

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет робототехніки та приладобудування
Комп'ютерно-інтегрованих оптичних та
навігаційних систем**

До захисту допущено:
Завідувач кафедри
_____ Надія БУРАУ
«___» _____ 20__ р.

**Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерно-інтегровані
системи та технології в приладобудуванні»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані
технології»
на тему: «Автоматизована система визначення рівня шуму в
приміщенні»**

Виконав:

студент IV курсу, групи ПГ-21

Левчук Олександр Сергійович _____

Керівник:

Цибульник Сергій Олексійович, _____

к.т.н., доцент _____

Рецензент:

асистент, PhD кафедри КІТВП _____

Бондарєв Д.В. _____

Засвідчую, що у цій дипломній
роботі немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2026 р.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет робототехніки та приладобудування
Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем
Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»
Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувачка кафедри

_____ Надія БУРАУ
«___» _____ 2026 р.

ЗАВДАННЯ

на дипломну роботу студенту
Левчуку Олександр Сергійовичу

1. Тема роботи «Автоматизована система визначення рівня шуму в приміщенні», керівник роботи Цибульник Сергій Олексійович, к.т.н., доцент, затверджені наказом по університету від «___» _____ 20__ р. № _____
2. Термін подання студентом роботи 05.06.2026р.
3. Вихідні дані до роботи: мінімальний рівень шуму для вимірювання – 30 дБ; максимальний рівень шуму для вимірювання – 100 дБ; чутливість датчика – не менше 3 дБ; частота оновлення даних графічного інтерфейсу користувача – 2 Гц.
4. Зміст роботи:
 - 1) Огляд стану проблеми.
 - 2) Огляд та вибір елементної бази.
 - 3) Проектування елементів складальної одиниці.
 - 4) Розроблення графічного інтерфейсу користувача.
 - 5) Апаратне та програмне налагодження системи.
 - 6) Перевірка працездатності системи та аналіз результатів.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): таблиці, графіки, рисунки, тощо.
6. Дата видачі завдання 16.03.2026р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
	Огляд стану проблеми	23.03.2026	
	Огляд та вибір елементної бази	30.03.2026	
	Проектування елементів складальної одиниці	06.04.2026	
	Розроблення графічного інтерфейсу користувача	20.04.2026	
	Апаратне та програмне налагодження системи	18.05.2026	

	Перевірка працездатності системи та аналіз результатів	25.05.2026	
	Оформлення пояснювальної записки	05.06.2026	

Студент

Олександр ЛЕВЧУК

Керівник

Сергій ЦИБУЛЬНИК

АНОТАЦІЯ

У кваліфікаційній роботі розглянуто питання розроблення автоматизованої системи для визначення рівня шуму в приміщенні, робота виконана з урахуванням вимог до швидкості та точності вимірювання.

У першому розділі розглянуто історію створення засобів для вимірювання рівня звуку, проведено огляд існуючих рішень, принципів вимірювання рівня технічного шуму та класифікацію приладів для вимірювання шуму. Також було підібрано елементну базу для реалізації власної автоматизованої системи для вимірювання рівня шуму.

У другому розділі було розглянуто принципи роботи мікрофону, інтерфейсів передачі даних, спроектовано елементи складальної одиниці у вигляді плати та корпусу пристрою, розроблено графічний інтерфейс користувача та проведено апаратне і програмне налагодження системи. Також було розглянуто алгоритми обробки та перетворення сигналів.

Основну увагу в роботі приділено проєктуванню та розробці пристрою для вимірювання рівня шуму, реалізації алгоритму вимірювання, обробки і використання вимірянних даних та перевірки працездатності пристрою.

Ключові слова: шумомір, автоматизована система, мікроконтролер, комутація, алгоритм, звук.

ABSTRACT

The qualification work considers the issue of developing an automated system for determining the noise level in a room, the work was carried out taking into account the requirements for the speed and accuracy of measurement.

The first section considers the history of the creation of sound level measuring devices, reviews existing solutions, principles of measuring the level of technical noise and classification of noise measuring devices. Also, an element base was selected for the implementation of our own automated system for measuring the noise level.

The second section considers the principles of operation of the microphone, data transmission interfaces, designs the elements of the assembly unit in the form of a board and a device housing, develops a graphical user interface and performs hardware and software debugging of the system. Signal processing and conversion algorithms are also considered.

The main attention in the work is paid to the design and development of a device for measuring the noise level, implementation of the measurement algorithm, processing and use of measured data and testing the device's performance.

Keywords: sound level meter, automated system, microcontroller, switching, algorithm, sound.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ОГЛЯД ПРОБЛЕМИ НАДМІРНОГО РІВНЯ ШУМУ ТА МЕТОДІВ ЙОГО ВИМІРЮВАННЯ	9
1.1 Історія винайдення засобів для вимірювання рівня шуму.....	9
1.2 Огляд актуальності проблеми надмірного рівня шуму та його вимірювання	10
1.3 Класифікація шумомірів за методом вимірювання.....	11
1.4 Класифікація шумомірів за принципом роботи	13
1.5 Огляд аналогічних рішень	16
1.6 Вибір елементної бази	18
РОЗДІЛ 2. РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИМІРЮВАННЯ РІВНЯ ШУМУ	29
2.1 Проєктування елементів складальної одиниці	29
2.2 Розроблення графічного інтерфейсу користувача.....	35
2.3 Апаратне та програмне налагодження системи.....	40
2.4 Перевірка працездатності системи та аналіз результатів.....	59
2.5 Висновки до другого розділу.....	64
ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТОК А	75

ВСТУП

В умовах зростаючих вимог до контролю норм, безпеки, комфорту або технічної діагностики особливої уваги здобуває питання вимірювання рівня шуму в певних процесах.

Зростання рівня техногенного навантаження та інтенсивний розвиток міської інфраструктури супроводжуються збільшенням рівня шуму, що спричиняє акустичний вплив на людину. Надмірний шум, що присутній у житлових приміщеннях, громадських і виробничих підприємствах є фактором, що негативно впливає на самопочуття, працездатність та загальний стан здоров'я [1]. У зв'язку з цим, питання контролю рівня шуму в навколишньому середовищі набуває практичного значення у побутовій і у професійній сферах.

Існуючі засоби вимірювання рівня шуму дають можливість отримувати достовірні результати, однак вони не завжди підходять по вартості, гнучкості використання або можливості інтеграції з іншими системами. Це створює необхідність розроблення спеціалізованого пристрою, що буде орієнтованим на конкретні умови застосування та побудований на базі сучасних елементів.

Саме тому метою даної роботи є створення автоматизованої системи для визначення рівня шуму в приміщенні з урахуванням вимог до точності отриманих результатів та швидкості їх вимірювання.

Для досягнення даної цілі було проведено аналіз вже існуючих підходів до вимірювання шуму, розглянуто особливості частотного зважування та методів оцінювання акустичних параметрів, виконано проєктування структурної та електричної схем пристрою, підібрано алгоритм, що використовуватиметься системою для вимірювання та обробки сигналу, а також здійснено калібрування розробленої автоматизованої системи з подальшою перевіркою точності отриманих результатів.

При виконанні роботи було застосовано аналітичні методи опрацювання науково-технічних джерел, а також експериментальні методи для перевірки характеристик створеної автоматизованої системи. Практична цінність полягає у можливості використання для контролю акустичних умов у приміщеннях різного

призначення, таких як на виробництві для оцінки умов роботи працівників та визначенню потреби у використанні засобів захисту від надмірного рівня шуму чи у житлових і громадських приміщеннях для оцінки акустичного комфорту. Також дана автоматизована система може використовуватись для тестування шумоізоляції транспортних засобів, перевірки рівня зашумленості довкілля, вимірювання якості звуку в приміщеннях та наукових досліджень.

РОЗДІЛ 1

ОГЛЯД ПРОБЛЕМИ НАДМІРНОГО РІВНЯ ШУМУ ТА МЕТОДІВ ЙОГО ВИМІРЮВАННЯ

1.1 Історія винайдення засобів для вимірювання рівня шуму

Шумомір – це прилад для об'єктивного вимірювання рівня звуку [2]. Фактично шумомір складається з мікрофону, до якого під'єднаний вольтметр. Оскільки електричний сигнал на виході з мікрофону пропорційний до вихідного звукового сигналу, то приріст рівня звукового тиску, який впливає на мембрану мікрофона викликати відповідний приріст напруги електричного струму на вході до вольтметра, що і відображається за допомогою іншого індикаторного пристрою, проградуйованого в децибелах [2].

Історія створення шумомірів бере свій початок ще в далекому 17 столітті, коли вчені вперше почали спроби визначити фізичні характеристики звуку [3]. Протягом великої кількості часу звук вважався одним із явищ, які неможливо точно виміряти, проте розвиток фізики, механіки та електроніки поступово привів до створення шумомірів.

Одними з перших дослідженнями швидкості звуку займалися П'єр Гассенді та Марен Мерсенн з Франції у 17 столітті [4]. Вони вимірювали час між спалахом вибуху та моментом, коли спостерігач міг почути його на певній дистанції [5]. Ісаак Ньютон був першим, хто, хоча і неточно, описав швидкість звуку. У 18 столітті Французька академія наук провела більш точні вимірювання, отримавши результати, що є набагато ближчими до сучасних значень [3, 6].

Одним із важливих етапів було дослідження частоти звуку. Галілео Галілей помітив, що у звуку є так званий тон [7], а Роберт Гук пізніше створив пристрій, який дозволяв генерувати звуки різної частоти. Пізніше, у 19 столітті Фелікс Савар встановив зв'язок між висотою тону та частотою коливань звуку [8].

Незабаром Жюль Ліссажу створив механічний осцилограф, який дозволяв спостерігати співвідношення частоти, амплітуди та фази звукових коливань, що стало основою для подальшого розвитку акустичних вимірювальних систем [9].

Однак найбільшою проблемою того часу було вимірювання амплітуди звуку, тобто визначення його гучності [6]. У 1882 році британській фізик Джон Вільям Стретт створив диск Релея, який реагував на рух повітря під дією звукової хвилі. Хоча цей прилад і був дуже неточним, але він став ще одним важливим кроком до створення шумомірів [6, 10]. Наприкінці 19 століття почалася електрична ера акустичних вимірювань. Томас Едіссон та Еміль Берлінер майже одночасно винайшли вугільний мікрофон, який перетворював звукові коливання на електричні сигнали [6, 11]. В цей ж період Жак-Арсен д'Арсонваль створив гальванометр – пристрій для вимірювання слабких електричних струмів [10, 12]. Після чого американський вчений Джордж Пірс поєднав мікрофон з гальванометром і вперше в історії здійснив електричне вимірювання звуку.

Вже в 1917 році інженери компанії AT&T розробили перший електронний вимірювач рівня звуку, який поєднував у собі гальванометр, термофон, вакуумну лампу-підсилювач та винайдений Едвардом Венте у 1916 році конденсаторний мікрофон [13]. Використання вакуумних ламп дозволило підсилювати слабкі електричні сигнали, що зробило можливим створення практичних шумомірів.

Подальший розвиток електроніки дуже сильно вдосконалив шумовимірювальну техніку. Коли у 1947 році винайшли транзистори, прилади стали компактнішими, надійнішими та точнішими, а у другій половині 20 століття їх почали оснащувати цифровими схемами, мікроконтролерами та системами автоматичної обробки сигналів, що дозволило отримати рівень точності, схожий на сучасні шумоміри [2, 14].

1.2 Огляд актуальності проблеми надмірного рівня шуму та його вимірювання

У сучасних реаліях проблема шумового забруднення в навколишньому середовищі є актуальною проблемою нашого суспільства. Люди знаходяться під його впливом майже завжди, на роботі, на вулиці, слухаючи голосну музику чи відвідуючи певні розважальні заходи чи місця великого скупчення людей.

Щоб спричинити тимчасову або навіть постійну шкоду для людського слухового апарату, достатньо тривалого впливу шуму в 85 дБ. Цей рівень часто спостерігається на вулицях з дуже активним транспортним рухом чи у промислових зонах [15]. Деякі двигуни, як-от в компресорах, можуть досягати рівня до 100 дБ, а рівні звуку на концертах можуть досягати до 110-120 дБ. Це є причиною, чому під час планування міст їх розділяють на зони, наприклад, приватний сектор чи промислова зона [15].

Для регулювання шумового забруднення Міністерством охорони здоров'я України було затверджено державні санітарні норми допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови [16]. Наприклад, для житлових приміщень вдень дозволений діапазон зазвичай сягає від 25 до 40 дБ, а вночі від 20 до 30 дБ. В торговельних залах магазинів, пасажирських залах аеропортів і вокзалів, приймальних пунктах підприємств побутового обслуговування дозволений рівень цілодобово сягає до 60 дБ [16]. Для контролю дотримання цих правил використовують шумоміри.

1.3 Класифікація шумомірів за методом вимірювання

За методами вимірювання шумоміри можна класифікувати на 3 основних типи [17]: шумоміри прямого вимірювання, інтегруючі шумоміри та дозиметричні шумоміри.

1.3.1 Шумоміри прямого вимірювання

Це найпоширеніший тип шумомірів, який вимірює поточний рівень звуку в реальному часі [18]. Його принцип роботи полягає у перетворенні мікрофоном акустичних коливань в електричний сигнал, який підсилюється, перетворюється в логарифмічну величину та відображається як числове значення в дБ.

Їх основною особливістю є те, що вони відразу відображають рівень шуму в даний момент, що робить їх зручними для швидкого вимірювання поточного рівня шумового забруднення, наприклад у приміщеннях, біля обладнання або на

вулиці. Проте вони не підходять для оцінки довготривалого впливу на людину, бо не враховують тривалість впливу шуму.

1.3.2 Інтегруючі шумоміри

Інтегруючі шумоміри, на відміну від прямих, мають можливість вимірювати середній рівень шуму протягом певного часу [19]. Основним їх показником є не рівень звуку в дБ, а еквівалентний рівень шуму L_{Aeq} , який враховує зміну звуку в часі та дає узагальнену оцінку шуму.

Такі прилади найчастіше використовують спеціальні методи інтегрування сигналу, що і дозволяє “накопичувати” енергію звуку за весь період вимірювання. Це використовується в екологічних дослідженнях і на виробництвах, де шум має змінний характер, щоб оцінити відповідність санітарним нормам.

1.3.4 Дозиметричні шумоміри

Шумоміри дозиметричного типу призначені для того, щоб контролювати вплив шуму на окрему одиницю, наприклад, людину [19]. Найчастіше вони закріплюються на одязі працівника і безперервно вимірюють рівень шуму, якому той піддається. Після завершення роботи за допомогою приладу можна побачити загальну дозу шумового навантаження, отриману за день (рис.1.1, [62]).

🔍 TWA CALCULATION EXAMPLE

A worker is exposed to:

- 95 dBA for 2 hours
- 90 dBA for 3 hours
- 85 dBA for 3 hours

Using OSHA's formula: $TWA = 16.61 \times \log(D/100) + 90$

Where D = noise dose percentage calculated from each exposure segment against OSHA's permissible duration table.

At 95 dBA: OSHA allows 4 hours → $2 \text{ hours} \div 4 \text{ hours} = 50\%$

At 90 dBA: OSHA allows 8 hours → $3 \div 8 = 37.5\%$

At 85 dBA: OSHA allows 16 hours → $3 \div 16 = 18.75\%$

Total D = $50 + 37.5 + 18.75 = 106.25\%$

This exceeds 100% — mandatory hearing protection and [audiometric testing](#) required.

Рисунок 1.1 – Приклад розрахунку дози шуму

Основна перевага цих пристроїв полягає в тому, що вони дають змогу вирахувати вплив шуму на окрему людину в динамічних умовах, тому часто використовуються в промисловості, авіації, будівництві та інших сферах, де наявне підвищене шумове навантаження для певних працівників. Результати таких вимірювань дають змогу оцінити ризики втрати слуху та дотримання норм охорони праці.

1.3 Класифікація шумомірів за принципом роботи

З розвитком історії змінювались і принципи роботи шумомірів, поступово замінюючи неточні та інертні механічні на швидкі, точні та компактні цифрові шумоміри. Загалом їх можна поділити на три категорії [6]: механічні, електромеханічні та цифрові шумоміри.

1.4.1 Механічні шумоміри

Одними з найперших пристроїв для оцінки звуку є механічні шумоміри [6]. Найчастіше вони складались з мембран або пластин, які коливались під дією звукового тиску, передаючи цей рух на стрілку або інший індикатор, що і давало змогу виміряти силу звуку (рис. 1.2, [63]).

Головним їхнім недоліком були низька точність та дуже велика інерційність. Вони не могли вимірювати швидкі зміни звуку та не дозволяли робити заміри стандартизовано в децибелах.

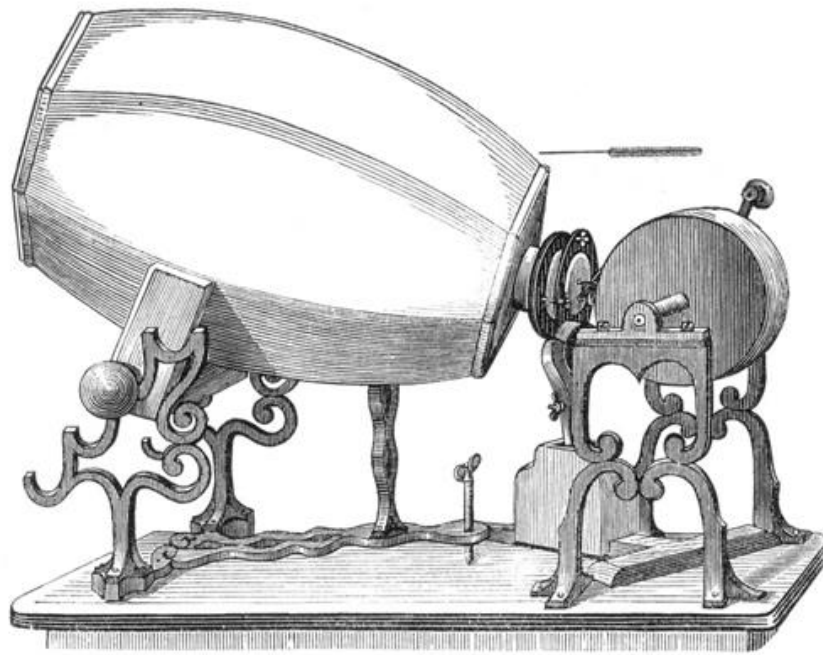


Рисунок 1.2 – Пристрій для запису коливань звуку

1.4.2 Електромеханічні шумоміри

Електромеханічні (або аналогові) шумоміри вимірювали звук за допомогою мікрофону, який перетворював вхідний шум на електричний сигнал, який після цього оброблявся за допомогою аналогових схем (підсилювачів та фільтрів) та керував стрілочним вимірювальним механізмом (рис. 1.3, [64]) [2].



