

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____ Ірина ЖУРАВСЬКА

«__» _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

Система визначення нот
на основі частоти звукових коливань

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач

_____ Андрій ПЕТРОВ
підпис

«__» _____ 202__ р.

Керівник ст. викладач кафедри КІ

_____ Катерина ОБУХОВА
підпис

«__» _____ 202__ р.

Миколаїв – 2025

Факультет	Комп'ютерних наук
Кафедра	Комп'ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерної інженерії
Ірина ЖУРАВСЬКА
«_____» _____ 2024 р.

ЗАВДАННЯ на кваліфікаційну роботу здобувача

Петров Андрій Романович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Система визначення нот на основі частоти звукових коливань

Затверджена наказом по ЧНУ ім. Петра Могили від 30.10.2024 № 297.

2. Строк представлення кваліфікаційної роботи «_____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Створення системи для визначення нот на основі частоти звукових коливань, яка забезпечуватиме швидко та точно розпізнавання нот у реальному часі. Система повинна бути простою в експлуатації та зрозумілою у використанні в порівнянні з іншими існуючими аналогами для визначення звуку.

4. Перелік питань, що підлягають розробці:

– проведення аналізу існуючих систем розпізнавання звуку, їхніх переваг та недоліків;

– ознайомлення з принципами дії швидкого перетворення Фур'є та його використанням у цифровій обробці сигналів, дослідження властивостей звукових коливань;

– розробка необхідного обладнання на основі Arduino, з використанням модуля електронного мікрофону та дисплея;

– реалізація програмного забезпечення, яке буде оброблювати цифровий сигнал з мікрофону, мікроконтролер буде обробляти інформацію і виводити на дисплей готовий результат;

– тестування створеної системи визначення нот, її точності, швидкості реагування та стабільності у різних умовах.

5. Перелік графічних матеріалів

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи

Особистий підпис

_____ Катерина ОБУХОВА

Власне ім'я ПРІЗВИЩЕ

Здобувач

Особистий підпис

_____ Андрій ПЕТРОВ

Власне ім'я ПРІЗВИЩЕ

Дата видачі завдання « _____ » _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема: Система визначення нот на основі частоти звукових коливань

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	30.10.2024	30.10.2024	Виконано
2	Огляд літератури за темою роботи	26.03.2025	08.04.2025	Виконано
3	Складання календарного плану КБР	12.04.2025	16.04.2025	Виконано
4	Аналіз предметної області	18.04.2025	05.05.2025	Виконано
5	Розробка проектних рішень	05.05.2025	10.05.2025	Виконано
6	Моделювання та конструювання АПЗ	11.05.2025	22.05.2025	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	24.05.2025	02.06.2025	Виконано
8	Відгук керівника КБР	02.06.2025	04.06.2025	Виконано
9	Оформлення КБР та презентації	04.06.2025	07.06.2025	Виконано
10	Попередній захист	21.05.2025	02.06.2025	Виконано
11	Рецензування	08.06.2025	09.06.2025	Виконано
12	Завершення оформлення КБР та презентації	09.06.2025	11.06.2025	Виконано
13	Захист кваліфікаційної бакалаврської роботи	24.06.2025	24.06.2025	Виконано

Керівник роботи

Особистий підпис

_____ Катерина ОБУХОВА

Власне ім'я ПРИЗВИЩЕ

Здобувач

Особистий підпис

_____ Андрій ПЕТРОВ

Власне ім'я ПРИЗВИЩЕ

« ____ » _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Система визначення нот на основі частоти звукових коливань»

Студент 405 гр.: Петров Андрій Романович

Керівник: ст. викладач Обухова Катерина Олександрівна

Кваліфікаційна бакалаврська робота (КБР) присвячена створенню системи для визначення нот на основі частоти звукових коливань із використанням швидкого перетворення Фур'є. Цей алгоритм дозволяє ефективно аналізувати звуковий сигнал для виявлення домінуючих частот, що дає змогу визначити відповідні музичні ноти. **Актуальність** дослідження полягає у розробці зручної, ефективної та доступної системи, що забезпечує оперативне отримання результатів в реальному часі та сприяє полегшеному засвоєнню музичної грамоти через аналіз нот.

Об'єкт дослідження – процес аналізу звукових коливань з метою визначення музичних нот на основі їх частотних характеристик.

Предмет дослідження – розробка та програмна реалізація системи розпізнавання нот шляхом аналізу частоти звукових сигналів із застосуванням алгоритму швидкого перетворення Фур'є.

Мета роботи – створення функціональної системи, здатної визначати музичні ноти на основі частотного аналізу звукових коливань із використанням методу швидкого перетворення Фур'є.

Для реалізації поставленої мети були досліджені й застосовані сучасні методи аналізу частотних характеристик звуку, розглянуто основні технології обробки аудіо сигналів, а також підібрано відповідні апаратні компоненти, які були використані при створенні системи.

Розроблена система включає мікрофонний модуль для зчитування аналогового сигналу, мікроконтролер Arduino, який здійснює цифрову обробку сигналів і керує роботою пристрою, а також дисплей для відображення відповідних нот у зручному форматі.

Пояснювальна записка КБР складається зі вступу, трьох розділів, висновків, переліку джерел посилання та 3 додатків.

У вступі розкрито актуальність теми, визначено об'єкт і предмет дослідження, сформульовано мету та завдання роботи, окреслено сферу застосування системи, а також наведено дані щодо апробації результатів.

У першому розділі здійснено аналітичний огляд проблеми та сучасних підходів до її вирішення, проаналізовано переваги й недоліки існуючих аналогічних систем, обґрунтовано вимоги до проєктованої системи та визначено її технічні специфікації. Окрему увагу приділено розгляду моделей і методів розпізнавання звукових сигналів.

У другому розділі було проаналізовано кожен компонент системи, здійснено обґрунтований вибір елементів шляхом порівняння з доступними аналогами, а також подано їхні ключові характеристики та принципи функціонування.

У третьому розділі детально описано логіку роботи системи, представлено схему взаємозв'язку її складових частин та проведено тестування для перевірки працездатності прототипу.

У висновках підсумовано результати виконаної роботи, проаналізовано ефективність розробленої системи та сформульовано основні висновки на основі проведених досліджень.

В цілому КБР містить 72 сторінки (без додатків), 64 рисунок, 8 таблиць, 32 джерел посилання.

Ключові слова: звукові коливання, музичні ноти, аналіз частотних характеристик звуку, швидке перетворення Фур'є, цифрова обробка сигналу, Arduino.

ABSTRACT

of the Bachelor's Thesis

"Note detection system based on the frequency of sound vibrations"

Student: Petrov Andrii Romanovich

Supervisor: senior teacher Obukhova Kateryna

The bachelor's thesis is dedicated to the development of a system for detecting musical notes based on the frequency of sound vibrations using the Fast Fourier Transform (FFT). This algorithm enables efficient analysis of audio signals to identify dominant frequencies, allowing the corresponding musical notes to be determined. The relevance of the study lies in the creation of a convenient, efficient, and accessible system that provides real-time results and facilitates the learning of music theory through note analysis.

The object of the study – the process of analyzing sound vibrations in order to determine musical notes based on their frequency characteristics.

The subject of the study – the development and software implementation of a system for recognizing musical notes by analyzing the frequency of sound signals using the Fast Fourier Transform algorithm.

The purpose of the bachelor's thesis – to create a functional system capable of determining musical notes based on the frequency analysis of sound vibrations using the FFT method.

To achieve the stated goal, modern methods for analyzing frequency characteristics of sound were studied and applied, the main technologies of audio signal processing were reviewed, and appropriate hardware components were selected and used in the system development.

The developed system includes a microphone module for reading analog signals, an Arduino microcontroller that performs digital signal processing and manages device operations, as well as a display for conveniently showing the corresponding notes.

The explanatory note of the thesis consists of an introduction, three chapters, conclusions, a list of references, and three appendices.

The introduction outlines the relevance of the topic, defines the object and subject of the study, formulates the purpose and objectives of the work, identifies the application areas of the system, and presents information about the testing and validation of the results.

The first chapter provides an analytical review of the problem and modern approaches to its solution, analyzes the advantages and disadvantages of existing similar systems, substantiates the requirements for the designed system, and defines its technical specifications. Particular attention is paid to the study of models and methods for recognizing audio signals.

The second chapter presents a comprehensive analysis of each system component, a reasoned selection of elements through comparison with available alternatives, as well as a description of their key characteristics and operating principles.

The third chapter describes the system's logic in detail, presents a diagram of the interconnection between its components, and outlines the stages of testing used to verify the functionality of the prototype.

The conclusions summarize the results of the completed work, analyze the effectiveness of the developed system, and present the main findings based on the research.

The paper consists of 72 pages (excluding appendices), 8 tables, 64 figures, 32 references, and 3 appendices.

Keywords: sound vibrations, musical notes, frequency characteristics analysis of sound, Fast Fourier Transform, digital signal processing, Arduino.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 ЗАСОБИ ОТРИМАННЯ, ОБРОБКИ ТА АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ З ВИКОРИСТАННЯМ ЗВУКОВИХ КОЛИВАНЬ	6
1.1 Природа звукових хвиль їх основні характеристики.....	6
1.2 Основи музичних нот	7
1.3 Огляд наявних альтернативних рішень, їх недоліки та переваги.....	9
1.4 Алгоритми розпізнавання звуку.....	14
1.5 Аналіз вимог до системи.....	19
Висновки до розділу 1	21
2 ПРОЄКТУВАННЯ СИСТЕМИ ТА ВИБІР СКЛАДОВИХ СИСТЕМИ ВИЗНАЧЕННЯ НОТ	22
2.1 Концепція системи визначення нот	22
2.2 Мікроконтролер	23
2.3 Мікрофон	28
2.4 Дисплей.....	33
Висновки до розділу 2	41
3 РОЗРОБКА АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВИЗНАЧЕННЯ НОТ	42
3.1 З'єднання компонентів пристрою	42
3.2 Налаштування Arduino IDE	49
3.3 Тестування системи.....	56
3.4 Методи покращення системи	65
Висновок до розділу 3.....	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	69
ДОДАТОК А Блок-схема алгоритму роботи системи	73
ДОДАТОК Б Код системи.....	74
ДОДАТОК В Матеріали апробації роботи.....	77

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП – аналогово-цифровий перетворювач

КБР – кваліфікаційна бакалаврська робота

ADC – Analog-to-digital converter

AGC – Automatic Gain Control,

DFT – Discrete Fourier Transform

FFT – Fast Fourier Transform

GUI – Graphical user interface

I2C – Inter-Integrated Circuit

MEMS – Micro-Electro-Mechanical Systems

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver-Transmitter

ВСТУП

У світі музики та акустики визначення нот на основі частоти звукових коливань стало невід'ємною частиною сучасних технологій. Ця технологія широко використовується для автоматичного налаштування музичних інструментів, аналізу звукових записів, а також у навчальних програмах для музикантів. Вона дозволяє забезпечити високу точність та швидкість розпізнавання нот, що є надзвичайно важливим в умовах реального часу, наприклад, під час живих виступів або при використанні музичних інструментів.

З розвитком цифрових технологій та аудіо обробки застосування систем визначення нот стало доступним не лише професіоналам, але й аматорам і студентам музичних навчальних закладів. Вони можуть використовувати подібні системи як інструмент для самостійного навчання та тренування слуху.

Визначення нот за допомогою аналізу частоти звукових коливань ґрунтується на розумінні фізичної природи звуку. Кожна музична нота має певну частоту, яка відповідає конкретному тону. Ці частоти можна виявити шляхом математичної обробки сигналу, що дозволяє розкласти складний звуковий сигнал на окремі гармоніки. Це дає можливість для точного визначення нот у складних звукових середовищах, навіть за наявності фонових шумів або одночасному звучанні кількох нот.

Актуальність дослідження полягає у створенні зручної, ефективної та недорогої системи, яка дозволить швидко отримувати результати в реальному часі та допоможе з вивченням музичних нот. Це сприятиме покращенню навичок сприйняття нот на слух, що є важливим у світі музики. Більше того, такий пристрій можна інтегрувати у навчальні курси для цифрової обробки сигналів.

Об'єктом дослідження є процес аналізу звукових коливань для визначення музичних нот на основі їх частотних характеристик.

Предметом дослідження є реалізація системи визначення нот на основі звукових коливань із використанням швидкого перетворення Фур'є.

Мета дослідження полягає у створенні системи для визначення музичних нот на основі аналізу частоти звукових коливань із використанням швидкого перетворення Фур'є (англ. Fast Fourier Transform, FFT). Така система буде компактною, недорогою, зручною та простою у використанні, здатною працювати в режимі реального часу, забезпечуючи точне розпізнавання частотного спектру звукового сигналу та його подальше перетворення у вигляді музичних нот.

Система буде працювати на базі мікроконтролера Arduino, що є популярною платформою завдяки простоті використання, доступності та великій кількості різноманітних модулів. Arduino широко використовується як в навчальних так і в професійних проєктах, що робить її гарним вибором для реалізації таких систем.

Відповідно до мети дослідження визначено такі завдання:

- провести аналіз існуючих систем розпізнавання звуку, їхніх переваг та недоліків;
- ознайомитися з принципами дії швидкого перетворення Фур'є та його використанням у цифровій обробці сигналів;
- дослідити та проаналізувати властивості звукових коливань;
- зібрати необхідне обладнання на основі Arduino, з використанням модуля електронного мікрофону та дисплея;
- реалізувати програмне забезпечення, яке буде оброблювати цифровий сигнал з мікрофону, мікроконтролер буде обробляти інформацію і виводити на дисплей готовий результат;
- провести тестування створеної системи визначення нот, її точності, швидкості реагування та стабільності у різних умовах.

Сфера застосування: систему для визначення музичних нот можна використовувати для навчання музикантів, тренування слуху, цифрової обробки сигналів, візуалізації нот у реальному часі, також як основу для розробки подібних освітніх систем.

Апробація результатів кваліфікаційної бакалаврської роботи (КБР) відбулася під час XVI Міжнародної науково-практичної конференції «Free and Open Source Software» (Харків, 2025 р.) [3].

1 ЗАСОБИ ОТРИМАННЯ, ОБРОБКИ ТА АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ З ВИКОРИСТАННЯМ ЗВУКОВИХ КОЛИВАНЬ

У наш час, коли цифрові технології дедалі більше інтегруються у всі сфери життя, розробка простої, компактної та недорого вартісної системи для визначення музичних нот на основі частоти звукових коливань є актуальною не лише для музикантів, а й для освітнього процесу, оскільки вона сприяє підвищенню рівня слухового сприйняття та розвитку навичок налаштування інструментів.

1.1 Природа звукових хвиль їх основні характеристики

Звукова хвиля, також відома як акустична хвиля – це тип механічної хвилі, що поширюється через середовище – таке як повітря, вода або тверді тіла – внаслідок вібрації об'єкта. Вона характеризується здатністю переносити енергію та інформацію через середовище, проявляючись у таких фізичних властивостях, як частота, амплітуда, довжина хвилі та швидкість. Звукові хвилі є в основі поздовжніми хвилями, де звіщення частинок середовища відбувається паралельно до напрямку поширення хвилі, що призводить до утворення зон стиснення та розрідження [24].

Термін «хвиля звукового тиску» часто вживається в широкому сенсі для позначення хвиль, які знаходяться в межах діапазону чутності людини – приблизно від 20 Гц до 20 кГц. Натомість термін «акустична хвиля» частіше використовується в науковому, інженерному та технічному контекстах і охоплює як чутний звук, так і хвилі тиску повітря за межами діапазону людського слуху, включаючи інфразвук (нижче 20 Гц) та ультразвук (понад 20 кГц). Це розмежування підкреслює багатофункціональну природу звукових хвиль, які мають вирішальне для музики та взаємодії з навколишнім середовищем [27].

Властивості акустичних хвиль

Частота – кількість коливань або циклів, які відбуваються в звуковій хвилі за секунду, вимірюється в герцах. Частота визначає висоту звуку: чим вища частота – тим вищий звук, чим нижче – тим нижчий.

Довжина хвилі – фізична відстань між двома послідовними точками хвилі, які знаходяться в однаковій фазі.

Амплітуда – максимальне відхилення частинок від положення рівноваги внаслідок проходження хвилі. В акустичних хвилях амплітуда пов'язана з гучністю: чим більша амплітуда – тим голосніший звук.

Швидкість – швидкість, з якою звукова хвиля поширюється в середовищі. Вона залежить від властивостей середовища і є найвищою в твердих тілах, нижчою – у рідинах і найнижчою – у газах.

Інтенсивність – потужність, яку переносить звукова хвиля на одиницю площі, вимірюється у ватах на квадратний метр. Інтенсивність відображає енергію хвилі та залежить як від амплітуди, так і від відстані до джерела, визнаючи, наскільки гучним сприймається звук.

Фаза – описує положення точно в часі на циклі хвилі. У випадку взаємодії кількох хвиль, різниця фаз може призводити до конструктивної або деструктивної інтерференції, що суттєво впливає на амплітуду та інтенсивність результуючого звуку.

Оскільки розроблювальна система буде взаємодіяти зі звуковими хвилями, важливо розуміти їх природу та основні характеристики, зокрема частоту, амплітуду та інтенсивність.

1.2 Основи музичних нот

Музичні ноти є основою музичної грамоти, що дозволяє музикантам у всьому світі спілкуватися універсальною мовою. Незалежно від інструменту чи жанру, розуміння нотної системи є необхідним для тих, хто прагне глибше зануритися у світ музики [4].

Властивості музичних нот

Музичні ноти – це умовні символи, що позначають звуки певної висоти та тривалості. Основні характеристики нот:

- висота звуку визначається положенням ноти на нотному стані та відповідає частоті звукових коливань;

- тривалість звуку визначається формою ноти та позначає, як довго має звучати звук;
- динаміка: додаткові позначки, що вказують на гучність виконання.

Тривалість нот

Ноти мають різну тривалість звучання, яка вказується за допомогою їх графічного вигляду [28]:

- ціла нота: має найдовшу тривалість;
- половинна нота звучить у два рази коротше за цілу;
- чверть нота триває чверть від цілої;
- восьма, шістнадцята, тридцять друга ноти мають відповідно ще коротшу тривалість.

Хроматична шкала та рівномірна температура

В основі західної музичної системи лежить рівномірно темперований стрій, що поділяє кожную октаву на 12 однакових частин півтонів. Цей коефіцієнт, відомий як дванадцятий корінь з двох, дорівнює приблизно 1,0595. Саме він використовується для розрахунку частот нот у межах однієї октави [19]:

$$f_n = f_0 * 2^{n/12}, \quad (1.1)$$

де f_n – частота шуканої ноти;

f_0 – базова частота;

n – кількість півтонів вгору або вниз від базової частоти.

Після 12 послідовних кроків ми отримуємо ноту, що має вдвічі більшу частоту – тобто наступну однойменну ноту вищої октави. Такий принцип дає змогу обчислювати частоти всіх нот у будь-якому напрямку – як вгору, так і вниз по шкалі [26].

Хроматична шкала

Комплект із 12 півтонів в межах однієї октави формує хроматичну шкалу, що охоплює всі доступні ноти в сучасному музикуванні [21].

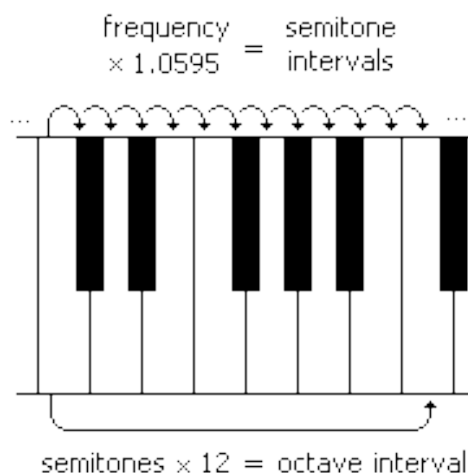


Рисунок 1.1 – Хроматична шкала [23]

Використання рівномірної темперації, стало важливим кроком у розвитку музики. Воно замінило попередні методи налаштування, які базувалися на простих дробових відношеннях частот, але були обмеженими для транспонування мелодій між тональностями [20].

Завдяки рівномірному поділу октави стало можливим використовувати будь-яку тональність без спотворення інтервалів, що суттєво розширило музичні можливості для композиторів і виконавців.

1.3 Огляд наявних альтернативних рішень, їх недоліки та переваги

Наразі у цифровому середовищі існує велика кількість альтернативних рішень для визначення музичних нот та налаштування інструментів. Прикладами таких систем є мобільні додатки, що працюють на базі Android, а також комп'ютерне програмне забезпечення для обробки звуку.

Tuner – gStrings

gStrings – це хроматичний тюнер, який призначений для вимірювання висоти та інтенсивності звуку. Додаток дозволить налаштувати будь-який музичний інструмент (скрипку, альт, віолончель, бас, гітару, фортепіано, духові інструменти, а також власний голос/спів).

