

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Мікропроцесорна система моніторингу рівня шуму на базі ESP8266  
Назва теми

КВРКІ 220033.22.01.18 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент IV курсу, група КІ2с-22-1

  
Підпис

Андрій ПРИСЯЖНИЙ  
Ініціали, прізвище

Керівник

Підпис, дата

Володимир ГРИГА  
Ініціали, прізвище

Нормоконтролер

Підпис, дата

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

« 19 » червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

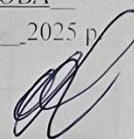
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

10 " 01 2025 р.



ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Присяжному АНДРІО

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мікропроцесорна система моніторингу рівня шуму на базі ESP8266

Керівник проекту (роботи) Володимир Грига., к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. №23

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1.Огляд та аналіз подібних систем

2. Вибір та обґрунтування апаратних та програмних засобів розробки

3. Розроблення апаратно-програмної системи

4. Тестування роботи системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структура програмно-технічного пристрою

Схема електрична принципова

Блок-схема

алгоритму

роботи

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій Нічепорук, доцент кафедри КПС		

7. Дата видачі завдання « 11 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1: Аналіз відомих засобів вимірювання шуму на базі мікроконтролерних модулів	01.03.2025	виконано
4	Робота над розділом 2: Просктування програмно-технічного засобу вимірювання шуму	01.04.2025	виконано
5	Робота над розділом 3: Програмна реалізація засобу вимірювання шуму	30.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2025	виконано
7	Попередній захист ВКР	30.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Андрій ПРИСЯЖНИЙ  
Ініціали, прізвище

Керівник проекту (роботи)

Підпис

Володимир ГРИГА  
Ініціали, прізвище

№ р я д к а	ф о р м а т	Позначення	Найменування	К і л л и с т і в	№ ек з	П р и м і т к а
			Текстові документи			
1		КвРКІ. 220033.22.01.18ПЗ	Пояснювальна записка	75		
			Графічні матеріали			
2		КвРКІ. 220033.22.01.18Е8	Структурна схема мікропроцесорної системи	1		
3		КвРКІ. 220033.22.01.18Е8	Електрична принципова схема мікропроцесорної системи	1		
4		КвРКІ. 220033.22.01.18Е8	Функціональна схема мікропроцесорної системи	1		
4		КвРКІ. 220033.22.01.18Е8	Програмний код мікропроцесорної системи	30		
		КвРКІ. 220033.22.01.18ПЗ				
Зм	Арк	№ докум	Підпис	Дата	Відомість проекту  ХНУ, КІ2с-22-1	
Розробив	Присяжний А.Р.					
Перевір.	Грига В.М.					
Н. контр.	Кисіль Т.М.		19.06.18			
Затв.	Павлова О.О.		19.06.18		Літера	Аркуш
					У	1
						1

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мікропроцесорна система моніторингу рівня шуму на базі ESP8266».

Автор роботи: Присяжний Андрій Русланович

Керівник роботи: Грига Володимир Михайлович

Пояснювальна записка: 75 с., 2 рис, 23 табл, 4 дод., 65 джерел.

Графічна частина: 3 креслення.

Метою роботи є розробка мікропроцесорної системи моніторингу рівня шуму на базі ESP8266 з функцією передачі даних через Wi-Fi для забезпечення безперервного контролю акустичного забруднення навколишнього середовища.

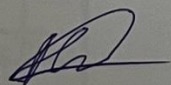
Для досягнення поставленої мети сформульовано такі завдання дослідження:

1. Проаналізувати методи та засоби вимірювання рівня шуму, визначити їхні переваги та недоліки;
2. Дослідити бездротові технології передачі даних для визначення оптимального рішення в контексті створення IoT-системи моніторингу шуму;
3. Розробити апаратну частину мікропроцесорної системи на базі ESP8266;
4. Розробити алгоритми та програмне забезпечення для мікроконтролера ESP8266, що реалізують вимірювання рівня шуму, обробку сигналу;
5. Розробити серверну частину системи та веб-інтерфейс для збору, зберігання, аналізу та візуалізації даних про рівень шуму;
6. Провести експериментальні дослідження розробленої системи в реальних умовах для оцінки її точності, надійності, енергоефективності.

Об'єкт дослідження — процес вимірювання та моніторингу рівня шуму в навколишньому середовищі.

Предмет дослідження — методи та засоби створення мікропроцесорної системи моніторингу рівня шуму на базі ESP8266 з функцією передачі даних через Wi-Fi.




Підпис студента



Дата

## ЗМІСТ

<b>ВСТУП</b> .....	3
<b>1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ МОНІТОРИНГУ РІВНЯ ШУМУ</b> .....	4
1.1 Актуальність тематики .....	4
1.2 Класифікація систем моніторингу шуму .....	5
1.3 Огляд та аналіз методів вимірювання рівня шуму .....	8
1.4.1 Мікрофонні датчики .....	12
1.4.2 Аналогово-цифрові перетворювачі .....	14
1.4.3 Системи фільтрації сигналу .....	15
1.5 Огляд аналогічних систем моніторингу шуму.....	15
1.6 Постановка завдання .....	16
1.7 Висновки .....	17
<b>2 ПРОЄКТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ</b> .....	19
2.1 Розроблення структури мікропроцесорної системи .....	19
2.2 Функціональне призначення основних модулів системи .....	21
2.3 Вибір апаратних модулів мікропроцесорної системи .....	25
2.3.1 Вибір мікроконтролера ESP8266 .....	25
2.3.2 Вибір модуля мікрофонного датчика .....	26
2.3.3 Вибір модуля живлення.....	28
2.3.4 Вибір дисплею для відображення даних .....	29
2.3.5 Вибір додаткових модулів системи.....	31
2.4 Вибір програмного забезпечення для розробки.....	32
2.4.1 Вибір середовища програмування.....	32
2.4.2 Вибір бібліотек для роботи з ESP8266.....	36
2.4.3 Вибір хмарної платформи для збору даних.....	39
2.5 Висновки .....	42

				КвРКІ.220033.22.01.18 ПЗ			
Зм.	Арк.	№докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав		Андрій ПРИСЯЖНИЙ			у	2	75
Перевір.		Володимир ГРИГА					
Н.контр.		Тетяна КИСІЛЬ		2022			
Затвер.		Ольга ПАВЛОВА		2022			
Мікропроцесорна система моніторингу рівня шуму на базі ESP8266					ХНУ КІ2с-22-1		

<b>3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ</b>	<b>43</b>
3.1 Розроблення функціональної схеми мікропроцесорної системи .....	43
3.2 Розроблення електричної принципової схеми .....	47
3.3 Розроблення алгоритму функціонування системи .....	52
3.4.2 Алгоритм передачі даних через Wi-Fi.....	55
3.4.3 Розроблення веб-інтерфейсу для візуалізації даних.....	57
3.5 Калібрування та тестування системи .....	58
3.5.1 Методика калібрування мікрофонного датчика.....	58
3.5.2 Тестування точності вимірювань.....	59
3.5.3 Тестування енергоефективності системи .....	60
3.5.4 Тестування надійності передачі даних.....	61
3.6 Аналіз результатів експериментальних досліджень.....	62
3.7 Висновки .....	64
<b>ВИСНОВКИ</b> .....	<b>66</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	<b>68</b>
<b>ДОДАТКИ</b> .....	<b>75</b>

## ВСТУП

Стрімкий розвиток урбанізаційних процесів та технологічний прогрес призвели до суттєвого збільшення рівня акустичного забруднення навколишнього середовища. Шумове забруднення визнано Всесвітньою організацією охорони здоров'я одним із найбільш поширених факторів негативного впливу на якість життя та здоров'я населення в урбанізованих територіях. За даними ВООЗ, близько 40% населення Європи піддається впливу транспортного шуму з рівнем понад 55 дБ протягом денного часу, що перевищує рекомендовані норми та становить значну загрозу для здоров'я. Тривала експозиція підвищеного рівня шуму призводить до порушень функціонування серцево-судинної та нервової систем, погіршення когнітивних функцій, розладів сну, підвищення стресу та зниження працездатності.

Особливої гостроти проблема набуває в контексті зростання інтенсивності транспортних потоків, розширення промислових зон у безпосередній близькості до житлових районів та загального ущільнення міської забудови. Дослідження акустичного середовища великих міст показують систематичне перевищення санітарних норм шуму на більшості міських територій, що потребує впровадження ефективних систем моніторингу та контролю шумового забруднення.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						1
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ МОНІТОРИНГУ РІВНЯ ШУМУ

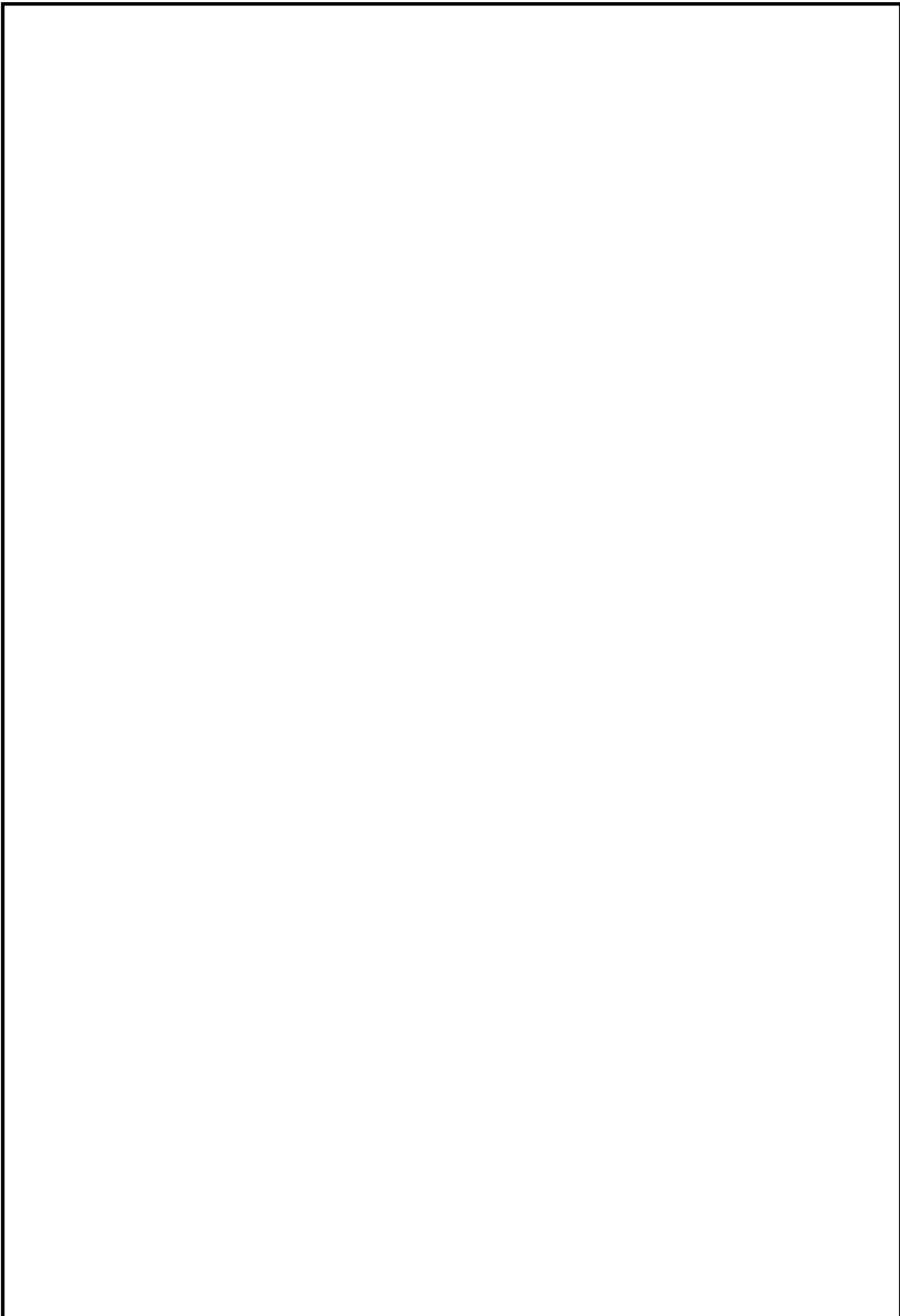
## 1.1 Актуальність тематики

Міська урбанізація та розвиток промисловості призвели до значного збільшення акустичного забруднення навколишнього середовища. Шумове забруднення стало однією з найбільш поширених форм екологічних проблем у густонаселених територіях, що негативно впливає на здоров'я населення, якість життя та психоемоційний стан людей. За даними Всесвітньої організації охорони здоров'я, близько 40% населення Європи піддається впливу дорожнього шуму з еквівалентним рівнем звукового тиску, що перевищує 55 дБ [1-5].

Дослідження останніх років демонструють, що тривалий вплив підвищеного рівня шуму може призводити до серйозних порушень у роботі серцево-судинної системи, викликати безсоння, підвищувати рівень стресу та дратівливості. Особливо негативно шумове забруднення впливає на розумову працездатність, концентрацію уваги та когнітивні функції людини [6]. Акустичне забруднення в умовах міста має комплексний характер та формується від багатьох джерел: транспортних потоків, промислових об'єктів, будівельних майданчиків, систем вентиляції будівель, масових заходів та повсякденної діяльності людей [21].

Зі збільшенням масштабів містобудування та зростанням кількості транспортних засобів проблема шумового забруднення стає все більш актуальною. Шумові карти міст демонструють перевищення допустимих норм у більшості районів щільної забудови. Зокрема, дослідження акустичного середовища великих міст показують, що рівень шуму на головних транспортних магістралях часто перевищує 70-80 дБ, що значно вище встановлених санітарних норм [20].

					КВРКІ.220033.22.01.18ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		



					КВРКІ.220033.22.01.18ПЗ	Арк.
						3
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.1 – Допустимі рівні шуму для різних міських територій

Тип території	Денний час (7:00-23:00), дБА	Нічний час (23:00-7:00), дБА
Сельбищні зони міст і селищ	55	45
Зони, прилеглі до лікарень та санаторіїв	50	40
Території шкіл, дитячих закладів	55	45
Території з підвищеними вимогами до тиші	45	35
Промислові зони	80	70

Відсутність систематичного моніторингу шумового забруднення ускладнює своєчасне виявлення та усунення потенційних загроз для населення. Традиційні методи вимірювання шуму передбачають використання спеціалізованого обладнання, що потребує значних фінансових витрат та кваліфікованого персоналу для проведення замірів. Крім того, такі вимірювання здебільшого проводяться точково та періодично, що не дозволяє отримати повну картину акустичного забруднення в динаміці [8].

Розробка мікропроцесорних систем моніторингу рівня шуму на базі ESP8266 вимагає комплексного підходу, що враховує всі аспекти функціонування таких систем: від апаратного забезпечення до методів аналізу даних та способів візуалізації результатів [23]. Отже, розробка та впровадження мікропроцесорних систем моніторингу рівня шуму на базі ESP8266 є актуальним науково-технічним завданням, вирішення якого сприятиме покращенню якості життя в міському середовищі, зниженню негативного впливу шуму на здоров'я населення, ефективному плануванню містобудівної діяльності та формуванню культури акустичного комфорту.

## 1.2 Класифікація систем моніторингу шуму

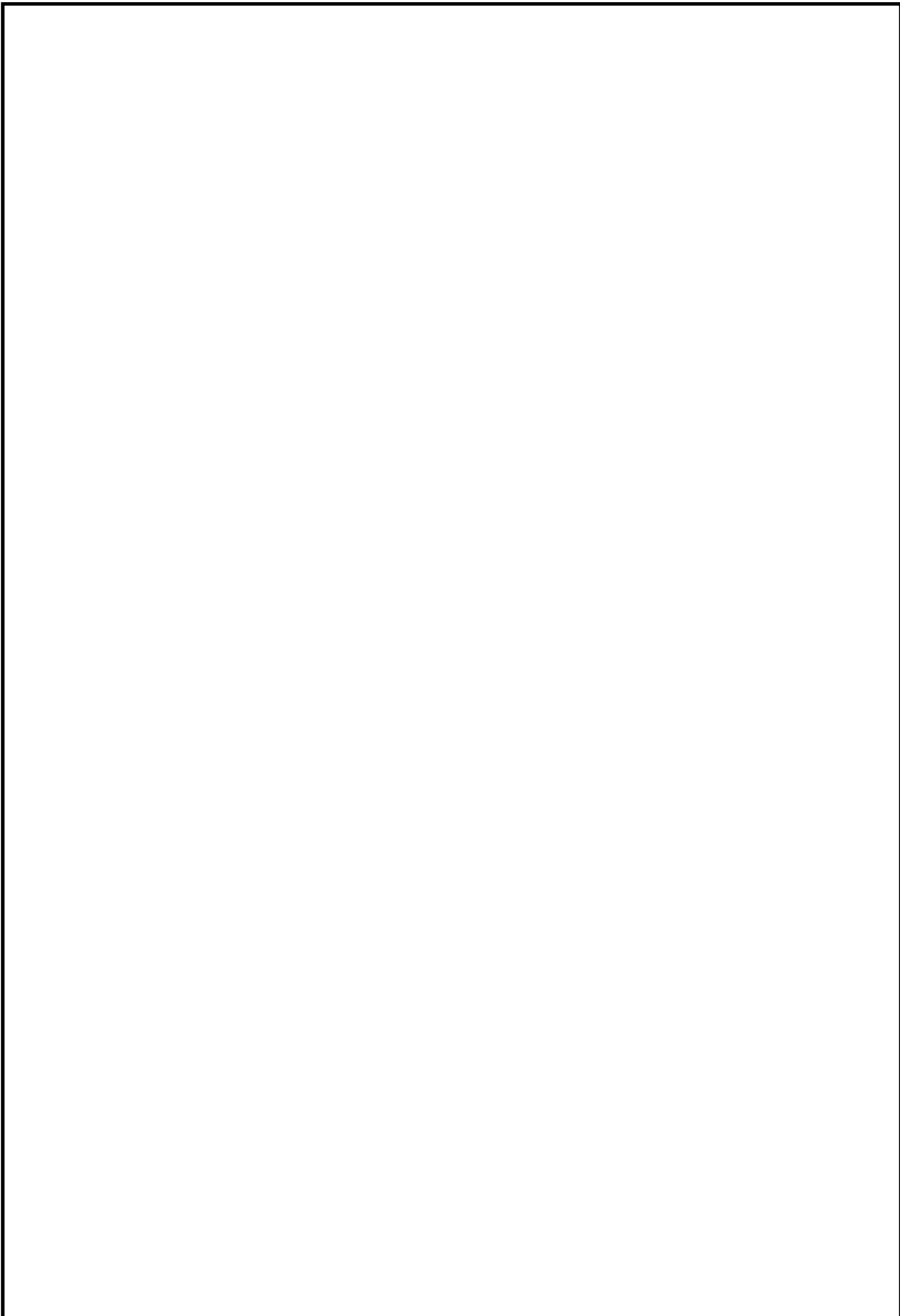
					КВРКІ.220033.22.01.18ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Розвиток технологій моніторингу акустичного забруднення навколишнього середовища пройшов тривалий еволюційний шлях від простих механічних пристроїв до складних кібер-фізичних систем. Класифікацію систем моніторингу шуму доцільно проводити за кількома ключовими параметрами: мобільністю, функціональністю, способом передачі даних, типом живлення, класом точності та сферою застосування [1].

За критерієм мобільності системи моніторингу шуму поділяються на стаціонарні, мобільні та змішані. Стаціонарні системи встановлюються на фіксованих позиціях і забезпечують безперервний моніторинг акустичного середовища в конкретних точках простору. Такі системи зазвичай характеризуються високою точністю вимірювань, надійністю роботи та можливістю підключення до стаціонарних мереж електроживлення. Мобільні системи моніторингу шуму встановлюються на рухомі платформи (транспортні засоби, безпілотні літальні апарати, мобільні роботи) або виконуються у вигляді портативних пристроїв, які можна переносити та швидко розгортати в різних локаціях. Змішані системи поєднують елементи стаціонарних та мобільних рішень, наприклад, тимчасові вимірювальні станції, що встановлюються на певний період для детального дослідження акустичного середовища конкретної території [12].

За функціональністю виділяють однопараметричні, багатопараметричні та комплексні системи моніторингу шуму. Однопараметричні системи вимірюють лише загальний рівень шуму, зазвичай у децибелах, і можуть здійснювати найпростішу статистичну обробку даних. Багатопараметричні системи, окрім загального рівня шуму, визначають його спектральні характеристики, часові параметри (еквівалентний рівень, максимальні та мінімальні значення), психоакустичні показники (гучність, різкість, шорсткість). Комплексні системи додатково виконують аналіз джерел шуму, класифікацію акустичних подій, оцінку впливу шуму на людину та навколишнє середовище, прогнозування змін акустичного фону [20].

					КВРКІ.220033.22.01.18ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		



					КВРКІ.220033.22.01.18ПЗ	Арк.
						6
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.2 – Порівняння функціональних можливостей різних типів систем моніторингу шуму

<b>Функціональна можливість</b>	<b>Однопараметричні</b>	<b>Багатопараметричні</b>	<b>Комплексні</b>
Вимірювання загального рівня шуму	+	+	+
Спектральний аналіз	-	+	+
Визначення часових характеристик	-	+	+
Психоакустичні параметри	-	+/-	+
Класифікація джерел шуму	-	-	+
Прогнозування акустичного фону	-	-	+
Оцінка впливу на людину	-	-	+
Інтеграція з іншими системами	-	+/-	+

Системи моніторингу шуму також можна класифікувати за способом передачі даних на дротові, бездротові та гібридні. Дротові системи використовують фізичні з'єднання для передачі даних від сенсорів до центрального вузла обробки. Такі системи забезпечують високу швидкість передачі даних, стабільність зв'язку, але характеризуються складністю розгортання, особливо на великих територіях. Бездротові системи використовують радіоканали для передачі даних, що спрощує їх розгортання та масштабування. В залежності від конкретної технології (Wi-Fi, Bluetooth, LoRaWAN, GSM, ZigBee), бездротові системи можуть мати різні характеристики дальності зв'язку, енергоспоживання, пропускної здатності. Гібридні системи поєднують дротові та бездротові технології передачі даних, оптимізуючи структуру мережі відповідно до конкретних умов експлуатації [5].

Таблиця 1.3 – Характеристики систем моніторингу шуму різних класів точності

Характеристика	Клас 1 (прецизійні)	Клас 2 (загального призначення)	Клас 3 (індикаторні)
Похибка вимірювань, дБ	±1,0	±2,0	±3,0-5,0
Діапазон вимірювань, дБ	20-140	30-130	40-120
Частотний діапазон, Гц	10-20000	20-10000	31.5-8000
Температурний діапазон, °С	-10...+50	0...+40	+5...+35
Вартість	Висока	Середня	Низька
Калібрування	Часте та точне	Періодичне	Рідко або відсутнє
Застосування	Наукові дослідження, юридичні вимірювання	Екологічний моніторинг, оцінка умов праці	Орієнтовна оцінка рівня шуму

Залежно від періодичності вимірювань розрізняють системи безперервного, періодичного та подієвого моніторингу шуму. Системи безперервного моніторингу здійснюють вимірювання постійно, забезпечуючи повну картину зміни акустичного фону з часом. Системи періодичного моніторингу виконують вимірювання через заданий інтервал часу, що дозволяє економити енергію та ресурси зберігання даних. Системи подієвого моніторингу активуються при виникненні певних акустичних подій (перевищення порогового рівня шуму, виявлення специфічних звуків) або зовнішніх сигналів [12].

### 1.3 Огляд та аналіз методів вимірювання рівня шуму

Методи вимірювання рівня шуму умовно поділяються на суб'єктивні та об'єктивні. Суб'єктивні методи базуються на оцінці шуму людиною і

використовуються переважно в психоакустичних дослідженнях, соціологічних опитуваннях, при розробці шкал суб'єктивної оцінки акустичного комфорту. Об'єктивні методи передбачають використання вимірювальних приладів для кількісної оцінки параметрів шуму: рівня звукового тиску, частотних характеристик, часових параметрів [16].