Рисунок 1.3 – Електромеханічний шумомір ВШВ-003-М2

Така схема давала можливість значно підвищити точність у порівнянні з механічними шумомірами та активно використовувалась до 1970-х років. Цей тип приладів міг використовувати стандартизовані шкали в децибелах та частотні корекції, але могли вимірювати тільки рівень шуму в реальному часі.

1.4.3 Цифрові шумоміри

Сучасні прилади вимірювання звуку повністю базуються на цифрових технологіях. Звук спочатку проходить через мікрофон, де він перетворюється у електричний сигнал, після чого він обробляється через аналого-цифровий перетворювач, а далі йде на мікроконтролер, який вже обробляє цю інформацію та виводить або зберігає її (рис. 1.4, [65]) [20].



Рисунок 1.4 – Цифровий шумомір BENETECH GM1357

Основною перевагою цього типу приладів є можливість не просто вимірювати рівень шуму, а будувати графіки, проводити спектральний аналіз, зберігати дані вимірювань для подальшої роботи з ними на інших пристроях та багато чого іншого. Зараз саме вони є стандартом у акустиці, промислових вимірюваннях та екологічному контролі.

1.5 Огляд аналогічних рішень

Зараз існує велика кількість різноманітних шумомірів, які відрізняються між собою не тільки точністю та функціоналом, але і сферою застосування. В той час як одні застосовуються для побутових задач, інші – для професійної акустики, промисловості або моніторингу екологічного стану довкілля. Нижче наведено деякі з сучасних моделей, які використовуються в різних цілях та мають різні можливості.

1.5.1 Tenmars TM-102

Це професійний шумомір середнього класу (рис. 1.5, [66]), що має діапазон вимірювань від 30 до 130 дБ. Найчастіше використовується на виробництвах та в охороні праці для перевірки відповідності санітарним нормам [21].

Основною перевагою над побутовими шумомірами є наявність А/С-корекції, режиму FAST/SLOW та вищої точності вимірювань.



Рисунок 1.5 - Шумомір Tenmars TM-102

1.5.2 Phonic PAA 3X

Цей пристрій це вже не звичайний шумомір, а портативний аналізатор (рис. 1.6, [67]). Він може виконувати різноманітні функції, такі як спектральний аналіз, аналіз акустики приміщення, перевірка частотних характеристик аудіосистем. Такі пристрої використовують при налаштуванні акустики в студіях звукозапису, концертних залах, тощо [22].

Вони значно складніші ніж звичайні шумоміри та дозволяють аналізувати структуру шуму по частотах. Проте, як і всяке інше професійне обладнання, мають доволі високу ціну.



Рисунок 1.6 - Спектральний аналізатор Phonic PAA 3X

1.5.3 Lutron DS-2013SD

Це професійний шумовий дозиметр із можливістю запису даних на SD-карту (рис. 1.7, [68]). Він дає можливість не просто вимірювати рівень шуму, а

проводити тривалий моніторинг шумового впливу. Найчастіше такі пристрої використовують при атестації робочих місць, оцінці умов праці та на промислових підприємствах [23].

Хоча це і шумомір, але області його використання дуже відрізняються від інших шумомірів, бо його задача це виміряти загальний вплив шуму на якусь одиницю (найчастіше людину) за певний відрізок часу.



Рисунок 1.7 - Шумовий дозиметр Lutron DS-2013SD

1.6 Вибір елементної бази

Для створення власного автоматизованого пристрою для вимірювання рівня шуму важливо підібрати правильну елементну базу, яка сприятиме швидкодії системи, невеликому енергоспоживанню та задовільній точності вимірювань. При підборі елементів враховувалися їхні технічні параметри, доступність, простота інтеграції та можливість в подальшому модернізувати систему.

1.6.1 Вибір елемента керування

В якості основного компонента керування необхідно обрати мікроконтролер, який має достатні обчислювальні можливості, вбудовані модулі обробки, наприклад, аналогово-цифровий перетворювач чи стабілізатор напруги та необхідні інтерфейси підключення. Основними претендентами на цю роль є розробницькі плати на основі мікроконтролерів ESP32, STM32 та плати Arduino.

Апаратні обчислювальні платформи для аматорського конструювання Arduino (рис. 1.8, [69]) широко використовуються для розробки електронних пристроїв та прототипування завдяки простоті використання та великій кількості доступних бібліотек [24]. Його використання дозволяє швидко реалізувати базову взаємодію між компонентами системи без необхідності глибокого налаштування апаратної частини. Наприклад, наявність цифрових і аналогових входів забезпечує можливість підключення різноманітних датчиків та модулів, підтримка інтерфейсів UART, SPI та I2C дозволяє організувати обмін даними з периферійними пристроями, а велика кількість готових програмних рішень значно скорочує час розробки [25].

Крім того, Arduino має зручне програмне середовище, що спрощує створення та налагодження програмного забезпечення. Проте обчислювальні можливості Arduino є обмеженими, особливо при виконанні ресурсоємних операцій цифрової обробки сигналів, що може ускладнювати реалізацію складних алгоритмів аналізу аудіоданих. Незважаючи на невисоку вартість та простоту використання, можливості даної платформи можуть бути недостатніми для реалізації пристрою шумоміра з розширеним функціоналом.

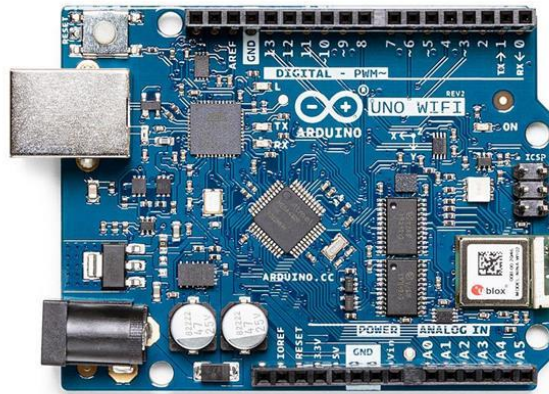


Рисунок 1.8 – Розробницька плата Arduino UNO

STM32 (рис. 1.9, [70]) - це сімейство високопродуктивних мікроконтролерів на основі архітектури ARM, має значні обчислювальні можливості та широкий набір вбудованих периферійних модулів. Його використання дозволяє реалізовувати складні алгоритми обробки сигналів та виконувати задачі в режимі реального часу.

Ці мікроконтролери оснащені високоточними аналого-цифровими перетворювачами, таймерами, модулями DMA для швидкої передачі даних без навантаження на процесор та підтримують інтерфейси UART, SPI, I2C, I2S і USB. Деякі моделі також мають апаратні блоки для математичних обчислень і цифрової обробки сигналів, що значно підвищує продуктивність при роботі з аудіоданими. Крім того, підтримується використання операційних систем реального часу, що дозволяє ефективно розподіляти задачі між процесами.

Проте налаштування STM32 та розробка програмного забезпечення для нього є складнішими порівняно з Arduino або ESP32 та потребують більш детального розуміння апаратної архітектури. Незважаючи на це, висока продуктивність і гнучкість роблять STM32 ефективним рішенням для створення вимірювальних систем і пристроїв цифрової обробки сигналів [26].

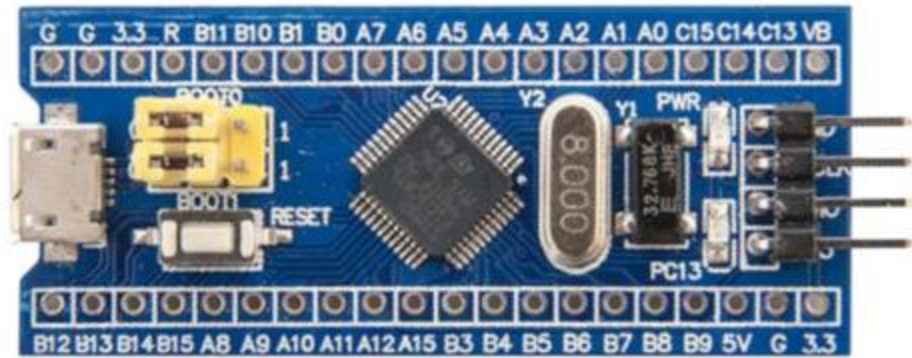


Рисунок 1.9 – Плата розробки STM32

ESP32 (рис. 1.10, [71]), який має достатню обчислювальну потужність для обробки сигналу та подальшої взаємодії з ним. Його використання дає багато можливостей для обробки, взаємодії та інтеграції з іншими елементами системи. Наприклад, наявність бездротових інтерфейсів (WiFi, Bluetooth) дають можливість взаємодіяти з пристроями дистанційно або через веб-інтерфейс, наявний режим deep sleep може значно знизити споживання заряду елементу живлення, вбудовані аналого-цифровий перетворювач та I2S спрощують взаємодію з іншими елементами, наявний RTOS може виконувати задачі в режимі реального часу та навіть обробляти задачі паралельно завдяки наявності двох ядер. Всі ці переваги в поєднанні з невеликою ціною та простотою у використанні роблять ESP32 очевидним вибором для створення пристрою [27].

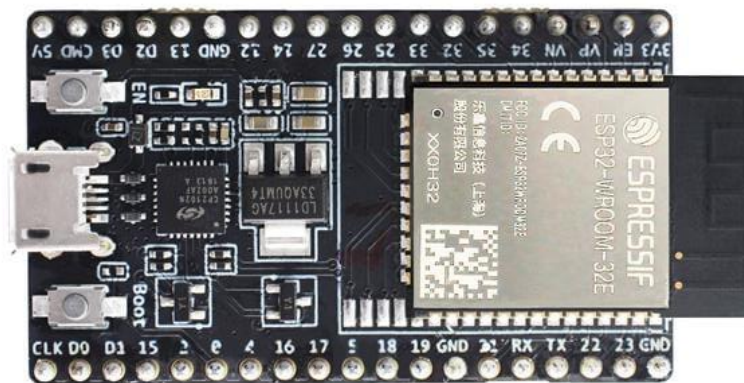


Рисунок 1.10 – Плата розробки ESP32

1.6.2 Вибір мікрофона

Для приймання звукового сигналу необхідно обрати мікрофон, який задовільнить вимоги до точності та діапазону вимірювання звуку та буде простим у підключенні та використанні. Найкращим варіантом є вибір готового модуля з розпаяним на ньому мікрофоном, що значно полегшить задачу його встановлення у готовий пристрій.

Модуль з цифровим MEMS-мікрофоном INMP441 та підключенням через інтерфейс I2S (рис. 1.11, [72]) є доцільним для систем вимірювання та обробки звукових сигналів завдяки поєднанню компактності, високої точності та простоти інтеграції з мікроконтролерними системами. Наприклад, мікрофон містить у своєму складі MEMS-чутливий елемент, вбудований аналого-цифровий перетворювач та цифрові схеми обробки сигналу, що дозволяє зменшити кількість зовнішніх компонентів і спростити схему пристрою. Підтримка інтерфейсу I2S забезпечує передачу вже оцифрованого сигналу без необхідності використання додаткових зовнішніх аналого-цифрових перетворювачів та зменшує вплив електромагнітних завад, характерних для аналогових ліній передачі.

Його робочий частотний діапазон становить 60 Гц – 15 кГц, що забезпечує можливість реєстрації більшості звукових сигналів у навколишньому середовищі. Максимальний рівень звукового тиску складає 120 дБ SPL, що дозволяє працювати з широким діапазоном інтенсивності звуку без значних спотворень. Номінальна чутливість становить –26 дБFS, а допустиме відхилення знаходиться в межах –29...–23 дБFS. Співвідношення сигнал/шум для даного мікрофона становить 61 дБА, що є достатнім для реалізації систем вимірювання рівня шуму [28].

Поєднання функціональних можливостей, простоти використання та невисокої вартості робить INMP441 ефективним рішенням для реалізації шумоміра.

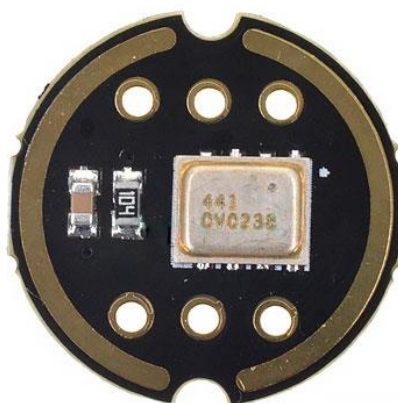


Рисунок 1.11 – Модуль з мікрофоном INMP441

Ще одним варіантом є модуль з цифровим MEMS-мікрофоном ICS 43434 та підключенням I2S (рис. 1.12, [73]). Його перевагами є його всеспрямованість, що дозволяє вимірювати звук в усіх напрямках, вбудовані MEMS-елемент, ADC та фільтри всередині корпусу, що дозволяє відкинути використання окремих фільтрів і аналого-цифрових перетворювачів та підтримка підключення по I2S, що значно спрощує його інтеграцію в пристрій.

Мікрофон ICS43434 також характеризується достатніми технічними параметрами для використання в системах вимірювання шуму. Згідно з технічними характеристиками, його робочий частотний діапазон становить від 60 Гц до 20 кГц, що охоплює основний спектр частот, які сприймаються людським слухом. Максимальний рівень акустичного навантаження становить 120 дБ SPL, що дозволяє використовувати його для вимірювання як побутових, так і виробничих шумів. Номінальна чутливість мікрофона складає $-26 \text{ dBFS} \pm 1 \text{ дБ}$, при цьому відхилення $\pm 1 \text{ дБ}$ можна розглядати як похибку між окремими екземплярами датчика. Співвідношення сигнал/шум (SNR) досягає 65 дБА, що забезпечує достатньо високу якість отриманого сигналу та зменшує вплив внутрішніх шумів під час вимірювання [29].

Хоча обидва мікрофони задовольняють умову технічного завдання, в даного екземпляра ширший робочий діапазон та краще співвідношення

сигнал/шум, тому він є дещо кращим варіантом у порівнянні з INMP441, що і стало умовою його вибору.

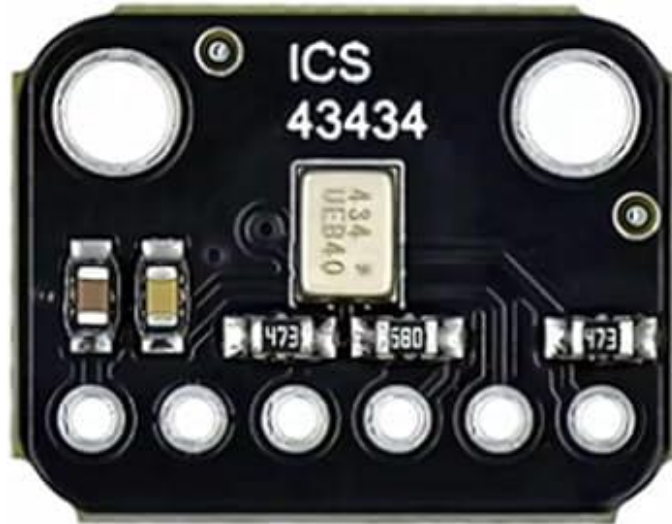


Рисунок 1.12 - Модуль з цифровим MEMS-мікрофоном ICS 43434

1.6.3 Вибір засобу для виведення інформації

Також важливим є вибір дисплея, що буде використовуватись в якості графічного інтерфейса користувача, основною вимогою до нього є частота оновлення інформації, проте також важливими є простота реалізації та можливість виведення великої кількості даних на екран.

Одним з варіантів для відображення інформації є рідкокристалічний дисплей LCD1602 (рис. 1.13, [74]). Даний дисплей дозволяє відображати 16 символів у 2 рядках і використовується переважно для виведення текстової інформації. Його основними перевагами є проста апаратна реалізація та широка підтримка з боку мікроконтролерних платформ. Споживання струму LCD1602 без підсвітки становить близько 1–2 мА, однак із увімкненою підсвіткою воно зростає до 20–100 мА, що суттєво впливає на енергоспоживання. LCD1602 (HD44780) має відносно низьку швидкість оновлення, яка зазвичай становить близько 30–100 мс на повне оновлення екрану, що обмежує можливість відображення швидко змінюваних даних. Крім того, обмежена кількість символів та відсутність можливості відображення графічних елементів значно

знижують інформативність інтерфейсу та ускладнюють реалізацію візуалізації вимірюваних даних [30].

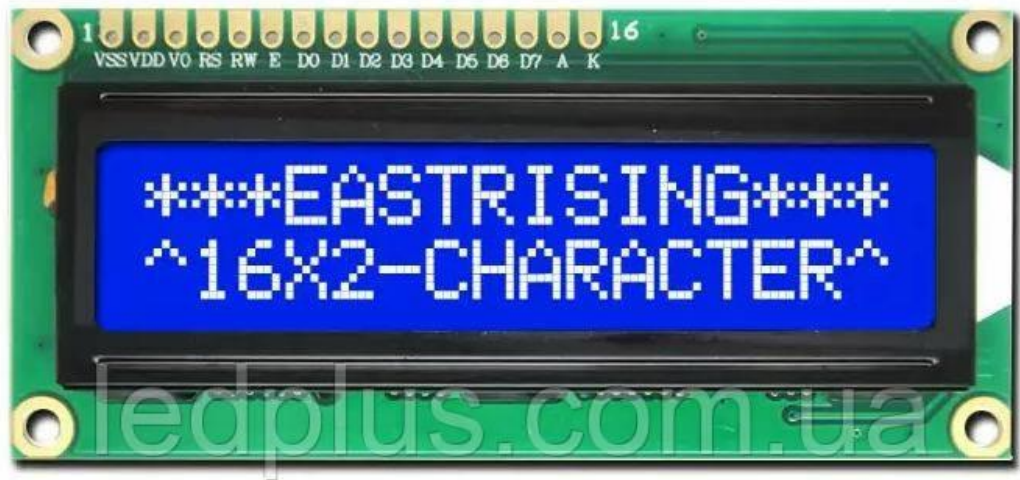


Рисунок 1.13 – Дисплей LCD1602

Для відображення результатів та роботи з пристроєм на графічному інтерфейсі користувача також можна використати OLED-дисплей SSD1306 малого розміру (рис. 1.14, [75]). Його застосування обумовлене низьким енергоспоживанням, компактним розміром, наявністю I2C підключення та високою якістю відображення інформації. Споживання електроенергії OLED-дисплея SSD1306 становить приблизно 0.4–1 мА у режимі відображення з мінімальною яскравістю та до 20–30 мА при повністю заповненому зображенні, а в режимі сну може знижуватися до десятків мікроампер (<0.01 мА) залежно від реалізації модуля. Також OLED-дисплей SSD1306 забезпечує значно вищу ефективну швидкість оновлення, яка може досягати приблизно 10–20 мс, залежно від інтерфейсу (I2C чи SPI) та реалізації драйвера [31].

Хоча обидва дисплея задовольняють умову технічного завдання (частота оновлення 2 Гц), було обрано саме SSD1306, у зв'язку з його компактністю, меншим енергоспоживанням, більшою частотою оновлення та простотою використання.



Рисунок 1.14 - OLED-дисплей SSD1306

1.6.4 Реалізація живлення пристрою

Останнім важливим елементом апаратної частини пристрою є його живлення. Його можна реалізувати або від постійного джерела живлення, наприклад, підключення до блоку живлення 5В або 12В, або від самостійного джерела живлення, такого як батарея або акумулятор. Для створення пристрою перевагу було надано саме самостійним джерелам живлення, бо їх використання дозволяє не тільки створити автономний пристрій, що не потребує постійного підключення до зовнішнього джерела живлення, а й дає змогу значно полегшити завдання стабілізації вхідного струму, бо навідміну від зовнішніх джерел, таких як USB-підключення до ПК чи підключення до розетки через блок живлення, акумулятор не створює різкі перепади в напрузі, видаючи стабільну напругу, яка майже не потребує окремих елементів для її обробки. Також наявність вбудованого в плату розробника ESP32 стабілізатора напруги дає змогу з легкістю перетворити вхідну напругу на 3.3В, необхідні для живлення дисплея та мікрофона [27].

Для живлення пристрою було обрано акумулятор BL-5C (рис. 1.15, [76]) разом з контролером заряду на базі елементу TP4056 (рис. 1.16, [77]) з захистом від перезаряду та перерозряду. Комбінація цих елементів дає можливість отримати стабільне живлення для пристрою з можливістю його зарядки або заміни елементу живлення за необхідності [32, 33].

Альтернативним варіантом живлення пристрою є використання одноразових батарей (наприклад, AA/AAA або 9V), які забезпечують просту реалізацію живлення без необхідності додаткових схем заряду, або набору акумуляторів формату AA/AAA. Однак у порівнянні з акумулятором BL-5C у поєднанні з модулем TP4056, батарейки мають суттєві недоліки, зокрема обмежену ємність та неможливість перезаряджання, що призводить до необхідності їх регулярної заміни та підвищує експлуатаційні витрати, а акумулятори форм-фактору AA/AAA хоча і є задовільним вибором, проте мають дещо більший розмір, ніж BL-5C. Напруга одноразових батарей також знижується у процесі розряду, що може негативно впливати на стабільність роботи електроніки без додаткових стабілізаторів. У той же час акумулятор BL-5C забезпечує багаторазове використання, а TP4056 дозволяє реалізувати контрольований процес заряду з захистом від перезаряду та глибокого розряду, що підвищує надійність і автономність пристрою.



Рисунок 1.15 - Акумулятор BL-5C

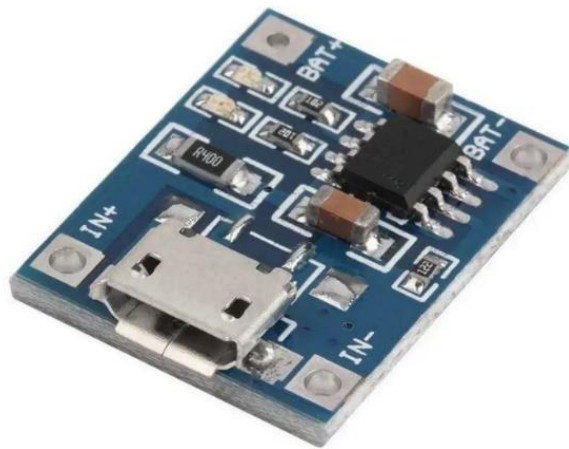


Рисунок 1.16 - Контролер заряду на базі елементу TP4056

Обраний склад елементної бази забезпечує реалізацію необхідних функціональних можливостей пристрою згідно поставленого технічного завдання при відносно невеликій вартості компонентів та простоті практичної реалізації. Використання сучасних цифрових компонентів дозволяє підвищити точність вимірювань і забезпечити можливість подальшого розширення функціоналу системи.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИМІРЮВАННЯ РІВНЯ ШУМУ

2.1 Проєктування елементів складальної одиниці

Розглянемо структурну схему для автоматизованої системи вимірювання рівню шуму. На рис.2.1 представлена структурна схема приладу, вона відображає основні функціональні вузли та їх взаємодію.

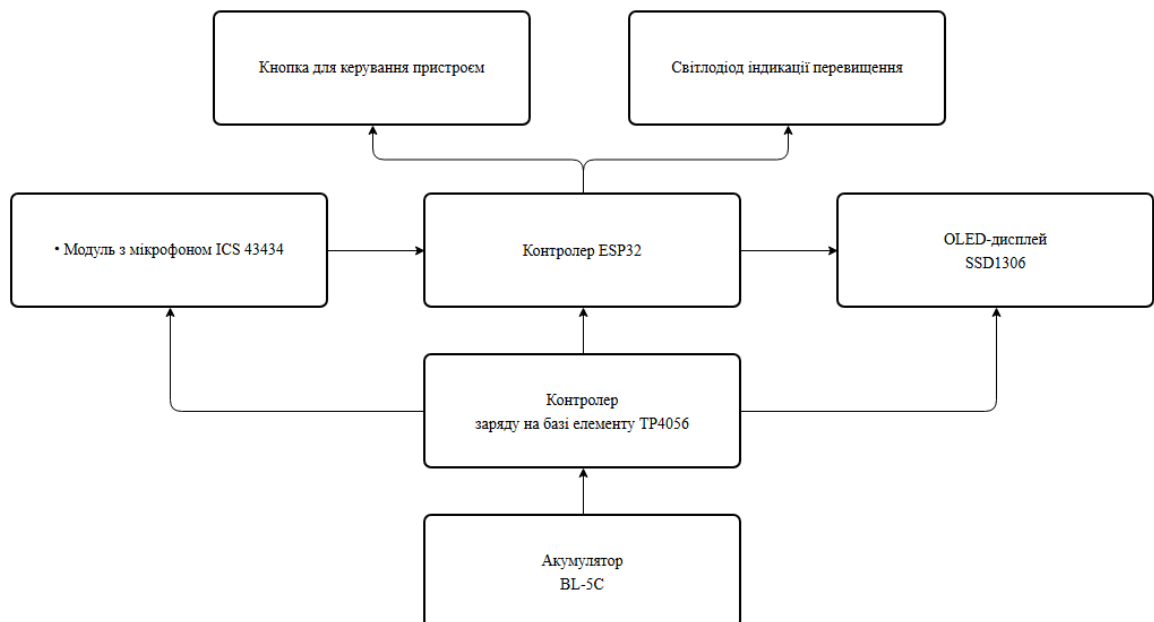


Рисунок 2.1 – Структурна схема пристрою

Структурна схема пристрою складається з 7 елементів, а саме:

- Контролер ESP32;
- Модуль з мікрофоном ICS 43434;
- OLED-дисплей SSD1306;
- Акумулятор BL-5C;
- Контролер заряду на базі елементу TP4056;
- Кнопка для керування та налаштування пристрою;
- Світлодіод для індикації перевищення заданого рівня шуму.

Живлення системи відбувається за допомогою акумулятора BL-5C та контролера заряду TP4056, що на виході дає напругу в $4.2V \pm 1\%$ [32, 33]. Такий рівень підходить для підключення до ESP32 та SSD1306, але для підключення модуля з мікрофоном ICS 43434 необхідно 3.3V, тому подача напруги відбувається через вбудований стабілізатор напруги в ESP32 DevKit [27]. Використання контролера заряду також дає такі переваги, як контроль процесу заряджання акумулятора, захист від перезаряду та перерозряду, захист від короткого замикання та контроль напруги заряду.

Модуль мікрофона підключається до ESP32 за допомогою цифрового інтерфейсу I2S (рис 2.2, [78]), який забезпечує передачу цифрових аудіоданих без використання додаткових аналогових підсилювачів або аналогово-цифрових перетворювачів. Основні канали інтерфейсу це WS (Word Select), SCK (Serial Clock) та SD (Serial Data). WS використовується для синхронізації передачі аудіоданих та визначення каналу, SCK передає тактові імпульси, які допомагають визначити швидкість обміну даними між мікрофоном та ESP32, а SD - це лінія цифрової передачі звукового сигналу від мікрофона до мікроконтролера [34].

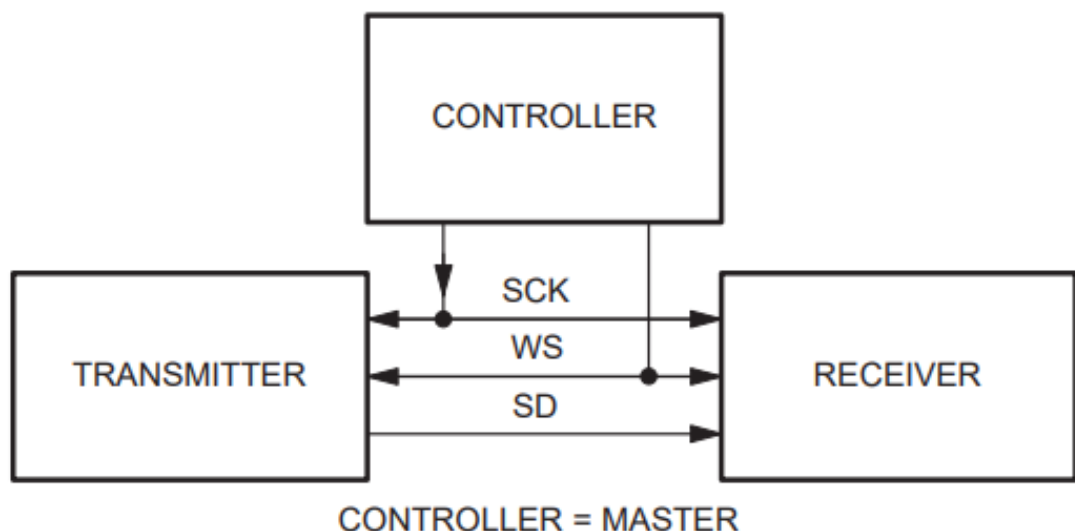


Рисунок 2.2 – Схематичне зображення принципу роботи інтерфейсу I2S

Після передачі інформації з мікрофона, мікроконтролер обробляє її згідно заданих параметрів та передає інформацію на дисплей SSD1306 за допомогою інтерфейсу I2C, а саме ліній передачі SCL (Serial Clock Line) та SDA (Serial Data Line). SCL відповідає за синхронізацію частоти роботи дисплея та мікроконтролера, в той час як через SDA передаються такі дані як поточний рівень шуму, пороговий рівень шуму та інші графічні елементи [35].

Для взаємодії користувача з пристроєм передбачено використання кнопки керування, яка дозволяє задавати та змінювати необхідні параметри системи, зокрема встановлювати граничне значення рівня шуму, після перевищення якого спрацюватиме світлодіодна індикація. У схемі використовується внутрішній підтягувальний резистор мікроконтролера ESP32, тому додатковий зовнішній резистор не потрібен [27]. Принцип роботи полягає в тому, що у нормальному стані на вході мікроконтролера підтримується логічна одиниця, а при натисканні кнопки вхід замикається на землю, що дає змогу отримати логічний нуль. Саме цю зміну і “помічає” ESP32, що дає змогу використовувати її для подальших цілей, таких як входу до меню налаштувань, зміни порогового значення рівня шуму, перемикання режимів роботи чи підтвердження вибраних користувачем параметрів.

Для візуального інформування користувача про перевищення заданого рівня шуму використовується світлодіод. Принцип роботи полягає в тому, що при перевищенні певного рівня шуму мікроконтролер встановлює високий логічний рівень на одному з GPIO-виходів, у результаті чого світлодіод починає світитись. Але для підключення необхідно обмежити струм за допомогою резистора номіналом в 200-300Ом, який запобігатиме пошкодженню світлодіода.

Схема підключення пристрою представляє собою дві окремі умовні частини. Перша – це частина, що відповідає за живлення плати мікроконтролера. До неї відносяться контролер заряду та батарея. Другою є частина з мікрофоном, елементами керування та графічним дисплеєм користувача (рис. 2.3). Проєктування схеми підключення та друкованої плати відбувалось в середовищі

EasyEDA [36]. Можливості цього середовища проєктування дають змогу швидко та зручно створити схему, перетворити її в друковану плату та багато іншого. Також наявність бібліотеки користувацьких елементів значно полегшує пошук та створення схеми чи друкованої плати.

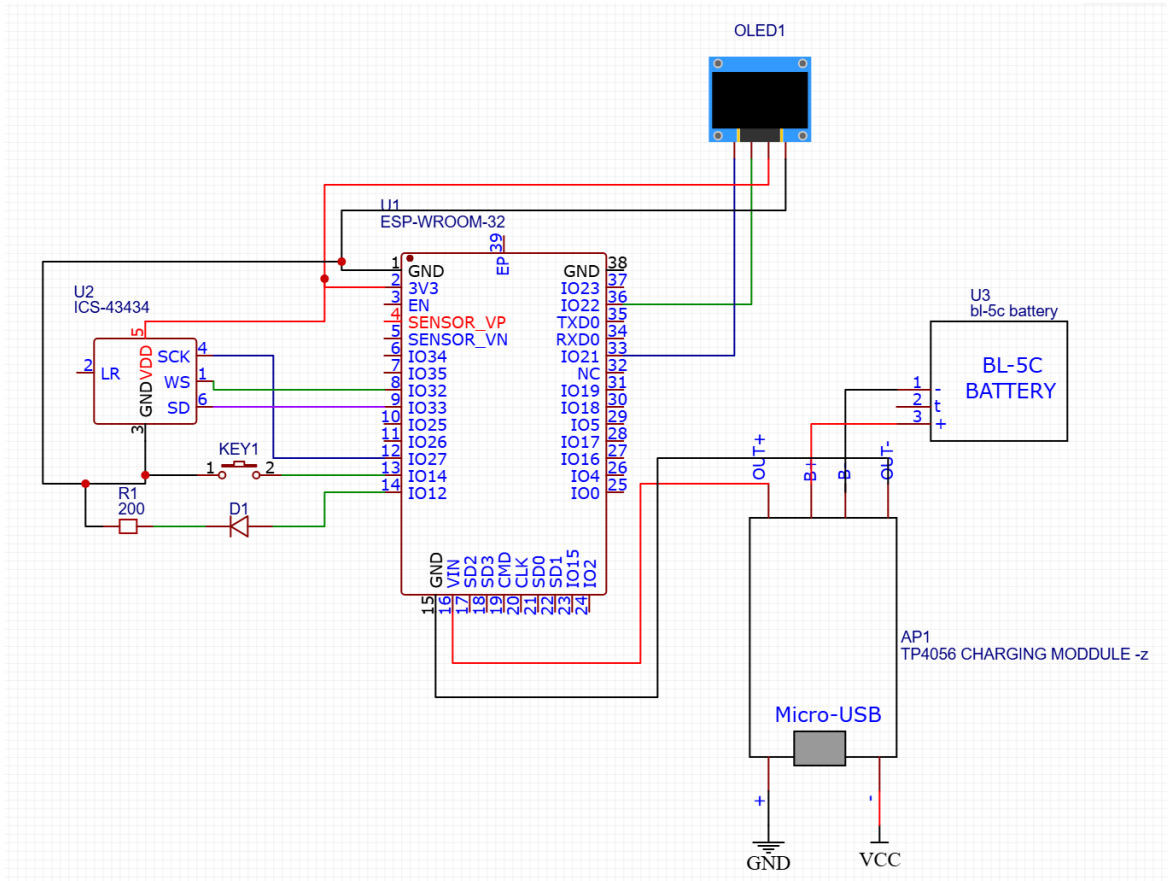


Рисунок 2.3 – Схема підключення елементів пристрою

Після створення схеми пристрою та розуміння принципу підключення його елементів можна перейти до створення друкованої плати для нього. (рис.2.4) Цей етап складається з проєктування самої плати та її фізичного створення. Під час розроблення важливо звернути увагу на віддзеркалювання елементів, бо для зручної збірки плати та запаювання всіх елементів необхідно розміщати елементи з протилежної сторони плати від розведених доріжок. В такому випадку всі контакти на підключеннях елементів будуть віддзеркалені, неврахування чого може спричинити повну недієздатність та навіть поломку деяких з елементів.

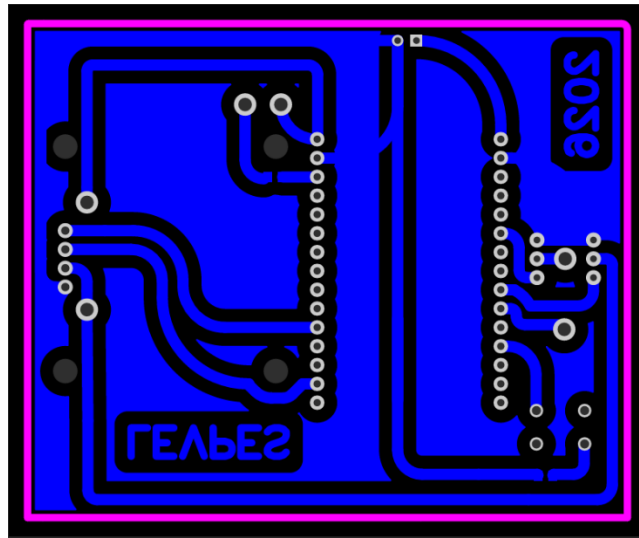


Рисунок 2.4 - Схема друкованої плати

Також для пристрою необхідно розробити корпус, який забезпечуватиме захист від прямого попадання води на плату, захист від пошкоджень та стабільне і надійне кріплення елементів пристрою під час його роботи. Для розроблення корпусу використовувалось середовище SolidWorks [52]. Його використання зумовлене проходженням курсу “Комп’ютерна графіка” під час навчання в національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» та наявності студентської ліцензії на використання даного середовища, що дало необхідні можливості та навички для створення надійного корпусу для шумоміру.

Головним питанням створення корпусу є вибір матеріалу, з якого він буде створений. У випадку портативного шумоміру пристрій повинен бути легким, захищеним від вологи та пошкоджень та зручним у використанні, тому було розглянуто кілька варіантів для створення корпусу пристрою, а саме: деревина, акрил, метал, ABS-пластик та фотополімерна смола.

Перевагою деревини є її дешевизна, низька вага та легкість в обробці. Для створення корпусу з деревини необхідно мати лише базові інструменти та навички. Проте недоліком даного матеріалу є низька вологостійкість, що може призвести до швидкої деградації матеріалу корпусу та навіть попадання вологи в пристрій, що може призвести до його повної поломки [37].

Поліметилметакрилат (акрил) також є непоганим вибором для створення корпусу завдяки своїй легкості та гнучкості, але основною проблемою його використання є необхідність обробки за допомогою спеціалізованих станків, що є негативною ознакою для пристрою, створюваного в домашніх умовах [38].

Використання металів є доцільним на шумомірах настільного типу, наприклад, раніше показаний ВШВ-003-М2 (рис.1.5). Цей матеріал є міцним, стійким та легким в обробці (якщо виготовляти пристрій з листів металу), проте його вага унеможливорює створення компактного пристрою для ручного носіння [39].

ABS-пластик є одним з найкращих варіантів для створення корпусу пристрою у зв'язку з його дешевизною, легкістю, міцністю та доступністю, пов'язаною з розвитком 3-Д друку. Корпуси з ABS-пластику широко застосовуються під час створення пристроїв не тільки в промислових масштабах, а і для саморобних пристроїв. Проте великим недоліком 3-Д друку за допомогою ABS-пластику є друк шарами. При виборі неідеальних налаштувань друку можуть спостерігатись розкладання корпусу на окремі шари, що також може бути викликане при фізичному впливі на корпус, наприклад, якщо гострим предметом сильно натиснути між шарами корпусу, вони можуть дати тріщину або розділитись в цьому місці повздовж лінії між шарами пластику [40].

Для вирішення цієї проблеми можна застосувати іншу технологію 3-Д друку, а саме друк за допомогою фотополімерної смоли. Її принцип полягає в тому, що смола під час дії на неї ультрафіолетових променів застигає, що дозволяє плавно “нарощувати” корпус з рідкої смоли. Такий підхід прибирає поділення корпусу на шари та дає змогу отримати надійний, легкий та герметичний корпус для пристрою. Проте для використання даного методу необхідно мати спеціальний принтер та смолу, ціна на яких є доволі високою, що значно знижує доступність цього методу для саморобних проєктів [41].

У моєму випадку було використано друк фотополімерною смолою у зв'язку з її очевидними перевагами та наявністю необхідного обладнання.

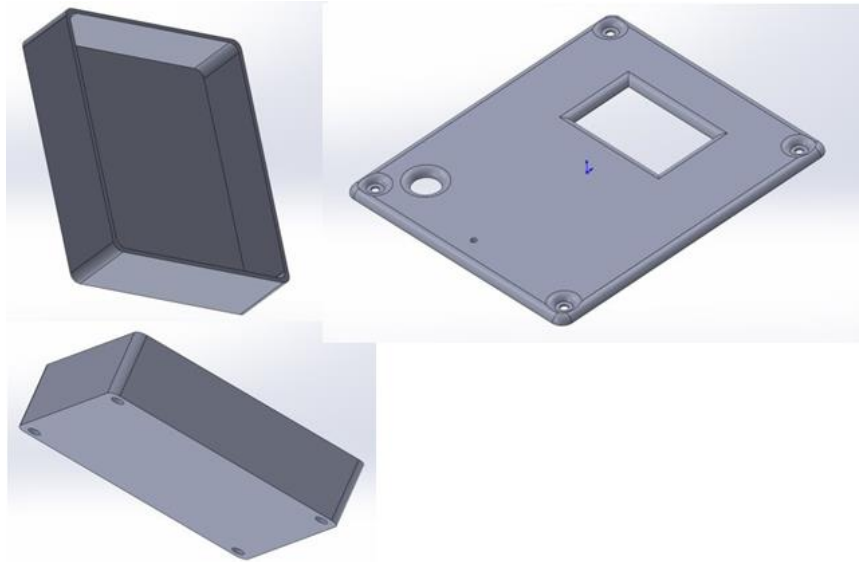


Рисунок 2.5 – Модель корпусу для пристрою

2.2 Розроблення графічного інтерфейсу користувача

Графічний інтерфейс користувача є важливою складовою вимірювального пристрою, оскільки саме він забезпечує взаємодію користувача із системою та відображення результатів вимірювань у зручному для сприйняття вигляді. Якість реалізації інтерфейсу безпосередньо впливає на зручність експлуатації пристрою, швидкість отримання необхідної інформації та загальну ефективність роботи системи.

Під час розроблення графічного інтерфейсу основна увага приділялася простоті використання, зрозумілості відображуваних даних та мінімальному навантаженню на обчислювальні ресурси мікроконтролера. Оскільки в даному проєкті використовується мікроконтролер ESP32 та OLED-дисплей SSD1306 з роздільною здатністю 128×64 пікселі [31], інтерфейс повинен враховувати обмеження дисплея щодо кількості символів та доступної області відображення.

Однією з головних задач під час створення інтерфейсу було забезпечення швидкого сприйняття користувачем поточних параметрів роботи пристрою. З цією метою на екран виводяться лише найбільш важливі дані, які необхідні для контролю процесу вимірювання.

Відображення інформації організовано таким чином, щоб найбільш важливий параметр — поточне значення рівня шуму — займав центральну частину екрана та відображався контрастним кольором. Це дозволяє забезпечити зручне зчитування інформації навіть при значній відстані між користувачем та пристроєм.

Під час програмної реалізації графічного інтерфейсу використовувалася спеціалізована бібліотека для роботи з OLED-дисплеєм SSD1306 від ThingPulse (рис. 2.6, [79]). Дана бібліотека забезпечує зручне створення текстових та графічних елементів, керування шрифтами, формування геометричних примітивів та оновлення вмісту дисплея. Основною її перевагою є підтримка I2C інтерфейсу [35] та легка інтеграція в програмну частину пристрою [42].



Рисунок 2.6 – Можливості бібліотеки ThingPulse OLED SSD1306

Окрему увагу було приділено оптимізації швидкості оновлення інформації. Згідно технічного завдання необхідно забезпечити частоту оновлення дисплею в 2 Гц, тобто кожні 0.5 секунди. Для забезпечення цього рівня швидкодії пристрій має встигати обчислювати рівень шуму за заданий час, щоб вчасно виводити актуальну інформацію про рівень шуму в даний момент. Для виконання цього завдання в коді необхідно буде підібрати правильні параметри для розміру буфера та швидкості обчислень даних, щоб правильно та достатньо швидко обраховувати та виводити дані на дисплей.

Структура графічного інтерфейсу користувача складається з кількох екранів: екрану запуску пристрою, основного екрану вимірювань, екранів налаштування порогового значення рівня шуму та екрану вибору режиму вимірювання. Екран запуску пристрою відображається під час початку роботи з ним та для індикації запуску пристрою (рис.2.7). На основному екрані відображається інформація про поточний рівень шумового навантаження (рис.2.8), на екрані налаштування ліміту звуку можна змінити рівень порогового значення шумового навантаження (рис.2.9), а на екрані перемикання спрацювання даного ліміту - увімкнути або вимкнути індикацію перевищення цього порогового значення (рис.2.10). На екрані зміни режиму роботи пристрою можна обрати принцип фільтрації шумів, а саме увімкнути A – weighting, або вимкнути його (Z – weighting) (рис.2.11). Використання фільтру A – weighting дозволяє імітувати людський слух та вимірювати частоти, які може сприймати людина, в той час як Z – weighting вимірює рівень шуму на всіх діапазонах частот, навіть невідчутних для людського слухового апарату [42].



Рисунок 2.7 – Екран запуску пристрою



Рисунок 2.8 – Екран налаштувань графічного інтерфейсу



Рисунок 2.9 – Екран налаштування ліміту звуку



Рисунок 2.10 – Екран перемикання спрацювання ліміту звуку

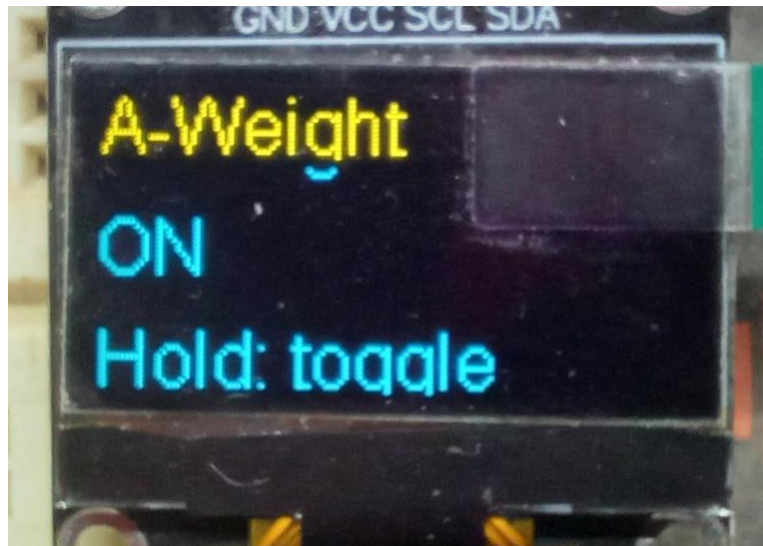


Рисунок 2.11 – Екран налаштування ліміту звуку

У результаті було створено компактний, інформативний та зручний апаратний графічний інтерфейс користувача, який дозволяє ефективно взаємодіяти з пристроєм та забезпечує наочне відображення результатів вимірювання рівня шуму. Реалізований інтерфейс відповідає функціональним вимогам системи та може бути розширений у майбутньому шляхом додавання нових елементів керування або режимів відображення інформації.

Також для даної автоматизованої системи було розроблено графічний веб – інтерфейс, на якому відображається графік вимірювання шуму за останні 50 секунд. Для його створення використовувався наявний в ESP32 модуль WiFi та бібліотеки для роботи з мережею і веб-сервером. Для створення веб-сторінки використовувались бібліотеки WiFi.h [44] та WebServer.h [45]. За допомогою цих бібліотек було створено простий веб-інтерфейс, який дає змогу дистанційно спостерігати за вимірюваннями пристрою та фіксувати зміну рівня шуму за останні 50 секунд в реальному часі. Працездатність веб-інтерфейсу було перевірено на настільному ПК та мобільних пристроях (рис. 2.12).

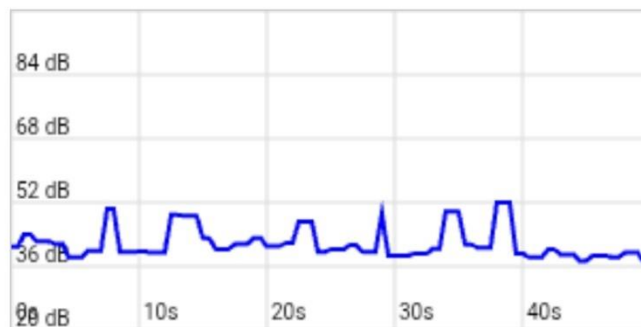
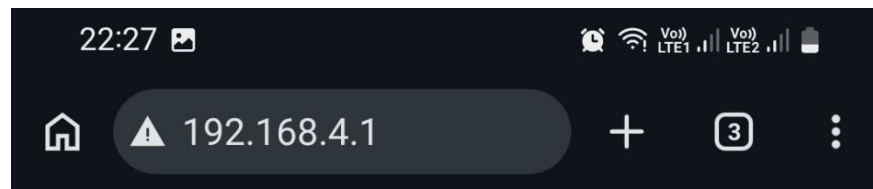


Рисунок 2.12 – Веб-інтерфейс з відображенням графіку вимірювань

2.3 Апаратне та програмне налагодження системи

Після завершення етапу проєктування структурної схеми пристрою та його інтерфейсу користувача, наступним кроком є створення та налагодження апаратної і програмної частин системи. Даний етап є найважливішим та найбільш трудомістким під час розроблення автоматизованої системи для вимірювання рівня шуму, оскільки саме на ньому здійснюється перевірка працездатності окремих компонентів та всієї системи в цілому. Основною метою налагодження є забезпечення стабільного функціонування пристрою, правильного обміну даними між модулями, коректної обробки звукового сигналу та відображення результатів вимірювання.

Під час реалізації апаратної частини особлива увага приділялася правильності підключення цифрового MEMS-мікрофона. Оскільки передавання звукових даних здійснюється за допомогою інтерфейсу I2S [34], необхідно було

забезпечити правильне з'єднання сигнальних ліній та синхронізацію обміну даними. При помилковому підключенні контактів можуть виникати неточності сигналу або повна відсутність даних на вході мікроконтролера. Ще одним важливим моментом, що проявив себе під час збірки апаратної частини пристрою на макетній платі є недопущення контакту проводів мікрофону до інших провідникових елементів пристрою, бо навіть найменші впливи магнітного поля можуть значним образом спотворювати дані, що отримуватиме головний контролер від мікрофону (рис. 2.13). Для виконання цієї умови важливо розмістити мікрофон так, щоб і він, і провідне підключення до нього знаходились на великій дистанції від любых провідникових елементів та, в особливості, елементів живлення системи.

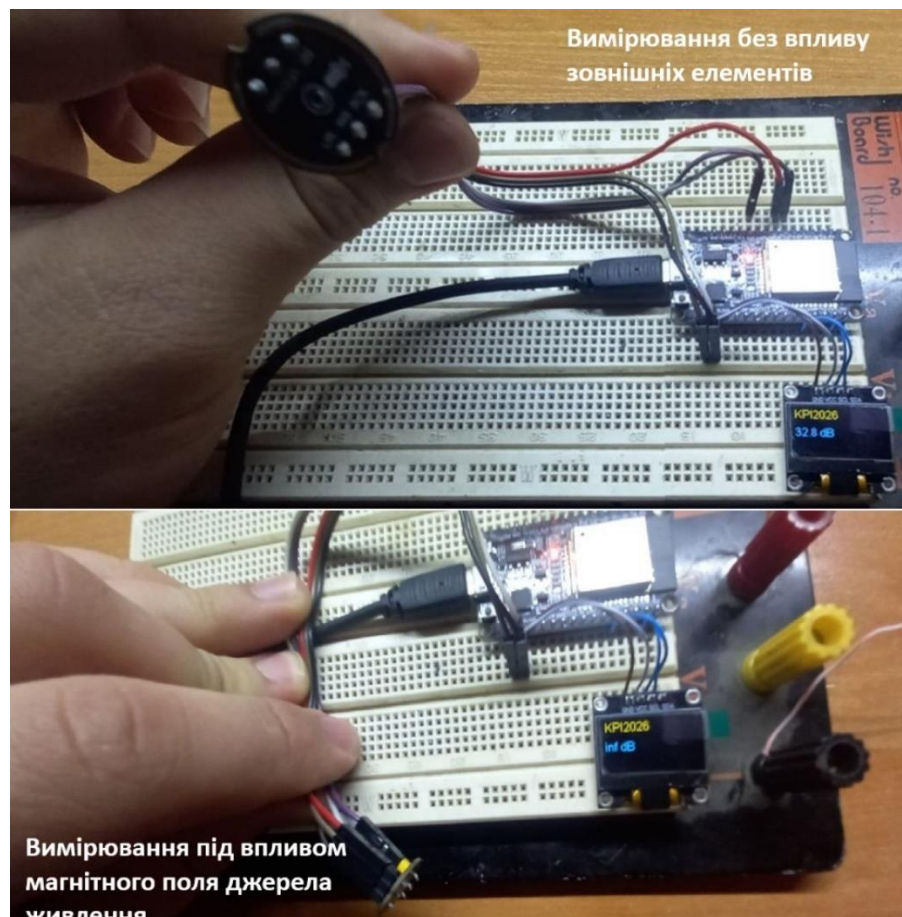


Рисунок 2.13 – Демонстрація впливу елементів живлення на точність вимірювань

Ще одним етапом для реалізації системи є підключення дисплея SSD1306. Даний модуль використовує інтерфейс I2C, який дозволяє зменшити кількість

сигнальних провідників [35]. Через дисплей здійснюється відображення результатів вимірювання рівня шуму, поточного стану системи, налаштувань та додаткової службової інформації. На етапі підключення було перевірено правильність роботи ліній передачі даних та синхронізації, а також відповідність адресації дисплея налаштуванням програмного забезпечення.

Для забезпечення взаємодії користувача з пристроєм було підключено кнопку керування. Основним її призначенням є зміна параметрів роботи системи, зокрема встановлення порогового значення рівня шуму, при перевищенні якого активується система сигналу попередження, режим роботи даного попередження та використання окремих фільтрів вимірювання звуку.

Також під час програмної реалізації необхідно врахувати явище брязкіту контактів, яке виникає при механічному натисканні кнопки та може спричинити багаторазове хибне спрацювання. Для усунення цього ефекту потрібно застосувати програмний механізм затримки натискання кнопки, який враховуватиме натискання кнопки тільки у тому випадку, якщо вона була утримана в цьому стані довше певного періоду часу (наприклад, для короткого натискання можна встановити період в 200мс). Таким чином можна відфільтрувати явище брязкіту контакту кнопки та спрацьовувати лише один раз при натисканні.

Для індикації перевищення встановлюваного користувачем рівня шуму використовується світлодіодний індикатор. При перевищенні заданого користувачем порогового значення в апаратному інтерфейсі мікроконтролер формує керуючий сигнал, який активує світлодіод та оповіщає користувача. Реалізація такого механізму дозволяє користувачу швидко отримувати інформацію про перевищення вказаного рівня шуму без необхідності постійного контролю показників на дисплеї.

Особлива увага приділялася організації системи живлення. Для забезпечення автономної роботи використовується акумулятор BL-5C та модуль заряджання TP4056. Оскільки напруга літій-іонного акумулятора змінюється в процесі його роботи, необхідно забезпечити стабільне живлення всіх

електронних компонентів системи [32]. Для цього використовується вбудована в плату контролера ESP32 зміна рівня напруги, яка забезпечує підтримання необхідного рівня живлення для мікроконтролера та периферійних модулів [27].

Після завершення розроблення апаратної частини пристрою було виконано програмне налагодження системи, основною метою якого є забезпечення коректної взаємодії між усіма компонентами та усунення можливих помилок у процесі функціонування пристрою. Даний етап є одним із найважливіших під час створення електронних вимірювальних систем, оскільки навіть за правильно спроектованої апаратної частини помилки в програмному забезпеченні можуть призвести до некоректних результатів вимірювань або нестабільної роботи пристрою.

Процес налагодження здійснювався поетапно. На початковому етапі перевірялася робота окремих модулів системи незалежно один від одного. Такий підхід дозволяє локалізувати можливі несправності та спростити пошук причин їх виникнення. Для перевірки дисплею достатньо підключити його до контролера через інтерфейс I2C та зконфігурувати його підключення в середовищі програмування, після чого, за допомогою бібліотеки для роботи з дисплеями SSD1306 [42], вивести на нього тестове повідомлення та перевірити можливість адекватно та вчасно оновлювати інформацію на дисплеї під час роботи (рис.2.14).



Рисунок 2.14 – Перевірка працездатності дисплею

Також під час виконання цього етапу було перевірено запуск та базову роботу мікроконтролера ESP32. Для подальшої перевірки працездатності плати контролера було перевірено можливість перетворення рівня вхідної напруги з батареї (4.2В) на необхідні для роботи елементів 3.3В, підключивши USB-кабель до плати контролера, на виході 3V3 має з'явитись рівень напруги величиною приблизно в 3.3В, що було підтверджено мультиметром під час тестування (рис.2.15). Також необхідно запевнитись в працездатності вбудованого модулю WiFi, тому було використано тестовий код, який давав можливість модулю створити свою бездротову мережу (рис.2.16).

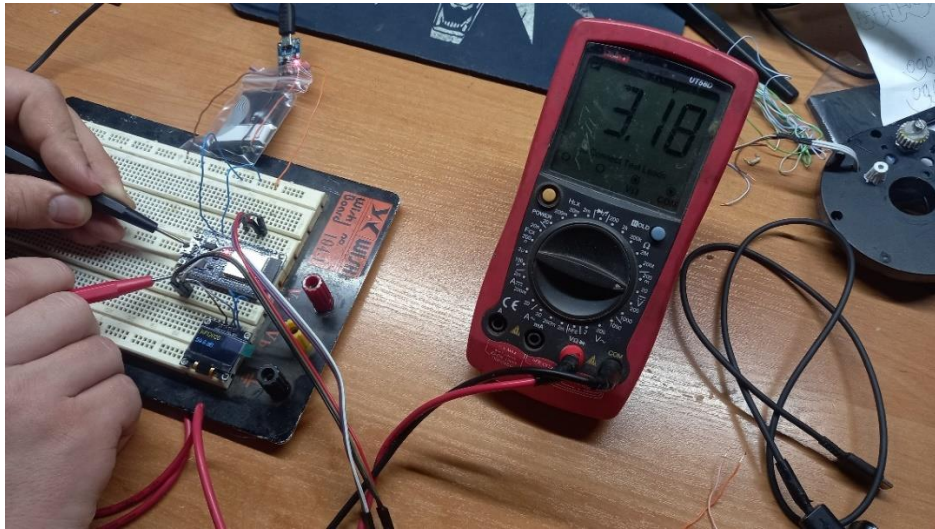


Рисунок 2.15 – Перевірка живлення

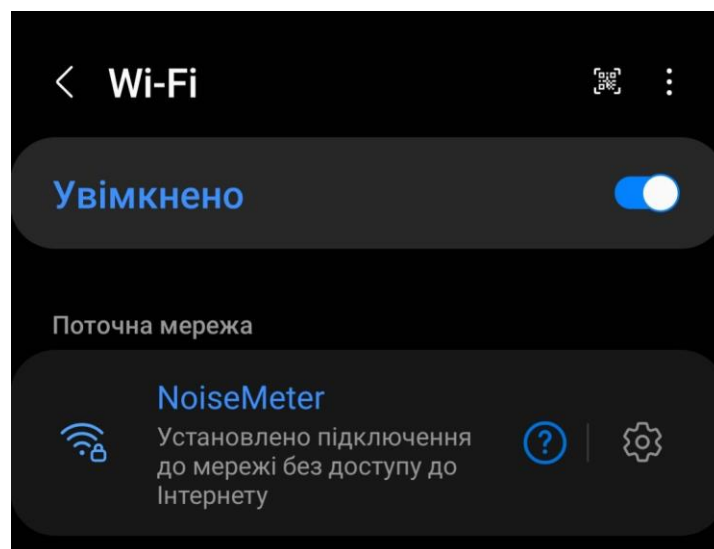


Рисунок 2.16 – Перевірка працездатності WiFi модуля

Наступним та найважливішим етапом перевірки працездатності елементів стала перевірка коректності роботи цифрового MEMS-мікрофона ICS-43434. Даний мікрофон використовує інтерфейс I2S для передачі цифрового аудіосигналу. Для розуміння принципу роботи цього мікрофону та інтерфейсу передачі даних важливо детально розуміти, яким чином ці дані вимірюються та обробляються.

Все починається з того, що в мікрофоні знаходиться тонка кремнієва пластина, під якою знаходиться ще одна пластина. Разом вони утворюють конденсатор, що складається з однієї рухомої та однієї нерухомої пластини. Під дією звукового тиску, завдяки своїй надзвичайній тонкості, яка вимірюється в кількох одиницях мікрометрів, кремнієва (рухома) пластина вигинається, що дозволяє змінити ємність цього конденсатора (рис.2.16). Проте ця зміна ємності є надзвичайно малою та вимірюється в фемто- (10^{-15}) чи піко- (10^{-12}) фарадах, тому виміряти її напряму майже неможливо [46].

Тому наступним етапом є перетворення різниці ємності в різницю напруги. Це виконується для того, щоб в подальшому можна було підсилити цей сигнал до достатньо високого значення для обробки. Отриманий сигнал вимірюється в мікрровольтах, тому його підсилюють до мілівольт, що є достатнім рівнем для рєстрації в АЦП. Але перед тим, як перетворювати цей сигнал, проводиться його антиаліасинг. Аліасинг відбувається коли високочастотний сигнал є занадто високим для частоти дискретизації, тому не встигає правильно обраховуватись і в результаті є невірним. Для перешкодження такому явищу є формула розрахунку максимальної допустимої частоти сигналу при певній частоті дискретизації, теорема Найквіста [47]. Вона виглядає так:

$$f_s \geq 2f_{max}$$

де f_s – частота дискретизації (в нашому випадку 48кГц), а f_{max} – максимальна допустима частота сигналу.

З цієї формули виходить, що для цього мікрофона необхідно мати частоту сигналу не більше 24 кГц. Цю функцію і виконує модуль антиаліасингу, він пропускає частоти до 24кГц, та пригнічує всі частоти, що його перевищують.

Після цього сигнал йде в блок АЦП Sigma-Delta. Його суть полягає в перетворенні сигналу на надшвидкий потік двійкових даних, який після цього обробляється наступним блоком, дециматором, який зменшує частоту дискретизації з Sigma-Delta, яка може сягати 3 мГц до 48 кГц, беручи групи бітів з потоку АЦП та обраховуючи їх середнє значення. Наприклад, він бере уривок з потоку даних 001010100010, який складається з чотирьох одиниць та 8 нулів (насправді ці уривки є набагато більшими). Він вираховує середнє значення, що для наведеного прикладу є 0.33(3), після чого оцифровує його. Для оцифровки він порівнює два діапазони, перший – від 0 до 1, це діапазон вимірюваного середнього значення, а другий – від -8388608 до 838607, це діапазон значень АЦП (саме для 24-бітного АЦП). Після цього він підставляє значення з одного діапазону в інший та отримує вже значення у тому вигляді, в якому вони будуть передаватись на мікроконтролер та переводитись у децибели [46].

Для передачі даних з мікрофона на мікроконтролер використовується інтерфейс передачі I2S. На своєму виході мікрофон дає оброблену інформацію про відхилення в діапазоні в від -8388608 до 838607, але робить це в 24-бітному форматі. Проте під час передачі мікроконтролер не може ніяк розділити де закінчується одне значення, яке передає мікрофон і починається нове, тому для цього і застосовується цей інтерфейс. Раніше було згадано, що він складається з трьох ліній, а саме BCLK, SD та WS. Для справної передачі даних необхідна взаємодія цих трьох ліній, бо кожна з них виконує свою важливу функцію в передачі даних. По лінії SD (Serial Data) постійно передаються значення в 24-бітному форматі, які видає сам мікрофон, проте для коректного зчитування кожного біта даних мікроконтролеру необхідно знати з якою частотою ці дані надсилаються. Для цього використовується лінія BCLK (Bit Clock), вона з певною частотою з однаковими інтервалами передає мікроконтролеру сигнал, який дає зрозуміти, що настав час зчитувати наступний біт інформації, після чого

мікроконтролер зчитує її з лінії SD. Проте хоч він і зчитує дані, він всеодно не може їх правильно розділити. Для цього і є лінія WS (Word Select) кожного разу як передається повне число лінією SD, він змінює значення на 0 або 1, що і дає мікроконтролеру можливість зрозуміти, коли закінчується та починається нове слово [46].

Особливістю роботи I2S є необхідність правильного налаштування параметрів синхронізації, оскільки навіть незначне відхилення параметрів може призвести до спотворення сигналу або повної відсутності даних. Під час прийому даних, необхідно враховувати, що дані з мікрофона складаються з 24-бітних слів [34], в той час як контролер ESP32 опрацьовує їх в 32-бітний масив (int32), тому необхідно обробляти отримані дані з мікрофона, роблячи їх зсув на 8 біт. Також важливо враховувати, що дані через I2S передаються в сирому форматі, а перетворення в дБ відбувається вже програмним чином з використанням формул та урахуванням особливостей мікрофона. Для перевірки мікрофона достатньо спостерігати за отриманими даними, вони мають коливатись в невеликому діапазоні (від -2000 до 2000 умовних одиниць в тихому приміщенні) та значно збільшуватись (в обидві сторони, наприклад, від -15000 до 15000) при збільшеному впливу звуку на мікрофон [46].

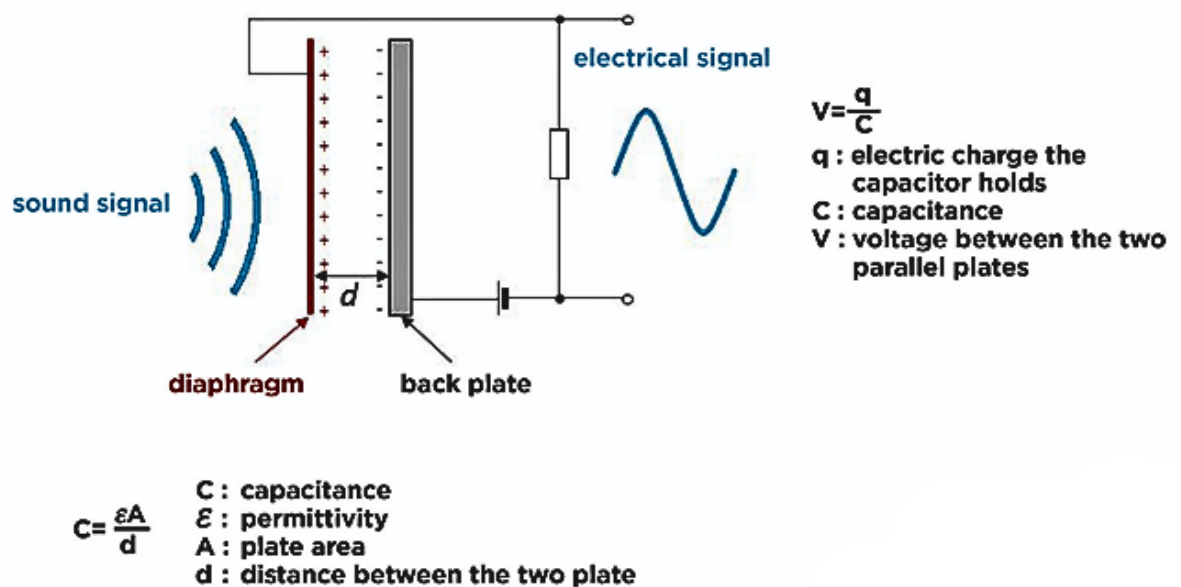


Рисунок 2.17 – Принцип роботи MEMS-мікрофона

Для перевірки працездатності мікрофона спочатку використовувалося виведення необроблених значень аудіосигналу у послідовний монітор. Отримані значення аналізувалися при різних рівнях шуму в навколишньому середовищі. При відсутності звукового сигналу покази повинні були залишатися близькими до нульових, тоді як при появі звуку повинна спостерігатися значна зміна амплітуди отриманих значень. Після перевірки було підтверджено реакцію мікрофона на вплив звукових хвиль, що є основною умовою для подальшого вимірювання та обробки рівня шуму.

Останнім етапом перевірки працездатності елементів була перевірка елементів живлення, а саме контролеру заряду та батареї. Спочатку батарея була під'єднана до контролера живлення, плюсовий контакт батареї був підключений до контакту BAT+ на платі заряду, а мінусовий контакт батареї був підключений на контакт BAT- [33]. Після цього на порт мікро-usb контролера заряду було підключено кабель з живленням 5В.

Очікуваним результатом є світлова індикація червоним світлодіодом на платі заряду, який позначає, що заряджання батареї відбувається успішно. Під час процесу заряджання можна перевірити рівень живлення на контактах BAT+ та BAT-, який у разі заряджання має становити приблизно 4.2В (рис. 2.18) [33]. Після цього необхідно залишити контролер підключеним до живлення на деякий час, доки на платі не засвітиться зелений світлодіод, який позначає кінець заряджання (рис. 2.19).



Рисунок 2.18 - Перевірка процесу заряджання батареї

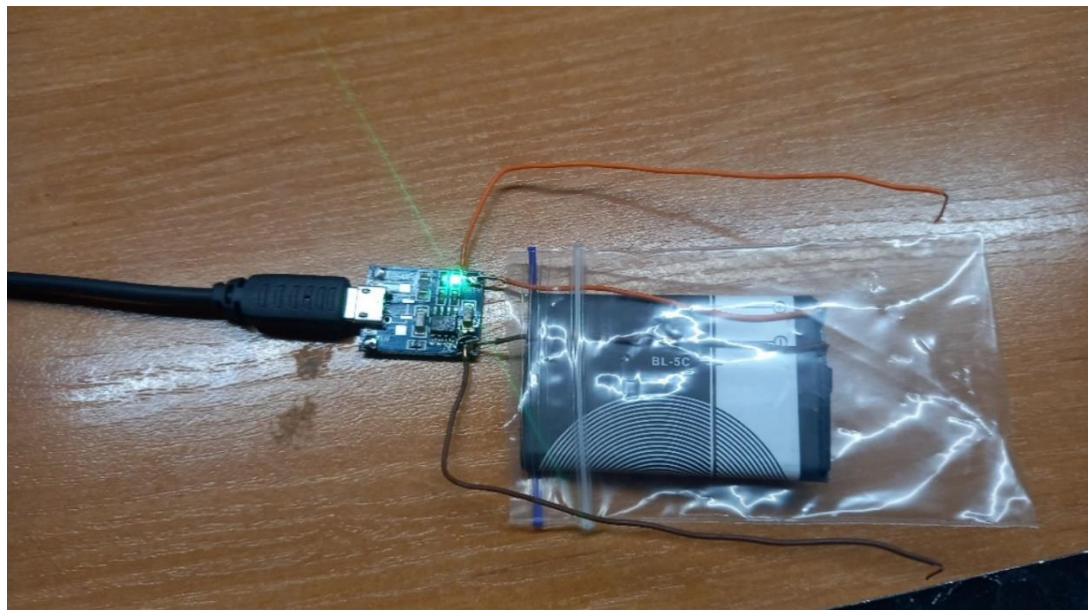


Рисунок 2.19 – Індикація повного заряду батареї

Наступним після підтвердження працездатності всіх елементів є розробка апаратної та програмної частини безпосередньо самої автоматизованої системи для вимірювання рівня шуму.

Спочатку необхідно підключити всі елементи та налаштувати взаємодію між ними. Для зручності розробки пристрою на етапі апаратного та програмного налаштування використовувалась макетна плата, що дозволяло швидко вносити необхідні зміни до будови пристрою (рис.2.20). Найважливішим на цьому етапі є правильне підключення всіх ліній живлення та передачі інформації, неправильно підключені компоненти будуть надавати недостовірну інформацію або можуть загалом вийти з ладу.

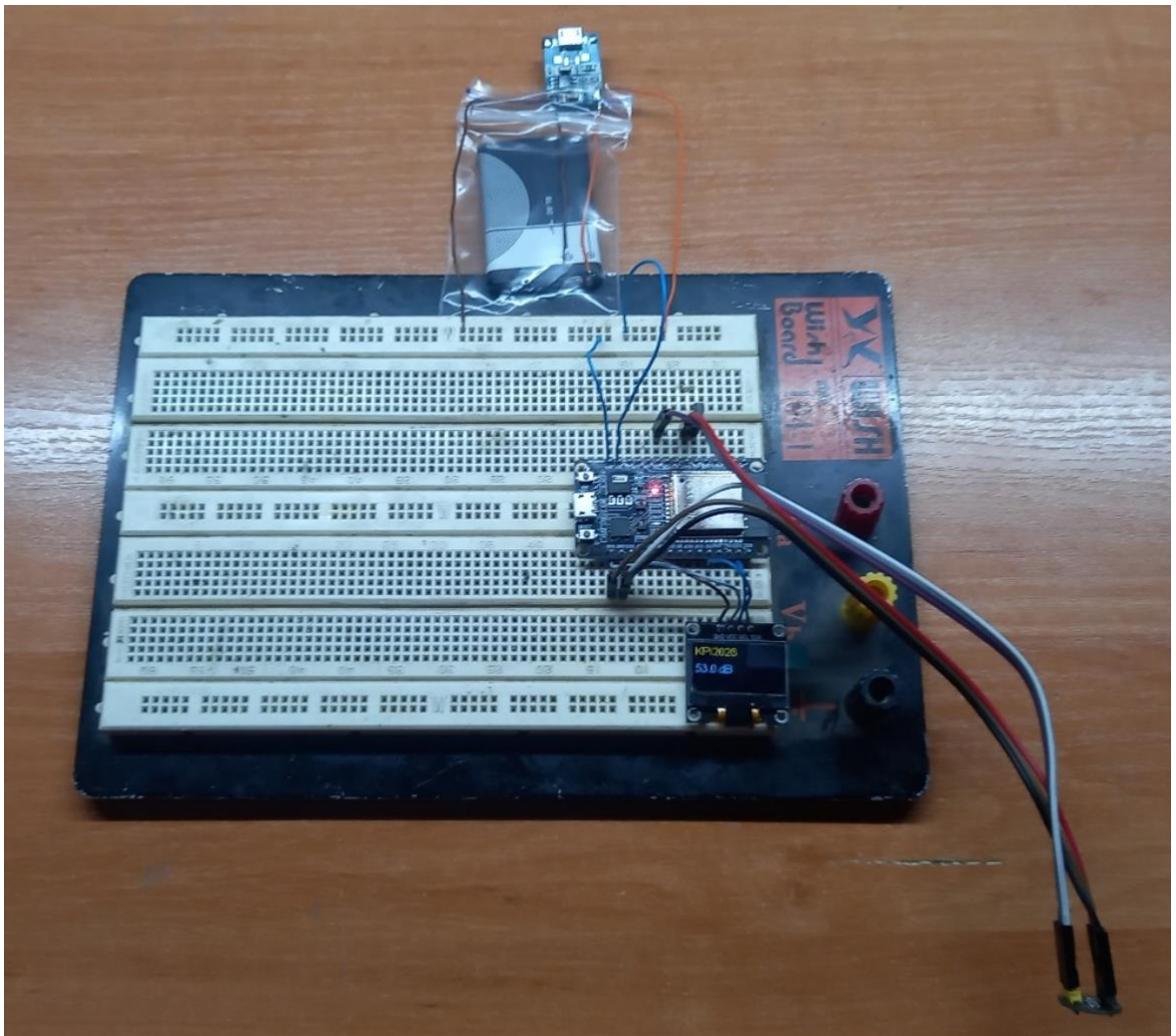


Рисунок 2.20 – Зібраний макет пристрою

Після того, як макет пристрою було зібрано на макетній платі, необхідно написати програмний код для взаємодії та обробки компонентів пристрою. Було прийнято рішення писати код для взаємодії елементів в порядку дисплей – мікрофон – графічний інтерфейс (керування кнопкою та веб-інтерфейс) –

сигналізація перевищення заданого користувачем рівня шуму (керування світлодіодом). Код, що відповідає за підключення та роботу дисплею включає в себе підключення бібліотеки для роботи з дисплеєм, ініціалізацію дисплею, створення списку екранів та роботу з виведенням вимірних даних і переключення між різними екранами (рис. 2.21).

```
#include <Wire.h> //бібліотека для роботи з I2C
#include <SSD1306Wire.h> //бібліотека для SSD1306
SSD1306Wire display(0x3c, SDA, SCL); //ініціалізація дисплею
enum MenuState { //створення списку екранів користувача
    MAIN_SCREEN,
    MENU_AWEIGHT,
    MENU_ALARM,
    MENU_THRESHOLD
};

MenuState menuState = MAIN_SCREEN; //вибір початкового при повному запуску пристрою

display.init(); //ініціалізація екрану
display.flipScreenVertically(); //перевертання даних на дисплеї
display.setFont(ArialMT_Plain_16); //вибір шрифту

display.drawString(0, 0, "Shumomir"); //вказування даних, які необхідно вивести на дисплей
display.drawString(0, 24, "Starting...");
display.drawString(0, 48, "KPI2026");
display.display(); //команда виведення заданих даних на дисплей

display.clear(); //очистити дисплей

if (menuState == MAIN_SCREEN) { //залежно від вибраного екрану
    display.drawString(0, 0, "KPI2026"); //вказування даних про звукове навантаження
    display.drawString(0, 24, String(currentDB, 1) + " dB");
}

else if (menuState == MENU_AWEIGHT) { //залежно від вибраного екрану
    display.drawString(0, 0, "A-Weight"); //вказування даних про вибір режиму фільтрації
    display.drawString(0, 24, aWeightEnabled ? "ON" : "OFF");
    display.drawString(0, 48, "Hold: toggle");
}

else if (menuState == MENU_ALARM) { //залежно від вибраного екрану
    display.drawString(0, 0, "Alarm"); //вказування даних про статус попередження про перевищення рівня шуму
    display.drawString(0, 24, alarmEnabled ? "ON" : "OFF");
    display.drawString(0, 48, "Hold: toggle");
}

else if (menuState == MENU_THRESHOLD) { //залежно від вибраного екрану
    display.drawString(0, 0, "Threshold"); //вказування даних про обраний максимальний дозволений рівень шуму
    display.drawString(0, 24, String(noiseThreshold, 1));
    display.drawString(0, 48, "Hold: +1 dB");
}

display.display(); //виведення даних на дисплей
```

Рисунок 2.21 – Частина коду, що відповідає за роботу з дисплеєм

Для початку роботи з дисплеєм необхідно ініціалізувати бібліотеки Wire.h для роботи з інтерфейсом підключення I2C та SSD1306Wire.h для роботи з безпосередньо дисплеєм SSD1306. Після цього за допомогою цих ж бібліотек вказуються параметри дисплею, а саме піни його ліній підключення та адресу I2C. Після цього створюється список екранів користувача та обирається один з

екранів, який буде початковим при повному запуску пристрою. Далі відбувається ініціалізація дисплею та виведення на нього екрану завантаження, який включає в себе надписи “Shumomir”, “Starting...” та “KPI2026”, що виводяться один за одним в стовпчик (рис. 2.5).

Після повного запуску пристрою за замовчуванням показується стан шумового навантаження в даний момент (`menuState == MAIN_SCREEN`) (рис.2.6), але за допомогою короткого натискання кнопки можна переключитись на екрани налаштувань, а саме на екран налаштування режиму фільтрації (рис.2.9), екрану увімкнення попередження про перевищення заданим користувачем рівня допустимого шуму (рис.2.8) та екрану задавання цього рівня (рис.2.7).

Після цього необхідно написати код для роботи з мікрофоном та вимірювання рівня шуму. Це є найважливішою частиною коду та включає в себе кілька різних математичних формул для перетворення необроблених даних з мікрофону в величину дБ. Як було описано раніше, необроблені дані мікрофона це просто величина відхилення мембрани, тому очікуваним результатом виводу необроблених даних є набір чисел, що мають майже однакову модульну величину з невеликим відхиленням (наприклад, -150, 148, -155, 122).

До цих даних застосовуються формули обробки сигналу, а саме формула середнього квадратичного (RMS) [48] та формула переведення в децибели [49].

$$V_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

де x_i — миттєві значення сигналу, а N — кількість відліків у вибірці.

$$L = 20 \log_{10} \left(\frac{V_{RMS}}{V_{ref}} \right)$$

Формула RMS в цьому випадку використовується для того, щоб визначити середнє значення шуму за деякий невеликий період часу, бо якщо брати напряму виміряні величини та відразу їх обробляти окремо, то це спричинить велику кількість стрибків в виміряному рівню шуму та надмірне навантаження на мікропроцесор. Також розмір масиву має забезпечувати достатню швидкість

вимірювань, щоб бути меншою ніж швидкість оновлення дисплею, що становить 2 Гц, тобто 2 рази за секунду. Це означає, що масив даних не має перевищувати кількість вимірів з мікрофону/2 вимірів. Для створення шумоміру було обрано масив даних величиною в 24000 вимірів з частотою вимірювання в 48000 одиниць за секунду. Обумовлено це тим, що навіть якщо збільшити частоту вимірювання рівня шуму, швидкість оновлення дисплею не дасть змогу нам його побачити, тому в цьому немає сенсу.

Після цього, значення RMS необхідно підставити в формулу для перетворення в децибели, але також необхідно знайти значення V_{ref} . Це значення для мікрофонів вираховується при впливу на нього звуку рівнем в 94дБ [46] та його усереднення за допомогою формули RMS. Його необхідність полягає в тому, що мікрофон ICS43434 не має вбудованого еталону, який прив'язував би відсутність шуму до 0 дБ, тому якщо не вказати це значення, то виміряний рівень шуму буде значно здвинутий по логарифмічній шкалі та, в результаті, буде невірним. Для цього мікрофона при дії на нього звуку силою в 94 дБ було виміряно значення RMS в 8388607 одиниць, що і стане еталонним значенням V_{ref} .

В результаті після обробки сигналу мікрофон повинен правильно виводити розраховані дані в дБ, що можна підтвердити експериментальним чином, використавши інші пристрої та програми, наприклад, телефон для генерації звуку. Тому наступним етапом стало написання коду для роботи з мікрофоном і розрахунку цих формул та перенесення його в мікроконтролер (рис.2.21).

```

#include <driver/i2s.h> //бібліотека для роботи з I2S
#include <math.h> //бібліотека для розрахування математики
#define I2S_WS 32 //налаштування пінів I2S
#define I2S_SCK 27
#define I2S_SD 33
#define I2S_PORT I2S_NUM_0

#define SAMPLE_RATE 48000 //налаштування швидкості вимірювання та розмір буферу сирих даних
#define BUFFER_SIZE 4096
int32_t samples[BUFFER_SIZE];

const double MIC_REF_AMPL = 8388607.0; //значення Vref мікрофону
float currentDB = 0; //створення змінних для обробки та виведення рівня шуму
void setupI2S() { //ініціалізація та налаштування I2S
    i2s_config_t config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
        .sample_rate = SAMPLE_RATE,
        .bits_per_sample = I2S_BITS_PER_SAMPLE_32BIT,
        .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_I2S,
        .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
        .dma_buf_count = 8,
        .dma_buf_len = 512,
        .use_apll = false
    };

    i2s_pin_config_t pins = {
        .bck_io_num = I2S_SCK,
        .ws_io_num = I2S_WS,
        .data_out_num = -1,
        .data_in_num = I2S_SD
    };

    i2s_driver_install(I2S_PORT, &config, 0, NULL);
    i2s_set_pin(I2S_PORT, &pins);
}

void loop() { //ініціалізація I2S
    size_t bytesRead;
    i2s_read(I2S_PORT, samples, sizeof(samples), &bytesRead, portMAX_DELAY); //зчитування даних з мікрофону

    int count = bytesRead / sizeof(int32_t); //вирахування кількості зчитаних даних
    for (int i = 0; i < count; i++) { //розрахунок RMS
        sumSq += x * x;
        samplesCount++;
    }

    #define WINDOW_SIZE (SAMPLE_RATE / 2) //змінна, що дозволяє змінити кількість накопичених семплів, з яких відбудеться розрахунок в дБ

    if (samplesCount >= WINDOW_SIZE) {
        double mean = sumSq / samplesCount;
        double rms = sqrt(mean); //розрахунок RMS

        float LAeq = //розрахунок значення в дБ
            20.0 * log10(rms/MIC_REF_AMPL) + CAL_DB;

        currentDB = LAeq;
    }
}

```

Рисунок 2.22 – Частини коду, що відповідають за роботу з мікрофоном

Після перевірки працездатності коду необхідно написати код для взаємодії з ним. Він включає в себе роботу з кнопкою, світлодіодом та веб-сторінкою.

Для початку було написано код для взаємодії з кнопкою, що дозволить змінювати налаштування пристрою. Під час написання коду було враховано явище брязкіту контакту кнопки, що може виникнути в момент її натискання і призводити до багатократного спрацювання. Для цього в коді і було вказано, що коротке натискання кнопки – це натискання протягом 200-800мс. Такий підхід дозволяє повністю прибрати можливість хибного спрацювання кнопки. Для довгого натискання необхідно тримати кнопку більш як 800мс., що дозволить змінити користувачеві налаштування пристрою (рис.2.22).

```

#define BTN_PIN 14 //налаштування піна для кнопки 25
unsigned long holdStart = 0; //створення змінних для кнопки
unsigned long lastRepeat = 0;
bool holding = false;
bool lastBtn = 1;
unsigned long btnPressTime = 0;

void handleButton() { //створення функції для обробки натискання кнопки

    bool btn = digitalRead(BTN_PIN);

    if (lastBtn == HIGH && btn == LOW) { //реєстрація моменту натиску на кнопку
        btnPressTime = millis();
    }

    if (lastBtn == LOW && btn == HIGH) { //реєстрація моменту відпускання кнопки

        unsigned long dt = millis() - btnPressTime; //вимірювання кількості часу, поки кнопка була натиснута

        if (dt > 800) { //дії переключення налаштувань при довгому натисканні (0.8сек)

            if (menuState == MENU_THRESHOLD) {
                noiseThreshold += 1;
                if (noiseThreshold > 100) noiseThreshold = 30;
            }

            else if (menuState == MENU_ALARM) {
                alarmEnabled = !alarmEnabled;
            }

            else if (menuState == MENU_AWEIGHT) {
                aWeightEnabled = !aWeightEnabled;
            }

            else if (menuState == MAIN_SCREEN) {
                menuState = MENU_AWEIGHT;
            }
        }

        else if (dt > 200) { //дії переключення екранів при короткому натисканні (0.2сек)

            if (menuState == MAIN_SCREEN) menuState = MENU_AWEIGHT;
            else if (menuState == MENU_AWEIGHT) menuState = MENU_ALARM;
            else if (menuState == MENU_ALARM) menuState = MENU_THRESHOLD;
            else if (menuState == MENU_THRESHOLD) menuState = MAIN_SCREEN;
        }
    }

    lastBtn = btn;
}

handleButton(); //перевірка стану кнопки

```

Відбувається одноразово під час завантаження пристрою

Розраховується постійно під час роботи пристрою

Рисунок 2.23 – Частина коду, що відповідають за роботу з кнопкою

Код світлодіода складається з кількох рядків, а саме налаштування його підключення до плати мікроконтролера та умови його спрацювання (рис.2.23). Він спрацьовуватиме лише тоді, коли задане кнопкою значення noiseThreshold (рис.2.24) буде перевищено.

```

#define LED_PIN 2 //налаштування піна для світлодіода
if (alarmEnabled && currentDB > noiseThreshold) { //перевірка виконання умови для спрацювання світлодіода
    digitalWrite(LED_PIN, HIGH);
} else {
    digitalWrite(LED_PIN, LOW);
}

void handleButton() { //створення функції для обробки натискання кнопки

```

Розраховується постійно під час роботи пристрою

Рисунок 2.24 – Частина коду, що відповідають за роботу світлодіода

Останнім етапом написання коду основних функцій пристрою є створення веб-сторінки для перегляду графіку вимірювань пристрою. Для цього було використано бібліотеки для роботи з вбудованим WiFi модулем в ESP32 [50] та бібліотеки для створення веб-серверу [51]. Для створення цього типу інтерфейсу необхідно налаштувати ряд параметрів, таких як створення бездротової мережі, створення веб сторінки, передача даних на веб-сторінку в реальному часі та створення самого графіку, який показуватиме виміряні дані (рис.2.25).

```
const char* ssid = "SHUMOMIRWEB"; //створення назви та паролю для бездротової мережі
const char* password = "levpes";
WebServer server(80); //вибір порта для веб-сервера
#define HISTORY_SIZE 100 //створення масиву даних для побудови графіку
float dbHistory[HISTORY_SIZE];
int historyIndex = 0;
void handleButton() { //створення функції для обробки натискання кнопки
//створення веб-сторінки через HTML
const char PAGE[] = R"rawliteral(
<!DOCTYPE html>
<html>
тут знаходиться код сторінки та побудови графіку
</html>
)rawliteral";

void handleRoot() { //функція перевірки входу на веб-сторінку та передача даних про неї
server.send(200, "text/html", PAGE);
}

void handleData() { //функція передачі даних на сторінку
String json = "{";
json += "\"db\":" + String(currentDB, 1);
json += "}";
server.send(200, "application/json", json);
}

void setupWeb() { //функція створення бездротової мережі та маршрутів сторінки
WiFi.softAP(ssid, password);

server.on("/", handleRoot);
server.on("/data", handleData);

server.begin();
}

setupWeb();
server.handleClient();
```

Виконується
одноразово при
завантаженні
пристрою

Виконується
постійно під час
роботи
пристрою

Рисунок 2.25 – Частини коду, що відповідають за веб-інтерфейс

Після цього необхідно перевірити працездатність пристрою, порівнявши його вихідні значення з іншими вимірювальними пристроями та зробити фізично і перенести всі елементи на друковану плату.

Для виготовлення плати розглядались три різні варіанти: витравлювання, замовлення готової плати онлайн та виготовлення плати на ЧПУ-станку.

Перший варіант є оптимальним та підходить для виконання в домашніх умовах, оскільки для нього необхідно мати при собі лише текстоліт, засіб для захисту міді від кислоти (засіб для рисування плати) та кислоту, якою можна витравити лишню мідь, наприклад, мідний купорос CuSO_4 або азотну кислоту HNO_3 . Процес виготовлення плати таким чином полягає в підготовці схеми друкованої плати, перенесення її на лист текстоліту з мідним покриттям та витравлення лишніх ділянок міді.

Перенесення схеми на текстоліт можна робити кількома шляхами, перший – це захистити необхідні місця (мідні доріжки, посадочні місця і т.л.) за допомогою засобу, на який не впливатиме кислота. Це є найшвидшим та найпростішим рішенням, але воно не забезпечує необхідну точність для створення посадочних місць для компонентів, в яких ніжки знаходяться дуже близько одна до одної. Наприклад, в контролера ESP32 з обох сторін по 15 ніжок, між центрами яких дистанція в 0.254 см, тому ручне нанесення захисної плівки тут майже неможливе.

Іншим варіантом є друк порошковим чорнилом схеми плати на лист глянцевого паперу та перенесення чорнила на текстоліт за допомогою розігрівання, наприклад, праски. Такий метод дозволяє дещо сповільнити роз'їдання доріжок в тих місцях, де нанесений порошковий покрив і є чудовим варіантом для виготовлення друкованих плат в домашніх умовах.

Іншим варіантом виготовлення плати є її купівля, проте це може бути затратно з погляду на ціну та час доставки, тому такий вибір хоч і дає неймовірну точність в створенні плати, але не завжди може бути підходящим варіантом.

Останнім методом виготовлення є виготовлення плати на ЧПУ-станку, що дає велику точність під час виготовлення, дозволяє надійно робити доріжки шириною в 1мм і менше, що є майже неможливим в ручному перенесенні, але важливим при тісному компонуванні, хоч і не є загальнодоступним варіантом у зв'язку з великими цінами на такі станки.

У випадку створюваного пристрою було використано метод з ЧПУ-станком у зв'язку з його наявністю (рис.2.26).



Рисунок 2.26 – Виготовлений пристрій на друкованій платі

Після перенесення пристрою на друковану плату необхідно помістити все в корпус, раніше виготовлений на фотополімерному принтері. Кріплення елементів відбувається за допомогою гвинтів з латунними термогайками та двохстороннього скотчу.

Встановлення в корпус термогайок відбувається за допомогою попереднього їх нагріву та притискання до точки в корпусі, де її потрібно встановити. Для нагрівання та встановлення термогайки використовувався паяльник з рівним конічним жалом, на який надівалась термогайка та під прямим кутом вставлялась в необхідне місце.

За допомогою двохстороннього скотчу всередині корпусу було закріплено акумуляторну батарею на призначене для неї місце в корпусі. Адгезивних властивостей скотчу виявилось достатньо для надійного закріплення АКБ всередині пристрою.

Після встановлення всіх термогайок, приклеювання батареї, прикручування головної плати пристрою та контролера заряду і повної збірки корпусу пристрою на мікрофон також було прикріплено захисний фільтр з поролону, що дає змогу перекрити пряме попадання вітру на мембрану мікрофону та таким чином значно понизити похибку вимірювань в місцях з рухомих потоком повітря, наприклад під кондиціонером чи надворі.

2.4 Перевірка працездатності системи та аналіз результатів

Після повного налаштування та збірки пристрою було проведено його перевірку. Для цього необхідно обрати кілька джерел стабільного звуку та виміряти його величину за допомогою різних засобів, включаючи шумомір. Генераторами шуму було обрано кілька побутових приборів, а саме фен ARDESTO Violet Pro [58] та міксер Bosch Clevermixx [59], також було виміряно рівень шумового навантаження в тихій кімнаті. Вимірювання здійснювались на різних відстанях від джерел шуму, але відстані до кожного приладу для вимірювання шуму залишались однаковими відносно джерела.

Приладами для реєстрації виміряного рівня шумового навантаження було обрано розроблений шумомір, телефон Redmi A5 з встановленою програмою для вимірювання рівня шуму та, в якості еталонного пристрою, для вимірювання було обрано шумомір I – класу SCARLET ST-15D [60] (рис. 2.27), який відповідає стандартам IEC 61672-1:2013 [17] та IEC 61260-1:2014 [61]. Його використання дало змогу точно оцінити відхилення результатів вимірювань розробленого пристрою в порівнянні з професійними варіантами.



Рисунок 2.27 – Шумомір SCARLET ST-15D

Для початку було виміряно та порівняно рівень шумового навантаження в тихій кімнаті без гучних пристроїв (рис. 2.28).

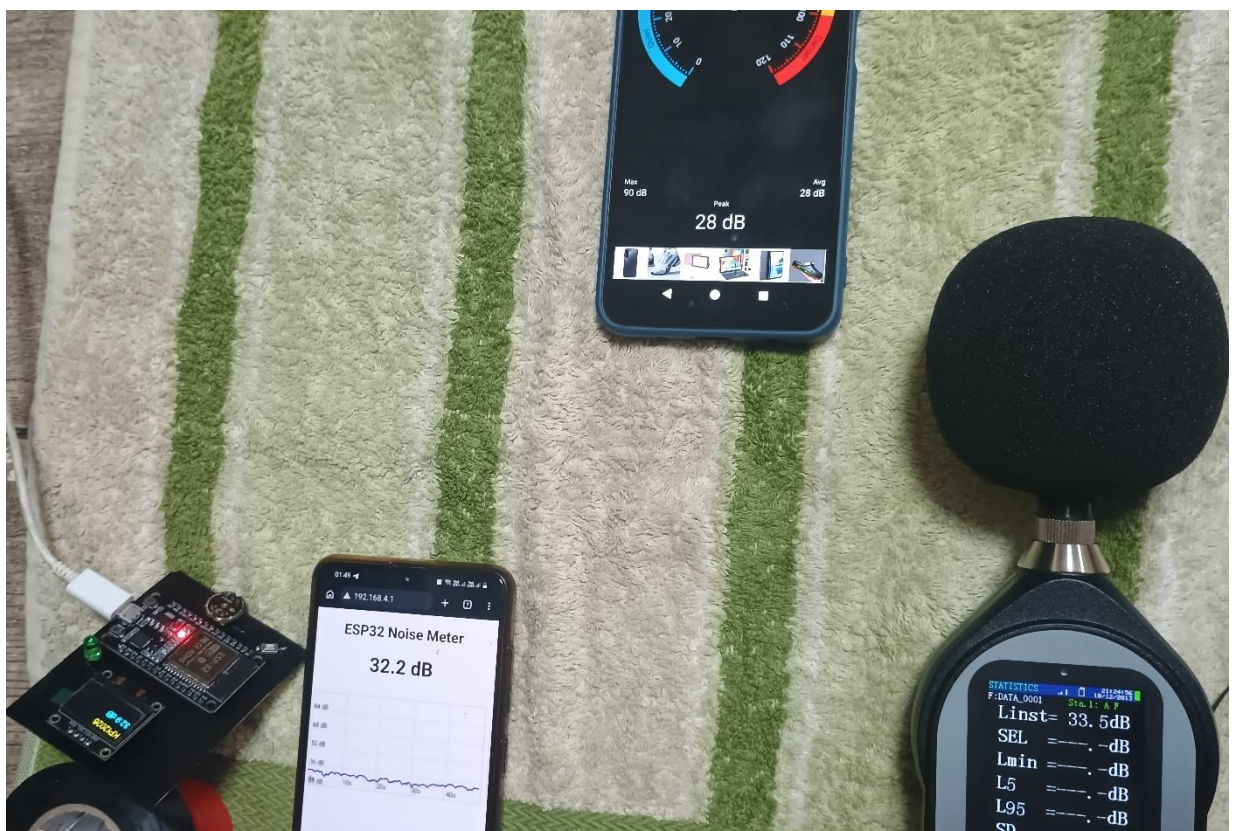


Рисунок 2.28 – Вимірювання рівня шуму в тихій кімнаті

Таблиця 2.1 – Результати вимірювання рівня шуму в тихій кімнаті

Пристрій для вимірювання	Розроблений шумомір	Мобільний телефон	SCARLET ST-15D
Результати вимірювань	32.2±1дБ	28±1.5дБ	33.5±0.3дБ

В результаті вимірювань можна спостерігати, що на низьких діапазонах шумового навантаження максимальне відхилення розробленої автоматизованої системи відрізняється від показів еталонного шумоміру не більш як на 2.6 дБ, що задовольняє умови технічного завдання (точність вимірювання до 3 дБ). В той ж час виміряне значення телефоном відрізняється від еталонного на 3.7 - 7.3 дБ, що є значним відхиленням.

Після цього було порівняно вимірюваний рівень шуму, що генерується роботою міксеру Bosch Clevermixx. Вимірювання здійснювалось безпосередньо біля джерела звуку, направивши всі пристрої мікрофонами в бік генератора шуму (рис. 2.29).



Рисунок 2.29 – Вимірювання рівня шуму, створюваного міксером Bosch Clevermixx

Таблиця 2.2 – Результати вимірювання шуму від міксеру Bosch Clevermixx

Пристрій для вимірювання	Розроблений шумомір	Мобільний телефон	SCARLET ST-15D
Результати вимірювань	76.7 ± 0.8 дБ	73 ± 1 дБ	77.4 ± 0.2 дБ

В результаті цього вимірювання можна спостерігати, що в середньому діапазоні рівня шуму у всіх пристроїв значно понизилась похибка вимірювань, що дало змогу розробленому шумоміру досягти максимальної похибки від еталонного пристрою у 1.7 дБ, в той час як вимірювання телефону показало максимальне відхилення від еталону в 5.6 дБ. Ці результати показують, що вимірювання на середніх рівнях шуму для всіх пристроїв є значно стабільнішим, ніж на низьких рівнях шуму.

Після цього було виміряно шумовий вплив фену для волосся ARDESTO Violet Pro безпосередньо біля джерела шуму (рис. 2.30) та на дистанції в 1 метр від нього (рис. 2.31).

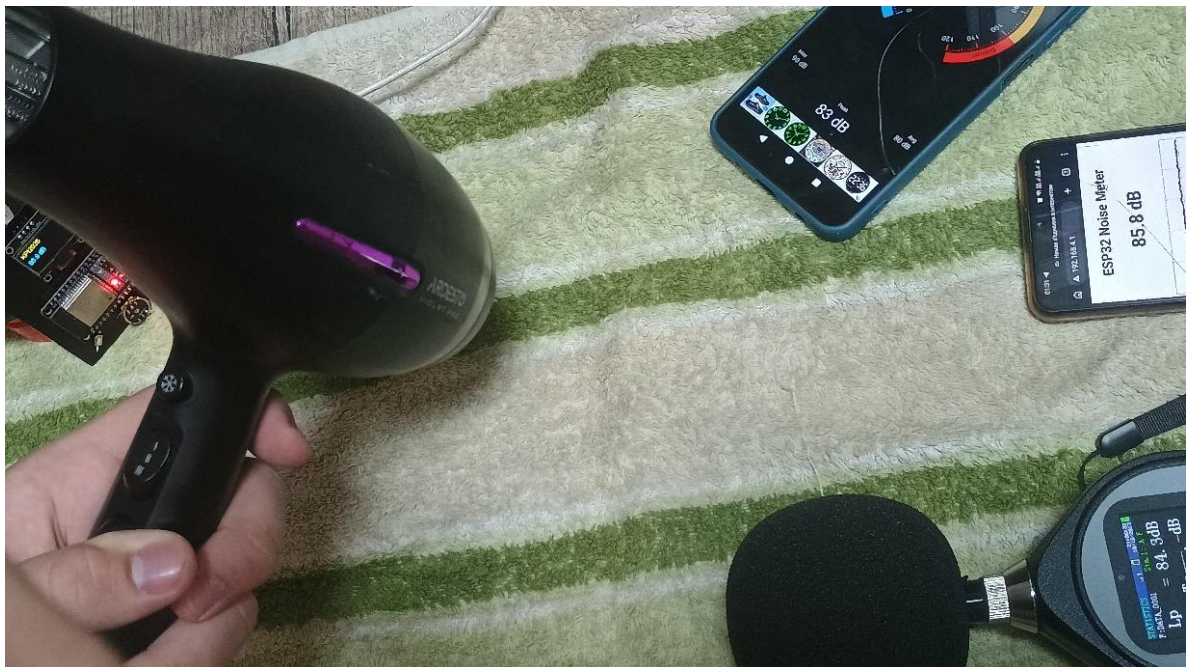


Рисунок 2.30 – Вимірювання рівня шуму створюваного феном ARDESTO Violet Pro



Рисунок 2.31 – Вимірювання рівня шуму, створюваного феном ARDESTO Violet Pro на дистанції в 1 м.

Таблиця 2.3 – Результати вимірювання шуму від фену ARDESTO Violet Pro

Пристрій для вимірювання	Розроблений шумомір	Мобільний телефон	SCARLET ST-15D
Результати вимірювань біля генератора шуму	85.8±0.9дБ	83±1.3дБ	84.3±0.3дБ
Результати вимірювань на дистанції в 1м.	78.1±1.2дБ	70±3дБ	77.2±0.4дБ

В результаті вимірювань можна помітити, що на вищих рівнях шуму показники розробленого шумоміру починають перевищувати показники еталонного приладу для вимірювання рівня шуму, тому похибка вимірювань сягає 2.7 дБ при вимірюванні безпосередньо біля джерела шуму та 2.5 дБ на дистанції в 1 м. від джерела шуму. В той ж час мобільний телефон непогано показав себе при вимірюванні рівня шуму безпосередньо біля джерела, з

похибкою відносно еталону до 2.9 дБ, але на дистанції в 1 м. до джерела звуку вимірювання почали сильно коливатись та відображатись невірною, що дає змогу зробити висновок, що даний пристрій не може вірно вимірювати рівень звуку на великих дистанціях.

Після проведених вимірювань було підтверджено, що створений прилад задовольняє умову технічного завдання щодо точності вимірювань не менше 3 дБ та має змогу коректно вимірювати як низькі рівні шуму, так і високі на різних дистанціях, в той час як використання мобільних телефонів показало себе не ефективним для використання у вимірюваннях рівня шуму, особливо якщо об'єктом вимірювання є не одне джерело, а середовище.

2.5 Висновки до другого розділу

В другому розділі було описано процес створення автоматизованої системи для вимірювання рівня шуму в приміщенні, її налагодження та тестування.

Для пристрою було сконструйовано печатну плату за допомогою ЧПУ-станка та попередньо розробленої схеми підключення елементів пристрою, що забезпечило велику точність у створенні друкованої плати.

Також було розроблено графічний інтерфейс користувача, що складався з фізичного засобу виведення інформації та налаштувань і веб-інтерфейсу для перегляду графіку вимірюваних даних.

Після цього було налагоджено апаратну частину, що включало в себе підключення всіх елементів до плати та програмну частину, виконання якої дало змогу вимірювати, перетворювати та відображати на графічному інтерфейсі дані про силу шуму.

В кінці було перевірено працездатність системи, порівняно результати її вимірювань з професійним пристроєм для вимірювання рівня шуму та звичайним телефоном і проаналізовано отримані результати, що дало змогу запевнитись у точності та правильності вимірюваних значень.

В результаті проведених робіт було розроблено та налагоджено автоматизовану систему для вимірювання рівня шуму, яка задовольняє всі вимоги згідно технічного завдання та забезпечує стабільне вимірювання величини кількості шуму на коротких та довгих дистанціях.

ВИСНОВКИ

Під час виконання дипломної роботи було проведено аналіз сучасних методів вимірювання рівня шуму в навколишньому середовищі, досліджено їх класифікації та проведено аналіз існуючих засобів для виконання цього завдання. Основною метою було дослідження принципів та алгоритмів вимірювання звуку для подальшого їх використання під час створення власної автоматизованої системи.

У першому розділі було розглянуто історію досліджень та створення систем для дослідження звуку, їх класифікацію за методами вимірювання та принципами роботи, проведено огляд існуючих рішень для вимірювання рівня шуму та підібрано елементи для створення власної автоматизованої системи для вимірювання рівня шуму.

У другому розділі розглянуто процес створення апаратної та програмної частин для автоматизованої системи, проєктування складальних одиниць системи, її налагодження та проведено перевірку працездатності. Було досліджено процес цифрової обробки звукового сигналу, зокрема дискретизацію, обчислення середньоквадратичного значення сигналу та визначення рівня звукового тиску в децибелах. Також проведено калібрування пристрою шляхом порівняння результатів вимірювань з еталонним шумоміром та визначення коригувального коефіцієнта.

За результатами експериментальних досліджень було встановлено, що розроблений шумомір забезпечує стабільне вимірювання рівнів шуму в діапазоні, достатньому для контролю акустичних умов побутових і виробничих приміщень та відповідному технічному завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шум: як він впливає на організм людини? Чи можна знизити його рівень? | Південне міжрегіональне управління Державної служби з питань праці [Електронний ресурс]. — Режим доступу: <https://pd.dsp.gov.ua/news/shum-iak-vin-vplyvaie-na-orhanizm-liudyny-chy-mozhna-znyzyty-ioho-riven/>
2. Прилад для вимірювання шуму — Вікіпедія [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Sound_level_meter
3. Швидкість звуку — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Швидкість_звуку
4. Marin Mersenne – Mathematics and Universal Harmony [Електронний ресурс]. — Режим доступу: https://mathshistory.st-andrews.ac.uk/SH/merсенне_sh.pdf
5. A Brief History of the Speed of Sound [Електронний ресурс]. — Режим доступу: https://acoustics.org.nz/wp-content/uploads/Hannah_L_NZA2006.pdf
6. The History of Audio and Sound Measurement | Pro Audio Encyclopedia [Електронний ресурс]. — URL: <https://proaudioencyclopedia.com/the-history-of-audio-and-sound-measurement>
7. The History of Sound Waves | The Music Around us [Електронний ресурс]. — Режим доступу: <https://musicalsoundwaves.wordpress.com/the-history-of-sound-waves/>
8. Зубчате колесо Савара — Вікіпедія [Електронний ресурс]. — Режим доступу: https://ru.wikipedia.org/wiki/Зубчатое_колесо_Савара
9. Jules Lissajous – Linda Hall Library [Електронний ресурс]. — Режим доступу: <https://www.lindahall.org/about/news/scientist-of-the-day/jules-lissajous/>
10. Диск Релея — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Диск_Релея
11. Вугільний мікрофон — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Вугільний_мікрофон

12. Galvanometers | The Physics Teacher | AIP Publishing Library [Електронний ресурс]. — Режим доступу: <https://pubs.aip.org/aapt/pte/article-abstract/35/7/423/271994/Galvanometers>
13. Sound Recording Research at Bell Labs [Електронний ресурс]. — Режим доступу: <https://www.aes-media.org/historical/html/recording.technology.history/bell-labs.html>
14. Цей день в історії : 23 грудня 1947 : Винайдення транзистора [Електронний ресурс]. — Режим доступу: <https://www.jnsm.com.ua/h/1223P>
15. Вплив шуму на наш слух | Долинська міська рада [Електронний ресурс]. — Режим доступу: <https://rada-dolyna.gov.ua/news/vplyv-shumu-na-nash-slukh>
16. Про затвердження Державних санітарних норм допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови | від 22.02.2019 № 463 [Електронний ресурс]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0281-19#n14>
17. IEC-61672-1-2013 [Електронний ресурс]. — Режим доступу: <https://cdn.standards.iteh.ai/samples/17900/df52d949fc904f329404e965b6268258/IEC-61672-1-2013.pdf>
18. Sound Level Meters: How They Work, Best Models & More | Guide [Електронний ресурс]. — Режим доступу: <https://mfe-is.com/sound-level-meters/>
19. CCOHS: Noise - Measurement of Workplace Noise [Електронний ресурс]. — Режим доступу: https://www.ccohs.ca/oshanswers/phys_agents/noise/noise_measurement.html
20. Sound level meter basics: how does it work? | What is used for? [Електронний ресурс]. — Режим доступу: <https://svantek.com/academy/sound-level-meter/>
21. TM-102 / TM-103 Sound Level Meter Manual [Електронний ресурс]. — Режим доступу: <https://www.egeratedocuments.com/markalar/tenmars/Manual/tm102-tm103-manual.pdf>

22. PAA3X User Manual [Электронный ресурс]. — Режим доступа: https://www.phonic.com/images/products-II/PAA3X/um_PAA3X_en_es.pdf?type=file
23. Operation Manual for DS-2013SD [Электронный ресурс]. — Режим доступа: https://inspectusa.com/lutron/manuals/ds-2013sd-manual.pdf?srsId=AfmBOordbOIyuv9uT0w_ilPWh8ltX6E1h_OwGzCbrTjAp_hImA_8d8xLo
24. About Arduino | Arduino [Электронный ресурс]. — Режим доступа: <https://www.arduino.cc/en/about/>
25. Arduino — Вікіпедія [Электронный ресурс]. — Режим доступа: <https://uk.wikipedia.org/wiki/Arduino>
26. STM32 Arm Cortex MPUs [Электронный ресурс]. — Режим доступа: <https://www.st.com/en/microcontrollers-microprocessors/stm32-arm-cortex-mpus/documentation.html>
27. ESP32 Series Datasheet [Электронный ресурс]. — Режим доступа: https://documentation.espressif.com/esp32_datasheet_en.pdf
28. INMP441 Datasheet [Электронный ресурс]. — Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/pdf/1244625/ETC1/INMP441.html>
29. EMMIC-ICS43434 Datasheet [Электронный ресурс]. — Режим доступа: <https://www.cdiweb.com/datasheets/invensense/emmic-ics43434-ds.pdf>
30. LCD1602 Datasheet [Электронный ресурс]. — Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/view/2215051/DFROBOT/LCD1602.html>
31. SSD1306 Datasheet [Электронный ресурс]. — Режим доступа: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
32. BL-5C Datasheet [Электронный ресурс]. — Режим доступа: <https://docs.rs-online.com/ab5a/0900766b80cf37fc.pdf>
33. TP4056 Datasheet [Электронный ресурс]. — Режим доступа: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>

- 34.I2S - Wikipedia [Електронний ресурс]. — Режим доступу:
<https://en.wikipedia.org/wiki/I2S>
35. I2C – Вікіпедія [Електронний ресурс]. — Режим доступу:
<https://uk.wikipedia.org/wiki/I2C>
- 36.EasyEDA - Online PCB design & circuit simulator [Електронний ресурс].
— Режим доступу: <https://easyeda.com>
- 37.Електронний посібник з дисципліни: БУДІВЕЛЬНЕ
МАТЕРІАЛОЗНАВСТВО [Електронний ресурс]. — Режим доступу:
https://elib.lntu.edu.ua/sites/default/files/elib_upload/Задорожнікова/page13.html
- 38.Пластик ПММА: що це? - Новини [Електронний ресурс]. — Режим
доступу: <https://ua.cnpolychemer.com/news/pmma-plastic-what-is-it-74766030.html>
- 39.Метали — Вікіпедія [Електронний ресурс]. — Режим доступу:
<https://uk.wikipedia.org/wiki/Метали>
- 40.Що таке ABS-пластик: властивості, плюси та мінуси, сфери застосування
[Електронний ресурс]. — Режим доступу:
<https://easy3dprint.com.ua/uk/shho-take-abs-plastik/>
- 41.Фотополімерна смола: що це таке в 2025 [Електронний ресурс]. — Режим
доступу: <https://easy3dprint.com.ua/uk/shcho-take-photopolymer-resin/>
- 42.GitHub - ThingPulse/esp8266-oled-ssd1306: Driver for the SSD1306 and
SH1106 based 128x64, 128x32, 64x48 pixel OLED display running on
ESP8266/ESP32 · GitHub [Електронний ресурс]. — Режим доступу:
<https://github.com/ThingPulse/esp8266-oled-ssd1306>
- 43.Frequency Weightings - A-Weighted, C-Weighted or Z-Weighted
[Електронний ресурс]. — Режим доступу:
<https://www.noisemeters.com/help/faq/frequency-weighting>
- 44.arduino-esp32/libraries/WiFi/src/WiFi.h at master · espressif/arduino-esp32 ·
GitHub [Електронний ресурс]. — Режим доступу:

- <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h>
45. arduino-esp32/libraries/WebServer/src/WebServer.h at master · espressif/arduino-esp32 · GitHub [Електронний ресурс]. — Режим доступу: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/src/WebServer.h>
46. ICS-43434 Datasheet [Електронний ресурс]. — Режим доступу: https://cdn-shop.adafruit.com/product-files/6049/6049_DS-000069-ICS-43434-v1.2.pdf
47. Частота Найквіста — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Частота_Найквіста
48. Середнє квадратичне — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Середнє_квадратичне
49. Децибел — Вікіпедія [Електронний ресурс]. — Режим доступу: <https://uk.wikipedia.org/wiki/Децибел>
50. WiFi | Arduino Documentation [Електронний ресурс]. — Режим доступу: <https://docs.arduino.cc/libraries/wifi/>
51. arduino-esp32/libraries/WebServer/src/WebServer.h at master · espressif/arduino-esp32 · GitHub [Електронний ресурс]. — Режим доступу: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/src/WebServer.h>
52. Software Solutions for 3D CAD, Design and Product Development | SOLIDWORKS [Електронний ресурс]. — Режим доступу: <https://www.solidworks.com>
53. DUB185RT User Manual [Електронний ресурс]. — Режим доступу: <https://www.notice-facile.com/en/manual/487027/makita%2Bdub185rt>
54. MAKITA 6305 User Manual [Електронний ресурс]. — Режим доступу: <https://makitauae.com/product/6305-drill>
55. BOSCH GBH2-26DFR User Manual [Електронний ресурс]. — Режим доступу: <https://www.manuals.co.uk/bosch/gbh-2-26-dfr-professional/manual?p=18>

56. ZELMER ZVC415ST User manual [Електронний ресурс]. — Режим доступу: <https://www.manualslib.com/manual/1099542/Zelmer-Zvc415st.html?page=41>
57. SENKOR SHD 0045BK User Manual [Електронний ресурс]. — Режим доступу: <https://manualsfile.com/product/k2xug8n8gs7.html>
58. ARDESTO Фен Violet PRO HD-Y222PRO [Електронний ресурс]. — Режим доступу: <https://prom.ua/p2473031945-ardesto-fen-violet.html>
59. Миксер Bosch CleverMixx MFQ3030 [Електронний ресурс]. — Режим доступу: <https://epicentrk.ua/shop/mikser-bosch-clevermixx-mfq3030.html>
60. Class 1 Sound Analyzer ST-15D with GPS [Електронний ресурс]. — Режим доступу: <https://scarlet-tech.com/products/class-1-sound-analyzer-st-15d/>
61. IEC-61260-1-2014 [Електронний ресурс]. — Режим доступу: <https://cdn.standards.iteh.ai/samples/13383/3c4ae3e762b540cc8111744cb8f0ae8e/IEC-61260-1-2014.pdf>
62. OSHA TWA Calculation: How to Calculate Noise Dose for 29 CFR 1910.95 | Soundtrace [Електронний ресурс]. — Режим доступу: <https://www.soundtrace.com/blog/time-weighted-average-twa-noise-dose-oshacalculation>
63. Файл: Phonautograph 1859.jpg — Вікіпедія [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Файл:Phonautograph_1859.jpg
64. Купити Шумомір ВШВ-003-М2 гурт, ціна | Система Оптимум [Електронний ресурс]. — Режим доступу: <https://www.systopt.com.ua/item-shumomir-vshv-003-m2>
65. Цифровий вимірювач рівня шуму (шумомір), фільтр А/С BENETECH GM1357, ціна 1786.84 ₴: купити на Prom.ua | Україна, Київ [Електронний ресурс]. — Режим доступу: <https://prom.ua/ua/p1130030719-tsifrovoj-izmeritel-urovnya.html>
66. Шумомір компактний (2-й клас) TENMARS TM-102 купити Контрольно-вимірювальна техніка SIMVOLT 288 [Електронний ресурс]. — Режим доступу: <https://simvolt.ua/shumomr-kompaktniy-2-y-klas-tenmars-tm-102.html/>

67. PAA3X [Електронний ресурс]. — Режим доступу: <https://www.phonic.com/index.php/products/test-instruments/product/106-paa3x>
68. Lutron DS-2013SD Noise Dosimeter – Karya Mandiri Techindo [Електронний ресурс]. — Режим доступу: <https://karyamandiritechindo.com/product/lutron-ds-2013sd-noise-dosimeter/>
69. Плата Arduino UNO: продаж, ціна в Україні. Набори та компоненти для самостійної збірки електроніки [Електронний ресурс]. — Режим доступу: <https://arduino kit.com.ua/ua/p2443129646-plata-arduino-uno.html>
70. STM32F103C8T6 ARM Cortex-M3 STM32 MicroUSB [Електронний ресурс]. — Режим доступу: <https://prom.ua/p1401062645-stm32f103c8t6-arm-cortex.html>
71. ESP32-DEVKITC-32E - РКС Компоненти - РАДІОМАГ [Електронний ресурс]. — Режим доступу: https://www.rcscomponents.kiev.ua/product/esp32-devkitc-32e_195673.html
72. INMP441 - MEMS мікрофон I2S - Купити в Україні, Харків [Електронний ресурс]. — Режим доступу: <https://1wire.com.ua/inmp441-mems-mikrofon-i2c.html>
73. ICS 43434 Digital MEMS Microphone Module I2S Port Small Size for Robotics and Intelligent Home Devices Low Cost Digital Microphone Module Board [Електронний ресурс]. — Режим доступу: <https://www.amazon.ca/Digital-Microphone-Robotics-Intelligent-Devices/dp/B0FH2YR1SK>
74. Індикатор РКІ 1602 СИНІЙ з підсвіткою LCD1602: продаж, ціна у Одесі. [Електронний ресурс]. — Режим доступу: <https://ledplus.com.ua/ua/p86789847-indikator-zhki-1602.html>
75. OLED 0.96 inch 128x64 SSD1306 I2C/SPI Display - REA012864D-01 – Raustar [Електронний ресурс]. — Режим доступу: <https://www.raustar-optronics.com/oled-graphic-display-module/oled-0.96-inch-i2c-spi.html>
76. Акумулятор BL-5C, без логотипа, для Nokia 1100, 1101, 1110, 1280, 1600, 1616, 202 Asha, (Li-ion 3.6V 1050мАч) (153344) купити в Україні

[Електронний ресурс]. — Режим доступу: <https://gsm-komplekt.ua/product/77200/?srsltid=AfmBOoolYlwYoYyPdHTEQ29Rj-pcnSefGEqlRberFJqi45jQqeb0g8we>

77.Модуль TP4056 microUSB із захистом контролер заряду Li-ion акумулятора 5V 1A [Електронний ресурс]. — Режим доступу: https://shcaf.com.ua/ua/p1779894372-modul-tp4056-microusb.html?srsltid=AfmBOopHGgUORV4G10YIOlHA98zQq35OK1_A7fBej4m4ctszc8uYkbSN

78.Introduction to the I2S Interface - Technical Articles [Електронний ресурс]. — Режим доступу: <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2s-interface>

79.GitHub - ThingPulse/esp8266-oled-ssd1306: Driver for the SSD1306 and SH1106 based 128x64, 128x32, 64x48 pixel OLED display running on ESP8266/ESP32 · GitHub [Електронний ресурс]. — Режим доступу: <https://github.com/ThingPulse/esp8266-oled-ssd1306>

80.

ДОДАТОК А

```

#include <driver/i2s.h>    //бібліотека для роботи з I2S
#include <math.h>          //бібліотека для розрахування математики
#include <WiFi.h>
#include <WebServer.h>
#include <Wire.h>          //бібліотека для роботи з I2C
#include <SSD1306Wire.h>  //бібліотека для SSD1306

const char* ssid = "SHUMOMIRWEB";    //створення назви та паролю для бездротової мережі
const char* password = "levpes";

WebServer server(80);                //вибір порта для веб-сервера

#define I2S_WS  32    //налаштування пінів I2S
#define I2S_SCK 27
#define I2S_SD  33
#define I2S_PORT I2S_NUM_0

#define SAMPLE_RATE 48000 //налаштування швидкості вимірювання та розмір буферу сирих даних
#define BUFFER_SIZE 4096

int32_t samples[BUFFER_SIZE];

SSD1306Wire display(0x3c, SDA, SCL); //ініціалізація дисплею

#define BTN_PIN 14    //налаштування піна для кнопки
#define LED_PIN 2     //налаштування піна для світлодіода

unsigned long holdStart = 0;    //створення змінних для кнопки
unsigned long lastRepeat = 0;
bool holding = false;

enum MenuState { //створення списку екранів користувача
    MAIN_SCREEN,
    MENU_AWEIGHT,
    MENU_ALARM,
    MENU_THRESHOLD
};

MenuState menuState = MAIN_SCREEN; //вибір початкового при повному запуску пристрою

float noiseThreshold = 70.0;
bool alarmEnabled = false;
bool aWeightEnabled = true;

bool lastBtn = 1;
unsigned long btnPressTime = 0;

#define CAL_DB 10.0
const double MIC_REF_AMPL = 420426.0; //значення Vref мікрофону

#define HISTORY_SIZE 100    //створення масиву даних для побудови графіку
float dbHistory[HISTORY_SIZE];
int historyIndex = 0;

float currentDB = 0;        //створення змінних для обробки та виведення рівня шуму

//створення веб-сторінки через HTML
const char PAGE[] = R"rawliteral(

```

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body { font-family: Arial; text-align: center; }
canvas { border: 1px solid #ccc; margin-top: 10px; }
</style>
</head>

<body>

<h2>ESP32 Noise Meter</h2>
<h1 id="db">-- dB</h1>

<canvas id="graph" width="300" height="150"></canvas>

<script>

let canvas = document.getElementById("graph");
let ctx = canvas.getContext("2d");

let data = [];
const MAX = 100;

function drawGraph() {

    ctx.clearRect(0, 0, canvas.width, canvas.height);

    let maxDB = 100;
    let minDB = 20;

    let w = canvas.width;
    let h = canvas.height;

    ctx.strokeStyle = "#ddd";
    ctx.lineWidth = 1;
    ctx.font = "10px Arial";
    ctx.fillStyle = "black";

    for (let i = 0; i <= 5; i++) {

        let y = (h / 5) * i;
        let dbVal = maxDB - (i * (maxDB - minDB) / 5);

        ctx.beginPath();
        ctx.moveTo(0, y);
        ctx.lineTo(w, y);
        ctx.stroke();

        ctx.fillText(dbVal.toFixed(0) + " dB", 2, y - 2);
    }

    let secondsTotal = data.length * 0.5;

    for (let i = 0; i <= 5; i++) {

        let x = (w / 5) * i;
        let sec = (secondsTotal / 5) * i;

        ctx.beginPath();
        ctx.moveTo(x, 0);
        ctx.lineTo(x, h);
    }

```

```

    ctx.stroke();

    ctx.fillText(sec.toFixed(0) + "s", x + 2, h - 5);
}

ctx.beginPath();
ctx.strokeStyle = "blue";
ctx.lineWidth = 2;

let step = w / MAX;

for (let i = 0; i < data.length; i++) {

    let x = i * step;
    let y = h - ((data[i] - minDB) / (maxDB - minDB)) * h;

    if (i === 0) ctx.moveTo(x, y);
    else ctx.lineTo(x, y);
}

ctx.stroke();
}

```

```

async function update() {

    try {
        let res = await fetch('/data');
        let json = await res.json();

        let db = json.db;

        document.getElementById("db").innerText =
            db.toFixed(1) + " dB";

        data.push(db);

        if (data.length > MAX) {
            data.shift();
        }

        drawGraph();

    } catch (e) {
        console.log("error", e);
    }
}

```

```
setInterval(update, 200);
```

```
</script>
```

```
</body>
```

```
</html>
```

```
)rawliteral";
```

```

неї void handleRoot() {                                     //функція перевірки входу на веб-сторінку та передача даних про
    server.send(200, "text/html", PAGE);
}

```

```

void handleData() {                                         //функція передачі даних на сторінку
    String json = "{}";
    json += "\"db\":\"" + String(currentDB, 1);
}

```

```

    json += "}";
    server.send(200, "application/json", json);
}

void setupWeb() {                                     //функція створення бездротової мережі та маршрутів
сторінки
    WiFi.softAP(ssid, password);

    server.on("/", handleRoot);
    server.on("/data", handleData);

    server.begin();
}

void setupI2S() {                                     //ініціалізація та налаштування I2S
    i2s_config_t config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
        .sample_rate = SAMPLE_RATE,
        .bits_per_sample = I2S_BITS_PER_SAMPLE_32BIT,
        .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_I2S,
        .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
        .dma_buf_count = 8,
        .dma_buf_len = 512,
        .use_apll = false
    };

    i2s_pin_config_t pins = {
        .bck_io_num = I2S_SCK,
        .ws_io_num = I2S_WS,
        .data_out_num = -1,
        .data_in_num = I2S_SD
    };

    i2s_driver_install(I2S_PORT, &config, 0, NULL);
    i2s_set_pin(I2S_PORT, &pins);
}

void handleButton() {                                 //створення функції для обробки натискання кнопки

    bool btn = digitalRead(BTN_PIN);

    if (lastBtn == HIGH && btn == LOW) {               //реєстрація моменту натиску на кнопку
        btnPressTime = millis();
    }

    if (lastBtn == LOW && btn == HIGH) {               //реєстрація моменту відпускання кнопки

        unsigned long dt = millis() - btnPressTime;   //вимірювання кількості часу, поки кнопка була натиснута

        if (dt > 800) {                                //дії переключення налаштувань при довгому натисканні (0.8сек)

            if (menuState == MENU_THRESHOLD) {
                noiseThreshold += 1;
                if (noiseThreshold > 100) noiseThreshold = 30;
            }

            else if (menuState == MENU_ALARM) {
                alarmEnabled = !alarmEnabled;
            }

            else if (menuState == MENU_AWEIGHT) {
                aWeightEnabled = !aWeightEnabled;
            }
        }
    }
}

```

```

    }

    else if (menuState == MAIN_SCREEN) {
        menuState = MENU_AWEIGHT;
    }
}

else if (dt > 200) { //дії переключення екранів при короткому натисканні (0.2сек)

    if (menuState == MAIN_SCREEN) menuState = MENU_AWEIGHT;
    else if (menuState == MENU_AWEIGHT) menuState = MENU_ALARM;
    else if (menuState == MENU_ALARM) menuState = MENU_THRESHOLD;
    else if (menuState == MENU_THRESHOLD) menuState = MAIN_SCREEN;
}
}

lastBtn = btn;
}

double sumSq = 0;
int samplesCount = 0;

void setup() {

    Serial.begin(115200);

    setupI2S(); //ініціалізація I2S
    setupWeb();

    display.init(); //ініціалізація екрану
    display.flipScreenVertically(); //перевертання даних на дисплеї
    display.setFont(ArialMT_Plain_16); //вибір шрифту

    display.drawString(0, 0, "Shumomir"); //вказування даних, які необхідно вивести на дисплей
    display.drawString(0, 24, "Starting...");
    display.drawString(0, 48, "KPI2026");
    display.display(); //команда виведення заданих даних на дисплей

    pinMode(BTN_PIN, INPUT_PULLUP);
    pinMode(LED_PIN, OUTPUT);
    digitalWrite(LED_PIN, LOW);

    delay(1000);
}

void loop() {

    server.handleClient();
    handleButton(); //перевірка стану кнопки

    size_t bytesRead;
    i2s_read(I2S_PORT, samples, sizeof(samples), &bytesRead, portMAX_DELAY); //зчитування даних з
мікрофону

    int count = bytesRead / sizeof(int32_t); //вираховування кількості зчитаних даних
    for (int i = 0; i < count; i++) { //розрахунок суми квадратичних значень в RMS
        // 1. normalize PCM
        float x = (float)(samples[i] >> 8);
        sumSq += x * x;
        samplesCount++;
    }
}

```

```

#define WINDOW_SIZE (SAMPLE_RATE / 2) //змінна, що дозволяє змінити
кількість накопичених семплів, з яких відбудеться розрахунок в дБ

if (samplesCount >= WINDOW_SIZE) {

    double mean = sumSq / samplesCount;
    double rms = sqrt(mean); //розрахунок RMS

    float LAeq = //розрахунок значення в дБ
        20.0 * log10(rms/MIC_REF_AMPL) + CAL_DB;

    currentDB = LAeq;
}
if (alarmEnabled && currentDB > noiseThreshold) { //перевірка виконання умови для
спрацювання світлодіода
    digitalWrite(LED_PIN, HIGH);
} else {
    digitalWrite(LED_PIN, LOW);
}

dbHistory[historyIndex] = currentDB;
historyIndex = (historyIndex + 1) % HISTORY_SIZE;

sumSq = 0;
samplesCount = 0;

display.clear(); //очистити дисплей

if (menuState == MAIN_SCREEN) { //залежно від вибраного екрану

    display.drawString(0, 0, "KPI2026"); //вказування даних про звукове навантаження
    display.drawString(0, 24, String(currentDB, 1) + " dB");

}

else if (menuState == MENU_AWEIGHT) { //залежно від вибраного екрану

    display.drawString(0, 0, "A-Weight"); //вказування даних про вибір режиму фільтрації
    display.drawString(0, 24, aWeightEnabled ? "ON" : "OFF");
    display.drawString(0, 48, "Hold: toggle");

}

else if (menuState == MENU_ALARM) { //залежно від вибраного екрану

    display.drawString(0, 0, "Alarm"); //вказування даних про статус попередження про
перевищення рівня шуму
    display.drawString(0, 24, alarmEnabled ? "ON" : "OFF");
    display.drawString(0, 48, "Hold: toggle");

}

else if (menuState == MENU_THRESHOLD) { //залежно від вибраного екрану

    display.drawString(0, 0, "Threshold"); //вказування даних про обраний максимальний
дозволений рівень шуму
    display.drawString(0, 24, String(noiseThreshold, 1));
    display.drawString(0, 48, "Hold: +1 dB");

}

display.display(); //виведення даних на дисплей
}
}

```