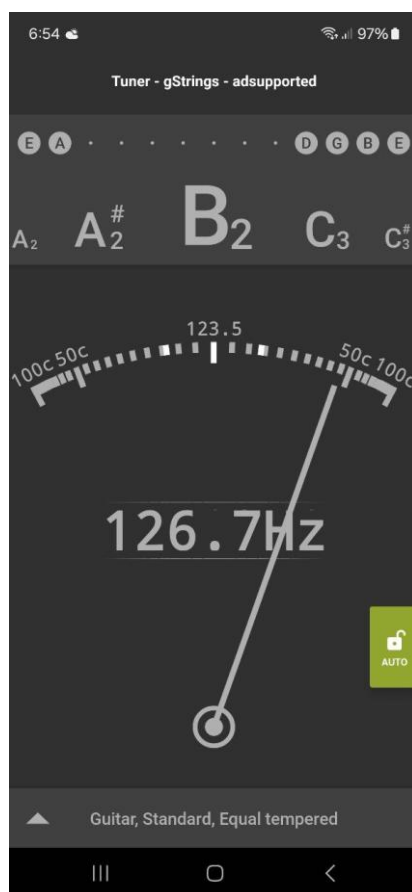


Рисунок 1.2 – Інтерфейс Tuner – gStrings

До ключових функціональних переваг застосунку можна віднести:

- підтримка кількох вбудованих налаштувань для різних інструментів;
- можливість користувацького налаштування параметрів;
- довгий список вбудованих темперацій (піфагорійська, середньо тонова, кома та інші);
- оркестрова налаштування з можливістю зміщення частот;
- наявність «тональної труби» для відтворення еталонних частот.

До технічних обмежень застосунку можна зарахувати:

- відсутність апаратної реалізації, тобто обмежена автономність;
- може бути недостатньо точним в умовах фонових шумів.

DaTuner: Tuner & Metronome

DaTuner – це універсальний, точний та чуйний хроматичний тюнер і метроном для пристроїв Android. Додаток вирізняється інтуїтивно зрозумілим інтерфейсом і простотою у використанні.



Рисунок 1.3 – Інтерфейс DaTuner: Tuner & Metronome

До ключових функціональних переваг застосунку можна віднести:

- висока точність та чутливість;
- автоматичне налаштування чутливості;
- зміна кольору екрану при досягненні правильної частоти;
- можливість фіксації ноти навіть при значному відхиленні від еталонної частоти;
- ручне або автоматичне встановлення опорної частоти;
- широкий діапазон частоти дискретизації (8–48 кГц).

До технічних обмежень застосунку можна зарахувати:

- призначений переважно для індивідуального використання;
- відсутність фізичних елементів для безпосередньої інтеграції з іншими пристроями.

GuitarTuna: Tune & Play Guitar

GuitarTuna – один із найпопулярніших мобільних тюнерів, який завантажено понад 100 мільйонів разів, і ця кількість тільки зростає. Додаток орієнтований переважно на гітаристів і включає навчальні функції.

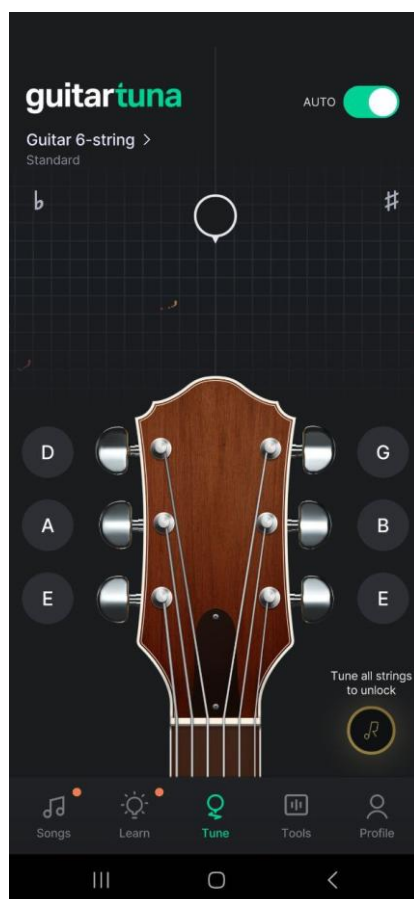


Рисунок 1.4 – Інтерфейс GuitarTuna: Tune & Play Guitar

До ключових функціональних переваг застосунку можна віднести:

- 1) просте та швидке налаштування різних струнних інструментів;
- 2) плавне перемикання між тюнером для гітари, тюнером для укулеле, тюнером для бас-гітари тощо;
- 3) підтримка великої бібліотеки акордів і табулятур (понад 20000 пісень);
- 4) навчальні посібники та персоналізований зворотний зв'язок;
- 5) включає в себе такі інструменти як:
 - метроном, що дозволить встановити темп;
 - тренажер для слуху, що покращить музичні навички в розпізнаванні акордів на слух;
 - режим для ліворуких користувачів.

До технічних обмежень застосунку можна зарахувати:

- основна орієнтація: лише на струнні інструменти;
- безкоштовна версія має обмежений функціонал.

Audacity

Audacity – багатофункціональне безкоштовне програмне забезпечення з відкритим кодом для запису та редагування аудіо.

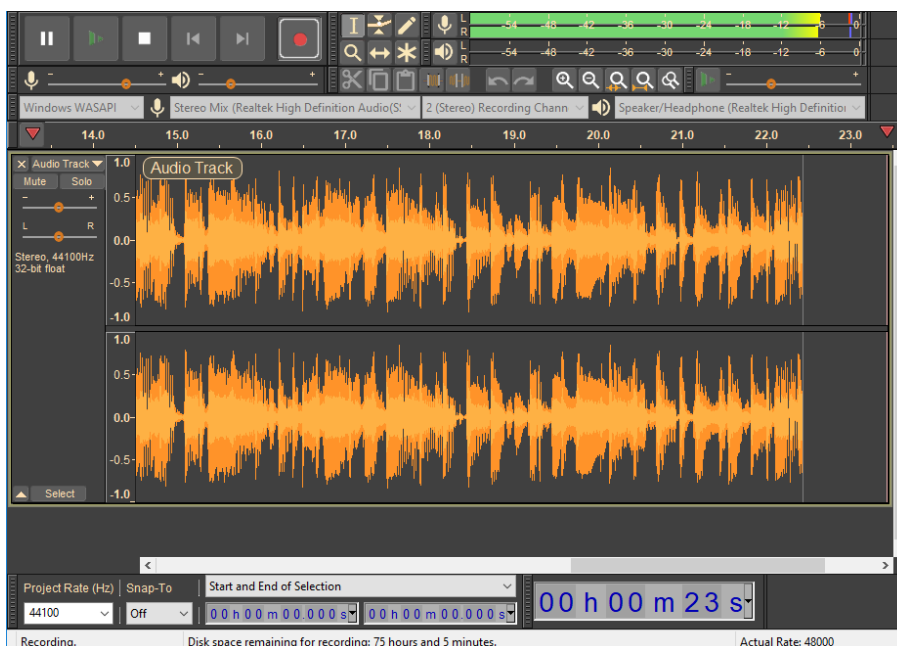


Рисунок 1.5 – Інтерфейс Audacity

До ключових функціональних переваг застосунку можна віднести:

- повна підтримка редагування аудіо у різних форматах;
- велика кількість ефектів і плагінів;
- відкритий вихідний код;
- можливість спектрального аналізу звуку.

До технічних обмежень застосунку можна зарахувати:

- потребує навичок і знань у сфері аудіо обробки;
- не призначений спеціально для музичного налаштування.

Серед переваг застосунків для визначення звуку можна виокремити високу точність, підтримку різних інструментів, можливість індивідуального налаштування, а також наявність додаткових функцій, які полегшують процес налаштування музичного інструменту.

Водночас, серед основних недоліків застосунків варто визначити залежність від мобільного або комп'ютерного пристрою (що знижує автономність системи),

зменшення точності в умовах фонових шумів, а також обмежений функціонал у безкоштовних версіях.

1.4 Алгоритми розпізнавання звуку

Алгоритми розпізнавання звуку – це методи аналізу та інтерпретації звукових сигналів з метою виділення характеристик. Вони використовуються у таких сферах, як налаштування музичного обладнання, розпізнавання мови, аудіо аналіз.

1.4.1 Автокореляція

Автокореляція, також відома як серійна кореляція – це кореляція сигналу із його зміщеною копією як функція затримки. Неофіційно це подібність між спостереженнями залежно від тимчасового зсуву між ними. Аналіз автокореляції є математичним інструментом для виявлення повторюваних шаблонів, таких як періодичного сигналу, прихованого шуму, або виявлення відсутньої основної частоти сигналу, яка впливає з його гармонійних складових. Автокореляція часто використовується в обробці сигналів для аналізу функцій або рядів значень, такий як сигнали у часовій області [9].

Принцип дії автокореляційного методу:

- розбиття аудіо сигналу на короткі часові вікна для локального аналізу;
- для кожного фрейму обчислюється автокореляційна функція, яка визначає ступінь подібності сигналу до самого себе;
- визначення піку автокореляції, що відповідає найбільшій подібності та вказує на період основної частоти;
- обчислення частоти.

Метод автокореляції вирізняється простотою реалізації, що дозволяє застосувати його у пристроях з обмеженими обчислювальними ресурсами. Завдяки здатності ефективно виявляти періодичність сигналу, він забезпечує високу точність з голосовими даними або музичними інструментами, у яких присутня чітка періодична структура.

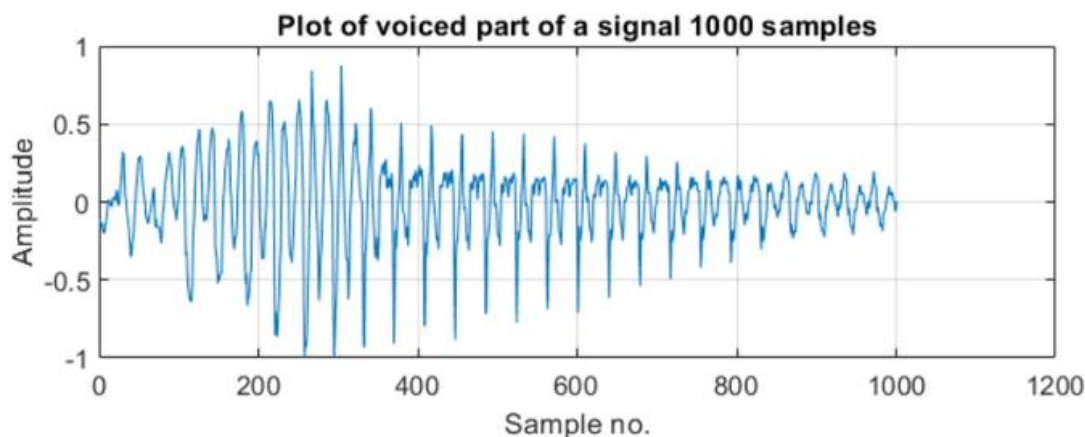


Рисунок 1.6 – Графік озвученої частини сигналу (1000 відліків) [7]

Разом із тим метод має і певні обмеження, він демонструє високу чутливість до фонових шумів, що може призводити до виникнення помилкових піків в автокореляційній функції, крім того, метод менш ефективний для обробки сигналів з низькою або відсутньою періодичністю, що обмежує його застосування у випадках складної акустичної структури.

1.4.2 Метод періодограм

Періодограма використовується для виявлення домінантних періодів (або частот) часових рядів. Це може бути корисним інструментом для виявлення основної циклічної поведінки в ряді, особливо коли цикли не пов'язані зі звичними коливаннями [29].

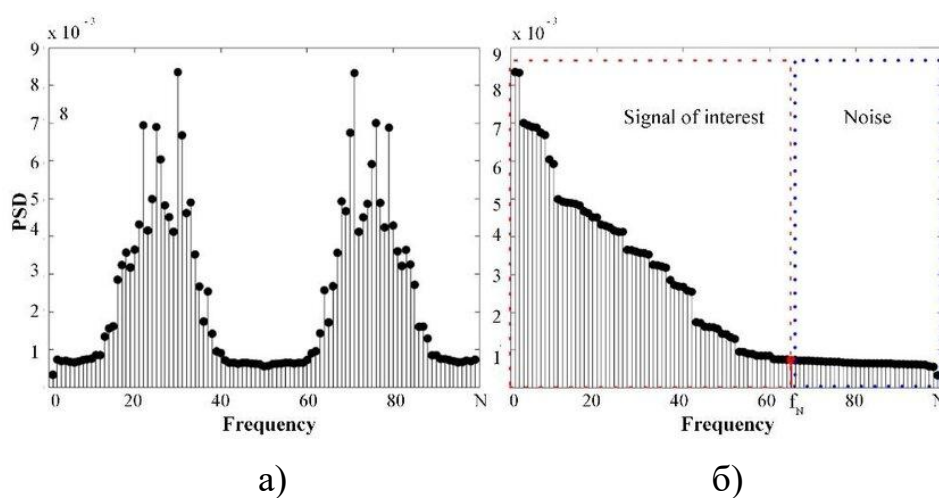


Рисунок 1.7 – Метод оцінки потужності шуму: а) періодограма сигналу;
б) відсортована за спаданням періодограми [13]

Принцип дії методу періодограми:

- сигнал ділиться на короткі часові фрейми для локального спектрального аналізу;
- до кожного фрейму застосовується віконна функція для зменшення спектральних витікань;
- для кожного фрейму обчислюється квадрат модуля дискретного перетворення Фур'є (англ. Discrete Fourier Transform, DFT), що дає оцінку спектральної щільної потужності;
- періодограми всіх фреймів усереднюються для отримання загальної оцінки спектру сигналів.

Серед основних переваг методу періодограм можна визначити високу роздільну здатність, яка забезпечує детальний спектральний аналіз, що дозволяє точно визначати компоненти сигналу. Метод також простий у реалізації, оскільки періодограма є прямим способом оцінки спектру без необхідності складних математичних моделей.

Однак метод має й деякі обмеження. Зокрема, він є чутливим до шуму, особливо при обробці коротких фреймів, що може призвести до значних варіацій в оцінках спектру. Крім того, при використанні коротких фреймів роздільна здатність за частотою знижується, що спричиняє злиття близьких частотних компонентів і ускладнює точне визначення спектральних характеристик сигналу.

1.4.3 Швидке перетворення Фур'є

Для реалізації системи планується використати FFT – алгоритм, який розкладає звуковий сигнал на його частотні складові, що дозволяє точно визначити частоту сигналу (рис. 1.8).

FFT використовує принцип, за яким множення в частотній області відповідає згортці у часовій області. Спочатку вихідний сигнал перетворюється у частотну область за допомогою DFT, потім його спектр множиться на частотну характеристику фільтра, і після цього результат повертається у часову область через обернене DFT.

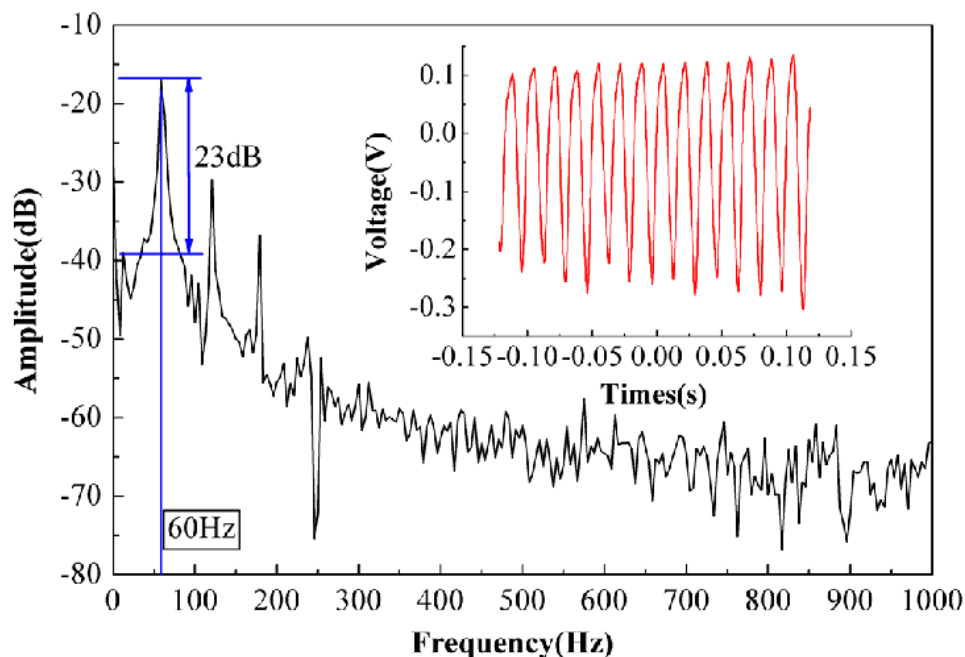


Рисунок 1.8 – Спектр у часовій та частотній області [32]

Цей підхід був відомий з часів Фур'є, проте практично не використовувався через складність обчислень. Ситуація змінилася у 1965 році, коли було розроблено алгоритм швидкого перетворення Фур'є. Він дозволив значно скоротити кількість обчислень і зробив частотну обробку сигналів більш ефективною.

Обчислювальна складність. Пряма реалізація дискретного перетворення Фур'є має обчислювальну складність $O(N^2)$, що робить її повільною при великій кількості точок. Саме для цього використовується алгоритм FFT, який реалізує ту ж операцію, але з набагато меншою складністю $O(N \log N)$. Це стало можливим завдяки ефективному розбиттю обчислень на менші підзадачі, використовуючи симетрію та періодичність експоненціальних функцій [31].

Математичною основою FFT є перетворення Фур'є, яке розкладає функції часу (сигналу) на складові частоти [5]:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt, \quad (1.2)$$

де $X(f)$ – представлення в частотній області;

f – частота;

t – час;

$e^{-j2\pi ft}$ – комплексна експоненціальна функція.



Рисунок 1.9 – Блок-схема аналізатора спектру FFT

Алгоритм FFT ефективно обчислює це перетворення для дискретних сигналів, що робить його можливим для обробки навіть у застосунках реального часу. У процесі роботи системи також буде використано акустичний спектр, який являє собою сукупність частот звукових хвиль, аналіз яких допоможе досягти максимальної точності при розпізнаванні нот (рис. 1.9).

1) Зчитування сигналу з мікрофона – система приймає аналоговий звуковий сигнал з мікрофона. Це може бути звук музичного інструмента, людського голосу чи будь-яке інше акустичне джерело.

2) Аналого-цифрове перетворення (АЦП) – перед оцифруванням сигнал проходить підсилення або ослаблення за допомогою підсилювача з автоматичним регулюванням гучності. Потім, якщо потрібно, застосовується фільтр нижніх

частот для усунення шумів. Далі сигнал дискретизується та перетворюється на цифровий – система отримує послідовність чисел, що відображають амплітуду сигналу в часі.

3) Буферизація сигналу – цифрові значення зберігаються у буфері до накопичення достатньої кількості, необхідної для виконання частотного аналізу.

4) Застосування віконної функції – до зібраного масиву зразків застосовується віконна функція, яка зменшує ефекти “витікання спектру” при подальшому перетворенні.

5) Обчислення FFT – за допомогою алгоритму FFT сигнал перетворюється з часової області у частотну. Це дозволяє визначити наявні у сигналі частоти.

6) Розрахунок амплітуди для кожної частоти – з комплексних значень, отриманих після FFT, обчислюються амплітуди кожної частоти ці значення показують, наскільки сильно виражена кожна з частот у сигналу.

7) Нормалізація та масштабування значень спектру – для кращої візуалізації або інтерпретації амплітудні значення масштабуються або нормалізуються до зручного діапазону.

8) Виведення спектру – результати частотного аналізу виводяться на дисплей у вигляді спектра, значень частот або назв відповідних музичних нот.

Порівнюючи розглянуті методи, можна зробити висновок, що кожен з них має свої переваги та обмеження. Автокореляційний метод є простим у реалізації та ефективним для періодичних сигналів, але чутливий до шуму. Метод періодограми забезпечує детальний спектральний аналіз, але може мати високу варіацію при коротких фреймах. FFT, у свою чергу, поєднує в собі високу ефективність та точність, що робить його оптимальним вибором для задач розпізнавання нот у реальному часі.

1.5 Аналіз вимог до системи

У проєкті передбачається створення системи для визначення музичних нот на основі частоти звукових коливань. Для цього необхідно врахувати деякі аспекти, деякі з них як апаратні та програмні вимоги.

Загальні функціональні вимоги

Система повинна мати можливість аналізувати звукові коливання, які сприймає мікрофон, і визначати відповідну музичну ноту. Для цього повинно бути реалізовано FFT, який дозволить виділити основну частоту сигналу.

Обробку сигнали повинна здійснюватися в реальному часі без помітних затримок – це забезпечить точне та своєчасне визначення частоти звуку та її відповідності музичній ноті.

Результати аналізу мають виводитися на OLED-дисплей з інтерфейсом I2C, що гарантує швидку передачу даних між мікроконтролером і дисплеєм.