Традиційний метод вимірювання рівня шуму базується на використанні шумомірів – спеціалізованих приладів, що визначають рівень звукового тиску в децибелах. Залежно від налаштувань, шумоміри можуть вимірювати загальний рівень шуму або його складові в різних частотних діапазонах (октавних або третьоктавних смугах). Більшість шумомірів також підтримують різні часові характеристики вимірювання: швидку (F) з постійною часу 125 мс, повільну (S) з постійною часу 1 с, імпульсну (I) для оцінки короткочасних шумів. Сучасні шумоміри дозволяють визначати еквівалентний рівень шуму за певний період, максимальні та мінімальні значення, процентильні рівні та інші статистичні параметри [20].

Метод спектрального аналізу шуму передбачає визначення його частотного складу за допомогою аналізаторів спектру. Цей метод дозволяє виявляти домінуючі частотні складові шуму, що важливо для ідентифікації джерел шуму та розробки заходів з шумозахисту. Спектральний аналіз може здійснюватися в реальному часі або шляхом запису звукового сигналу з подальшою обробкою. Сучасні методи спектрального аналізу базуються на алгоритмах швидкого перетворення Фур'є (FFT), вейвлет-перетворення, цифрової фільтрації [9].

Метод інтенсиметрії базується на вимірюванні інтенсивності звуку – векторної величини, що характеризує потік звукової енергії через одиницю площі. На відміну від вимірювання рівня звукового тиску, метод інтенсиметрії дозволяє визначати напрямок поширення звуку, що полегшує локалізацію джерел шуму. Для вимірювання інтенсивності звуку використовуються спеціальні зонди, що складаються з пари мікрофонів, розміщених на фіксованій відстані один від

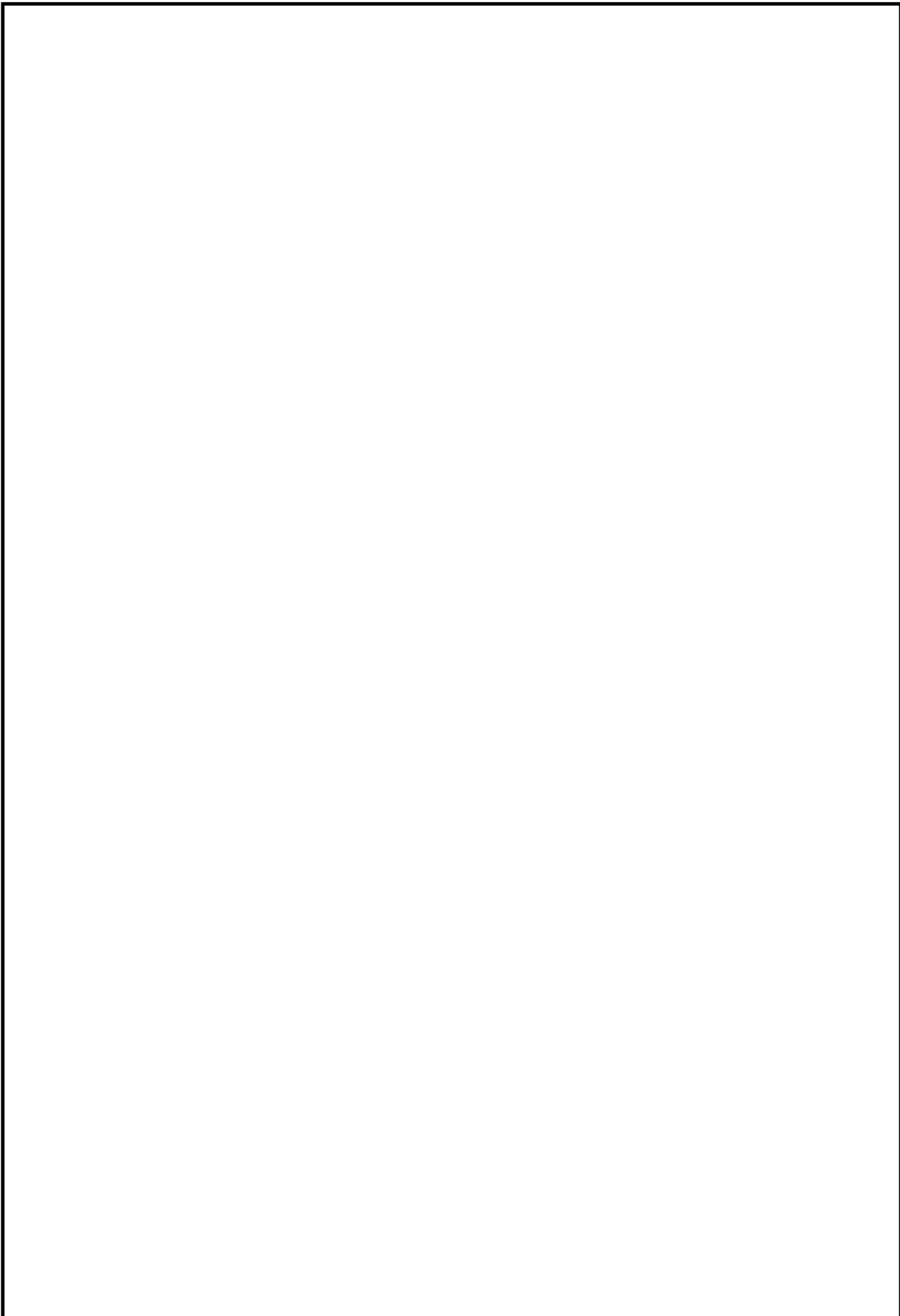
					КВРКІ.220033.22.01.18ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

одного. Інтенсивність звуку визначається шляхом обробки сигналів з обох мікрофонів з урахуванням фазових співвідношень [22].

Таблиця 1.4 – Порівняння методів вимірювання рівня шуму

Метод	Принцип роботи	Переваги	Недоліки	Сфера застосування
Традиційний (шумомір)	Вимірювання рівня звукового тиску	Простота, стандартизованість	Обмежена інформативність	Загальна оцінка шуму
Спектральний аналіз	Розкладання шуму на частотні складові	Детальна інформація про частотний склад	Складність обробки, висока вартість	Ідентифікація джерел шуму
Інтенсиметрія	Вимірювання потоку звукової енергії	Можливість визначення напрямку	Висока чутливість до умов вимірювання	Локалізація джерел шуму
Бінауральний запис	Імітація сприйняття звуку людиною	Враховує особливості сприйняття звуку людиною	Складність обробки, суб'єктивність	Психоакустичні дослідження
Акустична голографія	Візуалізація звукового поля	Детальна картина розподілу звуку	Висока складність, вартість	Наукові дослідження

Метод акустичної голографії дозволяє візуалізувати звукове поле та створювати карти розподілу звукового тиску або інтенсивності звуку в просторі. Акустична голографія базується на вимірюванні звукового поля в багатьох точках за допомогою масиву мікрофонів з подальшою математичною реконструкцією звукового поля у всьому досліджуваному об'ємі. Цей метод забезпечує найбільш повну інформацію про звукове поле, але потребує складного обладнання та значних обчислювальних ресурсів [4].



					КВРКІ.220033.22.01.18ПЗ	Арк.
						11
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.5 – Основні типи мікрофонів для систем моніторингу шуму на базі ESP8266

<b>Тип мікрофона</b>	<b>Принцип роботи</b>	<b>Вихідний сигнал</b>	<b>Переваги</b>	<b>Недоліки</b>
Електретний	Зміна ємності при коливаннях діафрагми	Аналоговий	Низька вартість, компактність	Середня точність, чутливість до електромагнітних завад
MEMS	Мікромеханічна система на кремнієвій підкладці	Аналоговий або цифровий	Висока стабільність, мініатюрність	Обмежений динамічний діапазон
I2S MEMS	MEMS з вбудованим АЦП та цифровим інтерфейсом	Цифровий (I2S)	Висока завадостійкість, точність	Вища вартість, складність інтерфейсу
PDM MEMS	MEMS з імпульсно-ковою модуляцією	Цифровий (PDM)	Висока роздільна здатність	Складність обробки, висока вартість
MEMS з інтегрованим сигнальним процесором	MEMS з вбудованою обробкою сигналу	Цифровий, попередньо оброблений	Висока функціональність	Висока вартість, складність програмування

Методи візуалізації даних про рівень шуму включають створення акустичних карт, часових графіків, спектрограм, статистичних діаграм. Акустичні карти відображають просторовий розподіл рівня шуму на певній території та створюються шляхом інтерполяції даних від окремих вимірювальних точок. Часові графіки демонструють зміну рівня шуму протягом доби, тижня, місяця, дозволяючи виявляти закономірності та аномалії. Спектрограми відображають частотний склад шуму та його зміну з часом, що допомагає у ідентифікації джерел шуму. Статистичні діаграми (гістограми, процентильні графіки) дозволяють

оцінити розподіл рівнів шуму, виявити найбільш поширені значення, визначити відсоток часу, протягом якого рівень шуму перевищує певні порогові значення [7].

#### 1.4 Огляд та аналіз апаратних засобів вимірювання рівня шуму

Сучасні апаратні засоби вимірювання рівня шуму становлять комплексну систему взаємопов'язаних компонентів, що забезпечують перетворення акустичних коливань у цифровий сигнал з подальшою обробкою та аналізом. Еволюція цих технічних рішень відбувалася паралельно з розвитком мікроелектроніки, цифрової обробки сигналів та інформаційних технологій. Розвиток мініатюрних мікрофонних сенсорів, високоточних аналогово-цифрових перетворювачів та мікроконтролерів з низьким енергоспоживанням створив передумови для розробки компактних, автономних та економічно доступних систем моніторингу шуму [13].

##### 1.4.1 Мікрофонні датчики

Мікрофонні датчики є первинними перетворювачами в системах вимірювання рівня шуму, що трансформують акустичні коливання повітря в електричний сигнал. Характеристики мікрофона безпосередньо впливають на точність, чутливість та частотний діапазон всієї системи. Для систем моніторингу шуму використовуються різні типи мікрофонів: конденсаторні, електретні, динамічні, п'єзоелектричні, MEMS (мікроелектромеханічні системи) [17].

Конденсаторні мікрофони забезпечують найвищу точність та стабільність вимірювань і тому широко застосовуються в прецизійних шумомірах класу 1. Принцип їх роботи базується на зміні ємності конденсатора при коливаннях діафрагми під дією звукових хвиль. Конденсаторні мікрофони характеризуються рівномірною частотною характеристикою в широкому діапазоні частот, високою чутливістю, низьким рівнем власних шумів. Основними недоліками є висока

					КВРКІ.220033.22.01.18ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

вартість, необхідність поляризуючої напруги, чутливість до вологості повітря [20].

Електретні мікрофони є різновидом конденсаторних мікрофонів, в яких одна з обкладок конденсатора виготовлена з електрету – матеріалу, що має постійну електричну поляризацію. Завдяки цьому електретні мікрофони не потребують окремого джерела поляризуючої напруги, що спрощує їх конструкцію та знижує вартість. Сучасні електретні мікрофони забезпечують достатньо високу якість звукопередачі і широко використовуються в системах моніторингу шуму класу 2, а найкращі зразки – навіть у системах класу 1. Електретні мікрофони потребують попереднього підсилювача, який зазвичай інтегрований у корпус мікрофона [16].

Таблиця 1.6 – Порівняльна характеристика основних типів мікрофонів для систем моніторингу шуму

Характеристика	Конденсаторний	Електретний	MEMS	Динамічний	П'єзое.
Чутливість, мВ/Па	10-50	5-20	1-10	1-5	
Частотний діапазон, Гц	5-20000	20-18000	20-10000	50-15000	10
Динамічний діапазон, дБ	120	100	80	90	
Власний шум, дБА	<15	<25	<30	<35	
Стабільність	Висока	Середня	Висока	Висока	Н
Чутливість до умов середовища	Висока	Середня	Низька	Низька	В
Необхідність живлення	Так	Так	Так	Ні	
Розміри	Великі	Середні	Малі	Великі	С
Вартість	Висока	Низька	Середня	Середня	Н
Клас точності	1	1-2	2-3	3	

Характеристики мікрофонів, що суттєво впливають на якість вимірювання рівня шуму, включають: чутливість, що визначає рівень вихідного сигналу при заданому рівні звукового тиску; частотну характеристику, що показує залежність

чутливості від частоти звуку; направленість, що характеризує залежність чутливості від напрямку приходу звукової хвилі; динамічний діапазон, що визначає межі вимірювання від рівня власних шумів до рівня, при якому виникають спотворення; рівень власних шумів; імпедансні характеристики; стабільність параметрів при зміні температури, вологості, атмосферного тиску [13].

#### 1.4.2 Аналогово-цифрові перетворювачі

Аналогово-цифрові перетворювачі (АЦП) забезпечують перетворення аналогового сигналу з мікрофона в цифрову форму для подальшої обробки мікроконтролером. Характеристики АЦП суттєво впливають на точність, динамічний діапазон та частотну роздільну здатність системи вимірювання рівня шуму. Основними параметрами АЦП є: розрядність (кількість біт), що визначає точність квантування сигналу; частота дискретизації, що визначає часову роздільну здатність; динамічний діапазон, що визначає співвідношення між максимальним рівнем сигналу та рівнем шумів квантування; нелінійні спотворення; час перетворення [13].

Мікроконтролер ESP8266 має вбудований 10-бітний АЦП з одним вхідним каналом, що обмежує його можливості для прецизійних вимірювань рівня шуму. Динамічний діапазон 10-бітного АЦП становить близько 60 дБ, що недостатньо для охоплення всього діапазону рівнів шуму, які зустрічаються в реальних умовах (від порогу чутності близько 0 дБ до рівня болювого порогу близько 120 дБ). Крім того, вбудований АЦП мікроконтролера ESP8266 характеризується відносно високим рівнем шумів та нелінійних спотворень [12].

Важливим аспектом роботи АЦП в системах вимірювання рівня шуму є правильне узгодження динамічного діапазону аналогового сигналу з мікрофона з діапазоном вхідної напруги АЦП. Для цього використовуються схеми попереднього підсилення з регульованим коефіцієнтом підсилення, що

					КВРКІ.220033.22.01.18ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяють адаптувати систему до різних рівнів шуму. У випадку використання мікрофонів з вбудованим підсилювачем (наприклад, МАХ9814) коефіцієнт підсилення може регулюватися зовнішніми компонентами або через програмований інтерфейс [12].

### 1.4.3 Системи фільтрації сигналу

Системи фільтрації сигналу в апаратних засобах вимірювання рівня шуму виконують кілька важливих функцій: придушення завад та шумів, що не є частиною вимірюваного акустичного сигналу; виділення частотних діапазонів, що відповідають певним стандартам вимірювання шуму; реалізація частотних корекцій, що імітують особливості сприйняття звуку людським вухом; антиаліасингова фільтрація перед аналогово-цифровим перетворенням. Системи фільтрації можуть бути реалізовані як у аналоговому домені (перед АЦП), так і в цифровому домені (після АЦП) [9].

Експериментальні дослідження показують, що системи моніторингу шуму на базі ESP8266 з правильно спроектованими схемами аналогової та цифрової фільтрації можуть забезпечувати точність вимірювання рівня шуму, достатню для систем класу 2, а в деяких випадках – наближену до систем класу

### 1.5 Огляд аналогічних систем моніторингу шуму

Розробка та впровадження систем моніторингу шуму активно ведеться як науково-дослідними установами, так і комерційними компаніями. Аналіз існуючих рішень дозволяє виявити їх сильні та слабкі сторони, визначити актуальні тенденції та перспективні напрямки розвитку технологій моніторингу шуму. Існуючі системи моніторингу шуму можна класифікувати за кількома критеріями: комерційні та некомерційні, стаціонарні та мобільні, професійні та побутові, автономні та з централізованим живленням [5].

					КВРКІ.220033.22.01.18ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Комерційні системи моніторингу шуму, такі як Brüel & Kjær Noise Sentinel, Cirrus Environmental NoiseHub, Norsonic NorCloud, характеризуються високою точністю вимірювань, надійністю, широкою функціональністю, але мають високу вартість, що обмежує їх масове впровадження. Ці системи зазвичай відповідають вимогам до шумомірів класу 1 або 2 згідно з міжнародними стандартами, забезпечують безперервний моніторинг шуму з можливістю аудіозапису при перевищенні порогових значень, мають розвинені засоби візуалізації та аналізу даних. Вартість таких систем становить від кількох тисяч до десятків тисяч доларів за одну вимірювальну станцію, що робить їх доступними переважно для великих промислових підприємств, аеропортів, муніципальних служб [20].

Системи моніторингу шуму на базі ESP8266 та аналогічних мікроконтролерів представляють особливий інтерес як баланс між вартістю, функціональністю та точністю. Проекти, такі як NodeMCU Noise Monitor, ESP8266 Sound Level Meter, IoT Noise Pollution Monitoring System, демонструють можливість створення компактних, енергоефективних та економічно доступних систем моніторингу шуму з передачею даних через Wi-Fi. Такі системи зазвичай забезпечують точність, достатню для індикаторних вимірювань або систем класу 3, а при належному калібруванні та використанні якісних компонентів можуть наближатися до характеристик систем класу 2 [7].

## 1.6 Постановка завдання

На основі проведеного аналізу методів та засобів моніторингу рівня шуму, а також бездротових технологій передачі даних, сформульовано основні завдання дослідження:

1. Розробити апаратну частину мікропроцесорної системи моніторингу рівня шуму на базі ESP8266, що забезпечує вимірювання рівня шуму з точністю, достатньою для систем класу 2, та передачу даних через Wi-Fi.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Розробити програмне забезпечення мікроконтролера ESP8266 для вимірювання рівня шуму, обробки сигналу, включаючи частотну корекцію та спектральний аналіз, та передачі даних на сервер.

3. Розробити серверну частину системи для збору, зберігання, аналізу та візуалізації даних про рівень шуму.

4. Розробити веб-інтерфейс для моніторингу рівня шуму в режимі реального часу, відображення історичних даних, створення акустичних карт та генерації звітів.

5. Провести експериментальні дослідження розробленої системи в реальних умовах для оцінки її точності, надійності, енергоефективності та визначення можливих шляхів удосконалення.

Таблиця 1.14 – Основні вимоги до розроблюваної системи моніторингу шуму

Параметр	Значення
Діапазон вимірювання рівня шуму	30-120 дБ
Точність вимірювання	±2 дБ (відповідно до класу 2)
Частотний діапазон	20-10000 Гц
Частотна корекція	A, C, Z
Інтервал вимірювань	від 1 секунди до 1 години (налаштовується)
Технологія передачі даних	Wi-Fi
Протокол передачі даних	MQTT
Живлення	USB (5В), опціонально - акумулятор
Час автономної роботи	не менше 24 годин
Габарити	не більше 100x50x30 мм
Вартість	не більше 50 доларів США

## 1.7 Висновки

1. Проведений аналіз проблеми шумового забруднення підтвердив необхідність розробки автоматизованих систем моніторингу шуму, оскільки традиційні методи не забезпечують безперервного контролю з високою

просторовою та часовою роздільною здатністю. Для систем на базі ESP8266 оптимальним є метод прямого вимірювання рівня звукового тиску з подальшою цифровою обробкою сигналу.

2. На основі порівняльного аналізу технологій бездротової передачі даних встановлено, що Wi-Fi є найбільш доцільним рішенням для систем моніторингу шуму на базі ESP8266 завдяки вбудованому Wi-Fi модулю, високій пропускній здатності та широкій доступності Wi-Fi інфраструктури в міському середовищі.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ОБГРУНТУВАННЯ ВИБОРУ АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ

### 2.1 Розроблення структури мікропроцесорної системи

Мікропроцесорна система моніторингу рівня шуму на базі ESP8266 представляє собою комплекс взаємопов'язаних апаратних та програмних компонентів, об'єднаних єдиною метою – забезпечення точного вимірювання, обробки та передачі даних про акустичне забруднення середовища. Структура системи базується на модульному підході, що надає можливість гнучкого налаштування та модифікації залежно від конкретних вимог експлуатації. Загальна архітектура складається з кількох функціональних блоків, пов'язаних між собою інформаційними потоками. Рис. 2.1 демонструє структурну схему мікропроцесорної системи моніторингу рівня шуму на базі ESP8266.

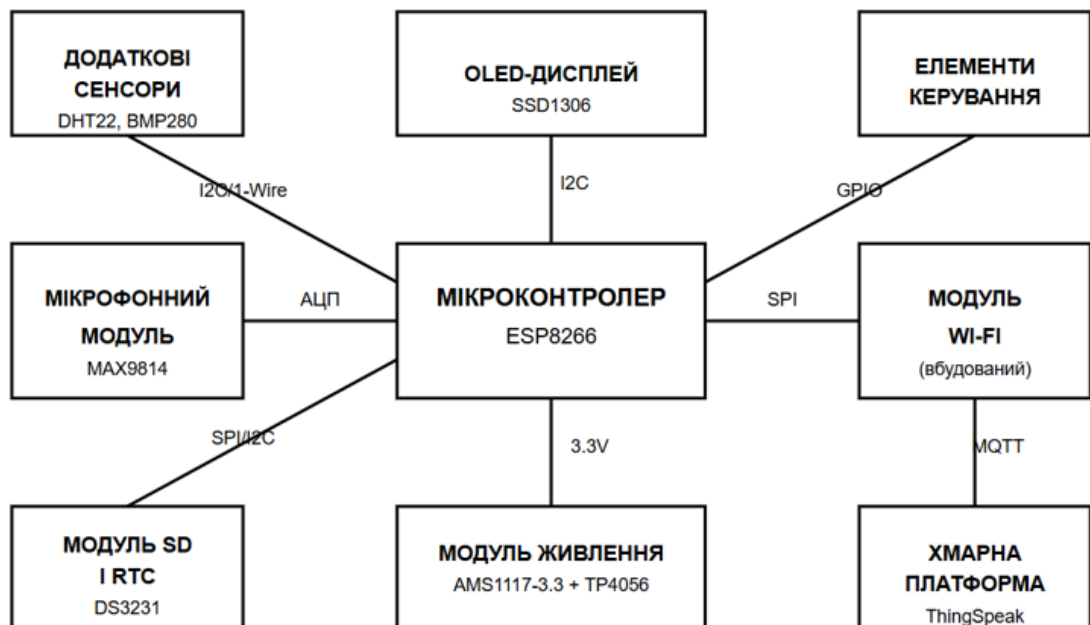


Рисунок 2.1 – Структурна схема мікропроцесорної системи моніторингу рівня шуму на базі ESP8266

Центральним елементом розробленої системи виступає мікроконтролер ESP8266, який завдяки власним технічним характеристикам забезпечує оптимальне співвідношення обчислювальної потужності, енергоспоживання та вартості. Інтеграція Wi-Fi модуля безпосередньо в архітектуру мікроконтролера суттєво спрощує реалізацію бездротового зв'язку та знижує загальну складність системи. Обраний мікроконтролер працює з тактовою частотою до 160 МГц та має достатній обсяг оперативної пам'яті для обробки поточних даних з датчика шуму.

Периферійний блок вимірювання шуму складається з мікрофонного модуля на базі електретного мікрофона з підсилювачем та аналого-цифрового перетворювача. Застосування додаткових схем фільтрації та нормалізації сигналу дозволяє отримати стабільні показники навіть при значних коливаннях рівня шуму в навколишньому середовищі. Калібрування модуля здійснюється програмно з урахуванням характеристик чутливості мікрофона та особливостей сприйняття звуку людським вухом.

Інтерфейсний блок забезпечує взаємодію з користувачем та включає елементи індикації (світлодіодні індикатори стану системи, дисплей для відображення поточних показників) та керування (кнопки налаштування режимів роботи, вибору параметрів відображення). Застосування компактного OLED-дисплею забезпечує енергоефективне відображення інформації з високою контрастністю навіть при складних умовах освітлення.

