) { // Перемикання режимів Запис/Відтворення
        if (!is_button_pressed) {
            HAL_Delay(10); // Антибрязкіт
            operation_mode = (operation_mode == CMD_RECORD) ? CMD_PLAY : CMD_RECORD;
            RepeatState = REPEAT_ON;
            is_button_pressed = 1;
        } else { is_button_pressed = 0; }
    }
    if (GPIO_Pin == GPIO_PIN_1) { // Пауза / Резюме
        static uint8_t pause_toggle = 0;
        if (pause_toggle) {
            PauseResumeStatus = RESUME_STATUS;
            pause_toggle = 0;
        } else {
            PauseResumeStatus = PAUSE_STATUS; pause_toggle = 1;
        } }
}

/* @brief Конфігурація таймера для LED індикації */
static void LedTimer_Peripheral_Init(void)
{
    uint32_t pclk1_freq;
    __HAL_RCC_TIM4_CLK_ENABLE();
    HAL_NVIC_SetPriority(TIM4_IRQn, 6, 0);
    HAL_NVIC_EnableIRQ(TIM4_IRQn);
    pclk1_freq = HAL_RCC_GetPCLK1Freq();
    uint16_t prescaler = (uint16_t)((pclk1_freq * 2) / 550000) - 1;
    htim_led_indicator.Instance = TIM4;
    htim_led_indicator.Init.Period = 65535;
    htim_led_indicator.Init.Prescaler = prescaler;
    htim_led_indicator.Init.ClockDivision = 0;
    htim_led_indicator.Init.CounterMode = TIM_COUNTERMODE_UP;
    if (HAL_TIM_OC_Init(&htim_led_indicator) == HAL_OK) {
        soc_led_config.OCMode = TIM_OCMODE_TIMING;
        soc_led_config.Pulse = timer_compare_val;
        soc_led_config.OCpolarity = TIM_OCPOLARITY_HIGH;
        HAL_TIM_OC_ConfigChannel(&htim_led_indicator,&soc_led_config, TIM_CHANNEL_1);
        HAL_TIM_OC_Start_IT(&htim_led_indicator, TIM_CHANNEL_1);
    }
}

```

```
void Error_Handler(void){  
    BSP_LED_On(LED3);  
    while (1);  
}
```