Програмне забезпечення повинно бути гнучким, передбачати можливість зміни порогових значень, оновлення бази частот та додавання нових функцій. Це забезпечить масштабування та адаптації системи до майбутніх потреб.

Апаратні вимоги

Мікрофонний модуль – це ключовий елемент для захоплення звуку. Він перетворює акустичні коливання на електричний сигнал. Важливо щоб мікрофон мав високу чутливість та низький рівень шумів та спотворень для точної і стабільної роботи системи.

Мікроконтролер відповідає за оцифрування сигналу та виконання обчислень за алгоритмом FFT для визначення частоти до відповідної ноти. Мікроконтролер повинен мати достатню обчислювальну потужність для роботи в реальному часі.

Дисплей використовується для виведення результатів – значення частоти та назви відповідної музичної ноти. Інтерфейс I2C забезпечуватиме швидке з'єднання з мікроконтролером і мінімальне навантаження на апаратну частину.

Програмне забезпечення

Arduino Integrated Development Environment (IDE) – містить текстовий редактор для написання коду, область повідомлень, текстову консоль, панель інструментів з кнопками для загальних функцій і ряд меню. Він підключається до обладнання Arduino для завантаження програм і зв'язку з ними [22].

fix_fft – це бібліотека для реалізації швидкого перетворення Фур'є з фіксованою комою в режимі «на місці» для мікроконтролерів Arduino. Хоча вона

жертвує певною точністю, але натомість працює набагато швидше, порівняно з реалізаціями на основі чисел з плаваючою комою. За допомогою цієї бібліотеки можна обчислити частоту дискредитованого сигналу [12].

Adafruit_SSD1306 – це бібліотека призначена для роботи з монохромними OLED-дисплеями на основі драйверів SSD1306. Дисплеї використовують інтерфейси I2C або SPI, для підключення потрібно від 2 до 5 контактів. Adafruit активно підтримує спільноту відкритого програмного забезпечення, тому бібліотека є безкоштовною та відкритою для використання й модифікації [10].

Adafruit GFX – це основна графічна бібліотека для дисплеїв, розроблених компанією Adafruit. Вона забезпечує спільний набір графічних примітивів таких як точки, лінії, кола тощо. Цю бібліотеку потрібно використовувати разом із бібліотекою, специфічною для апаратного забезпечення, для кожного пристрою відображення, який ми маємо (для обробки функцій нижчого рівня) [11].

Висновки до розділу 1

У першому розділі КБР було розглянуто концепцію створення системи для визначення музичних нот на основі аналізу звукових коливань. Проаналізовано основні фізичні принципи акустичних хвиль, методи їх перетворення в цифрові сигнали, а також технічні засоби для обробки цих сигналів.

Було проведено огляд існуючих альтернативних системи для визначення звуку, на основі проведеного аналізу сформульовано обґрунтування необхідності створення власної системи, яка враховує наявні проблеми та обмеження існуючих рішень.

Окрему увагу приділено визначенню вимог до майбутньої системи, зокрема: чутливості мікрофону, швидкодії мікроконтролера, точності алгоритму та ефективності виведення результатів на дисплей.

Таким чином використання мікроконтролера, мікрофонного модуля, дисплея та алгоритму FFT у цій системі допоможе забезпечити розпізнавання звуків у режимі реального часу. Завдяки своїй простоті та універсальності такий пристрій має потенціал широкого практичного застосування.

2 ПРОЄКТУВАННЯ СИСТЕМИ ТА ВИБІР СКЛАДОВИХ СИСТЕМИ ВИЗНАЧЕННЯ НОТ

У другому розділі було детально розглянуто кожен компонент системи, обґрунтовано вибір кожного з них шляхом порівняння з аналогічними елементами, а також наведено основні характеристики та принцип роботи відповідних модулів.

2.1 Концепція системи визначення нот

Система визначення нот – це концепція аналізу звукового спектра, що дозволяє отримати музичні ноти на основі вхідного аудіо сигналу. В основі роботи лежить зчитування аналогового сигналу з мікрофону, його оцифрування мікроконтролером, аналіз частотного спектра за допомогою алгоритму FFT (рис. 2.1, блок схему у повному розмірі винесено до Додатку А), та визначення домінуючої частоти. Відповідна нота яка збігається з цією частотою буде відображена на дисплеї.

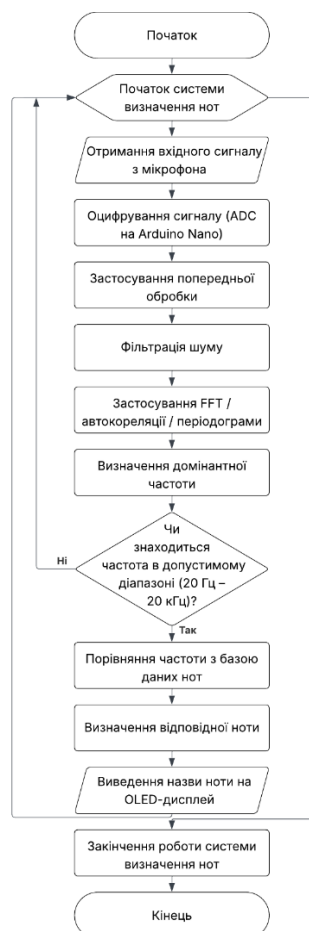


Рисунок 2.1 – Частина блок схеми системи

Для реалізації системи використовується мікроконтролер Arduino Uno, який дозволить пристрою функціонувати автономно у режимі реального часу, з мінімально можливими затримками. Такий підхід дозволить зробити систему зручною для використання для різних умов, зокрема як тюнер для налаштування музичних інструментів, а також для освітніх цілей, наприклад для вивчення нотного ряду та покращення сприйняття нот на слух.

Система включає в собі апаратну частину (мікроконтролер, мікрофон, дисплей) та програмну частину з використанням відповідних бібліотек що будуть відповідати за обробку сигналу і виведення результатів на екран, також систему можна буде модифікувати під різні завдання, замінити або покращити компоненти, що робить систему гнучкою та придатною для подальшого вдосконалення.

2.2 Мікроконтролер

Мікроконтролер – це спеціалізований мікроелектронний програмований прилад, що призначений для використання у керуючих пристроях, системах передачі даних та системах керування технологічними процесами. Він поєднує в собі центральний процесор, пам'ять і периферійні модулі введення та виведення інформації.

Завдяки своїй універсальності, мікроконтролери широко застосовують в побутовій техніці, медичних приладах, телефонах, електронних музичних інструментах, комп'ютерній периферії, світлофорах, автоматичних системах керування, автомобілях, роботах та промислових установках.

Принцип роботи мікроконтролера полягає в послідовному виконанні команд, записаних у його програмній пам'яті. Він зчитує сигнали із зовнішніх пристроїв, обробляє ці дані відповідно до заданої програми, та видає керуючі сигнали для виконання певних дій. Завдяки цьому мікроконтролер здатен автономно керувати складними електронними системами з мінімальним втручанням людини.

2.2.1 Вибір мікроконтролеру

Arduino Uno

Плата Arduino Uno є найпопулярнішою контролерною платою в сімействі Arduino і використовується в усьому світі. Вона є найкращим вибором для початківців. Arduino Uno – це плата мікроконтролера на основі ATmega328. Вона має 14 цифрових входів/виходів, 6 аналогових входів, кварцовий генератор на 16 МГц, USB-роз'єм, роз'єм живлення, заголовок ICSP та кнопку скидання (reset).



Рисунок 2.2 – Arduino Uno [6]

Таблиця 2.1 – Arduino Uno технічні характеристики

Мікроконтролер	ATmega328P
Робоча напруга, В	5,0
Рекомендована напруга живлення, В	7,0–12,0
Аналогові входи	6 (A0 – A5)
Цифрові I/O піни	14 (із них 6 підтримують PWM-вивід)
Макс. струм на пін I/O, мА	40
Струм на 3,3 В пін, мА	50
Flash-пам'ять, кбайт	32 (0.5 з яких використовується завантажувачем Bootloader)
SRAM, кбайт	2
EEPROM, кбайт	1
Тактова частота (Clock Speed), МГц	16
Інтерфейси зв'язку	I2C, SPI, USART

Uno відрізняється від усіх попередніх плат тим, що не використовує чіп FTDI для перетворення USB у послідовний інтерфейс. Натомість у ній використовується мікроконтролер Atmega8U2, запрограмований як перетворювач USB на UART.

Arduino Nano

Arduino Nano – це мініатюрна версія контролера для макетних плат із інтегрованим USB-інтерфейсом. Це найменша, повноцінна та зручна для використання на breadboard-платі (макетній платі) версія Arduino. Вона має все те ж саме, що й плати Diecimila/Duemilanove (з електричної точки зору), але з більшою кількістю аналогових входів та вбудованим перемикачем AREF +5V.

Фізично Nano не має роз'єму для живлення (power jack). Проте вона автоматично визначає джерело з вищою напругою і перемикається на нього немає потреби вручну обирати джерело живлення за допомогою джампера. Nano поєднує зручність підключення як у Boarduino та MiniBus, але при цьому має менший розмір, що дає більше вільного місця на макетній платі.

Схема розташування пінів (контактів) пристрою сумісна з Arduino Mini або Basic Stamp: TX, RX, ATN та GND розташовані з одного боку, а живлення та заземлення – з іншого.

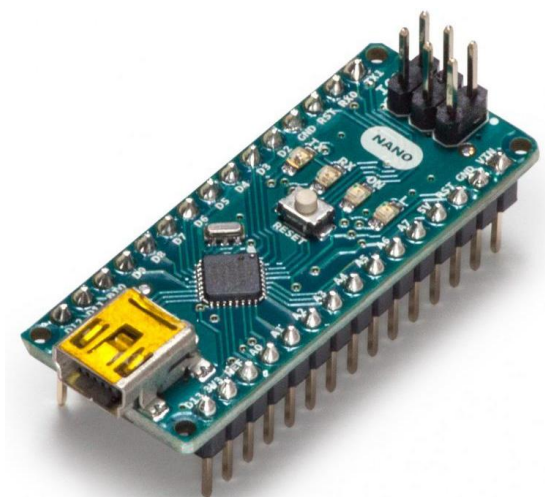


Рисунок 2.3 – Arduino Nano [6]

Arduino Nano побудований на базі мікроконтролера ATmega168 і має приблизно таку ж функціональність, як Arduino Duemilanove, але у компактнішому форматі. Єдиним недоліком є відсутність роз'єму живлення постійного струму, 2025 р.

проте він працює з кабелем Mini-USB замість стандартного USB. Nano був розроблений і виготовляється компанією Gravitech.

Таблиця 2.2 – Arduino Nano технічні характеристики

Мікроконтролер	ATmega328P
Робоча напруга, В	5,0
Рекомендована напруга живлення, В	7,0–12,0
Аналогові входи	6 (A0 – A5)
Цифрові I/O піни	14 (із них 6 підтримують PWM-вивід)
Макс. струм на пін I/O , мА	40
Струм на 3,3 В піні, мА	50
Flash-пам'ять, кбайт	32 (2 з яких використовується завантажувачем Bootloader)
SRAM, кбайт	2
EEPROM, кбайт	1
Тактова частота (Clock Speed), МГц	16
Інтерфейси зв'язку	I2C, SPI, USART

Arduino Mega

Arduino Mega 2560 – це плата мікроконтролера, побудована на основі ATmega2560. Вона має 54 цифрових входи/виходи (з яких 14 можуть використовуватись як PWM-виходи), 16 аналогових входів, 4 апаратні UART (послідовні порти), кварцовий резонатор на 16 МГц, USB-з'єднання, роз'єм живлення, ICSP-роз'єм та кнопку скидання. Mega сумісна з більшістю шилдів, розроблених для Arduino Duemilanove або Diecimila.

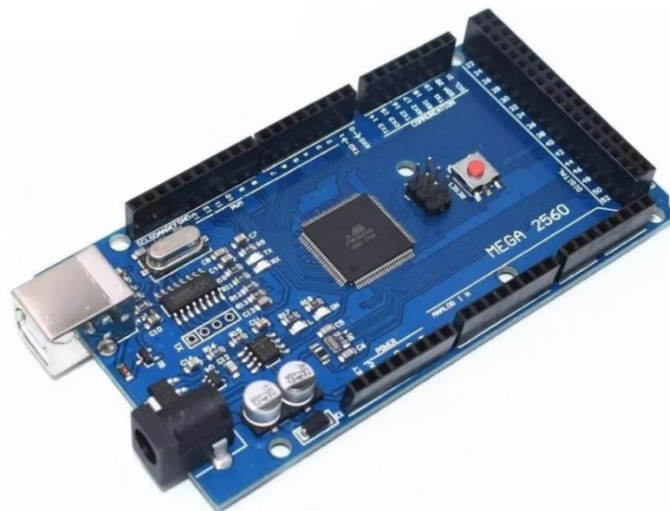


Рисунок 2.4 – Arduino Mega [6]

Таблиця 2.3 – ARDUINO MEGA технічні характеристики

Мікроконтролер	ATmega2560
Робоча напруга, В	5,0
Рекомендована напруга живлення, В	7,0–12,0
Аналогові входи	16 (A0 – A15)
Цифрові I/O піни	54 (із них 15 підтримують PWM-вивід)
Макс. струм на пін I/O, мА	40
Струм на 3,3 В піні, мА	50
Flash-пам'ять, кбайт	256 (8 з яких використовується завантажувачем Bootloader)
SRAM, кбайт	8
EEPROM, кбайт	4
Тактова частота (Clock Speed), МГц	16
Інтерфейси зв'язку	ІС, SPI, USART

Для розробки системи визначення нот на основі звукових коливань було обрано Arduino Nano. Основним критерієм вибору стала компактність плати, що дозволяє зручно розміщувати її на макетній платі разом з іншими компонентами, такими як мікрофонний модуль, OLED-дисплеєм та з іншими допоміжними елементами.

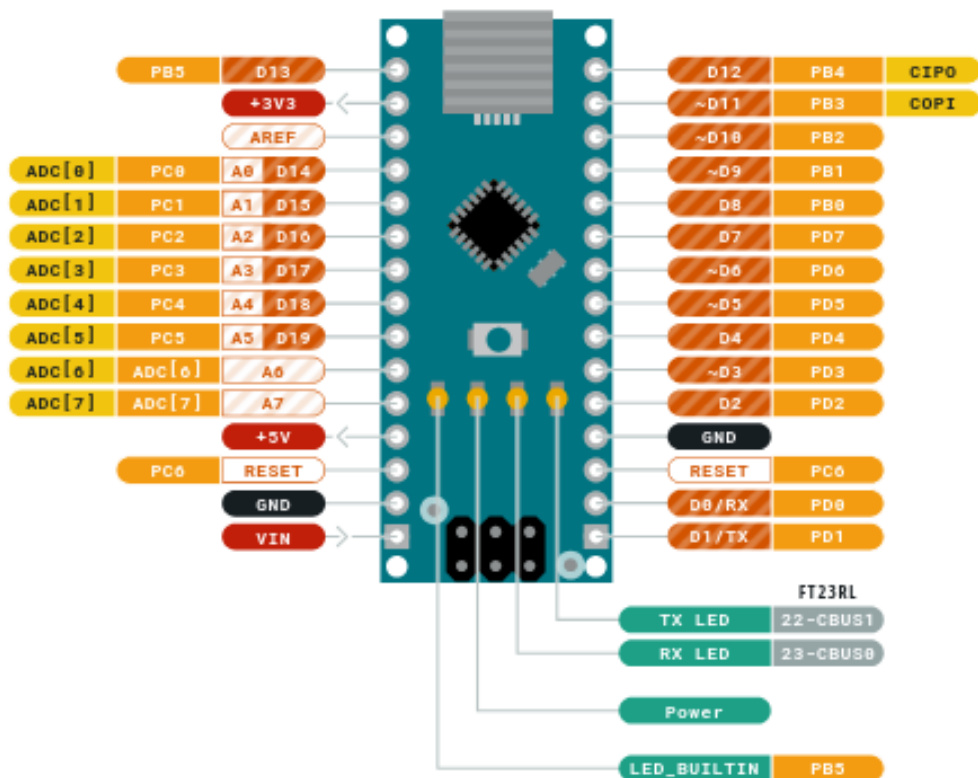


Рисунок 2.5 – Розпіновка плати [17]

Arduino Nano має ті ж самі технічні характеристики, що й Arduino Uno – мікроконтролер ATmega328P, 14 цифрових пінів (з них 6 підтримують PWM), 6 аналогових входів, підтримку I2C, SPI та UART інтерфейсів, а також робочу частоту 16 МГц.

Arduino Nano забезпечує достатній обсяг пам'яті (32 КБ flash, 2 КБ SRAM), щоб зберігати необхідну логіку обробки сигналу, аналізу частот та відображення результатів на екрані.

Таким чином, Arduino Nano є оптимальним вибором для проєкту: вона забезпечує всі необхідні апаратні ресурси, має зручний форм-фактор для компактного монтажу та сумісна з численними бібліотеками і сенсорами, необхідними для реалізації функціоналу системи.

2.3 Мікрофон

Мікрофон – це пристрій, який захоплює звук і перетворює його на електричний сигнал. Він відіграє ключову роль у записі аудіо, підсиленні звуку та забезпеченні зв'язку через телефони, комп'ютери, динаміки та інші пристрої. Мікрофони широко використовуються в таких сферах, як музика, телерадіомовлення, публічні виступи, подкасти та відеозйомка.

Принцип роботи мікрофона полягає в тому, що звукові хвилі спричиняють коливання тонкої мембрани всередині пристрою. Ці коливання передаються на котушку, яка рухається в магнітному полі та генерує електричний струм. Таким чином створюється електричний сигнал, що відповідає звуковому, і який можна записати або передати далі.

Робота мікрофона ґрунтується на законі збереження енергії: енергія звукових хвиль перетворюється на електричну. Це дозволяє точно фіксувати акустичну інформацію та використовувати її для подальшої цифрової обробки або відтворення.

2.3.1 Види мікрофонів

Аналогові мікрофони

Аналогові мікрофони виводять безперервний електричний сигнал, який відображає амплітуду звукових хвиль. Їх легко підключати до аналогових входів мікроконтролера, вони часто використовуються у простих проєктах.

До переваг аналогового мікрофону можна віднести:

- простота підключення та використання;
- підходить для проєктів, де важливий контроль над аналоговим сигналом.

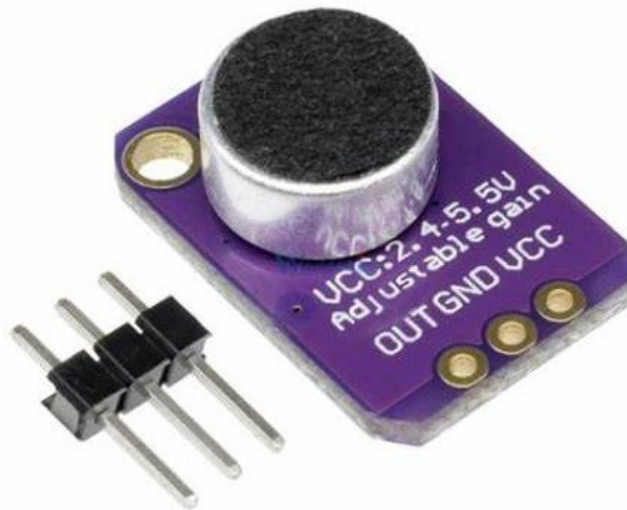


Рисунок 2.6 – Аналоговий мікрофон [16]

До недоліків можна зарахувати:

- вразливість до шуму та перешкод;
- вимагає аналогово-цифрового перетворення (англ. Analog-to-digital converter, ADC), що може зменшити точність, якщо мікроконтролер має обмежену розрядність;
- залежність від рівня сигналу може вимагати зовнішнього підсилювача.

Цифрові мікрофони

Цифрові мікрофони, виводять уже оцифрований сигнал, який легко аналізувати за допомогою відповідного програмного забезпечення. Вони

забезпечують високу точність і стабільність сигналу, оскільки не залежать від якості АЦП мікроконтролера.



Рисунок 2.7 – Цифровий мікрофон [25]

До переваг цифрового мікрофону можна віднести:

- висока точність та низький рівень шуму;
- пряма передача цифрового сигналу менше спотворень.

До недоліків можна зарахувати:

- потребує мікроконтролера з підтримкою I2S або PDM інтерфейсу (наприклад, ESP32, STM32);
- ускладнене програмне забезпечення для роботи з цифровими потоками.