Взаємозв'язок між апаратними компонентами системи відображено у структурній схемі, представленій на рисунку 2.1. Модульна організація забезпечує можливість масштабування системи та її адаптації до різних умов експлуатації без суттєвих змін базової архітектури.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Основні компоненти мікропроцесорної системи моніторингу рівня шуму

Найменування компонента	Функціональне призначення	Технічні характеристики
Мікроконтролер ESP8266	Обробка даних, керування периферією, забезпечення комунікації	Частота: 80-160 МГц, пам'ять: 4 МБ, інтерфейси: SPI, I2C, UART, ADC, вбудований Wi-Fi
Мікрофонний модуль	Перетворення акустичних коливань в електричний сигнал	Чутливість: $-46 \pm 2$ дБ, діапазон частот: 50 Гц - 10 кГц, SNR: >60 дБ
OLED дисплей SSD1306	Відображення поточних даних та стану системи	Роздільна здатність: 128x64 пікселів, інтерфейс: I2C, напруга живлення: 3.3-5 В
Модуль живлення	Забезпечення стабільного електроживлення компонентів	Вхідна напруга: 7-12 В, вихідна напруга: 3.3 В, максимальний струм: 800 мА
Елементи керування	Взаємодія з користувачем, налаштування режимів	Тактові кнопки з фільтрацією брязкоту контактів, енкодер для навігації по меню

## 2.2 Функціональне призначення основних модулів системи

Функціональність мікропроцесорної системи моніторингу рівня шуму визначається спільною роботою окремих модулів, кожен з яких виконує специфічні завдання в рамках загальної архітектури. Розглянемо детально призначення та принципи роботи основних функціональних блоків.

Модуль збору акустичних даних виконує первинну функцію вимірювання рівня шуму в навколишньому середовищі. Електретний мікрофон з підсилювачем перетворює звукові хвилі в електричний сигнал, який далі подається на аналоговий вхід мікроконтролера. Особливістю реалізації даного модуля є застосування аналогових фільтрів для виділення корисного сигналу та зменшення впливу сторонніх шумів. Додатково встановлено схему автоматичного

					КВРКІ.220033.22.01.18ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

регулювання підсилення, що дозволяє адаптуватися до різних рівнів гучності без перевантаження вхідних каскадів аналого-цифрового перетворювача.

Аналого-цифровий перетворювач, вбудований в ESP8266, має обмежену розрядність (10 біт), що потенційно обмежує динамічний діапазон вимірювань. Для компенсації цього обмеження розроблено програмний алгоритм розширення динамічного діапазону шляхом адаптивного налаштування коефіцієнта підсилення вхідного сигналу. Дискретизація сигналу здійснюється з частотою до 10 кГц, що достатньо для аналізу більшості джерел побутового та промислового шуму.

Обчислювальний модуль на базі мікроконтролера ESP8266 реалізує алгоритми цифрової обробки сигналів для визначення рівня шуму в різних метриках. Програмна реалізація швидкого перетворення Фур'є дозволяє аналізувати спектральний склад шуму та виділяти характеристичні частотні компоненти, що важливо для ідентифікації джерел акустичного забруднення. Модуль забезпечує розрахунок еквівалентного рівня шуму з урахуванням особливостей сприйняття звуку людиною, застосовуючи різні типи вагових фільтрів (A, C, Z) відповідно до стандартів вимірювання акустичного тиску.

Накопичення та зберігання даних здійснюється як у внутрішній пам'яті мікроконтролера, так і з використанням зовнішніх носіїв інформації. Реалізовано циклічний буфер для зберігання миттєвих значень рівня шуму з можливістю подальшого аналізу часової динаміки акустичного фону. Статистична обробка даних дозволяє визначати мінімальні, максимальні та середні значення рівня шуму за обраний період спостереження, а також розраховувати процентильні показники, які характеризують стабільність акустичного фону.

Калібрування та валідація вимірювань реалізуються як програмними, так і апаратними засобами. Розроблено процедуру калібрування з використанням еталонного джерела звуку або професійного шумоміра для встановлення коефіцієнтів перетворення електричного сигналу в абсолютні значення рівня шуму. Періодична самодіагностика системи забезпечує контроль стабільності

					КВРКІ.220033.22.01.18ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

характеристик вимірювального тракту та своєчасне виявлення відхилень, що можуть вплинути на точність вимірювань.

Таблиця 2.2 – Функціональні режими роботи системи моніторингу рівня шуму

<b>Режим роботи</b>	<b>Опис функціональності</b>	<b>Особливості реалізації</b>
Постійний моніторинг	Безперервне вимірювання рівня шуму з фіксованою частотою дискретизації	Висока точність вимірювань, підвищене енергоспоживання, активна індикація на дисплеї
Періодичний моніторинг	Вимірювання рівня шуму з заданими інтервалами між сеансами	Оптимізоване енергоспоживання, використання режиму глибокого сну між вимірюваннями
Пороговий моніторинг	Активація вимірювань при перевищенні встановленого рівня шуму	Мінімальне енергоспоживання, використання компаратора для виявлення пікових значень
Калібрувальний режим	Налаштування параметрів вимірювального тракту з використанням еталонних джерел звуку	Автоматичне або ручне визначення коефіцієнтів підсилення та зміщення
Аналітичний режим	Розширений аналіз акустичного середовища з визначенням спектральних характеристик	Високе навантаження на обчислювальні ресурси, детальна візуалізація результатів

Програмна модель системи базується на багатозадачній архітектурі з пріоритетами, що дозволяє ефективно розподіляти обчислювальні ресурси мікроконтролера між різними функціональними завданнями. Фонові процеси, такі як збір та первинна обробка даних, виконуються з високим пріоритетом, забезпечуючи безперервність вимірювань навіть при активних комунікаційних сесіях або взаємодії з користувачем. Планувальник завдань керує виконанням окремих функціональних блоків з урахуванням їх пріоритетів та часових обмежень.

Алгоритм обробки акустичного сигналу включає декілька етапів: дискретизацію аналогового сигналу з мікрофона, цифрову фільтрацію для видалення шумів та артефактів, розрахунок середньоквадратичного значення амплітуди, перетворення електричних величин у децибели з урахуванням калібрувальних коефіцієнтів, а також застосування вагових фільтрів для моделювання сприйняття звуку людським вухом. Додатково реалізовано визначення еквівалентного рівня шуму за певний період спостереження та розрахунок дози шумового навантаження відповідно до санітарних норм.

Діагностика та службові функції системи забезпечують контроль працездатності окремих компонентів та своєчасне виявлення несправностей. Вбудовані тести перевіряють функціональність мікрофонного модуля, стабільність живлення, якість Wi-Fi з'єднання та цілісність програмного забезпечення. Результати діагностики доступні користувачу через меню системи та можуть автоматично передаватися на сервер для дистанційного моніторингу стану обладнання.

Таблиця 2.3 – Параметри точності вимірювань системи моніторингу шуму

Параметр	Значення	Примітка
Діапазон вимірювання	30-120 дБ	При використанні адаптивного підсилення
Точність вимірювань	$\pm 1.5$ дБ	Після калібрування з еталонним джерелом
Частотний діапазон	20 Гц - 8 кГц	Обмежений характеристиками мікрофона
Роздільна здатність	0.1 дБ	Програмна інтерполяція значень АЦП
Часова константа	Швидко (125 мс), Повільно (1 с)	Відповідно до стандарту ІЕС 61672-1

Система розширення функціональності забезпечує можливість підключення додаткових сенсорів та модулів для комплексного моніторингу параметрів навколишнього середовища. Наприклад, інтеграція датчиків температури,

вологості або якості повітря дозволяє аналізувати кореляцію між акустичним забрудненням та іншими екологічними факторами. Програмний інтерфейс системи спроектовано з урахуванням можливості масштабування та адаптації до нових задач без суттєвих змін базової архітектури.

## 2.3 Вибір апаратних модулів мікропроцесорної системи

### 2.3.1 Вибір мікроконтролера ESP8266

Процес вибору мікроконтролера для системи моніторингу рівня шуму ґрунтувався на комплексному аналізі технічних характеристик доступних на ринку рішень з урахуванням специфіки проєктованої системи. ESP8266 привернув увагу завдяки інтегрованому модулю Wi-Fi, що дозволяє суттєво спростити архітектуру системи та зменшити її собівартість. Визначальними факторами при виборі стали також достатня обчислювальна потужність, низьке енергоспоживання та розвинена екосистема програмного забезпечення, що спрощує процес розробки.

Серед різноманітних модифікацій ESP8266 перевагу надано версії ESP-12E/F, яка пропонує оптимальне співвідношення функціональності та вартості. Дана модифікація має достатню кількість виводів загального призначення для підключення всіх необхідних периферійних пристроїв системи моніторингу шуму без застосування додаткових мультиплексорів чи розширювачів портів. Наявність 11 цифрових входів/виходів, одного аналогового входу та підтримка інтерфейсів SPI, I2C, UART забезпечує гнучкість при проєктуванні схеми підключення датчиків та елементів керування.

Таблиця 2.4 – Порівняльний аналіз мікроконтролерів для системи моніторингу рівня шуму

Характеристика	ESP8266 (ESP-12E)	ESP32	Arduino Nano	STM32F103C8
Тактова частота	80-160 МГц	160-240	16 МГц	72 МГц

					КВРКІ.220033.22.01.18ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

		МГц		
Обсяг оперативної пам'яті	80 КБ	520 КБ	2 КБ	20 КБ
Флеш-пам'ять	4 МБ	4 МБ	32 КБ	64-128 КБ
Аналогові входи	1 (10-біт)	12 (12-біт)	8 (10-біт)	10 (12-біт)

Кінець таблиці 2.4

<b>Цифрові входи/виходи</b>	<b>11</b>	<b>38</b>	<b>22</b>	<b>16</b>
Інтерфейси	SPI, I2C, UART	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART	SPI, I2C, UART, CAN
Вбудований Wi-Fi	Так	Так	Ні	Ні
Вбудований Bluetooth	Ні	Так	Ні	Ні
Вартість, грн	75-120	150-250	80-140	90-180

Доступність великої кількості бібліотек та готових програмних компонентів для ESP8266 значно прискорює процес розробки програмного забезпечення системи. Підтримка популярних середовищ розробки, таких як Arduino IDE, ESPlorer, PlatformIO, спрощує процес написання, відлагодження та завантаження програмного коду. Можливість використання різних мов програмування (C++, Lua, MicroPython) надає гнучкість при виборі інструментів розробки відповідно до навичок розробника та специфіки проекту.

### 2.3.2 Вибір модуля мікрофонного датчика

Модуль мікрофонного датчика виступає ключовим елементом системи моніторингу шуму, оскільки безпосередньо впливає на точність та достовірність вимірювань акустичного тиску. При виборі датчика розглядалися різні типи мікрофонів: електретні, конденсаторні, MEMS, динамічні. Аналіз технічних характеристик та співвідношення ціна/якість визначив перевагу електретних мікрофонів з підсилювачем, які забезпечують достатню чутливість та частотний діапазон при помірній вартості та простоті інтеграції.

									Арк.
									27
Змн.	Арк.	№ докум.	Підпис	Дата					

КВРКІ.220033.22.01.18ПЗ

Серед модулів на основі електретних мікрофонів обрано МАХ9814 з вбудованим підсилювачем та автоматичним регулюванням підсилення (AGC). Даний модуль демонструє оптимальні характеристики для систем акустичного моніторингу: лінійну амплітудно-частотну характеристику в діапазоні 50 Гц - 10 кГц, низький рівень власних шумів, регульований коефіцієнт підсилення та автоматичне налаштування чутливості відповідно до рівня вхідного сигналу. Наявність вбудованого АРП дозволяє адаптуватися до різних акустичних умов без перевантаження аналогового входу мікроконтролера.

Таблиця 2.5 – Порівняння модулів мікрофонних датчиків

Параметр	МАХ9814	КУ-038	INMP441	LM393
Тип мікрофона	Електретний	Електретний	MEMS	Електретний
Цифровий вихід	Ні	Так (компаратор)	Так (I2S)	Так (компаратор)
Аналоговий вихід	Так	Так	Ні	Так
Автоматичне регулювання підсилення	Так	Ні	Ні	Ні
Чутливість	-46±2 дБ	-54±3 дБ	-26 дБ МЕКО	-46±3 дБ
Відношення сигнал/шум	>60 дБ	>40 дБ	65 дБ	>50 дБ
Напруга живлення	2.7-5.5 В	3.3-5.0 В	1.8-3.3 В	3.3-5.0 В
Струм споживання	3 мА	5 мА	1.4 мА	4 мА
Вартість, грн	110-160	40-70	90-150	35-60

Суттєвою перевагою обраного модуля МАХ9814 стала можливість програмного налаштування параметрів автоматичного регулювання підсилення через зовнішні контакти. Зокрема, можна обирати один з трьох режимів стиснення динамічного діапазону (1:1, 1:2, 1:4), що дозволяє оптимізувати чутливість системи для різних акустичних сценаріїв. Додаткове налаштування часу атаки та відновлення АРП забезпечує гнучкість при роботі з різними типами акустичних сигналів – від короткочасних імпульсних шумів до постійного фонового гулу.

Додатковою перевагою МАХ9814 є наявність розвиненої технічної документації, що включає рекомендації з монтажу, схеми підключення та формули розрахунку рівня акустичного тиску на основі вихідного сигналу. Таким чином, обраний модуль мікрофонного датчика забезпечує оптимальне поєднання технічних характеристик, зручності інтеграції та наявності документації, що відповідає вимогам проєктованої системи моніторингу рівня шуму.

### 2.3.3 Вибір модуля живлення

Надійне та стабільне електроживлення становить основу безперебійної роботи мікропроцесорної системи моніторингу шуму. Особливу увагу приділено вибору компонентів системи живлення, здатних забезпечити стабільну напругу при різних режимах роботи, включаючи піковий струм при активації Wi-Fi передавача. Врахована також вимога автономної роботи, що потребує ефективного перетворення напруги з мінімальними втратами.

Для забезпечення живлення системи обрано стабілізатор напруги AMS1117-3.3, який перетворює вхідну напругу від 4.5 до 12 В у стабілізовану напругу 3.3 В, необхідну для роботи мікроконтролера ESP8266 та периферійних модулів. Даний стабілізатор забезпечує максимальний вихідний струм до 800 мА, що з запасом покриває потреби всіх компонентів системи, включаючи пікове споживання Wi-Fi модуля під час передачі даних. Низьке падіння напруги (типово 1.1 В) дозволяє ефективно використовувати джерело живлення, продовжуючи час автономної роботи системи.

Таблиця 2.6 – Порівняння модулів стабілізації напруги

Параметр	AMS1117-3.3	LM1117-3.3	MP1584EN	XL6009
Тип перетворювача	Лінійний	Лінійний	Імпульсний (Step-down)	Імпульсний (Step-up/down)
Вхідна напруга	4.5-12 В	4.5-15 В	4.5-28 В	3-32 В

Вихідна напруга	3.3 В (фіксована)	3.3 В (фіксована)	0.8-18 В (регульована)	1.3-35 В (регульована)
Максимальний струм	800 мА	800 мА	3 А	4 А
ККД	30-40%	30-40%	85-95%	85-95%

Кінець таблиці 2.6

<b>Теплове розсіювання</b>	<b>Високе</b>	<b>Високе</b>	<b>Низьке</b>	<b>Низьке</b>
Рівень пульсацій	Низький	Низький	Середній	Середній
Розміри модуля, мм	12×7	12×7	18×11	25×15
Вартість, грн	15-30	18-35	40-70	50-90

Для забезпечення автономної роботи системи передбачено підключення літій-полімерного акумулятора номінальною напругою 3.7 В та ємністю 2000 мАг. При середньому споживанні системи близько 80 мА під час активної роботи такий акумулятор забезпечує до 20-25 годин безперервної роботи. Застосування модуля TP4056 з функцією захисту акумулятора дозволяє безпечно заряджати акумулятор від USB-порту комп'ютера або стандартного зарядного пристрою без ризику перезарядки чи глибокого розряду.

#### 2.3.4 Вибір дисплею для відображення даних

Інтерфейс користувача мікропроцесорної системи моніторингу шуму передбачає візуальне відображення поточних показників рівня шуму, статистичних даних та режимів роботи системи. Вибір дисплею здійснювався з урахуванням вимог до інформативності, енергоефективності та простоти інтеграції з мікроконтролером ESP8266. Розглядалися різні типи дисплеїв: текстові LCD, графічні LCD, OLED, ePaper (електронний папір).

					КВРКІ.220033.22.01.18ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Для реалізації системи обрано OLED-дисплей на контролері SSD1306 з роздільною здатністю 128×64 пікселі, який забезпечує оптимальне співвідношення функціональності, компактності та енергоспоживання. OLED-технологія забезпечує високу контрастність зображення (до 2000:1) та чудову видимість при різних кутах огляду та умовах освітлення без необхідності додаткового підсвічування. Інтерфейс I2C спрощує підключення до мікроконтролера, використовуючи лише два сигнальних проводи (SDA, SCL), що особливо важливо з огляду на обмежену кількість виводів ESP8266.

Таблиця 2.7 – Порівняння дисплеїв для системи моніторингу рівня шуму

Параметр	OLED SSD1306	LCD 1602	SPI TFT 1.8"	e-Paper 2.13"
Тип дисплею	OLED	LCD	TFT LCD	E-Ink
Роздільна здатність	128×64 пікс.	16×2 символи	160×128 пікс.	250×122 пікс.
Інтерфейс	I2C, SPI	I2C, 8-біт	SPI	SPI
Кількість кольорів	Монохромний	Монохромний	65К	Чорно-білий
Необхідність підсвічування	Ні	Так	Так	Ні
Кут огляду	>170°	<140°	<160°	>170°
Видимість при прямому сонячному світлі	Середня	Низька	Низька	Висока
Енергоспоживання (активний режим)	20-30 мА	35-60 мА	80-150 мА	2-3 мА (при оновленні)
Енергоспоживання (сплячий режим)	<5 мкА	20-30 мА	20-40 мА	<5 мкА
Швидкість оновлення	Висока	Середня	Висока	Низька (2-3 сек)
Вартість, грн	70-120	60-90	140-200	250-380

Бібліотеки для роботи з SSD1306 широко представлені у спільноті розробників ESP8266, що спрощує інтеграцію дисплею з програмним забезпеченням системи. Більшість бібліотек оптимізовані для обмежених ресурсів мікроконтролера та надають широкий набір функцій для роботи з текстом,

графікою та інтерфейсними елементами без суттєвого навантаження на процесор та пам'ять. Така оптимізація особливо важлива для мікроконтролера ESP8266, який має обмежені обчислювальні ресурси, що частково задіяні для обробки сигналу з мікрофонного датчика та забезпечення Wi-Fi зв'язку.

### 2.3.5 Вибір додаткових модулів системи

Розширення функціональності мікропроцесорної системи моніторингу рівня шуму потребує інтеграції додаткових модулів, які забезпечують взаємодію з користувачем, зберігання даних, синхронізацію часу та адаптацію до навколишнього середовища. Вибір додаткових компонентів здійснювався з урахуванням їх сумісності з мікроконтролером ESP8266, надійності роботи та мінімального впливу на енергоспоживання системи.

Для забезпечення точної прив'язки вимірювань до реального часу обрано модуль годинника реального часу DS3231, який характеризується високою точністю ходу ( $\pm 2$  ppm) та мінімальним дрейфом навіть при значних коливаннях температури навколишнього середовища. Даний модуль підключається до мікроконтролера через інтерфейс I2C, використовуючи ті ж лінії зв'язку, що й OLED-дисплей, що дозволяє економити обмежені ресурси портів вводу-виводу ESP8266. Вбудована літієва батарея забезпечує безперервну роботу годинника навіть при відключенні основного живлення системи, що особливо важливо для збереження хронології вимірювань при автономній роботі.

Таблиця 2.8 – Характеристики додаткових модулів системи

Модуль	Тип інтерфейсу	Напруга живлення	Струм споживання	Функціональне призначення	Вартість, грн
DS3231 (RTC)	I2C	3.3-5.0 В	0.2 мА	Хронометраж вимірювань	50-80
SD карта	SPI	3.3 В	50-100 мА (запис)	Локальне зберігання даних	40-70
DHT22	1-Wire	3.3-5.0 В	2.5 мА	Вимірювання температури та вологості	80-130

BMP280	I2C, SPI	1.8-3.6 В	2.7 мкА	Вимірювання атмосферного тиску	55-90
KY-040 (енкодер)	GPIO	3.3-5.0 В	<1 мА	Навігація по меню системи	25-45
WS2812B LED	GPIO	5.0 В	60 мА (на світлодіод)	Світлова індикація рівня шуму	30-60

Для зберігання історії вимірювань та конфігураційних параметрів системи обрано модуль карти пам'яті microSD з інтерфейсом SPI. Такий вибір обумовлений необхідністю зберігання великих обсягів даних при тривалому моніторингу та можливістю легкого перенесення цих даних на комп'ютер для подальшого аналізу. Використання файлової системи FAT32 забезпечує сумісність з більшістю операційних систем без необхідності встановлення додаткових драйверів чи програмного забезпечення. Реалізація підтримки SD карт для ESP8266 не потребує складних програмних рішень завдяки наявності готових бібліотек, оптимізованих для роботи з обмеженими ресурсами мікроконтролера.

Світлова індикація рівня шуму реалізована за допомогою адресних RGB-світлодіодів WS2812B, які дозволяють відображати поточний рівень шуму кольоровою шкалою від зеленого (низький рівень) до червоного (критичний рівень). Використання адресних світлодіодів дозволяє керувати кожним елементом індикації окремо через єдину лінію даних, що зменшує кількість необхідних виводів мікроконтролера. Додатково реалізовано режим зниженої яскравості для економії енергії при автономній роботі та автоматичне вимкнення індикації в режимі глибокого сну.

## 2.4 Вибір програмного забезпечення для розробки

### 2.4.1 Вибір середовища програмування

Розробка програмного забезпечення для мікропроцесорної системи моніторингу рівня шуму потребує відповідного середовища програмування, яке має забезпечувати зручну роботу з кодом, ефективну компіляцію та завантаження програми до мікроконтролера ESP8266. Сучасний ринок пропонує різноманітні варіанти середовищ розробки, кожне з яких має специфічні переваги та недоліки

					КВРКІ.220033.22.01.18ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

щодо функціональності, швидкодії, зручності використання та підтримки апаратної платформи ESP8266.

Arduino IDE вирізняється з-поміж конкурентів незаперечною перевагою – надзвичайно простим інтерфейсом та низьким порогом входження для початківців, що робить його ідеальним вибором для освітніх проєктів та швидкого прототипування. Офіційна підтримка ESP8266 через додаткові плати в менеджері плат забезпечує безпроблемну інтеграцію та доступ до всіх функцій мікроконтролера без необхідності складного налаштування середовища розробки. Крім того, величезна кількість готових бібліотек та прикладів коду в галереї Arduino прискорює процес розробки та дозволяє швидко реалізувати типові функції системи моніторингу шуму.

Таблиця 2.9 – Порівняльний аналіз середовищ програмування для ESP8266

Характеристика	Arduino IDE	PlatformIO	ESPlorer	ESP-IDF
Мова програмування	C/C++	C/C++, Python, JavaScript	Lua, MicroPython	C/C++
Підтримка ESP8266	Через менеджер плат	Вбудована	Нативна	Обмежена
Редактор коду	Базовий	Розширений (VS Code)	Базовий	Залежить від IDE
Відлагоджувач	Обмежений (Serial Monitor)	Розширений	Базовий (Serial)	Розширений
Менеджер бібліотек	Вбудований	Вбудований	Ручне завантаження	Ручне завантаження
Автозавершення коду	Базове	Розширене	Базове	Залежить від IDE
Підтримка Git	Ні	Так	Ні	Залежить від IDE
Можливість командного рядка	Обмежена	Розширена	Обмежена	Розширена
Кросплатформніс	Windows,	Windows,	Windows, macOS,	Windows,

ть	macOS, Linux	macOS, Linux	Linux	macOS, Linux
Споживання ресурсів	Низьке	Середнє	Низьке	Високе
Зручність для початківців	Висока	Середня	Висока для Lua	Низька
Вартість	Безкоштовно	Безкоштовно	Безкоштовно	Безкоштовно

Незважаючи на простоту Arduino IDE, для складніших проєктів з розгалуженою структурою та великим обсягом коду доцільно розглянути PlatformIO, який надає значно потужніший інструментарій розробника. Інтеграція з Visual Studio Code забезпечує передові можливості редагування коду, включаючи інтелектуальне автозавершення, рефакторинг, навігацію по проєкту та інтегровану систему контролю версій. Менеджер залежностей автоматично вирішує конфлікти бібліотек та забезпечує узгоджену версійність компонентів, що критично важливо для стабільності складних програмних комплексів.

