

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
Обчислювальний пристрій аналізатора аудіо спектру
08-54.БКР.032.00.000 ПЗ

Виконав: студент 4 курсу, групи 2КІ—23мс
спеціальності

123 Комп'ютерна інженерія,
(шифр і назва напряму підготовки, спеціальності)
освітня програма
«Системне програмування»

Ярський Я.О.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ОТ
Савицька Л.А.
(прізвище та ініціали)

Рецензент: доц. каф. ПЗ
Хошаба О.М.
(прізвище та ініціали)

ДОПУЩЕНО

Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д
Азаров
“10” 06 2025 р.

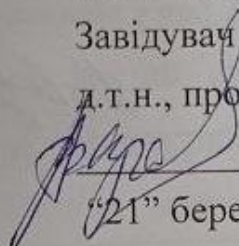
Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо—кваліфікаційний рівень бакалавр
Галузь знань – 12 – Інформаційні технології
Спеціальність – 123 – Комп'ютерна інженерія
Освітньо—професійна програма – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.


"21" березня 2025 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Ярському Ярославу Олеговичу

1 Тема роботи «Обчислювальний пристрій аналізатора аудіо спектру», керівник роботи к.т.н., доц. каф. ОТ Савицька Л.А. затверджені наказом ВНТУ від «20» березня 2025 року №70.

2 Термін подання студентом роботи: 13.05.2025

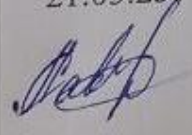
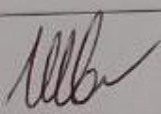
3 Вихідні дані до роботи: технічний опис мікропроцесорної системи спектрального аналізу аудіосигналу. Технології розробки: мова програмування C для Arduino, бібліотеки для роботи з серійним зв'язком, LCD—дисплеєм та швидким перетворенням Фур'є; мобільні сервіси, технології для інтеграції з мобільними застосунками. Цільова платформа системи – Arduino, середовище розробки – Arduino IDE.

4 Зміст розрахунково—пояснювальної записки: вступ; аналіз існуючих прототипів та визначення вимог до створюваної системи; обґрунтування структури та принципів роботи системи; розробка й тестування апаратно—програмної системи; висновки.

5 Перелік графічного матеріалу: функціональна схема пристрою, блок-схема роботи алгоритму. Ілюстративний матеріал — прототип мікропроцесорної системи.

6 Консультанти розділів наведені у таблиці 1

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1—3	к.т.н., доц. каф. ОТ Савицька Л.А. 	21.03.25 	13.05.25 
Нормоконтроль	ас. каф. ОТ Швець С.І. 	14.05.25	30.05.25

7. Дата видачі завдання 21.03.2025 р.

Календарний план наведений у таблиці 2.

Таблиця 2 — Календарний план

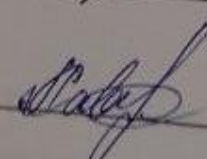
№ з/п	Назва етапів роботи	Строк виконання	Підп
1	Постановка задачі роботи	21.03.25	
2	Пошук матеріалів та інформаційних джерел	21.03.25— 30.03.25	
3	Аналіз існуючих прототипів та визначення вимог до створюваної системи	21.03.25— 30.03.25	
4	Обґрунтування структури та принципів роботи системи	31.03.25— 13.04.25	
5	Розробка й тестування апаратно—програмної системи	14.04.25— 27.04.25	
7	Оформлення пояснювальної записки та ілюстративного матеріалу	28.04.25— 11.05.25	
8	Перевірка якості виконання бакалаврської кваліфікаційної роботи та усунення недоліків	13.05.25	

Студент



Ярський Я.О.

Керівник роботи



Савицька Л.А.

АНОТАЦІЯ

УДК 621.3

Ярський Я.О. Обчислювальний пристрій аналізатора аудіо спектру. Бакалаврська кваліфікаційна робота зі спеціальності 123 «Комп'ютерна інженерія», освітня програма «Комп'ютерна інженерія». Вінниця: ВНТУ, 2025. 68 с.

На укр. мові. Бібліогр.: 41 назв; рис.: 26; табл. 1.

В роботі розроблено графічний аналізатор аудіоспектру на базі мікроконтролера Arduino, що дозволяє в режимі реального часу візуалізувати спектр аудіосигналів на графічному дисплеї. Розглянуто існуючі аналоги пристроїв, проаналізовано їхні переваги та недоліки, на основі чого обрано оптимальну апаратну конфігурацію. Реалізовано програмне забезпечення з використанням бібліотек для швидкого перетворення Фур'є (FFT) та графічного виводу.

У результаті створено функціональний прототип, який може застосовуватися в аудіосистемах, домашніх студіях або навчальних проектах для аналізу звукового спектру.

Ключові слова: Arduino, аудіоспектр, FFT, графічний дисплей, аналіз аудіосигналів, візуалізація.

ABSTRACT

UDC 621.3

Yarskyi Y.O. Computing device of the audio spectrum analyser. Bachelor's qualification project in speciality 123 'Computer Engineering', educational programme 'Computer Engineering'. Vinnytsia: VNTU, 2025. 68 c.

In Ukrainian. Bibliography: 41 titles; Figures: 26; Table 1.

The Arduino Graphical Audio Spectrum Analyser is a device that can analyse audio signals and visualise their spectrum on a graphical display. It can be useful in home studio equipment, audio systems, as well as in projects related to sound processing. In the project, a review of analogues was carried out, and the type of construction and all elements were selected taking into account their advantages and disadvantages.

In this work, a graphical audio spectrum analyser based on the Arduino microcontroller was developed, which allows real—time visualisation of the audio signal spectrum on a graphical display. The existing analogues of the devices are considered, their advantages and disadvantages are analysed, and the optimal hardware configuration is chosen. The software was implemented using libraries for fast Fourier transform (FFT) and graphical output.

As a result, a functional prototype was created that can be used in audio systems, home studios, or educational projects to analyse the sound spectrum.

Keywords: Arduino, audio spectrum, FFT, graphical display, audio signal analysis, visualisation.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Аналіз завдання та порівняльний огляд аналогів	6
1.2 Особливості мікропроцесорних платформ для аналізу аудіосигналів.....	9
1.3 Тенденції у цифровій обробці звуку та спектральному аналізі.....	11
1.3.1 Розвиток методів цифрового спектрального аналізу.....	11
1.3.2 Принципи та переваги швидкого перетворення Фур'є (FFT).....	13
1.3.3 Вагові вікна.....	15
1.3.4 Використання мікроконтролерів Arduino для обробки аудіосигналів.....	18
2 ПРОЄКТУВАННЯ МІКРОПРОЦЕСОРНОГО ПРИСТРОЮ АУДИОСПЕКТРАЛЬНОГО АНАЛІЗУ	20
2.1 Розробка структурної схеми	20
2.2 Вибір електронних компонентів для реалізації FFT—аналізатора та розробка функціональної схеми	20
2.3 Інтеграція LCD1602 для виведення спектральної інформації.....	24
2.3.1 Особливості роботи з LCD1602 через I2C	26
2.3.2 Формат і логіка візуалізації спектра на дисплеї	26
2.3.3 Переваги використання LCD1602 у мікропроцесорних пристроях	27
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	29
3.1 Огляд Arduino IDE як середовища розробки	29
3.2 Розробка алгоритму та програмного забезпечення.....	33
4 МОДЕЛЮВАННЯ ТА МАКЕТУВАННЯ РОБОТИ.....	36
4.1 Моделювання пристрою у Proteus.....	36
4.2 Тестування пристрою у середовищі Proteus.....	38

4.3 Складення електронної схеми на макетній платі.....	38
4.4 Тестування пристрою.....	40
4.5 Вибір компонентів та їх характеристика.....	48
4.6 Інструкція користувача.....	52
4.6.1 Увімкнення та режим роботи пристрою.....	52
4.6.2 Обслуговування, технічні зауваження та рекомендації.....	53
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А Технічне завдання.....	57
ДОДАТОК Б Протокол перевірки навчальної (кваліфікаційної) роботи.....	61
ДОДАТОК В Графічна частина.....	62
ДОДАТОК Г Ілюстративна частина.....	64
ДОДАТОК Д Лістинг програмного коду	66

ВСТУП

Сучасний розвиток мікроелектроніки та цифрових технологій значно розширює можливості створення доступних та функціональних пристроїв для збору, аналізу та візуалізації даних, зокрема у сфері аудіоінженерії. Однією з важливих задач у цій галузі є спектральний аналіз звуку, що дозволяє досліджувати структуру аудіосигналу за його частотними компонентами. Такий підхід широко застосовується в професійних студіях, системах моніторингу, медичних пристроях, а також у навчальному процесі для демонстрації принципів цифрової обробки сигналів.

На сьогодні існує велика кількість промислових спектроаналізаторів, проте більшість з них мають високу вартість, складні в експлуатації або вимагають спеціалізованого програмного забезпечення та знань для налаштування. Це обмежує їх доступність для аматорів, студентів технічних спеціальностей, а також користувачів побутових аудіосистем, які прагнуть не лише слухати, але й бачити звук у вигляді графічного спектра. У зв'язку з цим, актуальною є розробка простого та бюджетного пристрою, який здатен у режимі реального часу виконувати аналіз частотного спектру аудіосигналу з подальшою його візуалізацією на графічному дисплеї.

Актуальність теми дипломного проекту зумовлена зростанням інтересу до засобів цифрової обробки сигналів, поширенням платформ для швидкого прототипування, таких як Arduino, а також необхідністю у створенні наочних навчальних та побутових пристроїв. Реалізація спектроаналізатора на базі Arduino дає змогу поєднати простоту програмування, доступність апаратного забезпечення та широкі можливості для модифікацій і розширень функціоналу пристрою.

Метою даної роботи є розробка графічного аналізатора аудіоспектру на базі мікроконтролера Arduino, що забезпечує спектральний аналіз вхідного аудіосигналу за допомогою алгоритму швидкого перетворення Фур'є (FFT), а також вивід отриманого спектра на рідкокристалічний дисплей типу LCD1602. Пристрій має бути простим у використанні, енергоефективним, компактним та

економічно доступним для широкого кола користувачів.

У межах даного проєкту вирішуються такі основні **задачі**:

- проведення огляду сучасних методів спектрального аналізу та аналіз існуючих аналогічних пристроїв;
- обґрунтування вибору апаратної платформи для реалізації аналізатора.
- розробка структурної та принципової схеми пристрою.
- вибір необхідних електронних компонентів.
- розробка програмного забезпечення для збору аудіосигналу, обчислення спектра та графічного відображення результатів.
- виготовлення, налагодження та тестування прототипу пристрою в умовах реального середовища.

Об'єктом дослідження виступають процеси цифрової обробки сигналів у мікропроцесорних системах реального часу.

Предметом дослідження є методи спектрального аналізу аудіосигналів та технічні засоби реалізації цих методів на платформі Arduino.

Результатом виконаної роботи є функціональний пристрій — аналізатор аудіоспектру, який може бути використаний у побутових аудіосистемах, навчальних лабораторіях або аматорських технічних проєктах. Простота конструкції, мінімальна кількість компонентів та використання відкритого програмного забезпечення роблять даний проєкт привабливим для практичного використання, розширення або комерційної адаптації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз завдання та порівняльний огляд аналогів

У сучасному світі цифрова обробка аудіосигналів є важливою складовою як професійних, так і аматорських систем. Швидкий розвиток мікропроцесорної техніки дозволив створювати пристрої для аналізу спектру звуку на базі простих мікроконтролерів, доступних навіть початківцям. Серед основних сфер застосування таких пристроїв — музичні студії, системи візуалізації аудіо в реальному часі, діагностичні прилади, навчальні лабораторії та побутова електроніка.

Актуальність розробки мікропроцесорного спектроаналізатора зумовлена необхідністю недорогих, портативних, простих у реалізації засобів аналізу аудіосигналу. Більшість існуючих рішень або надто дорогі, або складні для розуміння та адаптації з боку користувача. З іншого боку, деякі дешеві пристрої мають занадто примітивну реалізацію (наприклад, лише світлодіодну індикацію), що не дозволяє отримати достатню точність та гнучкість.

Вибір теми зумовлений наступними факторами: широке поширення аудіопристроїв і потреба у візуалізації звукового сигналу; зацікавленість користувачів у проєктах, які поєднують роботу з мікроконтролерами, сенсорами, обробкою сигналу та відображенням інформації; доступність недорогих компонентів для реалізації функціоналу аналізатора спектру на базі Arduino та сумісних платформ; можливість практичного застосування проєкту в освітніх цілях, у звукотехніці та аматорських розробках.

Проаналізуємо кілька аналогічних пристроїв, які представлені на ринку, наприклад SparkFun Spectrum Shield.

Розширення до Arduino, що використовує два MSGEQ7 чіпи для розділення аудіосигналу на 7 частотних смуг.

Переваги:

- просте підключення до Arduino;
- наявність бібліотек для швидкого старту;

- обробка стереосигналу;

Недоліки:

- примітивна обробка спектру – фіксовані частотні смуги;

- відсутність можливості гнучкого налаштування або зміни алгоритму обробки;

- низька роздільна здатність спектру.

Другим пристроєм візьмемо Audio Spectrum Visualizer Kit на WS2812B (рисунок 1.1)



Рисунок 1.1 — Audio Spectrum Visualizer Kit

Популярний DIY — набір, що поєднує мікроконтролер (зазвичай ESP32 або Arduino), мікрофонний модуль та LED—стрічку.

Переваги:

- яскрава індикація спектру;

- багато варіантів дизайну;

- можливість підключення до Bluetooth або Wi-Fi.

Недоліки:

- візуалізація часто не відповідає реальній частотній структурі сигналу;

- обмежена точність вимірювань;

- орієнтований на декоративне застосування, а не вимірювальну точність.

Ще можна взяти трішкий інший за реалізацією пристрій. Наприклад miniDSP UMIK—1 з програмним спектроаналізатором (рисунок 1.2)



Рисунок 1.2 — miniDSP UMIK-1

Калібрований USB — мікрофон, який використовується з комп'ютерним ПЗ типу REW або Room EQ Wizard.

Переваги:

- висока точність спектрального аналізу;
- професійне використання для акустичних вимірювань;
- можливість калібрування.

Недоліки:

- висока вартість;
- не підходить для автономного або портативного використання.

Далі розглянемо DIY Fast Fourier Transform (FFT) аналізатори на Arduino Nano (рисунок 1.3)

Багато ентузіастів створюють прості аудіоаналізатори на основі Arduino Nano з використанням бібліотеки FFT (наприклад, `arduinoFFT`).

Переваги:

- відкритий код;

- гнучкість у налаштуванні;
- низька вартість.

Недоліки:

- обмежена продуктивність Arduino Nano для великих обсягів спектральної обробки;
- невелика кількість бінів FFT;
- залежність якості аналізу від правильного підбору фільтрації, вибірки тощо.



Рисунок 1.3 — DIY Fast Fourier Transform

Запропонований у даному проєкті пристрій буде базуватись на мікроконтролері Arduino (або його функціональному аналогу), використовуватиме AUX порт, бібліотеку FFT для аналізу сигналу та виводитиме результат на LCD-дисплей. Завдяки цьому досягаються наступні переваги:

- низька вартість;
- гнучкість;
- наочність;
- компактність.

Відкритий код дозволяє змінювати параметри обробки сигналу, використання доступних компонентів дозволяє знизити ціну порівняно з комерційними аналогами, спектр відображається в реальному часі на LCD-дисплеї, що дозволяє використовувати пристрій у навчальних цілях, вся система вміщується в невеликому корпусі.

Таким чином, розробка є актуальною, практично цінною, і має потенціал до подальшого вдосконалення. На відміну від аналогів, запропонований пристрій орієнтований не лише на індикацію, а на реальний спектральний аналіз звуку в доступній для користувача формі.

1.2 Особливості мікропроцесорних платформ для аналізу аудіосигналів

Для реалізації аналізу аудіосигналів у сучасних пристроях широко використовуються мікропроцесорні платформи, які забезпечують обробку цифрових даних у реальному часі. Вибір відповідної платформи визначає ефективність, точність та функціональні можливості аудіоспектроаналізатора.

Основні особливості мікропроцесорних платформ для аналізу аудіосигналів включають:

- потужність обробки і швидкодія;
- наявність периферії для роботи з аналоговими сигналами;
- обсяг оперативної пам'яті;
- інтерфейси для виводу інформації;
- енергоспоживання;
- програмна підтримка та доступність бібліотек;
- можливості розширення.

Для виконання алгоритмів перетворення Фур'є (FFT) та інших обчислювальних операцій необхідна достатня обчислювальна потужність. Мікроконтролери з високою тактовою частотою і апаратною підтримкою DSP— операцій значно пришвидшують обробку сигналу та забезпечують виведення спектру в реальному часі.

Важливою складовою є вбудований аналогово—цифровий перетворювач (АЦП), який здійснює оцифрування аудіосигналу. Висока розрядність і швидкість АЦП впливають на точність відтворення спектру та дозволяють аналізувати широкий діапазон частот.

Для збереження буферів аудіоданих, проміжних результатів та програмних алгоритмів необхідний достатній об'єм оперативної пам'яті. Недостатня пам'ять може призводити до уповільнення роботи або обмеження частоти дискретизації.

Для відображення спектру аудіосигналу часто використовуються різні дисплеї (LCD, OLED), а також можливість передачі даних на зовнішні пристрої через UART, SPI, I2C або USB. Обрана платформа має підтримувати необхідні інтерфейси для підключення візуальних та комунікаційних модулів.

У портативних пристроях важливо мінімізувати споживання енергії, тому мікроконтролери з низьким енергоспоживанням та можливістю роботи в режимах енергозбереження є перевагою.

Наявність готових бібліотек для реалізації FFT, роботи з АЦП і дисплеями значно спрощує розробку. Популярні платформи, такі як Arduino, мають розвинену екосистему, що дозволяє швидко створювати прототипи та тестувати алгоритми.