Звукові сенсори

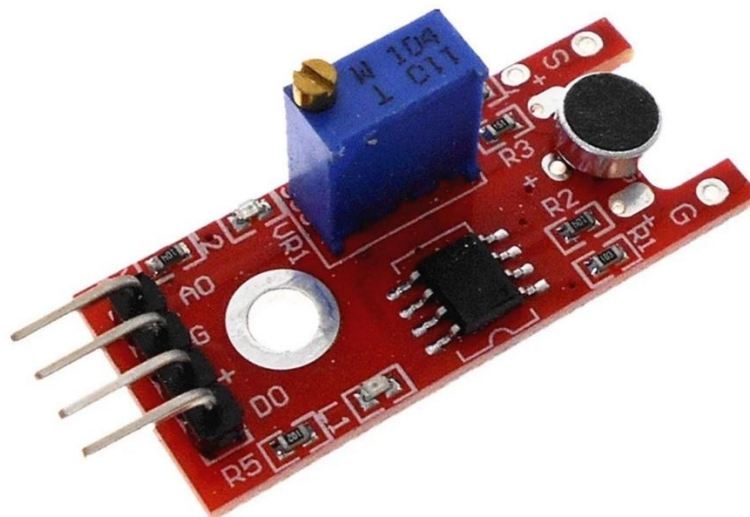


Рисунок 2.8 – Звуковий сенсор [15]

Ці модулі не передають звукову інформацію, а лише повідомляють про наявність звуку (наприклад, клацання або гучний шум). Найчастіше використовуються в автоматизації або сигналізації.

До переваг звукових сенсорів можна віднести:

- дуже дешеві та прості у використанні;
- ідеальні для виявлення наявності або відсутності звуку;
- не потребують обчислювальних ресурсів для обробки сигналу.

До недоліків можна зарахувати:

- неможливо визначити частоту або амплітуду звуку;
- низька точність та чутливість;
- не підходять для музичних або аудіо аналітичних застосувань.

Micro-Electro-Mechanical Systems (MEMS) мікрофони

Це сучасні мікрофони, які поєднують високу якість, компактність і надійність. Вони можуть бути як аналоговими, так і цифровими. MEMS мікрофони широко використовуються у смартфонах, ноутбуках та портативних пристроях через їх точність і стабільність.



Рисунок 2.9 – MEMS мікрофон [1]

До переваг MEMS можна віднести:

- компактні розміри, низьке енергоспоживання;
- підвищена чутливість і точність;
- доступні в обох форматах аналоговому та цифровому.

До недоліків можна зарахувати:

- можуть бути дорожчими за звичайні аналогові мікрофони;
- деякі моделі потребують точного налаштування живлення або спеціального підключення.

У розділі проаналізовано різні типи мікрофонів, з-поміж яких для реалізації системи було обрано аналоговий мікрофон через його простоту підключення, сумісність із мікроконтролером та можливість гнучкого контролю над вхідним сигналом, що є достатнім для задачі визначення музичних нот.

2.3.2 Вибір мікрофону

Для розробки системи було обрано мікрофонний модуль MAX9814 – аналоговий мікрофон із вбудованою функцією автоматичного керування підсиленням (англ. Automatic Gain Control, AGC). Завдяки AGC гучні звуки автоматично приглушуються, щоб уникнути перевантаження сигналу, а тихі – підсилюються для кращої чутності. Це дозволяє адаптуватися до змін рівня звуку, що є важливим для точної фіксації частоти звукових коливань. Така особливість робить MAX9814 ефективним рішенням для роботи в умовах зі змінною гучністю, без необхідного ручного регулювання підсилення.

Таблиця 2.4 – Характеристики мікрофону

Напруга живлення, В	2,7– 5,5
Споживаний струм, мА	3
Вихідний сигнал, В	2 від піку до піку при зміщенні 1,25
Частотна характеристика, Гц	20–20000
Коефіцієнт атаки та спаду	Програмований
Автоматичне посилення, дБ	Вибір між 40, 50 або 60
Вхідна густина шуму, нВ/Гц	30
Розміри, мм	25 × 14, діаметр – 2, відстань від центру до центра – 10

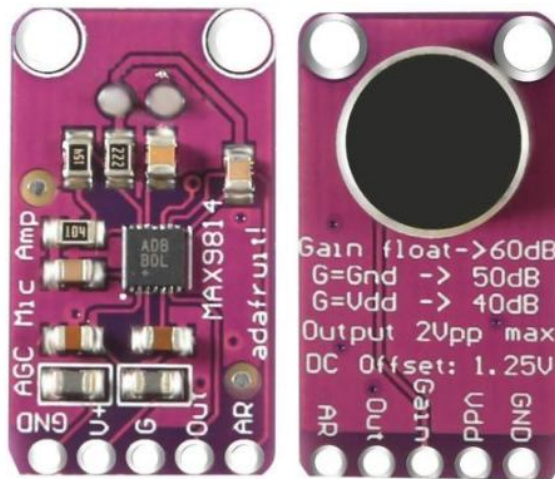


Рисунок 2.10 – Мікрофонний модуль MAX9814 [2]

Порівняно з типовими аналоговими мікрофонами, MAX9814 забезпечує нижчий рівень шуму та не потребує зовнішнього підсилювача. На відміну від цифрових мікрофонів, він легко підключається до аналогових входів Arduino Nano, не вимагаючи використання спеціалізованих інтерфейсів. Усе це робить MAX9814 оптимальним вибором для системи, яка потребує простого підключення, стабільного сигналу та високої чутливості.

2.4 Дисплей

Дисплей – це екран, який відображає зображення та текст, отримані від комп'ютера або іншого пристрою. Він дозволяє користувачам взаємодіяти з електронними системами, такими як комп'ютери, смартфони чи телевізори, надаючи візуальну інформацію у вигляді текстів, графіки чи відео. Для керування пристроєм зазвичай використовують клавіатури, сенсорні панелі або інші засоби введення.

Дисплей складається з великої кількості крихітних точок – пікселів, які світяться різними кольорами для формування зображення. Кожен піксель утворюється трьома субпікселями (червоним, зеленим і синім), які змінюють свою яскравість для створення потрібного кольору. Роздільна здатність екрана залежить від кількості пікселів на одиницю площі – чим їх більше, тим чіткіше зображення.

Принцип роботи дисплея залежить від його типу, але загальна ідея полягає в управлінні світлом або кольорами пікселів. Наприклад в OLED-дисплеях кожен піксель самостійно випромінює світло, завдяки чому зображення виходить яскравішим, а енергоспоживання – ефективнішим.

2.4.1 Інтерфейси підключення

Універсальні інтерфейси – це стандартизовані способи з'єднання та обміну даними між різними електронними пристроями або компонентами системи. Вони дозволяють здійснювати передачу інформації незалежно від виробника обладнання, завдяки чому забезпечується сумісність та взаємодія між модулями, сенсорами, мікроконтролерами, дисплеями тощо [14].

Ці інтерфейси, такі як I2C, SPI, UART тощо, широко використовуються в електроніці та вбудованих системах. Вони дозволяють оптимізувати зв'язок між пристроями за кількістю ліній, швидкістю передачі даних, енергоспоживанням та складністю реалізації. Завдяки універсальності таких інтерфейсів, інженери можуть гнучко комбінувати різні компоненти в єдину функціональну систему.

Serial Peripheral Interface (SPI)

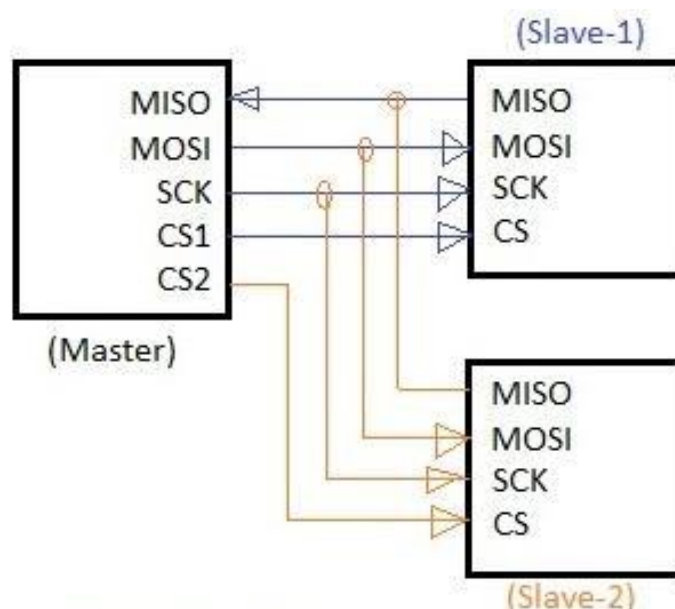


Рисунок 2.11– SPI Інтерфейс [30]

Інтерфейс SPI – це послідовний інтерфейс, який використовується для обміну даними між головним пристроєм, який у SPI називається Master (ведучий), та пристроями Slave (ведені). Один головний пристрій може взаємодіяти з багатьма підлеглими. Щоб вибрати певний пристрій Slave, використовується лінія вибору мікросхеми. Для передачі даних використовуються дві лінії: вихід Master і вхід Master. Обов'язковою умовою є визначення тактового сигналу, що синхронізує дані, оскільки це інтерфейс з синхронізацією по тактовому імпульсу.

Inter-Integrated Circuit (I2C)

Інтерфейс I2C – це послідовний інтерфейс, який дозволяє обмінюватися даними між мікроконтролером та іншим пристроями. Він використовує лише дві лінії: одна передає тактовий сигнал (SCL), а інша – дані (SDA). Обидві лінії з'єднують усі пристрої на шині, що дозволяє будувати прості й ефективні схеми підключення.

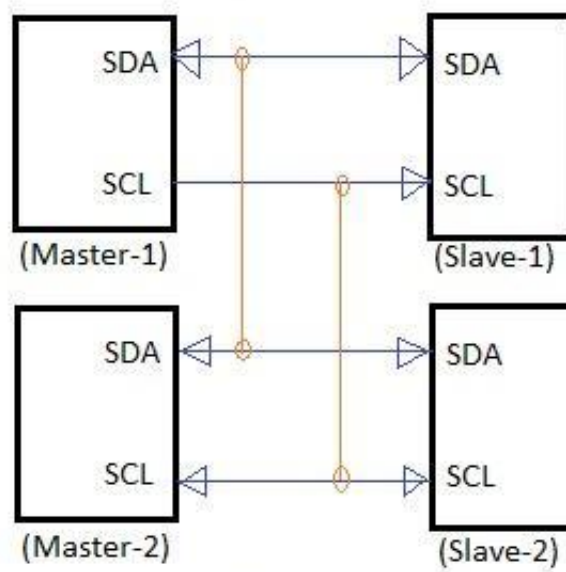


Рисунок 2.12 – I2C Інтерфейс [30]

Перед початком передачі даних ведучий пристрій надсилає унікальну адресу підлеглого пристрою, до якого має бути здійснено звернення. Після цього відбувається обмін інформацією, яка може йти як від ведучого до підлеглого, так і навпаки. Важливо, що всі пристрої в мережі мають різні адреси, що дозволяє вести комунікацію з кількома пристроями одночасно через одну пару ліній.

Інтерфейс 12C має помірну швидкість передавання даних, однак цього зазвичай достатньо для роботи з датчиками, пам'яттю, годинниками реального часу або сенсорними екранами. Простота підключення та підтримка багатьох пристроїв на одній шині робить цей інтерфейс дуже популярним у вбудованих системах.

Universal Asynchronous Receiver-Transmitter (UART)

Інтерфейс UART – це асинхронний послідовний інтерфейс, який забезпечує обмін даними між двома пристроями за допомогою двох ліній: одна для передачі даних (TX), інша – для прийому (RX). На відміну від синхронних інтерфейсів, UART не використовує окремий тактовий сигнал, а обидва пристрої повинні бути налаштовані на однакову швидкість обміну – baud rate.

Перед передачею даних обидві сторони домовляються про параметри комунікації: швидкість, кількість біт даних, наявність бітів парності та стоп-бітів. Потім передача відбувається блоками – кожен байт даних супроводжується початковим і стоп-бітом, що дозволяє приймачу визначити межі інформації.

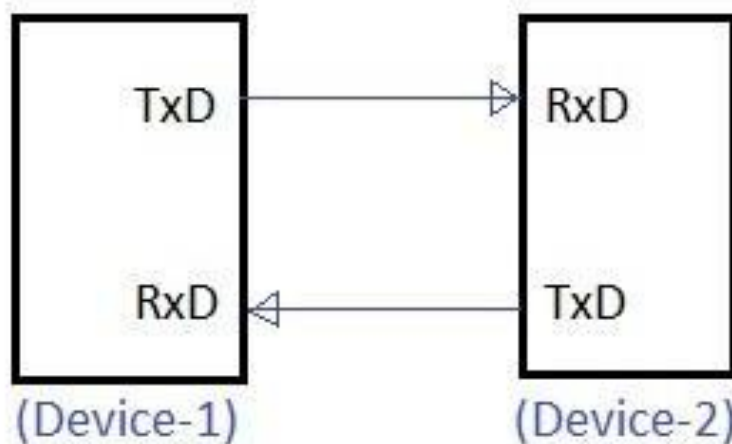


Рисунок 2.13 – UART Інтерфейс [30]

UART широко використовується для внутрішнього обміну даними між мікроконтролерами, модулями Bluetooth, GPS, Wi-Fi та іншими пристроями. Завдяки простій реалізації та достатній для більшості додатків швидкості, цей інтерфейс є базовим і зручним способом зв'язку у мікроелектроніці.

Таблиця 2.5 – UART vs SPI vs I2C [30]

Характеристика	UART	SPI	I2C
Позначення пінів	TxD: Передача даних RxD: Прийом даних	SCLK: Серійний тактовий сигнал MOSI: Master вихід, Slave вхід MISO: Master вхід, Slave вихід SS: Slave Select	SDA: Серійні дані SCL: Серійний тактовий сигнал
Швидкість передавання	Асинхронна передача потребує однакових швидкостей передачі. Максимум: ~230-460 Кбіт/с.	Не має визначеного максимуму. Типово: 10-20 Мбіт/с	100 Кбіт/с, 400 Кбіт/с, 3.4 Мбіт/с. Інші варіанти: 10 Кбіт/с і 1 Мбіт/с.
Дальність передачі	Низька, ~15 метрів	Висока	Висока
Тип зв'язку	Асинхронний	Синхронний	Синхронний
Кількість майстрів	Один	Один	Один або більше
Тактовий сигнал	Немає спільного тактового сигналу. Кожен пристрій має власний годинник	Спільний тактовий сигнал між майстром і підлеглим	Спільний тактовий сигнал між кількома майстрами та підлеглими
Складність апаратної частини	Найменша	Менша	Більша
Протокол	8 бітів даних + 1 стартовий біт + 1 стоповий біт	Протоколи залежать від виробника. Потрібно переглядати технічну документацію	Стартовий і стоповий біти. АСК після кожних 8 бітів даних
Програмна адресація	Не потрібна. Пряме з'єднання один-до-одного	Адресація підлеглих через сигнал вибору (SS). Для n пристроїв потрібно n SS-ліній	Можливість спілкування між кількома майстрами та підлеглими. До 127 пристроїв на шині.
Переваги	Проста, популярна, підтримується багатьма пристроями через роз'єм RS232. Малий обсяг протоколу	Повноцінний дуплекс. Окремі SS-лінії дозволяють використовувати кілька однакових чипів. Вища швидкість, більші відстані. Менше енергоспоживання, ніж у I2C	Низькі перехідні процеси завдяки відкритому колектору. Підтримка кількох майстрів. Лише 2 дроти. Проста адресація без SS. Гнучкість по напрузі. Використовує контроль потоку.
Недоліки	Підходить лише для з'єднання між двома пристроями. Потрібна однакова швидкість.	Зі збільшенням кількості підлеглих зростає кількість SS-ліній. Потрібно більше апаратних ресурсів і змін у ПЗ для додавання пристрою. Майстер/підлеглий – жорстка структура. Немає контролю потоку.	Зі збільшенням кількості пристроїв схема ускладнюється. Напівдуплекс. Потрібна програмна реалізація протоколу, що збільшує навантаження.

Інтерфейси UART, SPI та I2C мають свої унікальні особливості й сфери застосування: UART підходить для простих з'єднань один-до-одного, SPI забезпечує високу швидкість і гнучкість, а I2C дозволяє підключати багато пристроїв з мінімальною кількістю ліній.

2.4.2 Види дисплею

Існує два основні типи дисплеїв, які можна використовувати: LCD та OLED. Обидва дисплеї відрізняються за принципом роботи, вихідними характеристиками та робочою напругою, але виконують одну й ту ж функцію відображення зображень або тексту. Є також ще одна категорія дисплеїв – TFT LCD дисплеї. Вони зазвичай оснащені сенсорними панелями та використовуються в застосунках з графічним інтерфейсом (англ. Graphical user interface, GUI). [8]

LCD

LCD – це один з найпоширеніших типів екранів, які використовуються у мікроконтролерних проєктах. На рис. 2.14 показано LCD-дисплей розміром 16x2, що може виводити 2 рядки по 16 символів. LCD-дисплеї мають просту конструкцію, невелику вартість і достатню енергоефективність.

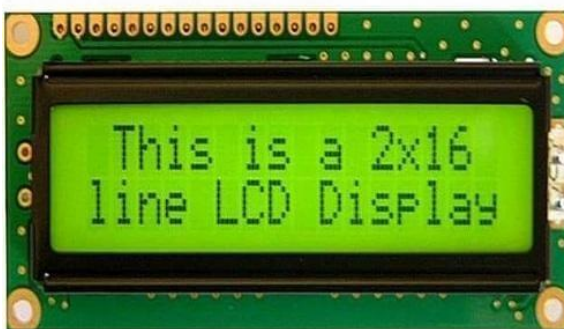


Рисунок 2.14 – LCD дисплей [8]

Вони часто використовуються для виведення тексту, простих повідомлень або меню користувача. Проте LCD дисплеї мають обмежену контрастність та кут огляду, для роботи з ними часто потрібен зовнішній резистор для регулювання яскравості, а також підключення через велику кількість пінів.

TFT LCD

TFT LCD – це різновид LCD-дисплеїв, який використовується в застосунках, де потрібен інтерактивний інтерфейс користувача. TFT-екрани також добре підходять для відображення анімацій та виведення графічних повідомлень.



Рисунок 2.15 – TFT LCD дисплей [8]

Зазвичай TFT-модулі постачаються у вигляді шилдів для Arduino або накладок (hats) для Raspberry Pi. Ці модулі забезпечують кращу передачу кольорів та оснащені сенсорною панеллю, проте мають вищу складність підключення та більші габарити, також споживають більше енергії, що не завжди сумісно з проектами на базі енергообмежених мікроконтролерів.

OLED

OLED-дисплеї (органічні світлодіоди) дещо дорожчі за згадані вище варіанти. Однак вони чудово підходять для проектів, де важливе енергозбереження.



Рисунок 2.16 – OLED-дисплей [8]

Ці дисплеї дуже прості у використанні, оскільки працюють за протоколом I2C. Зазвичай вони функціонують при напрузі 3,3–6,0 В, що робить їх дуже енергоефективними. Подібно до LCD-дисплеїв, на OLED-екранах можна створювати цікаві анімації. Крім того дисплеї мають більш сучасну архітектуру, оскільки кожен піксель в ньому є самосвітним, що дозволяє досягати високої

контрастності, широких кутів огляду та кращої видимості навіть при сонячному освітленні.

2.4.3 Вибір дисплею

У контексті розробки системи визначення нот на основі частоти звукових коливань було прийнято рішення на користь OLED-дисплею.



Рисунок 2.17 – OLED дисплей 0.96" I2C 128x64 [18]

Таблиця 2.6 – Характеристики дисплею

Розмір екрану, "	0.96
Тип дисплея	OLED
Роз'єм підключення	4-pin (VCC/GND/SCL/SDA)
Роздільна здатність дисплея	128 × 64
Розміри, мм	25 × 27 × 4

Таблиця 2.7 – Специфікації дисплею

Драйвер OLED модуля	SSD1306
Кут огляду, градуси	>160
Напруга живлення, В	3.3
Рівні входних сигналів, В	3.3
Цоколювка роз'єму	GND: Загальний VCC: Напруга живлення SDA: Шина даних SCL: Шина тактування даних

Це маленький дисплей, лише близько 1 дюйма по діагоналі, але дуже добре читаємий завдяки високому контрасту. Дисплей складається зі 128×64 окремих OLED-пікселів, кожен з яких вмикається або вимикається контролерним чіпом.

Драйверний чіп SSD1306 може передавати дані кількома способами, зокрема через I2C, SPI та 8-бітний паралельний інтерфейс.

Таким чином, вибір OLED-дисплею є доцільним через його компактним розміром, низьким енергоспоживанням, високою чіткістю виведення текстової та графічної інформації, а також простотою підключення через інтерфейс I2C.

Висновки до розділу 2

У другому розділі було здійснено проектування системи визначення музичних нот, з обґрунтуванням вибору основних апаратних компонентів та аналізом їх технічних характеристик. Визначено загальну концепцію системи, що базується на прийомі звукового сигналу, його цифровій обробці із застосуванням FFT та подальшому виведенні результату на дисплей.

Після порівняння характеристик і специфікацій Arduino Nano, Uno, Mega, було обрано Arduino Nano як оптимальний варіант для проєкту. Він забезпечує всі необхідні апаратні ресурси та його компактним та зручним для розміщення в системі з обмеженим простором.