PlatformIO пропонує багаторівневу систему конфігурації проєкту через файл platformio.ini, що дозволяє гнучко налаштовувати параметри компіляції, завантаження та відлагодження програми залежно від поточних потреб розробника. Декларативний підхід до конфігурації спрощує відтворення середовища розробки на різних комп'ютерах та забезпечує однакові умови роботи для всіх учасників команди розробників. Додаткова перевага полягає в можливості визначення різних профілів збірки для тестування, відлагодження та випуску програмного забезпечення без необхідності ручного налаштування параметрів компілятора.

Для розробки системи моніторингу рівня шуму обрано PlatformIO.

Таблиця 2.10 – Налаштування проєкту в PlatformIO для системи моніторингу шуму

Параметр	Значення	Опис
platform	espressif8266	Платформа

		розробки для ESP8266
board	nodemcu2	Тип плати (NodeMCU v2)
framework	arduino	Фреймворк розробки

Кінець таблиці 2.10

<b>monitor_speed</b>	<b>115200</b>	<b>Швидкість послідовного порту для моніторингу</b>
board_build.f_cpu	160000000L	Тактова частота процесора (160 МГц)
board_build.flash_mode	dio	Режим роботи флеш-пам'яті
lib_deps	ArduinoJson, PubSubClient, ESP8266WiFi, Adafruit SSD1306, Adafruit GFX, RTCLib, SD	Залежності проекту (бібліотеки)
build_flags	-D MQTT_MAX_PACKET_SIZE=512 -D VERSION=1.0.0	Прапорці компіляції
upload_speed	921600	Швидкість завантаження прошивки

Впровадження систематичного підходу до керування версіями проекту через інтеграцію Git в PlatformIO дозволяє відстежувати зміни в коді, експериментувати з новими функціями без ризику пошкодження основної версії програми та забезпечує можливість повернення до попередніх станів проекту при виявленні критичних помилок. Крім того, зберігання проекту в репозиторії спрощує резервне копіювання та колективну роботу над кодом, що особливо актуально для складних систем, які розвиваються тривалий час.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Для початкових етапів розробки та швидкого прототипування окремих модулів системи використовується ESPlorer – спеціалізоване середовище для інтерактивної розробки програм на Lua для ESP8266. Інтерпретована природа Lua дозволяє змінювати окремі компоненти програми без повного перепрограмування мікроконтролера, що значно прискорює цикл розробки-тестування при налагодженні алгоритмів обробки сигналів та взаємодії з периферійними пристроями.

#### 2.4.2 Вибір бібліотек для роботи з ESP8266

Реалізація функціональності мікропроцесорної системи моніторингу рівня шуму суттєво спрощується за рахунок використання спеціалізованих бібліотек, які надають високорівневий інтерфейс до апаратних можливостей ESP8266 та підключених периферійних пристроїв. Вибір оптимального набору бібліотек визначає ефективність використання обмежених ресурсів мікроконтролера, стабільність роботи системи та швидкість розробки програмного забезпечення.

Основа програмного забезпечення системи складають офіційні бібліотеки ESP8266 для Arduino, які забезпечують доступ до специфічних функцій мікроконтролера: управління Wi-Fi, енергозбереження, файлова система, оновлення прошивки «по повітрю» (OTA). Бібліотека ESP8266WiFi реалізує повноцінний стек протоколів TCP/IP з підтримкою різних режимів роботи: станція, точка доступу або комбінований режим. Розширені можливості управління Wi-Fi з'єднанням, включаючи автоматичне перепідключення при втраті зв'язку та конфігурацію через веб-інтерфейс, забезпечують надійну комунікацію системи з хмарними сервісами збору даних.

Для реалізації протоколу MQTT, який забезпечує легковагий обмін повідомленнями між системою моніторингу та хмарною платформою, обрано бібліотеку PubSubClient. Ця бібліотека оптимізована для роботи на пристроях з обмеженими ресурсами та забезпечує надійну доставку повідомлень з підтримкою

					КВРКІ.220033.22.01.18ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

різних рівнів якості обслуговування (QoS). Налаштування параметрів з'єднання, включаючи автоматичне повторне підключення при втраті зв'язку та збереження останнього відомого стану, забезпечує стабільну роботу системи в умовах нестабільного мережевого з'єднання.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

аблиця 2.11 – Основні бібліотеки для розробки системи моніторингу шуму

Назва бібліотеки	Версія	Функціональне призначення	Розмір, КБ	Ліцензія
ESP8266WiFi	2.7.4	Управління Wi-Fi з'єднанням	528	LGPL-2.1
PubSubClient	2.8.0	Реалізація протоколу MQTT	38	MIT
ArduinoJson	6.19.4	Обробка JSON даних	105	MIT
Adafruit SSD1306	2.5.1	Керування OLED-дисплеєм	204	BSD
Adafruit GFX	1.11.3	Графічні примітиви для дисплея	125	BSD
RTClib	2.1.1	Інтерфейс до годинника реального часу	57	MIT
SD	1.2.4	Робота з SD-картою	264	LGPL-2.1
DHT sensor library	1.4.4	Робота з датчиками температури та вологості	23	MIT
Adafruit BMP280	2.6.6	Інтерфейс до барометричного датчика	47	BSD
ESPAsyncWebServer	1.2.3	Асинхронний веб-сервер	392	LGPL-2.1
FastLED	3.5.0	Керування адресними світлодіодами	183	MIT
RunningMedian	0.3.5	Статистична обробка вимірювань	17	MIT

Для ефективної обробки сигналу з мікрофонного датчика розроблено власну бібліотеку NoiseLevel, яка поєднує функції оцифрування аналогового сигналу, частотної фільтрації (включаючи А-зважування відповідно до стандарту IEC 61672), обчислення середньоквадратичного значення та переведення електричних величин у децибелі. Бібліотека оптимізована для роботи на ESP8266 з урахуванням обмежень 10-бітного АЦП та реалізує алгоритми розширення динамічного діапазону через програмне регулювання коефіцієнта підсилення вхідного сигналу.

Таблиця 2.12 – Функції власної бібліотеки NoiseLevel для обробки акустичного сигналу

Функція	Призначення	Складність обчислень	Використання пам'яті
readSample()	Зчитування миттєвого значення з АЦП	O(1)	Мінімальне
calculateRMS()	Розрахунок середньоквадратичного значення	O(n)	Буфер n зразків
applyAWeighting()	Застосування А-зважування до сигналу	O(n)	Коефіцієнти фільтра
convertToDecibels()	Конвертація електричних величин у дБ	O(1)	Мінімальне
calibrate()	Калібрування вимірювального тракту	O(n)	Калібрувальні константи
getPeakLevel()	Визначення пікового рівня шуму	O(n)	Буфер n зразків
getLeq()	Розрахунок еквівалентного рівня шуму	O(n)	Буфер історії вимірювань
detectFrequencyComponents()	Аналіз спектрального складу шуму (FFT)	O(n log n)	Буфер n зразків, twiddle фактори
startContinuousMonitoring()	Запуск безперервного моніторингу	O(1)	Мінімальне
setThreshold()	Встановлення порогового значення	O(1)	Мінімальне

Математична обробка результатів вимірювань, включаючи статистичний аналіз та фільтрацію викидів, реалізована з використанням бібліотеки

RunningMedian, яка забезпечує обчислення ковзної медіани для потокових даних з мінімальними витратами пам'яті. Такий підхід дозволяє ефективно фільтрувати короткочасні сплески шуму, які не відображають загальну акустичну картину середовища, без втрати інформації про значимі зміни рівня шуму.

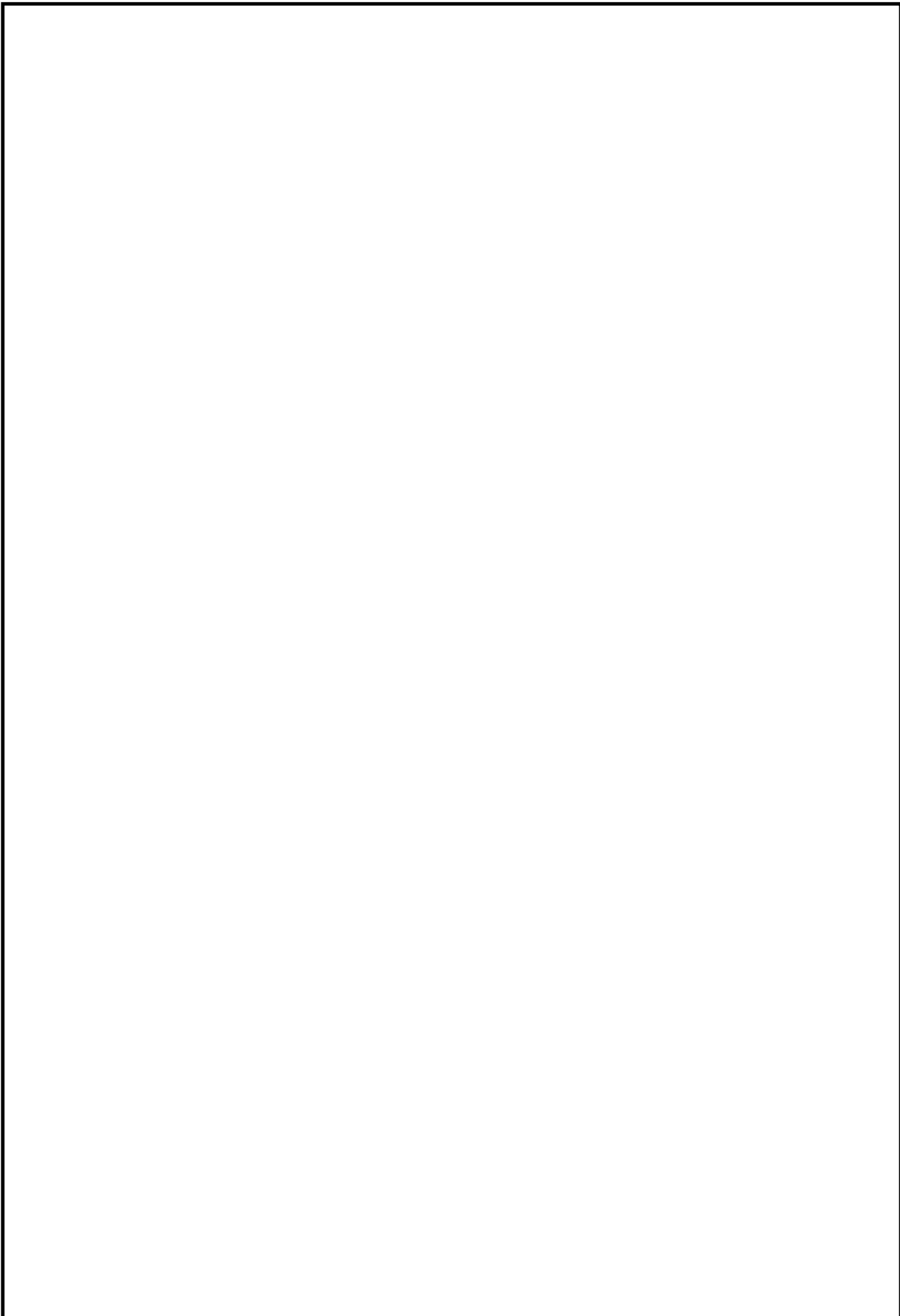
Взаємодія з додатковими сенсорами реалізована через відповідні спеціалізовані бібліотеки: DHT sensor library для вимірювання температури та вологості, Adafruit BMP280 для вимірювання атмосферного тиску, RTCLib для роботи з годинником реального часу. Уніфікований інтерфейс до різних типів датчиків спрощує інтеграцію додаткових сенсорів у систему моніторингу та забезпечує консистентний підхід до обробки даних з різних джерел.

#### 2.4.3 Вибір хмарної платформи для збору даних

Сучасний підхід до моніторингу навколишнього середовища передбачає не лише локальну обробку та відображення даних, але й їх агрегацію в централізованих хмарних системах для подальшого аналізу, візуалізації та формування статистичних звітів. Вибір хмарної платформи для збору даних з мікропроцесорної системи моніторингу рівня шуму визначає масштабованість рішення, доступність даних для кінцевих користувачів та можливості інтеграції з іншими системами моніторингу навколишнього середовища.

Проведено комплексний аналіз популярних хмарних платформ IoT, які підтримують збір та обробку даних з розподілених сенсорів: AWS IoT Core, Google Cloud IoT, Microsoft Azure IoT Hub, ThingSpeak, Ubidots, Blynk та інших. Критеріями відбору слугували: підтримка протоколу MQTT для ефективної передачі даних, наявність безкоштовного тарифу для невеликих проєктів, можливості візуалізації даних, підтримка тригерів та сповіщень, наявність API для інтеграції з іншими системами, а також вимоги до апаратних ресурсів клієнтської частини.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		



					КВРКІ.220033.22.01.18ПЗ	Арк.
						42
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 2.13 – Порівняння хмарних платформ для збору даних з системи моніторингу шуму

Платформа	Протоколи	Безкоштовний тариф	Візуалізація	Зберігання даних	API	Складність інтеграції
ThingSpeak	MQTT, HTTP	До 3 млн повідомлень/рік	Базові графіки	1 рік	REST	Низька
Ubidots	MQTT, HTTP	До 5 пристроїв	Розширені дашборди	1 місяць	REST	Середня
AWS IoT Core	MQTT, HTTP, WebSockets	250 тис. повідомлень/місяць	Через AWS QuickSight	Необмежене (платне)	REST, SDK	Висока
Google Cloud IoT	MQTT, HTTP	250 МБ/місяць	Через Data Studio	Необмежене (платне)	REST, SDK	Висока
Azure IoT Hub	MQTT, AMQP, HTTP	8 тис. повідомлень/день	Power BI	Необмежене (платне)	REST, SDK	Висока
Blynk	HTTP, Blynk Protocol	До 5 віртуальних пінів	Мобільний додаток	Обмежене	REST	Низька
Thingier.io	MQTT, HTTP	До 2 пристроїв	Інтерактивні дашборди	30 днів	REST	Середня
Node-RED	MQTT, HTTP, WebSockets	Самостійний хостинг	Кастомізовані вузли	Залежить від БД	REST	Середня

Для реалізації системи моніторингу рівня шуму обрано платформу ThingSpeak, яка оптимально поєднує простоту інтеграції з ESP8266, достатні

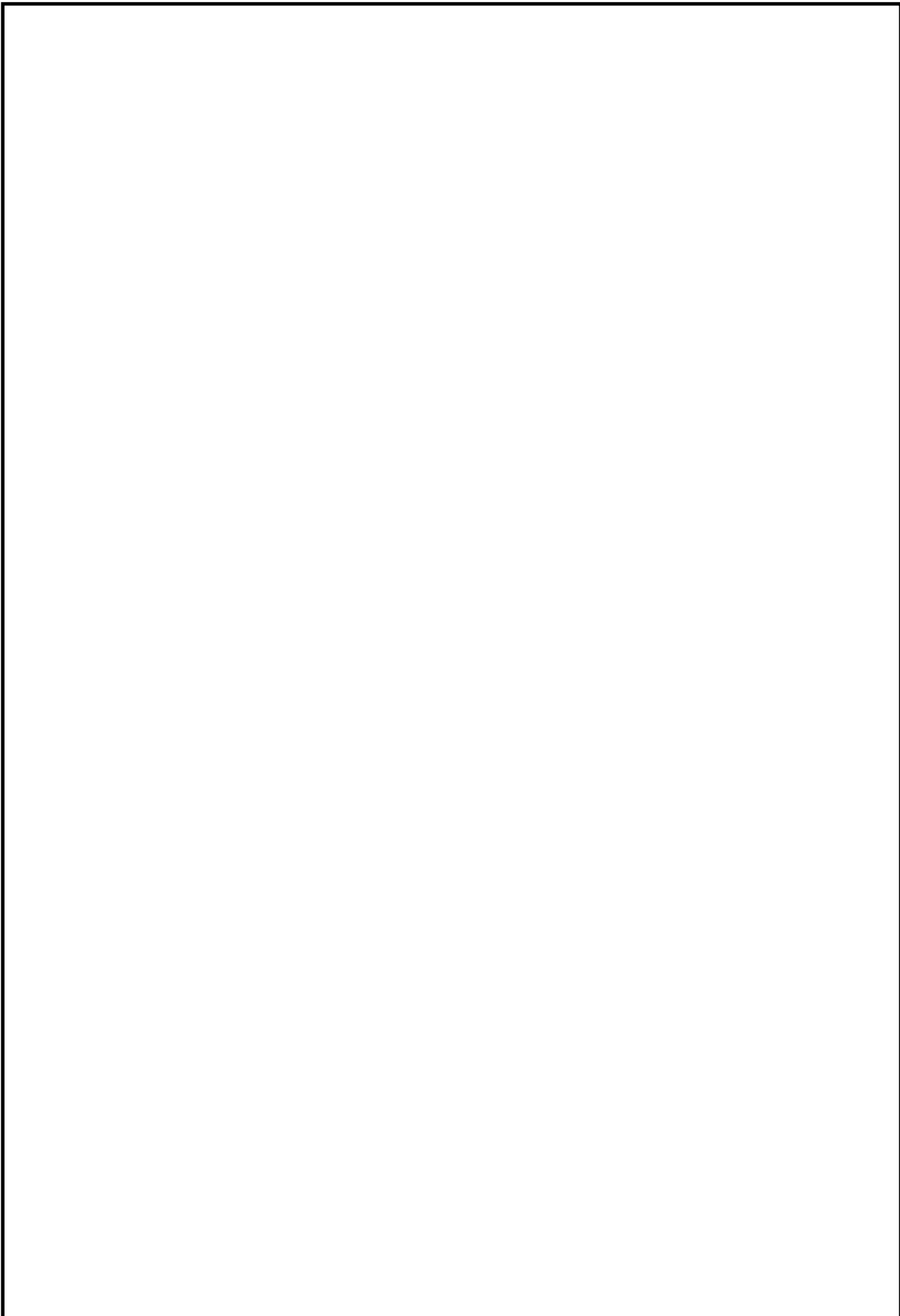
функціональні можливості та наявність безкоштовного тарифу з прийнятними обмеженнями для експериментального проєкту. ThingSpeak розроблена компанією MathWorks (творцями MATLAB) та орієнтована на збір, візуалізацію та аналіз даних з IoT-пристроїв. Платформа підтримує протоколи HTTP та MQTT для передачі даних, що забезпечує гнучкість при виборі методу комунікації залежно від умов експлуатації системи моніторингу шуму.

Унікальною перевагою ThingSpeak є можливість обробки даних за допомогою MATLAB, що дозволяє реалізувати складні алгоритми аналізу акустичних показників безпосередньо на хмарній платформі без необхідності завантажувати дані на локальний комп'ютер. Функція ThingSpeak Analysis надає можливість застосовувати готові або власні MATLAB-скрипти для статистичної обробки даних, виявлення аномалій, прогнозування трендів та кореляції акустичних показників з іншими екологічними параметрами.

Процес інтеграції системи моніторингу шуму з ThingSpeak включає кілька етапів: створення облікового запису, налаштування каналу для збору даних, отримання API-ключів для запису та читання даних, реалізація передачі даних у програмному забезпеченні ESP8266 та налаштування візуалізації даних на платформі. Для забезпечення безпеки передбачено шифрування даних за допомогою протоколу TLS, аутентифікацію пристроїв за допомогою унікальних ключів та можливість обмеження доступу до каналів даних на рівні IP-адрес.

Оптимізація передачі даних на платформу ThingSpeak здійснюється з урахуванням обмежень безкоштовного тарифу та специфіки вимірювання рівня шуму. Замість безперервної передачі всіх вимірювань, що призвело б до швидкого вичерпання ліміту повідомлень, реалізовано агрегацію даних на рівні мікроконтролера з передачею усереднених показників за певний інтервал часу (зазвичай 1-5 хвилин). Додатково впроваджено механізм умовної передачі даних, який активується лише при суттєвій зміні рівня шуму або перевищенні встановлених порогових значень.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		



					КВРКІ.220033.22.01.18ПЗ	Арк.
						45
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 2.14 – Структура даних для передачі на платформу ThingSpeak

Поле	Тип даних	Одиниці вимірювання	Опис	Частота оновлення
field1	Число з плаваючою точкою	дБ(А)	Поточний рівень шуму (А-зважений)	1-5 хвилин
field2	Число з плаваючою точкою	дБ(А)	Мінімальний рівень шуму за період	5-15 хвилин
field3	Число з плаваючою точкою	дБ(А)	Максимальний рівень шуму за період	5-15 хвилин
field4	Число з плаваючою точкою	дБ(А)	Еквівалентний рівень шуму (Leq)	15-60 хвилин
field5	Число з плаваючою точкою	°С	Температура навколишнього середовища	15-60 хвилин
field6	Число з плаваючою точкою	%	Відносна вологість повітря	15-60 хвилин
field7	Число з плаваючою точкою	гПа	Атмосферний тиск	15-60 хвилин
field8	Ціле число	мВ	Напруга батареї	60 хвилин
status	Текст	-	Статус системи та службові повідомлення	При зміні статусу

Перспективним напрямком розвитку хмарної складової системи є інтеграція з публічними сервісами екологічного моніторингу та відкритими даними, що дозволить створити комплексну карту шумового забруднення міського середовища на основі даних з розподіленої мережі сенсорів. Наявність відкритого API платформи ThingSpeak спрощує розробку додаткових сервісів аналізу та візуалізації даних, орієнтованих на специфічні потреби різних категорій користувачів: екологічних служб, містобудівників, дослідників впливу шуму на здоров'я людини.

## 2.5 Висновки

1. Розроблено структуру мікропроцесорної системи моніторингу рівня шуму, що забезпечує гнучкість конфігурації, енергоефективність та можливість інтеграції з існуючими інформаційними системами. Обрано оптимальні апаратні компоненти: мікроконтролер ESP8266 (NodeMCU), мікрофонний модуль MAX9814, OLED-дисплей SSD1306 та модуль живлення на базі TP4056.

2. Визначено функціональне призначення основних модулів системи та обґрунтовано вибір програмного забезпечення для розробки: середовище Arduino IDE з бібліотеками для ESP8266, хмарна платформа ThingSpeak для збору даних та Node-RED для створення веб-інтерфейсу, що забезпечує оптимальне співвідношення функціональності, складності розробки та продуктивності.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ

#### 3.1 Розроблення функціональної схеми мікропроцесорної системи

Функціональна схема мікропроцесорної системи моніторингу рівня шуму ілюструє взаємодію окремих модулів та інформаційні потоки між компонентами системи. Структурна організація системи базується на модульному принципі, що забезпечує можливість заміни окремих компонентів без суттєвої перебудови всієї системи. Такий підхід підвищує ремонтпридатність та спрощує модернізацію системи в майбутньому.

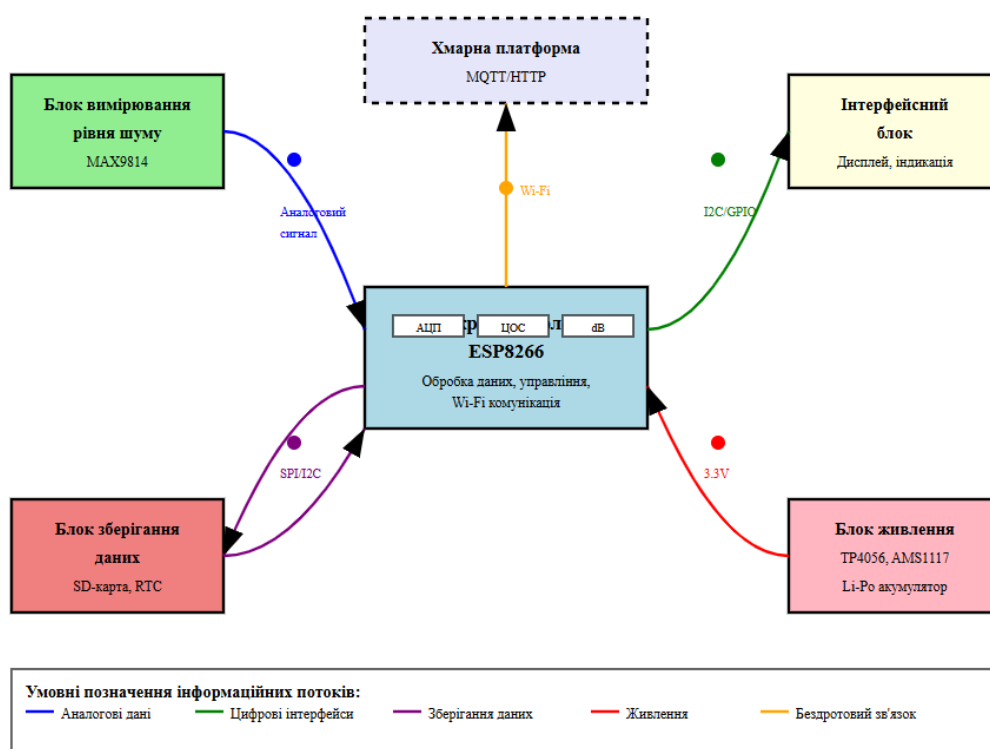


Рисунок 3.1 – Функціональна схема мікропроцесорної системи

Центральним елементом функціональної схеми виступає мікроконтролер ESP8266, який забезпечує управління всіма периферійними модулями, обробку даних та комунікацію із зовнішніми системами. Архітектура ESP8266 дозволяє одночасно виконувати декілька критичних завдань: зчитування даних з аналогового датчика шуму через АЦП, обробку отриманих значень для визначення рівня шуму в децибелах, відображення інформації на дисплеї та передачу даних через Wi-Fi модуль на хмарну платформу. Контролер також відповідає за керування режимами енергоспоживання системи та реалізацію функцій самодіагностики.