Підтримка додаткових модулів, датчиків і периферійних пристроїв дозволяє розширювати функціонал аналізатора, додаючи нові можливості, наприклад, бездротову передачу даних або інтеграцію з комп'ютерними системами.

Підсумовуючи, вибір мікропроцесорної платформи для аудіоспектроаналізатора повинен базуватися на балансі між продуктивністю, енергоспоживанням, наявністю периферії та простотою розробки. Найпоширенішими рішеннями є платформи на базі мікроконтролерів сімейств ARM Cortex—M, AVR (Arduino) та DSP—процесорів, які здатні забезпечити необхідну швидкість і точність аналізу аудіосигналів.

1.3 Тенденції у цифровій обробці звуку та спектральному аналізі

1.3.1 Розвиток методів цифрового спектрального аналізу

Цифровий спектральний аналіз є основою для обробки та дослідження сигналів у багатьох технічних та наукових галузях. Його розвиток тісно пов'язаний із прогресом у галузі цифрової обробки сигналів (ЦОС) та комп'ютерних технологій.

Перші методи спектрального аналізу базувалися на аналогових фільтрах, що мали обмежену точність і гнучкість. З появою цифрових обчислювальних пристроїв з'явилася можливість більш точної, швидкої і гнучкої обробки сигналів.

Основними етапами розвитку цифрового спектрального аналізу є перетворення Фур'є — ключовим математичним інструментом спектрального аналізу є перетворення Фур'є, що дозволяє розкласти сигнал на складові частоти. Спочатку для обчислень використовували дискретне перетворення Фур'є (DFT), яке є ресурсомістким і повільним.

Швидке перетворення Фур'є (FFT) — в 1965 році була запропонована ефективна алгоритмічна реалізація DFT — швидке перетворення Фур'є (FFT), що значно знизило обчислювальні витрати. FFT дозволило виконувати спектральний аналіз у реальному часі навіть на обмежених за ресурсами мікропроцесорних системах.

Віконні методи — для аналізу сигналів з часовою варіацією спектру були розроблені методи, що використовують віконні функції (наприклад, Хенінга, Хемінга). Віконні методи допомагають зменшити спотворення спектру, пов'язані з розривами сигналу.

Періодограми та спектральна оцінка — для оцінки спектру сигналів розроблено різні підходи, такі як класична періодограма, усереднені періодограми та методи гладкої спектральної оцінки, що підвищують точність аналізу та зменшують шум.

Мультиплікативні та адаптивні методи — сучасні методи спектрального аналізу включають адаптивні алгоритми, які можуть змінювати параметри під час обробки сигналу, що дозволяє ефективніше виявляти зміни спектру в часі.

Сьогодні цифровий спектральний аналіз є невід’ємною частиною систем аудіообробки, телекомунікацій, радіотехніки, медичної діагностики та багатьох інших сфер. Постійне вдосконалення алгоритмів і зростання обчислювальних можливостей забезпечують все більшу точність і швидкість аналізу, що відкриває нові можливості для розвитку інтелектуальних систем обробки сигналів. У данному дипломному проєкті будуть використовуватись такі методи як віконні та швидке перетворення Фур’є

1.3.2 Принципи та переваги швидкого перетворення Фур’є (FFT)

Перетворення Фур’є — це математичний апарат розкладання сигналів на синусоїдальні коливання. Наприклад, якщо сигнал $x(t)$ безперервний і нескінченний за часом, його можна у вигляді інтеграла Фур’є:

$$x(t) = \int_0^{\infty} X_{\omega} \cos(\omega t + \varphi_{\omega}) d\omega$$

Інтеграл Фур’є збирає сигнал $x(t)$ з нескінченної множини синусоїдальних складових різних частот ω , що мають амплітуди X_{ω} і фази φ_{ω} [6].

Насправді нас більше цікавить аналіз кінцевих за часом звуків. Оскільки музика перестав бути статичним сигналом, її спектр змінюється у часі. Тому при спектральному аналізі нас цікавлять окремі короткі фрагменти сигналу. Для аналізу таких фрагментів цифрового аудіосигналу існує дискретне перетворення Фур’є:

$$x(n) = \sum_{k=0}^{N/2} X_k \cos \frac{2\pi k(n + \varphi_k)}{N}$$

де N — відліки дискретного сигналу;

$x(n)$ — інтервал часу від 0 до $N - 1$.

Синтезуються як сума кінцевого числа синусоїдальних коливань з амплітудами X_k та фазами ϕ_k . Частоти цих синусоїд дорівнюють kF/N , де F — частота дискретизації сигналу, а N — число відліків вихідного сигналу $x(n)$ на аналізованому інтервалі. Набір коефіцієнтів X_k називається амплітудним спектром сигналу. Як видно з формули, частоти синусоїд, на які розкладається сигнал, рівномірно розподілені від 0 (постійна складова) до $F/2$ максимально можливої частоти в цифровому сигналі. Таке лінійне розташування частот відрізняється від розподілу смуг третьоктавного аналізатора.

Переваги FFT

- висока швидкість обробки;
- ефективне використання ресурсів;
- висока точність;
- універсальність застосування;
- підтримка апаратного прискорення.

FFT забезпечує значне прискорення обчислень порівняно з класичним DFT, що дозволяє виконувати спектральний аналіз у реальному часі навіть на ресурсомістких пристроях.

Зниження обчислювальної складності сприяє зменшенню енергоспоживання та дозволяє використовувати менш потужні процесори, що важливо для вбудованих та мобільних систем.

FFT дає точний спектральний розклад сигналу, що є критичним для якісного аналізу звукових, радіохвильових та інших типів сигналів.

FFT використовується в широкому спектрі задач: аудіо— та відеообробка, радіозв'язок, медична діагностика, геофізика, обробка зображень, машинне навчання та інші.

Сучасні мікроконтролери і процесори часто мають апаратні блоки або інструкції для прискореного виконання FFT, що покращує продуктивність системи.

Швидке перетворення Фур'є є базовим інструментом цифрового спектрального аналізу, що забезпечує оптимальний баланс між швидкодією та точністю. Його використання дозволяє ефективно реалізувати аналіз аудіосигналів у мікропроцесорних системах, що є основою розробки обчислювального пристрою аналізатора аудіо спектру.

1.3.3 Вагові вікна

Один із найпростіших звукових сигналів — синусоїдальний тон. Як виглядатиме його спектр на FFT-аналізаторі — це залежить від частоти тону. Ми знаємо, що FFT розкладає сигнал не за тими частотами, які насправді є у сигналі, а за фіксованою рівномірною сіткою частот. Наприклад, якщо частота дискретизації дорівнює 48 кГц і розмір вікна FFT вибрано 4096 відліків, то FFT розкладає сигнал по 2049 частот: 0 Гц, 11.72 Гц, 23.44 Гц, ..., 24000 Гц.

Якщо частота тону збігається з однією із частот сітки FFT, то спектр буде виглядати «ідеально»: єдиний гострий пік вкаже на частоту та амплітуду тону (рисунок 5, білий графік).

Якщо ж частота тону не збігається з жодною із частот сітки FFT, то FFT «збере» тон із наявних у сітці частот, скомбінованих з різними вагами.

Графік спектра у своїй розмивається по частоті (рисунок 1.4, зелений графік). Таке розмиття зазвичай небажане, оскільки воно може закрити собою слабкіші звуки на сусідніх частотах.

Можна також помітити, що амплітуда максимуму зеленого графіка нижче за реальну амплітуду аналізованого тону. Це з тим, що потужність аналізованого тону дорівнює сумі потужностей коефіцієнтів спектра, у тому числі цей тон складено.

Можна також помітити, що амплітуда максимуму зеленого графіка нижча за реальну амплітуду аналізованого тону. Це пов'язано з тим, що коли частота сигналу не збігається точно з частотою жодного з бінів FFT, його енергія розподіляється між кількома частотними бін-фільтрами. У результаті жоден окремий бін не містить повної потужності сигналу, і тому максимум спектра не досягає реального значення амплітуди синусоїди.

Щоб зменшити ефект розмиття спектру, сигнал перед обчисленням FFT множиться на вагові вікна — гладкі функції, схожі на гауссіани, що спадають до країв інтервалу. Вони зменшують розмиття діапазону рахунок деякого погіршення частотного дозволу. Якщо розглядати FFT як набір смугових фільтрів, вагові вікна регулюють взаємне проникнення частотних смуг.

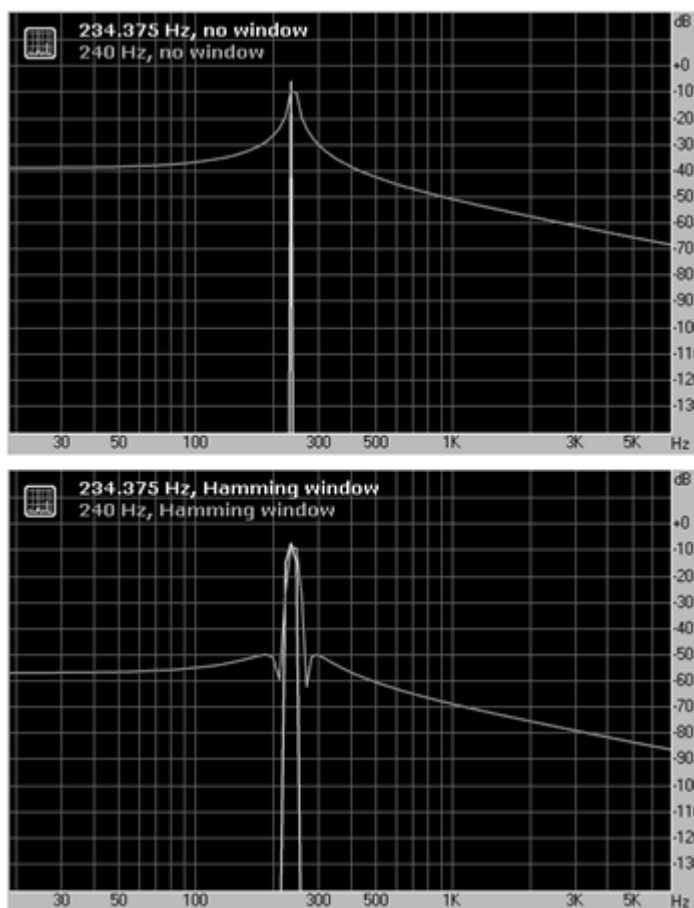


Рисунок 1.4 — Спектр синусоїдального тону різних частот з ваговими вікнами та без них

Найпростіше вікно — прямокутне, це константа 1, яка не змінює сигналу. Воно еквівалентне відсутності вагового вікна. Одне з найпопулярніших вікон — вікно Хеммінга. Воно зменшує рівень розмиття спектру приблизно на 40 дБ щодо головного піку.

Вагові вікна розрізняються за двома основними параметрами: ступенем розширення головного піку та ступенем придушення розмиття спектру («бічних пелюсток»). Чим сильніше ми хочемо придушити бічні пелюстки, тим ширшим буде основний пік. Прямокутне вікно найменше розмиває верхівку піку, але має найвищі бічні пелюстки. Вікно Кайзера (рисунок 1.5) має параметр, який дозволяє вибирати потрібний ступінь придушення бічних пелюсток.

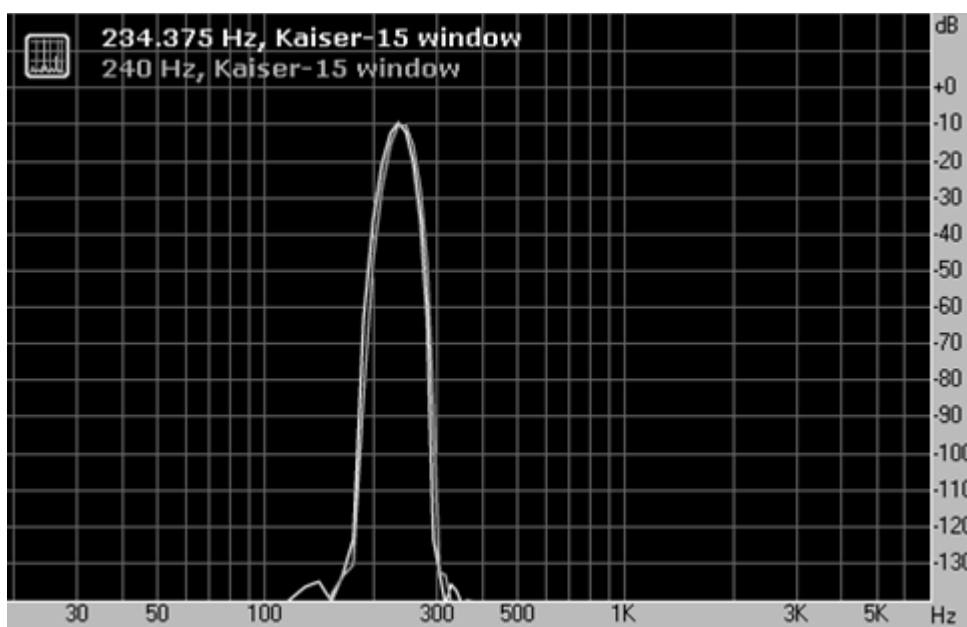


Рисунок 1.5 — Вікно Кайзера

Інший популярний вибір — вікно Хана (рисунок 1.6). Воно пригнічує максимальна бічна пелюстка слабша, ніж вікно Хеммінга, зате інші бічні пелюстки швидше спадають при віддаленні від головного піку. Вікно Блекмана (рисунок 1.7) має більш сильне придушення бічних пелюсток, ніж вікно Хана.

Більшість завдань не дуже важливо, який саме вид вагового вікна використовувати. Головне, щоб воно було. Популярний вибір — Хан чи Блекман. Використання вагового вікна зменшує залежність форми спектра від конкретної частоти сигналу та її збігу з сіткою частот FFT.

Платформи на базі мікроконтролерів AVR (наприклад, Arduino Uno, Nano) мають 10—бітний АЦП, що дозволяє здійснювати оцифрування аналогового аудіосигналу з частотою дискретизації від 10 кГц до 15 кГц. Цього достатньо для аналізу низькочастотних аудіосигналів, наприклад, мовлення, басові частоти музики.

1.3.4 Використання мікроконтролерів Arduino для обробки аудіосигналів

Більш потужні моделі, як Arduino Due або плати на базі ESP32, мають 12-бітні або 16-бітні АЦП і вищу тактову частоту, що значно розширює можливості аналізу сигналу. Зокрема, ESP32 має два ядра, 32-бітну архітектуру і апаратні блоки для математичних операцій, що робить його придатним для реалізації алгоритмів FFT.

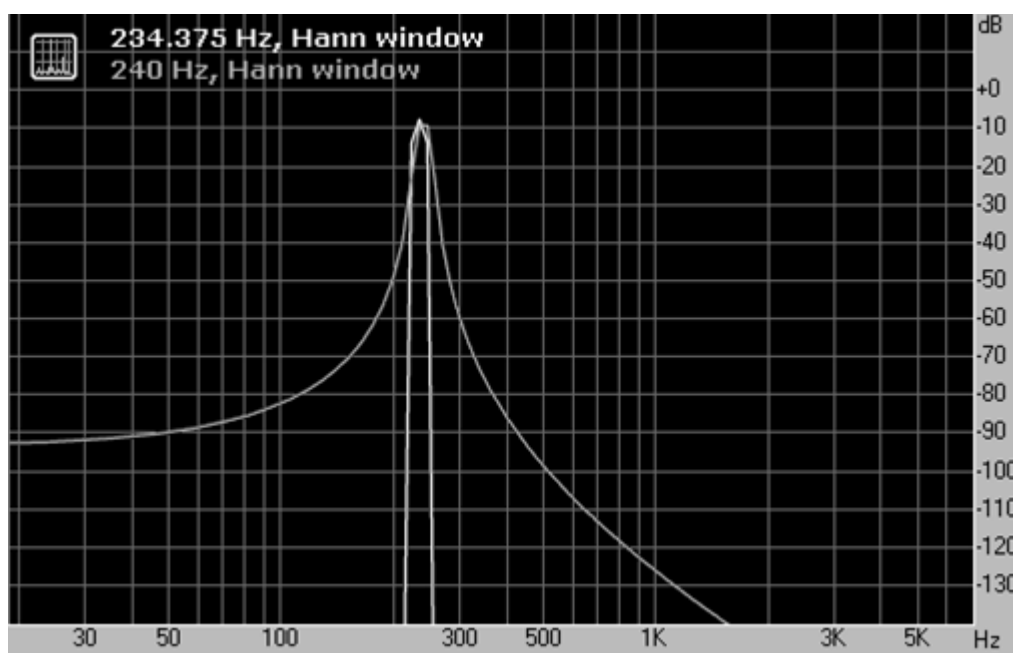


Рисунок 1.6 — Вікно Ханна

Приклади застосування:

- візуалізація спектру звуку;
- виявлення частотних компонентів;
- фільтрація шуму.