Серед мікрофонів було розглянуто декілька типів, включаючи аналогові та цифрові. У підсумку для реалізації системи обрано аналоговий модуль MAX9814, який має вбудований підсилювач, автоматичне регулювання підсилення і високу чутливість. Це забезпечує надійну фіксацію звуків у широкій діапазоні гучностей, що є важливим для точного визначення частоти звуку.

Для виведення результатів обрано OLED-дисплей з драйвером SSD1306, який відзначається високим контрастом та низьким енергоспоживанням.

Таким чином, було сформовано технічну основу для створення прототипу системи, здатної визначати музичні ноти в реальному часі. Обрані компоненти забезпечать необхідний баланс між функціональністю, точністю та простотою реалізації.

3 РОЗРОБКА АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ВИЗНАЧЕННЯ НОТ

У попередньому розділі було обґрунтовано вибір ключових компонентів: Arduino Nano, аналогового модуля мікрофона MAX9814, OLED-дисплея на базі контролера SSD1306, а також було вирішено додати електролітичного конденсатора JCCON 100 мкФ 250 В. Далі докладно розглянуто, функціональну схему роботи системи, схему підключення цих елементів між собою та тестування системи.

3.1 З'єднання компонентів пристрою

Функціональна схема роботи системи представлена на рис. 3.1, на якому зображено принцип роботи системи визначення нот, що відображає логічну структуру пристрою, спрощуючи розуміння взаємодії між апаратними компонентами та покрокової обробки сигналу в системі.

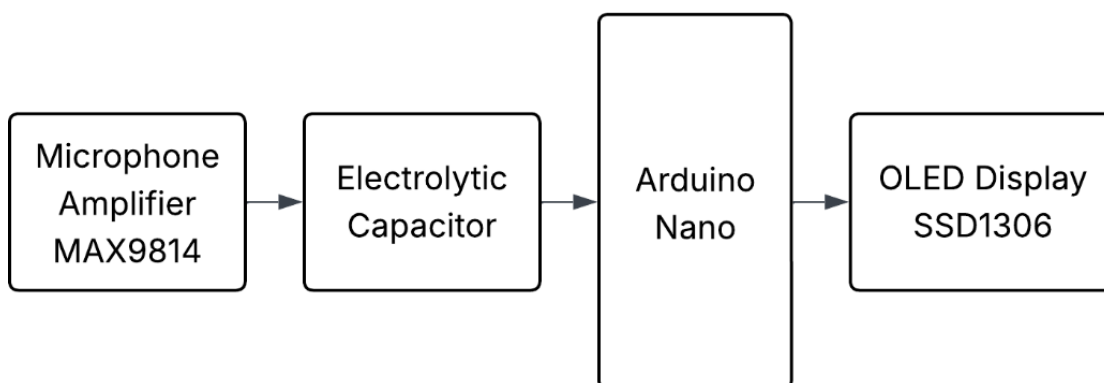


Рисунок 3.1 – Функціональна схема принципу роботи системи визначення нот

Система розпочинає роботу з приймання аналогового аудіо сигналу через підсилювач мікрофона MAX9814, який забезпечує автоматичне регулювання підсилення залежно від гучності звуку. Далі сигнал проходить через електролітичний конденсатор, що виконує функцію фільтрації та стабілізації. Оброблений сигнал надходить до мікроконтролера Arduino Nano, який оцифровує дані, виконує частотний аналіз сигналу (наприклад, за допомогою FFT) і визначає відповідну музичну ноту. Отриманий результат передається на OLED-дисплей SSD1306, де виводиться назва ноти або графік спектру сигналу.

Для створення функціональної схеми використовувався онлайн-сервіс Lucidchart, який дозволяє зручно проєктувати технічні діаграми, блок-схеми та візуалізації систем. Функціональна схема слугує базою для подальшої реалізації як апаратної, так і програмної частини системи.

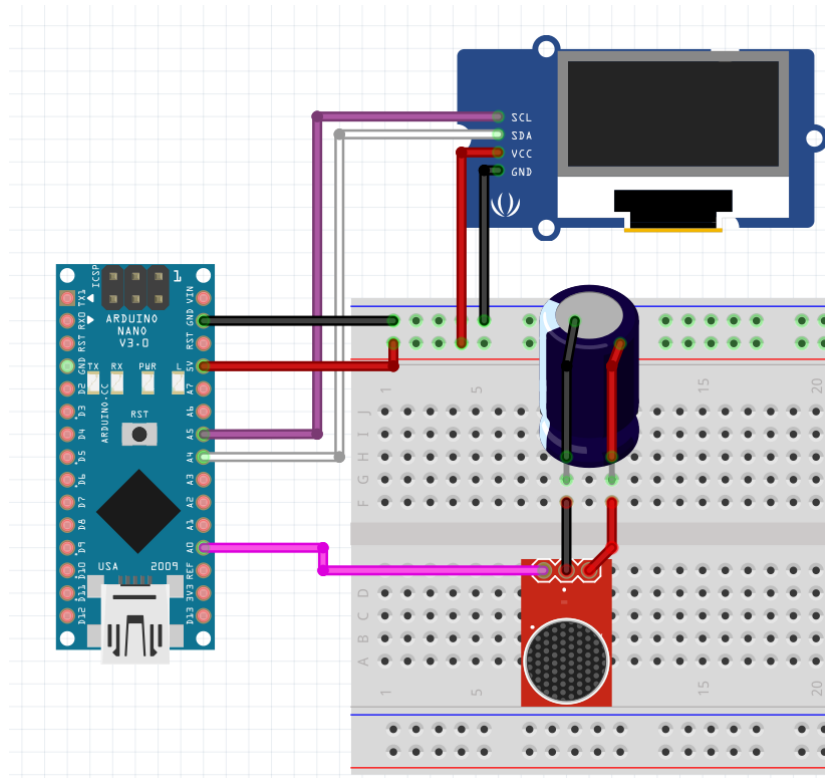


Рисунок 3.2 – Схема підключення пристрою

Схему підключення пристрою було розроблено в Fritzing – це безкоштовне програмне забезпечення, призначене для створення схем електронних пристроїв та макетів, що імітують з'єднання на макетній платі.

Однією з головних переваг Fritzing є можливість візуалізувати схему в режимі Breadboard View, де компоненти зображені у вигляді реалістичних 3D-елементів із чітко позначеними контактами та шлейфами.

Підключення дисплея OLED SSD1306

Дисплей SSD1306 реалізує протокол зв'язку I2C і має такі чотири виводи:

- 1) GND – земля;
- 2) VCC – живлення дисплея (3,3–5,0 В);
- 3) SCL – тактова лінія I2C;
- 4) SDA – лінія даних I2C.

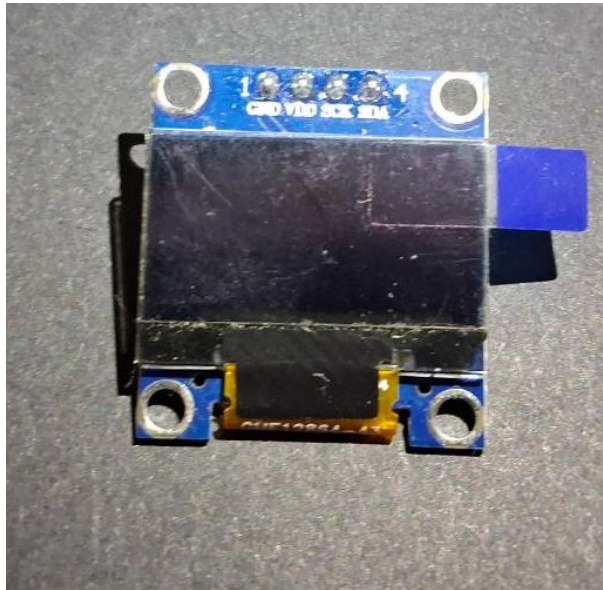


Рисунок 3.3 – Дисплей на чіпі SSD1306

Він підключається до Arduino Nano таким чином:

- 1) VCC – до виводу 5V на платі Arduino;
- 2) GND – до загальної землі (GND);
- 3) SCL – до аналогового порту A5;
- 4) SDA – до аналогового порту A4.

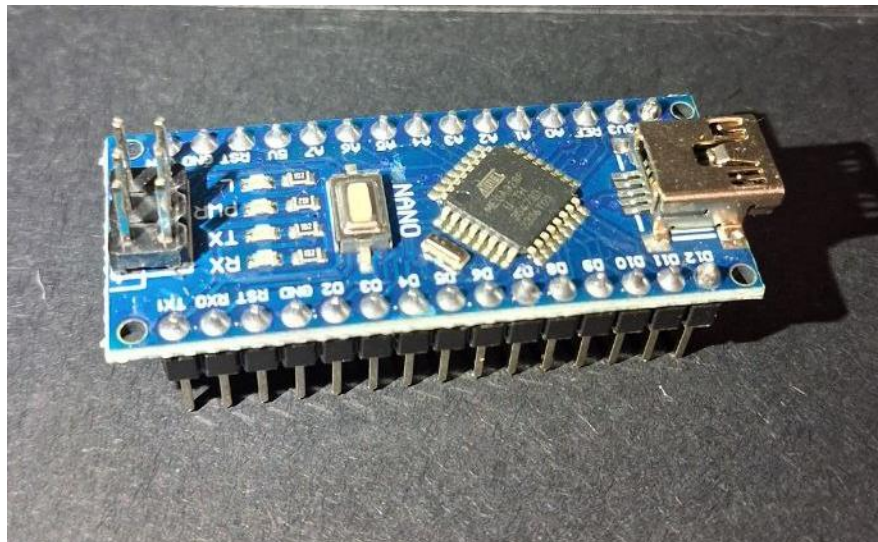


Рисунок 3.4 – Arduino Nano V3.0 AVR ATmega328P

Оскільки Arduino Nano має апаратну підтримку протоколу I2C через ці порти, підключення саме до A4 і A5 є стандартом для забезпечення правильної роботи з дисплеями цього типу.

Підключення мікрофонного модуля MAX9814

Модуль MAX9814 використовується для перетворення звукових хвиль у аналоговий електричний сигнал. Він має три основні виводи:

- 1) VCC – для живлення модуля (5 В);
- 2) GND – загальна земля;
- 3) OUT – аналоговий вихід, який передає посилений сигнал.



Рисунок 3.5 – Модуль електричного мікрофона MAX9814

У першочерговому порядку мікрофонний модуль підключається до мікроконтролера:

- 1) VCC – підключається до виводу 5 В на платі Arduino Nano;
- 2) GND – до одного з виводів GND мікроконтролера;
- 3) OUT – до аналогового входу A0.

Перед модулем MAX9814 у лінію живлення (VCC) додатково встановлено електролітичний конденсатор 100 мкФ.



Рисунок 3.6 – Конденсатор електролітичний JCCON 100 мкФ 250 В

Це покращує фільтрацію живлення, запобігає коливанням напруги та стабілізує роботу підсилювача, особливо у портативних або чутливих до шумів пристроях. Конденсатор підключається паралельно: його плюсовий контакт до VCC, мінусовий до GND.

Усі компоненти отримують живлення безпосередньо від лінії 5V на Arduino Nano, яка живиться через порт USB кабелем (рис. 3.7).



Рисунок 3.7 – Кабель mini-USB 300мм

Компоненти системи з'єднані у відповідності до їхніх електричних характеристик та вимог щодо інтерфейсів.

Підключення виконуються за допомогою Dupont-кабелів (рис. 3.8) з відповідними роз'ємами, які дозволяють швидко й без потреби пайки з'єднати компоненти на макетній платі (рис. 3.9).

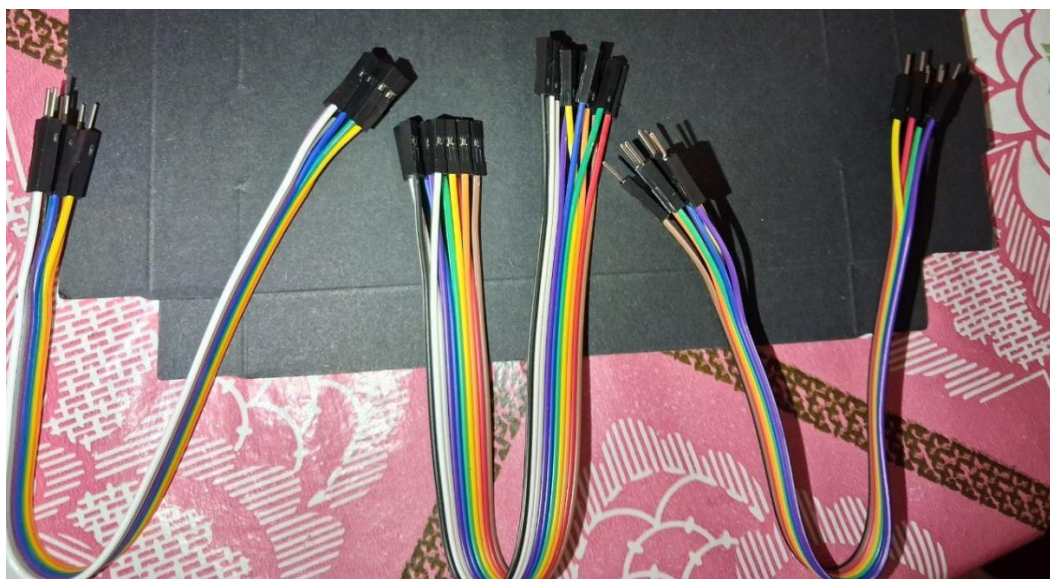


Рисунок 3.8 – Кабелі типу Dupont

Кабелі типу Dupont є одними з найбільш розповсюджених компонентів у прототипуванні електронних систем, особливо в сфері використання мікроконтролерів, сенсорів та модулів. Вони забезпечують швидке, надійне та багаторазове з'єднання між контактами компонентів без потреби пайки, що особливо важливо при створенні макетів і тестових збірок.

Dupont-кабелі класифікуються залежно від типу роз'ємів на обох кінцях:

- жіночий-жіночий (female-to-female): мають гнізда з обох боків, призначені для підключення до пінів (штирів) на платах або між модулями з зовнішніми контактами;
- чоловічий-чоловічий (male-to-male): оснащені штирями з обох кінців, використовуються для підключення макетних плат або в тих випадках, коли необхідно з'єднати два гніздові роз'єми;
- чоловічий-жіночий (male-to-female): універсальний тип, який дозволяє поєднувати плату з модулем, макетну плату з піновими роз'ємами тощо.

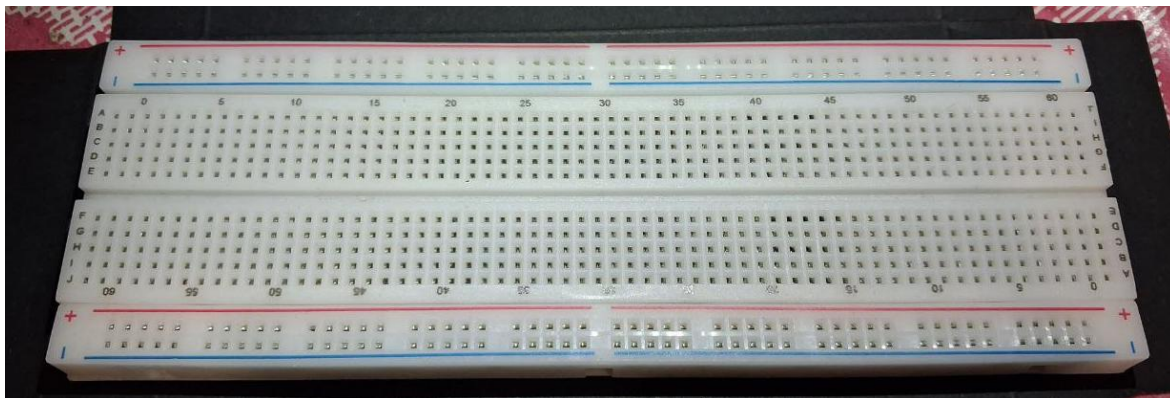


Рисунок 3.9 – Макетна плата

Макетна плата призначена для швидкого й зручного збирання електронних схем без потреби пайки. Вона використовується на етапі розробки для перевірки працездатності схеми, дозволяючи легко змінювати з'єднання та компонування елементів. Контакти на платі мають стандартну структуру з'єднань, що полегшує підключення до мікроконтролерів, датчиків, модулів живлення та інших компонентів.

На рисунку 3.10 представлено повністю змонтовану апаратну частину системи. Усі компоненти з'єднані відповідно до схеми, що дозволяє системі

стабільно працювати та точно визначати частоту звукового сигналу в реальному часі.

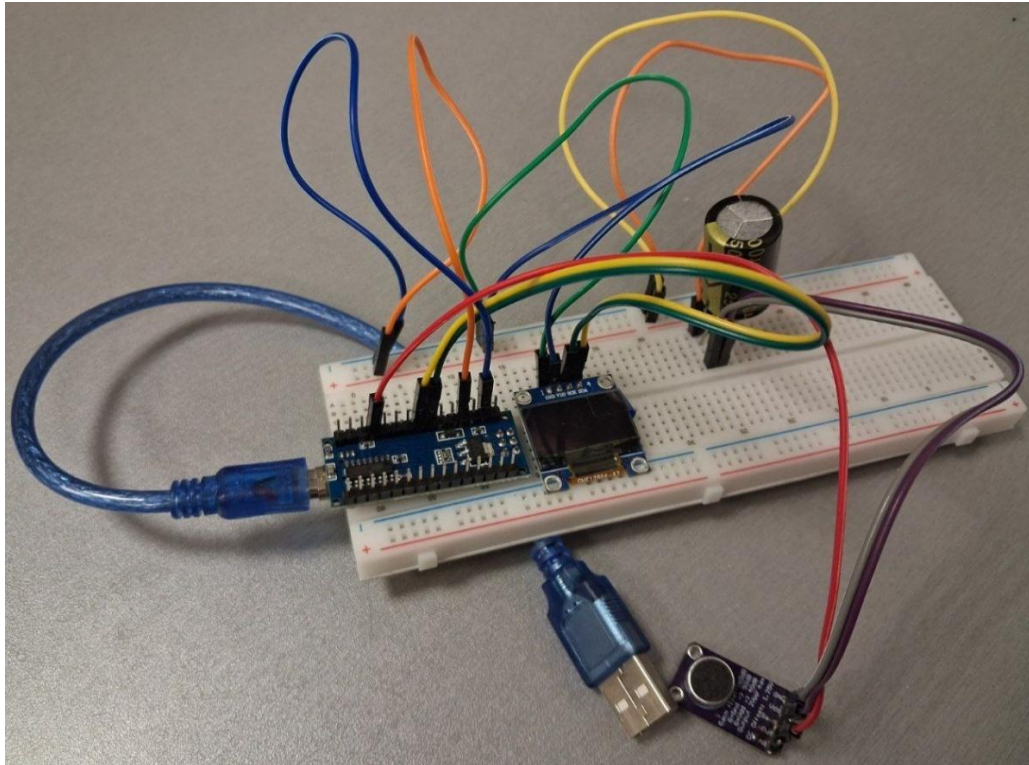


Рисунок 3.10 – Кінцевий результат

Для створення даної системи було використано недорогі, але ефективні компоненти, які забезпечать точність, стабільність і компактність. Усі елементи є загальнодоступними на ринку та легко інтегруються у єдину систему. У кошторис включено вартість основних компонентів та кабелів підключення.

Таблиця 3.1 – Кошторис обладнання для системи визначення нот

Найменування	Кількість	Од. вим.	Ціна, грн	Вартість, грн
Arduino Nano	1	шт.	189,00	189,00
Oled-дисплей SSD1306	1	шт.	124,00	124,00
Мікрофоний модуль MAX9814	1	шт.	106,00	106,00
Конденсатор електrolітичний JCCON 100мкФ	1	шт.	26,00	26,00
Кабель mini-USB	1	шт.	29,00	29,00
З'єднувальні проводи	1	набір	40,00	40,00
Макетна плата	1	шт.	45,00	45,00
			Разом	559,00

У результаті реалізації апаратної частини системи визначення нот було досягнуто надійного з'єднання між основними компонентами, мікроконтролером, дисплеєм та мікрофонним модулем з фільтрувальним конденсатором. Загальна схема з'єднання є логічна впорядкованою, зручною для налагодження і подальшого розширення системи.

3.2 Налаштування Arduino IDE

3.2.1 Мова програмування

Мова програмування Arduino базується на мові C/C++, однак її синтаксис спрощено для зручності початківців. Разом із базовими елементами мов C/C++, Arduino надає набір вбудованих функцій, які реалізують основні можливості мікроконтролера: керування цифровими та аналоговими входами/виходами, робота з таймерами, серійна комунікація, а також виконання математичних і логічних операцій.