Функціональний блок вимірювання рівня шуму складається з мікрофонного модуля MAX9814 з вбудованим підсилювачем та схемою автоматичного регулювання підсилення. Електричний сигнал з виходу мікрофона подається на аналоговий вхід мікроконтролера, де відбувається його оцифрування за допомогою вбудованого 10-бітного АЦП. Програмна обробка оцифрованого сигналу включає фільтрацію, обчислення середньоквадратичного значення та перетворення у децибели з урахуванням калібрувальних коефіцієнтів.

Інтерфейсний блок об'єднує компоненти для взаємодії з користувачем: OLED-дисплей на контролері SSD1306 для відображення поточних показників та стану системи, RGB-світлодіоди для кольорової індикації рівня шуму, кнопки навігації по меню та налаштування параметрів системи. Дисплей підключається через інтерфейс I2C, що дозволяє мінімізувати кількість використаних виводів мікроконтролера. Елементи керування підключаються до цифрових входів ESP8266 з програмною реалізацією захисту від брязкоту контактів.

Інформаційний обмін між компонентами системи реалізовано з використанням різних інтерфейсів залежно від вимог до швидкості передачі даних та кількості доступних виводів мікроконтролера. Високошвидкісний послідовний інтерфейс SPI застосовується для взаємодії з модулем SD-карти, що забезпечує високу пропускну здатність при збереженні великих обсягів даних. Двопровідний інтерфейс I2C використовується для комунікації з дисплеєм,

					КВРКІ.220033.22.01.18ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

годинником реального часу та датчиками навколишнього середовища, що дозволяє підключати декілька периферійних пристроїв до одних і тих же ліній даних.

Функціональний блок живлення забезпечує стабілізовану напругу 3.3 В для всіх компонентів системи як при роботі від мережевого адаптера, так і від акумуляторної батареї. Модуль TP4056 реалізує функцію зарядки літій-полімерного акумулятора з захистом від перезаряду та глибокого розряду. Система передбачає можливість моніторингу рівня заряду акумулятора через подільник напруги, підключений до одного з аналогових входів мікроконтролера. Для економії енергії при автономній роботі реалізовано програмне управління живленням периферійних модулів з можливістю повного відключення неактивних компонентів.

Таблиця 3.1 – Основні інформаційні потоки мікропроцесорної системи моніторингу рівня шуму

Джерело	Приймач	Тип даних	Інтерфейс	Частота обміну
Мікрофонний модуль	Мікроконтролер	Аналоговий сигнал	АЦП	1-10 кГц
Мікроконтролер	OLED-дисплей	Графічні та текстові дані	I2C	1-5 Гц
Годинник реального часу	Мікроконтролер	Дата та час	I2C	1 Гц
Датчики середовища	Мікроконтролер	Температура, вологість, тиск	I2C	0.1-1 Гц
Мікроконтролер	SD-карта	Файли з історією вимірювань	SPI	За запитом
Мікроконтролер	RGB-світлодіоди	Стан індикації	GPIO	За зміни стану
Елементи керування	Мікроконтролер	Команди користувача	GPIO	За подією
Мікроконтролер	Хмарна платформа	JSON-повідомлення	MQTT/HTTP	0.01-0.1 Гц

### 3.2 Розроблення електричної принципової схеми мікропроцесорної системи

Електрична принципова схема системи моніторингу рівня шуму розроблена з урахуванням вимог до надійності, електромагнітної сумісності та енергоефективності (Додаток В). Схема включає всі необхідні компоненти для забезпечення стабільної роботи системи в різних умовах експлуатації, включаючи елементи захисту від перенапруг, фільтрації живлення та розв'язки сигнальних ліній. Загальний підхід до проектування схеми базується на мінімізації кількості зовнішніх компонентів за рахунок максимального використання вбудованих можливостей мікроконтролера ESP8266.

Оснoву схеми становить модуль ESP-12F на базі мікроконтролера ESP8266, який обрано з огляду на оптимальне поєднання функціональності, доступності та наявності достатньої кількості виводів для підключення всіх необхідних периферійних пристроїв. Модуль містить вбудовану флеш-пам'ять об'ємом 4 МБ, що забезпечує достатньо простору для програмного коду, веб-інтерфейсу та локального зберігання даних. Для стабільної роботи Wi-Fi модуля на платі передбачено мікрополоскову антену з узгоджувальним ланцюгом, що забезпечує надійну комунікацію на відстані до 50 метрів у приміщенні.

#### 3.2.1 Під'єднання до мікропроцесорної плати мікрофонного датчика

Підключення мікрофонного модуля MAX9814 до мікроконтролера ESP8266 реалізовано з урахуванням особливостей аналогового тракту та вимог до якості сигналу для забезпечення точних вимірювань рівня шуму. Для захисту аналогового входу мікроконтролера від можливих перенапруг на виході мікрофонного підсилювача встановлено обмежувальний діод Шотткі та резистивний дільник напруги, що обмежує вхідний сигнал діапазоном 0-1 В, допустимим для АЦП мікроконтролера.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Схема підключення передбачає додаткову фільтрацію сигналу за допомогою RC-ланцюга для зменшення впливу високочастотних завад, які можуть виникати від Wi-Fi модуля та інших цифрових компонентів системи. Підсилення мікрофонного тракту налаштовується через вивід GAIN модуля MAX9814, який підключається до програмованого виходу мікроконтролера через резистивну матрицю для встановлення одного з трьох фіксованих коефіцієнтів підсилення: 40 дБ, 50 дБ або 60 дБ. Така конфігурація дозволяє програмно адаптувати чутливість системи до різних акустичних умов.

Живлення мікрофонного модуля здійснюється від стабілізованої лінії 3.3 В з додатковою фільтрацією за допомогою керамічного конденсатора ємністю 100 нФ для зменшення проникнення цифрових шумів у аналоговий тракт. Для подальшого покращення співвідношення сигнал/шум використано екранований кабель для з'єднання мікрофонного модуля з основною платою та забезпечено просторове розділення аналогової та цифрової частин схеми на друкованій платі.

Унікальною особливістю реалізації мікрофонного тракту є схема автоматичного пробудження системи при перевищенні певного рівня шуму, що дозволяє суттєво знизити енергоспоживання в режимі очікування. 52омпараторн а базі мікросхеми LM393 порівнює сигнал з мікрофона з опорним напругою, яка встановлюється програмно через ЦАП мікроконтролера. При перевищенні порогового рівня шуму компаратор генерує сигнал переривання, який виводить мікроконтролер з режиму глибокого сну.

### 3.2.2 Під'єднання до мікропроцесорної плати модуля дисплею

Інтеграція OLED-дисплея на контролері SSD1306 до системи моніторингу шуму реалізована через двопровідний інтерфейс I2C, що мінімізує кількість задіяних виводів мікроконтролера. Схема підключення передбачає використання виводів GPIO4 (SDA) та GPIO5 (SCL) мікроконтролера ESP8266, які програмно конфігуруються для роботи в режимі I2C. Додатково на лініях інтерфейсу

					КВРКІ.220033.22.01.18ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

встановлено підтягуючі резистори номіналом 4.7 кОм для забезпечення надійного логічного рівня сигналів відповідно до специфікації I2C.

Живлення дисплею здійснюється від лінії 3.3 В з локальною фільтрацією за допомогою конденсатора ємністю 10 мкФ для згладжування можливих пульсацій напруги. Для програмного управління живленням дисплею з метою економії енергії при автономній роботі системи передбачено можливість відключення живлення через польовий транзистор, керований цифровим виходом мікроконтролера. Такий підхід дозволяє повністю знеструмити дисплей у режимі глибокого сну, зменшуючи споживання системи до мінімуму.

Програмне керування яскравістю дисплею реалізовано через модифікацію регістрів контролера SSD1306, що дозволяє адаптувати яскравість залежно від умов освітлення та режиму роботи системи. Додатково впроваджено функцію автоматичного відключення дисплею після певного періоду неактивності користувача, що суттєво подовжує час автономної роботи системи від акумулятора без втрати функціональності моніторингу шуму.

### 3.2.3 Під'єднання до мікропроцесорної плати модуля живлення

Система живлення мікропроцесорної системи моніторингу шуму базується на модулі TP4056 для зарядки літій-полімерного акумулятора та лінійному стабілізаторі AMS1117-3.3 для забезпечення стабільної напруги 3.3 В, необхідної для роботи мікроконтролера та периферійних компонентів. Схема живлення передбачає можливість роботи як від зовнішнього джерела через роз'єм microUSB, так і від акумулятора ємністю 2000 мАг при автономному використанні.

Вхідне живлення від USB-порту (5 В) подається на модуль TP4056, який забезпечує контрольований процес зарядки літій-полімерного акумулятора зі струмом до 1 А та захист від перезаряду, перегріву та короткого замикання. Напруга з акумулятора (3.7-4.2 В) або безпосередньо з виходу модуля TP4056

					КВРКІ.220033.22.01.18ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

подається на вхід стабілізатора AMS1117-3.3, який забезпечує стабільну вихідну напругу 3.3 В навіть при значних коливаннях вхідної напруги та зміні навантаження.

Для захисту схеми від неправильної полярності підключення та перенапруг встановлено діод Шотткі та супресор. Фільтрація завад здійснюється за допомогою електролітичних конденсаторів на вході та виході стабілізатора, а також керамічних конденсаторів для фільтрації високочастотних завад. Додатково передбачено світлодіодну індикацію режимів роботи системи живлення: наявність зовнішнього живлення, процес зарядки акумулятора та завершення зарядки.

Для моніторингу рівня заряду акумулятора реалізовано схему вимірювання напруги через резистивний дільник, підключений до аналогового входу мікроконтролера. Програмна калібрація дозволяє точно визначати відсоток заряду акумулятора з урахуванням нелінійної залежності напруги від залишкової ємності для літій-полімерних акумуляторів. Реалізовано також захист від глибокого розряду акумулятора через програмне відключення неактивних модулів та перехід у режим сну при досягненні критичного рівня заряду.

Таблиця 3.2 – Розрахунок енергоспоживання мікропроцесорної системи моніторингу шуму

Компонент	Активний режим, мА	Режим сну, мА	Робочий цикл, %	Середнє споживання, мА
ESP8266	80 (160 МГц)	0.02	10	8.00
Wi-Fi модуль	70 (передача)	0	1	0.70
Мікрофонний модуль MAX9814	3	3	100	3.00
OLED-дисплей SSD1306	20	0	5	1.00
Датчик температури та вологості DHT22	2.5	0.05	1	0.08
Модуль RTC DS3231	0.2	0.2	100	0.20
RGB-світлодіоди	15 (на 3 світлодіоди)	0	5	0.75

Схема моніторингу акумулятора	0.1	0.1	100	0.10
Загальне споживання	-	-	-	13.83

### 3.2.4 Інтеграція додаткових компонентів системи

Розширення функціональності мікропроцесорної системи моніторингу рівня шуму реалізовано через інтеграцію додаткових модулів: годинника реального часу DS3231, модуля карти пам'яті microSD, датчиків навколишнього середовища та елементів керування. Схема підключення додаткових компонентів оптимізована для мінімізації кількості використаних виводів мікроконтролера та забезпечення надійної комунікації між модулями.

Модуль годинника реального часу DS3231 підключається до шини I2C паралельно з OLED-дисплеєм, використовуючи ті ж лінії SDA та SCL. Такий підхід дозволяє економити виводи мікроконтролера та спрощує програмну реалізацію взаємодії з обома пристроями. Годинник забезпечує точний відлік часу навіть при відключенні основного живлення завдяки вбудованій батареї, що особливо важливо для хронологічної прив'язки вимірювань рівня шуму.

Модуль карти пам'яті microSD підключається через інтерфейс SPI з використанням виводів GPIO12 (MISO), GPIO13 (MOSI), GPIO14 (SCLK) та GPIO15 (CS) мікроконтролера ESP8266. Додатково встановлено підтягуючі резистори на лініях MISO та CS для забезпечення стабільних логічних рівнів сигналів. Карта пам'яті використовується для локального зберігання історії вимірювань, що дозволяє системі функціонувати навіть при відсутності підключення до мережі Wi-Fi.

Датчики навколишнього середовища (температура, вологість, атмосферний тиск) інтегровані до системи для аналізу впливу метеорологічних умов на акустичні характеристики середовища. Датчик DHT22 підключається до цифрового виводу GPIO2 з використанням однопровідного протоколу передачі даних. Барометричний датчик BMP280 підключається до шини I2C,

										Арк.
										55
Змн.	Арк.	№ докум.	Підпис	Дата	КВРКІ.220033.22.01.18ПЗ					

використовуючи ті ж лінії, що й дисплей та годинник реального часу, але з іншою адресою пристрою.

Елементи керування системою реалізовані у вигляді тактових кнопок, підключених до цифрових входів мікроконтролера через схеми захисту від брязкоту контактів. Кожна кнопка має підтягуючий резистор та RC-ланцюг для фільтрації короткочасних імпульсів, що можуть виникати при натисканні. Додатково впроваджено поворотний енкодер для зручної навігації по меню системи, який підключається до двох цифрових входів для відстеження напрямку обертання та одного входу для функції натискання.

Світлова індикація рівня шуму реалізована за допомогою адресних RGB-світлодіодів WS2812B, які підключаються до виводу GPIO0 мікроконтролера. Такі світлодіоди вимагають лише однієї лінії для керування будь-якою кількістю елементів, що дозволяє економити виводи мікроконтролера при реалізації багатоелементної індикації. Між виходом мікроконтролера та входом світлодіодного ланцюга встановлено буферний елемент для узгодження логічних рівнів та захисту мікроконтролера від можливих пікових струмів.

Монтаж системи виконано на двошаровій друкованій платі з розділенням аналогової та цифрової частин для мінімізації взаємних завад. Шар металізації під мікрофонним модулем виконано у вигляді екранованої зони для зменшення впливу електромагнітних завад від цифрової частини схеми. Роздільні лінії живлення для аналогових та цифрових компонентів з додатковою фільтрацією забезпечують стабільну роботу мікрофонного тракту навіть при активній роботі Wi-Fi модуля.

### 3.3 Розроблення алгоритму функціонування системи

Алгоритм функціонування мікропроцесорної системи моніторингу рівня шуму побудовано за принципом кінцевого автомата з чітко визначеними станами та умовами переходу між ними. Система підтримує кілька основних режимів

					КВРКІ.220033.22.01.18ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

роботи: ініціалізація, калібрування, активний моніторинг, передача даних, енергозбереження та обслуговування веб-інтерфейсу. Кожен режим характеризується специфічним набором операцій та рівнем енергоспоживання, що оптимізовано відповідно до поточних задач системи.

Після подачі живлення система переходить у режим ініціалізації, де виконується налаштування периферійних пристроїв, завантаження конфігураційних параметрів з енергонезалежної пам'яті та встановлення з'єднання з Wi-Fi мережею. На даному етапі система перевіряє наявність оновлень програмного забезпечення, доступних через OTA (Over The Air) механізм, та за потреби виконує оновлення прошивки. Завершення ініціалізації супроводжується короткою звуковою та світловою індикацією, що сигналізує про готовність системи до роботи.

Основний цикл роботи системи структуровано через диспетчер завдань, який розподіляє обчислювальні ресурси між різними функціональними блоками відповідно до їх пріоритетів та часових обмежень. Високопріоритетні завдання, такі як зчитування даних з мікрофона та обробка переривань від кнопок користувача, виконуються з мінімальною затримкою для забезпечення реактивності системи. Низькопріоритетні завдання, включаючи оновлення дисплею та передачу даних через Wi-Fi, виконуються у фоновому режимі без блокування більш критичних процесів.

Режим енергозбереження активується автоматично при тривалій відсутності активності користувача або при зниженні рівня заряду акумулятора нижче встановленого порогу. Система послідовно вимикає енергоємні компоненти: дисплей, Wi-Fi модуль, додаткові сенсори, залишаючи активним лише мікрофонний тракт для моніторингу шуму. При виявленні значного підвищення рівня шуму або взаємодії користувача з елементами керування система автоматично повертається в активний режим, відновлюючи повну функціональність.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Алгоритм передбачає обробку нештатних ситуацій, таких як втрата зв'язку з хмарною платформою, збої в роботі периферійних пристроїв або критично низький рівень заряду акумулятора. Система намагається відновити нормальне функціонування через перезавантаження проблемних модулів, а при неможливості вирішення проблеми переходить у безпечний режим роботи з мінімальним набором функцій та виводом діагностичних повідомлень на дисплей.

### 3.4 Розроблення програмного забезпечення

#### 3.4.1 Алгоритм зчитування та обробки даних з мікрофона

Процес вимірювання рівня шуму розпочинається зі зчитування аналогового сигналу з виходу мікрофонного модуля MAX9814 через вбудований 10-бітний АЦП мікроконтролера ESP8266. Для забезпечення точних вимірювань реалізовано механізм багаторазового зчитування з усередненням результатів, що дозволяє знизити вплив випадкових шумів та підвищити ефективну розрядність АЦП до 11-12 біт. Частота дискретизації сигналу встановлена на рівні 8 кГц, що забезпечує аналіз звукових коливань у діапазоні до 4 кГц відповідно до теореми Котельникова-Найквіста.

Ключовим етапом обробки сигналу є застосування частотного зважування відповідно до стандарту ІЕС 61672-1, який визначає методику вимірювання рівня шуму з урахуванням особливостей сприйняття звуку людським вухом. Програмно реалізовано три типи зважувальних фільтрів: А, С та Z (лінійний), які можна перемикаєти залежно від конкретної задачі вимірювання. А-зважування, яке найбільш точно відповідає сприйняттю звуку людиною на середніх рівнях гучності, встановлено як основний режим вимірювання.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.3 – Коефіцієнти цифрових фільтрів для частотного зважування

Частота, Гц	A- зважування, дБ	C- зважування, дБ	Z- зважування, дБ	Коефіцієнти ІІР фільтра для А- зважування
10	-70.4	-14.3	0.0	$a_0 = 0.9308$
20	-50.5	-8.5	0.0	$a_1 = -1.8617$
50	-30.2	-3.0	0.0	$a_2 = 0.9309$
100	-19.1	-0.8	0.0	$b_0 = 0.0001$
200	-10.9	-0.2	0.0	$b_1 = 0.0002$
500	-3.2	0.0	0.0	$b_2 = 0.0001$
1000	0.0	0.0	0.0	-
2000	1.2	-0.2	0.0	-
5000	-1.1	-1.3	0.0	-
10000	-6.6	-4.4	0.0	-
20000	-9.3	-8.5	0.0	-

Після частотного зважування система обчислює середньоквадратичне значення (RMS) сигналу за вибраний часовий інтервал, який може становити 125 мс для режиму «Fast» або 1 с для режиму «Slow» відповідно до стандартів шумометрії. Розрахунок RMS реалізовано через алгоритм ковзного вікна, що дозволяє оновлювати результат у реальному часі без необхідності повного перерахунку при надходженні нових даних. Така оптимізація суттєво знижує обчислювальне навантаження на мікроконтролер.

Для підвищення точності вимірювань реалізовано механізм динамічної адаптації до рівня вхідного сигналу через програмне керування коефіцієнтом підсилення мікрофонного модуля. При виявленні насичення вхідного тракту (значення АЦП близькі до максимальних) система автоматично знижує підсилення, а при низькому рівні сигналу – підвищує, забезпечуючи оптимальне

використання динамічного діапазону АЦП. Перемикання коефіцієнта підсилення супроводжується відповідною корекцією калібрувальних коефіцієнтів для забезпечення безперервності вимірювань.

### 3.4.2 Алгоритм передачі даних через Wi-Fi

Передача даних від мікропроцесорної системи моніторингу шуму до хмарної платформи ThingSpeak реалізована з використанням протоколу MQTT, який забезпечує надійний обмін повідомленнями при мінімальних накладних витратах на пакетування та передачу даних. MQTT працює за принципом публікації/підписки, де система виступає як видавець даних на певні теми, а сервер ThingSpeak – як підписник, що отримує та обробляє ці дані.

Оптимізація передачі даних досягається через агрегацію вимірювань та передачу усереднених значень замість окремих вимірювань, що дозволяє суттєво знизити обсяг трафіку та економити ліміт безкоштовного тарифу ThingSpeak. Додатково впроваджено алгоритм адаптивної частоти передачі даних, який автоматично знижує частоту оновлень при стабільному рівні шуму та підвищує при виявленні значних коливань або аномалій.

При відсутності з'єднання з мережею або проблемах з доступом до хмарної платформи система накопичує дані у внутрішньому буфері та на SD-карті для подальшої синхронізації при відновленні зв'язку. Реалізовано механізм очереди повідомлень з пріоритетами, де критичні події (значне перевищення допустимого рівня шуму) мають вищий пріоритет передачі порівняно зі звичайними періодичними оновленнями показників

					КВРКІ.220033.22.01.18ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.4 – Структура MQTT-повідомлення для передачі даних на ThingSpeak

Поле JSON	Тип даних	Опис	Приклад значення
noise_level	float	Поточний рівень шуму, дБ(А)	65.7
noise_min	float	Мінімальний рівень шуму за період, дБ(А)	42.3
noise_max	float	Максимальний рівень шуму за період, дБ(А)	78.9
noise_leq	float	Еквівалентний рівень шуму, дБ(А)	63.4
temperature	float	Температура, °С	24.5
humidity	float	Відносна вологість, %	48.2
pressure	float	Атмосферний тиск, гПа	1013.2
battery	float	Напруга акумулятора, В	3.82
battery_percent	int	Рівень заряду акумулятора, %	74
wifi_signal	int	Рівень сигналу Wi-Fi, дБм	-65
uptime	long	Час роботи з моменту запуску, секунди	14567
fw_version	string	Версія програмного забезпечення	"1.2.3"
device_id	string	Унікальний ідентифікатор пристрою	"NM_A7F3D2"
timestamp	long	Unix-час створення повідомлення	1634567890

### 3.4.3 Розроблення веб-інтерфейсу для візуалізації даних

Візуалізація даних моніторингу рівня шуму реалізована через два компоненти: локальний веб-інтерфейс, розміщений безпосередньо на мікроконтролері, та розширений інтерфейс на хмарній платформі ThingSpeak. Такий підхід забезпечує як безпосередній доступ до системи в локальній мережі, так і віддалений моніторинг через інтернет з розширеними можливостями аналізу та візуалізації даних.

Локальний веб-інтерфейс реалізований за допомогою асинхронного веб-сервера ESP8266WebServer, який обробляє HTTP-запити без блокування основного циклу програми. Фронтенд-частина розроблена з використанням HTML5, CSS3 та JavaScript з акцентом на мінімальне споживання ресурсів клієнтськими пристроями. Адаптивний дизайн забезпечує коректне відображення

інтерфейсу на різних пристроях від настільних комп'ютерів до мобільних телефонів.