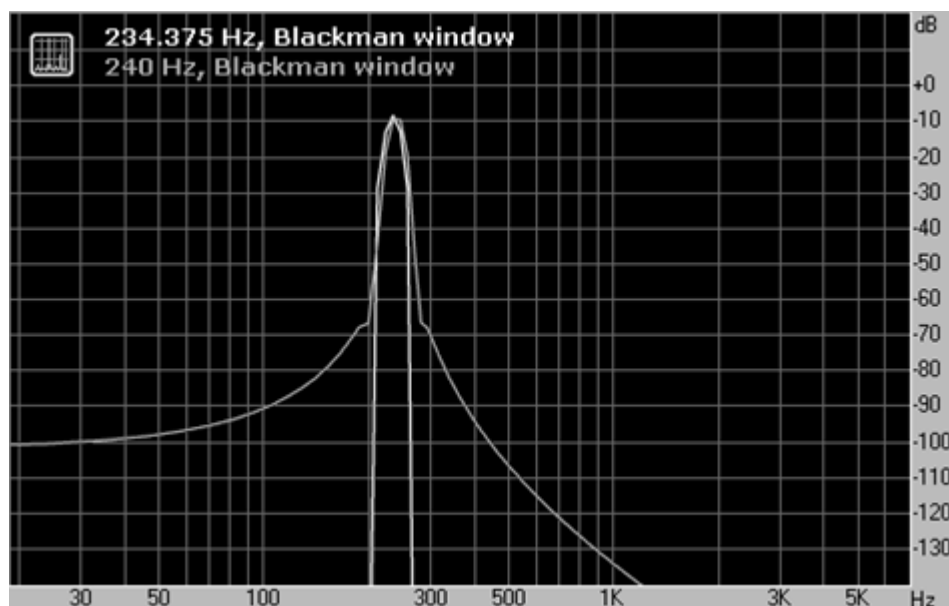


Рисунок 1.7 — Вікно Блекмана

Arduino може приймати аналоговий сигнал через аудіопідсилювач, виконувати перетворення Фур'є (через бібліотеку Arduino FFT) та виводити спектр на матрицю світлодіодів (LED), OLED-дисплей або через серійний порт.

Алгоритми на основі FFT дозволяють Arduino виявляти домінантні частоти, що корисно для реалізації музичних візуалізаторів, тюнерів, детекторів голосу, систем розпізнавання подій (удари, сигнали тощо).

Програмна реалізація цифрових фільтрів (наприклад, середньозваженого або фільтра нижніх частот) дозволяє здійснювати елементарну обробку сигналу перед аналізом.

Обмеження:

- обмежена тактова частота і оперативна пам'ять;
- низька розрядність АЦП;
- відсутність апаратного цифрового сигнального процесора (DSP).

Наприклад, мікроконтролер ATmega328P, який використовується в Arduino Uno та Nano, працює на частоті 16 МГц та має лише 2 КБ оперативної пам'яті (SRAM). Такий об'єм критично малий для виконання операцій швидкого перетворення Фур'є (БПФ) над великими масивами даних. Це означає, що довжина вибірки сигналу, який можна обробити, обмежується десятками або

сотнями значень, що знижує частотну роздільну здатність та точність спектрального аналізу. Крім того, паралельне використання пам'яті бібліотеками дисплея, введення/виведення та інтерфейсів (I2C, UART) ще більше зменшує доступний обсяг для зберігання та обробки сигналу.

Arduino Uno/Nano оснащені 10-бітним аналого-цифровим перетворювачем, що забезпечує всього 1024 рівні квантування. У практиці це означає, що амплітуди аналогового сигналу дискретизуються з кроком, який часто є надто грубим для точного вимірювання слабких сигналів. Це створює високий рівень шуму квантування, який проявляється у вигляді спотворень в спектрі, особливо в нижніх частотах і при роботі з низькоамплітудними джерелами, такими як мікрофони без підсилення.

У мікроконтролерах Arduino всі обчислювальні операції — множення, обчислення синусів/косинусів для БПФ, фільтрація, згладжування — виконуються програмно, за допомогою арифметики з плаваючою комою (або фіксованою точкою в оптимізованих реалізаціях). Це призводить до значного навантаження на процесор, що знижує частоту оновлення аналізу та робить неможливою обробку сигналу в реальному часі при високій дискретизації або з більшим числом частотних бінів. Наприклад, навіть реалізація 64-точкової БПФ може займати десятки мілісекунд — надто довго для задач на кшталт розпізнавання миттєвих звукових подій.

Arduino може бути ефективним інструментом для початкового аналізу аудіосигналів у простих або навчальних проєктах. Його відкритість, наявність бібліотек (наприклад, `ArduinoFFT`, `FixFFT`), модулів (аудіопідсилювачі, мікрофони) та низька вартість роблять його привабливою платформою для реалізації компактних пристроїв аналізу спектру звуку. У складніших випадках доцільно використовувати більш продуктивні платформи, зокрема STM32, ESP32 або Raspberry Pi.

2 ПРОЄКТУВАННЯ МІКРОПРОЦЕСОРНОГО ПРИСТРОЮ АУДІОСПЕКТРАЛЬНОГО АНАЛІЗУ

2.1 Розробка структурної схеми пристрою

Розробка структурної схеми пристрою є важливим етапом його розробки. За допомогою структурної схеми можна зрозуміти принцип роботи пристрою, знайти помилки у його роботі. Структурна схема пристрою повинна мати джерело живлення, в ролі якого може бути комп'ютер (який буде також джерелом сигналу) або будь-яке джерело з вихідною напругою 5В, потрібен мікроконтролер для обробки сигналу та дисплей для виводу даних. Загалом структурна схема буде мати наступний вигляд (рисунок 2.1).

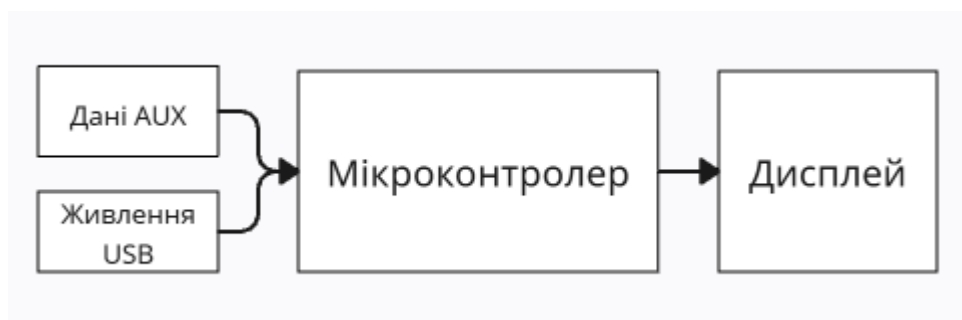


Рисунок 2.1 — Структурна схема пристрою

На даній схемі наглядно можна побачити принцип роботи нашого пристрою. Спочатку ми отримуємо дані з пристрою (комп'ютер, телефон та ін.) та забезпечуємо живлення, потім за допомогою мікроконтролера перетворюємо дані в спектр. Наступним етапом передаємо отримані дані на дисплей.

2.2 Вибір електронних компонентів для реалізації FFT-аналізатора та розробка функціональної схеми

Графічний аналізатор аудіо спектру в першу чергу призначений для розкладання аналогового звукового сигналу на код і подальшу передачу цього коду на елемент, що відтворює, а саме матрицю світлодіодів або екран.

Основним виконуючим елементом є мікроконтролер, на якому і буде базуватися побудова розроблюваного графічного аналізатора спектру, його потужності повинно вистачати для виконання програмного, швидкого

розкладання Фур'є, без допоміжних засобів, оскільки на цьому принципі буде працювати аналізатор. Модуль зв'язку повинен забезпечити з'єднання пристрою з гаджетом (телефон, планшет, персональний комп'ютер та ін.), через AUX та подачі живлення та команд управління на мікроконтролер через USB. LCD-дисплей є кінцевим елементом пристрою, що відображає основну суть його роботи, управління їй відбувається через мікроконтролер. Пристрій повинен мати постійну напругу живлення номіналом у 5В, для функціонування мікроконтролера, та світлодіодної матриці, джерелом може виступити порт USB 3.0.

В ролі керуючого мікроконтролера мій вибір впав на Arduino nano v3 (рисунок 2.2), її технічні характеристики повністю задовільняють необхідні вимоги для данного пристрою.

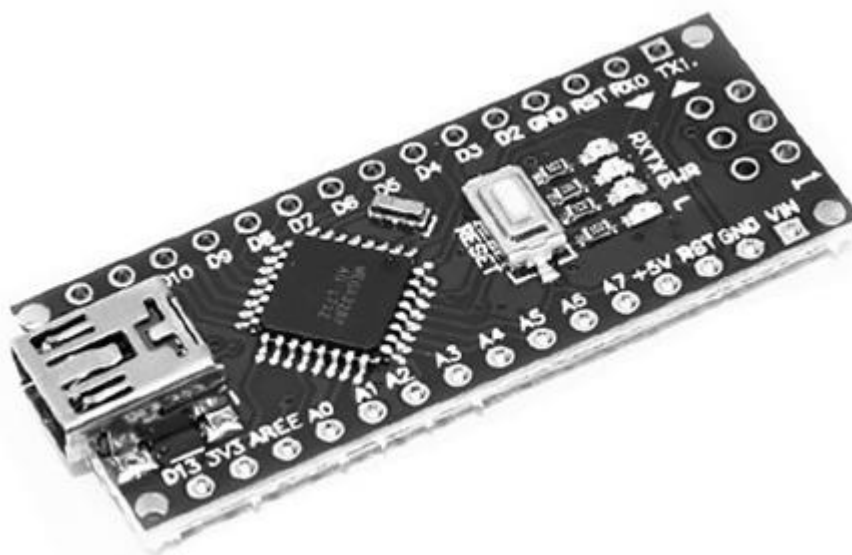


Рисунок 2.2 — Arduino nano v3

Arduino Nano — це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптований для використання з макетної плати. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю. Arduino Nano розроблено і випускається фірмою Gravitech.

В ролі компоненту виведення я обрав дисплей LCD1602 (рисунок 2.3), LCD

1602 має 16 символів і два рядки, вже впаяний модуль інтерфейсу I2C/ПС. Цей індикатор добре зарекомендував себе в наборах Arduino, що дозволяє швидко підключати його до проектів і програмувати за допомогою готової бібліотеки.

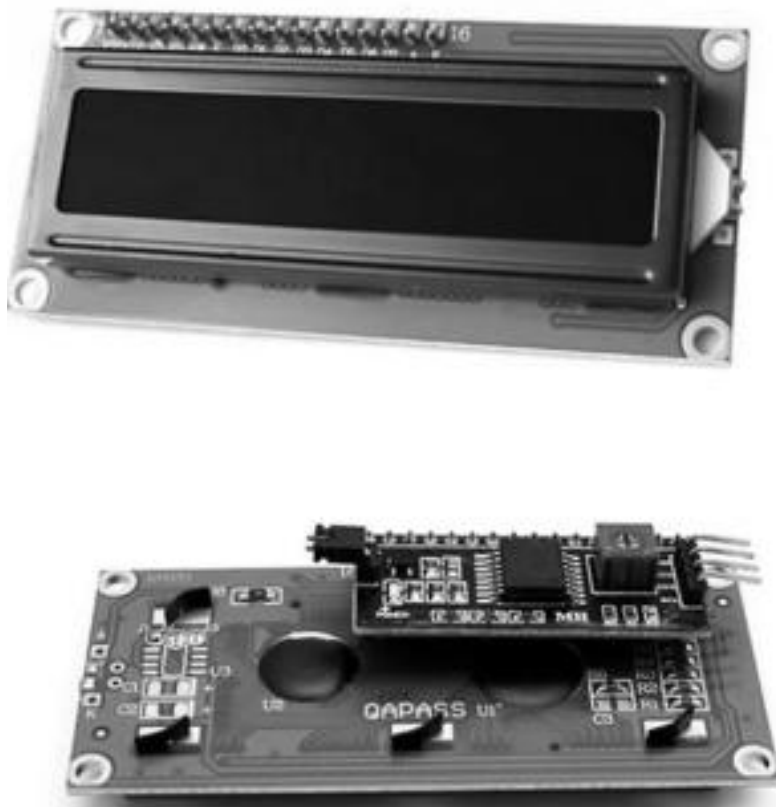


Рисунок 2.3 — LCD 1602 з інтерфейсом I2C/ПС

Lcd display Arduino 1602 використовується в багатьох проектах, схемах і пристроях так як дозволяє відображати спеціальні і навіть символи користувача. Ще одним плюсом цього дисплею є те, що він може житись від Arduino.

Для отримання аудіо сигналу потрібен роз'єм AUX (рисунок 2.4). Яких є велике різноманіття, конкретно для данного проекту немає якихось особливих критерій відбору, так як навіть наявність стерео не обов'язкова. Загалом краще обрати моно варіант, але при умові його відсутності підійде і стерео.

На рисунку 2.4 показаний стерео варіант порту, для правильної роботи пристрою потрібно буде з'єднати лівий та правий канал, при умові 2-х контактного порту цієї маніпуляції виконувати не прийдеться.



Рисунок 2.4 — Aux порт

При роботі над функціональною схемою я урахував не ідеальність умов експлуатації, можливі шуми та збої. Тому для виготовлення фізичного пристрою було додано 3 резистори опором 10кОм та один конденсатор, який має ємність 10нФ.

При реальному використанню пристрою можуть бути деякі проблеми, які зв'язані з не ідеальним спаюванням компонентів, погана якість аудіо-кабелю та навіть електро-магнітне випромінювання. Використання резисторів та конденсатору дасть можливість зменшити ці шуми та суттєво не вплине на ціну приладу.

Розглянувши варіант реального пристрою, функціональна схема буде виглядати наступним чином (рисунок 2.5).

Як ми можемо бачити, використання дисплею lcd1602 дало нам можливість жити його від самого мікроконтролера, без необхідності додаткового живлення. Це спрощує як конструкцію пристрою, так і його використання.

2.3 Інтеграція LCD1602 для виведення спектральної інформації

2.3.1 Особливості роботи з LCD1602 через I2C

LCD1602 — це рідкокристалічний дисплей, що широко використовується в мікропроцесорних системах для виводу текстової інформації. Він має два рядки по 16 символів кожен і підтримує як паралельний інтерфейс, так і послідовний

інтерфейс I2C при використанні додаткового адаптера.

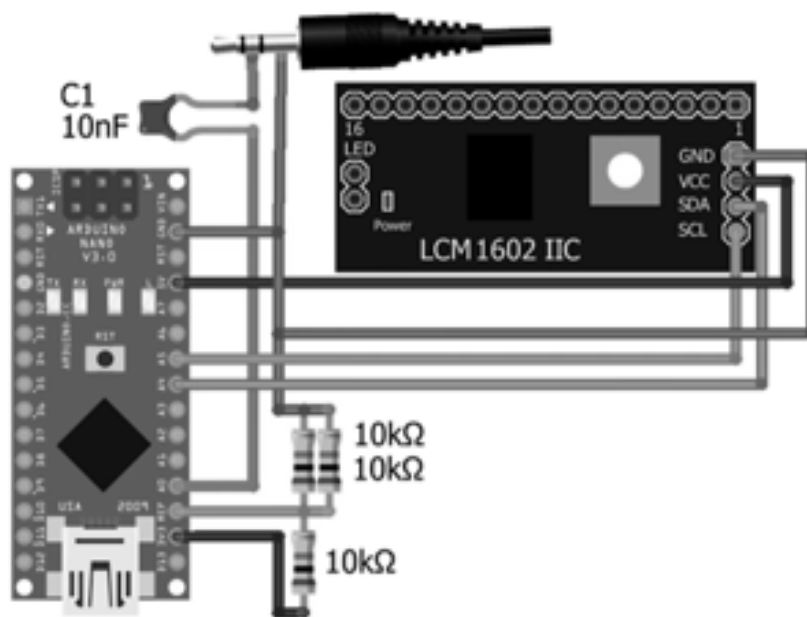


Рисунок 2.5 — Функціональна схема

Причини використання інтерфейсу I2C:

— у стандартному режимі підключення LCD1602 потребує щонайменше 6 цифрових ліній вводу/виводу мікроконтролера (RS, E, D4–D7, а також живлення й підсвічування), це суттєво зменшує доступну кількість портів для інших пристроїв;

— використання I2C-адаптера (на базі мікросхеми PCF8574) дозволяє підключити дисплей лише двома лініями — SDA (дані) та SCL (тактування), що значно спрощує схему та зменшує навантаження на мікроконтролер.

Принцип роботи — Адаптер I2C перетворює послідовні сигнали в команди для LCD1602. Кожна передача включає:

- адресу пристрою I2C (стандартно 0x27 або 0x3F);
- командний байт (інструкції управління курсором, відображенням тощо);
- дані символів, які необхідно вивести.