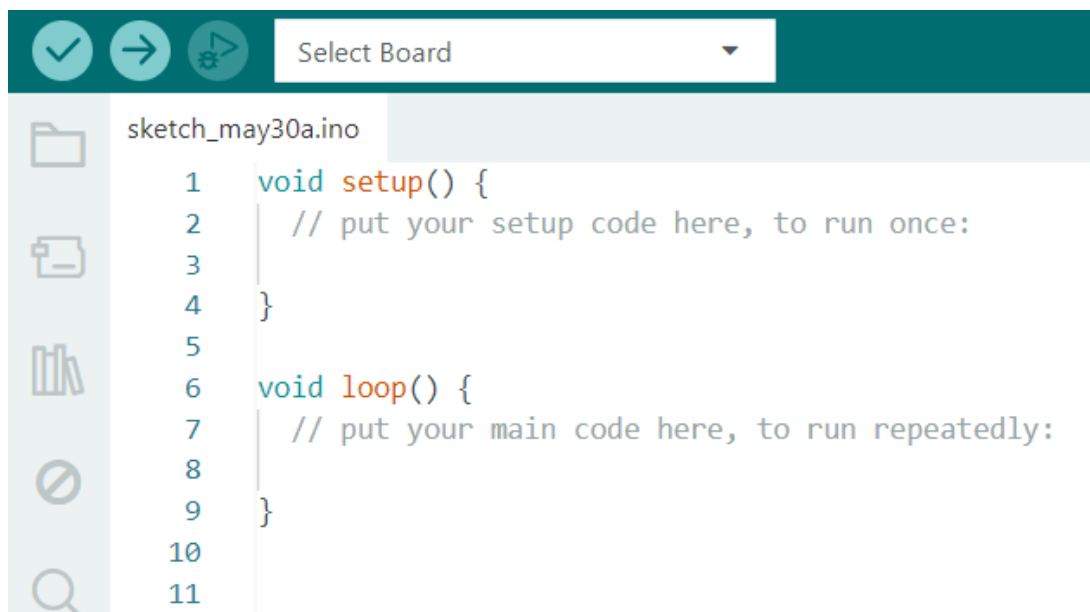


Рисунок 3.11 – Початковий скетч в Arduino IDE

Основу будь-якої програми (скетч) на платформі Arduino становлять дві ключові функції: *setup()* та *loop()*. Перша з них викликається один раз при ввімкненні або перезапуску мікроконтролера та призначена для первинної ініціалізації, зокрема налаштування пінів введення/виведення, підключення

бібліотек і периферійних пристроїв. Друга функція – виконується циклічно протягом усього часу роботи пристрою, забезпечуючи реалізацію основної логіки програми.

До категорії функції керування введенням/виведенням належать:

- *pinMode(pin, mode)*: встановлює режим роботи піну (вхід, вихід або вхід із внутрішнім підтягуванням);
- *digitalWrite(pin, value)*: задає логічний рівень на цифровому піні;
- *digitalRead(pin)*: зчитує логічний рівень із цифрового піна;
- *analogWrite(pin, value)*: здійснює широтно-імпульсну модуляцію;
- *analogRead(pin)*: зчитує аналогове значення з відповідного входу.

Крім того, Arduino надає засоби для роботи з часовими інтервалами:

- *delay(ms)*, *delayMicroseconds(us)*: реалізують затримку виконання програми на заданий інтервал;
- *millis()*, *micros()*: повертають кількість мілісекунд або мікросекунд від початку виконання програми відповідно.

Суттєвим інструментом для відлагодження й обміну даними є серійна комунікація через UART. Arduino забезпечує простий інтерфейс для її використання:

- *Serial.begin(baudRate)*: ініціалізує серійний порт із заданою швидкістю передавання даних;
- *Serial.print(data)*, *Serial.println(data)*: виводять дані у серійний монітор без або з переходом на новий рядок;
- *Serial.read()*, *Serial.available()*: забезпечують приймання даних із серійного порту.

Таким чином, стандартні функції Arduino надають зручний інтерфейс для програмування мікроконтролерів, приховуючи складні апаратні реалізації за простим синтаксисом.

3.2.2 Підключення плати Arduino Nano

Підключення мікроконтролерної плати Arduino Nano до комп'ютера здійснюється за допомогою стандартного USB-кабелю типу Mini-USB. Після фізичного під'єднання операційна система автоматично визначає пристрій та, за потреби, інсталує відповідні драйвери. Цей етап є необхідним для забезпечення стабільного зв'язку між комп'ютером та мікроконтролером з метою подальшого завантаження програмного коду, а також відлагодження роботи пристрою.

Після підключення мікроконтролера необхідно вказати тип використовуваної плати середовищі розробки Arduino IDE. Для цього у меню Tools -> Board -> Board Manager обираємо Arduino Nano.

Правильний вибір типу плати дозволяє компілятору IDE застосовувати відповідні параметри компіляції, зокрема налаштування тактової частоти, доступної пам'яті та конфігурації пінів вводу/виводу

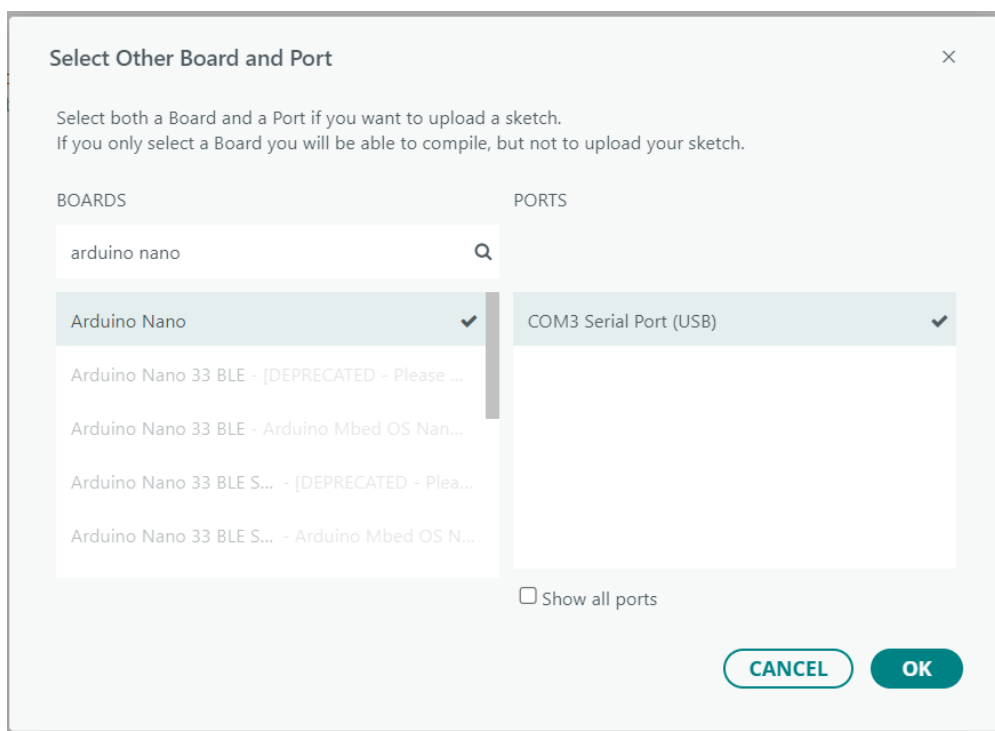


Рисунок 3.12 – Вибір типу плати та COM-порту

З метою успішного завантаження програмного коду на плату, а також забезпечення серійної комунікації, необхідно обрати правильний COM-порт, до якого підключена плата Arduino Nano.

3.2.3 Встановлення необхідних бібліотек та їх методи

Бібліотеки в середовищі Arduino IDE це попередньо написані модулі програмного коду, які розширюють базовий функціонал Arduino та спрощують взаємодію з периферійними пристроями, такими як дисплеї, сенсори чи мікросхемні модулі.

Застосування бібліотек дозволяє зменшити обсяг ручного кодування, підвищити надійність програмного забезпечення та забезпечити уніфікований підхід до роботи з різними апаратними компонентами.

Процес встановлення бібліотек складається з декількох етапів, для початку необхідно перейти до меню Sketch -> Include Library -> Manage Libraries (рис. 3.13).

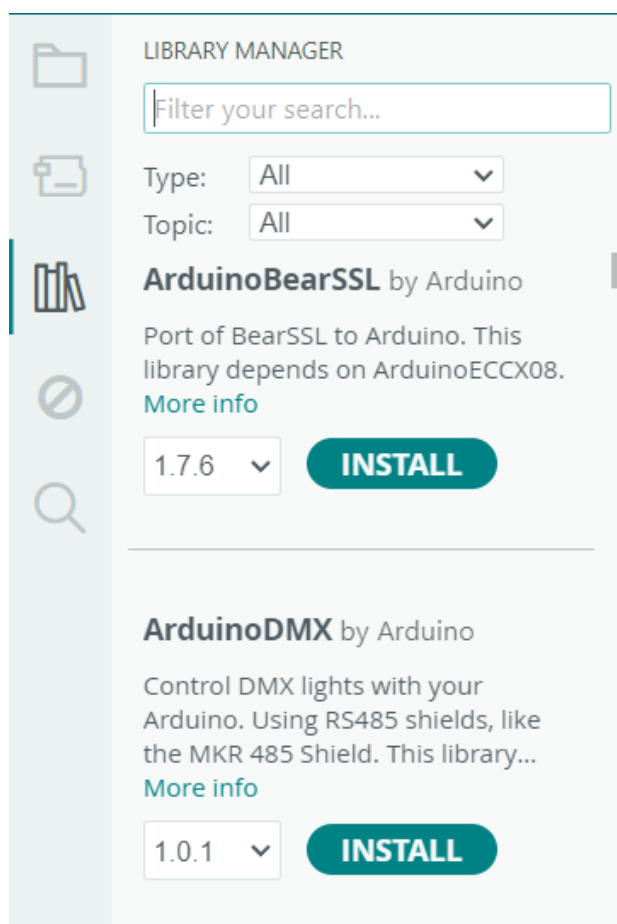


Рисунок 3.13 – Менеджер бібліотек

Наступним кроком у вікні пошуку ввести назву бібліотеки, далі обрати необхідну версію бібліотеки та натиснути кнопку «Install» поруч з назвою бібліотеки (рис. 3.14).

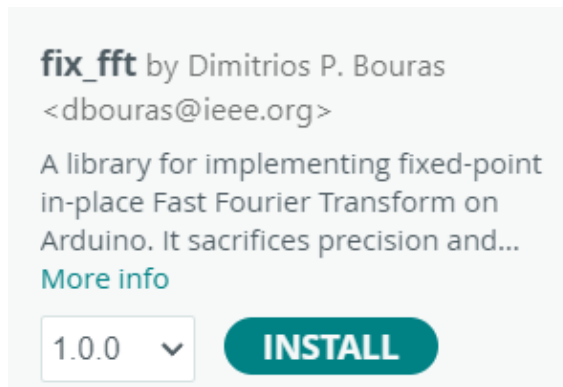


Рисунок 3.14 – Встановлення бібліотеки fix_fft

Встановлюємо всі інші необхідні бібліотеки:

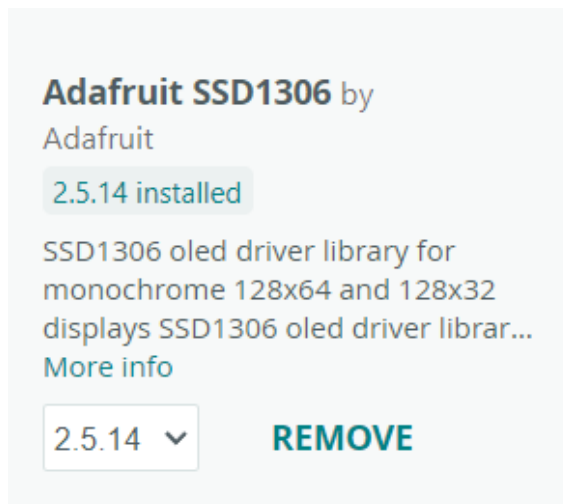


Рисунок 3.15 – Встановлення бібліотеки Adafruit SSD1306

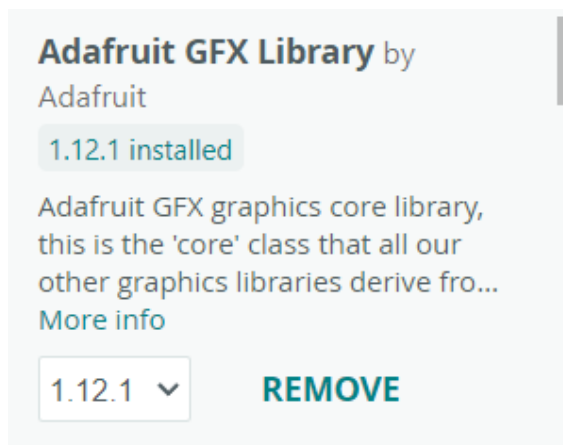


Рисунок 3.16 – Встановлення бібліотеки Adafruit GFX Library

У разі потреби Arduino IDE автоматично запропонує встановити всі залежні бібліотеки, що сильно спрощує етап інсталяції залежностей (рис. 3.17).

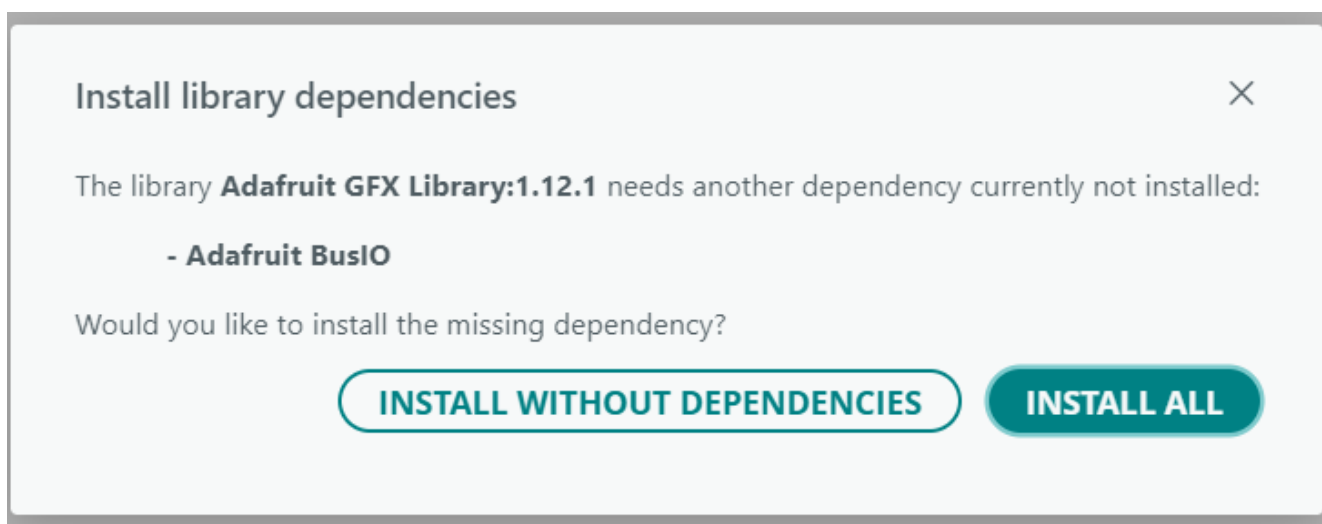


Рисунок 3.17 – Встановлення залежностей для бібліотеки Adafruit GFX Library

Після встановлення, бібліотеки необхідно підключити на початок скетчу Arduino за допомогою директив.

Наприклад:

```
#include "fix_fft.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Рисунок 3.18 – Підключення директив

У межах проєкту використовуються такі основні бібліотеки:

1) Wire – забезпечує роботу з протоколом I2C, який використовується для підключення OLED-дисплея до мікроконтролера. Основні методи бібліотеки Wire:

- *Wire.begin()*: ініціалізує I2C-шину.
- *Wire.beginTransmission(address)*: починає передачу даних до пристрою з заданою адресою.
- *Wire.write(data)*: записує байт (або масив байтів) до пристрою.
- *Wire.endTransmission()*: завершує передачу даних.
- *Wire.requestFrom(address, quantity)*: зчитує вказану кількість байтів з пристрою за I2C;
- *Wire.read()*: читає отриманий байт з шини;

2) Adafruit GFX Library – забезпечує базові засоби для виведення графіки на дисплеї, сумісні з драйвером SSD1306. Вона надає функції для малювання точок, ліній, прямокутників, кіл, а також для виведення тексту. Основні методи бібліотеки Adafruit GFX:

- *drawPixel(x, y, color)*: малює піксель у заданій точці.
- *drawLine(x0, y0, x1, y1, color)*: малює лінію між двома точками.
- *setCursor(x, y)*: встановлює позицію курсора для виведення тексту.
- *setTextSize(size)*: задає розмір тексту.
- *print() / println()*: виводить текст у поточну позицію курсора;

3) Adafruit SSD1306 – відповідає за безпосереднє керування OLED-дисплеєм із драйвером SSD1306 через інтерфейс I2C. Вона працює у зв'язці з бібліотекою GFX і реалізує специфічні функції ініціалізації та оновлення екрану. Основні методи бібліотеки Adafruit SSD1306:

- *begin(SSD1306_SWITCHCAPVCC, address)*: ініціалізує дисплей із вказаною I2C-адресою.
- *clearDisplay()*: очищає екран.
- *display()*: оновлює дисплей для відображення графічного буфера.
- *dim(true/false)*: зменшує або повертає яскравість дисплея.
- *invertDisplay(true/false)*: інвертує кольори дисплея;

4) *fix_fft* – реалізує алгоритм FFT з фіксованою комою, що дозволяє проводити спектральний аналіз аналогового сигналу в реальному часі, навіть на обмежених апаратних ресурсах. Основні методи бібліотеки *fix_fft*:

- *fix_fft(real[], imag [], log2point, inverse)*: основна функція для виконання перетворення Фур'є. Приймає масиви дійсних та уявних частин сигналу, логарифм кількості точок та прапорець напрямку перетворення (пряме/обернене);
- *fix_mpy(a, b)*: допоміжна функція множення з фіксованою точкою.

Завдяки використанню бібліотек вдалося досягти стабільної роботи пристрою в реальному часі навіть за умов обмежених апаратних ресурсів.

3.2.4 Завантаження скетчу та верифікація роботи

Після завершення налаштувань параметрів можна завантажити скетч до плати. Для цього необхідно натиснути кнопку «Upload» або скористатися комбінацією клавіш Ctrl + U.

Arduino IDE автоматично компілює програму, перетворюючи її у машинний код, перевіряє зв'язок із платою через інтерфейс UART і передає згенерований код на мікроконтролер за допомогою завантажувача (bootloader). Після завершення передавання мікроконтролер автоматично запускає виконання нової програми.

У разі успішного завантаження у вікні Arduino IDE з'являється повідомлення «Upload complete» і плата миттєво переходить до виконання скетчу.

Після завантаження бажано перевірити працездатність скетчу за допомогою серійного монітора (Serial Monitor) або шляхом спостереження за роботою периферійних пристроїв. В Arduino IDE серійний монітор відкривається через меню Tools -> Serial Monitor, або клавішею Ctrl + Shift + M.

3.3 Тестування системи

Тестування – це процес перевірки працездатності системи, програмного забезпечення або окремих компонентів із метою виявлення помилок, дефектів чи невідповідностей між фактичними та очікуваними результатами. Тестування є невід'ємною складовою життєвого циклу розробки будь-яких апаратних або програмних рішень, оскільки сприяє забезпеченню якості, надійності та відповідності кінцевого продукту встановленим технічним вимогам і стандартам.

У контексті створення системи, зокрема системи визначення музичних нот на основі акустичних коливань, тестування виконує низку ключових функцій:

- перевірка функціональної коректності кожного модуля: зокрема, тестується точність передачі аудіосигналу мікрофоном, правильність реалізації алгоритму обробки сигналу, а також достовірність виведення результатів на дисплей;

- оцінка взаємодії між апаратними та програмними компонентами системи: включаючи узгодженість роботи мікроконтролера, мікрофона, дисплея та програмного коду, який здійснює обробку даних і управління виводом;
- аналіз точності та продуктивності системи: зокрема, визначається, наскільки система точно розпізнає частоту звуку та відповідну їй музичну ноту, з якою затримкою відбувається реакція на вхідний сигнал, а також наскільки ефективно використовуються ресурси мікроконтролера;
- оцінка стабільності та надійності функціонування в різних експлуатаційних умовах: перевіряється робота системи за наявності фоновому шуму, при зміні гучності вхідного сигналу, у нестабільних умовах живлення або при тривалому безперервному функціонуванні.

Подальше тестування буде зосереджене на перевірці точності визначення нот у різних діапазонах частот і в умовах підвищеного шуму.

3.3.1 Тестування мікрофонного модуля MAX9814

Для перевірки працездатності мікрофонного модуля MAX9814 було здійснено тестове зчитування аналогових значень із вихідного контакту пристрою. Метою цього етапу є підтвердження наявності коректної реакції модуля на звукові коливання, а також перевірка динаміки сигналу, що надходить на аналоговий вхід мікроконтролера (рис. 3.19).

```
Тест мікрофона MAX9814...  
Vp-p: 66 | Напруга: 0.322 V  
Vp-p: 74 | Напруга: 0.361 V  
Vp-p: 64 | Напруга: 0.312 V  
Vp-p: 161 | Напруга: 0.786 V  
Vp-p: 182 | Напруга: 0.889 V  
Vp-p: 27 | Напруга: 0.132 V  
Vp-p: 368 | Напруга: 1.797 V  
Vp-p: 68 | Напруга: 0.332 V  
Vp-p: 136 | Напруга: 0.664 V  
Vp-p: 258 | Напруга: 1.260 V  
Vp-p: 257 | Напруга: 1.255 V  
Vp-p: 362 | Напруга: 1.768 V
```

Рисунок 3.19 – Результат тестування

Під час тестування Arduino Nano зчитує аналогові значення з мікрофона протягом короткого інтервалу часу (50 мс) і обчислює амплітуду сигналу за методом пік-пік (peak-to-peak). Отримані результати виводяться у вигляді числового значення амплітуди та еквівалентної напруги у вольтах через серійний монітор. Результати тестування наведено на рис. 3.20.

```
sketch_may30a.ino
1  #define MIC_PIN A0          // Пін, до якого підключено мікрофон
2  #define SAMPLE_WINDOW 50    // Вікно збору сигналу (мс)
3
4  void setup() {
5      Serial.begin(115200);
6      Serial.println("Тест мікрофона MAX9814...");
7  }
8
9  void loop() {
10     unsigned int signalMax = 0;
11     unsigned int signalMin = 1023;
12     unsigned int sample;
13
14     unsigned long startMillis = millis();
15     while (millis() - startMillis < SAMPLE_WINDOW) {
16         sample = analogRead(MIC_PIN);
17         if (sample < 1024) {
18             if (sample > signalMax) signalMax = sample;
19             if (sample < signalMin) signalMin = sample;
20         }
21     }
22
23     unsigned int peakToPeak = signalMax - signalMin;
24     float volts = (peakToPeak * 5.0) / 1024.0;
25
26     Serial.print("Vp-p: ");
27     Serial.print(peakToPeak);
28     Serial.print(" | Напруга: ");
29     Serial.print(volts, 3);
30     Serial.println(" V");
31
32     delay(200);
33 }
```

Рисунок 3.20 – Код тестування

У процесі експерименту мікрофонний модуль успішно зареєстрував зміну амплітуди сигналу в залежності від інтенсивності звуку, що підтверджує його працездатність та чутливість (рис. 3.19). Відповідно, модуль готовий до подальшої інтеграції у систему обробки звукових сигналів.

3.3.2 Тест OLED-дисплея

На етапі тестування дисплейного модуля з драйвером SSD1306 було перевірено працездатність виводу графічної та текстової інформації за допомогою бібліотеки Adafruit SSD1306, що реалізує підтримку OLED-дисплеїв із роздільною здатністю 128×64 пікселів через інтерфейс I2C (рис. 3.21, 3.22).

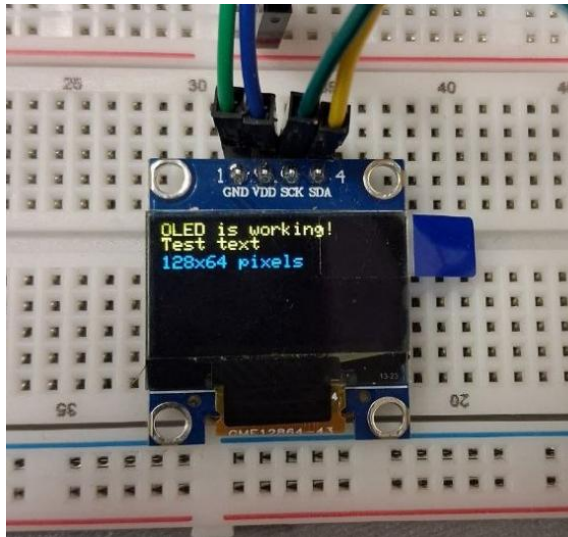


Рисунок 3.21 – Перевірка текстового заповнення

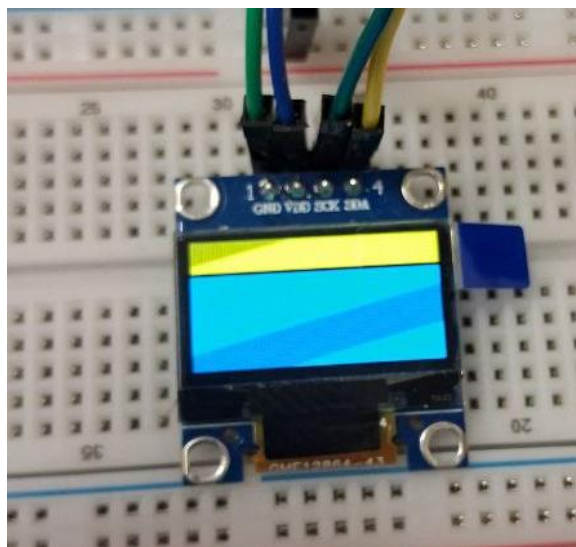


Рисунок 3.22 – Перевірка графічного заповнення

Метою тесту є перевірка ініціалізації пристрою, можливості виведення тексту різного розміру, графічних елементів та функцій очищення і оновлення дисплея.

```
1  #include <Wire.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_SSD1306.h>
4
5  #define SCREEN_WIDTH 128
6  #define SCREEN_HEIGHT 64
7  #define OLED_ADDR 0x3C // типова I2C-адреса для SSD1306
8
9  Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
10
11 void setup() {
12     Serial.begin(115200);
13
14     if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
15         Serial.println(F("OLED не знайдено! Перевір підключення."));
16         while (true); // зупинка, якщо дисплей не знайдено
17     }
18
19     display.clearDisplay();
20     display.setTextSize(1); // розмір тексту
21     display.setTextColor(SSD1306_WHITE); // білий текст
22     display.setCursor(0, 0); // верхній лівий кут
23     display.println(F("OLED is working!"));
24     display.println(F("Test text"));
25     display.println(F("128x64 пікселів"));
26     display.display(); // показати все на дисплеї
```

Рисунок 3.23 – Код тестування

```
27
28     delay(3000); // затримка на 3 секунди
29
30     // Намалювати прямокутник
31     display.clearDisplay();
32     display.drawRect(0, 0, 128, 64, SSD1306_WHITE);
33     display.display();
34     delay(2000);
35
36     // Намалювати коло
37     display.clearDisplay();
38     display.drawCircle(64, 32, 30, SSD1306_WHITE);
39     display.display();
40     delay(2000);
41
42     // Заповнити дисплей
43     display.clearDisplay();
44     display.fillRect(0, 0, 128, 64, SSD1306_WHITE);
45     display.display();
46     delay(1000);
47
48     // Очистити дисплей
49     display.clearDisplay();
50     display.display();
51 }
52
53 void loop() {
54 }
```

Рисунок 3.24 – Код тестування

За результатами тестування було встановлено, що дисплей коректно реагує на команди виводу, зображення формуються без спотворень, а графічна частина працює стабільно (рис. 3.21, 3.22). Це підтверджує успішну інтеграцію пристрою у систему.

3.3.3 Тест системи визначення нот

Після розробки програмної частини системи було проведено тестування її працездатності на практиці.

У процесі тестування виявлено ряд обмежень, пов'язаних як із апаратними, так і програмними чинниками.

По-перше, Arduino Nano V3, яка використовується в проєкті, має обмежений обсяг оперативної пам'яті (всього 2 кбайт SRAM). Це суттєво впливає на якість обробки сигналу, зокрема на можливість зберігання достатньої кількості семплів для обчислення FFT. Для досягнення компромісу між точністю та ресурсами мікроконтролера було обрано 128 семплів, що, хоча й дозволяє здійснювати базове визначення частоти, водночас обмежує роздільну здатність спектрального аналізу.

По-друге, бібліотека нот була обмежена лише кількома значеннями (C4-B4) через обмеження пам'яті. Це призводить до ситуації, коли частоти, близькі до цих нот, можуть бути відображені як найближча з доступних, навіть якщо насправді звучить інша нота.

Також виявлено значний вплив зовнішніх і внутрішніх шумів на точність вимірювання. Навколишній шум, механічні вібрації, а також електричні перешкоди від самої плати Arduino або дисплея OLED все це спотворює сигнал, ускладнюючи точне визначення частоти.

Ще однією проблемою стало те, що мікрофон MAX9814 не забезпечує високої точності в умовах сильного шуму або при роботі з людським голосом. Найкраще система реагує на чисті синусоїдальні сигнали з частотного генератора або інструментів з вираженою гармонічною структурою. Тому у випадках, коли підключався генератор звукових частот, система показувала частоту, яка приблизно відповідала заданому сигналу, з прийнятною похибкою.

Розглянемо ключові частини коду, що забезпечують функціонування системи в режимі тестування (повний код наведено у Додатку В).

```
#include <Arduino.h>
#include "fix_fft.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <avr/pgmspace.h>
#include <math.h>
```

Рисунок 3.25 – Підключення необхідних бібліотеки

```
const int N = 128; // Кількість точок дискретизації для БПФ
const int analogPin = A0; // Аналоговий пін для зчитування аудіосигналу
const int samplingFreq = 10000; // Частота дискретизації (10 кГц)
const unsigned long sampleInterval = 100; // Інтервал між зразками (в мікросекундах)
```

Рисунок 3.26 – Встановлення глобальні змінні

```
char vReal[N];
char vImag[N];
```

Рисунок 3.27 – Ініціалізація масивів

Масиви дійсної та уявної частин сигналу використовуються для виконання FFT. Значення зчитуються з аналогового входу, перетворюються та записуються у вигляді симетричного сигналу.

```
// OLED setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET -1
#define SSD1306_NO_SPLASH

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

Рисунок 3.28 – Ініціалізація дисплею

```
const char noteNames[][4] PROGMEM = {
  "C4", "D4", "E4", "F4", "G4", "A4", "B4"
};

const float noteFreqs[] PROGMEM = {
  261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88
};
```

Рисунок 3.29 – Бібліотека нот

```
const char* getNoteName(float freq) {  
    float minDiff = 1e6;  
    uint8_t closestIndex = 0;  
    for (uint8_t i = 0; i < noteCount; i++) {  
        float noteFreq = pgm_read_float(&noteFreqs[i]);  
        float diff = fabs(freq - noteFreq);  
        if (diff < minDiff) {  
            minDiff = diff;  
            closestIndex = i;  
        }  
    }  
  
    static char nameBuffer[4];  
    strcpy_P(nameBuffer, (PGM_P)noteNames[closestIndex]);  
    return nameBuffer;  
}
```

Рисунок 3.30 – Функція визначення ноти

Ця функція порівнює частоту, отриману після FFT, з табличними значеннями частот нот, і повертає назву найближчої ноти.

```
for (int i = 0; i < N; i++) {  
    while (micros() - lastSampleTime < sampleInterval);  
    lastSampleTime = micros();  
  
    int analogValue = analogRead(analogPin);  
    analogValue -= 512;  
    analogValue = constrain(analogValue, -127, 127);  
    vReal[i] = (char)analogValue;  
    vImag[i] = 0;  
}
```

Рисунок 3.31 – Зчитування сигналу з мікрофону

```
// FFT  
fix_fft(vReal, vImag, 7, 0);  
  
// Find peak  
int peakIndex = 0;  
int peakValue = 0;  
for (int i = 1; i < N / 2; i++) {  
    int mag = sqrt((int)vReal[i] * vReal[i] + (int)vImag[i] * vImag[i]);  
    if (mag > peakValue) {  
        peakValue = mag;  
        peakIndex = i;  
    }  
}
```

Рисунок 3.32 – Виконання FFT та визначення домінуючої частоти

Виконується пошук найбільшої амплітуди в спектрі для виявлення основної частоти.

```
if (peakValue < 4) {  
    Serial.println("No signal");  
    display.println("No signal");  
} else {  
    float frequency = (float)peakIndex * samplingFreq / N;  
    const char* noteName = getNoteName(frequency);  
  
    Serial.print("Frequency: ");  
    Serial.print(frequency, 1);  
    Serial.print(" Hz, Amplitude: ");  
    Serial.print(peakValue);  
    Serial.print(", Note: ");  
    Serial.println(noteName);  
  
    display.print("Freq: ");  
    display.print(frequency, 1);  
    display.println(" Hz");  
  
    display.print("Amp: ");  
    display.println(peakValue);  
  
    display.print("Note: ");  
    display.println(noteName);  
}
```

Рисунок 3.33 – Обробка та вивід результатів

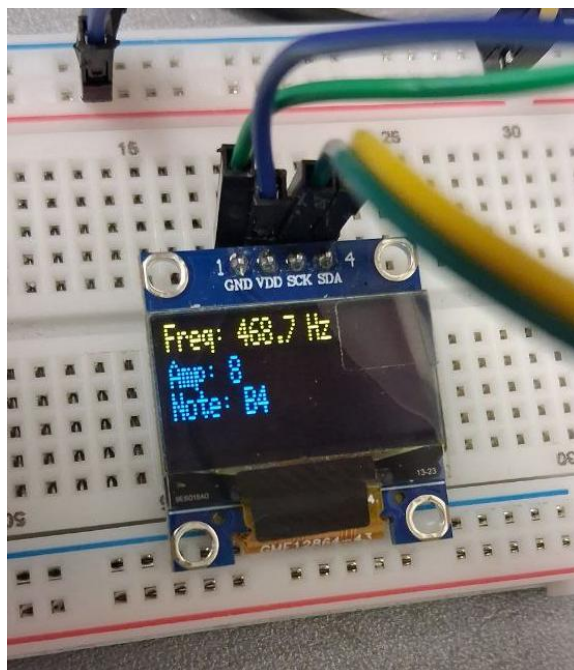


Рисунок 3.34 – Результат роботи системи

У загальному підсумку, незважаючи на технічні обмеження, система продемонструвала свою здатність розпізнавати частоти й визначати ноти, що є доказом працездатності реалізованого прототипу (рис. 3.34).

Таким чином, тестування є критично важливим етапом проєктування, який забезпечує відповідність системи вимогам до точності, ефективності та стійкості до зовнішніх чинників. Результати тестування також можуть використовуватись для подальшої оптимізації апаратної конфігурації або програмного забезпечення.

3.4 Методи покращення системи

У процесі реалізації системи визначення музичних нот було досягнуто основної мети створено працездатний прототип, здатний розпізнавати звукові частоти та відображати відповідну ноту на екрані. Проте, для розширення функціональності, підвищення точності та надійності, система може бути вдосконалена в декількох напрямках:

Для зменшення впливу перешкод та стабілізації вхідного сигналу доцільно встановити цифрові фільтри, зокрема фільтри низьких частот, середніх частот або фільтри Калмана. Це дозволить покращити точність визначення основної частоти.

Оптимізувати продуктивність можна шляхом видалення надлишкових обчислень, використання табличних значень замість складних формул і вдосконалення реалізації циклів.

Використання більш потужних мікроконтролерів, таких як ESP32, STM32 або Raspberry Pi, дозволяє працювати з більшою кількістю семплів, розширити бібліотеку нот та зменшити похибки завдяки значно більшій оперативній пам'яті й обчислювальним можливостям.

Реалізація графічного спектра аналізу у вигляді гістограми частот на дисплеї допоможе візуалізувати звукову інформацію в реальному часі.

Додаткові індикатори точності, наприклад кольорові підказки або анімації, зроблять процес налаштування музичних інструментів зручнішим та точнішим.

Запис частотних даних на SD-карту дозволить зберігати інформацію про частоту і час для подальшого аналізу або побудови графіків.

Інтеграція модулів Bluetooth або Wi-Fi забезпечить можливість передачі даних до смартфонів чи комп'ютерів для віддаленого моніторингу, аналізу або збереження статистики.

Модифікація алгоритмів обробки сигналів дозволить здійснювати аналіз складних звуків, що містять кілька нот одночасно, тобто акорди.

Перехід на автономне живлення із впровадженням енергозберігаючих режимів, таких як режим сну або низькочастотне сканування, забезпечить мобільність системи. Додатково доцільно розробити компактний і зручний корпус, що зробить пристрій придатним для використання під час репетицій, концертів або музичних занять.

Впровадження наведених удосконалень дозволить не лише усунути існуючі недоліки прототипу, але й перетворити його на повноцінний інструмент для точного визначення нот, з розширеною функціональністю, високою точністю та зручним інтерфейсом для кінцевого користувача.

Висновок до розділу 3

У третьому розділі було представлено практичну реалізацію системи визначення музичних нот на основі звукових коливань із використанням мікроконтролера Arduino Nano V3, мікрофонного модуля MAX9814 та OLED-дисплея. У процесі розробки було створено працездатний прототип, здатний розпізнавати частоти звукових сигналів та виводити відповідні ноти на екран у режимі реального часу.

Проведене тестування показало, що система загалом виконує поставлену задачу, однак має низку обмежень, пов'язаних із апаратною платформою.

Незважаючи на це, система адекватно реагує на генератори синусоїдальних сигналів і музичні інструменти з вираженими гармоніками, що підтверджує правильність вибраного підходу та архітектури.

У завершальній частині розділу запропоновано низку напрямів для подальшого покращення системи, серед яких: застосування точніших алгоритмів визначення частоти, впровадження фільтрації шуму, оптимізація програмного коду та перехід на продуктивніші мікроконтролери.

Таким чином, розділ демонструє практичну реалізацію, об'єктивно оцінює її обмеження та формує чітке підґрунтя для подальшого вдосконалення проєкту.

ВИСНОВКИ

У результаті виконання КБР було розроблено та реалізовано функціональну систему для визначення музичних нот на основі частотного аналізу звукових коливань із застосуванням алгоритму швидкого перетворення Фур'є. Система забезпечує можливість визначати домінуючі частоти вхідного звукового сигналу в режимі реального часу, зіставляючи їх із відповідними нотами музичної шкали.

Для досягнення поставленої мети було виконано наступні завдання:

- проведено аналіз існуючих систем розпізнавання звуку, їх переваг та недоліків;
- ознайомлено з принципами дії швидкого перетворення Фур'є та його використанням у цифровій обробці сигналів;
- досліджено та проаналізовано властивості звукових коливань;
- зібрано необхідне обладнання на основі Arduino, з використанням модуля електронного мікрофону та дисплея;
- реалізовано програмне забезпечення, яке оброблює цифровий сигнал з мікрофона, що передається з мікроконтролера, який у свою чергу здійснює обчислення та виводить результат на дисплей;
- проведено тестування створеної системи визначення нот, оцінено її точність, швидкість реагування та стабільність роботи за різних умов.

У ході дослідження було проаналізовано наявні аналогічні рішення, виявлено їх переваги та недоліки, що дало змогу сформулювати технічні та функціональні вимоги до розроблюваного пристрою. Важливу роль у досягненні поставленої мети відіграло обґрунтоване рішення щодо вибору апаратних компонентів: мікроконтролера Arduino Nano, модуля мікрофона MAX9814 із автоматичним посиленням сигналу та OLED-дисплея для візуалізації результатів. Ці компоненти було обрано з огляду на їхню доступність, сумісність між собою, компактність і відповідність вимогам до продуктивності.

Після реалізації апаратної частини системи було створено відповідне програмне забезпечення, що включає збір звукового сигналу, його цифрову

обробку за допомогою FFT, аналіз частотного спектра та інтерпретацію отриманих частот у вигляді музичних нот.

Система пройшла тестування у різних умовах, що дало змогу оцінити її точність, швидкодію та загальну стабільність. Отримані результати підтвердили працездатність розробленого рішення та його здатність до ефективного функціонування в реальному часі.

Практичне значення створеної системи полягає у можливості її широкого використання в освітньому процесі – зокрема, для навчання нотної грамоти, тренуванні музичного слуху, а також як наочного інструменту для демонстрації принципів цифрової обробки сигналів. Крім того, система може використовуватися для самонавчання музикантами, початківцями та студентами.

Результати дослідження повністю відповідають поставленій у вступі меті та завданням. Реалізована система є ефективною, функціонально завершеною та потенційно масштабованою. Вона демонструє інтеграцію апаратних та програмних рішень на основі відкритих технологій, що відповідає сучасним тенденціям у розробці доступних цифрових пристроїв.

У перспективі доцільним є розширення функціональних можливостей системи. Зокрема, її можна доповнити підтримкою:

- розпізнавання акордів і гармонічного контексту;
- аналізу поліфонічних звукових сигналів;
- збереження історії визначених нот на карту пам'яті;
- передачі даних по бездротових каналах зв'язку.