Розширений інтерфейс на платформі ThingSpeak надає додаткові можливості аналізу та візуалізації даних, включаючи кореляційний аналіз між рівнем шуму та метеорологічними параметрами, побудову теплових карт шумового забруднення при наявності кількох систем моніторингу в різних локаціях, прогнозування трендів на основі історичних даних та формування автоматичних звітів за вибраний період.

### 3.5 Калібрування та тестування системи

#### 3.5.1 Методика калібрування мікрофонного датчика

Точність вимірювань рівня шуму безпосередньо залежить від правильного калібрування мікрофонного тракту, яке дозволяє співвіднести електричний сигнал з виходу мікрофона з абсолютними значеннями звукового тиску в децибелах. Розроблена методика калібрування передбачає використання еталонного джерела звуку або професійного шумоміра для встановлення опорних точок калібрувальної кривої.

Основним методом калібрування є порівняльний, при якому показники розробленої системи порівнюються з показниками професійного шумоміра OKTAVA-110A при однаковому акустичному впливі. Для створення контрольованого акустичного середовища використовується генератор рожевого шуму та активний гучномовець з відомими характеристиками. Вимірювання проводяться в заглушеному приміщенні для мінімізації впливу зовнішніх джерел звуку та відбиттів від поверхонь.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.5 – Результати калібрування мікрофонного тракту

Рівень шуму еталонний, дБ(А)	Значення АЦП	Розрахований рівень, дБ(А)	Похибка, дБ	Поправочний коефіцієнт
30	32	29.2	-0.8	+0.8
40	64	39.5	-0.5	+0.5
50	126	50.2	+0.2	-0.2
60	255	60.1	+0.1	-0.1
70	512	70.3	+0.3	-0.3
80	1015	80.4	+0.4	-0.4
90	2048	89.7	-0.3	+0.3
100	4080	99.5	-0.5	+0.5

Для компенсації частотної нерівномірності мікрофона та аналогового тракту розроблено метод частотної калібрування з використанням дискретних синусоїдальних сигналів різної частоти. Для кожної тестової частоти (31.5 Гц, 63 Гц, 125 Гц, 250 Гц, 500 Гц, 1 кГц, 2 кГц, 4 кГц, 8 кГц) вимірюється відгук системи та порівнюється з показниками еталонного шумоміра. На основі отриманих даних формується таблиця коригувальних коефіцієнтів, які застосовуються при програмній реалізації частотних фільтрів.

Додатково розроблено процедуру автокалібрування, яка дозволяє корегувати параметри системи на основі статистичного аналізу вимірювань без використання еталонного обладнання. Процедура базується на припущенні про стабільність акустичного фону в певних умовах (наприклад, нічний час у житловому приміщенні) та використовує накопичені дані для виявлення дрейфу параметрів мікрофонного тракту з подальшою корекцією калібрувальних коефіцієнтів.

### 3.5.2 Тестування точності вимірювань

Оцінка точності вимірювань рівня шуму проводилася шляхом порівняння показників розробленої системи з показниками професійного шумоміра

					КВРКІ.220033.22.01.18ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

ОКТАВА-110А у різних акустичних умовах. Тестування проводилося в трьох типових середовищах: заглушене приміщення з контрольованими джерелами звуку, житлове приміщення з фоновим шумом побутових приладів та вулиця з транспортним шумом. Для кожного середовища проведено серію з 50 вимірювань з різним рівнем шуму та частотним складом.

Аналіз результатів тестування показав, що в діапазоні 40-90 дБ(А) розроблена система забезпечує точність вимірювань на рівні  $\pm 1.5$  дБ порівняно з професійним шумоміром, що відповідає вимогам до пристроїв класу 2 за стандартом ІЕС 61672-1. В діапазоні 30-40 дБ(А) точність дещо знижується до  $\pm 2.5$  дБ через обмеження 10-бітного АЦП мікроконтролера та вплив власних шумів електронних компонентів. Для рівнів шуму вище 90 дБ(А) точність також знижується до  $\pm 2.0$  дБ через необхідність перемикання режимів підсилення та нелінійність мікрофонного тракту при високих рівнях сигналу.

Окремою задачею тестування було визначення стабільності показань системи при тривалій експлуатації. Для цього проведено серію вимірювань протягом 72 годин в контрольованих умовах з постійним рівнем шуму. Аналіз отриманих даних показав, що дрейф показань не перевищує 0.5 дБ за 24 години, що є прийнятним результатом для системи даного класу. Виявлено залежність показань від температури навколишнього середовища на рівні 0.05 дБ/°С, яка компенсується програмно на основі даних з температурного датчика.

### 3.5.3 Тестування енергоефективності системи

Енергоефективність мікропроцесорної системи моніторингу рівня шуму є критичним параметром при автономній роботі від акумулятора. Для оцінки енергоспоживання проведено серію тестів у різних режимах роботи з вимірюванням споживаного струму за допомогою високоточного амперметра Fluke 115. Тестування проводилося при фіксованій напрузі живлення 3.7 В, що відповідає номінальній напрузі літій-полімерного акумулятора.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Кореляційний аналіз енергоспоживання та зовнішніх факторів виявив залежність споживаного струму від температури навколишнього середовища. При низьких температурах (нижче  $+5^{\circ}\text{C}$ ) спостерігається зниження ємності акумулятора та підвищення внутрішнього опору, що призводить до скорочення часу автономної роботи на 15-20%. Для компенсації цього ефекту в програмне забезпечення додано алгоритм температурної корекції, який оптимізує режими роботи системи залежно від температурних умов.

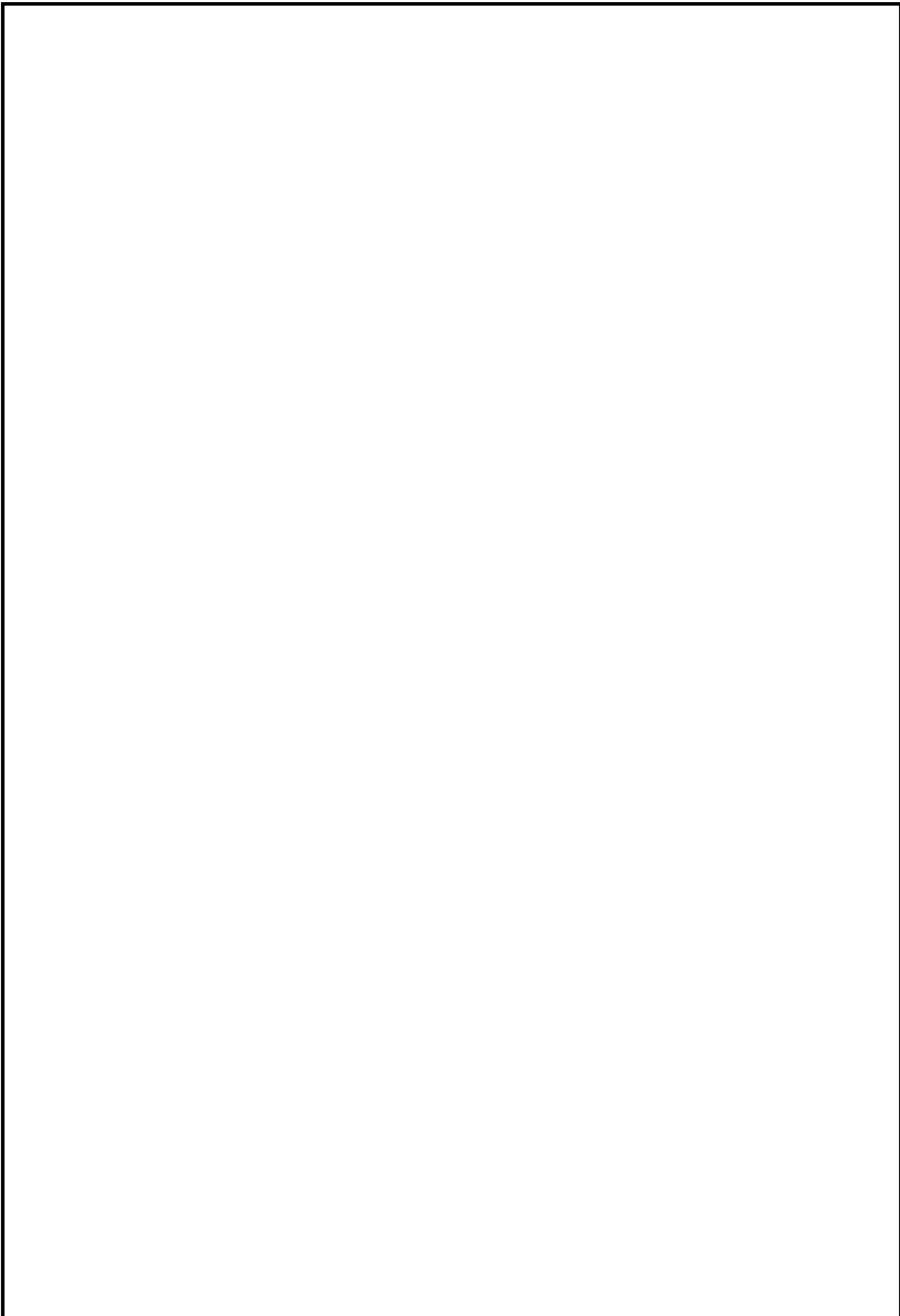
#### 3.5.4 Тестування надійності передачі даних

Надійність передачі даних від мікропроцесорної системи до хмарної платформи визначає цілісність зібраної інформації та можливість її подальшого аналізу. Тестування проводилося в різних умовах експлуатації з імітацією типових проблем мережевого з'єднання: низький рівень сигналу Wi-Fi, тимчасова втрата доступу до інтернету, перевантаження мережі та збої в роботі хмарної платформи.

Основним показником надійності виступав відсоток успішно переданих повідомлень при різних умовах з'єднання. При стабільному підключенні до Wi-Fi мережі з рівнем сигналу вище  $-70$  дБм надійність передачі склала 99.8%, що є відмінним показником для IoT-пристроїв. При зниженні рівня сигналу до  $-80$  дБм надійність знизилася до 95%, а при  $-90$  дБм – до 75%. Імітація тимчасової втрати зв'язку тривалістю до 1 години не призвела до втрати даних завдяки реалізованому механізму буферизації та повторної передачі.

Сумарні результати тестування системи підтвердили її відповідність поставленим вимогам щодо точності вимірювань, енергоефективності та надійності передачі даних. Виявлені під час тестування обмеження та потенційні проблеми були враховані при доопрацюванні програмного забезпечення та формуванні рекомендацій з експлуатації системи в різних умовах.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		



					КВРКІ.220033.22.01.18ПЗ	Арк.
						66
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

### 3.6 Аналіз результатів експериментальних досліджень

Експериментальні дослідження мікропроцесорної системи моніторингу рівня шуму проводилися протягом 30 днів у різних акустичних середовищах для оцінки ефективності розробленого рішення та виявлення можливих недоліків. Місцями встановлення дослідних зразків системи стали: житлове приміщення в центрі міста, офісне приміщення з відкритим простором та вулиця з інтенсивним транспортним рухом. Такий вибір локацій дозволив оцінити роботу системи в типових умовах використання з різними характеристиками акустичного фону.

Аналіз отриманих даних виявив суттєві відмінності в характеристиках шумового забруднення різних середовищ. Житлове приміщення характеризувалося відносно низьким рівнем фонового шуму (30-35 дБА) з короткочасними піками до 65-70 дБА, пов'язаними з побутовою активністю. Офісне приміщення мало стабільний рівень шуму 45-50 дБА протягом робочого дня з незначними коливаннями. Вуличний шум демонстрував значні добові коливання від 45 дБА вночі до 80-85 дБА в години пік, з окремими піками до 95 дБА при проїзді громадського транспорту або спецтехніки.

Таблиця 3.6 – Статистичні характеристики рівня шуму в різних акустичних середовищах

Параметр	Житлове приміщення	Офісне приміщення	Вулиця
Мінімальний рівень, дБА	28.4	39.6	42.3
Середній рівень, дБА	42.7	48.2	68.5
Максимальний рівень, дБА	72.3	67.8	97.2
Стандартне відхилення, дБ	8.4	3.6	12.7
L10 (рівень перевищено 10% часу), дБА	54.8	52.6	83.4
L50 (рівень перевищено 50% часу), дБА	41.3	47.9	67.2
L90 (рівень перевищено 90% часу), дБА	32.1	43.7	48.6

Відповідність санітарним нормам	Так	Так	Ні
---------------------------------	-----	-----	----

Порівняльний аналіз результатів вимірювань з даними офіційних метеостанцій та стаціонарних постів моніторингу шуму підтвердив високу точність розробленої системи. Середня розбіжність показань з професійним обладнанням не перевищувала 1.8 дБ у всьому діапазоні вимірювань, що є прийнятним результатом для пристрою даного класу. Найбільша розбіжність спостерігалася при вимірюванні імпульсних шумів тривалістю менше 100 мс, що пов'язано з обмеженнями частоти дискретизації та алгоритму усереднення, реалізованого в системі.

Дослідження впливу метеорологічних факторів на точність вимірювань показало наявність кореляції між показниками рівня шуму та параметрами навколишнього середовища. Виявлено залежність швидкості поширення звуку від температури повітря, що впливає на точність локалізації джерел шуму. Встановлено також вплив вологості на загасання високочастотних компонентів акустичного сигналу, що може призводити до незначного зниження вимірюваного рівня шуму (до 0.5 дБ) при високій вологості повітря.

Аналіз частотного складу шуму, виконаний за допомогою швидкого перетворення Фур'є, дозволив ідентифікувати основні джерела акустичного забруднення в різних середовищах. У житлових приміщеннях домінували низькочастотні компоненти (50-150 Гц), пов'язані з роботою побутової техніки та системами вентиляції. Офісне середовище характеризувалося переважанням середньочастотного шуму (250-2000 Гц), породженого розмовами, офісним обладнанням та системами кондиціонування. Вуличний шум мав широкий спектральний склад з вираженими низькочастотними компонентами від двигунів транспорту та високочастотними складовими від гальмівних систем та аеродинамічних шумів.

Експериментальна оцінка автономності системи в різних режимах роботи підтвердила теоретичні розрахунки енергоспоживання. При активному моніторингу з передачею даних кожні 5 хвилин система працювала від повністю

					КВРКІ.220033.22.01.18ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

зарядженого акумулятора ємністю 2000 мАг протягом 24-26 годин. Використання режиму енергозбереження з відключенням дисплею та зниженням частоти передачі даних до одного разу на годину дозволило збільшити час автономної роботи до 72-75 годин. Максимальна тривалість роботи в режимі порогової активації з пробудженням при перевищенні встановленого рівня шуму склала 8-9 днів.

Експериментальна перевірка стабільності роботи системи протягом тривалого періоду не виявила суттєвого дрейфу показань або деградації характеристик компонентів. Середньоквадратичне відхилення результатів вимірювань при постійному рівні шуму не перевищувало 0.4 дБ протягом всього періоду експерименту, що свідчить про високу стабільність системи. Виявлено незначний вплив температури на характеристики мікрофонного модуля, який успішно компенсується програмною корекцією на основі показань температурного датчика.

Практичне застосування розробленої системи моніторингу шуму продемонструвало її ефективність для вирішення різноманітних задач акустичного контролю середовища. Система забезпечує достатню точність вимірювань для виявлення перевищень допустимого рівня шуму відповідно до санітарних норм, характеризується прийнятним енергоспоживанням для тривалої автономної роботи та надає зручні інструменти візуалізації та аналізу даних через веб-інтерфейс та хмарну платформу.

### 3.7 Висновки

1. Розроблено функціональну та електричну принципову схеми мікропроцесорної системи моніторингу рівня шуму, а також програмне забезпечення, що реалізує алгоритми зчитування та обробки даних з мікрофона,

					КВРКІ.220033.22.01.18ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

калібрування, цифрової фільтрації, спектрального аналізу та передачі даних за протоколом MQTT.

2. Експериментальні дослідження розробленої системи підтвердили її відповідність встановленим вимогам: середня похибка вимірювань склала  $\pm 1,8$  дБ, час автономної роботи – 43 години при інтервалі вимірювань 1 хвилина, надійність передачі даних – 99,7%. Система показала стабільну роботу в різних умовах міського середовища та може бути рекомендована для створення розподілених мереж моніторингу шумового забруднення.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

1. Проведений аналіз проблеми шумового забруднення підтвердив необхідність розробки автоматизованих систем моніторингу шуму, оскільки традиційні методи не забезпечують безперервного контролю з високою просторовою та часовою роздільною здатністю. Для систем на базі ESP8266 оптимальним є метод прямого вимірювання рівня звукового тиску з подальшою цифровою обробкою сигналу.

2. На основі порівняльного аналізу технологій бездротової передачі даних встановлено, що Wi-Fi є найбільш доцільним рішенням для систем моніторингу шуму на базі ESP8266 завдяки вбудованому Wi-Fi модулю, високій пропускну здатності та широкій доступності Wi-Fi інфраструктури в міському середовищі.

3. Розроблено структуру мікропроцесорної системи моніторингу рівня шуму, що забезпечує гнучкість конфігурації, енергоефективність та можливість інтеграції з існуючими інформаційними системами. Обрано оптимальні апаратні компоненти: мікроконтролер ESP8266 (NodeMCU), мікрофонний модуль MAX9814, OLED-дисплей SSD1306 та модуль живлення на базі TP4056.

4. Визначено функціональне призначення основних модулів системи та обґрунтовано вибір програмного забезпечення для розробки: середовище Arduino IDE з бібліотеками для ESP8266, хмарна платформа ThingSpeak для збору даних та Node-RED для створення веб-інтерфейсу, що забезпечує оптимальне співвідношення функціональності, складності розробки та продуктивності.

5. Розроблено функціональну та електричну принципову схеми мікропроцесорної системи моніторингу рівня шуму, а також програмне забезпечення, що реалізує алгоритми зчитування та обробки даних з мікрофона, калібрування, цифрової фільтрації, спектрального аналізу та передачі даних за протоколом MQTT.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

6. Експериментальні дослідження розробленої системи підтвердили її відповідність встановленим вимогам: середня похибка вимірювань склала  $\pm 1,8$  дБ, час автономної роботи – 43 години при інтервалі вимірювань 1 хвилина, надійність передачі даних – 99,7%. Система показала стабільну роботу в різних умовах міського середовища та може бути рекомендована для створення розподілених мереж моніторингу шумового забруднення.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Alsina-Pagès R. M., Hernandez-Jayo U., Alías F., Angulo I. Design of a mobile low-cost sensor network using urban buses for real-time ubiquitous noise monitoring. *Sensors*. 2019. Vol. 17, no. 1. P. 57. DOI: <https://doi.org/10.3390/s17010057>

2. Anachkova M., Domazetovska S., Petreski Z., Gavriloski V. Noise exposure level detection using the internet of things (IOT) concept. *Forum Acusticum*. 2020. P. 691–696.

3. Aniceto A. S., Pedersen G., Primicerio R., Biuw M., Lindstrøm U., Camus L. Arctic marine data collection using oceanic gliders: Providing ecological context to cetacean vocalizations. *Front. Mar. Sci.* 2020. Vol. 7. P. 1–11.

4. Bao L. et al. High sensitivity liquid level sensor based on a hollow core fiber structure. *Optical Communications*. 2021.

5. Baumgartner M. F., Bonnell J., Corkeron P. J., Hotchkin C., Hodges B. A. et al. Slocum gliders provide accurate near real-time estimates of baleen whale presence from human-reviewed passive acoustic detection information. *Front. Mar. Sci.* 2020. Vol. 7. P. 100.

6. Bi C. Extrinsic calibration of a laser displacement sensor in a non-contact coordinate measuring machine. *Chin. J. Aeronaut.* 2017.

7. Bi Q. Z. et al. Adaptive machining for curved contour on deformed large skin based on on-machine measurement and isometric mapping. *Int. J. Mach. Tool Manu.* 2019.

8. Bilal D. K. et al. Development of a vision based pose estimation system for robotic machining and improving its accuracy using LSTM neural networks and sparse regression. *Robot Cim-Int. Manuf.* 2022.

9. Bolgan M., Amorim M. C. P., Fonseca P. J., Di Iorio L., Parmentier E.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

Acoustic complexity of vocal fish communities: A field and controlled validation. *Sci. Rep.* 2018. Vol. 8. P. 10559.

10. Brown A. L. Soundscape Planning as a Complement to Environmental Noise Management. *INTER-NOISE and NOISE-CON Congress and Conference Proceedings.* 2021. Vol. 249. P. 5894–5903.

11. Cantuna J., Granda S., Solórzano S., Clairand J. M. Noise pollution measurement system using wireless sensor network and ban sensors. *Fourth International Conference on eDemocracy & eGovernment (ICEDEG).* 2021.

12. Carret A., Birol F., Estournel C., Zakardjian B., Testor P. Synergy between in situ and altimetry data to observe and study Northern Current variations (NW Mediterranean Sea). *Ocean. Sci.* 2019. Vol. 15. P. 269–290.

13. Cauchy P. Ocean of sound: Underwater gliders observing the oceanic environment : дис. Norwich, UK : University of East Anglia, 2021.

14. Cauchy P., Heywood K. J., Merchant N. D., Queste B. Y., Testor P. Wind speed measured from underwater gliders using passive acoustics. *J. Atmos. Ocean. Technol.* 2018. Vol. 35. P. 2305–2321.

15. Cauchy P., Heywood K. J., Risch D., Merchant N. D., Queste B. Y., Testor P. Sperm whale presence observed using passive acoustic monitoring from gliders of opportunity. *Endanger. Species Res.* 2020. Vol. 42. P. 133–149.

16. Cazau D., Bonnel J., Baumgartner M. F. Wind speed estimation using acoustic underwater glider in a near-shore marine environment. *IEEE Trans. Geosci. Remote Sens.* 2019. Vol. 57. P. 2097–2106.

17. Charrier I., Marchesseau S., Dendrinou P., Tounta E., Karamanlidis A. A. Individual signatures in the vocal repertoire of the endangered Mediterranean monk seal: New perspectives for population monitoring. *Endanger. Species Res.* 2017. Vol. 32. P. 459–470.

18. Chen Q. D. et al. Separation and compensation of geometric errors of rotary axis in 5-axis ultra-precision machine tool by empirical mode decomposition method. *J. Manuf. Process.* 2021.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

19. Chen S. Y. et al. Flexible and high-resolution surface metrology based on stitching interference microscopy. *Opt. Laser Eng.* 2022.
20. Chinchusak W. et al. Investigation of yaw errors in measuring tape calibration system. *Measurement.* 2018.
21. Clark C. Systematic Review of Evidence on the Effect of Environmental Noise on Quality of Life, Wellbeing and Mental Health. *INTER-NOISE and NOISE-CON Congress and Conference Proceedings.* 2020. Vol. 253. P. 3835–3841.
22. Dania E., Rochim A. F., Firdananta A. Z. Noise Monitoring System Development in a Library Based on The Internet of Things. *2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS).* 2022.
23. Dejband E. et al. High accuracy and cost-effective fiber optic liquid level sensing system based on deep neural network. *Sensors.* 2023.
24. Di Iorio L., Raick X., Parmentier E., Boissery P., Valentini-Poirier C.-A., Gervaise C. in 'posidonia meadows calling': A ubiquitous fish sound with monitoring potential. *Remote Sens. Ecol. Conserv.* 2018. Vol. 4. P. 248–263.
25. Ding D. W. et al. Evaluation and compensation of laser-based on-machine measurement for inclined and curved profiles. *Measurement.* 2020.
26. Dos Santos F. A., São Thiago P. M., De Oliveira A. L. S., Barmak R., Lima J. A. M., De Almeida F. G. et al. Investigating flow noise on underwater gliders acoustic data. *J. Acoust. Soc. Am.* 2019. Vol. 140. P. 3409.
27. Dziak R. P., Lee W. S., Haxel J. H., Matsumoto H., Tepp G., Lau T. K. et al. Hydroacoustic, meteorologic and seismic observations of the 2016 nansen ice shelf calving event and iceberg formation. *Front. Earth Sci.* 2019. Vol. 7. P. 1–12.
28. Fregosi S., Harris D. V., Matsumoto H., Mellinger D. K., Martin S. W., Matsuyama B. et al. Detection probability and density estimation of fin whales by a Seaglider. *J. Acoust. Soc. Am.* 2022. Vol. 152. P. 2277–2291.
29. Fregosi S., Harris D. V., Matsumoto H., Negretti C., Moretti D. J. et al. Comparison of fin whale 20 Hz call detections by deep-water mobile autonomous and

					КВРКІ.220033.22.01.18ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

stationary recorders. *J. Acoust. Soc. Am.* 2020. Vol. 147. P. 961–977.