Для взаємодії з пристроєм у середовищі Arduino зазвичай використовується бібліотека `LiquidCrystal_I2C`, яка забезпечує абстракцію низькорівневих команд. Основні команди включають:

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // Ініціалізація дисплея
lcd.begin();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Hello, world!");
```

Переваги I2C-з'єднання:

- скорочення кількості з'єднань, використовується лише дві лінії зв'язку;
- можливість підключення кількох пристроїв до однієї шини, завдяки унікальним I2C-адресам;
- простота реалізації, підтримка великої кількості бібліотек;
- низьке енергоспоживання, особливо в режимах зниженої активності.

Обмеження:

- невисока швидкість оновлення, для деяких застосувань може бути критично;
- чутливість до довжини дроту, при великій довжині сигнал може спотворюватися;
- обмежена довжина повідомлення, максимум 16 символів на рядок без прокрутки.

Загалом, використання LCD1602 з I2C-інтерфейсом є оптимальним рішенням для компактних і простих систем виведення текстової інформації, таких як аудіоаналізatori, де необхідно зекономити порти вводу/виводу мікроконтролера.

2.3.2 Формат і логіка візуалізації спектра на дисплеї

У межах проєкту для відображення результатів аналізу аудіосигналу використовується LCD1602 із I2C-інтерфейсом. Через обмежений розмір дисплея (2 рядки по 16 символів) необхідно оптимально формувати спектральну інформацію та обирати логіку її візуалізації.

Принцип формування спектра полягає в тому, що після обробки аудіосигналу алгоритмом швидкого перетворення Фур'є (FFT) одержується набір амплітуд у частотних діапазонах. Кількість таких діапазонів (бінів) залежить від реалізації FFT (наприклад, 64, 128 або 256 точок) і вибраної глибини аналізу.

Оскільки дисплей може вивести лише 16 символів у рядку, доцільно використовувати графічне представлення спектру у вигляді стовпців різної висоти. Кожен стовпець відповідає одному частотному діапазону, а висота символу — рівню сигналу в цьому діапазоні.

Формат відображення:

- кожен символ на дисплеї відповідає одному частотному біну (всього 16 бінів, попередньо агрегованих з усієї FFT);
- два рядки використовуються для побудови двоповерхових гістограм, верхній рядок — високі рівні, нижній — низькі або середні;
- користувацькі символи (custom characters) дозволяють створити до 8 різних форм «стовпців» для побудови графіків;
- висота кожного стовпця визначається шляхом нормалізації амплітуди FFT у межах шкали (наприклад, від 0 до 8).

Логіка відображення:

- FFT обчислюється з вхідного аудіосигналу;
- отримані амплітуди агрегуються в 16 груп (за кількістю колонок);
- для кожної групи обчислюється середня амплітуда;
- результати масштабуються до значень від 0 до 8;
- для кожної позиції дисплея виводиться відповідний користувацький символ;
- екран оновлюється з певною періодичністю від 10 до 20 Гц.

Формат візуалізації спектра на дисплеї LCD1602 обмежений апаратними можливостями, однак завдяки використанню кастомних символів можна реалізувати просту, але інформативну гістограму спектрального аналізу. Це

дозволяє реалізувати візуальний контроль аудіоінформації навіть на недорогих та обмежених апаратних засобах.

2.3.3 Переваги використання LCD1602 у мікропроцесорних пристроях

LCD1602 — це один із найпоширеніших текстових дисплеїв, який широко використовується у мікропроцесорних системах завдяки своїй простоті, надійності та низькій вартості. Він має 2 рядки по 16 символів кожен і підтримує до 8 користувацьких символів.

Основні переваги:

- низька вартість та доступність;
- простота підключення через I2C;
- низьке енергоспоживання;
- підтримка користувацьких символів;
- широка підтримка серед розробників;
- надійність у роботі;
- інтуїтивне сприйняття текстової інформації.

Використання LCD1602 у мікропроцесорному пристрої аналізатора аудіо спектра є доцільним вибором через баланс між функціональністю, енергоспоживанням, простотою керування та вартістю. Цей дисплей дозволяє реалізувати просту та наочну візуалізацію результатів спектрального аналізу без складних ресурсомістких графічних компонентів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Огляд Arduino IDE як середовища розробки

Платформа Arduino — це сімейство мікроконтролерів на базі процесорів Atmel, STM и ARM. Усі мікроконтролери програмуються на мові C/C++ в середовищі розробки Arduino IDE. Більшість плат Arduino (окрім деяких маленьких, таких як micro, pro mini) мають ідентичне розміщення виводів (піни, pins) та дозволяють підключення уніфікованих сторонніх модулів, що звуться шилдами (Shield). На всіх платах є набір цифрових та аналогових пінів, а також інтерфейси SPI і I2C. Для роботи зі сторонніми модулями середовище розробки має менеджер бібліотек, де зібрано найчастіше використовувані для Arduino бібліотеки.

Більшість мікроконтролерів мають підтримку USB (вбудовану або зовнішню як окремий перетворювач) та підключаються до операційної системи як послідовний порт (COM порт). COM порт — спеціальний порт для послідовної передачі даних між пристроями. Може бути апаратним (спеціальний COM роз'єм на системній платі ПК), або емулюється поверх іншого апаратного протоколу (наприклад, поверх USB, як у випадку з Arduino). Послідовний порт використовується для завантаження програми на мікроконтролер, а також може використовуватися для взаємодії ПК и програми для мікроконтролера (інша назва програми в Arduino IDE — "скетч").

Скетч для мікроконтролера Arduino уявляє собою файл з розширенням .ino, який містить код на мові C/C++.

Код включає декілька основних блоків:

- підключення файлів бібліотек;
- оголошення глобальних змінних, констант, макроозначень;
- оголошення функцій, структур та класів.

Функція `setup()` — викликається один раз на початку роботи програми, саме тут потрібна пройти ініціалізація контролера — налаштування пінів, запуск послідовного порту, SPI і I2C інтерфейсів. Функція `loop()` — викликається мікроконтролером в нескінченному циклі, тут відбувається основна робота.

Після завантаження на мікроконтролер програма зберігається навіть після від'єднання Arduino, до тих пір, поки вона не буде перезаписана новою програмою.

Встановлення середовища розробки — завантажити потрібну версію середовища розробки для встановлення на ПК. Запустити на виконання програму встановлення середовища розробки (рис. 3.1).

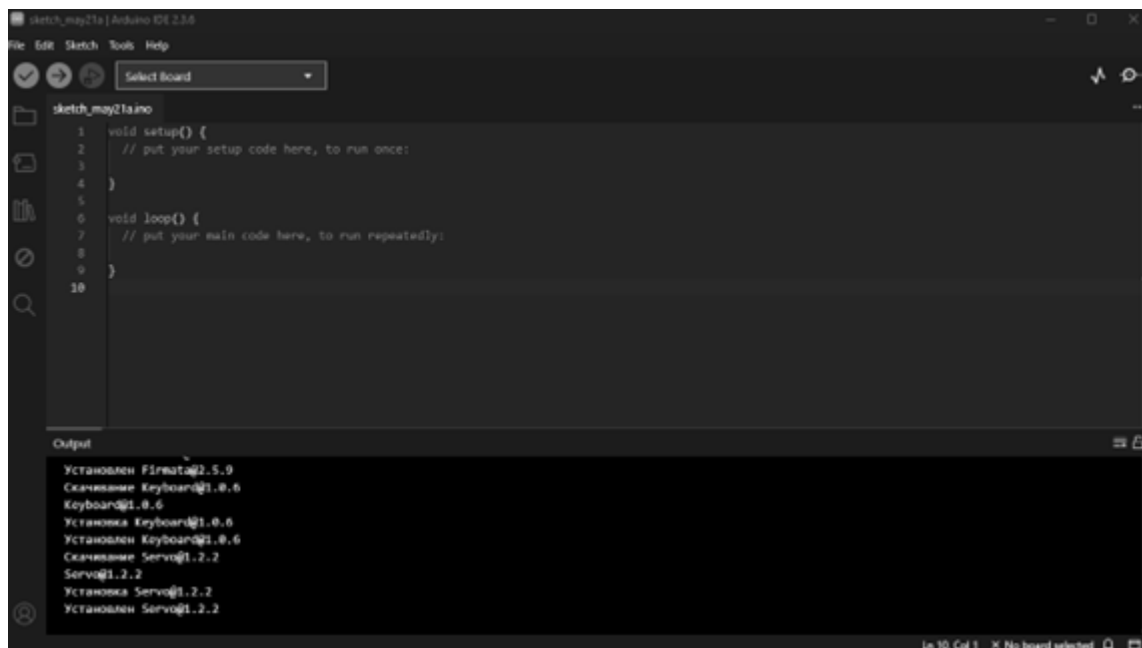


Рисунок 3.1 — Інтерфейс Arduino IDE

Підключення плати до середовища розробки — підключити плату до ПК USB кабелем. В середовище розробки обрати потрібну версію плати (зазвичай на платі вказано її найменування) (рис. 3.2, рис. 3.3).

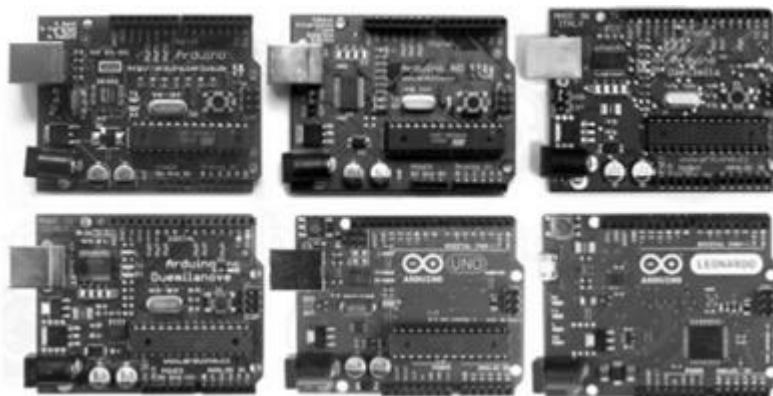


Рисунок 3.2 — Плати Arduino

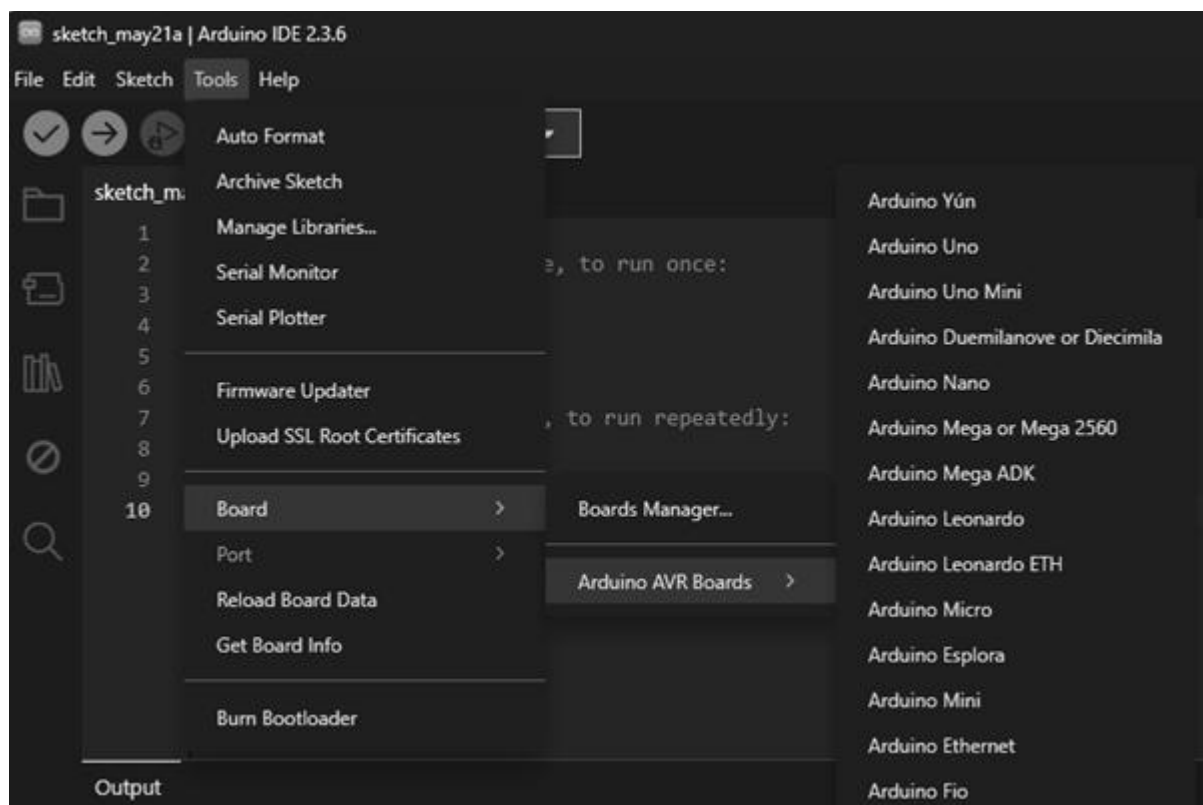


Рисунок 3.3 — Обрання версії плати Arduino в середовищі IDE

Далі обрати потрібний порт, В ОС Windows порти відображаються з назвами виду COM1, COM2

Завантаження тестового прикладу — Завантажте скетч Blink зі стандартних бібліотек прикладів (FILE —> Examples —> 0.1Basics —> Blink). Перевірте обраний порт та плату. Натисніть кнопку Upload або оберіть пункт меню "Sketch —> Upload". Після успішного завантаження в нижній частині вікна середовища з'явиться напис: «Upload completed», а на платі почне мерехтити світлодіод.

Керування бібліотеками, іноді, в процесі роботи, виникає потреба в підключенні сторонніх бібліотек, наприклад, для роботи з різноманітними датчиками. Бібліотека уявляє собою набір заголовкових файлів (з розширенням .h) та файлів з кодом (розширення .c або .cpp). У разі підключення бібліотеки в скетч додаються підключення потрібних заголовкових файлів (рядок виду `#include`). Наприклад, підключімо стандартну бібліотеку Mouse. Обираємо " Sketch —> Include library —> Mouse". При цьому IDE автоматично

добавить зміни в скетч, що є необхідними для підключення бібліотеки (рис. 3.4). У верхній частині скетча з'явиться рядок `#include`.

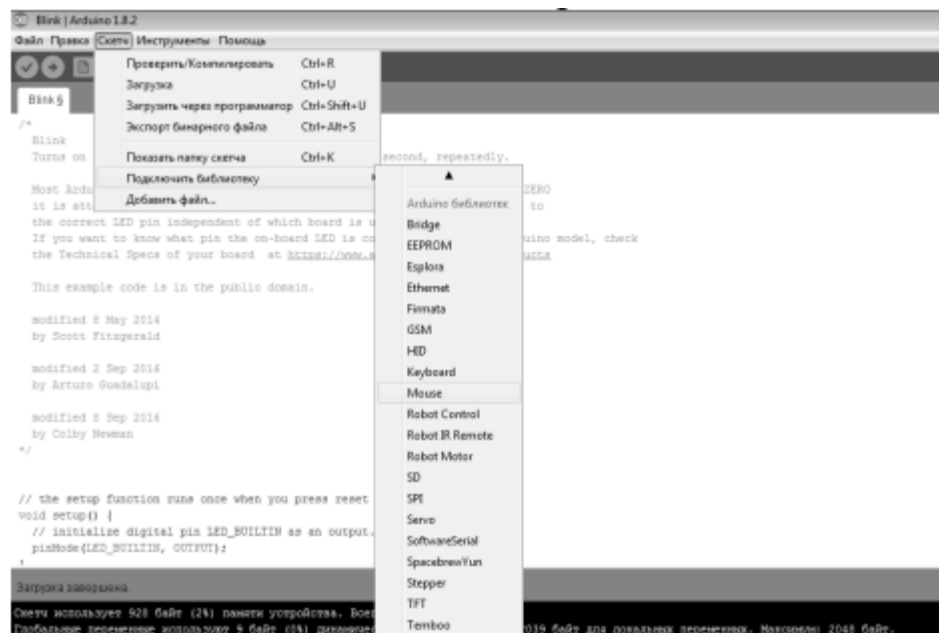


Рисунок 3.4 — Підключення стандартних бібліотек

Оразу після встановлення в середовище розробки вже доступний базовий набір бібліотек. Інші бібліотеки спочатку потрібно завантажити в IDE, для цього використовується менеджер бібліотек "Sketch —> Include library —> Manage libraries". В менеджері присутні бібліотеки для роботи с різноманітними модулями/протоколами, тощо. Після завантаження бібліотеки вона становиться доступної для підключення к скетчу.

В верхній частині менеджера знаходяться два поля фільтрації по типу і темі, а також поле текстового пошуку за назвою та описом бібліотеки (рисунок 3.5). Якщо обрати бібліотеку з переліку з'явиться кнопка "Install" (для деяких бібліотек також можна обрати версію, якщо є декілька версій).

3.2 Розробка алгоритму та програмного забезпечення

Для роботи правильної роботи пристрою необхідно щоб він отримав аудіо сигнал, потім розклав його у спектр та вивів результат на дисплей. Для розкладу в спектр можна було б використати швидке перетворення Фур'є, але це не

доцільно тому, що для Arduino існують готові бібліотеки оцифрування звуку та швидке перетворення Хартлі, яке працює ще швидше ніж швидке перетворення Фур'є. У прошивці для естетичного виведення частот без різких переходів потрібно правильно реалізувати діаграму виділення частот, її взято з фабричного аналізатору (рисунок 3.6).

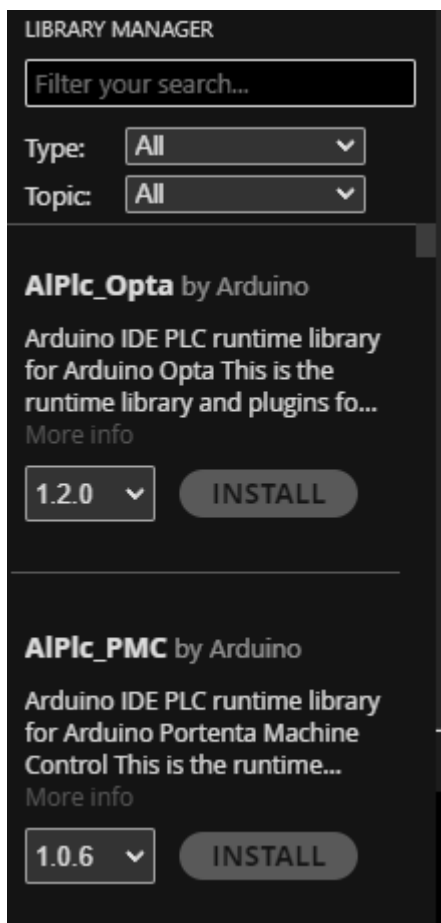


Рисунок 3.5 — Менеджер бібліотек

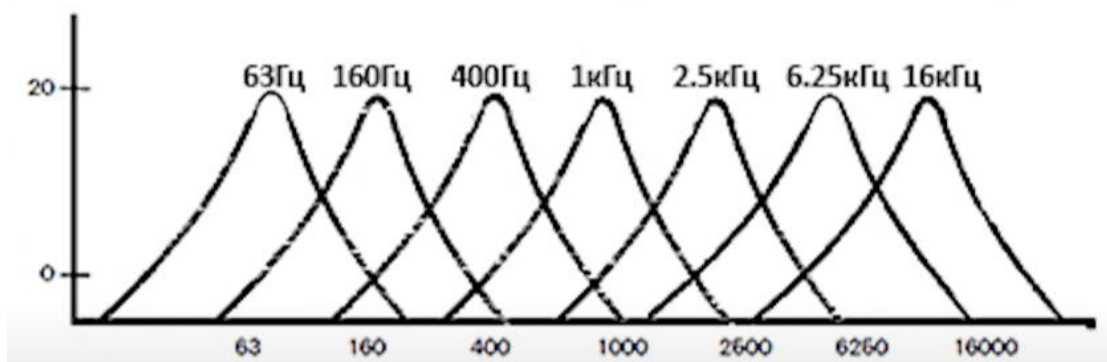


Рисунок 3.6 — Діаграма виділення частот

Спочатку були задані параметри налаштування системи, які визначають базову конфігурацію пристрою. Вони включають вибір драйвера дисплея (їх є два типи, і без правильного маркування дисплеїв не запрацює), а також режими регулювання гучності: ручний, автоматичний або з фіксованим значенням. Також встановлюються кілька порогових значень — поріг тиші (LOW_PASS) і чутливості (VOL_THR), які визначають, наскільки слабкий сигнал ще вважається значущим для аналізу. Окремо задається максимальний поріг гучності, що використовується лише тоді, коли авто- і ручне регулювання вимкнені. Окрім цього, вказано довжину вибірки для перетворення Хартлі — чим більше значення, тим точніше аналіз, але більше навантаження на мікроконтролер:

```
#define DRIVER_VERSION 0
#define GAIN_CONTROL 0
#define AUTO_GAIN 0
#define VOL_THR 35
#define LOW_PASS 30
#define DEF_GAIN 80
#define FHT_N 256
```

Після цього було вручну сформовано масив із позицій спектру (posOffset), який відповідає 16 різним частотним зонам, що відображатимуться на дисплеї. Цей масив визначає, які частоти будуть використані для побудови кожної з колонок спектрограми:

```
byte posOffset[16] = {2, 3, 4, 6, 8, 10, 12, 14, 16, 20, 25, 30, 35, 60, 80, 100};
```

Далі визначено пін, з якого зчитується аналоговий сигнал — це аудіовхід пристрою. У цьому випадку використовується аналоговий вхід A0:

```
#define AUDIO_IN 0
```

Після цього до прошивки були підключені всі необхідні бібліотеки. Серед них — бібліотека для перетворення Хартлі (FHT), яка дозволяє проводити частотний аналіз сигналу в реальному часі. Також підключено бібліотеки для

роботи з I2C-дисплеєм. Додатково заданий макрос printByte, який спрощує виведення кастомних символів на LCD:

```
#define LOG_OUT 1
#include <FHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define printByte(args) write(args);
double prevVolts = 100.0;
```

Далі реалізовано автоматичне визначення моделі дисплея: залежно від того, яке значення встановлено в DRIVER_VERSION, ініціалізується дисплей з адресою 0x27 або 0x3f. Це дає змогу використовувати прошивку з різними модулями без потреби вручну змінювати код.:

```
#if (DRIVER_VERSION)
LiquidCrystal_I2C lcd(0x27, 16, 2);
#else
LiquidCrystal_I2C lcd(0x3f, 16, 2);
#endif
```

Наступним кроком у прошивці було створення набору спеціальних символів, що формують стовпці спектрограми — від найнижчого (v1) до повністю заповненого (v8). Кожен символ — це 8-бітова матриця, яка відображається на дисплеї як вертикальний рівень. Їхнє використання дозволяє візуалізувати амплітуду кожної частотної смуги у вигляді «барів», що динамічно змінюються залежно від гучності сигналу в цій смузі:

```
byte v1[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111};
byte v2[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111};
byte v3[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111};
byte v4[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111};
```

```
byte v5[8] = {0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
```

```
byte v6[8] = {0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
```

```
byte v7[8] = {0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
```

```
byte v8[8] = {0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
```

Останнім важливим елементом в цьому фрагменті є механізм адаптивного налаштування рівня підсилення сигналу. Якщо ввімкнено автоматичне регулювання (AUTO_GAIN), система постійно відстежує максимальне значення амплітуди сигналу та поступово адаптує коефіцієнт підсилення (gain) на основі згладженого значення. Згладжування відбувається через експоненційне усереднення, що дозволяє уникати різких стрибків та забезпечує більш стабільну візуалізацію. Кожні півтори секунди (1500 мс) проводиться оновлення значення підсилення, якщо нове значення перевищує поріг гучності, воно приймається як новий коефіцієнт, інакше система повертається до стандартного рівня підсилення. Це дозволяє системі динамічно реагувати на зміну гучності вхідного сигналу, але без зайвої чутливості до короткочасних шумів:

```
if (AUTO_GAIN) {
    maxValue_f = maxValue * k + maxValue_f * (1 — k);
    if (millis() — gainTimer > 1500) {
        if (maxValue_f > VOL_THR) gain = maxValue_f;
        else gain = 100;
        gainTimer = millis();
    }
}
```

4 МОДЕЛЮВАННЯ ТА МАКЕТУВАННЯ РОБОТИ

4.1 Моделювання пристрою у Proteus

Для перевірки правильності побудови електричної принципової схеми, а також коректності розробленої прошивки, був виконаний запуск моделі пристрою у програмному середовищі Proteus (рисунок 4.1). Це програмне середовище дозволяє здійснювати віртуальне моделювання роботи електронних схем, що значно спрощує процес тестування та пошуку помилок на ранніх етапах розробки. Завдяки емуляції стало можливим здійснити попередню оцінку працездатності мікроконтролера, перевірити відповідність електричних з'єднань, а також проаналізувати наявність сигналів на виводах компонентів відповідно до заданої проектної логіки.

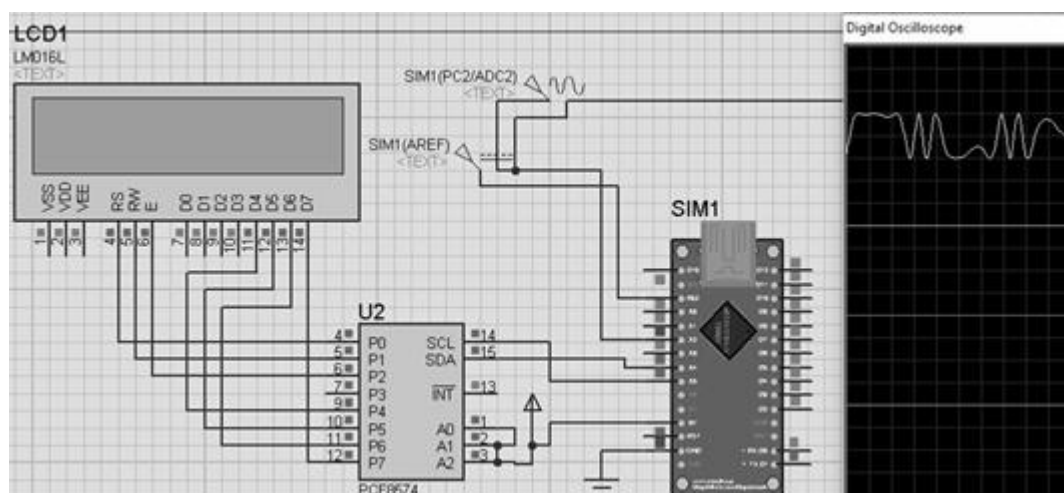


Рисунок 4.1 — Модель з підключеним осцилографом

Під час моделювання пристрій успішно проходив фазу ініціалізації, що свідчить про правильну реалізацію схеми живлення, коректне з'єднання мікроконтролера з периферійними елементами та відсутність критичних помилок у схемотехнічній частині. Також це підтверджує, що базова конфігурація мікроконтролера була реалізована згідно з вимогами проекту, що є необхідною умовою для подальшої розробки та розгортання функціональної логіки пристрою.

Однак у процесі тестування виявилось, що інформація не виводиться на дисплей, що унеможливлювало візуалізацію даних. Поглиблений аналіз виявленої проблеми показав, що її причиною є помилка у програмному забезпеченні, зокрема — некоректне налаштування процедури ініціалізації дисплейного модуля. Через це дисплей не реагував на команди мікроконтролера, хоча електричні з'єднання були вірними.

Виявлення цієї помилки на етапі програмного моделювання виявилось вкрай корисним, оскільки дозволило вчасно внести необхідні зміни у код прошивки без потреби у фізичному втручанні в апаратну частину. Це значно зменшило ризики подальших ускладнень на етапі виготовлення реального пристрою та прискорило загальний процес розробки. В результаті, після коригування ініціалізації дисплея, система успішно продовжила функціонування відповідно до запланованого алгоритму, що підтвердило ефективність використання програмного середовища Proteus для перевірки як електронної, так і програмної частин проекту.

Під час перевірки правильності написання програмного забезпечення критичних помилок виявлено не було. Однак у структурі прошивки реалізовано механізм автоматичного налаштування під тип використовуваного дисплея, що пов'язано з тим, що існують два різновиди дисплеїв стандарту 1602, які мають відмінності в логіці ініціалізації. Цей функціонал необхідний для універсальності роботи пристрою з різними апаратними конфігураціями.

Проте симуляційне середовище Proteus не підтримує обидва варіанти дисплеїв у повному обсязі, що унеможливлювало коректне моделювання з увімкненим автоналаштуванням. З метою забезпечення стабільної роботи пристрою у віртуальному середовищі зазначений функціонал було тимчасово вилучено.

Після внесення відповідних змін до прошивки пристрій почав функціонувати згідно з очікуваннями (рисунок 4.2): на дисплеї коректно виводиться розроблене привітальне повідомлення, що підтверджує працездатність системи у симульованому режимі.

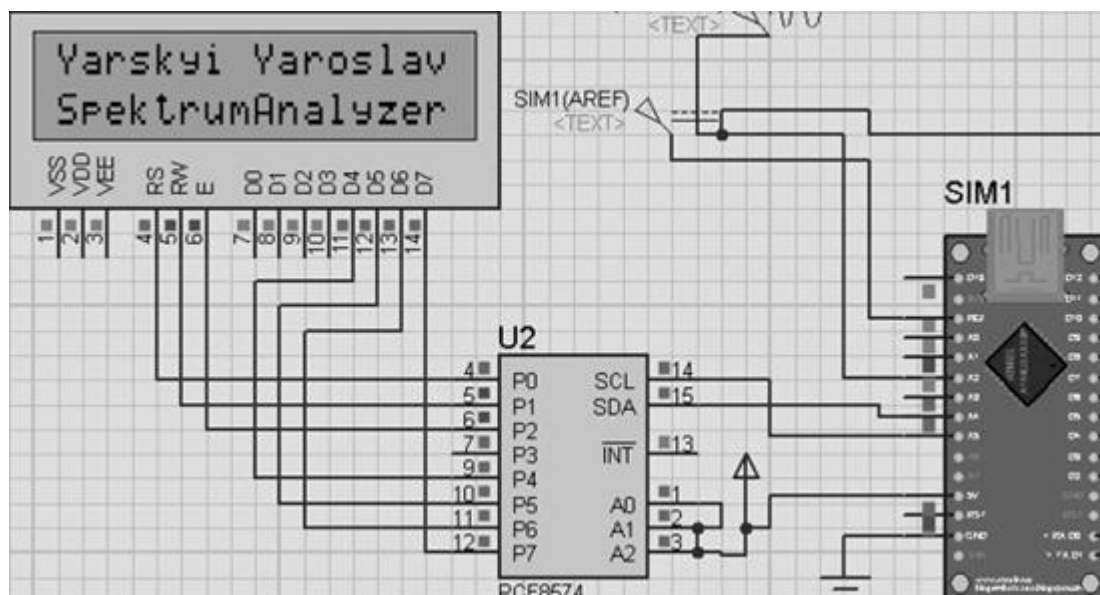


Рисунок 4.2 — Робота пристрою

4.2 Тестування пристрою у середовищі Proteus

На наступному етапі було проведено тестування роботи пристрою з використанням генератора сигналів (рисунок 4.3). Метою даного етапу було перевірити реакцію системи на вхідний сигнал заданої частоти та переконатися у коректності його обробки пристроєм.

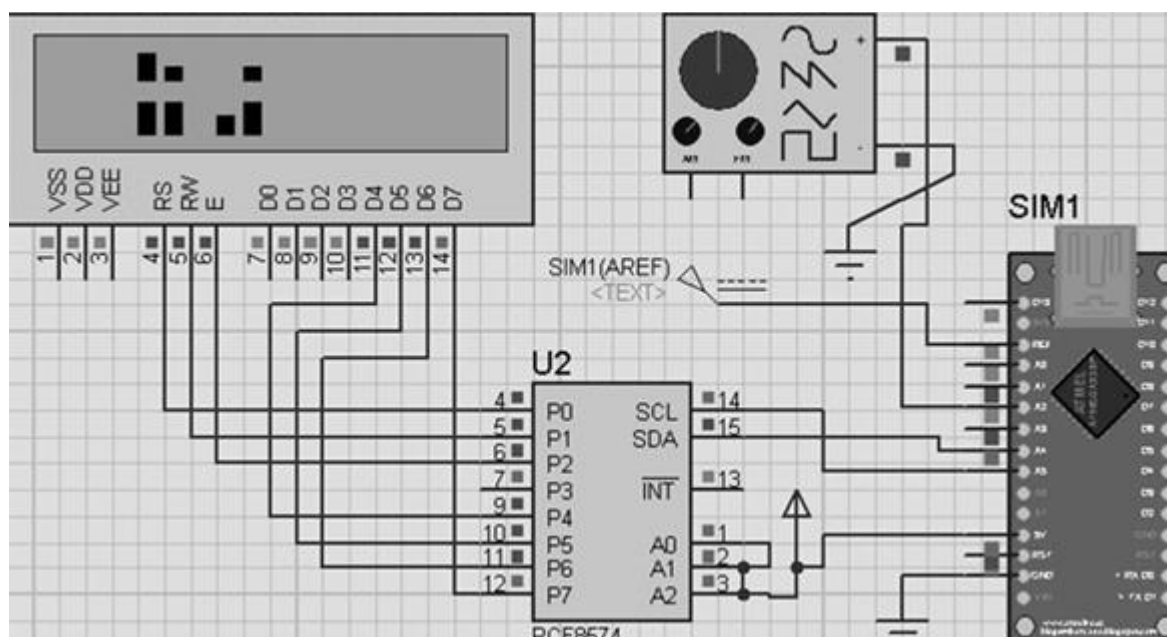


Рисунок 4.3 — Тест з генератором частот

До вхідного каналу аналізатора було підключено функціональний генератор, який формував гармонійні сигнали з різною частотою та амплітудою. Під час тестування відстежувалась стабільність роботи мікроконтролера, коректність обчислення параметрів сигналу та відображення відповідної інформації на дисплеї. За результатами випробувань встановлено, що пристрій коректно реагує на зміни частоти вхідного сигналу, демонструючи відповідність роботи очікуваному алгоритму функціонування.

4.3 Складення електронної схеми на макетній платі

Спочатку необхідно приєднати три резистори номіналом 10 кОм до мікроконтролера Arduino згідно зі схемою, зображеною на рисунках 4.4 та 4.5. Кожен резистор слід підключити до відповідного входу мікроконтролера. Це дозволить стабілізувати сигнали, уникнути шумів і забезпечити коректне зчитування значень. Правильне розташування резисторів та їх підключення згідно зі схемою є критично важливим для забезпечення надійної роботи системи.

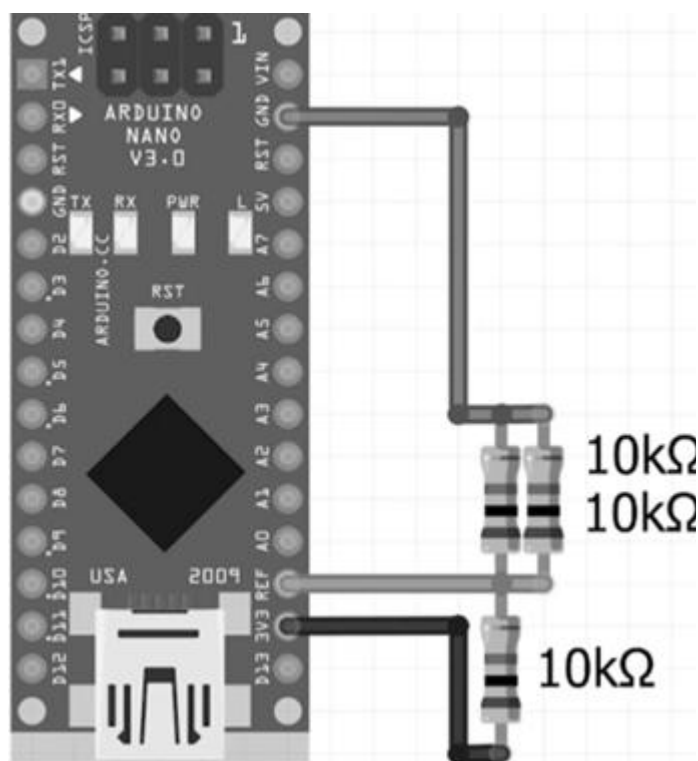


Рисунок 4.4 — Схематичне зображення підключення резисторів

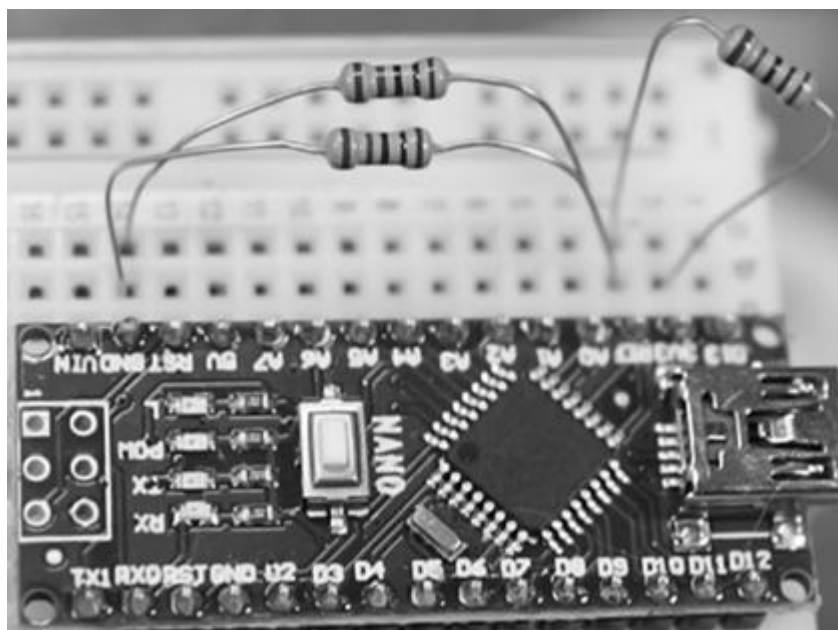


Рисунок 4.5 — Підключення резисторів

Потім підключаємо вхід AUX. Для цього мінусовий провід слід з'єднати з загальною землею (масою) системи, а позитивний провід підключити до аналогового входу мікроконтролера Arduino через конденсатор ємністю 10 нФ (рисунок 4.6).

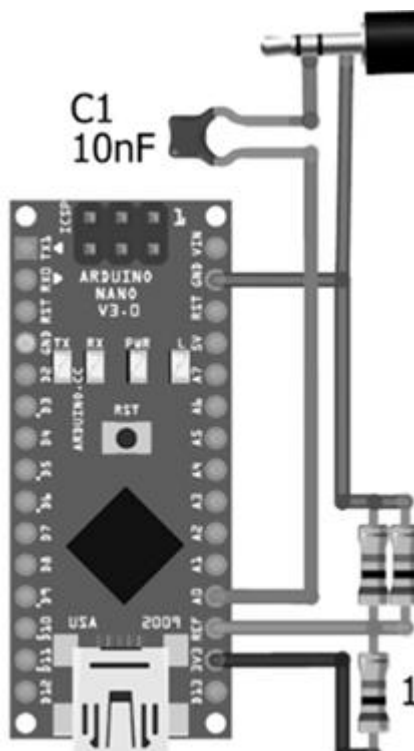


Рисунок 4.6 — Підключення конденсатору

Конденсатор виконує функцію фільтрації високочастотних шумів і захищає вхід від раптових імпульсних сигналів, що покращує якість зчитування аналогового сигналу з AUX-входу. Така схема підключення забезпечує стабільне та чисте зняття сигналу для подальшої обробки.

4.4 Тестування пристрою

Для початку був проведений первинний тест приладу без підключення дисплею (рисунок 4.7). Такий підхід дозволив сфокусуватися на перевірці базової функціональності основних апаратних модулів пристрою без додаткових перешкод і потенційних джерел помилок, пов'язаних з передачею даних на дисплей. У ході цього тесту основну увагу було приділено коректності зчитування сигналів з аналогових входів, а також стабільності роботи мікроконтролера в умовах, наближених до реальних, але без візуального виводу.

Як видно на рисунку 4.7, прилад справно обробляє вхідні аналогові дані — це підтверджує, що пристрій функціонує правильно, і основна логіка програми виконується без критичних збоїв. Проте на цьому етапі було виявлено, що амплітуда сигналу, що надходить на вхід, є недостатньою для повноцінного аналізу і візуалізації спектру. Така ситуація могла бути викликана як слабким рівнем сигналу від джерела.

З метою усунення цього недоліку був проведений наступний етап тестування — із увімкненою функцією автоматичного налаштування гучності (рисунок 4.8).

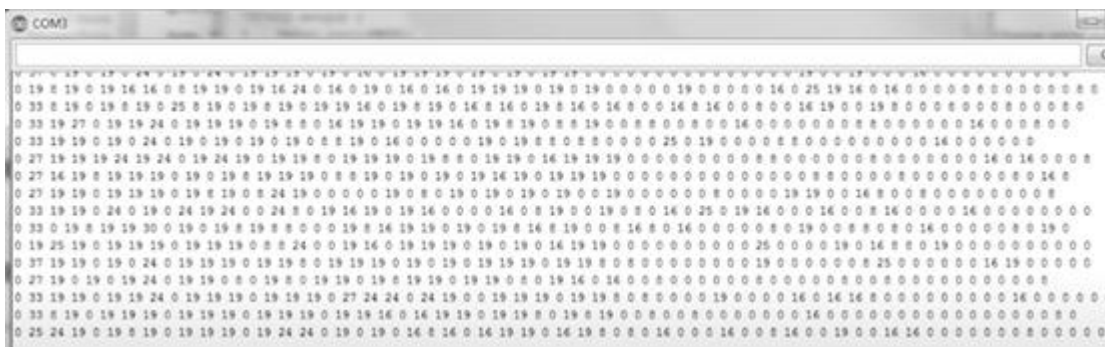


Рисунок 4.7 — Тест без автоматичного налаштування гучності

Цей режим дозволяє пристрою адаптивно підлаштовувати коефіцієнт підсилення сигналу на основі його поточного рівня, що забезпечує більш точну і стабільну реакцію системи на зміну гучності вхідного сигналу.

Ця функція дозволяє автоматично підсилювати або ослаблювати вхідний сигнал до оптимального рівня, що забезпечує більш стабільну та якісну обробку аудіоданих незалежно від початкової сили сигналу. Результати тесту підтверджують ефективність такого підходу у поліпшенні роботи пристрою.

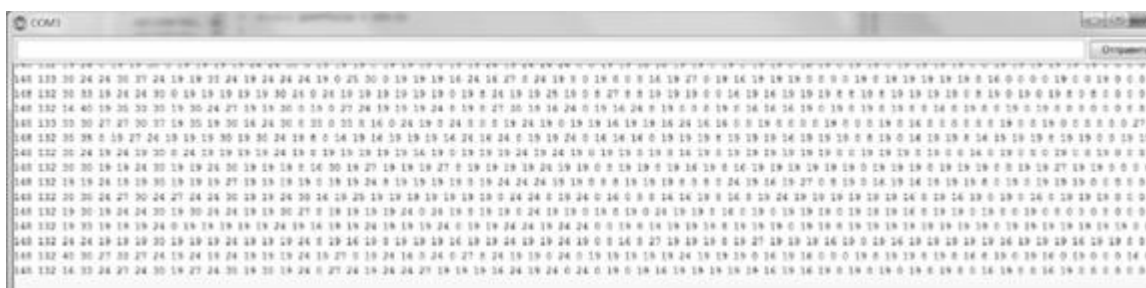


Рисунок 4.8 — Тест після включення автоматичного налаштування гучності

Далі були проведені тести роботи пристрою з підключеним дисплеєм (рисунок 4.9). Метою цього етапу було перевірити коректність відображення оброблених даних на екрані, а також оцінити взаємодію між мікроконтролером та дисплеєм у реальному часі. Під час тестування було підтверджено стабільність роботи інтерфейсу.

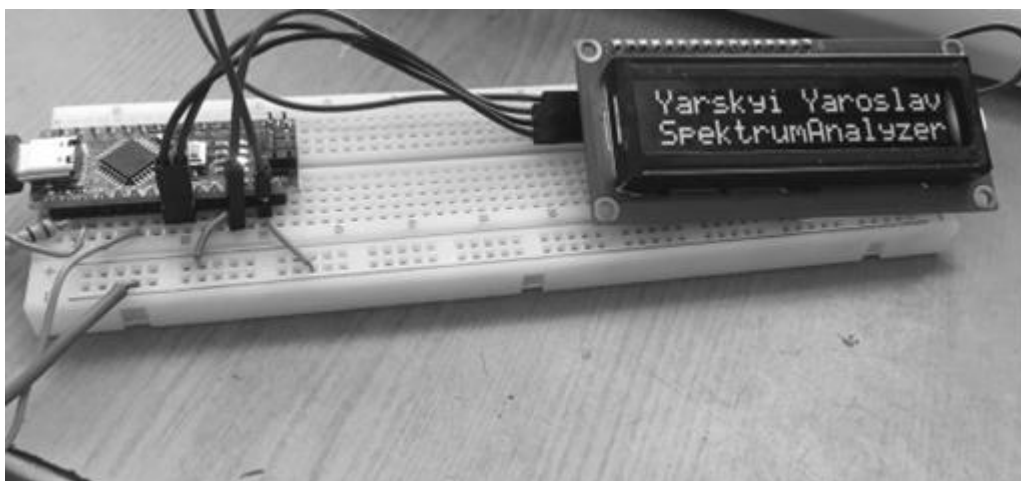


Рисунок 4.9 — Ініціалізація прототипу

Після цього був проведений тест шумів з метою коректного налаштування алгоритмів заглушення шумів у прошивці (рисунок 4.10). У процесі випробування проводився збір даних з аналогових входів пристрою в умовах навмисно змодельованого та природного шумового фону. Аналіз результатів дозволив ідентифікувати рівень фонових перешкод, частотні характеристики шумових сигналів, а також їхню мінливість у часі. Було зафіксовано найбільш критичні діапазони впливу, у яких спостерігалось зниження точності вимірювань.

Ці заходи дозволили значно підвищити точність зчитування даних, зменшити кількість хибних спрацьовувань, а також забезпечити стабільність роботи пристрою навіть за наявності суттєвих електричних перешкод. Таким чином, тест шумів став ключовим етапом у підвищенні надійності та функціональної стійкості системи в умовах реального середовища експлуатації.

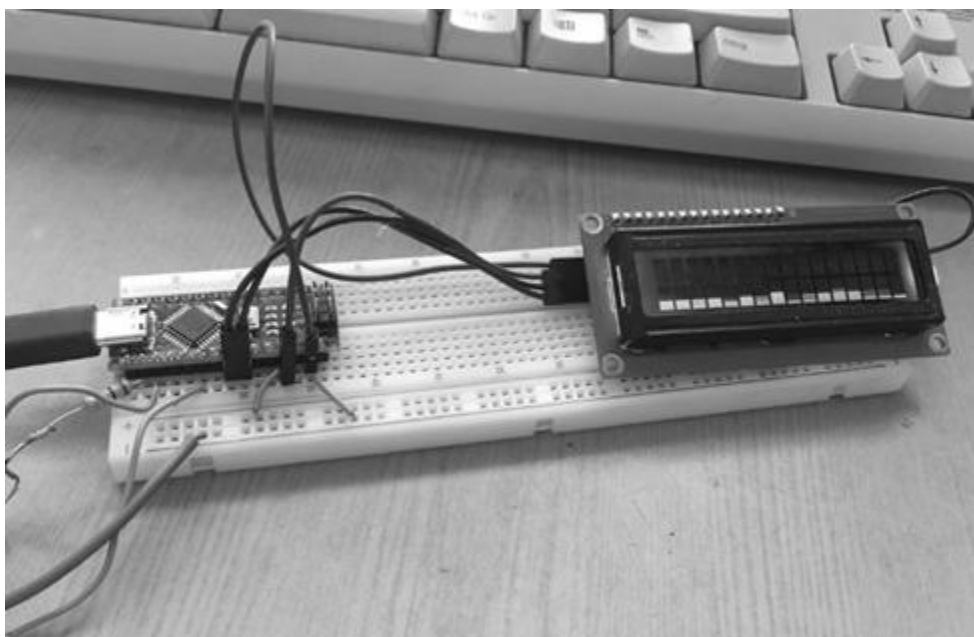


Рисунок 4.10 — Перевірка рівня шуму

Наступним був проведений тест із використанням генератора частотних коливань для оцінки точності та стабільності роботи пристрою при заданих частотах. Перший тест здійснювався на частоті 2178 Гц (рисунки 4.11 та 4.12).



Рисунок 4.11 — Тест №1

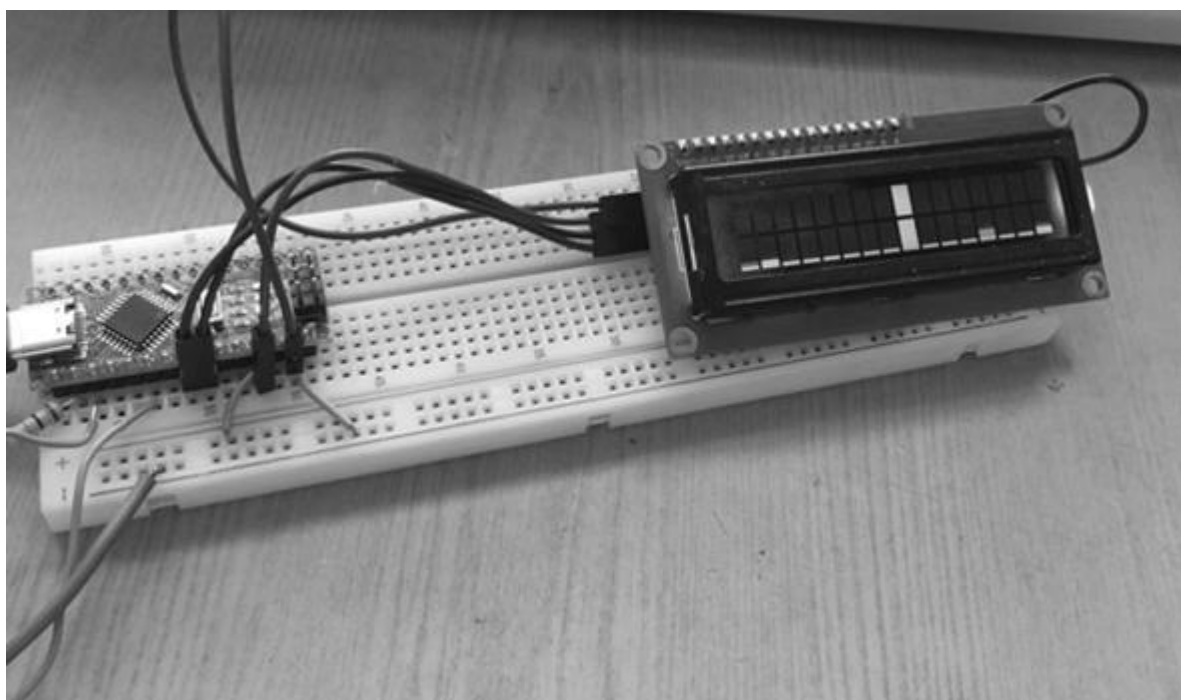


Рисунок 4.12 — Результати тесту №1

Метою цього тесту було перевірити коректність виявлення та обробки сигналу заданої частоти, а також оцінити відповідність вихідних даних очікуваним результатам. При цьому були помічені незначні шуми, що виникли через неідеальність конструкції пристрою та вплив зовнішніх умов. Аналіз отриманих показників дозволив підтвердити загальну точність роботи пристрою

на конкретній частоті та ефективність алгоритмів аналізу аудіо спектру з урахуванням цих шумових факторів.

Другий тест був проведений на частоті 7903 Гц (рисунки 4.13 та 4.14).

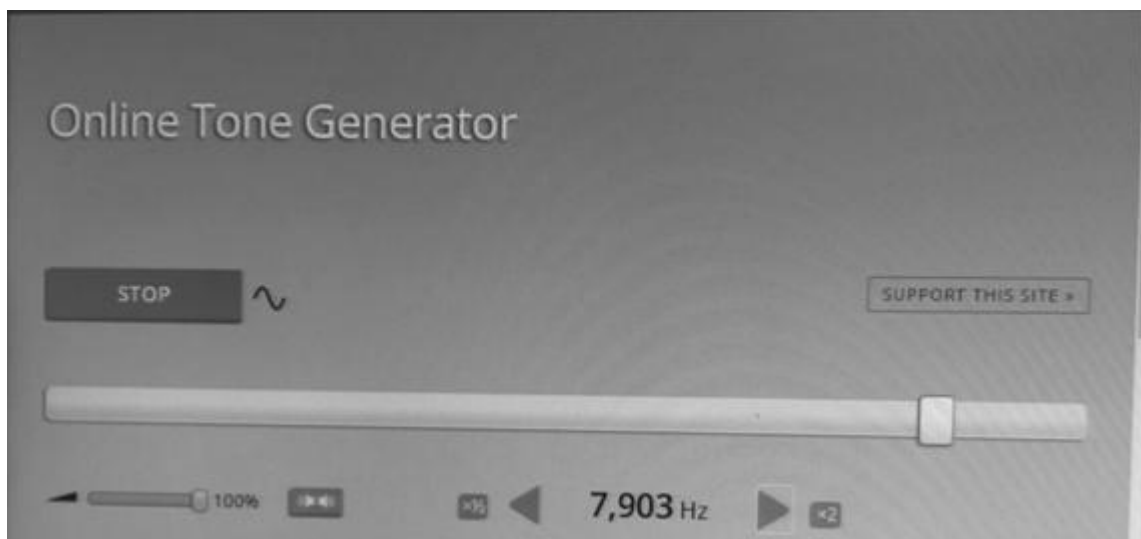


Рисунок 4.13 — Тест №2

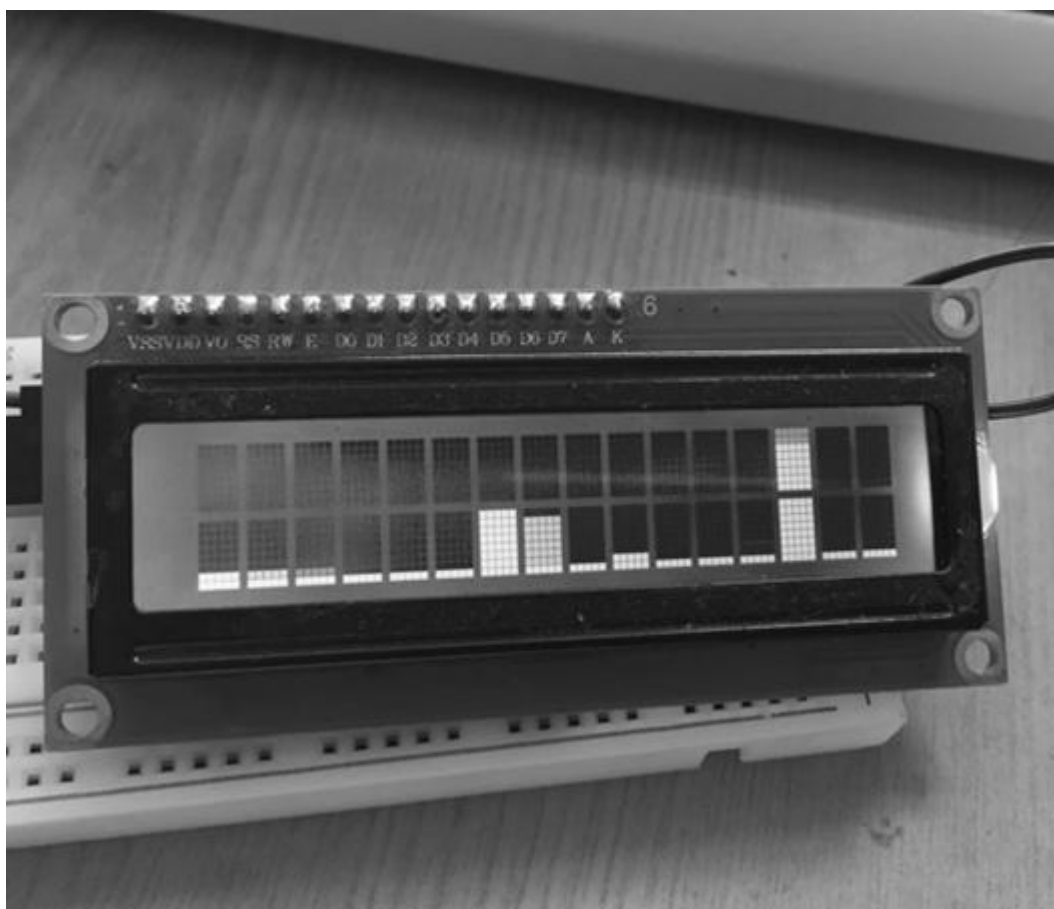


Рисунок 4.14 — Результати тесту №2

На цій частоті спостерігалось більш виражене проявлення шумів, що пов'язано як із фізичними особливостями конструкції пристрою, так і з умовами проведення експерименту. Незважаючи на підвищений рівень шумового фону, прилад зберіг здатність коректно фіксувати основний сигнал.

Третій тест був проведений із використанням музичної композиції (рисунки 4.15 та 4.16). Метою цього тесту було оцінити роботу пристрою в реальних умовах, при обробці складного аудіосигналу з різноманітними частотними компонентами. Результати тесту показали здатність системи коректно аналізувати спектр аудіо, відображаючи основні частотні характеристики композиції.



Рисунок 4.15 — Тест №3

Тестування підтвердило працездатність пристрою в різних режимах роботи, від базового зчитування сигналів до аналізу складних аудіокомпозицій. Незначні шуми, виявлені під час тестів, зумовлені конструктивними особливостями та зовнішніми умовами, проте не впливають суттєво на точність обробки сигналу. Загалом, результати підтверджують ефективність апаратної реалізації та програмного забезпечення для аналізу аудіоспектру.

4.5 Вибір компонентів та їх характеристика

Для розробки обчислювального пристрою аналізатора аудіо спектру було ретельно підібрано комплект основних компонентів, що забезпечують необхідну функціональність, надійність та оптимальну вартість системи.

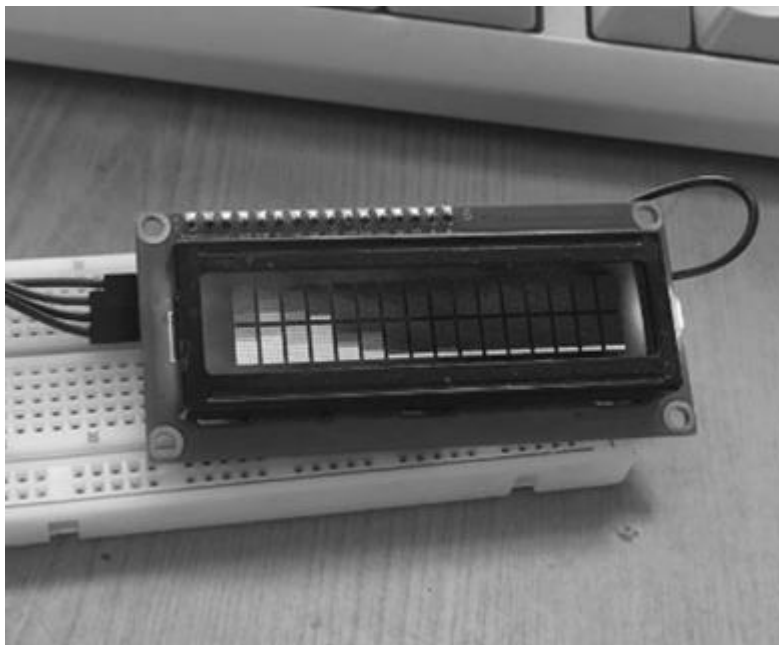


Рисунок 4.16 — Результати тесту №3

Мікроконтролер Arduino Nano (ATmega328P) має наступні характеристики:

- частота роботи — 16 МГц;
- оперативна пам'ять (SRAM) — 2 КБ;
- постійна пам'ять (Flash) — 32 КБ;
- кількість аналогових входів — 8 (10—бітне АЦП);
- компактний форм-фактор (розміри приблизно 45 мм × 18 мм).

Обґрунтування вибору — Arduino Nano (рисунок 4.17) має всі необхідні ресурси для реалізації FFT-аналізу аудіосигналів у режимі реального часу, включаючи достатню обчислювальну потужність, швидкодію та доступ до аналогових входів для зчитування аудіосигналу через мікрофонний підсилювач або інші датчики. На відміну від більш громіздкої Arduino Uno, плата Nano

значно менша за розміром, що дозволяє легко інтегрувати її в компактні конструкції без втрати функціональності.

Крім того, Arduino Nano підтримує стандартну екосистему бібліотек Arduino, у тому числі й бібліотеки для обчислення БПФ (наприклад, ArduinoFFT), що суттєво спрощує програмну реалізацію аналізу сигналу. Навіть з обмеженим обсягом оперативної пам'яті (2 КБ SRAM у версії з мікроконтролером ATmega328), Nano може обробляти вхідні дані з достатньою точністю для побудови базового спектра в діапазоні аудіочастот.

Використання Arduino Nano забезпечує компактність пристрою без втрати обчислювальних можливостей. Вибір компонентів базується на забезпеченні надійної, точної роботи аналізатора з мінімальними розмірами, що важливо для практичного застосування та зручності користування.

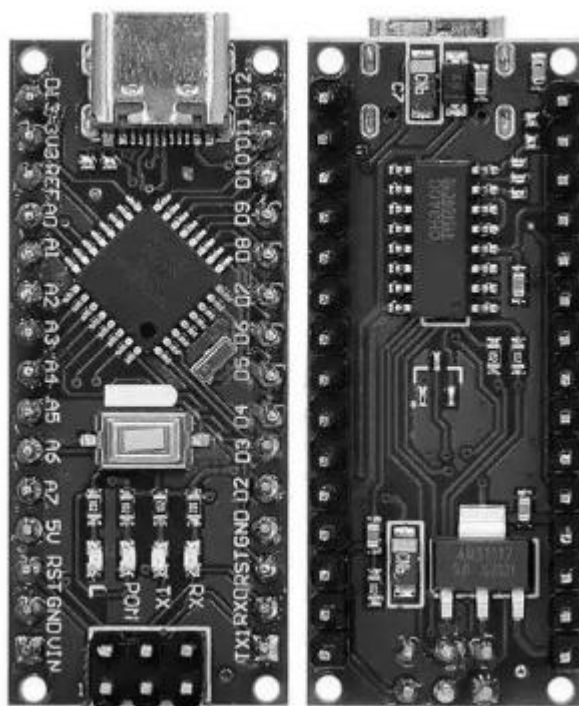


Рисунок 4.17 — Arduino nano v3

Модуль аудіосигналу (вхід AUX) має наступні характеристики:

- діапазон частот — від 20 Гц до 20 кГц;
- вхідний рівень — від 0,1В до 1В RMS.

Обґрунтування вибору — AUX (рисунок 4.18) забезпечує якісний захват аудіосигналу, сумісний з аналоговим входом Arduino, дешева ціна.



Рисунок 4.18 — AUX порт

ЖК-дисплей LCD1602 з інтерфейсом I2C (Рисунок 4.19) має наступні характеристики:

- розмір дисплея — 16 символів на 2 рядки;
- інтерфейс — I2C (два проводи SDA, SCL);
- живлення — 5 В.

Обґрунтування вибору — використання дисплея типу LCD1602 з I2C-інтерфейсом є раціональним рішенням для даного пристрою з кількох технічних і практичних причин. Насамперед, дисплей формату 16×2 (16 символів на 2 рядки) є одним із найпоширеніших та доступних модулів для виводу інформації у вбудованих системах. Його розмір є цілком достатнім для виводу результатів спектрального аналізу в стислому, але зрозумілому вигляді.

Важливою перевагою є наявність I2C-інтерфейсу, який суттєво спрощує підключення дисплея до мікроконтролера. Завдяки цьому дисплей потребує лише два сигнальні проводи (SDA та SCL), що, у свою чергу, зменшує кількість зайнятих пінів на мікроконтролері.

Джерело живлення має наступні характеристики:

- напруга — 5 В;
- струм — до 1 А (з урахуванням запасу).

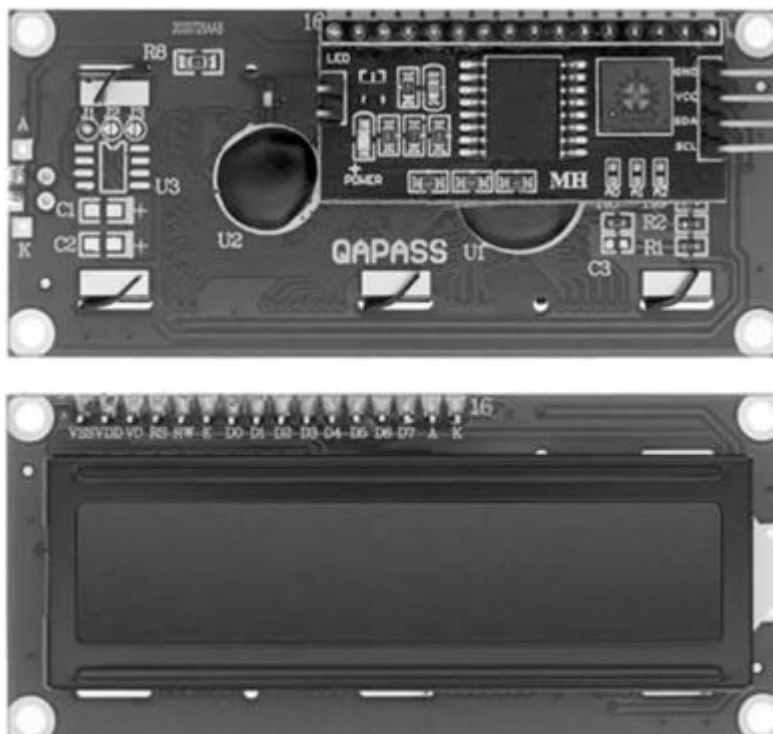


Рисунок 4.19 — LCD 1602 з інтерфейсом I2C/ПС

Обґрунтування вибору—стандартне USB-живлення забезпечує стабільну напругу для мікроконтролера та периферії. Запас струму дозволяє уникнути проблем із нестачею живлення.

Додаткові компоненти: резистори, конденсатори, стабілізатори напруги — для фільтрації сигналів, стабілізації живлення та захисту від перешкод.

Корпус — забезпечує захист електроніки від механічних пошкоджень та пилу.

4.6 Інструкція користувача

4.6.1 Увімкнення та режим роботи пристрою

Підключення пристрою до джерела живлення здійснюється через USB-порт або за допомогою зовнішнього адаптера з напругою 5 В. Після підключення

живлення на платі Arduino Nano загоряється індикаторний світлодіод, що свідчить про успішну подачу напруги та готовність пристрою до роботи.

Після ввімкнення пристрій автоматично переходить у робочий режим — відбувається ініціалізація мікроконтролера, дисплея та всіх необхідних програмних модулів. Без необхідності додаткового втручання з боку користувача запускається алгоритм збору, обробки та аналізу аудіосигналу.

Після запуску пристрій починає прийом звуку через вхід AUX. Отримані дані проходять спектральний аналіз, після чого результати виводяться на LCD-дисплей формату 16×2. Інформація подається у вигляді числових або графічних значень, які змінюються кожні 0.15 секунди, забезпечуючи оновлення інформації майже в реальному часі. Це дозволяє отримати візуальне уявлення про зміну характеристик аудіосигналу — наприклад, зростання або зменшення гучності на певних частотах.

На екрані також відображається поточний рівень гучності сигналу, що надходить, а також його частотний спектр, поданий у вигляді стовпчиків або умовних індикаторів. Така візуалізація дає змогу користувачу оперативно оцінити якість сигналу, виявити наявність шумів, сплесків або переключень у частотному діапазоні.

Для завершення роботи пристрою достатньо відключити його від джерела живлення — через USB або адаптер. Після цього всі процеси автоматично припиняються, і пристрій повертається у неактивний стан до наступного запуску.

4.6.2 Обслуговування, технічні зауваження та рекомендації

Для забезпечення довготривалої та стабільної роботи пристрою рекомендується регулярно здійснювати його базове обслуговування. Зовнішню частину корпусу слід очищати від пилу, бруду та можливих забруднень за допомогою м'якої сухої тканини. У разі накопичення пилу на вентиляційних отворах або поблизу елементів роз'ємів це може призвести до перегріву або зниження якості контактів.

Окрім цього, варто періодично перевіряти цілісність з'єднувальних проводів, контактів та роз'ємів. Наявність пошкоджених або ослаблених з'єднань може стати причиною нестабільної роботи пристрою або повного припинення передачі сигналу. Надійне електричне з'єднання — запорука стабільної передачі даних та коректного функціонування всіх модулів.

Важливим технічним застереженням є необхідність захисту пристрою від потрапляння вологи. Будь-який контакт з рідиною може спричинити коротке замикання, що призведе до виходу з ладу окремих елементів або всієї системи. Також рекомендується використовувати лише рекомендовані джерела живлення з вихідною напругою 5 В та струмом не менше 0,5 А. Невідповідність параметрів живлення може спричинити перегрів, нестабільну роботу або пошкодження електронних компонентів. Пристрій не слід піддавати сильним механічним ударам, вібраціям або падінням — це може пошкодити мікросхеми, дисплей або інші вразливі елементи конструкції.

Для оптимальної експлуатації пристрій бажано розміщувати у приміщеннях із стабільною температурою в межах від +5 °С до +40 °С. Надмірна вологість або перегрів можуть негативно вплинути на стабільність роботи та тривалість служби електронних компонентів. Слід також уникати розміщення пристрою поблизу джерел сильних електромагнітних полів, таких як високовольтне обладнання, трансформатори або радіопередавачі, оскільки це може викликати завади, збої у вимірюваннях або порушення зв'язку між модулями.

ВИСНОВКИ

В рамках даного дипломного проекту було створено ефективний та універсальний аналізатор аудіоспектру, що базується на платформі Arduino.

Його ключові особливості такі:

- необхідність такого пристрою виникає з розширеного попиту на ринку на аналіз звукових сигналів та обробку аудіоданих;
- аналізатор аудіоспектру дозволяє вимірювати та візуалізувати спектральний склад аудіосигналів;
- програмований на Arduino Nano пристрій дозволяє розробляти та впроваджувати нові функції для аналізу звукових сигналів;
- аналізатор живиться від комп'ютера через USB-перехідник, що гарантує безпеку та зручність використання.

У процесі розробки даного аналізатора аудіоспектру було проведено дослідження програмування мікроконтролерів, здійснено експериментальні випробування, вивчено архітектуру мікроконтролерів та їх особливості, а також здійснено порівняння мікропроцесорів і мікроконтролерів, зокрема модулів Arduino.

При розробці даного пристрою, керувалися важливими принципами, використання доступних та економічно вигідних компонентів, простота збірки та налаштування. Крім того, надавалася велика увага надійності роботи, зручності використання для користувача будь—якого віку, естетичному дизайну та актуальності пристрою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arduino Nano V3.0 AVR ATmega328P. URL: <https://arduino.ua/prod166—arduino—nano—v3—0—avr—atmega328p—s—raspayannimi—razemami> (дата звернення: 12.05.2025)
2. Arduino Nano. URL: <https://evo.net.ua/arduino—nano/> (дата звернення: 12.05.2025)
3. LCD—дисплей особливості. URL: <http://teg.com.ua/lcd—displei—osoblivosti—harakteristiki—i—vidgyki/> (дата звернення: 12.05.2025)
4. Коржик М. В. Мікропроцесорна техніка: Комп'ютерний практикум. Київ: КПІ ім. Ігоря Сікорського, 2021. 47 с.
5. Основи аналізу спектра у реальному масштабі часу. Частина 4. URL: https://gtest.com.ua/uk/statti/osnovy—analiza—spektra—v—realnommasshtabe—vremeni—chast_4.html (дата звернення: 12.05.2025)
6. Кириленко О.І. Цікава електроніка: навчальний посібник. Львів: Новий Світ – 2000, 2016. 312 с..
7. Розклад функції в ряд Фур'є. URL: <https://yukhym.com/uk/ryadi—ta—jikh—zbizhnist/rozklad—funktsiji—v—ryad—fur—e.html> (дата звернення: 12.05.2025)
8. Спектроаналізатор — що ми бачимо на ньому. URL: <https://prosound.ixbt.com/education/spektr—analys.shtml> (дата звернення: 12.05.2025)
9. Шидловська Т.І., Ткач Ж.М.. Загальні вимоги до оформлення дипломних та курсових проектів технічних спеціальностей. Методичні вказівки. Вінниця: Вінницький технічний коледж, 2020. 40с.
10. Шликов В. В. Мікропроцесорна техніка: Практикум Київ : КПІ ім. Ігоря Сікорського, 2018. 144 с
11. Horowitz P., Hill W. The Art of Electronics. 3rd ed. Cambridge: Cambridge University Press, 2015.
12. Mazidi M.A., Naimi S., Naimi M. AVR Microcontroller and Embedded Systems: Using Assembly and C. Pearson Education, 2014.

13. ArduinoFFT Library. URL: <https://www.arduino.cc/reference/en/libraries/arduinofft/> (дата звернення: 12.05.2025)
14. Основи цифрової обробки сигналів. URL: <https://mathworld.org.ua/cifrova-obrobka-signaliv/> (дата звернення: 12.05.2025)
15. Спектральний аналіз звукових сигналів. URL: <https://eduportal.com.ua/articles/spectral-audio-analysis.html> (дата звернення: 12.05.2025)
16. Технічна документація Arduino Nano. URL: <https://docs.arduino.cc/hardware/nano> (дата звернення: 12.05.2025)
17. Smith S. W. The Scientist and Engineer's Guide to Digital Signal Processing. 2nd ed. California Technical Publishing, 1999.
18. Принципи роботи FFT. URL: <https://electronics.ua/fft-principles.html> (дата звернення: 12.05.2025)
19. Основи роботи з мікроконтролерами AVR. URL: <https://microavr.org/osnovy-roboty-z-avr/> (дата звернення: 12.05.2025)
20. DSP для початківців. URL: <https://digitalsignal.org.ua/dsp-dlya-pochatkivtsiv/> (дата звернення: 12.05.2025)

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.

21 березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи

«Обчислювальний пристрій аналізатора аудіо спектру»

08-54.БКР.032.00.000 ТЗ

Науковий керівник к.т.н., доцент

Савицька Л.А. _____

Студент групи 2КІ-23мс

Ярський Я.О. _____

1 Підстави для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Актуальність дослідження обумовлена потребою у вдосконаленні засобів аналізу аудіосигналів у режимі реального часу, що зумовлено широким застосуванням аудіоспектрального аналізу в системах технічної діагностики, звукозапису, телекомунікацій та побутової електроніки. Зокрема, важливість дослідження визначається необхідністю створення доступного та ефективного обчислювального пристрою для візуалізації частотного спектра аудіосигналів на базі мікроконтролерних платформ.

1.2 Тема БКР затверджена наказом ректора університету №123 від 09.05.2025 року.

2 Мета БКР і призначення розробки

2.1 Метою проекту є розробка обчислювального пристрою для аналізу аудіоспектра на основі мікроконтролера Arduino Nano з використанням методу швидкого перетворення Фур'є (FFT). Проект передбачає створення ефективного та доступного засобу для реального часу візуалізації частотних характеристик аудіосигналу з аналогового входу з виведенням результатів на LCD-дисплей.

2.2 Призначення розробки полягає у створенні мікропроцесорної системи, здатної здійснювати реальний аналіз спектра аудіосигналу в режимі реального часу. Пристрій повинен забезпечувати обробку сигналу з аналогового аудіовходу, його спектральне розкладання методом FFT та відображення результатів на LCD-дисплеї для подальшого візуального контролю та аналізу.

3 Вихідні дані для виконання БКР

3.1 Проведення аналізу сучасних методів та технологій спектрального аналізу аудіосигналів, що використовуються в побутових, промислових та наукових приладах в Україні та за кордоном.

3.2 Розробка структурної та функціональної схеми мікропроцесорної системи для спектрального аналізу аудіосигналу передбачає побудову логічно

взаємопов'язаної системи, основною метою якої є прийом, обробка та візуалізація параметрів аудіосигналу.

3.3 Здійснення розрахунків та моделювання елементів системи, а також написання мікропрограми для керуючого мікроконтролера.

4 Вимоги до виконання БКР

Головна вимога — функціонування мікропроцесорної системи спектрального аналізатора полягає у використанні методу обробки даних, що базується на порівнянні поточних значень частотних компонентів аудіосигналу з еталонними спектральними шаблонами.

5. Етапи БКР та очікувані результати

5.1 Підготовчий етап — межах дослідження проведено аналіз літературних і наукових джерел, присвячених тематиці спектрального аналізу аудіосигналів. Особливу увагу приділено вивченню сучасних технологій цифрової обробки сигналів, алгоритмів швидкого перетворення Фур'є (FFT), а також мікропроцесорних платформ, які застосовуються для вбудованих систем аудіоаналізу.

5.2 Етап розробки концепції та створення структури та функціональної схеми мікропроцесорної системи. Вибір відповідного мікроконтролера та компонентів. Очікувані результати включають концептуальну модель системи та визначення основних компонентів і їх взаємодії.

5.3 Етап проектування та розробка детальної схеми мікропроцесорної системи, написання програмного забезпечення для мікроконтролера, моделювання та тестування роботи окремих компонентів. Очікувані результати включають готову схему системи та функціональне програмне забезпечення.

5.4 Етап виготовлення прототипу та збірка прототипу системи з вибраних компонентів та інтеграція програмного забезпечення з апаратною частиною. Очікуваний результат — робочий прототип обчислювального аналізатора аудіо спектру.

5.5 Етап тестування, налагодження та проведення серії тестів для перевірки точності та надійності вимірювань, налагодження системи для оптимальної роботи у реальних умовах. Очікувані результати включають тестові звіти та внесення коригувань для покращення роботи системи.

5.6 Етап впровадження, апробації та введення системи в реальні умови експлуатації, збір та аналіз даних для подальшого удосконалення системи. Очікувані результати включають реальні дані та відгуки користувачів.

6 Матеріали, що подаються до захисту БКР

До захисту подаються: пояснювальна записка БКР, графічні та ілюстративні матеріали, протокол попереднього захисту на кафедрі, відгуки наукового керівника та опонента, протоколи складання державних екзаменів, анотації до БКР українською та іноземною мовами.

7 Порядок контролю виконання та захисту БКР

Виконання етапів документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання БКР

8.1 При оформлюванні БКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія».

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК В

ГРАФІЧНА ЧАСТИНА

ОБЧИСЛЮВАЛЬНИЙ ПРИСТРІЙ АНАЛІЗАТОРА АУДІО СПЕКТРУ

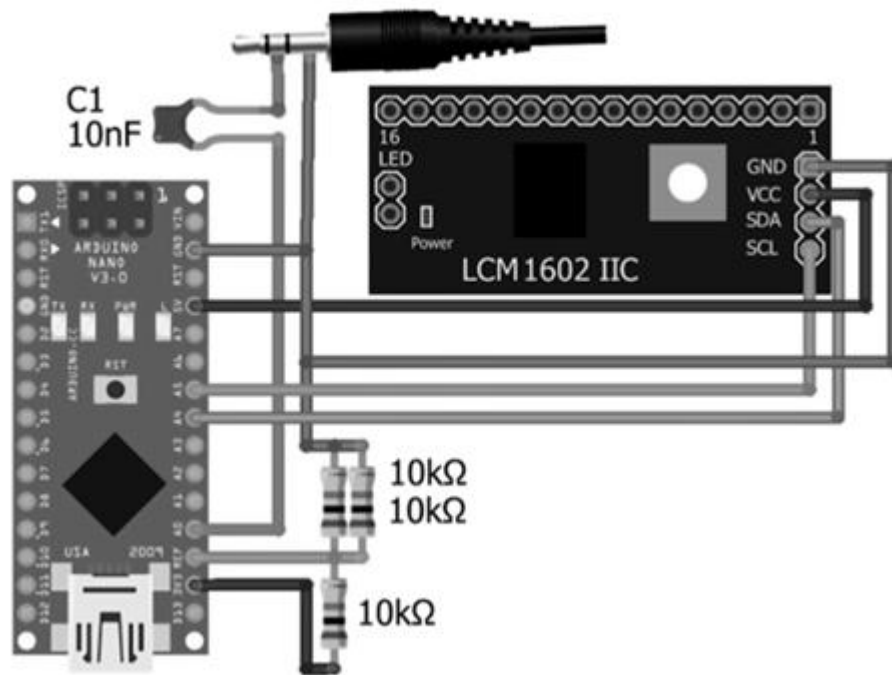


Рисунок В.1 — Функціональна схема пристрою

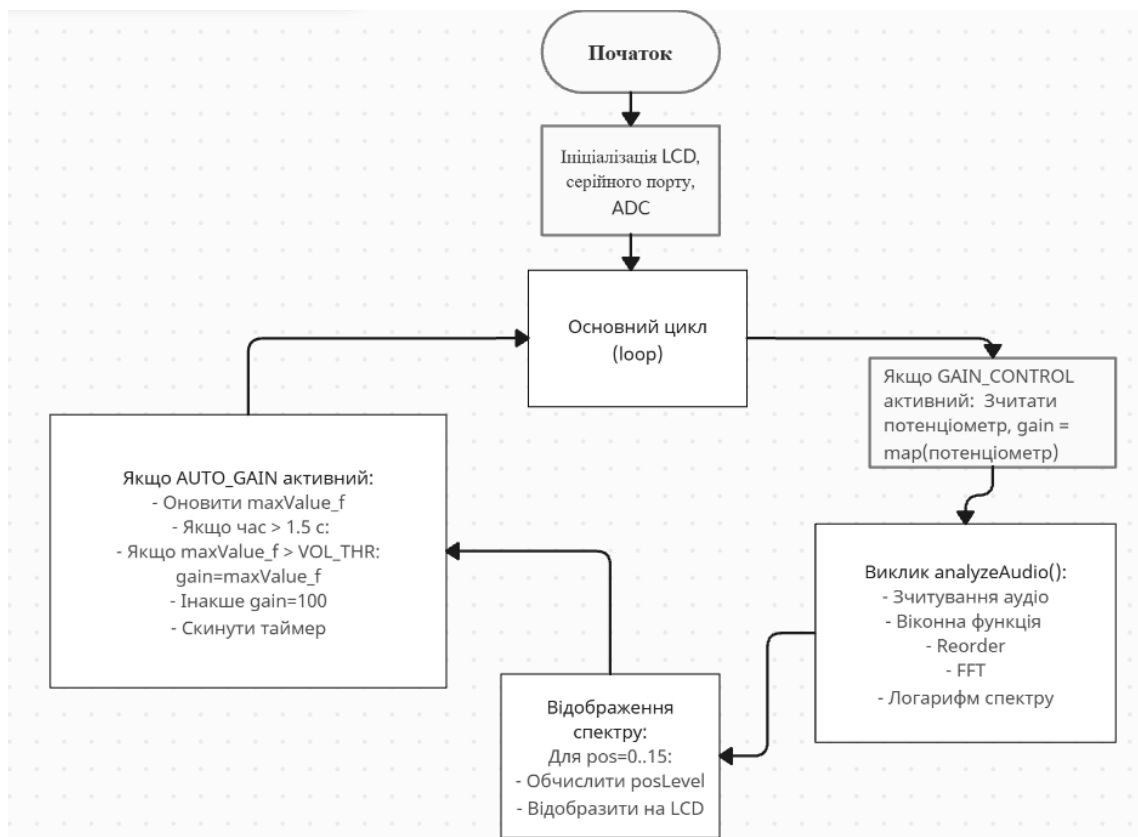


Рисунок В.2 — Блок-схема роботи алгоритму

ДОДАТОК Г

ІЛЮСТРАТИВНА ЧАСТИНА

ОБЧИСЛЮВАЛЬНИЙ ПРИСТРІЙ АНАЛІЗАТОРА АУДІО СПЕКТРУ

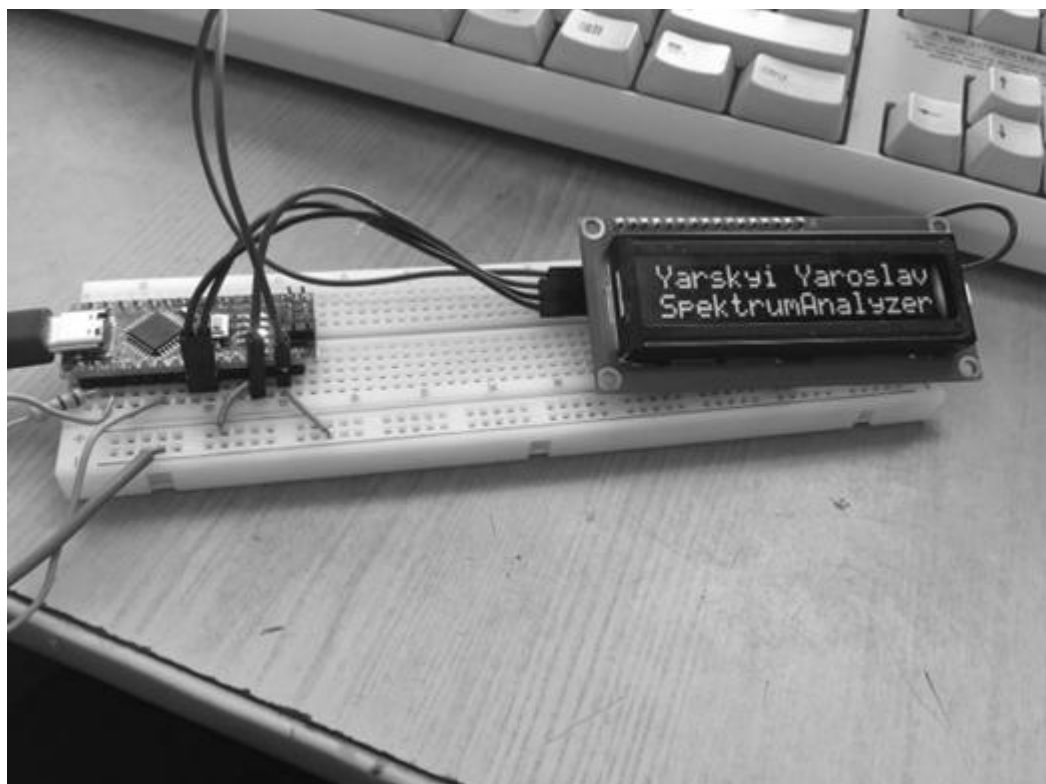


Рисунок Г.1 — Прототип мікропроцесорної системи

ДОДАТОК Д

Лістинг програмного коду

```

#define DRIVER_VERSION 0
#define GAIN_CONTROL
#define AUTO_GAIN
#define VOL_THR 35
#define LOW_PASS 30
#define DEF_GAIN 80
#define FHT_N 256
byte posOffset[16] = {2, 3, 4, 6, 8, 10, 12, 14, 16, 20, 25, 30, 35, 60, 80, 100};
#define AUDIO_IN 0
#define POT_PIN 7
#define LOG_OUT 1
#include <FHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define printByte(args) write(args);
double prevVolts = 100.0;
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#if (DRIVER_VERSION)
LiquidCrystal_I2C lcd(0x27, 16, 2);
#else
LiquidCrystal_I2C lcd(0x3f, 16, 2);
#endif
byte v1[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000,
0b00000, 0b11111};
byte v2[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000,
0b11111, 0b11111};
byte v3[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111,
0b11111, 0b11111};
byte v4[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111,
0b11111, 0b11111};
byte v5[8] = {0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
byte v6[8] = {0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
byte v7[8] = {0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
byte v8[8] = {0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111,
0b11111, 0b11111};
byte gain = DEF_GAIN;
unsigned long gainTimer;
byte maxValue, maxValue_f;
float k = 0.1;

```

```

void setup() {
  sbi(ADCSRA, ADPS2);
  cbi(ADCSRA, ADPS1);
  sbi(ADCSRA, ADPS0);
  analogReference(EXTERNAL);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Yarskyi Yaroslav");
  lcd.setCursor(0, 1);
  lcd.print("SpektrumAnalyzer");
  lcdChars();
  delay(2000);
}

void loop() {
  if (GAIN_CONTROL) gain = map(analogRead(POT_PIN), 0, 1023, 0, 150);

  analyzeAudio();
  for (int pos = 0; pos < 16; pos++) {
    if (fht_log_out[posOffset[pos]] > maxValue) maxValue =
fht_log_out[posOffset[pos]];

    lcd.setCursor(pos, 0);

    int posLevel = map(fht_log_out[posOffset[pos]], LOW_PASS, gain, 0, 15);
    posLevel = constrain(posLevel, 0, 15);
    if (posLevel > 7) {
      lcd.printByte(posLevel — 8);
      lcd.setCursor(pos, 1);
      lcd.printByte(7);
    } else {
      lcd.print(" ");
      lcd.setCursor(pos, 1);
      lcd.printByte(posLevel);
    }
  }
}

if (AUTO_GAIN) {
  maxValue_f = maxValue * k + maxValue_f * (1 — k);
  if (millis() — gainTimer > 1500) {

    if (maxValue_f > VOL_THR) gain = maxValue_f;
    else gain = 100;
    gainTimer = millis();
  }
}

```

```

    }
  }
}

void lcdChars() {
  lcd.createChar(0, v1);
  lcd.createChar(1, v2);
  lcd.createChar(2, v3);
  lcd.createChar(3, v4);
  lcd.createChar(4, v5);
  lcd.createChar(5, v6);
  lcd.createChar(6, v7);
  lcd.createChar(7, v8);
}

void analyzeAudio() {
  for (int i = 0 ; i < FHT_N ; i++) {
    int sample = analogRead(AUDIO_IN);
    fht_input[i] = sample; // put real data into bins
  }
  fht_window();
  fht_reorder();
  fht_run();
  fht_mag_log();
}

```