Таким чином, створена система має реальний потенціал подальшого вдосконалення та адаптації до різноманітних сфер застосування: від музичної освіти до інженерних і наукових досліджень у галузі обробки звукових сигналів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Всеспрямований мікрофонний модуль INMP441. *beegreen*. URL: <https://beegreen.com.ua/vsespryamovaniy-mikrofonnij-modul-inmp441-i2c-z-nizkim-energospozhivannyam-esp32-17822> (дата звернення: 25.05.2025).
2. Модуль електретного мікрофона MAX9814. *Arduino в Україні*. URL: <https://arduino.ua/prod3912-modyl-mikrofona-max9814> (дата звернення: 25.05.2025).
3. Петров А. Р., Обухова К. О. Система визначення нот на основі частоти звукових коливань. *Free and Open Source Software* : тези доп. XVI Міжнар. наук. - практ. конф. Харків, 13-14 лют. 2025 р. Харків : ХНЕУ ім. Семе́на Кузне́ця, 2025. С.173–174. URL: <https://repository.hneu.edu.ua/bitstream/123456789/35624/1/foss-2025-theses.pdf> (дата звернення: 25.05.2025).
4. Що таке музичні ноти: основи музичної грамоти. *sentire-voice.com.ua*. URL: <https://sentire-voice.com.ua/chto-takoe-muzykalnye-noty/> (дата звернення: 25.05.2025).
5. Ahmad M. S. Deep learning 101: lesson 23: the basics of audio signal processing with FFT. *Medium*. URL: <https://muneebsa.medium.com/deep-learning-101-lesson-23-the-basics-of-audio-signal-processing-with-fft-ffef65689c1d> (Last accessed: 10.05.2025).
6. Arduino uno vs nano vs mega, pinout, and technical specifications. *Electronic Clinic*. URL: <https://www.electronicclinic.com/arduino-uno-vs-nano-vs-mega-pinout-and-technical-specifications/#:~:text=Arduino%20Uno%20Vs%20Nano%20Vs%20Mega-In%20this> (Last accessed: 25.05.2025).
7. Pitch Detection of Speech Signal Using Autocorrelation Method in MATLAB. *MATLAB Helper*. URL: <https://matlabhelper.com/blog/matlab/pitch-detection-of-speech-signal-using-autocorrelation-method-in-matlab/> (Last accessed: 10.05.2025).

8. Display modules for your electronics projects. *CircuitMix*. URL: <https://www.circuitmix.com/display-modules-for-your-electronics-projects/> (Last accessed: 25.05.2025).
9. Fast and efficient pitch detection: bitstream autocorrelation. *Cycfi Research*. URL: <https://www.cycfi.com/2018/03/fast-and-efficient-pitch-detection-bitstream-autocorrelation/> (Last accessed: 10.05.2025).
10. Adafruit_SSD1306: Arduino library for SSD1306 monochrome 128x64 and 128x32 OLEDs. *GitHub*. URL: https://github.com/adafruit/Adafruit_SSD1306 (Last accessed: 10.05.2025).
11. Adafruit-GFX-Library: Adafruit GFX graphics core Arduino library, this is the 'core' class that all our other graphics libraries derive from. *GitHub*. URL: <https://github.com/adafruit/Adafruit-GFX-Library> (Last accessed: 10.05.2025).
12. fix_fft: fix_fft library. *GitHub*. URL: https://github.com/kosme/fix_fft (Last accessed: 10.05.2025).
13. Hernández M. G., Gómez J. T., Hourné E. A. P. Digital Spectrum Sensing Technique implemented on FPGA devices for Digital Television Applications. *Revista científica de ingeniería electrónica, automática y comunicaciones (RIELAC)*. 2018. Vol. XXXIX, No. 2. P. 10–23. ISSN: 1815-5928.
14. Interfaces in LCD display modules SPI, I2C, LVDS, MIPI, VX1, EDP, etc. *Riverdi*. URL: <https://riverdi.com/blog/interfaces-in-lcd-display-modules-spi-i2c-lvds-mipi-vx1-edp-and-others> (Last accessed: 25.05.2025).
15. KY-038 sound detection sensor module. *All Top Notch*. URL: <https://alltopnotch.co.uk/product/ky-038-sound-detection-sensor-module/> (Last accessed: 25.05.2025).
16. MAX4466 electret microphone pre amplifier module adjustable gain 2.4-5V DC. *rarecomponents*. URL: <https://rarecomponents.com/store/max4466-m> (Last accessed: 25.05.2025).
17. Nano. *Arduino*. URL: <https://docs.arduino.cc/hardware/nano/> (Last accessed: 25.05.2025).

18. OLED дисплей 0.96" I2C 128x64 (жовто-синій). *Arduino в Україні*. URL: <https://arduino.ua/prod1178-oled-displei-0-96-i2c-128x64-jelto-sinii> (дата звернення: 25.05.2025).
19. Senocak A., Ryu H., Kim J., Oh T., Pfister H., Chung J. S. Toward interactive sound source localization: better align sight and sound! *IEEE transactions on pattern analysis and machine intelligence*. 2025. Is. 5. P. 1–17. DOI: 10.1109/tpami.2025.3573994.
20. Shi Y. Image-based efficient music note detection by advanced lightweighted YOLO. *Applied and computational engineering*. 2024. Vol. 110, No. 1. P. 91–95. DOI: 10.54254/2755-2721/110/2024melb0054.
21. Sinin A. E., Hamdan S., Mohamad Said K. A., Sawawi M., Kamarudin K. A. D., Musib A. F. Malay rebab: Sound analysis of the Kelantan traditional musical instrument. *BioResources*. 2024. Vol. 20, No. 1. P. 1318–1332. DOI: 10.15376/biores.20.1.1318-1332.
22. Software. *Arduino – Home*. URL: <https://www.arduino.cc/en/software/> (Last accessed: 10.05.2025).
23. Sound and music. *How Music Works*. URL: <https://www.howmusicworks.org/109/Sound-and-Music/The-Chromatic-Scale> (Last accessed: 25.05.2025).
24. Sound wave definition, characteristics, and use in acoustics. *SVANTEK – Sound and Vibration*. URL: <https://svantek.com/academy/sound-wave/> (Last accessed: 11.05.2025).
25. SPH0645 I2S MEMS microphone breakout module. *Amazon*. URL: <https://www.amazon.co.uk/RAKSTORE-SPH0645-Microphone-Breakout-SPH0645LM4H/dp/B0C1N16SK2> (Last accessed: 25.05.2025).
26. Stafura A., Nagy S., Poničan J., Pavlík Ľ. Sound frequency spectrum versus legth of closed organ pipes. *Akustika*. 2025. Vol. 48, No. 48. DOI: 10.36336/akustika20254821.
27. Stanfill M., Morris J. W., Sterne J., Razlogova E., Murray S. When, where, and how is digital sound? *The 19th Annual Conference of the Association of Internet* 2025 p.

Researchers. Montréal, Canada, Oct. 10–13 2018. P. 1–13.
DOI: 10.5210/spir.v2018i0.10468.

28. Tang G. A digital signal processing-based multi-channel acoustic emission acquisition system with a simplified analog front-end. *Sensors*. 2025. Vol. 25, No. 10. P. 1–23. DOI: 10.3390/s25103206.

29. STAT 510 | Applied Time Series Analysis. *PennState: Statistics Online Courses*. URL: <https://online.stat.psu.edu/stat510/lesson/6/6.1> (Last accessed: 10.05.2025).

30. UART vs SPI vs I2C: Interface Comparison. *RF Wireless World*. URL: <https://www.rfwireless-world.com/terminology/data-communication/uart-vs-spi-vs-i2c-comparison#:~:text=This%20article%20compares%20UART,%20SPI,%20and%20I2C%20interfaces,,clocking,%20hardware%20and%20software%20complexity,%20advantages,%20and%20disadvantages> (Last accessed: 25.05.2025).

31. Yunitasari Y., Saifulloh S., Harly D. A. P. Application of FFT and KNN methods for the process of identifying sound signals. *IC-ITECHS*. 2024. Vol. 5, No. 1. P. 637–645. DOI: 10.32664/ic-itechs.v5i1.1599.

32. Yusong L., Wenjun N., Liuyang Y., Siyun H. Real-time spectral interferometry enables ultrafast acoustic detection. *Applied physics letters*. 2023. Vol. 123, No. 21. DOI: 10.1063/5.0178453.

ДОДАТОК А

Блок-схема алгоритму роботи системи

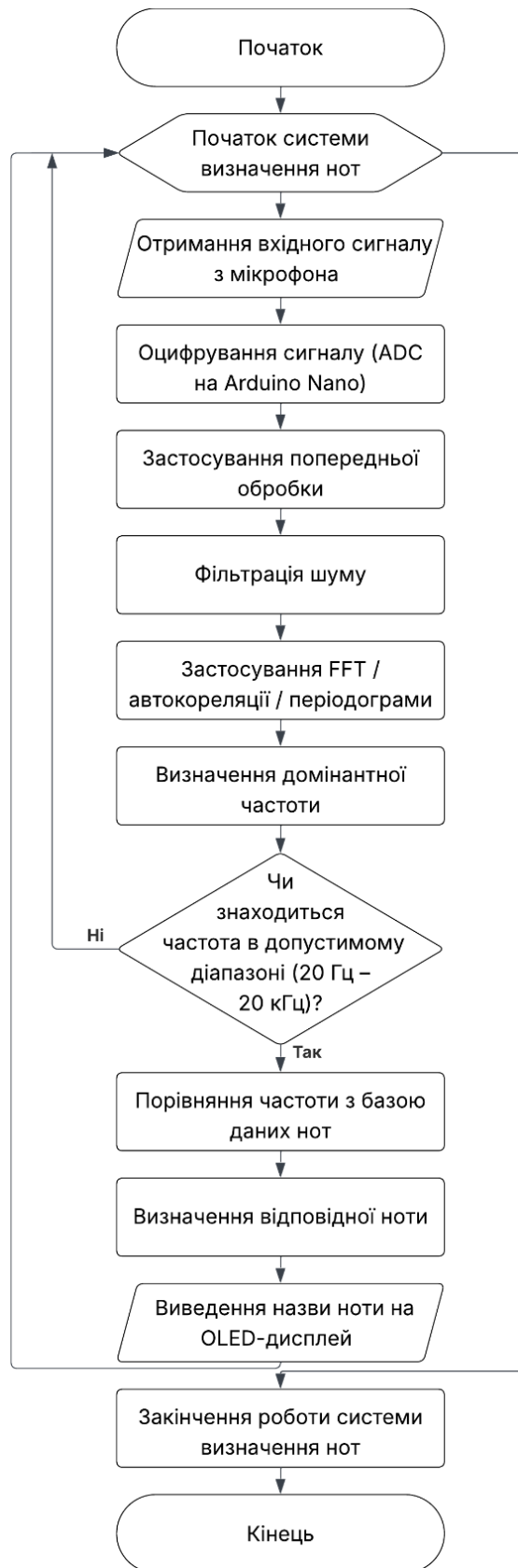


Рисунок А.1 – Блок-схема алгоритму роботи системи

ДОДАТОК Б

Код системи

```
#include <Arduino.h>
#include "fix_fft.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <avr/pgmspace.h>
#include <math.h>

const int N = 128;
const int analogPin = A0;
const int samplingFreq = 10000;
const unsigned long sampleInterval = 100; // 100  $\mu$ s = 10 kHz

char vReal[N];
char vImag[N];
unsigned long lastSampleTime = 0;

// OLED setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET -1
#define SSD1306_NO_SPLASH

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// Note names and frequencies (C4 to B4) in Flash
const char noteNames[][4] PROGMEM = {
  "C4", "D4", "E4", "F4", "G4", "A4", "B4"
};

const float noteFreqs[] PROGMEM = {
  261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88
};

const uint8_t noteCount = sizeof(noteFreqs) / sizeof(noteFreqs[0]);

// Get closest note name
const char* getNoteName(float freq) {
  float minDiff = 1e6;
  uint8_t closestIndex = 0;
  for (uint8_t i = 0; i < noteCount; i++) {
    float noteFreq = pgm_read_float(&noteFreqs[i]);
    float diff = fabs(freq - noteFreq);
    if (diff < minDiff) {
      minDiff = diff;
      closestIndex = i;
    }
  }
}
```

```
static char nameBuffer[4];
strcpy_P(nameBuffer, (PGM_P)noteNames[closestIndex]);
return nameBuffer;
}

void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("Starting frequency + note detection...");

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Display not found!"));
    while (true);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Ready to measure");
  display.display();
  delay(1000);
}

void loop() {
  // Sample input signal
  for (int i = 0; i < N; i++) {
    while (micros() - lastSampleTime < sampleInterval);
    lastSampleTime = micros();

    int analogValue = analogRead(analogPin);
    analogValue -= 512;
    analogValue = constrain(analogValue, -127, 127);
    vReal[i] = (char)analogValue;
    vImag[i] = 0;
  }

  // FFT
  fix_fft(vReal, vImag, 7, 0);

  // Find peak
  int peakIndex = 0;
  int peakValue = 0;
  for (int i = 1; i < N / 2; i++) {
    int mag = sqrt((int)vReal[i] * vReal[i] + (int)vImag[i] * vImag[i]);
    if (mag > peakValue) {
      peakValue = mag;
      peakIndex = i;
    }
  }

  display.clearDisplay();
  display.setCursor(0, 0);
```

```
if (peakValue < 4) {
    Serial.println("No signal");
    display.println("No signal");
} else {
    float frequency = (float)peakIndex * samplingFreq / N;
    const char* noteName = getNoteName(frequency);

    Serial.print("Frequency: ");
    Serial.print(frequency, 1);
    Serial.print(" Hz, Amplitude: ");
    Serial.print(peakValue);
    Serial.print(", Note: ");
    Serial.println(noteName);

    display.print("Freq: ");
    display.print(frequency, 1);
    display.println(" Hz");

    display.print("Amp: ");
    display.println(peakValue);

    display.print("Note: ");
    display.println(noteName);
}

display.display();
delay(100);
}
```

ДОДАТОК В

Матеріали апробації роботи



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

КАФЕДРА КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАТЕРІАЛИ

XVI-ої МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
«FREE AND OPEN SOURCE SOFTWARE»



Дякуємо за підтримку



13-14 лютого 2025 р.
м. Харків

Рисунок В.1 – Матеріали апробації роботи

СИСТЕМА ВИЗНАЧЕННЯ НОТ НА ОСНОВІ ЧАСТОТИ ЗВУКОВИХ КОЛИВАНЬ

Петров А. Р.

Керівник: Обухова К. О.

E-mail: miray7772@gmail.com

Миколаїв, Чорноморський національний університет імені Петра Могили

У світі музики та акустики визначення нот на основі частоти звукових коливань стало невід'ємною частиною сучасних технологій. Ця технологія широко використовується для автоматичного налаштування музичних інструментів, аналізу звукових записів, а також в навчальних програмах для музикантів. Вона дозволяє забезпечити високу точність та швидкість розпізнавання нот, що є надзвичайно важливим в умовах реального часу, наприклад, під час живих виступів або при використанні музичних інструментів.

З розвитком цифрових технологій та аудіообробки, застосування систем визначення нот стало доступним не лише професіоналам, але й аматорам та студентам музичних навчальних закладів. Тому дослідження цієї технології має велике практичне значення для музичної індустрії, а також для розвитку нових методів у музичному навчанні та аудіо обробці.

Метою цієї роботи є розробка апаратно-програмного комплексу для визначення нот на основі частоти звукових коливань, що дозволяє забезпечити точне і швидке розпізнавання музичних нот в реальному часі. Для цього необхідно дослідити та застосувати сучасні методи аналізу частотних характеристик звуку, вивчити основні технології для розробки таких систем та визначити апаратні компоненти, які будуть використовуватися при створенні комплексу.

Для реалізації системи планується використати швидке перетворення Фур'є (FFT) – алгоритм, який розкладає звуковий сигнал на частотні складові, що дозволяє визначити його частоту (рис. 1).

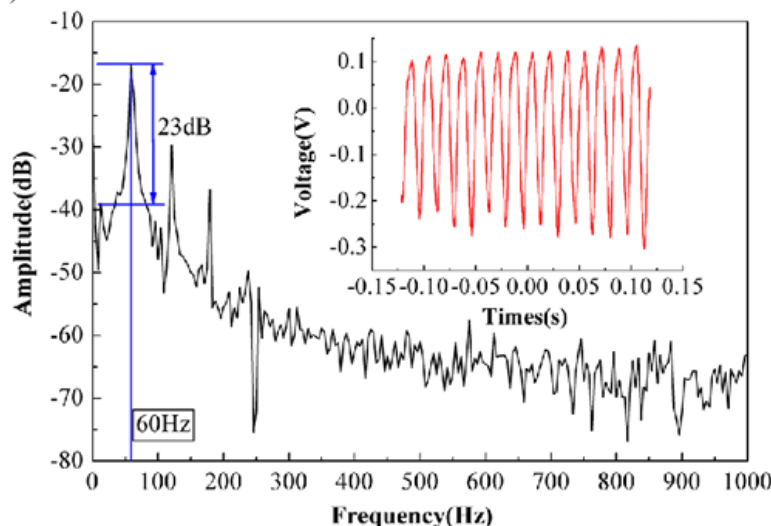


Рисунок 1 – Спектр у часовій та частотній області [1]

FFT використовується для перетворення часового аудіосигналу в частотне представлення. Математичною основою FFT є перетворення Фур'є, яке розкладає функцію часу (сигнал) на складові частоти [2]:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

де $X(f)$ – представлення в частотній області;
 f – частота;
 t – час;
 $e^{-j2\pi ft}$ – комплексна експоненціальна функція.

Алгоритм швидкого перетворення Фур'є ефективно обчислює це перетворення для дискретних сигналів, що робить його можливим для обробки навіть у застосунках реального часу. У процесі роботи системи також буде використано акустичний спектр, який являє собою сукупність частот звукових хвиль, аналіз яких допоможе досягти максимальної точності при розпізнаванні нот (рис. 2) [3].

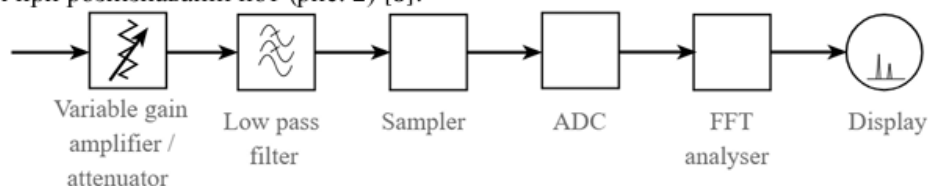


Рисунок 2 – Блок-схема аналізатора спектру FFT

Основою пристрою буде система розпізнавання звуку, яка вимірюватиме частоту звукових коливань і визначатиме відповідні музичні ноти. Основним елементом є мікрофонний модуль, що буде записувати звук та перетворювати його на електричний сигнал. Важливо, щоб мікрофон мав високу чутливість і мінімальні спотворення, оскільки точність визначення частоти залежить від якості вхідного сигналу. Для обробки звуку потрібен мікроконтролер – плата Arduino, що дозволить виконувати розрахунки для визначення частоти та ноти. Крім того, буде використано дисплей, для відображення нот, що відповідають частоті.

Алгоритм роботи системи визначення нот на основі частоти звукових коливань включає наступні кроки:

- мікрофонний модуль вловлює звукові коливання та перетворює їх на цифровий сигнал;
- цей сигнал обробляється плата Arduino, що реалізує алгоритм FFT, для виділення основної частоти;
- отримана частота порівнюється з базою даних музичних нот;
- на дисплеї виводиться точна нота, що відповідає цій частоті, разом з її значенням.

Використання Arduino, мікрофонного модуля, дисплея та алгоритму FFT у цій системі допоможе забезпечити розпізнавання звуків у режимі реального часу. Завдяки простоті та універсальності такий пристрій може знайти широке застосування. Система допоможе як музикантам, так і тим, хто просто цікавиться музикою. Вона стане в пригоді для поглиблення розуміння нотної системи та правильного налаштування музичних інструментів. Також система сприятиме вивченню основ музики та допоможе краще розрізняти ноти на слух.

Література

- [1] Liu Yu., Ni W., Yang L., Huang S., Liu H., Sun Yi., Xia R., Yao Y., Yan, L., Luo, Yi., Xu Zh., Xu G., Sun Q., Tang X., Shum P. Real-time spectral interferometry enables ultrafast acoustic detection. *Applied Physics Letters*, 2023. Vol. 123, Is. 21. DOI: 10.1063/5.0178453.
- [2] Ahmad M. Deep Learning 101: Lesson 23: The Basics of Audio Signal Processing with FFT. [Електронний ресурс] – Режим доступу до ресурсу: <https://muneebsa.medium.com/deep-learning-101-lesson-23-the-basics-of-audio-signal-processing-with-fft-ffef65689c1d#:~:text=In%20audio%20processing%2C%20FFT%20is,signal%20into%20its%20frequency%20components> (Last accessed: 10.02.2025).
- [3] Poole I. FFT Spectrum Analyzer. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.electronics-notes.com/articles/test-methods/spectrum-analyzer/fft-fast-fourier-transform-spectrum-analyser.php> (Last accessed: 10.02.2025).