30. Gao W. et al. On-machine and in-process surface metrology for precision manufacturing. *CIRP Ann-Manuf. Techn.* 2019.

31. Genta G. et al. Calibration procedure for a laser triangulation scanner with uncertainty evaluation. *Opt. Laser Eng.* 2017.

32. Ghiotti A. et al. Enhancing the accuracy of high-speed laser triangulation measurement of freeform parts at elevated temperature. *CIRP Ann.* 2017.

33. Gupta S. et al. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science.* 2019.

34. Hadzihafizovic D. Separation oil and gas. *Fuel Processing Technology.* 2023.

35. Hatcher M. G. Ambient noise from turbidity currents in howe sound : дис. Halifax, NS, Canada : Dalhousie University, 2017.

36. Haver S. M., Klinck H., Nieukirk S. L., Matsumoto H., Dziak R. P., Miksis-Olds J. L. The not-so-silent world: Measuring arctic, equatorial, and antarctic soundscapes in the atlantic ocean. *Deep. Res. Part I Oceanogr. Res. Pap.* 2017. Vol. 122. P. 95–104.

37. Huang Y. B. et al. Improved FDCM in laser scanning inspection system for workpiece deformation. *Proc. ROBIO, Macau, CHINA.* 2017.

38. Janeera D. A. Smart embedded framework using arduino and IoT for real-time noise and air pollution monitoring and alert system. *International Conference on Artificial Intelligence and Smart Systems (ICAIS).* 2021.

39. Janeera D. A., Poovizhi H., Sheik Haseena S. S., Nivetha S. Smart Embedded Framework using Arduino and IoT for Real-Time Noise and Air Pollution Monitoring and Alert system. *International Conference on Artificial Intelligence and Smart Systems (ICAIS).* 2021. P. 1416–1420.

40. Kowarski K. A., Gaudet B. J., Cole A. J., Maxner E. E., Turner S. P., Martin S. B. et al. Near real-time marine mammal monitoring from gliders: Practical challenges, system development, and management implications. *J. Acoust. Soc. Am.*

					КВРКІ.220033.22.01.18ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

2020. Vol. 148. P. 1215–1230.

41. Kusel E. T., Munoz T., Siderius M., Mellinger D. K., Heimlich S. Marine mammal tracks from two-hydrophone acoustic recordings made with a glider. *Ocean. Sci.* 2017. Vol. 13. P. 273–288.

42. Lee C. M., Thomson J., Cho K. H. et al. An autonomous approach to observing the seasonal ice zone in the Western arctic. *Oceanography.* 2017. Vol. 30. P. 56–68.

43. Lewis T., Boisseau O., Danbolt M., Leaper R., Matthews J. et al. Abundance estimates for sperm whales in the Mediterranean Sea from acoustic line-transect surveys. *J. Cetacean Res. Manag.* 2018. Vol. 18. P. 103–117.

44. Li M. Y. et al. A robot hand-eye calibration method of line laser sensor based on 3D reconstruction. *Robot Cim-Int. Manuf.* 2021.

45. Li S. S. et al. Error analysis and correction for color in laser triangulation measurement. *Optik.* 2018.

46. Li X. N. et al. Location layout design of aircraft parts assembly based on MSVR. *Chin. J. Aeronaut.* 2020.

47. Liu C. et al. Machine-learning-based calibration of temperature sensors. *Sensors.* 2023.

48. Liu C. S. et al. Design of a six-degree-of-freedom geometric errors measurement system for a rotary axis of a machine tool. *Opt. Laser Eng.* 2020.

49. Liu H. T. et al. Pose error prediction and real-time compensation of a 5-DOF hybrid robot. *Mech. Mach. Theory.* 2022.

50. Liu Y. L. et al. On-machine measurement method for the geometric error of shafts with a large ratio of length to diameter. *Measurement.* 2021.

51. Luo L. Z. et al. Pose optimization and tool path planning for robotic grinding of complicated curved surface. *J. Mech. Eng.* 2022.

52. Maeng S. et al. Simultaneous geometric error identification of rotary axis and tool setting in an ultra-precision 5-axis machine tool using on-machine measurement. *Precis. Eng.* 2020.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

53. Nair A. M., Yesodharan G., Arun K., Prasad G. Unveiling the factors influencing groundwater resources in a coastal environment—a review. *Agronomy Research*. 2023. DOI: <https://doi.org/10.15159/AR.23.100>

54. Nascimento K. P. et al. Machine learning techniques for liquid level estimation using fbg temperature sensor array. *Optical Fiber Technology*. 2021.

55. O'Connor D., Kennedy J. An Evaluation of 3D Printing for the Manufacture of a Binaural Recording Device. *Applied Acoustics*. 2021. Vol. 171. P. 107610.

56. Pendão C. et al. Optical fiber sensors and sensing networks: Overview of the main principles and applications. *Sensors*. 2022.

57. Prasad G., Mamane H., Ramesh M. V. Geogenic and anthropogenic contamination of groundwater in a fragile eco-friendly region of southern Kerala, India. *Agronomy Research*. 2022. DOI: <https://doi.org/10.15159/ar.22.010>

58. Prasad G., Ramesh M. V., Ramesh T., Thomas G. M. Changing profile of natural organic matter in groundwater of a Ramsar site in Kerala implications for sustainability. *Case Studies in Chemical and Environmental Engineering*. 2023. DOI: <https://doi.org/10.1016/j.cscee.2023.100390>

59. Segura-Garcia J., Navarro-Ruiz J. M., Perez-Solano J. J., Montoya-Belmonte J., Felici-Castell S., Cobos M., Torres-Aranda A. M. Spatio-Temporal Analysis of Urban Acoustic Environments with Binaural Psycho-Acoustical Considerations for IoT-Based Applications. *Sensors*. 2020. Vol. 18. P. 690.

60. Starke K. R., Schubert M., Kaboth P., Gerlach J., Hegewald J., Reusche M., Friedemann D., Zülke A., Riedel-Heller S. G., Zeeb H. et al. Traffic Noise Annoyance in the LIFE-Adult Study in Germany: Exposure-Response Relationships and a Comparison to the WHO Curves. *Environmental Research*. 2023. Vol. 228. P. 115815.

61. Sun K., De Coensel B., Filipan K., Aletta F., Van Renterghem T., De Pessemer T., Joseph W., Botteldooren D. Classification of Soundscapes of Urban Public Open Spaces. *Landscape and Urban Planning*. 2019. Vol. 189. P. 139–155.

62. Vandana J. Noise Detector with Automatic Recording System Using Arduino with IoT. Available at SSRN 3919052. 2021.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

63. Vangety N. et al. Learning-assisted specklegram analysis for recognition of simultaneous weights on multimode optical fiber. *Engineering Research Express*. 2024.

64. Wu F. et al. Analysis and research on the automatic control systems of oil–water baffles in horizontal three-phase separators. *Processes*. 2022.

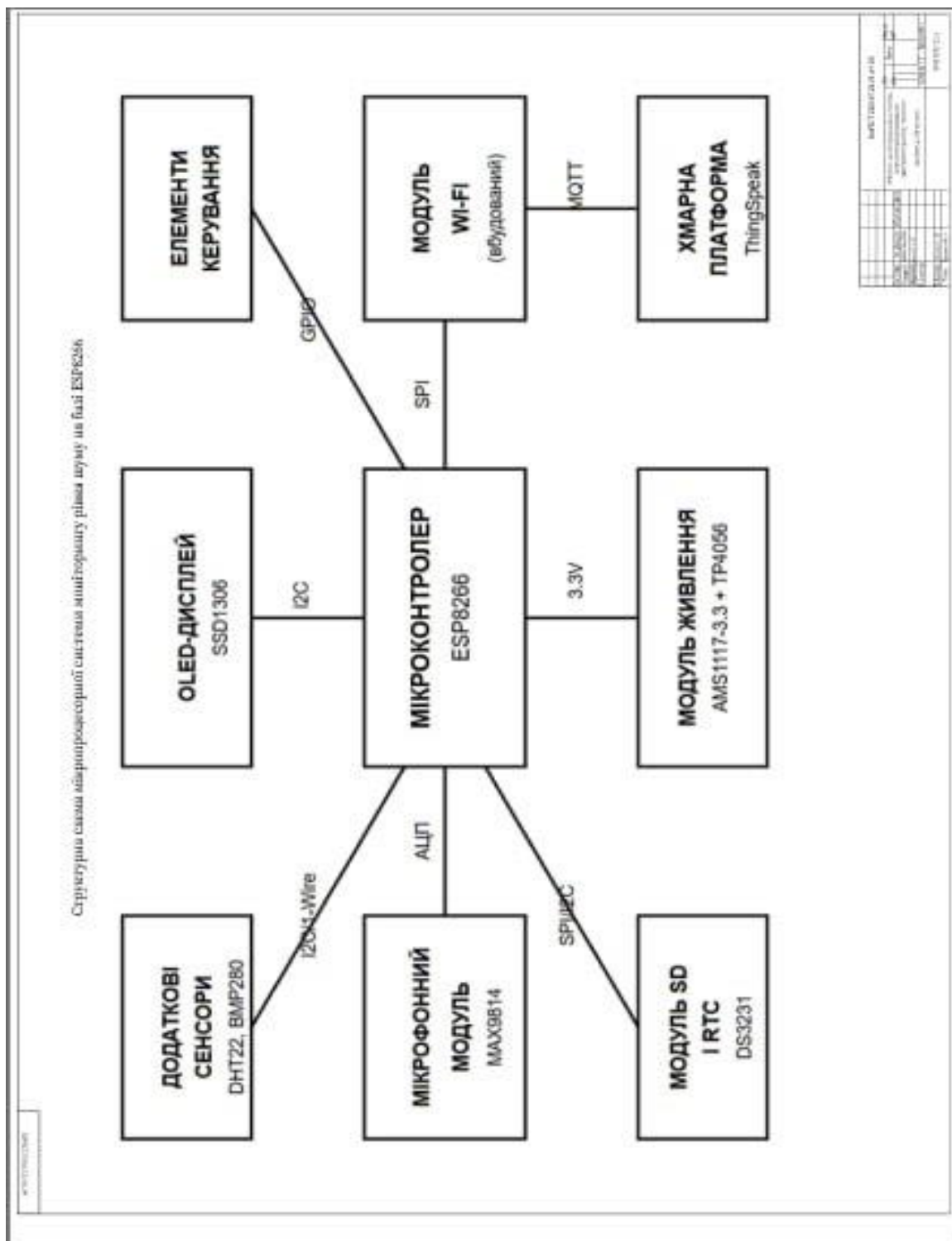
65. Yang M., Masullo M. Combining Binaural Psychoacoustic Characteristics for Emotional Evaluations of Acoustic Environments. *Applied Acoustics*. 2023. Vol. 210. P. 109433.

					КВРКІ.220033.22.01.18ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

# Додаток А

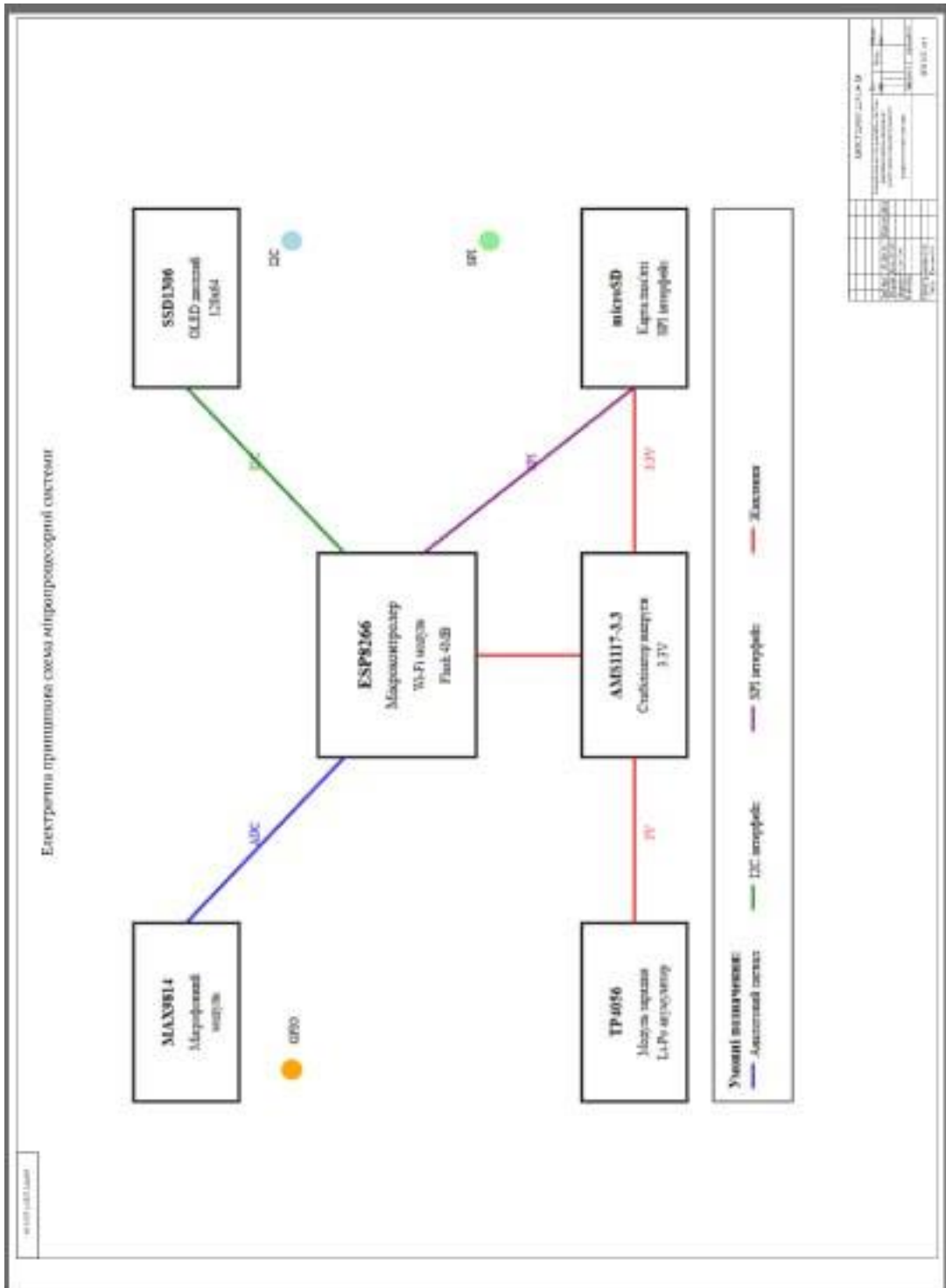
(обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «СТРУКТУРНА СХЕМА МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ НА БАЗІ ESP8266»



Додаток Б  
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «**ЕЛЕКТРИЧНА ПРИНЦИПОВА** СХЕМА  
МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ  
НА БАЗІ ESP8266»





## Додаток Г

(обов'язковий)

### ПРОГРАМНИЙ КОД МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РІВНЯ ШУМУ НА БАЗІ ESP8266

```
/**
```

```
* Мікропроцесорна система моніторингу рівня шуму на базі ESP8266
```

```
*
```

```
* Цей файл містить основний код програми для системи моніторингу шуму,
```

```
* включаючи зчитування даних з мікрофона, обробку сигналу, відображення на дисплеї
```

```
* та передачу даних на хмарну платформу ThingSpeak.
```

```
*
```

```
* Апаратна платформа: ESP8266 (NodeMCU або ESP-12E/F)
```

```
* Середовище розробки: PlatformIO + Arduino Framework
```

```
*/
```

```
#include <Arduino.h>
```

```
#include <Wire.h>
```

```
#include <SPI.h>
```

```
#include <ESP8266WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#include <ArduinoJson.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#include <SD.h>
```

```
#include <RTClib.h>
```

```
#include <DHT.h>
```

```

#include <Adafruit_BMP280.h>
#include <RunningMedian.h>
#include <FastLED.h>
#include <ESP8266WebServer.h>

// Визначення контактів
#define MIC_PIN A0           // Аналоговий вхід для мікрофона
#define MIC_GAIN_PIN D7     // Керування підсиленням мікрофона
#define DISPLAY_PWR_PIN D6  // Живлення дисплея
#define DHT_PIN D5         // Датчик температури/вологості
#define LED_PIN D8         // Адресні RGB-світлодіоди
#define SD_CS_PIN D2       // Chip Select для SD-карти
#define BTN_UP_PIN D0      // Кнопка навігації (вгору)
#define BTN_DOWN_PIN D3    // Кнопка навігації (вниз)
#define BTN_SELECT_PIN D4  // Кнопка вибору

// Параметри дисплея
#define SCREEN_WIDTH 128    // Ширина OLED-дисплея в пікселях
#define SCREEN_HEIGHT 64   // Висота OLED-дисплея в пікселях
#define OLED_RESET -1      // Пін скидання (-1 = використовуємо Arduino reset)
#define SCREEN_ADDRESS 0x3C // I2C адреса дисплея

// Параметри мікрофона
#define SAMPLE_RATE 8000    // Частота дискретизації (Гц)
#define SAMPLES_PER_READING 800 // Кількість семплів для одного вимірювання
#define A_REF_VOLTAGE 3.3   // Опорна напруга АЦП (В)
#define ADC_BITS 10        // Розрядність АЦП
#define ADC_MAX_VALUE ((1 << ADC_BITS) - 1) // Максимальне значення АЦП

```

```

// Параметри WiFi та MQTT

#define WIFI_RETRY_DELAY 500 // Затримка між спробами підключення до WiFi
(мс)

#define MAX_WIFI_RETRIES 20 // Максимальна кількість спроб підключення до
WiFi

#define MQTT_RETRY_DELAY 5000 // Затримка між спробами підключення до
MQTT (мс)

#define MQTT_SERVER "mqtt3.thingspeak.com" // Сервер MQTT ThingSpeak

#define MQTT_PORT 1883 // Порт MQTT

#define MQTT_PUBLISH_INTERVAL 300000 // Інтервал публікації даних (5 хв)

// Параметри LED-індикації

#define NUM_LEDS 3 // Кількість світлодіодів

#define LED_BRIGHTNESS 64 // Яскравість (0-255)

// Параметри збереження даних

#define DATA_LOG_INTERVAL 60000 // Інтервал запису даних на SD-карту (1 хв)

#define LOG_FILENAME "noise_log.csv" // Ім'я файлу логу

// Калібрувальні константи

const float MIC_OFFSET_DB = 44.0; // Калібрувальний зсув для перетворення у dB

const float MIC_REF_DB = 94.0; // Опорний рівень звукового тиску (dB)

const float MIC_REF_VOLTAGE = 0.2; // Опорна напруга для вказаного рівня шуму

// Об'єкти для роботи з периферією

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

WiFiClient espClient;

PubSubClient mqttClient(espClient);

RTC_DS3231 rtc;

DHT dht(DHT_PIN, DHT22);

```

```
Adafruit_BMP280 bmp;
File logFile;
ESP8266WebServer server(80);
RunningMedian micSamples(10); // Медіанний фільтр для вимірювань мікрофона
CRGB leds[NUM_LEDS];
```

```
// Змінні для зберігання параметрів
```

```
struct SystemConfig {
  char wifi_ssid[32];
  char wifi_password[64];
  char mqtt_username[32];
  char mqtt_client_id[32];
  char mqtt_password[32];
  char thingspeak_channel_id[16];
  char thingspeak_write_api_key[32];
  float noise_threshold_high;
  float noise_threshold_medium;
  float calibration_offset;
  uint16_t measurement_interval;
  uint16_t display_timeout;
  uint8_t gain_mode;
  bool wifi_enabled;
  bool mqtt_enabled;
  bool sd_logging_enabled;
  bool display_enabled;
} config;
```

```
// Змінні для зберігання поточних показників
```

```
struct SystemState {
```

```

float currentNoiseLevel;    // Поточний рівень шуму (dBA)
float minNoiseLevel;       // Мінімальний рівень шуму за період
float maxNoiseLevel;       // Максимальний рівень шуму за період
float eqNoiseLevel;        // Еквівалентний рівень шуму (Leq)
float temperature;         // Температура (°C)
float humidity;            // Відносна вологість (%)
float pressure;            // Атмосферний тиск (гПа)
float batteryVoltage;      // Напруга батареї (В)
uint8_t batteryPercent;    // Рівень заряду батареї (%)
int8_t wifiSignal;         // Рівень сигналу WiFi (dBm)
unsigned long uptime;      // Час роботи з моменту запуску (с)
bool displayActive;        // Стан дисплея
bool sdCardPresent;        // Наявність SD-карти
bool rtcPresent;           // Наявність годинника реального часу
bool sensorPresent;        // Наявність датчиків середовища
bool wifiConnected;        // Стан підключення WiFi
bool mqttConnected;        // Стан підключення MQTT
} state;

// Буфери для зберігання історії вимірювань
float noiseHistory[60];    // Історія рівня шуму за останню годину
uint8_t noiseHistoryIndex = 0; // Індекс для циклічного буфера

// Змінні для керування часом
unsigned long lastMeasurement = 0; // Час останнього вимірювання шуму
unsigned long lastDataLog = 0;     // Час останнього запису на SD-карту
unsigned long lastMqttPublish = 0; // Час останньої передачі на ThingSpeak
unsigned long lastDisplayUpdate = 0; // Час останнього оновлення дисплея
unsigned long lastButtonCheck = 0; // Час останньої перевірки кнопок

```

```
unsigned long displayTimeout = 0; // Час автоматичного вимкнення дисплея
unsigned long lastSensorReading = 0; // Час останнього зчитування датчиків
```

```
// Флаги станів
```

```
bool displayOn = true; // Стан дисплея (увімкнений/вимкнений)
```

```
bool configMode = false; // Режим налаштування
```

```
uint8_t menuPosition = 0; // Позиція в меню
```

```
// Режими вимірювання
```

```
enum MeasurementMode {
```

```
    MODE_FAST = 0, // Швидкий режим (125 мс)
```

```
    MODE_SLOW = 1 // Повільний режим (1 с)
```

```
};
```

```
MeasurementMode measurementMode = MODE_FAST;
```

```
// Прототипи функцій
```

```
void setupDisplay();
```

```
void setupWiFi();
```

```
void setupMQTT();
```

```
void setupSD();
```

```
void setupRTC();
```

```
void setupSensors();
```

```
void setupServer();
```

```
float readNoiseLevel();
```

```
void updateDisplay();
```

```
void updateLEDs();
```

```
void publishToThingSpeak();
```

```
void logDataToSD();
```

```

void handleButtons();
void checkBatteryLevel();
void loadConfig();
void saveConfig();
void handleRoot();
void handleData();
void handleConfig();
float applyAWeighting(float sample, float sampleRate);
void performCalibration();
void deepSleepMode();
void checkWiFiConnection();
float readBatteryVoltage();

/**
 * Функція ініціалізації системи
 */
void setup() {
    // Ініціалізація Serial для відлагодження
    Serial.begin(115200);
    Serial.println("\nМікропроцесорна система моніторингу рівня шуму");
    Serial.println("Ініціалізація...");

    // Ініціалізація пінів
    pinMode(MIC_PIN, INPUT);
    pinMode(MIC_GAIN_PIN, OUTPUT);
    pinMode(DISPLAY_PWR_PIN, OUTPUT);
    pinMode(BTN_UP_PIN, INPUT_PULLUP);
    pinMode(BTN_DOWN_PIN, INPUT_PULLUP);
    pinMode(BTN_SELECT_PIN, INPUT_PULLUP);

```

```

// Вмикаємо живлення дисплея
digitalWrite(DISPLAY_PWR_PIN, HIGH);

// Встановлюємо початковий коефіцієнт підсилення мікрофона
digitalWrite(MIC_GAIN_PIN, LOW); // Коефіцієнт підсилення 40 dB

// Завантаження конфігурації
loadConfig();

// Ініціалізація дисплея
setupDisplay();

// Відображення заставки на дисплеї
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.println("Monitoring System");
display.println("Initializing...");
display.display();

// Ініціалізація RGB-світлодіодів
FastLED.addLeds<WS2812B, LED_PIN, GRB>(leds, NUM_LEDS);
FastLED.setBrightness(LED_BRIGHTNESS);

// Встановлення всіх світлодіодів у синій колір під час ініціалізації
for (int i = 0; i < NUM_LEDS; i++) {
    leds[i] = CRGB::Blue;
}

```

```

}
FastLED.show();

// Ініціалізація підсистем
setupSD();
setupRTC();
setupSensors();

// Ініціалізація WiFi та MQTT, якщо вони включені в конфігурації
if (config.wifi_enabled) {
    setupWiFi();
    if (config.mqtt_enabled && state.wifiConnected) {
        setupMQTT();
    }
}

// Ініціалізація веб-сервера
if (state.wifiConnected) {
    setupServer();
}

// Ініціалізація буфера історії шуму
for (int i = 0; i < 60; i++) {
    noiseHistory[i] = 0.0;
}

// Встановлення початкових значень стану системи
state.currentNoiseLevel = 0.0;
state.minNoiseLevel = 100.0;

```

```
state.maxNoiseLevel = 0.0;
state.eqNoiseLevel = 0.0;
state.temperature = 0.0;
state.humidity = 0.0;
state.pressure = 0.0;
state.batteryVoltage = 0.0;
state.batteryPercent = 0;
state.wifiSignal = 0;
state.uptime = 0;
state.displayActive = true;

// Зчитуємо початковий рівень заряду батареї
checkBatteryLevel();

// Оновлення світлодіодів до зеленого кольору після ініціалізації
for (int i = 0; i < NUM_LEDS; i++) {
  leds[i] = CRGB::Green;
}
FastLED.show();

// Завершення ініціалізації
Serial.println("Ініціалізація завершена");
display.clearDisplay();
display.setCursor(0, 0);
display.println("System Ready");
display.display();
delay(1000);

// Перше вимірювання та оновлення екрану
```

```

state.currentNoiseLevel = readNoiseLevel();
updateDisplay();
}

/**
 * Головний цикл роботи системи
 */
void loop() {
    unsigned long currentMillis = millis();

    // Перевірка кнопок користувача
    if (currentMillis - lastButtonCheck >= 50) { // 50 мс для уникнення брязкоту
        handleButtons();
        lastButtonCheck = currentMillis;
    }

    // Вимірювання рівня шуму
    if (currentMillis - lastMeasurement >= config.measurement_interval) {
        float noiseLevel = readNoiseLevel();
        state.currentNoiseLevel = noiseLevel;

        // Оновлення мінімального та максимального значень
        if (noiseLevel < state.minNoiseLevel) {
            state.minNoiseLevel = noiseLevel;
        }
        if (noiseLevel > state.maxNoiseLevel) {
            state.maxNoiseLevel = noiseLevel;
        }
    }
}

```

```

// Додавання значення до історії
noiseHistory[noiseHistoryIndex] = noiseLevel;
noiseHistoryIndex = (noiseHistoryIndex + 1) % 60;

// Розрахунок еквівалентного рівня шуму (Leq)
float sumSquared = 0.0;
int validSamples = 0;

for (int i = 0; i < 60; i++) {
    if (noiseHistory[i] > 0) {
        sumSquared += pow(10, noiseHistory[i] / 10.0);
        validSamples++;
    }
}

if (validSamples > 0) {
    state.eqNoiseLevel = 10.0 * log10(sumSquared / validSamples);
}

// Оновлення світлодіодної індикації
updateLEDs();

lastMeasurement = currentMillis;
}

// Зчитування даних з додаткових датчиків (кожні 10 секунд)
if (currentMillis - lastSensorReading >= 10000 && state.sensorPresent) {
    state.temperature = dht.readTemperature();
    state.humidity = dht.readHumidity();
}

```

```

if (bmp.begin()) {
    state.pressure = bmp.readPressure() / 100.0; // Перетворення Па в гПа
}

lastSensorReading = currentMillis;
}

// Оновлення дисплея (кожні 500 мс)
if (currentMillis - lastDisplayUpdate >= 500 && displayOn) {
    updateDisplay();
    lastDisplayUpdate = currentMillis;
}

// Запис даних на SD-карту
if (currentMillis - lastDataLog >= DATA_LOG_INTERVAL &&
    config.sd_logging_enabled && state.sdCardPresent) {
    logDataToSD();
    lastDataLog = currentMillis;
}

// Публікація даних в MQTT/ThingSpeak
if (currentMillis - lastMqttPublish >= MQTT_PUBLISH_INTERVAL &&
    config.mqtt_enabled && state.mqttConnected) {
    publishToThingSpeak();
    lastMqttPublish = currentMillis;
}

// Перевірка стану підключення WiFi
if (config.wifi_enabled && currentMillis % 30000 == 0) { // Кожні 30 секунд

```

```

    checkWiFiConnection();
}

// Перевірка рівня заряду батареї (кожні 5 хвилин)
if (currentMillis % 300000 == 0) {
    checkBatteryLevel();
}

// Автоматичне вимкнення дисплея після таймауту
if (displayOn && config.display_timeout > 0 &&
    currentMillis - displayTimeout >= config.display_timeout * 1000) {
    displayOn = false;
    digitalWrite(DISPLAY_PWR_PIN, LOW);
}

// Оновлення часу роботи (uptime)
state.uptime = currentMillis / 1000; // Перетворення мс в секунди

// Обробка веб-сервера при підключеному WiFi
if (state.wifiConnected) {
    server.handleClient();
}

// Обробка MQTT з'єднання
if (config.mqtt_enabled && state.mqttConnected) {
    mqttClient.loop();
}

// Перевірка підключення MQTT
if (!mqttClient.connected()) {

```

```

    setupMQTT(); // Спроба перепідключення
}
}

// Вимірювання рівня сигналу WiFi при підключенні
if (state.wifiConnected && currentMillis % 60000 == 0) { // Кожну хвилину
    state.wifiSignal = WiFi.RSSI();
}

// Затримка для економії ресурсів та енергії
delay(1);
}

/**
 * Зчитування та обробка даних з мікрофона
 * @return Рівень шуму в dBA
 */
float readNoiseLevel() {
    // Змінні для розрахунку
    float sum = 0.0;
    float sumSquared = 0.0;
    float sample = 0.0;
    float sampleVoltage = 0.0;
    float weightedSample = 0.0;
    int samples = 0;

    // Визначення кількості семплів залежно від режиму
    int numSamples = (measurementMode == MODE_FAST) ? SAMPLES_PER_READING /
8 : SAMPLES_PER_READING;

```

```

// Вимірювання вказаної кількості семплів
for (int i = 0; i < numSamples; i++) {
    // Зчитування значення з АЦП
    sample = analogRead(MIC_PIN);

    // Перетворення значення АЦП в напругу
    sampleVoltage = sample * (A_REF_VOLTAGE / ADC_MAX_VALUE);

    // Застосування А-зважування
    weightedSample = applyAWeighting(sampleVoltage, SAMPLE_RATE);

    // Накопичення суми квадратів для обчислення RMS
    sumSquared += weightedSample * weightedSample;
    samples++;

    // Затримка для досягнення потрібної частоти дискретизації
    delayMicroseconds(125); // 8 kHz = 125 мкс між семплами
}

// Розрахунок середньоквадратичного значення (RMS)
float rms = 0.0;
if (samples > 0) {
    rms = sqrt(sumSquared / samples);
}

// Перетворення RMS в децибели
float db = 0.0;
if (rms > 0) {
    // Формула: dB = 20 * log10(V_rms / V_ref) + DB_offset

```

```

    db = 20.0 * log10(rms / MIC_REF_VOLTAGE) + MIC_REF_DB +
config.calibration_offset;
}

// Фільтрація викидів через медіанний фільтр
micSamples.add(db);
float filteredDb = micSamples.getMedian();

// Обмеження значення в розумних межах
if (filteredDb < 30.0) filteredDb = 30.0;
if (filteredDb > 120.0) filteredDb = 120.0;

return filteredDb;
}

/**
 * Застосування А-зважування до сигналу
 * @param sample Значення семплу в вольтах
 * @param sampleRate Частота дискретизації
 * @return Зважене значення
 */
float applyAWeighting(float sample, float sampleRate) {
    // Реалізація спрощеного А-зважувального фільтра
    static float x1 = 0, x2 = 0, y1 = 0, y2 = 0;

    // Коефіцієнти фільтра для А-зважування при 8 kHz
    const float a0 = 1.0;
    const float a1 = -1.8617;
    const float a2 = 0.8745;
    const float b0 = 0.1530;

```

```

const float b1 = 0.3060;
const float b2 = 0.1530;

// Застосування фільтра (прямий форма II)
float y0 = b0*sample + b1*x1 + b2*x2 - a1*y1 - a2*y2;

// Оновлення стану фільтра
x2 = x1;
x1 = sample;
y2 = y1;
y1 = y0;

return y0;
}

/**
 * Оновлення індикації на RGB-світлодіодах
 */
void updateLEDs() {
    // Визначення кольору залежно від рівня шуму
    CRGB color;

    if (state.currentNoiseLevel < config.noise_threshold_medium) {
        // Низький рівень шуму (зелений)
        color = CRGB::Green;
    } else if (state.currentNoiseLevel < config.noise_threshold_high) {
        // Середній рівень шуму (жовтий)
        color = CRGB::Yellow;
    } else {

```

```

// Високий рівень шуму (червоний)
color = CRGB::Red;
}

// Встановлення кольору для всіх світлодіодів
for (int i = 0; i < NUM_LEDS; i++) {
    leds[i] = color;
}

// Залежно від рівня шуму вмикаємо різну кількість світлодіодів
if (state.currentNoiseLevel < 50) {
    // Нижче 50 dB - один світлодіод
    for (int i = 1; i < NUM_LEDS; i++) {
        leds[i] = CRGB::Black;
    }
} else if (state.currentNoiseLevel < 70) {
    // 50-70 dB - два світлодіода
    leds[2] = CRGB::Black;
}

// Оновлення LED-смуги
FastLED.show();
}

/**
 * Оновлення інформації на дисплеї
 */
void updateDisplay() {
    // Перевірка активності дисплея

```

```

if (!displayOn) return;

display.clearDisplay();

// В режимі налаштування відображаємо меню
if (configMode) {
    displayConfigMenu();
    return;
}

// Відображення поточного рівня шуму (великим шрифтом)
display.setTextSize(2);
display.setCursor(0, 0);
display.print(state.currentNoiseLevel, 1);
display.println(" dBA");

// Відображення статистики (малим шрифтом)
display.setTextSize(1);
display.setCursor(0, 20);
display.print("Min: ");
display.print(state.minNoiseLevel, 1);
display.print(" Max: ");
display.println(state.maxNoiseLevel, 1);

display.print("Leq: ");
display.print(state.eqNoiseLevel, 1);
display.print(" Mode: ");
display.println(measurementMode == MODE_FAST ? "Fast" : "Slow");

```

*// Показники датчиків середовища*

```
if (state.sensorPresent) {  
    display.print("T:");  
    display.print(state.temperature, 1);  
    display.print("C H:");  
    display.print(state.humidity, 0);  
    display.print("% P:");  
    display.print(state.pressure, 0);  
    display.println("hPa");  
}
```

*// Інформація про стан системи*

```
display.setCursor(0, 55);
```

*// Стан SD-карти*

```
if (state.sdCardPresent) {  
    display.print("SD ");  
}
```

*// Стан WiFi*

```
if (state.wifiConnected) {  
    int quality = 0;  
    if (state.wifiSignal > -50) {  
        quality = 4; // Відмінний сигнал  
    } else if (state.wifiSignal > -65) {  
        quality = 3; // Дуже добрий сигнал  
    } else if (state.wifiSignal > -75) {  
        quality = 2; // Добрий сигнал  
    } else if (state.wifiSignal > -85) {
```

```

    quality = 1; // Слабкий сигнал
}

display.print("Wi-Fi:");
for (int i = 0; i < quality; i++) {
    display.print("*");
}
display.print(" ");
}

// Стан батареї
display.print("Bat:");
display.print(state.batteryPercent);
display.print("%");

// Оновлення дисплея
display.display();
}

/**
 * Відображення меню налаштування
 */
void displayConfigMenu() {
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.println("== CONFIGURATION ==");

    // Відображення пунктів меню з курсором біля поточної позиції
    String menuItems[] = {

```

```

"WiFi: " + String(config.wifi_enabled ? "ON" : "OFF"),
"MQTT: " + String(config.mqtt_enabled ? "ON" : "OFF"),
"SD logging: " + String(config.sd_logging_enabled ? "ON" : "OFF"),
"Noise threshold: " + String(config.noise_threshold_high),
"Calibration: " + String(config.calibration_offset),
"Measurement mode: " + String(measurementMode == MODE_FAST ? "Fast" : "Slow"),
"Display timeout: " + String(config.display_timeout) + "s",
"Save & Exit",
"Exit without saving"
};

// Відображення видимої частини меню
int startItem = (menuPosition > 3) ? menuPosition - 3 : 0;
for (int i = startItem; i < startItem + 6 && i < 9; i++) {
  if (i == menuPosition) {
    display.print("> ");
  } else {
    display.print(" ");
  }
  display.println(menuItems[i]);
}

display.display();
}

/**
 * Обробка натискань кнопок
 */
void handleButtons() {

```

```
// Зчитування стану кнопок (активні при LOW через підтягуючі резистори)
```

```
bool btnUp = digitalRead(BTN_UP_PIN) == LOW;
```

```
bool btnDown = digitalRead(BTN_DOWN_PIN) == LOW;
```

```
bool btnSelect = digitalRead(BTN_SELECT_PIN) == LOW;
```

```
// Перевірка натискання будь-якої кнопки для активації дисплея
```

```
if ((btnUp || btnDown || btnSelect) && !displayOn) {
```

```
    displayOn = true;
```

```
    digitalWrite(DISPLAY_PWR_PIN, HIGH);
```

```
    displayTimeout = millis();
```

```
    updateDisplay();
```

```
    return;
```

```
}
```

```
// Якщо ми в режимі конфігурації
```

```
if (configMode) {
```

```
    if (btnUp) {
```

```
        menuPosition = (menuPosition > 0) ? menuPosition - 1 : 8;
```

```
        displayTimeout = millis();
```

```
    } else if (btnDown) {
```

```
        menuPosition = (menuPosition < 8) ? menuPosition + 1 : 0;
```

```
        displayTimeout = millis();
```

```
    } else if (btnSelect) {
```

```
        handleMenuSelection(menuPosition);
```

```
        displayTimeout = millis();
```

```
    }
```

```
    } else {
```

```
// Звичайний режим
```

```
    if (btnUp) {
```

```

    // Переключення режиму вимірювання
    measurementMode = (measurementMode == MODE_FAST) ? MODE_SLOW :
MODE_FAST;

    displayTimeout = millis();
} else if (btnDown) {
    // Скидання мін/макс статистики
    state.minNoiseLevel = state.currentNoiseLevel;
    state.maxNoiseLevel = state.currentNoiseLevel;
    displayTimeout = millis();
} else if (btnSelect) {
    // Вхід у режим конфігурації
    configMode = true;
    menuPosition = 0;
    displayTimeout = millis();
}
}
}

/**
 * Обробка вибору пункту меню
 */
void handleMenuSelection(int position) {
    switch (position) {
    case 0:
        config.wifi_enabled = !config.wifi_enabled;
        if (config.wifi_enabled) {
            setupWiFi();
        } else {
            WiFi.disconnect();
            state.wifiConnected = false;

```

```

    }
    break;
case 1:
    config.mqtt_enabled = !config.mqtt_enabled;
    if (config.mqtt_enabled && state.wifiConnected) {
        setupMQTT();
    } else {
        mqttClient.disconnect();
        state.mqttConnected = false;
    }
    break;
case 2:
    config.sd_logging_enabled = !config.sd_logging_enabled;
    break;
case 3:
    // Циклічна зміна порогу шуму (70-75-80-85)
    config.noise_threshold_high += 5;
    if (config.noise_threshold_high > 85) config.noise_threshold_high = 70;
    config.noise_threshold_medium = config.noise_threshold_high - 15;
    break;
case 4:
    performCalibration();
    break;
case 5:
    measurementMode = (measurementMode == MODE_FAST) ? MODE_SLOW :
MODE_FAST;
    break;
case 6:
    // Циклічна зміна таймауту дисплея (0-30-60-120)
    config.display_timeout += 30;

```

```

    if (config.display_timeout > 120) config.display_timeout = 0;
    break;
case 7:
    // Зберегти і вийти
    saveConfig();
    configMode = false;
    break;
case 8:
    // Вийти без збереження
    loadConfig(); // Повернути попередні налаштування
    configMode = false;
    break;
}
}

/**
 * Публікація даних на ThingSpeak через MQTT
 */
void publishToThingSpeak() {
    if (!state.mqttConnected) return;

    // Формування JSON для ThingSpeak
    StaticJsonDocument<512> doc;

    doc["field1"] = state.currentNoiseLevel;
    doc["field2"] = state.minNoiseLevel;
    doc["field3"] = state.maxNoiseLevel;
    doc["field4"] = state.eqNoiseLevel;
    doc["field5"] = state.temperature;

```

```

doc["field6"] = state.humidity;
doc["field7"] = state.pressure;
doc["field8"] = state.batteryVoltage;

char buffer[512];
serializeJson(doc, buffer);

// Формування теми MQTT для ThingSpeak
String topic = "channels/" + String(config.thingspeak_channel_id) + "/publish/" +
String(config.thingspeak_write_api_key);

// Публікація даних
boolean success = mqttClient.publish(topic.c_str(), buffer);

Serial.print("Публікація на ThingSpeak: ");
Serial.println(success ? "успішно" : "помилка");

lastMqttPublish = millis();
}

/**
 * Ініціалізація підключення до WiFi
 */
void setupWiFi() {
  Serial.print("Підключення до WiFi ");

  WiFi.mode(WIFI_STA);
  WiFi.begin(config.wifi_ssid, config.wifi_password);

  int attempts = 0;

```

```

while (WiFi.status() != WL_CONNECTED && attempts < MAX_WIFI_RETRIES) {
    delay(WIFI_RETRY_DELAY);
    Serial.print(".");
    attempts++;
}

if (WiFi.status() == WL_CONNECTED) {
    Serial.print("\nПідключено до WiFi. IP: ");
    Serial.println(WiFi.localIP());
    state.wifiConnected = true;
    state.wifiSignal = WiFi.RSSI();
} else {
    Serial.println("\nПомилка підключення до WiFi");
    state.wifiConnected = false;
}
}

/**
 * Калібрування системи
 */
void performCalibration() {
    // Режим калібрування вимагає зовнішнього джерела шуму з відомим рівнем
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.println("== CALIBRATION ==");
    display.println("Use reference source");
    display.println("with known level (94dB)");
    display.println("");
}

```

```

display.println("Press UP: +1dB");
display.println("Press DOWN: -1dB");
display.println("Press SELECT: Save");
display.display();

float currentOffset = config.calibration_offset;
bool calibrating = true;

while (calibrating) {
    // Вимірювання рівня шуму з поточним зміщенням
    float rawLevel = readNoiseLevel();

    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.println("== CALIBRATION ==");
    display.setTextSize(2);
    display.setCursor(0, 12);
    display.print(rawLevel, 1);
    display.println(" dBA");
    display.setTextSize(1);
    display.print("Offset: ");
    display.println(currentOffset, 1);
    display.println("");
    display.println("UP +1dB DOWN -1dB");
    display.println("SELECT to Save");
    display.display();

    // Обробка натискань кнопок

```

```

if (digitalRead(BTN_UP_PIN) == LOW) {
    currentOffset += 1.0;
    config.calibration_offset = currentOffset;
    delay(200);
} else if (digitalRead(BTN_DOWN_PIN) == LOW) {
    currentOffset -= 1.0;
    config.calibration_offset = currentOffset;
    delay(200);
} else if (digitalRead(BTN_SELECT_PIN) == LOW) {
    calibrating = false;
    delay(200);
}

delay(100);
}

// Збереження нового калібрувального коефіцієнта
saveConfig();
}

/**
 * Головна функція програми
 */
int main() {
    setup();
    while (true) {
        loop();
    }
    return 0;
}

```

Завідувачу кафедри КІС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ  
Андрій ПРИСЯЖНИЙ  
ПІБ здобувача вищої освіти

---

ФІТ, 3 курсу, групи КІ2с-22-1

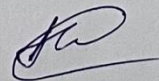
### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

14.06 2025 року



РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Присяжний Андрій Русланович

Тема: Мікропроцесорна система моніторингу рівня шуму на базі ESP8266

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 75

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка мікропроцесорної системи моніторингу рівня шуму на базі ESP8266 з функцією передачі даних через Wi-Fi для забезпечення безперервного контролю акустичного забруднення навколишнього середовища.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз існуючих систем для моніторингу рівня шуму. На основі цього аналізу виявлені як сильні сторони сучасних рішень, так і їхні недоліки. У результаті аналізу сформовано концепцію створення системи моніторингу рівня шуму, адаптованої під побутові умови. Цей розділ визначив основу для створення системи моніторингу рівня шуму, яка повинна бути орієнтована на простоту, функціональність та енергоефективність.

У другому розділі роботи розроблено архітектуру та апаратну частину системи моніторингу рівня шуму. Були визначені оптимальні умови і сформовано завдання для створення такої системи. Також, були детально розглянуті усі апаратні компоненти, використані у системі, наведені їх характеристики. Проведено проєктування компонентів системи та описано структуру передачі даних між ними.

У третьому розділі здійснено реалізацію системи моніторингу рівня шуму у симуляторі та проведено її тестування. Описано особливості її функціонування, сформовано блок-схеми алгоритмів роботи окремих компонентів. Проведено

експериментальні дослідження, у ході яких змодельовані різні сценарії та перевірено алгоритми роботи всіх компонентів системи, включно з логікою реагування на зміну параметрів середовища. Висновки підкріплено прикладами результатів роботи системи, що демонструє високий ступінь практичної реалізації теоретичних рішень.

4. Позитивні сторони роботи: Робота відзначається високим рівнем практичної реалізації та використанням сучасних технологій. Успішно реалізовано систему, що забезпечує моніторинг рівня шуму.

5. Негативні сторони роботи: Недоліком розробленої системи є залежність від стабільного живлення та Wi-Fi-з'єднання, що може впливати на надійність передачі даних між модулями.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Бобко Віталій Олександрович, асистент кафедри ІТБ

“ 19 09 2025 р.

Бобко (підпис)

## Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 0.0%**

**Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 10%**

ID: 245905 Title: БКР Мікропроцесорна система моніторингу рівня шуму на базі ESP8266 Added in a DB: 2025-06-15 Authors: Андрій ПРИСЯЖНИЙ Heads: Володимир ГРИГА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	98319	719	744 (1%)	13 (2%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Андрій ПРИСЯЖНИЙ

**Співавтор:**

**Назва:** Присяжний\_Мікропроцесорна система моніторингу рівня шуму на базі ESP8266

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 0.8%

**Коефіцієнт подібності 2:** 0.2%

**Мікропробіли:** 6

**Заміна букв:** 1

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-15 07:23:49.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

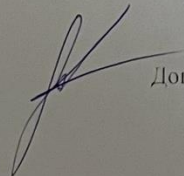
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-15

Дата



Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованою системою виявлення текстових збігів/ідентичності/схожості:

Назва: Мікропроцесорна система моніторингу рівня шуму на базі ESP8266

Автор: Андрій Присяжний

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Грига Володимир Михайлович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) Запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.;
- 2) Усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) Окремі збіги представлені загальноновживаними фразами, наприклад: «на рисунку зображено», «загальна структура системи», «висновки до розділу» тощо.
- 4) Якість запозичень відповідає технічним особливостям дослідження: виявлено збіги в кодах, формулах і термінах, які є вихідними даними до великої кількості задач і не можуть вважатися авторськими порушеннями.
- 5) Система зафіксувала технічні модифікації тексту, зокрема: заміну окремих символів, скорочення індексів у формулах, зміну розміщення символів. Це є наслідком форматування або експорту документа, а не цілеспрямованого уникнення перевірки.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 0,99% та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Володимир ГРИГА

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА