

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютерна система локального мовлення з використанням  
IoT-вузлів та IP-мережі

Виконав: студент IV курсу, групи СІ-41

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Бойчук Н.В.

(підпис)

(прізвище та ініціали)

Керівник

Луцків А.М.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Луцків Н.С.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Рецензент

Литвиненко Я.В.

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осхівська Г.М.

(підпис)

(прізвище та ініціали)

«25» квітня 2026 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

студенту Бойчуку Назарію Віталійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система керування транспортним перехрестям на основі аналізу відеопотоків

Керівник роботи Луцків Андрій Мирославович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» квітня 2026 року № 4.9-188

2. Термін подання студентом завершеної роботи 19.06.2026 р.

3. Вихідні дані до роботи Характеристики систем локального мовлення, мікроконтролер ESP32, принцип роботи HTTP-сервера, особливості протоколу MQTT

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз вимог до систем локального мовлення. 2. Проектування системи локального мовлення з використанням IoT-вузлів та IP-мережі. 3. Реалізація програмної складової комп'ютерної системи локального мовлення 4. Безпека життєдіяльності, основи охорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Структура аналогових і цифрових систем локального мовлення

2. Архітектура системи локального мовлення з використанням IoT-вузлів та IP-мережі

3. Функціональна схема IoT-вузла в системі локального мовлення

4. Блок схема алгоритму роботи системи локального мовлення

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Сенчишин В.С., к.т.н., доц., каф. МТ</i>		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	<i>26.01 – 02.02</i>	
2.	<i>Робота над першим розділом «Аналіз вимог до систем локального мовлення»</i>	<i>03.02 – 15.02</i>	
3.	<i>Робота над другим розділом «Проектування системи локального мовлення з використанням IoT-вузлів та IP-мережі»</i>	<i>20.04 – 25.04</i>	
4.	<i>Робота над третім розділом «Реалізація програмної складової комп'ютерної системи локального мовлення»</i>	<i>26.04 – 05.05</i>	
5.	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>07.05 – 25.05</i>	
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>26.05 – 7.06</i>	
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>26.06.2026</i>	

Студент

\_\_\_\_\_ (підпис)

*Бойчук Назарій Віталійович*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Луцків Андрій Мирославович*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Бойчук Н.В. Комп'ютерна система локального мовлення з використанням IoT-вузлів та IP-мережі: робота на здобуття ступеня бакалавра: спец. 123 – комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: комп'ютерна система, локальне мовлення, IoT-вузол, IP-мережа, аудіоповідомлення.

У кваліфікаційній роботі розроблено комп'ютерну систему локального мовлення з використанням IoT-вузлів та IP-мережі. Система призначена для централізованого керування аудіоповідомленнями, вибору зон мовлення, передавання команд керування та відтворення повідомлень у визначених приміщеннях або групах приміщень.

У роботі проаналізовано традиційні аналогові та IP-орієнтовані системи локального мовлення, визначено їхні переваги й обмеження. Обґрунтовано доцільність використання розподіленої архітектури, у якій сервер керування відповідає за вибір повідомлення, формування команд і журналювання подій, а IoT-вузли виконують безпосереднє відтворення аудіосигналу.

Для реалізації IoT-вузла обрано мікроконтролер ESP32, цифровий I2S-підсилювач MAX98357A та динамік Adafruit Speaker 3" 4 Ом / 3 Вт. Розроблено структурну схему системи, схему підключення компонентів вузла, модель мережевого підключення та логічну модель зон мовлення. Для передавання керуючих команд використано протокол MQTT, а для доступу до аудіофайлів — HTTP.

## ANNOTATION

Boichuk N.V. Computer System for Local Broadcasting Using IoT Nodes and an IP Network: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: computer system, local broadcasting, IoT node, IP network, audio message.

The qualification thesis presents the development of a computer system of local broadcasting using IoT nodes and an IP network. The system is intended for centralized management of audio messages, selection of broadcasting zones, transmission of control commands, and playback of messages in specific rooms or groups of rooms.

The thesis analyzes traditional analog and IP-oriented local broadcasting systems and identifies their advantages and limitations. The feasibility of using a distributed architecture is substantiated. In this architecture, the control server is responsible for selecting messages, generating commands, and logging events, while IoT nodes perform direct playback of the audio signal.

An ESP32 microcontroller, a MAX98357A digital I2S amplifier, and an Adafruit Speaker 3" 4 Ohm / 3 W loudspeaker were selected for the implementation of the IoT node. The system structure diagram, the node component connection diagram, the network connection model, and the logical model of broadcasting zones were developed. The MQTT protocol was used for transmitting control commands, while HTTP was used to access audio files.

## ЗМІСТ

ВСТУП	.....	8
РОЗДІЛ 1	АНАЛІЗ ВИМОГ ДО СИСТЕМ ЛОКАЛЬНОГО МОВЛЕННЯ.....	10
1.1	Призначення та особливості систем локального мовлення .....	10
1.2	Аналіз традиційних та IP-орієнтованих систем мовлення .....	15
1.3	Використання IoT-вузлів та мережевих протоколів у системах локального мовлення .....	20
РОЗДІЛ 2	ПРОЄКТУВАННЯ СИСТЕМИ ЛОКАЛЬНОГО МОВЛЕННЯ	3
	ВИКОРИСТАННЯМ IoT-ВУЗЛІВ ТА IP-МЕРЕЖІ.....	24
2.1	Визначення вимог та проєктування архітектури системи локального мовлення .....	24
2.2	Обґрунтування вибору IoT-вузла системи мовлення .....	30
2.3	Вибір підсилювача аудіосигналу в системі локального мовлення .....	33
2.4	Вибір динаміка та параметрів аудіовиходу .....	36
2.5	Схема підключення компонентів IoT-вузла .....	39
2.6	Структура мережевого підключення та модель зон мовлення .....	42
РОЗДІЛ 3	РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СКЛАДОВОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ЛОКАЛЬНОГО МОВЛЕННЯ.....	47
3.1	Проєктування архітектури програмного забезпечення системи локального мовлення .....	47
3.2	Програмна реалізація серверної частини .....	50
3.2.1	Реалізація HTTP-сервера для роботи з аудіофайлами .....	50

					КС КРБ 123.147.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Бойчук Н.В.			Комп'ютерна система локального мовлення з використанням IoT-вузлів та IP-мережі	Літ.	Арк.	Аркушів
Перевір.		Луцків А.М.					6	
Реценз.		Литвиненко Я.В.				ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

3.2.2	Реалізація MQTT-брокера та модуля передавання керуючих команд.....	54
3.2.3	Реалізація веб-інтерфейсу адміністратора .....	59
3.3	Програмна реалізація IoT-вузла на базі ESP32.....	61
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ....		64
4.1	Діяльність людини та її вплив на довкілля .....	64
4.2	Вплив шуму на організм людини та розробка заходів щодо його зниженню до допустимих величин.....	66
ВИСНОВКИ .....		70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		71
Додаток А Технічне завдання		
Додаток Б Блок схема алгоритму роботи IoT-вузла		

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## ВСТУП

У сучасних умовах функціонування навчальних закладів, адміністративних будівель, виробничих приміщень та інших об'єктів з великою кількістю людей важливого значення набувають системи оперативного інформування. Такі системи дають змогу швидко передавати службові оголошення, навчальні повідомлення, сигнали тривоги, інструкції щодо дій у надзвичайних ситуаціях або іншу важливу інформацію. Особливо актуальним це є для університетів, де одночасно можуть перебувати студенти, викладачі, працівники та відвідувачі, а приміщення розміщені у різних аудиторіях, поверхах або корпусах.

Традиційні системи локального мовлення здебільшого будуються на основі централізованих підсилювачів, аналогових ліній передавання звуку та гучномовців, підключених до окремих зон. Такі рішення можуть бути достатньо надійними, однак вони часто мають обмежену гнучкість, складно масштабуються та потребують значних витрат на прокладання кабельних ліній. Крім того, у класичних системах складніше реалізувати вибірккову трансляцію повідомлень, централізоване журналювання подій, віддалене керування або інтеграцію з іншими інформаційними системами.

Розвиток IP-мереж, бездротових технологій та пристроїв Інтернету речей відкриває можливість побудови більш гнучких комп'ютерних систем локального мовлення. У такому підході передавання аудіоданих може здійснюватися через існуючу локальну мережу, а функції відтворення звуку виконують окремі IoT-вузли, розміщені у відповідних зонах. Кожен такий вузол може мати власний ідентифікатор, належати до певної аудиторії, поверху або корпусу, отримувати команди від сервера та відтворювати задані аудіоповідомлення. Це дозволяє створити систему, яка є більш масштабованою, зручною в адмініструванні та придатною для подальшого розвитку.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю розроблення доступного та технічно обґрунтованого проєкту системи

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

локального мовлення, яка може працювати в межах IP-мережі та використовувати IoT-вузли для відтворення аудіоповідомлень у вибраних зонах. Така система може бути корисною для навчальних корпусів, лабораторій, офісних приміщень або інших об'єктів, де потрібно забезпечити централізоване керування повідомленнями та оперативне інформування людей.

Особливу увагу в такій системі доцільно приділити зонуванню трансляцій і підтримці пріоритету повідомлень. Зонування дає змогу надсилати аудіоповідомлення не на всі пристрої одночасно, а лише до певної групи вузлів, наприклад, на один поверх, в окрему аудиторію або в конкретний корпус.

Пріоритет повідомлень є важливим для випадків, коли службове або екстрене оголошення має перервати звичайну трансляцію. Завдяки цьому система може використовуватися не лише для повсякденних інформаційних оголошень, а й для повідомлень підвищеної важливості.

Метою кваліфікаційної роботи є розроблення комп'ютерної системи локального мовлення з використанням IoT-вузлів та IP-мережі, яка забезпечує централізоване керування аудіотрансляціями, вибір зон мовлення, передавання команд керування та підтримку пріоритету екстрених повідомлень.

					<i>КС КРБ 123.147.00.00 ПЗ</i>	Арк.
						9
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО СИСТЕМ ЛОКАЛЬНОГО МОВЛЕННЯ

### 1.1 Призначення та особливості систем локального мовлення

Системи локального мовлення є важливою складовою інженерної інфраструктури навчальних закладів, адміністративних будівель, підприємств, торгових центрів, транспортних об'єктів та інших приміщень, у яких необхідно забезпечити швидке передавання аудіоінформації певній групі людей [1]. Основним призначенням таких систем є трансляція голосових повідомлень, службових оголошень, попереджувальних сигналів, інструкцій щодо дій у надзвичайних ситуаціях, а також інших аудіоповідомлень у межах заданої території або окремих зон.

У загальному випадку система локального мовлення представляє собою сукупність технічних і програмних засобів, які забезпечують формування, передавання, керування та відтворення аудіосигналів. До її складу можуть входити джерела звуку, пристрої керування, канали передавання даних, підсилювачі, гучномовці, мережеві вузли, програмне забезпечення та засоби адміністрування [2]. Залежно від призначення, така система може використовуватися як для звичайного інформування користувачів, так і для оповіщення про події підвищеної небезпеки чи важливості.

Для університетського середовища локальне мовлення має особливе значення. Навчальні корпуси зазвичай складаються з великої кількості аудиторій, лабораторій, коридорів, адміністративних кабінетів та допоміжних приміщень. У таких умовах виникає потреба в оперативному інформуванні студентів, викладачів і працівників про початок або завершення занять, організаційні повідомлення або необхідність виконання певних дій у разі виникнення небезпечної ситуації [3].

					<b>КС КРБ 123.147.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Аналіз вимог до систем локального мовлення</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Бойчук Н.В.</i>						
<i>Перевір.</i>		<i>Луцків А.М.</i>					10	
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

При цьому не кожне повідомлення повинно транслюватися на весь корпус або кампус. Часто достатньо передати інформацію лише в окрему аудиторію, на певний поверх або в конкретну групу приміщень [1-2].

З огляду на це, сучасна система локального мовлення повинна забезпечувати не лише передавання аудіосигналу, а й гнучке керування напрямком трансляції. Однією з важливих функцій є зонування, тобто можливість об'єднувати окремі пристрої відтворення в логічні групи. Наприклад, у межах університетського корпусу можна виділити зони «перший поверх», «другий поверх», «лабораторії», «адміністративний блок», «усі аудиторії» або «весь корпус». Це дає змогу надсилати повідомлення саме туди, де воно є необхідним, не створюючи зайвого інформаційного шуму в інших приміщеннях [3].

У традиційних системах локального мовлення передавання звуку часто здійснюється за допомогою аналогових аудіоліній. Такі системи можуть містити центральний підсилювач, мікрофонний пульт, комутаційні пристрої та гучномовці, підключені кабельними лініями. Перевагою такого підходу є відносна простота принципу роботи та висока надійність за умови правильної інсталяції. Водночас аналогові системи мають низку обмежень. Зокрема, вони менш гнучкі щодо зміни конфігурації зон, складніші в масштабуванні, потребують прокладання окремих кабельних трас і не завжди дозволяють зручно реалізувати централізований облік подій або дистанційне керування [2-4].

Альтернативою традиційним рішенням є побудова систем локального мовлення на основі IP-мережі. У такому випадку команди керування та аудіодані можуть передаватися через локальну комп'ютерну мережу, а пристрої відтворення реалізуються у вигляді окремих мережевих вузлів. Такий підхід дозволяє використовувати вже наявну мережеву інфраструктуру будівлі, зменшити потребу в окремих аудіолініях, спростити масштабування системи та забезпечити централізоване програмне керування [5].

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, IP-орієнтована архітектура відкриває можливість інтеграції системи мовлення з вебінтерфейсом, базою даних, журналом подій, системами моніторингу та іншими інформаційними сервісами [5].

У межах даної кваліфікаційної роботи проектується система локального мовлення з використанням IoT-вузлів та IP-мережі. IoT-вузол мовлення можна розглядати як компактний мережевий пристрій, який має власний ідентифікатор, підключається до локальної мережі, отримує команди від сервера керування та забезпечує відтворення аудіоповідомлення через підключений динамік. Такий вузол може бути встановлений у конкретній аудиторії, коридорі або іншому приміщенні. Завдяки цьому система стає модульною: для розширення зони покриття достатньо додати новий вузол і зареєструвати його в серверному програмному забезпеченні [5].

Загальну ідею функціонування системи локального мовлення з використанням IoT-вузлів подано на рис. 1.1.

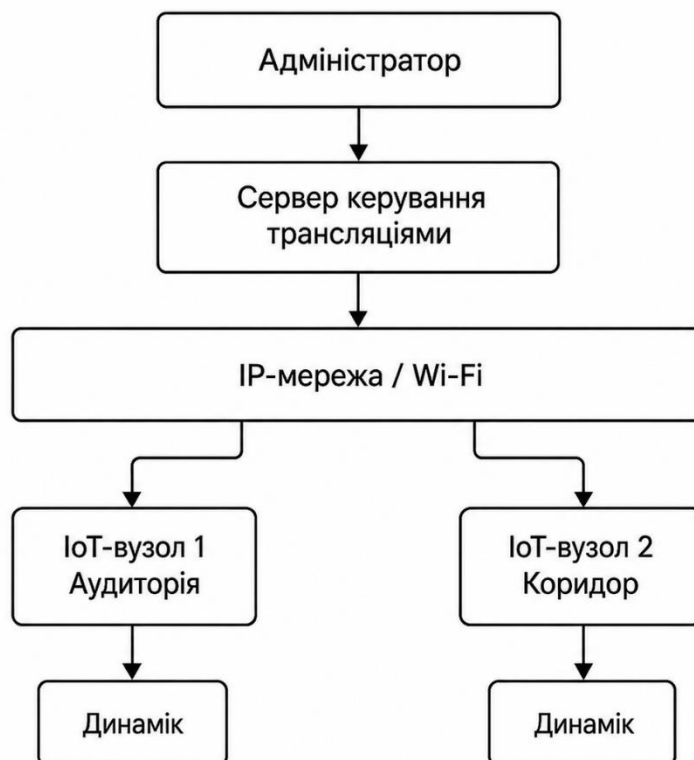


Рисунок 1.1 – Узагальнена структура системи локального мовлення з використанням IoT-вузлів

Основною перевагою такої архітектури є розподілений характер системи. Сервер виконує функції централізованого керування, тоді як відтворення звуку здійснюється безпосередньо на локальних вузлах. Це дозволяє відокремити логіку прийняття рішень від фізичного відтворення повідомлень. Сервер визначає, яке повідомлення потрібно відтворити, у якій зоні та з яким пріоритетом, а IoT-вузол виконує отриману команду.

Важливою особливістю систем локального мовлення є підтримка різних типів повідомлень. У найпростішому випадку система може використовуватися для запуску заздалегідь підготовлених аудіофайлів. Наприклад, це можуть бути повідомлення про початок занять, завершення перерви, технічні оголошення або стандартні інструкції. У складніших системах може бути реалізована трансляція голосу в реальному часі, потокове аудіо або автоматичне формування повідомлень на основі тексту. У кваліфікаційній роботі основна увага буде зосереджена на забезпеченні можливості відтворення попередньо підготовлених аудіофайлів.

Ще однією важливою властивістю системи є підтримка пріоритетів. У реальних умовах різні повідомлення можуть мати різну важливість. Наприклад, фонове або звичайне службове повідомлення може бути перерване екстремим оголошенням. Це означає, що система повинна не лише запускати аудіо, а й аналізувати поточний стан вузла мовлення. Якщо вузол уже відтворює повідомлення, нова команда повинна порівнюватися з поточним рівнем пріоритету. Повідомлення з вищим пріоритетом має перервати поточне відтворення, тоді як повідомлення з нижчим пріоритетом може бути відхилене або поставлене в чергу [5].

Для системи локального мовлення також важливе журналювання подій. Журнал дає змогу зберігати інформацію про те, коли було запущено повідомлення, у яку зону воно було направлено, які вузли його отримали, який мав пріоритет запуск і який результат виконання команди. Наявність журналу підвищує контрольованість системи та дозволяє аналізувати її роботу під час тестування або експлуатації.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Основні функції системи локального мовлення наведено в табл. 1.1.

Таблиця 1.1 – Основні функції системи локального мовлення

Функція системи	Зміст функції	Приклад використання
Передавання аудіо-повідомлень	Запуск голосових або службових повідомлень через динаміки	Оголошення про початок занять
Зонування трансляцій	Вибір окремої аудиторії, поверху, корпусу або групи вузлів	Повідомлення лише для другого поверху
Централізоване керування	Керування всіма вузлами з одного серверного модуля	Адміністратор запускає повідомлення з веб-інтерфейсу
Підтримка пріоритетів	Переривання менш важливих повідомлень більш важливими	Екстрене повідомлення перериває звичайне
Моніторинг стану вузлів	Контроль доступності та активності IoT-вузлів	Перевірка, чи вузол підключений до мережі
Журналювання подій	Збереження інформації про запуснені трансляції	Запис часу, зони, повідомлення та статусу
Масштабування	Можливість додавання нових вузлів мовлення	Додавання вузла для нової аудиторії

З технічної точки зору система локального мовлення повинна відповідати низці вимог. По-перше, вона має забезпечувати стабільне передавання команд між сервером і вузлами. По-друге, вузли повинні мати можливість відтворювати аудіо з достатньою якістю для розбірливого сприйняття голосових повідомлень. По-третє, система повинна мати зрозумілу структуру зон, щоб адміністратор міг легко вибрати напрям трансляції. По-четверте, важливим є забезпечення прийнятної затримки між запуском команди та початком відтворення аудіо [4].

Отже, системи локального мовлення виконують важливу функцію оперативного аудіоінформування в межах окремого об'єкта або групи приміщень.

Використання IoT-вузлів та IP-мережі дозволяє зробити таку систему більш гнучкою, масштабованою та зручною в керуванні. Саме тому обраний підхід є доцільним для реалізації комп'ютерної системи локального мовлення, яка проєктується в межах кваліфікаційної роботи.

## 1.2 Аналіз традиційних та IP-орієнтованих систем мовлення

Традиційні системи локального мовлення тривалий час були основним рішенням для організації аудіоінформування в будівлях. Вони, зазвичай, будуються на основі центрального підсилювача, мікрофонного пульта, комутаційного обладнання, кабельних аудіоліній та гучномовців, розміщених у різних приміщеннях. Джерелом аудіосигналу може бути мікрофон, аудіопрогравач, комп'ютер або інший пристрій, підключений до центрального блока керування. Далі сигнал підсилюється та передається до гучномовців через окремі кабельні лінії [1-3].

У таких системах керування зонами здебільшого реалізується за допомогою комутації аудіоліній. Наприклад, оператор може вибрати, у які приміщення або групи приміщень потрібно передати повідомлення. Проте така комутація часто залежить від фізичної структури кабельної мережі. Якщо потрібно змінити склад зон, додати нове розташування або переналаштувати маршрути передавання звуку, це може потребувати додаткового монтажу, зміни кабельних з'єднань або модернізації обладнання [2].

Перевагою традиційних систем є відносна простота принципу роботи. Аудіосигнал формується в одному місці, підсилюється і передається до гучномовців. Вони не завжди потребують складного програмного забезпечення, а їхня робота зрозуміла для обслуговуючого персоналу [3].

Однак традиційний підхід має низку суттєвих обмежень. Основним недоліком є залежність від фізичної кабельної структури. Для додавання нової зони потрібно прокласти додаткові лінії або змінювати підключення існуючих. Крім того, аналогові системи мають обмежені можливості програмного керування. У них складніше реалізувати гнучке зонування, автоматичне журналювання подій, віддалений моніторинг стану вузлів, інтеграцію з веб-інтерфейсом або централізоване керування через комп'ютерну мережу [4].

Узагальнену структуру традиційної системи локального мовлення подано на рис. 1.2.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.2 – Типова структура традиційної аналогової системи локального мовлення

Ще одним обмеженням аналогових систем мовлення є складність індивідуального керування окремими точками відтворення. У традиційній системі гучномовці часто об'єднуються в групи фізично, а не логічно. Тобто зона визначається тим, як прокладено та підключено кабелі. Якщо потрібно тимчасово змінити склад зони, наприклад, об'єднати кілька аудиторій або передати повідомлення лише в одну конкретну кімнату, така операція може бути неможливою без попередньої зміни схеми підключення [4,5].

На відміну від аналогових систем, IP-орієнтовані системи локального мовлення використовують комп'ютерну мережу як середовище передавання даних. У такій архітектурі аудіоповідомлення та команди керування можуть передаватися через Ethernet або Wi-Fi, а пристрої відтворення підключаються до мережі як окремі вузли. Кожен вузол має власну адресу, ідентифікатор, налаштування та може бути віднесений до однієї або кількох логічних зон [5].

У IP-орієнтованій системі центральний сервер виконує функції керування трансляціями. Він може зберігати список аудіофайлів, облік вузлів, структуру зон, журнал подій і правила пріоритетів. Адміністратор через програмний інтерфейс вибирає повідомлення, зону трансляції та рівень пріоритету. Після цього сервер надсилає відповідні команди вузлам мовлення. Самі вузли приймають команди через мережу, завантажують або отримують аудіодані та відтворюють звук через підключений динамік [3].

Структуру IP-орієнтованої системи локального мовлення подано на рис. 1.3.

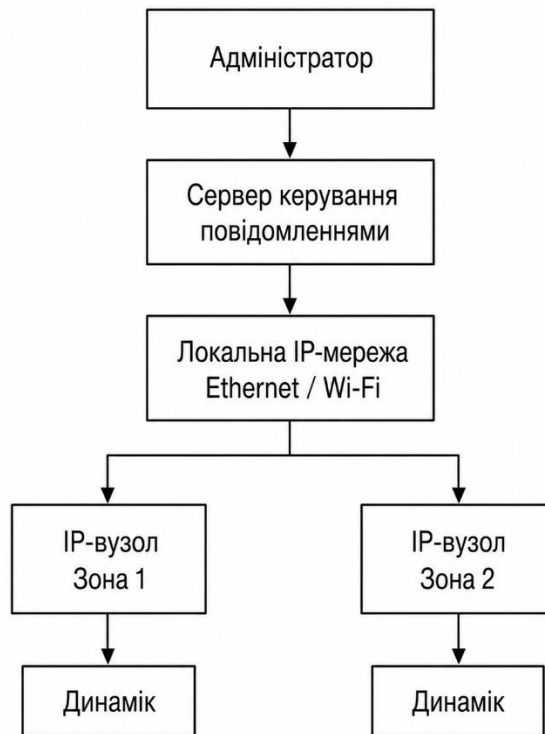


Рисунок 1.3 – Структура IP-орієнтованої системи локального мовлення

Головна перевага IP-орієнтованого підходу полягає в гнучкості [4]. Оскільки вузли підключаються до комп'ютерної мережі, то їх можна логічно об'єднувати в різні групи незалежно від фізичного розташування кабелів. Наприклад, один вузол може належати одночасно до зони «аудиторія», «поверх», «корпус» і «всі вузли». Це дозволяє реалізувати багаторівневе зонування, яке складно або незручно виконувати в аналогових системах.

Ще однією перевагою є можливість централізованого програмного керування. Сервер може не лише надсилати команди, а й зберігати історію трансляцій, фіксувати стан вузлів, відображати їх доступність, змінювати гучність, надсилати тестові сигнали або керувати пріоритетами повідомлень. Така функціональність є важливою для сучасних об'єктів, де система мовлення повинна бути не ізольованим технічним рішенням, а частиною ширшої інформаційної інфраструктури [3].

IP-орієнтовані системи також мають переваги з точки зору масштабування. Для додавання нового вузла не обов'язково прокладати окрему аналогову аудіолінію від центрального підсилювача. Достатньо забезпечити підключення вузла до мережі та зареєструвати його в системі керування. У разі використання Wi-Fi це може істотно спростити розгортання прототипу, особливо в умовах навчальної лабораторії або невеликого корпусу [2].

Водночас IP-орієнтовані системи мають і певні недоліки. Їхня працездатність залежить від якості мережевого з'єднання, стабільності Wi-Fi або Ethernet-інфраструктури, правильного налаштування мережевих протоколів і програмного забезпечення. У разі перевантаження мережі можуть виникати затримки або нестабільність відтворення. Крім того, для таких систем потрібно передбачати базові механізми безпеки: обмеження доступу до серверного інтерфейсу, контроль команд, захист від несанкціонованого запуску повідомлень [5].

Порівняння традиційних та IP-орієнтованих систем локального мовлення наведено в табл. 1.2.

Таблиця 1.2 – Порівняння традиційних та IP-орієнтованих систем мовлення

Критерій порівняння	Традиційна аналогова система	IP-орієнтована система
Середовище передавання	Аналогові аудіолінії	Локальна IP-мережа, Ethernet або Wi-Fi

Продовження таблиці 1.2

Критерій порівняння	Традиційна аналогова система	IP-орієнтована система
Основний принцип роботи	Центральне формування та підсилення аудіосигналу	Передавання команд і аудіоданих до мережевих вузлів
Керування зонами	Через фізичну комутацію ліній	Через програмне логічне групування вузлів
Масштабування	Потребує додаткових кабельних ліній	Виконується шляхом додавання нових мережевих вузлів
Гнучкість налаштування	Обмежена структурою підключення	Висока, з можливістю зміни зон у програмі
Журналювання подій	Зазвичай відсутнє або обмежене	Може реалізовуватися на сервері
Віддалене керування	Обмежене	Може здійснюватися через веб-інтерфейс або мережевий застосунок
Індивідуальне керування вузлами	Ускладнене	Доступне для кожного вузла окремо
Залежність від комп'ютерної мережі	Низька	Висока
Складність програмної реалізації	Невисока	Вища, потребує серверного ПЗ та протоколів

Як видно з табл. 1.2, традиційні системи мають переваги у простоті та незалежності від комп'ютерної мережі, однак поступаються IP-орієнтованим рішенням за гнучкістю, масштабованістю та можливостями програмного керування.

Окремо варто зазначити, що IP-системи локального мовлення можуть реалізовуватися різними способами. У професійних рішеннях часто використовуються спеціалізовані мережеві аудіопристрої, IP-гучномовці, апаратні контролери та програмні платформи керування.

У межах даної роботи доцільно використовувати архітектуру, у якій сервер керування відповідає за вибір аудіоповідомлення та зони відтворення аудіо, формування команди та фіксацію події.

IoT-вузол відповідає за приймання команди, отримання аудіоданих і відтворення звуку [4]. Такий розподіл функцій дозволяє уникнути надмірної

складності вузла та водночас забезпечити централізоване керування всією системою.

Для передавання команд керування в IP-орієнтованій системі можуть використовуватися різні протоколи. Одним із варіантів є MQTT, який побудований на моделі публікації та підписки. У такій моделі сервер публікує повідомлення в певну групу, а відповідні вузли, підписані на цю групу, отримують команду. Це добре підходить для системи з кількома вузлами, оскільки дозволяє надсилати команди як окремому вузлу, так і групі пристроїв. Для передавання самих аудіофайлів можна використати HTTP, оскільки цей протокол є простим, поширеним і зручним для отримання файлів із сервера.

### 1.3 Використання IoT-вузлів та мережевих протоколів у системах локального мовлення

У межах системи локального мовлення IoT-вузол можна розглядати як компактний мережевий пристрій, призначений для приймання команд керування та відтворення аудіосигналу в певній зоні [6]. Він підключається до локальної IP-мережі за допомогою Wi-Fi або Ethernet, отримує команди від сервера керування, завантажує або приймає аудіодані та відтворює їх через підключений динамік.

Основними складовими IoT-вузла мовлення є мікроконтролер, мережевий інтерфейс, аудіомодуль, підсилювач і динамік. У разі використання мікроконтролера з вбудованим Wi-Fi модулем, вузол може підключатися до безпроводної мережі, отримувати команди через мережевий протокол, обробляти їх програмно та передавати цифровий аудіосигнал на підсилювач через інтерфейс I2S. У такому випадку кожен вузол є окремим елементом комп'ютерної системи, який має власний ідентифікатор, належить до певної зони мовлення, може мати індивідуальні налаштування гучності та передавати серверу інформацію про свій стан [7].

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання IoT-вузлів дає змогу реалізувати модульну архітектуру системи. Це означає, що розширення системи не потребує повної перебудови її структури. Для додавання нової точки мовлення достатньо встановити новий вузол, підключити його до мережі та зареєструвати на сервері керування. Роль IoT-вузла в системі локального мовлення можна подати у вигляді табл. 1.3.

Таблиця 1.3 – Основні функції IoT-вузла мовлення

Функція IoT-вузла	Зміст функції
Підключення до мережі	Встановлення з'єднання з локальною IP-мережею через Wi-Fi або Ethernet
Приймання команд	Отримання команд запуску, зупинки, зміни гучності або пріоритету
Відтворення аудіо	Запуск аудіоповідомлення через підключений динамік
Ідентифікація	Використання унікального ідентифікатора вузла в системі
Належність до зони	Робота у складі однієї або кількох зон мовлення
Передавання стану	Надсилання серверу інформації про доступність або результат виконання команди
Обробка пріоритетів	Перевірка важливості нового повідомлення порівняно з поточним

Для ефективної роботи IoT-вузлів важливе значення має вибір мережевих протоколів. У системі локального мовлення потрібно передавати два основні типи даних: команди керування та аудіодані. Команди керування мають невеликий обсяг, але повинні доставлятися швидко та надійно. До таких команд належать запуск повідомлення, зупинка відтворення, зміна гучності, вибір пріоритету або перевірка стану вузла. Аудіодані, навпаки, мають значно більший обсяг і можуть передаватися як у вигляді файлів, так і у вигляді потокового сигналу [8].

Одним із найзручніших протоколів для передавання команд у системах Інтернету речей є MQTT. Цей протокол побудований на моделі “видавець – брокер – підписник”. Сервер або інший пристрій публікує повідомлення в певну групу, MQTT-брокер приймає це повідомлення і передає його тим клієнтам, які

підписані на відповідну групу. Такий підхід добре підходить для системи локального мовлення, оскільки дає змогу надсилати команди як окремому вузлу, так і групі вузлів [5-8].

Для передавання аудіофайлів доцільно використовувати HTTP. У цьому випадку аудіофайл зберігається на сервері, а IoT-вузол отримує команду, у якій міститься посилання на файл. Після цього вузол завантажує або відкриває цей файл і відтворює його. Крім того, HTTP не потребує складної додаткової інфраструктури і може бути реалізований за допомогою звичайного веб-сервера або серверного застосунку [7].

Окрім MQTT і HTTP, у системах мовлення можуть використовуватися й інші протоколи. Наприклад, WebSocket забезпечує постійне двостороннє з'єднання між клієнтом і сервером та може бути зручним для веб-інтерфейсу адміністратора. Протоколи RTP або UDP можуть застосовуватися для потокового передавання аудіо в реальному часі, однак їх реалізація є складнішою, особливо якщо потрібно забезпечити синхронізацію кількох вузлів. Порівняння можливих протоколів мережевої взаємодії подано в табл. 1.4.

Таблиця 1.4 – Порівняння протоколів для системи локального мовлення

Протокол	Призначення в системі	Переваги	Обмеження
MQTT	Передавання команд керування	Невеликий обсяг повідомлень, зручна модель підписок, придатність для IoT	Потребує MQTT-брокера
HTTP	Передавання або завантаження аудіофайлів	Простота реалізації, поширеність, зручність роботи з файлами	Не забезпечує точну синхронізацію потокового аудіо
WebSocket	Двосторонній обмін між інтерфейсом і сервером	Зручний для оновлення стану в реальному часі	Складніший за HTTP для простого прототипу
RTP/UDP	Потокове аудіо в реальному часі	Менші затримки для live-трансляцій	Складніша реалізація та контроль якості передавання

У межах розроблюваної системи доцільно застосувати комбінований підхід. Сервер керування формує команду, у якій зазначається тип дії, адреса аудіофайлу, зона трансляції, рівень гучності та пріоритет повідомлення. Ця команда передається до відповідних IoT-вузлів через MQTT. Після отримання команди вузол звертається до HTTP-сервера, отримує потрібний аудіофайл і запускає його відтворення. Така схема дозволяє розділити керування та передавання аудіоданих, що спрощує структуру системи.

					<i>КС КРБ 123.147.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

## РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ ЛОКАЛЬНОГО МОВЛЕННЯ З ВИКОРИСТАННЯМ ІоТ-ВУЗЛІВ ТА ІР-МЕРЕЖІ

### 2.1 Визначення вимог та проектування архітектури системи локального мовлення

Розроблювана комп'ютерна система локального мовлення призначена для передавання аудіо повідомлень у межах локальної ІР-мережі з використанням серверного модуля керування та ІоТ-вузлів мовлення [9]. Загальна логіка формування вимог базується на тому, що система повинна поєднувати три основні складові: сервер керування, локальну ІР-мережу та ІоТ-вузли мовлення. Основні функціональні вимоги до системи наведено в табл. 2.1.

Таблиця 2.1 – Функціональні вимоги до системи локального мовлення

Вимога	Зміст вимоги
Запуск аудіоповідомлення	Система повинна дозволяти адміністратору запускати вибране аудіоповідомлення
Вибір зони мовлення	Повідомлення має передаватися до одного вузла, групи вузлів або всіх вузлів
Керування відтворенням	Система повинна підтримувати команди запуску, зупинки та зміни гучності
Передавання команд	Команди керування повинні передаватися через локальну ІР-мережу
Підтримка пріоритетів	Екстрені повідомлення повинні мати можливість переривати звичайні
Журналювання подій	Система повинна зберігати інформацію про запуснені трансляції
Контроль стану вузлів	Сервер повинен мати можливість визначати доступність вузлів мовлення

					<b>КС КРБ 123.147.00.00 ПЗ</b>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Бойчук Н.В.</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Луцків А.М.</i>				24	
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>			<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Луцик Н.С.</i>					
<i>Затверд.</i>		<i>Осухівська Г.М.</i>					
					<i>Проектування комп'ютерної системи локального мовлення з використанням ІоТ-вузлів та ІР-мережі</i>		

До функціональних вимог також належить підтримка логічного зонування. Це означає, що фізичне розташування пристрою не повинно жорстко обмежувати сценарії його використання. Наприклад, один IoT-вузол може належати до зони конкретної аудиторії, зони поверху, зони корпусу або загальної зони всієї системи. Такий підхід дозволяє гнучко керувати трансляціями та не потребує зміни фізичної схеми підключення при зміні логіки роботи системи.

Апаратні вимоги визначають склад технічних засобів, які будуть використані для реалізації системи. Основним виконавчим елементом є IoT-вузол мовлення. Він повинен мати мережевий інтерфейс, засоби обробки команд, аудіовихід і можливість підключення до динаміка. Для реалізації прототипу доцільно використовувати мікроконтролер ESP32, оскільки він має вбудований Wi-Fi, достатню продуктивність для обробки команд і підтримує роботу з цифровим аудіоінтерфейсом I2S. Для формування звуку може бути використаний I2S-підсилювач, наприклад MAX98357A, який забезпечує передавання аудіосигналу на динамік [10-12].

Вимоги до програмного забезпечення стосуються серверної частини, IoT-вузлів і способів взаємодії між ними. Сервер повинен забезпечувати вибір аудіофайлів, формування команд управління, визначення списку вузлів відповідно до обраної зони, передавання команд у мережу та запис подій у журнал. Програмне забезпечення IoT-вузла повинно забезпечувати підключення до мережі, отримання команд, аналіз їх параметрів, запуск або зупинку відтворення, а також надсилання інформації про стан вузла.

Для передавання команд керування доцільно використовувати MQTT, оскільки цей протокол добре підходить для IoT-систем, має невеликий обсяг службових повідомлень і підтримує модель підписки. Для доступу до аудіофайлів доцільно використати HTTP, оскільки цей протокол є простим і зручним для передавання файлів у локальній мережі. Такий поділ дозволяє відокремити канал керування від каналу отримання аудіо даних [12].

До нефункціональних вимог, які висуваються до системи локального мовлення, належать простота розгортання, масштабованість, прийнятна

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

затримка запуску повідомлення, стабільність роботи в межах локальної мережі та можливість подальшого розширення [13].

Основні нефункціональні вимоги:

- система повинна мати просту та зрозумілу структуру;
- компоненти системи повинні бути доступними для реалізації навчального прототипу;
- затримка між запуском команди та початком відтворення має бути прийнятною для службових оголошень;
- система повинна підтримувати подальше додавання нових вузлів;
- програмна логіка має бути достатньо простою для тестування та пояснення;
- архітектура повинна дозволяти подальше вдосконалення.

Сформовані вимоги визначають основні компоненти, які повинні бути представлені у структурній схемі системи. До них належать адміністратор або користувацький інтерфейс, сервер керування, MQTT-брокер, HTTP-сервер аудіофайлів, локальна IP-мережа, IoT-вузли мовлення, аудіопідсилювачі та динаміки. Архітектура комп'ютерної системи локального мовлення з використанням IoT-вузлів та IP-мережі показана на рис. 2.1.

У поданій структурі адміністратор є користувачем системи, який виконує керування трансляціями. Він може працювати із серверним програмним забезпеченням через вебінтерфейс, локальний застосунок або інший засіб керування. Основними діями адміністратора є вибір повідомлення, визначення зони мовлення, встановлення пріоритету та запуск або зупинка трансляції.

Сервер керування є центральним елементом системи. Його завдання полягає в обробці дій адміністратора, зберіганні інформації про вузли та зони, формуванні команд керування, організації доступу до аудіофайлів і веденні журналу подій. У межах роботи сервер може бути реалізований на персональному комп'ютері або ноутбуці, підключеному до тієї самої локальної мережі, що й IoT-вузли [14]. Це спрощує розгортання системи та дозволяє проводити тестування без використання складної серверної інфраструктури.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

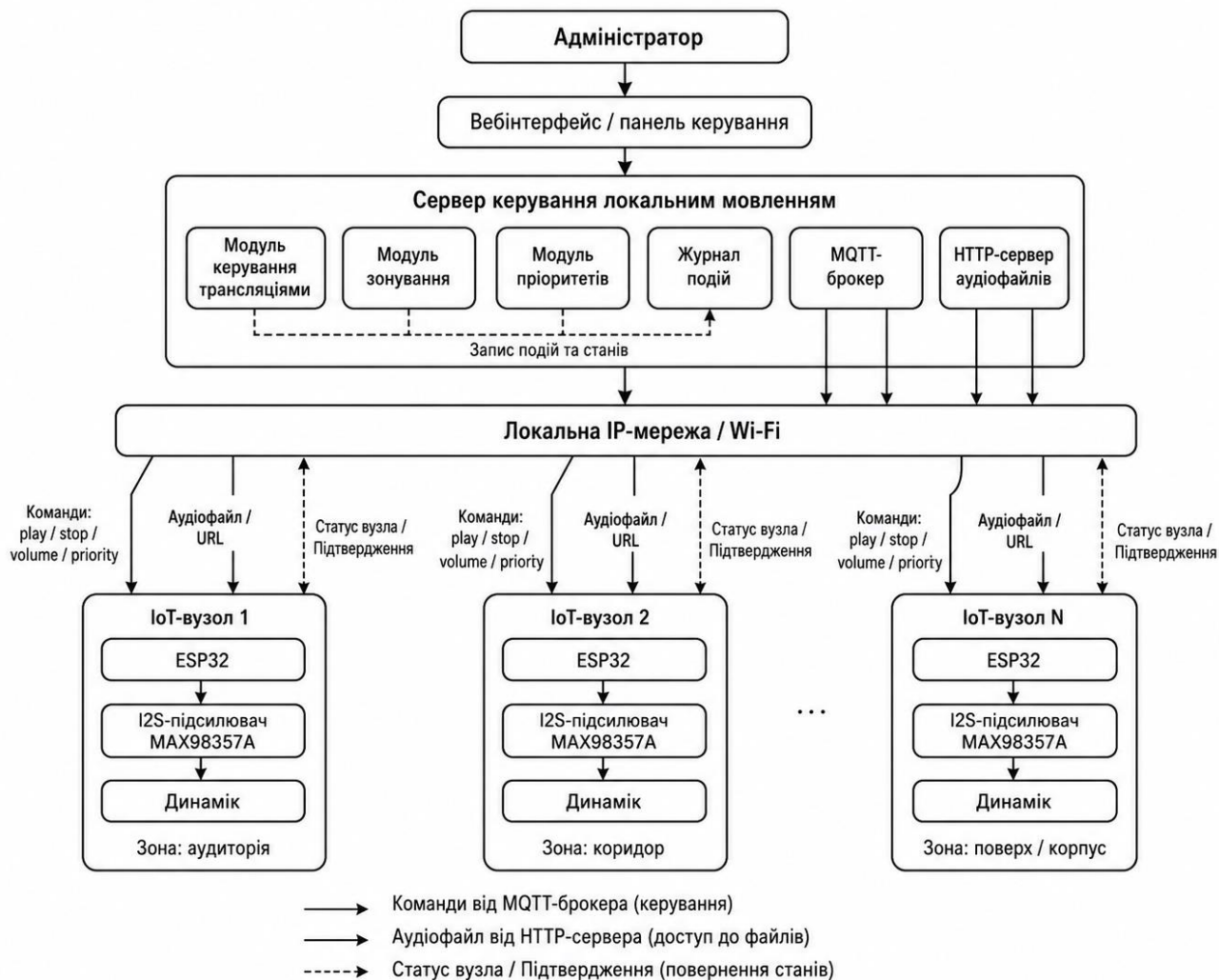


Рисунок 2.1 – Архітектура комп’ютерної системи локального мовлення

Локальна IP-мережа виконує роль середовища передавання даних між сервером та вузлами мовлення. У практичному варіанті вона може бути організована на основі Wi-Fi, оскільки не потребує прокладання додаткових кабельних ліній до кожного вузла. У більш надійному варіанті частина вузлів може підключатися через Ethernet, однак для базової реалізації достатньо бездротового підключення [15].

IoT-вузол мовлення є кінцевим виконавчим елементом системи. Його основне призначення полягає в отриманні команд від сервера і відтворенні аудіоповідомлень у відповідному приміщенні або зоні. У межах даної роботи доцільно використовувати вузол на базі ESP32, оскільки ця платформа має вбудований Wi-Fi і достатні можливості для реалізації простого мережевого аудіопристрою.

Для передавання аудіосигналу від IoT-вузла до динаміка використовується аудіопідсилювач. Оскільки вузол побудований на базі ESP32, то доцільним є застосування I2S-підсилювача MAX98357A. Він приймає цифровий аудіосигнал через інтерфейс I2S і формує сигнал для динаміка. Така схема зручна тим, що не потребує окремого цифро-аналогового перетворювача і добре підходить для компактного вузла локального мовлення [16].

У структурній схемі важливо розрізнити канал керування та канал передавання аудіоданих. Канал керування використовується для передавання коротких команд, наприклад запуску, зупинки, зміни гучності або встановлення пріоритету. Для цього доцільно використовувати MQTT, оскільки він добре підходить для IoT-систем і дозволяє надсилати повідомлення як окремому вузлу, так і групі вузлів. Канал передавання аудіоданих може бути реалізований через HTTP. У цьому випадку сервер зберігає аудіофайли, а вузол після отримання команди звертається до відповідного файлу за посиланням [15-17].

Такий поділ функцій робить систему простішою для реалізації та тестування. MQTT-команди мають невеликий об'єм і швидко передаються мережею, а аудіофайли можуть зберігатися окремо на сервері. Наприклад, команда, яку отримує IoT-вузол, може містити назву повідомлення, URL-адресу аудіофайлу, рівень гучності та пріоритет. Після цього вузол самостійно звертається до HTTP-сервера і запускає відтворення.

З погляду апаратної організації кожен IoT-вузол можна розглядати як окремий модуль, що складається з кількох функціональних частин. До них належать модуль живлення, мережевий модуль, обчислювальний модуль, аудіомодуль і динамік. Оскільки ESP32 поєднує в собі мікроконтролер і Wi-Fi-модуль, структура вузла спрощується. Йому додатково потрібні лише аудіопідсилювач, динамік і стабільне джерело живлення [16].

Функціональну структуру IoT-вузла мовлення та схему підключення його компонентів представлено на рис. 2.2.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

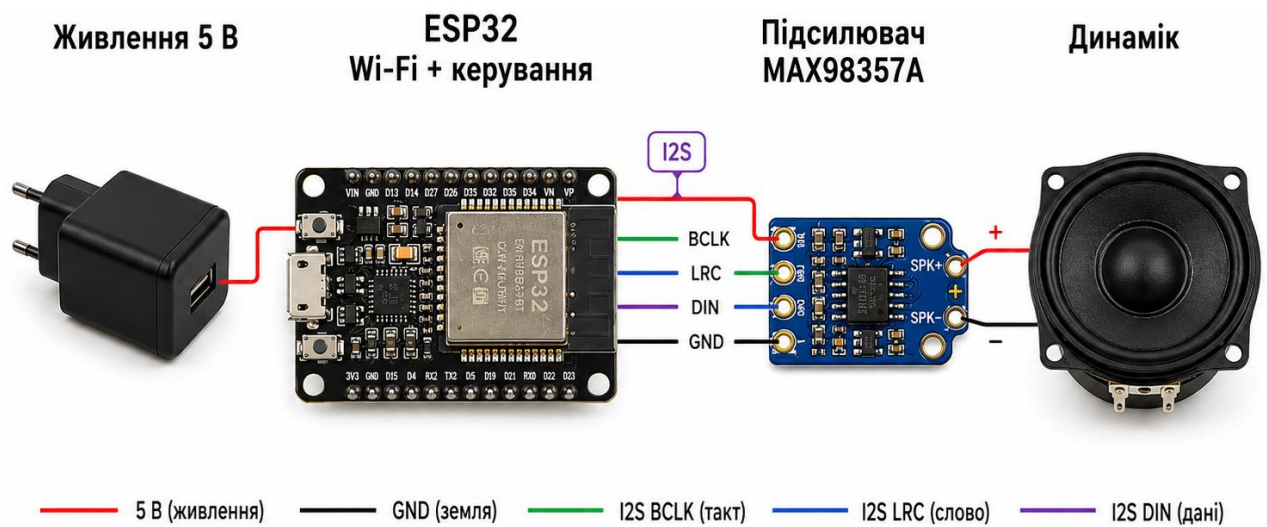


Рисунок 2.2 – Функціональна схема IoT-вузла системи локального мовлення

У такій структурі ESP32 виконує одразу кілька функцій: підключається до Wi-Fi-мережі, отримує команди від сервера, обробляє параметри повідомлення, керує відтворенням і передає цифровий аудіосигнал на підсилювач [14].

Підсилювач MAX98357A забезпечує перетворення цифрового аудіосигналу у форму, придатну для відтворення через динамік. Динамік є кінцевим елементом, який забезпечує озвучення повідомлення в певній зоні.

Важливою особливістю структурної схеми є можливість масштабування. Якщо потрібно розширити систему, до неї можна додати новий IoT-вузол. Для цього вузол підключається до мережі, отримує власний ідентифікатор і додається до відповідної зони на сервері. При цьому загальна логіка системи не змінюється. Це є однією з головних переваг IP-орієнтованої архітектури порівняно з традиційними аналоговими системами мовлення [16].

У кваліфікаційній роботі передбачається два типи зон: індивідуальна та групову. Індивідуальна зона відповідає одному вузлу, наприклад, окремій аудиторії. Групова зона може охоплювати кілька вузлів, наприклад, коридор і навчальні приміщення одного поверху. Додатково може бути створена загальна зона, яка включає всі вузли системи. Така організація дозволяє перевірити роботу системи в різних сценаріях. Приклад логічного поділу вузлів на зони наведено в табл. 2.2.

Таблиця 2.2 – Приклад організації зон мовлення

Зона мовлення	Склад зони	Призначення
Аудиторія	Один IoT-вузол	Передавання повідомлення в окреме приміщення
Коридор	Один або кілька вузлів	Оповіщення у спільному просторі
Поверх	Група вузлів	Передавання повідомлення на один поверх
Усі вузли	Усі зареєстровані пристрої	Загальне повідомлення для всієї системи

Запропонована структура забезпечує централізоване керування, підтримує роботу з кількома вузлами, дозволяє реалізувати зонування, використовує IP-мережу як середовище передавання даних і може бути побудована на доступних апаратних компонентах.

Отже, загальна структурна схема комп'ютерної системи локального мовлення передбачає взаємодію сервера керування, локальної IP-мережі та IoT-вузлів, які виконують відтворення аудіоповідомлень у заданих зонах. Основою системи є розподіл функцій між центральним сервером і кінцевими вузлами мовлення. Такий підхід дозволяє створити гнучкий, масштабований і технічно зрозумілий прототип, який поєднує апаратні засоби, мережеву взаємодію та програмне керування

## 2.2 Обґрунтування вибору IoT-вузла системи мовлення

Для реалізації IoT-вузла локального мовлення можна розглянути кілька апаратних варіантів. Найбільш доцільними для даної задачі є мікроконтролер на базі ESP32 та одноплатний комп'ютер Raspberry Pi. Обидва варіанти можуть бути використані для побудови мережевого аудіовузла, однак вони суттєво відрізняються за функціональністю, складністю програмування, вартістю та вимогами до живлення [11].

Плата ESP32 є мікроконтролерною платформою, орієнтованою на задачі Інтернету речей (рис. 2.3). Її важливою перевагою є наявність вбудованого Wi-

Fi-модуля, що дозволяє безпосередньо підключати пристрій до локальної мережі. Крім того, ESP32 підтримує інтерфейс I2S, який може використовуватися для передавання цифрового аудіосигналу на зовнішній аудіопідсилювач, зокрема MAX98357A. Завдяки цьому ESP32 може виконувати роль компактного мережевого аудіовузла, здатного приймати команди від сервера та запускати відтворення аудіоповідомлень [10-13].

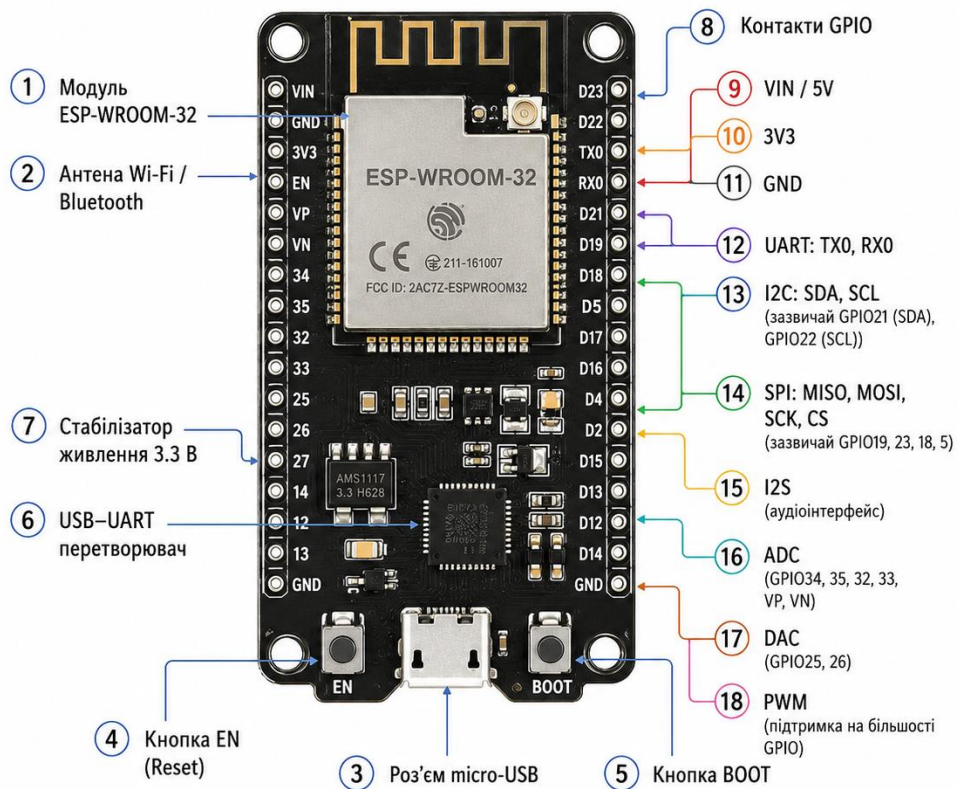


Рисунок 2.3 – Мікроконтролер ESP32

Raspberry Pi, на відміну від ESP32, є повноцінним одноплатним комп'ютером. Він працює під керуванням операційної системи, має значно більші обчислювальні можливості, підтримує різні способи роботи з аудіо та може легко обробляти потокове мовлення. Таке рішення є зручним для складніших систем, у яких потрібна робота з великими файлами, організація складної логіки обробки або розширений користувацький інтерфейс. Однак у цьому випадку застосування Raspberry Pi може бути надлишковим, оскільки має вищу вартість, більше енергоспоживання та потребує обслуговування операційної системи [14].

Порівняння потенційних апаратних платформ для IoT-вузла системи локального мовлення наведено в табл. 2.3.

Таблиця 2.3 – Порівняння апаратних платформ для IoT-вузла мовлення

Критерій	ESP32	Raspberry Pi
Тип пристрою	Мікроконтролерна платформа	Одноплатний комп'ютер
Наявність Wi-Fi	Є в більшості модулів	Є в окремих моделях
Підтримка аудіо	Через I2S-підсилювач або зовнішній аудіомодуль	Через аудіовихід, USB DAC, I2S DAC або HDMI
Обчислювальні можливості	Достатні для IoT-команд і простого аудіовідтворення	Високі, придатні для складної обробки
Складність програмної реалізації	Середня, потрібне програмування мікроконтролера	Нижча для аудіофайлів, але вища через ОС
Енергоспоживання	Невисоке	Вище
Вартість	Нижча	Вища
Масштабування	Економічно доцільне	Дорожче при великій кількості вузлів

З аналізу видно, що більш доцільним є використання ESP32. Основна причина полягає в тому, що система локального мовлення, яка проектується в межах роботи, не потребує складної обробки аудіо безпосередньо на вузлі. Основні функції вузла полягають у прийманні команди, перевірці її параметрів, отриманні аудіофайлу або потоку та відтворенні звуку. Для таких задач можливостей ESP32 достатньо [18].

Важливою перевагою ESP32 є компактність. Вузол на його основі може бути невеликого розміру та складатися з мінімальної кількості компонентів: плати ESP32, I2S-підсилювача MAX98357A, динаміка та джерела живлення. Така структура добре відповідає ідеї розподіленої системи, у якій окремі вузли встановлюються в різних приміщеннях або зонах. У разі необхідності кількість вузлів можна збільшувати без значного зростання вартості всієї системи [19].

Ще одним аргументом на користь ESP32 є підтримка інтерфейсу I2S. Цей інтерфейс призначений для передавання цифрового аудіосигналу між мікроконтролером і аудіопристроєм. У даній системі ESP32 може передавати

аудіодані на підсилювач МАХ98357А, який приймає цифровий сигнал, виконує його перетворення та формує сигнал для динаміка. Завдяки цьому не потрібно використовувати окремий цифрово-аналоговий перетворювач або складний аудіотракт.

З погляду мережевої взаємодії ESP32 також відповідає вимогам системи. Він може підключатися до Wi-Fi-мережі, працювати з MQTT-командами та звертатися до HTTP-сервера для отримання аудіофайлів. Це дозволяє реалізувати запропоновану архітектуру, у якій MQTT використовується для керування, а HTTP – для доступу до аудіоданих. Таким чином, вибрана платформа узгоджується як з апаратною, так і з програмною частиною системи.

З урахуванням виконуваних функцій, структурно IoT-вузол можна зобразити як поєднання чотирьох основних блоків: модуля мережевої взаємодії, модуля обробки команд, аудіомодуля та блока живлення. Модуль мережевої взаємодії забезпечує підключення до Wi-Fi та обмін даними із сервером. Модуль обробки команд відповідає за аналіз отриманих MQTT-повідомлень. Аудіомодуль реалізує передавання цифрового сигналу на підсилювач і подальше відтворення. Блок живлення забезпечує стабільну роботу ESP32 та підсилювача.

Отже, для реалізації IoT-вузла комп'ютерної системи локального мовлення доцільно використати апаратну платформу ESP32. Вона забезпечує необхідний рівень функціональності для роботи в локальній IP-мережі, підтримує бездротове з'єднання, може взаємодіяти з сервером керування, працювати з MQTT та HTTP, а також передавати цифровий аудіосигнал на I2S-підсилювач.

### 2.3 Вибір підсилювача аудіосигналу в системі локального мовлення

Безпосередньо підключати динамік до виводів ESP32 недоцільно. Це обумовлено його не здатністю до формування потужного звукового сигналу. Виходи ESP32 можуть передавати цифрові дані або керуючі сигнали, але не забезпечують необхідної потужності для роботи динаміків. Тому до складу вузла мовлення вводиться окремий аудіопідсилювальний модуль [15].

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

У даній системі обрано модуль MAX98357A, який є цифровим аудіопідсилювачем із підтримкою інтерфейсу I2S (рис. 2.4). Його основне призначення полягає у прийманні цифрового аудіосигналу від ESP32, перетворенні та підсиленні сигналу до рівня, достатнього для відтворення через динамік.

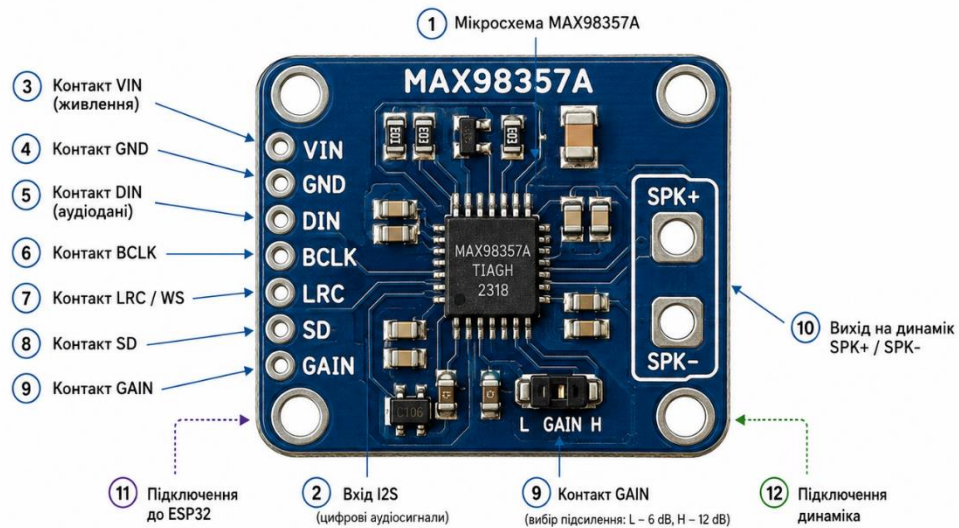


Рисунок 2.4 – Підсилювач аудіосигналу MAX98357A

Особливістю MAX98357A є те, що він працює без окремого зовнішнього цифро-аналогового перетворювача. У багатьох аудіосистемах цифровий сигнал спочатку потрібно перетворити в аналоговий, а потім подати на підсилювач. У випадку з MAX98357A цей процес спрощується, оскільки модуль приймає аудіодані безпосередньо у цифровому вигляді через I2S і формує вихідний сигнал для динаміка [16]. Це зменшує кількість компонентів у вузлі, спрощує схему підключення та робить пристрій компактнішим.

У складі вузла мовлення MAX98357A виконує такі основні функції:

- приймає цифровий аудіопотік від ESP32;
- забезпечує роботу з аудіоданими через інтерфейс I2S;
- формує сигнал для підключеного динаміка;
- підсилює аудіосигнал до рівня, достатнього для невеликої зони мовлення;
- спрощує побудову апаратної частини вузла;

- дозволяє реалізувати компактний мережевий пристрій відтворення.

Інтерфейс I2S передбачає використання кількох основних сигнальних ліній:

- BCLK – призначена для задання тактового сигналу передавання бітів;
- LRC або WS – використовується для визначення аудіоканалу.
- DIN — передає самі цифрові аудіодані від мікроконтролера до підсилювача.

Крім цих ліній, модуль потребує підключення живлення та спільної землі.

Для практичної реалізації вузла необхідно правильно підключити контакти ESP32 і MAX98357A [17-19]. Один із варіантів підключення наведено в табл. 2.4.

Таблиця 2.4 – Схема підключення підсилювача MAX98357A до ESP32

Контакт ESP32	Контакт MAX98357A	Призначення з'єднання
VIN / 5V	VIN	Живлення модуля підсилювача
GND	GND	Спільна земля схеми
GPIO27	BCLK	Тактовий сигнал інтерфейсу I2S
GPIO25	LRC / WS	Сигнал вибору аудіоканалу
GPIO26	DIN	Передавання цифрових аудіоданих

Наведений варіант підключення не є єдино можливим, оскільки ESP32 дозволяє програмно призначати різні GPIO для роботи з I2S. Під час вибору MAX98357A важливими є не стільки його максимальні аудіотехнічні характеристики, скільки його відповідність задачі локального мовлення [18].

Основна вимога до вузла полягає в розбірливому відтворенні голосових повідомлень. Для цього не потрібна складна стереосистема або високоякісне музичне відтворення. Достатньо забезпечити стабільну роботу, прийнятний рівень гучності та зрозуміле звучання в межах невеликої зони.

Під час використання модуля МАХ98357А слід враховувати кілька практичних моментів. Насамперед потрібно забезпечити спільну землю для ESP32 і підсилювача, оскільки без цього сигнальні лінії I2S можуть працювати некоректно. Також необхідно використовувати стабільне живлення, особливо якщо динамік працює на підвищеній гучності. Просідання напруги може впливати не лише на якість звуку, а й на стабільність роботи самого мікроконтролера.

У системі локального мовлення МАХ98357А також зручний тим, що кожен IoT-вузол може мати власний підсилювач і динамік. Це означає, що аудіовихід не формується централізовано на сервері або в одному спільному підсилювачі, а розміщується безпосередньо в зоні мовлення. Такий підхід відповідає розподіленій архітектурі системи: сервер лише керує трансляціями, а кожен вузол самостійно виконує відтворення.

З погляду експлуатації це має кілька переваг. По-перше, несправність одного вузла не обов'язково зупиняє роботу всієї системи. По-друге, різні вузли можуть мати різні рівні гучності залежно від розміру приміщення. По-третє, систему можна поступово масштабувати, додаючи нові вузли з власними підсилювачами та динаміками [18].

Отже, I2S-підсилювач МАХ98357А є доцільним компонентом для апаратної реалізації IoT-вузла локального мовлення. Він забезпечує зв'язок між цифровою частиною системи, представленою мікроконтролером ESP32, і виконавчим звуковим елементом – динаміком. Його використання дозволяє спростити аудіотракт, зменшити кількість зовнішніх компонентів і реалізувати компактний вузол, здатний приймати команди з IP-мережі та відтворювати аудіоповідомлення у визначеній зоні.

## 2.4 Вибір динаміка та параметрів аудіовиходу

Динамік є кінцевим виконавчим елементом IoT-вузла мовлення, оскільки саме він перетворює електричний сигнал, сформований аудіопідсилювачем, у звукові коливання. Якщо ESP32 відповідає за приймання команд і мережеву

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

взаємодію, а МАХ98357А – за формування підсиленого аудіосигналу, то динамік забезпечує безпосереднє відтворення повідомлення у вибраній зоні. Тому правильний вибір динаміка впливає на практичну придатність вузла мовлення, розбірливість голосових повідомлень і стабільність роботи апаратної частини.

У розроблюваній системі локального мовлення основним типом аудіоінформації є службові та інформаційні голосові повідомлення. Це можуть бути оголошення для студентів, повідомлення про початок або завершення занять, попередження, інструкції або екстрені звернення. Отже, для системи не є критичною висока якість музичного відтворення, широкий частотний діапазон або стереоефект. Важливішими є достатня гучність, чіткість мовлення, надійність і простота підключення.

У роботі доцільно використовувати малогабаритний динамік потужністю 3–5 Вт з опором 4 або 8 Ом. Його можна безпосередньо підключити до виходу підсилювача МАХ98357А, за умови дотримання допустимих параметрів навантаження та живлення.

Основні вимоги до динаміка для IoT-вузла локального мовлення наведено в табл. 2.5.

Таблиця 2.5 – Вимоги до динаміка IoT-вузла мовлення

Параметр	Рекомендоване значення / вимога	Обґрунтування
Тип динаміка	Малогабаритний широкосмуговий	Підходить для відтворення голосових повідомлень
Потужність	3–5 Вт	Достатньо для невеликої аудиторії або лабораторії
Опір	4 або 8 Ом	Сумісний з типовими малопотужними аудіопідсилювачами
Кількість каналів	Один канал, моно	Для службового мовлення стереовідтворення не є обов'язковим
Призначення	Голосові повідомлення	Основна вимога — розбірливість, а не музична якість
Розміщення	У корпусі вузла або окремо на стіні	Залежить від умов монтажу та зони покриття
Підключення	До виходу МАХ98357А	Забезпечує просту апаратну структуру вузла

Вибір опору динаміка має практичне значення для стабільної роботи вузла. Динаміки з опором 4 Ом, зазвичай, дозволяють отримати більшу вихідну потужність за однакової напруги живлення, однак вони створюють більше навантаження на підсилювач. Динаміки з опором 8 Ом можуть працювати з меншою вихідною потужністю, але є менш навантажувальними для підсилювального модуля. Для навчального прототипу обидва варіанти є прийнятними, але доцільно використовувати динамік, який відповідає рекомендаціям для конкретного модуля МАХ98357А [16].

Для реалізації аудіо виходу IoT-вузла обрано широкосмуговий динамік Adafruit Speaker.

Вибір саме цієї моделі пояснюється її відповідністю параметрам підсилювача. МАХ98357А використовується як малопотужний цифровий I2S-підсилювач для одного каналу відтворення. Тому застосування динаміка з опором 4 Ом і номінальною потужністю 3 Вт є технічно обґрунтованим. Потужніший динамік, наприклад на 5–10 Вт, не дав би суттєвої переваги без заміни підсилювача, оскільки фактична вихідна потужність усе одно обмежувалася б можливостями МАХ98357А та джерела живлення.

Основні характеристики обраного динаміка наведено в табл. 2.6, а його зовнішній вигляд показано на рис. 2.5.

Таблиця 2.6 – Характеристики динаміка Adafruit Speaker

Параметр	Значення
Модель	Adafruit Speaker 3" Diameter
Опір	4 Ом
Номінальна потужність	3 Вт
Розміри	77,8 × 77,8 × 25,49 мм
Маса	50,48 г
Тип використання	Відтворення голосових аудіоповідомлень
Призначення у системі	Динамік IoT-вузла локального мовлення



Рисунок 2.5 – Adafruit Speaker

У межах розроблюваної системи до одного підсилювача MAX98357A підключається один динамік. Це пов'язано з тим, що модуль MAX98357A є монофонічним підсилювачем і має один вихід на динамік – SPK+ та SPK-. Така схема є простою та зручною, оскільки кожен IoT-вузол можна розглядати як окрему автономну точку мовлення.

Отже, для IoT-вузла локального мовлення обрано динамік Adafruit Speaker, його параметри узгоджуються з використанням підсилювача MAX98357A, а потужності 3 Вт достатньо для відтворення голосових повідомлень у невеликій зоні. У межах кваліфікаційної роботи один підсилювач MAX98357A працює з одним динаміком, а масштабування системи до більших приміщень доцільно виконувати шляхом додавання нових IoT-вузлів у відповідні зони мовлення.

## 2.5 Схема підключення компонентів IoT-вузла

Після вибору основних апаратних компонентів необхідно визначити схему їх з'єднання в межах одного IoT-вузла мовлення. Такий вузол є автономною точкою відтворення аудіоповідомлень, яка отримує команди через IP-мережу та забезпечує озвучення певної зони. До його складу входять мікроконтролер ESP32, цифровий I2S-підсилювач MAX98357A, динамік Adafruit Speaker 3", а також джерело живлення.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

У розроблюваній системі ESP32 є центральним елементом вузла. Підсилювач MAX98357A, у свою чергу, приймає аудіодані через інтерфейс I2S і формує вихідний сигнал для динаміка. Динамік підключається безпосередньо до виходу підсилювача та виконує фізичне відтворення звуку.

Загальну схему підключення компонентів IoT-вузла мовлення можна подати так, як показано на рис. 2.6.

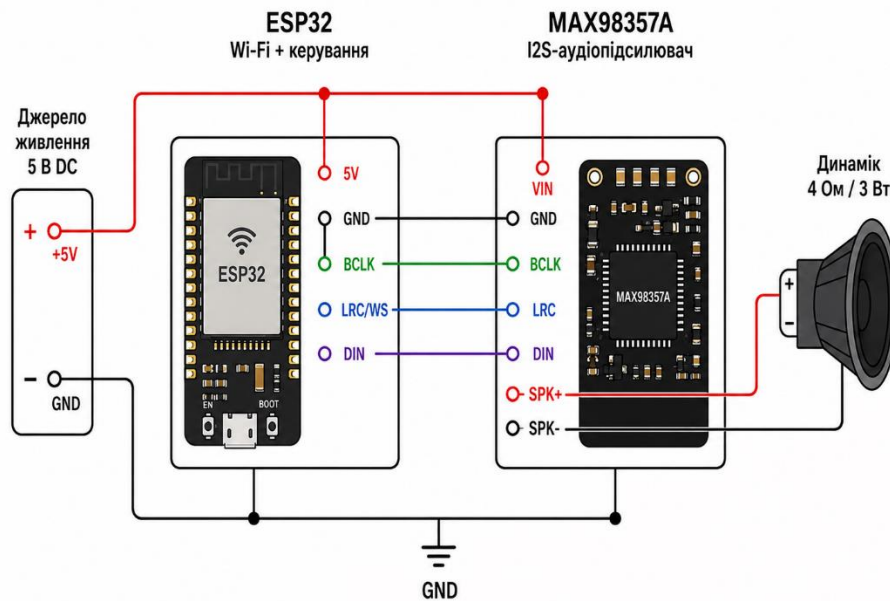


Рисунок 2.6 – Схема підключення компонентів IoT-вузла системи локального мовлення

У цій схемі всі компоненти мають спільну електричну основу через лінію GND. Це обов'язкова умова для правильної роботи цифрових сигналів між ESP32 і MAX98357A.

Окрему увагу потрібно приділити живленню компонентів вузла. ESP32 може житися через роз'єм micro-USB або через контакт VIN / 5V. У прототипі зручно використовувати джерело живлення постійного струму 5 В, яке забезпечує роботу мікроконтролера та підсилювача. Підсилювач MAX98357A також може житися від 5 В, що спрощує структуру живлення вузла.

Загальний принцип живлення компонентів IoT-вузла наведено в табл. 2.7.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Таблиця 2.7 – Організація живлення IoT-вузла

Компонент	Напруга живлення	Примітка
ESP32	5 В через micro-USB або VIN	На платі формується внутрішня напруга 3,3 В
MAX98357A	5 В	Живлення підсилювача від спільного джерела
Динамік	Отримує сигнал від підсилювача	Окремого живлення не потребує
Спільна земля	GND	Обов'язкова для ESP32 і MAX98357A

Використання одного джерела живлення для ESP32 і MAX98357A робить апаратну структуру вузла простішою, однак потребує достатнього запасу за струмом. Під час відтворення аудіо підсилювач може створювати пікове навантаження, тому блок живлення має бути стабільним. Тому тут варто використовувати джерело живлення 5 В з вихідним струмом не менше 1 А, а краще – 1,5–2 А, щоб уникнути просідання напруги під час роботи динаміка. Окрім цього, у схему вузла можна також помістити додаткові елементи індикації. Наприклад, до одного з GPIO ESP32 можна підключити світлодіод, який показуватиме стан вузла: підключення до мережі, очікування команди, відтворення аудіо або помилку. Такий елемент не є обов'язковим для базової роботи системи, але він корисний під час налагодження системи локального мовлення. Приклад додаткового підключення індикатора наведено в табл. 2.7.

Таблиця 2.7 – Додаткове підключення індикатора стану вузла

Вивід ESP32	Елемент	Призначення
GPIO2	Світлодіод через резистор	Індикація стану IoT-вузла
GND	Катод світлодіода	Завершення електричного кола

У такому варіанті світлодіод може світитися під час активної роботи вузла або блимати в різних режимах. Наприклад, повільне блимання може означати очікування підключення до Wi-Fi, постійне світіння – успішне підключення, а швидке блимання – отримання команди або відтворення повідомлення.

Монтаж компонентів системи може виконуватися шляхом їх з'єднання за допомогою макетної плати. У разі подальшого розвитку системи доцільно розробити друковану плату або компактний монтажний модуль, який об'єднає ESP32, підсилювач, роз'єм живлення, клеми динаміка та індикатори стану.

## 2.6 Структура мережевого підключення та модель зон мовлення

У структурі мережевого підключення, як було зазначено раніше, передбачаються такі основні компоненти:

- сервер керування локальним мовленням;
- MQTT-брокер для передавання команд;
- HTTP-сервер для доступу до аудіофайлів;
- Wi-Fi-маршрутизатор або локальний комутатор;
- IoT-вузли мовлення на базі ESP32;
- веб-інтерфейс адміністратора;
- локальна IP-адресація пристроїв.

У межах прототипу сервер керування, MQTT-брокер і HTTP-сервер можуть працювати на одному комп'ютері. Це спрощує налаштування системи та дозволяє зосередитися на перевірці основної логіки: виборі повідомлення, виборі зони, передаванні команди та відтворенні аудіо на вузлі.

Приклад структури мережевого підключення системи наведено на рис. 2.7.

У запропонованій схемі всі пристрої перебувають в одній локальній мережі. Сервер має постійну IP-адресу, за якою IoT-вузли можуть підключатися до MQTT-брокера та HTTP-сервера. Для прототипу можна використати приватний діапазон адрес, наприклад 192.168.1.0/24. Приклад адресації компонентів системи наведено в табл. 2.8.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.7 – Структура мережевого підключення та зонування системи локального мовлення

Таблиця 2.8 – IP-адресація компонентів системи з 3-ма IoT-вузлами

Пристрій	Приклад IP-адреси	Призначення
Сервер керування	192.168.1.10	Веб-інтерфейс, MQTT-брокер, HTTP-сервер
Wi-Fi-маршрутизатор	192.168.1.1	Організація локальної мережі
IoT-вузол 1	192.168.1.21	Відтворення повідомлень в аудиторії
IoT-вузол 2	192.168.1.22	Відтворення повідомлень у коридорі
IoT-вузол 3	192.168.1.23	Відтворення повідомлень на поверсі
Комп'ютер адміністратора	192.168.1.50	Доступ до панелі керування

Для стабільної роботи системи бажано, щоб сервер мав статичну IP-адресу. Це пояснюється тим, що IoT-вузли повинні знати адресу MQTT-брокера та HTTP-сервера. Якщо адреса сервера зміниться, вузли можуть втратити можливість отримувати команди або завантажувати аудіофайли. Адреси IoT-вузлів можуть призначатися автоматично через DHCP, однак для зручності тестування їх також можна закріпити в налаштуваннях маршрутизатора [21].

У системі використовується два основні типи мережевого обміну. Перший тип – передавання команд керування через MQTT. Через цей протокол сервер

надсилає вузлам команди запуску, зупинки, зміни гучності або встановлення пріоритету повідомлення. Другий тип – доступ до аудіофайлів через HTTP. У цьому випадку сервер передає у команді посилання на аудіофайл, а IoT-вузол самостійно звертається до HTTP-сервера та отримує файл для відтворення.

Основні типи мережевого обміну наведено в табл. 2.9.

Таблиця 2.9 – Основні типи мережевого обміну в системі

Напрямок обміну	Протокол	Зміст передавання
Сервер → IoT-вузол	MQTT	Команди play, stop, volume, priority
IoT-вузол → сервер	MQTT	Статус вузла, підтвердження виконання, повідомлення про помилку
IoT-вузол → HTTP-сервер	HTTP	Запит на отримання аудіофайлу
HTTP-сервер → IoT-вузол	HTTP	Передавання аудіофайлу
Адміністратор → сервер	HTTP / вебінтерфейс	Вибір повідомлення, зони та пріоритету

Поділ мережевої взаємодії на MQTT і HTTP робить систему простішою для реалізації. MQTT використовується для коротких службових повідомлень, які повинні швидко надходити до вузлів. HTTP використовується для передавання аудіофайлів, які мають більший обсяг. Завдяки цьому команди керування не перевантажуються аудіоданими, а аудіофайли можуть зберігатися на сервері окремо [22].

Мережеве підключення безпосередньо пов'язане з моделлю зон мовлення. У цій роботі під зоною мовлення розуміється окреме приміщення або група приміщень, у яких потрібно відтворити аудіоповідомлення. Зоною може бути аудиторія, коридор, поверх, корпус або вся система. Кожна зона містить один або кілька IoT-вузлів, а один вузол може належати одночасно до кількох зон.

Наприклад, вузол, встановлений в аудиторії, може належати до зони аудиторії, зони поверху, зони корпусу та загальної зони всієї системи. Завдяки

цьому адміністратор може запустити повідомлення як для окремого приміщення, так і для ширшої групи вузлів без зміни апаратної схеми підключення. Приклад логічної моделі зон мовлення наведено в табл. 2.10.

Таблиця 2.10 – Приклад логічної моделі зон мовлення

Ідентифікатор зони	Назва зони	Склад зони	Призначення
zone_room	Аудиторія	node_01	Повідомлення в окреме навчальне приміщення
zone_corridor	Коридор	node_02	Оповіщення у спільному проході
zone_floor	Поверх	node_01, node_02, node_03	Трансляція для групи приміщень одного поверху
zone_building	Корпус	node_01, node_02, node_03, node_04	Повідомлення в межах одного корпусу
zone_all	Усі вузли	Усі зареєстровані вузли	Загальне або екстрене повідомлення

Для адресації вузлів і зон у MQTT доцільно використовувати зрозумілу структуру тем. Індивідуальні команди можуть надсилатися у теми конкретних вузлів, а групові – у теми зон. Це дозволяє серверу працювати не лише з IP-адресами пристроїв, а й з логічними ідентифікаторами приміщень або груп вузлів.

Таблиця 2.11 – Відповідність зон мовлення та MQTT-тем

Об'єкт керування	MQTT-тема	Призначення
Вузол node_01	campus/audio/node_01/cmd	Команда лише для першого вузла
Вузол node_02	campus/audio/node_02/cmd	Команда лише для другого вузла
Зона аудиторії	campus/audio/zone/room/cmd	Команда для аудиторії
Зона коридору	campus/audio/zone/corridor/cmd	Команда для коридору
Зона поверху	campus/audio/zone/floor/cmd	Команда для всіх вузлів поверху
Загальна зона	campus/audio/all/cmd	Команда для всіх вузлів системи

Отже, структура мережевого підключення та модель зон мовлення формують єдиний логічний рівень системи. IP-мережа забезпечує обмін даними між сервером і IoT-вузлами, а модель зон визначає, яким саме вузлам потрібно передати повідомлення. Завдяки використанню MQTT, HTTP і логічного зонування система може передавати повідомлення в окрему аудиторію, коридор, на поверх, у корпус або на всі вузли одночасно.

					<i>КС КРБ 123.147.00.00 ПЗ</i>	Арк.
						46
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СКЛАДОВОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ЛОКАЛЬНОГО МОВЛЕННЯ

### 3.1 Проектування архітектури програмного забезпечення системи локального мовлення

Програмне забезпечення комп'ютерної системи локального мовлення з використанням IoT-вузлів та IP-мережі призначене для організації централізованого керування аудіоповідомленнями.

В основі архітектури програмного забезпечення лежить принцип розподіленості компонентів за функціональним призначенням. Це означає, що окремі функції системи виконуються різними програмними компонентами, які взаємодіють між собою за допомогою мережевих протоколів. Центральна частина системи відповідає за формування, збереження та передавання повідомлень, тоді як IoT-вузли виконують приймання команд і безпосереднє відтворення аудіосигналу через підключений підсилювач та динамік.

До складу програмного забезпечення системи входять такі основні компоненти:

- веб-інтерфейс адміністратора;
- сервер керування повідомленнями;
- MQTT-брокер для передавання керуючих команд;
- HTTP-сервер для передавання або надання доступу до аудіофайлів;
- програмне забезпечення IoT-вузлів на базі ESP32;
- модуль журналювання подій і контролю стану вузлів.

Схематично архітектуру програмного забезпечення системи локального мовлення можна зобразити, як показано на рис. 3.1.

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.147.00.00 ПЗ			
Розроб.		Бойчук Н.В.			Реалізація програмної складової комп'ютерної системи локального мовлення	Літ.	Арк.	Аркушів
Перевірив.		Луцків А.М.					47	
Реценз.		Литвиненко Я.В.				ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Луцик Н.С.						
Затверд.		Осухівська Г.М.						

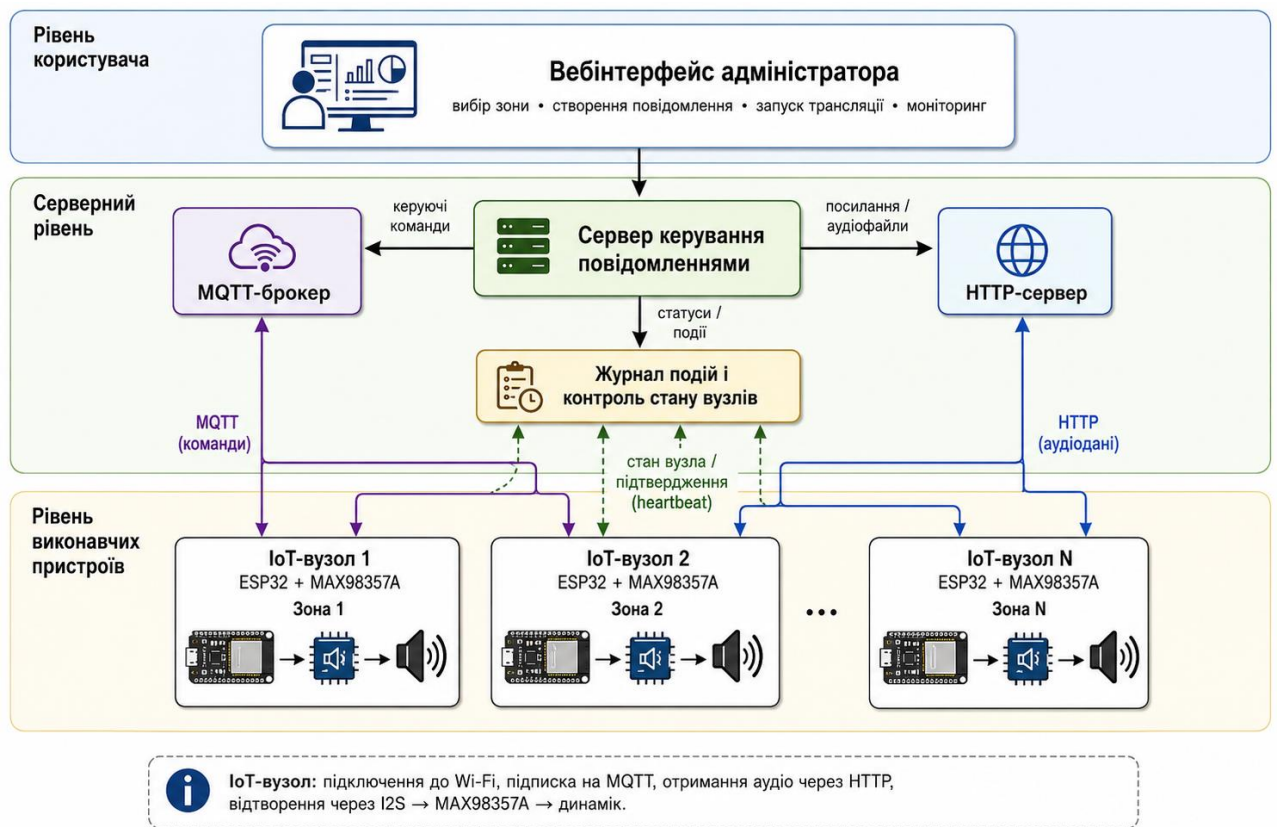


Рисунок 3.1 – Архітектура ПЗ системи локального мовлення

Веб-інтерфейс адміністратора є верхнім рівнем програмної системи. Через нього користувач має змогу обирати зону мовлення, формувати текстове або аудіоповідомлення, запускати трансляцію, переглядати стан вузлів і контролювати виконання команд. Такий підхід робить систему зручною для використання, оскільки адміністратору не потрібно безпосередньо взаємодіяти з окремими IoT-пристроями. Усі дії виконуються через єдину панель керування.

Сервер керування повідомленнями виконує роль центрального програмного елемента системи. Він приймає запити від веб-інтерфейсу, обробляє їх, визначає цільові зони мовлення та формує відповідні команди для IoT-вузлів [22]. Наприклад, якщо адміністратор обирає трансляцію повідомлення в аудиторію, сервер визначає вузол, який відповідає цій зоні, і надсилає йому команду на відтворення. Якщо потрібно виконати групову трансляцію, команда може бути надіслана одразу кільком вузлам.

Для передавання керуючих команд у системі використовується протокол MQTT, оскільки він забезпечує легкий механізм обміну повідомленнями між сервером і віддаленими пристроями [23].

MQTT працює за принципом публікації та підписки: сервер публікує повідомлення у відповідну тему, а IoT-вузли, підписані на цю тему, отримують команду. Наприклад, для окремої аудиторії може використовуватися тема `campus/audio/zone1`, а для коридору – `campus/audio/zone2`.

У запропонованій архітектурі MQTT не застосовується для передавання самого аудіопотоку. Натомість через MQTT передається команда, яка містить інформацію про те, яке повідомлення потрібно відтворити, його пріоритет і посилання на аудіофайл. Після отримання такої команди IoT-вузол може звернутися до HTTP-сервера та отримати необхідні аудіодані.

Загальна логіка роботи програмного забезпечення IoT-вузла зображена у додатку Б і полягає в тому, що після ввімкнення живлення пристрій автоматично підключається до локальної Wi-Fi-мережі, отримує IP-адресу та підключається до MQTT-брокера. Після цього вузол переходить у режим очікування команд. Коли сервер надсилає команду на відтворення, ESP32 перевіряє її параметри, визначає тип повідомлення, отримує аудіодані та запускає відтворення через підсилювач.

Важливою частиною програмної структури є механізм адресації зон. Кожен IoT-вузол має власну IP-адресу та логічну прив'язку до певної зони: аудиторії, коридору, поверху або іншої частини приміщення. Це дає змогу реалізувати як адресне мовлення, коли повідомлення надсилається лише в одну зону, так і групове мовлення, коли трансляція виконується одночасно в декількох зонах [21-23].

Для підвищення надійності роботи системи передбачається контроль стану IoT-вузлів. Кожен вузол може періодично надсилати на сервер службові повідомлення про свій стан: підключення до мережі, готовність до відтворення, зайнятість, завершення трансляції або виникнення помилки. Сервер, у свою чергу, може відображати цю інформацію у веб-інтерфейсі адміністратора. Це дозволяє оперативно виявляти несправності або відсутність зв'язку з окремими

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

вузлами. Окрему роль у програмному забезпеченні відіграє журналювання подій. До журналу можуть записуватися дії адміністратора, час запуску повідомлення, обрана зона мовлення, статус доставки команди, початок і завершення відтворення, а також повідомлення про помилки.

Таким чином, програмне забезпечення системи має модульну структуру, що спрощує розроблення, тестування та подальше розширення. У разі потреби до системи можна додати нові вузли мовлення, змінити логіку адресації зон або доповнити веб-інтерфейс новими функціями без повної перебудови архітектури.

### 3.2 Програмна реалізація серверної частини

#### 3.2.1 Реалізація HTTP-сервера для роботи з аудіофайлами

У запропонованій архітектурі HTTP-сервер не виконує безпосереднє керування вузлами мовлення, а відповідає саме за передавання аудіоданих. Керуючі команди, наприклад команда запуску трансляції, передаються через MQTT, тоді як сам аудіофайл завантажується IoT-вузлом за HTTP-посиланням.

Сервер має локальну IP-адресу – 192.168.1.10, а аудіофайли зберігаються в окремому каталозі – /server/audio/.

Після завантаження файлу адміністратором сервер формує для нього посилання такого вигляду: «[http://192.168.1.10:8000/audio/msg\\_001.wav](http://192.168.1.10:8000/audio/msg_001.wav)». Саме це посилання надалі передається IoT-вузлу у складі MQTT-команди.

Роботу HTTP-сервера можна поділити на два основні сценарії:

- завантаження аудіофайлу адміністратором;
- отримання аудіофайлу IoT-вузлом.

У першому випадку адміністратор через веб-інтерфейс вибирає аудіофайл, після чого він передається на сервер. Сервер перевіряє формат файлу, формує унікальну назву, зберігає файл у каталозі аудіоданих і повертає URL-адресу для подальшого використання.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

У другому випадку IoT-вузол отримує MQTT-команду, у якій міститься посилання на аудіофайл. Після цього ESP32 звертається до HTTP-сервера, отримує аудіодані та передає їх на відтворення через інтерфейс I2S.

Графічно алгоритм роботи HTTP-сервера можна подати у вигляді UML діаграми активностей, як показано на рис. 3.2.

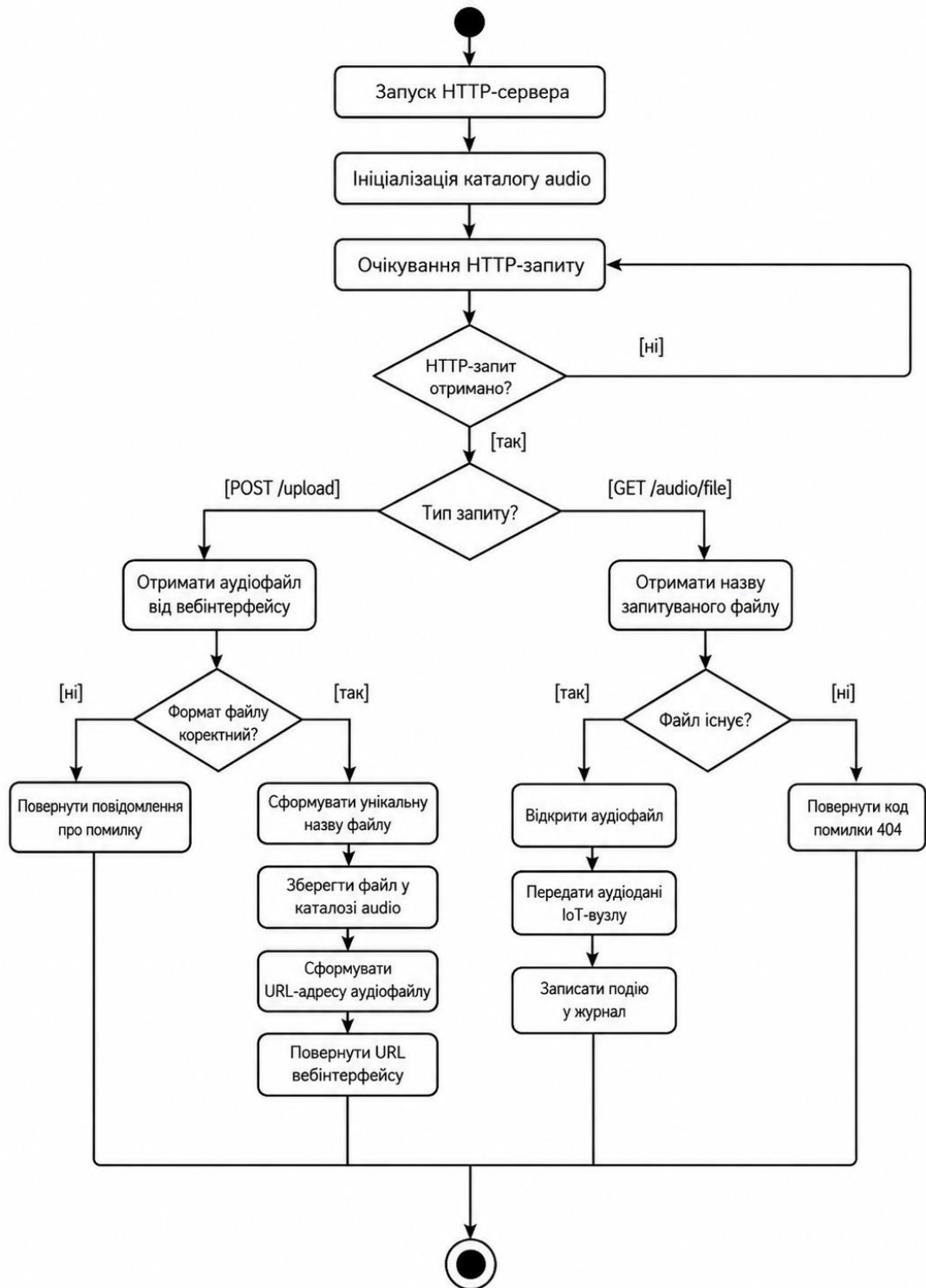


Рисунок 3.2 – UML-діаграма активностей HTTP-сервера

Змн.	Арк.	№ докум.	Підпис	Дата

Під час реалізації HTTP-сервера виділено набір основних функцій, які представлені у табл. 3.1. Це спрощує структуру програми та дозволяє окремо реалізувати перевірку файлів, їх збереження, формування URL-адрес і передавання аудіоданих.

Таблиця 3.1 – Основні функції HTTP-сервера у системи мовлення

Функція	Призначення
upload_audio()	Приймає аудіофайл від веб-інтерфейсу адміністратора
validate_audio_file()	Перевіряє формат і допустимість аудіофайлу
generate_filename()	Формує унікальну назву аудіофайлу
save_audio_file()	Зберігає аудіофайл у каталозі сервера
generate_audio_url()	Формує HTTP-посилання на збережений файл
get_audio_file()	Передає аудіофайл IoT-вузлу за запитом
log_event()	Записує службові події в журнал

Для реалізації HTTP-сервера використано мову програмування Python і мікрофреймворк Flask. Фрагмент програмного коду підключення бібліотек та опису мережевих параметрів HTTP-сервера показано на рис. 3.3.

```

from flask import Flask, request, send_from_directory, jsonify
import os
import uuid
from datetime import datetime

app = Flask(__name__)

SERVER_IP = "192.168.1.10"
SERVER_PORT = 8000
AUDIO_DIR = "audio"
ALLOWED_EXTENSIONS = {".wav"}

os.makedirs(AUDIO_DIR, exist_ok=True)

```

Рисунок 3.3 – Імпорт бібліотек та опис HTTP-сервера

Маршрут `/upload` використовується веб-інтерфейсом адміністратора. Після успішного завантаження сервер повертає URL-адресу аудіофайлу. Далі ця адреса може бути передана до MQTT-модуля для формування команди відтворення.

Маршрут `/audio/<filename>` використовується IoT-вузлом. Після отримання MQTT-команди ESP32 звертається до цієї адреси та отримує аудіодані для відтворення.

Маршрут `/health` може використовуватися для перевірки того, чи працює сервер. Це зручно під час тестування системи та діагностики проблем у локальній мережі.

Фрагмент коду, що реалізує завантаження і перевірку аудіофайлів представлено на рис. 3.4.

```
@app.route("/upload", methods=["POST"])
def upload_audio():
    """
    Обробник завантаження аудіофайлу від вебінтерфейсу.
    """
    if "audio" not in request.files:
        return jsonify({
            "status": "error",
            "message": "Аудіофайл не передано"
        }), 400

    audio_file = request.files["audio"]

    if audio_file.filename == "":
        return jsonify({
            "status": "error",
            "message": "Назва файлу порожня"
        }), 400

    if not validate_audio_file(audio_file.filename):
        return jsonify({
            "status": "error",
            "message": "Підтримується лише формат WAV"
        }), 400
```

Рисунок 3.4 – Завантаження аудіофайлу на HTTP-сервер

Після успішного завантаження аудіофайлу сервер повертає відповідь у форматі JSON, структуру якого показано на рис. 3.5.

```
{
  "status": "ok",
  "filename": "7f3b8f41-9a51-4f71-b1f4-1e7e25c8c2ab.wav",
  "audio_url": "http://192.168.1.10:8000/audio/7f3b8f41-9a51-4f71-b1f4-1e7e25c8c2ab.wav"
}
```

Рисунок 3.5 – Відповідь сервера на успішне завантаження аудіофайлу

Поле `audio_url` є ключовим, оскільки саме воно використовується під час формування MQTT-команди для IoT-вузла.

Після запуску HTTP-сервер створює каталог для зберігання аудіофайлів, якщо його ще не існує. Далі переходить у режим очікування запитів. Якщо надходить запит на завантаження файлу, програма перевіряє наявність файлу в запиті, його назву та формат. У разі успішної перевірки файл зберігається з унікальним іменем, після чого формується URL-адреса для доступу до нього. Якщо запит надходить від IoT-вузла, сервер перевіряє, чи існує відповідний файл у каталозі. Якщо файл знайдено, він передається клієнту. Якщо файл відсутній, сервер повертає повідомлення про помилку. Усі важливі події можуть записуватися в журнал, що полегшує діагностику роботи системи. Перевагою такої реалізації є простота та зрозумілість. HTTP-сервер не перевантажується додатковою логікою керування трансляціями, а виконує окрему чітко визначену функцію – зберігання та передавання аудіофайлів. Це відповідає модульному принципу побудови програмного забезпечення.

### 3.2.2 Реалізація MQTT-брокера та модуля передавання керуючих команд

Для роботи MQTT-брокера використовується стандартний порт: 1883. У системі локального мовлення використовується брокер Mosquitto [23]. Серверна програма підключається до MQTT-брокера як клієнт і публікує повідомлення у відповідні теми. IoT-вузли на базі ESP32 також підключаються до брокера та підписуються на теми, які відповідають їхнім зонам мовлення.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

MQTT використовується для передавання таких типів повідомлень:

- команда запуску відтворення аудіофайлу;
- команда зупинки поточного відтворення;
- команда аварійного оповіщення;
- повідомлення від IoT-вузлів щодо статусу команди;
- повідомлення про помилки або втрату зв'язку.

Для організації адресації зон у системі використовується ієрархічна структура MQTT-тем. Кожна зона має власну тему, а також передбачені службові теми для групового та аварійного оповіщення, як було раніше показано у п. 2.6.

MQTT-команда передається у форматі JSON. Такий формат є зручним, оскільки він зрозумілий як для серверної частини, так і для мікроконтролера ESP32. Приклад команди запуску відтворення аудіофайлу проілюстровано на рис. 3.6, а призначення команд – в табл. 3.2.

```
{
  "command": "play",
  "message_id": "msg_001",
  "audio_url": "http://192.168.1.10:8000/audio/msg_001.wav",
  "zone": "zone1",
  "priority": "normal"
}
```

Рисунок 3.6 – Команда запуску відтворення аудіофайлу

Таблиця 3.2 – Призначення команд у структурі JSON-повідомлення

Поле	Призначення
command	Тип команди, наприклад play, stop, emergency
message_id	Унікальний ідентифікатор повідомлення
audio_url	HTTP-посилання на аудіофайл
zone	Ідентифікатор зони мовлення
priority	Пріоритет повідомлення: звичайний або аварійний

Для аварійного повідомлення команда має вигляд, як показано на рис. 3.7.

```

{
  "command": "play",
  "message_id": "emergency_001",
  "audio_url": "http://192.168.1.10:8000/audio/emergency_001.wav",
  "zone": "all",
  "priority": "emergency"
}

```

Рисунок 3.7 – Команда аварійного (екстренного) повідомлення

У випадку екстреного повідомлення IoT-вузол повинен зупинити поточне відтворення звичайного повідомлення та перейти до відтворення аварійного аудіофайлу. Графічно алгоритм роботи MQTT-модуля подано на рис. 3.8.

MQTT-модуль серверної частини виконує дві основні функції. Перша полягає у формуванні та надсиланні команд до IoT-вузлів, а друга – у прийманні статусних повідомлень від вузлів.

Для реалізації MQTT-модуля визначено окремі функції, які відповідають за підключення до брокера, формування команд, публікацію повідомлень і обробку статусів від вузлів. У табл. 3.3 представлено необхідні функції і їх призначення.

Таблиця 3.3 – Функції MQTT-модуля

Функція	Призначення
connect_mqtt()	Підключення серверної програми до MQTT-брокера
build_play_command()	Формування JSON-команди для запуску відтворення
get_topic_by_zone()	Визначення MQTT-теми відповідно до вибраної зони
publish_command()	Публікація команди в MQTT-брокер
subscribe_status_topics()	Підписка на теми статусів IoT-вузлів
on_node_status()	Обробка статусного повідомлення від вузла
update_node_status()	Оновлення поточного стану вузла
log_event()	Запис події у журнал системи

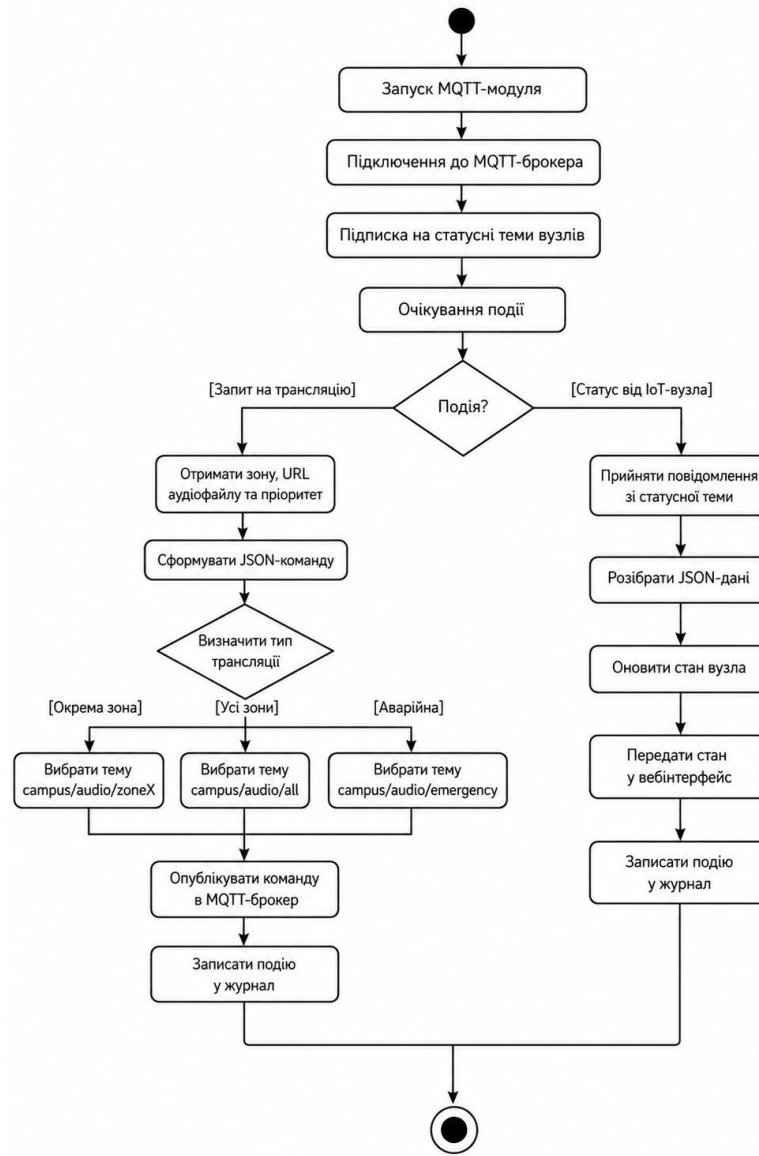


Рисунок 3.8 – Алгоритм роботи MQTT-модуля

Для реалізації MQTT-модуля використано бібліотеку paho-mqtt. На рис. 3.9 наведено підключення необхідних бібліотек та оголошення параметрів MQTT-брокера.

```
import json
from datetime import datetime
import paho.mqtt.client as mqtt

MQTT_BROKER = "192.168.1.10"
MQTT_PORT = 1883

node_status = {}
```

Рисунок 3.9 – Оголошення MQTT-брокера

Функція підключення до MQTT-брокера та функція визначення теми в залежності від зони та пріоритету повідомлення проілюстровані у лістингу на рис. 3.10.

```
def connect_mqtt():
    """
    Підключення до MQTT-брокера.
    """
    client = mqtt.Client()
    client.on_message = on_node_status
    client.connect(MQTT_BROKER, MQTT_PORT, 60)
    return client

def get_topic_by_zone(zone, priority="normal"):
    """
    Визначення MQTT-теми залежно від зони та пріоритету повідомлення.
    """
    if priority == "emergency":
        return "campus/audio/emergency"

    if zone == "all":
        return "campus/audio/all"

    return f"campus/audio/{zone}"
```

Рисунок 3.10 – Підключення до брокера та визначення теми

Якщо з веб-інтерфейсу надходить команда на трансляцію, сервер отримує параметри повідомлення: зону мовлення, URL аудіофайлу та пріоритет. На основі цих даних формується JSON-команда, після чого вона публікується у відповідну MQTT-тему.

Якщо трансляція призначена для однієї зони, команда надсилається у тему цієї зони. Якщо адміністратор обирає всі зони, використовується тема campus/audio/all. Якщо повідомлення має аварійний пріоритет, воно публікується у тему campus/audio/emergency. Паралельно MQTT-модуль приймає повідомлення про статус від IoT-вузлів. Це дозволяє серверу відстежувати, чи доступний вузол, чи він перебуває в режимі очікування, чи

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

виконує відтворення, чи завершив трансляцію. Отримані статуси відображаються у веб-інтерфейсі адміністратора.

Формування команди запуску на відтворення аудіо та публікації її у відповідну тему представлені на рис. 3.11.

```
def build_play_command(zone, message_id, audio_url, priority="normal"):
    """
    Формування JSON-команди для запуску відтворення.
    """
    command = {
        "command": "play",
        "message_id": message_id,
        "audio_url": audio_url,
        "zone": zone,
        "priority": priority
    }

    return json.dumps(command)

def publish_command(client, zone, message_id, audio_url, priority="normal"):
    """
    Публікація команди в MQTT-брокер.
    """
    topic = get_topic_by_zone(zone, priority)
    payload = build_play_command(zone, message_id, audio_url, priority)

    client.publish(topic, payload)

    log_event(f"MQTT-команду надіслано у тему {topic}")
```

Рисунок 3.11 – Функції запуску відтворення аудіо та публікації у тему

Отже, MQTT-брокер і MQTT-модуль серверної частини забезпечують передавання керуючих команд між сервером і IoT-вузлами. Використання MQTT дозволяє реалізувати адресне, групове та аварійне мовлення без прямого звернення до кожного пристрою за IP-адресою. Команди передаються у форматі JSON і містять лише службову інформацію, зокрема тип команди, зону, пріоритет і URL аудіофайлу.

### 3.2.3 Реалізація веб-інтерфейсу адміністратора

У запропонованій системі веб-інтерфейс працює як частина серверної програми. Основними функціями вебінтерфейсу є:

- вибір зони мовлення;

- завантаження аудіофайлу;
- вибір пріоритету повідомлення;
- запуск трансляції;
- перегляд стану IoT-вузлів;
- перегляд журналу подій;
- відображення результату виконання команди.

Загальна логіка роботи веб-інтерфейсу полягає в тому, що адміністратор відкриває сторінку керування, вибирає потрібні параметри трансляції та надсилає запит на сервер. Сервер перевіряє коректність введених даних, зберігає аудіофайл, формує URL-адресу для доступу до нього та передає команду через MQTT у відповідну тему.

Веб сторінка адміністратора повинна бути простою та зрозумілою, оскільки її основне завдання – швидкий запуск аудіотрансляції у визначену зону. Для цього доцільно передбачити одну головну сторінку з формою керування, таблицею стану вузлів і журналом подій. На рис. 3.12 показано макет та реалізацію даної веб-сторінки.

Після відкриття сторінки адміністратор бачить форму запуску трансляції. У полі вибору зони він може обрати конкретну зону, наприклад аудиторію або коридор, або запустити повідомлення для всіх зон одночасно. Далі адміністратор вибирає WAV-файл, який потрібно відтворити, та задає пріоритет повідомлення.

Після натискання кнопки «Запустити трансляцію» браузер надсилає POST-запит на маршрут /broadcast. Сервер приймає файл, зберігає його у каталозі audio, формує URL-адресу аудіофайлу та створює MQTT-команду. Після цього команда публікується у відповідну тему MQTT-брокера.

Паралельно у нижній частині веб сторінки відображається таблиця стану IoT-вузлів. У ній адміністратор може бачити IP-адресу кожного вузла, його логічну зону та поточний стан. Наприклад, стан ready означає, що вузол готовий до роботи, а стан playing – що вузол виконує відтворення аудіоповідомлення.

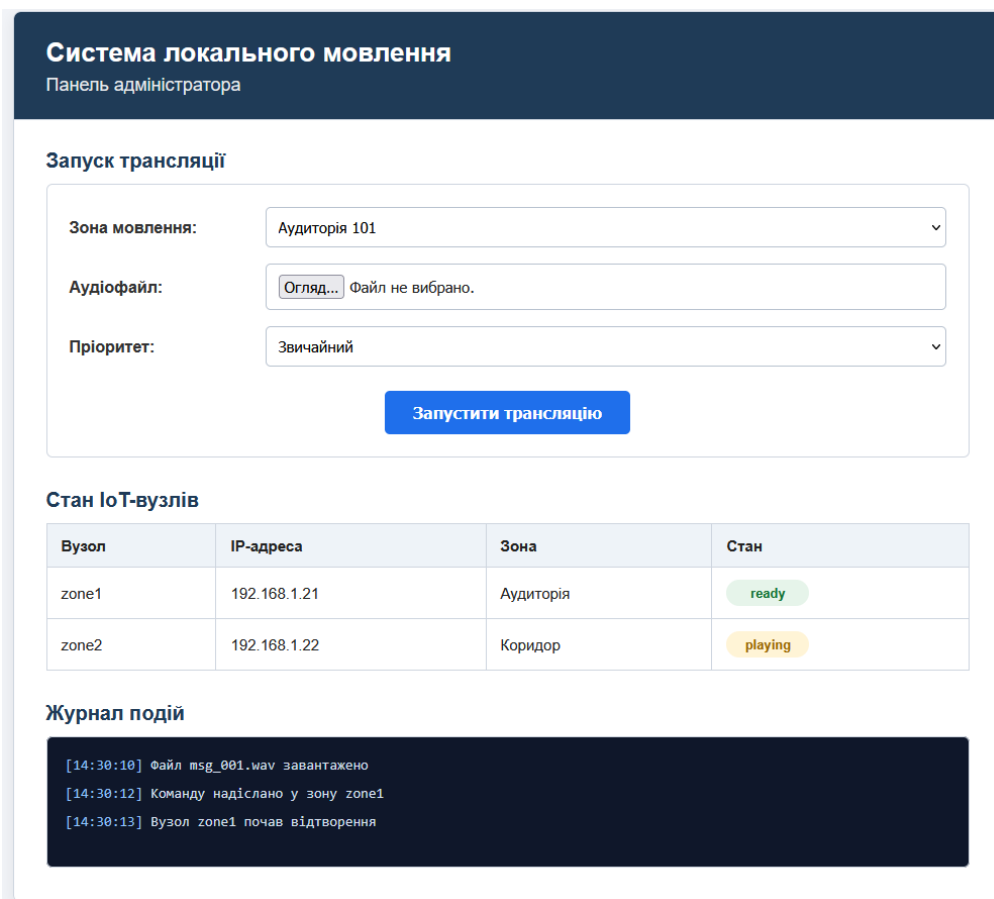


Рисунок 3.12 – Веб сторінка адміністратора системи локального мовлення

Журнал подій використовується для контролю виконання команд. У ньому можуть відображатися повідомлення про завантаження файлу, надсилання MQTT-команди, початок відтворення, завершення трансляції або виникнення помилки.

### 3.3 Програмна реалізація IoT-вузла на базі ESP32

ESP32 виконує роль контролера, який після ввімкнення підключається до мережі, встановлює з'єднання з MQTT-брокером і очікує команд від сервера.

Підключившись до Wi-Fi мережі та отримавши IP-адресу мікроконтролер підключається до MQTT-брокера за адресою сервера і підписується на MQTT-теми своєї зони. Тема `campus/audio/zone1` використовується для адресного повідомлення, `campus/audio/all` – для трансляції в усі зони, а `campus/audio/emergency` – для аварійного оповіщення.

Основні функції програми IoT-вузла реалізовано мовою програмування C++ та наведено в табл.3.4.

Таблиця 3.4 – Основні функції програми IoT-вузла

Функція	Призначення
connectWiFi()	Підключення ESP32 до Wi-Fi
connectMQTT()	Підключення до MQTT-брокера
subscribeTopics()	Підписка на MQTT-теми
mqttCallback()	Обробка отриманих команд
parseCommand()	Розбір JSON-команди
playAudioFromUrl()	Отримання та відтворення аудіофайлу
sendStatus()	Надсилання стану вузла на сервер

Фрагмент програми підключення ESP32 до Wi-Fi та MQTT-брокера показано на рис. 3.13.

```

#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "Campus_WiFi";
const char* password = "password123";
const char* mqtt_server = "192.168.1.10";
const int mqtt_port = 1883;

WiFiClient espClient;
PubSubClient client(espClient);

const char* node_id = "zone1";

void connectWiFi() {
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

void connectMQTT() {
  while (!client.connected()) {
    if (client.connect(node_id)) {
      client.subscribe("campus/audio/zone1");
      client.subscribe("campus/audio/all");
      client.subscribe("campus/audio/emergency");
    } else {
      delay(2000);
    }
  }
}

```

Рисунок 3.13 – Підключення до Wi-Fi та MQTT-брокера

Для обробки MQTT-команди використовується callback-функція. Вона викликається автоматично, коли вузол отримує повідомлення з теми, на яку він підписаний. На рис. 3.14 представлено програмну реалізацію функції при обробці MQTT-команди.

```
void mqttcallback(char* topic, byte* payload, unsigned int length) {
    String message = "";

    for (int i = 0; i < length; i++) {
        message += (char)payload[i];
    }

    if (message.indexOf("\command\":"play\") >= 0) {
        sendStatus("playing");

        // Тут виконується розбір JSON-команди
        // та отримання audio_url

        // playAudioFromUrl(audio_url);

        sendStatus("finished");
    }
}
```

Рисунок 3.14 – Функція обробки MQTT-команди

Функція передавання статусу (рис. 3.15) дозволяє серверній частині відстежувати стан кожного IoT-вузла. Статус надсилається у відповідну MQTT-тему, наприклад campus/audio/status/zone1.

```
void sendStatus(String status) {
    String payload = "{";
    payload += "\node_id\":"zone1\",";
    payload += "\ip\":" + WiFi.localIP().toString() + "\",";
    payload += "\status\":" + status + "\",";
    payload += "}";

    client.publish("campus/audio/status/zone1", payload.c_str());
}
```

Рисунок 3.15 – Функція передавання стану IoT-вузла

Програмна реалізація IoT-вузла забезпечує автоматичне підключення пристрою до мережі і виконання покладених на вузол функцій. Передавання повідомлень стану дозволяє серверу контролювати вузли і відображати цю інформацію у веб-інтерфейсі адміністратора.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Діяльність людини та її вплив на довкілля

В умовах науково-технічного прогресу значно ускладнились взаємовідносини суспільства з природою. Людина отримала можливість впливати на хід природних процесів, підкорила сили природи, почала опановувати майже всі доступні відновні і невідновні природні ресурси, але разом з тим забруднювати і руйнувати довкілля.

За оцінкою Всесвітньої організації охорони здоров'я (ВООЗ), із більш ніж 6 млн. відомих хімічних сполук практично використовується до 500 тис. сполук; із них біля 40 тис. мають шкідливі для людини властивості, а 12 тис. є токсичними [25].

До кінця ХХ століття початку ХХІ століття забруднення навколишнього середовища відходами, викидами, стічними водами всіх видів промислового виробництва, сільського господарства, комунального господарства міст набуло глобального характеру і поставило людство на грань екологічної катастрофи.

Втручання людини у природні процеси різко зростає і може спричиняти зміну режиму ґрунтових і підземних вод у цілих регіонах, поверхневого стоку, структури ґрунтів, інтенсифікацію ерозійних процесів, активізацію геохімічних та хімічних процесів у атмосфері, гідросфері та літосфері, зміни мікроклімату.

Сучасна діяльність, наприклад, будівництво гідротехнічних споруд, шахт, рудників, доріг, свердловин, водойм, дамб, деформація суші ядерними вибухами, будівництво гігантських міст, обводнення і озеленення пустель, та інші повсякденні аспекти діяльності людини, вже викликали значні видимі і приховані зміни довкілля.

					<b>КС КРБ 123.140.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		Бойчук Н.В.			Безпека життєдіяльності, основи охорони праці	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірив.</i>		Луцків А.М.					64	
<i>Консульт.</i>		Сенчишин В.С.				ТНТУ, каф. КС, гр. СІ-41		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затверд.</i>		Осухівська Г.М.						

В історичному плані виділяють декілька етапів зміни біосфери людством, які увінчались екологічними кризами та революціями, а саме:

- вплив людства на біосферу як звичайного біологічного виду;
- надінтенсивне полювання без змін екосистем у період становлення людства;
- зміни екосистем внаслідок процесів, що відбуваються природнім шляхом: випасання, посилення росту трав шляхом випалювання тощо;
- інтенсифікація впливу на природу шляхом розорювання ґрунтів та вирубування лісів;
- глобальні зміни всіх екологічних компонентів біосфери в цілому.

Вплив людини на біосферу зводиться до чотирьох головних форм:

- зміна структури земної поверхні (розорювання степів, вирубування лісів, меліорація, створення штучних водойм та інші зміни режиму поверхневих вод);
- зміна складу біосфери, кругообігу і балансу тих речовин, які її складають (добування корисних копалин, створення відвалів, викиди різних речовин у атмосферу та водойми);
- зміна енергетичного, зокрема теплового, балансу окремих регіонів земної кулі і всієї планети;
- зміни, які вносяться у біоту (сукупність живих організмів) внаслідок знищення деяких видів, руйнування їх природних місць існування, створення нових порід тварин та сортів рослин, переміщення їх на нові місця існування.

Під забрудненням навколишнього середовища розуміють надходження в біосферу будь-яких твердих, рідких і газоподібних речовин або видів енергії (теплоти, звуку, радіоактивності) у кількостях, що шкідливо впливають на людину, тварин і рослини як безпосередньо, так і непрямым шляхом.

Безпосередньо об'єктами забруднення (акцепторами забруднених речовин) є основні компоненти екотопу (місце існування біотичного угруповання): атмосфера, вода, ґрунт [25].

Опосередкованими об'єктами забруднення (жертвами забруднення) є складові біогеоценозу: рослини, тварини, гриби, мікроорганізми.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Втручання людини в природні процеси в біосфері, котре викликає небажані для екосистем антропогенні зміни, можна згрупувати за наступними видами забруднень:

- інградієнтне забруднення - забруднення сукупністю речовин, кількісно або якісно ворожих природним біогеоценозам (інградієнт - складова частина складної сполуки або суміші);
- параметричне забруднення пов'язане зі зміною якісних параметрів навколишнього середовища (параметр навколишнього середовища - одна з його властивостей, наприклад, рівень шуму, радіації, освітленості);
- біоценотичне забруднення полягає у впливі на склад та структуру популяції живих організмів;
- стаціонально-деструкційне забруднення (стація- місце існування популяції, деструкція - руйнування) викликає зміну ландшафтів та екологічних систем в процесі природокористування.

Фахівці по різному класифікують забруднення природного середовища, в залежності від того, який принцип беруть за основу класифікації, зокрема - за типом походження, за часом взаємодії з довкіллям, за способом впливу [25].

Джерела забруднення дуже різноманітні: серед них не тільки промислові підприємства і паливно-енергетичний комплекс, але і побутові відходи, відходи тваринництва, транспорту, а також хімічні речовини, які людина цілеспрямовано вводить до екосистеми для захисту корисних продуцентів і консументів від шкідників, хвороб і бур'янів.

#### 4.2 Вплив шуму на організм людини та розробка заходів щодо його зниженню до допустимих величин

Шум – це сукупність звуків різноманітної частоти та інтенсивності, що виникають у результаті коливального руху частинок у пружних середовищах (твердих, рідких, газоподібних). Шумом також вважають будь-який небажаний для людини звук [26].

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

Важливою характеристикою шуму є його частотний склад. Якщо в складі шуму переважають звуки з частотою коливань до 400 Гц, такий шум називається низькочастотним, якщо переважають звуки з частотою 400 – 1000 Гц – середньочастотним, якщо понад 1000 Гц – високочастотним [26].

Низькочастотний шум інтенсивністю до 100 дБ не викликає відчутної несприятливої дії на орган слуху; для середньочастотного шуму ця норма складає 85 – 90 дБ; для високочастотного – 75 – 85 дБ. Несприятливі суб'єктивні відчуття і вплив на організм людини зумовлює високочастотний шум [26].

Шум підступний, його шкідливий вплив на організм відбувається незримо, непомітно. Організм людини проти шуму практично беззахисний.

Вплив шуму на організм умовно поділяють на:

- специфічний, що спричиняє зміни в органі слуху;
- неспецифічний – з боку інших органів і систем.

Основну увагу приділяють стану органа слуху, тому що слуховий аналізатор першим сприймає звукові коливання і потерпає від впливу шуму на організм.

Дія шуму на організм людини пов'язана головним чином із застосуванням нового, високопродуктивного устаткування, з механізацією або автоматизацією трудових процесів: переходом на великі швидкості при експлуатації різних верстатів і агрегатів [26].

Джерелами шуму можуть бути двигуни, насоси, компресори, пневматичні та електричні інструменти, молоти, дробарки, верстати, центрифуги та інше обладнання, що має рухомі деталі.

Крім того, за останні роки, у зв'язку із значним розвитком міського транспорту, зростає інтенсивність шуму і в побуті.

Короткочасний, навіть одноразовий вплив шуму високої інтенсивності може спричинити повну загибель спірального органа або розрив барабанної перетинки, що супроводжується почуттям закладеності та різким болем у вухах. Наслідком баротравми нерідко буває повна втрата слуху [26].

У виробничих умовах такі травми спостерігаються надзвичайно рідко, здебільшого під час аварій чи вибухів.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Основною ознакою впливу шуму є зниження слуху по типу кохлеарного невриту. Професійне зниження слуху зазвичай буває двостороннім. Стійкі зміни слуху, як правило, розвиваються повільно, нерідко їм передують адаптація до шуму, яка характеризується нестійким зниженням слуху, що виникає безпосередньо після його впливу і зникає після припинення його дії.

Початкові прояви професійної приглухуватості найчастіше зустрічаються у осіб зі стажем роботи в умовах шуму близько 5 років. При високих рівнях шуму слухова чутливість падає вже через 1 – 2 роки, при середніх – виявляється набагато пізніше, через 5 – 10 років, тобто зниження слуху відбувається повільно, хвороба розвивається поступово [26].

У працюючих в умовах шуму основними скаргами є: зниження слуху, головний біль тупого характеру, відчуття важкості і шуму в голові, що виникають до кінця робочої зміни або після роботи, запаморочення при зміні положення тіла, підвищена дратівливість, швидка стомлюваність, зниження працездатності, уваги, підвищена пітливість, порушення ритму сну (сонливість вдень, тривожний сон у нічний час). Можуть спостерігатися неприємні відчуття в області серця у вигляді поколювань, серцебиття. Відзначається виражена нестійкість пульсу і артеріального тиску, особливо в період перебування в умовах шуму.

Ефективний захист працюючих від несприятливого впливу шуму вимагає здійснення комплексу організаційних, технічних і медичних заходів. Особливо важливо заздалегідь приймати відповідні заходи захисту від шуму.

З метою підвищення ефективності боротьби з шумом введено обов'язковий гігієнічний контроль об'єктів, що генерують шум, реєстрація фізичних факторів, що роблять шкідливий вплив на навколишнє середовище і негативно впливають на здоров'я людей. Ефективним шляхом вирішення проблеми боротьби з шумом є зниження його рівня в самому джерелі за рахунок зміни технології і конструкції машин. До заходів цього типу відносяться заміна гучних процесів безшумними, ударних – безударними, наприклад заміна клепки пайкою, кування і штампування – обробкою тиском, застосування віброізоляції, глушників, звукоізолюючих кожухів та інші. У деяких випадках зниження рівня

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

шуму досягається застосуванням звукопоглинальних пористих матеріалів, покритих перфорованими листами алюмінію, пластмас.

При необхідності підвищення коефіцієнта звукопоглинання в області високих частот звукоізолюючі шари покривають захисною оболонкою з дрібною і частою перфорацією, застосовують також штучні звукопоглиначі у вигляді конусів, кубів, закріплених над обладнанням, що є джерелом підвищеного шуму. У тих випадках, коли технічні засоби не забезпечують досягнення вимог чинних нормативів, необхідно обмеження тривалості впливу шуму та застосування засобів індивідуального захисту органу слуху. Їх використовують тоді, коли технічні засоби боротьби з шумом не забезпечують зниження його до безпечних меж. Засоби індивідуального захисту органів слуху поділяють на три типи: вкладиші, навушники і шоломи.

Важливе значення у попередженні розвитку шумової патології мають попередні (під час прийняття на роботу) і періодичні (протягом трудової діяльності) медичні огляди. Згідно з наказом Міністерства охорони здоров'я України від 21.05.2007 № 246 «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» таким оглядам підлягають особи, які працюють на виробництвах, де шум перевищує гранично допустимий рівень.

Медичними протипоказаннями до допуску на роботу, пов'язану з впливом інтенсивного шуму, крім загальних медичних протипоказань є наступні захворювання:

- стійке зниження слуху, хоча б на одне вухо, будь-якого походження;
- отосклероз і інші хронічні захворювання вуха з несприятливим прогнозом;
- порушення функції вестибулярного апарата, у тому числі хвороба Мен'єра;
- виражена вегетативно-судинна дистонія;
- гіпертонічна хвороба (всі стадії).

Кратність проведення періодичних медичних оглядів встановлюється в залежності від інтенсивності шуму. Огляди проводяться за участю отоларинголога, невропатолога і терапевта.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У кваліфікаційній роботі розроблено комп'ютерну систему локального мовлення з використанням IoT-вузлів та IP-мережі. Запропонована система призначена для централізованого керування аудіо повідомленнями, вибору зон мовлення, передавання команд керування та відтворення повідомлень у визначених приміщеннях або групах приміщень.

У першому розділі проведено аналіз особливостей систем локального мовлення. Розглянуто традиційні аналогові системи, які будуються на основі централізованих підсилювачів, комутації аудіоліній та гучномовців, а також IP-орієнтовані рішення, у яких передавання команд і аудіоданих здійснюється через локальну комп'ютерну мережу.

У другому розділі виконано проектування комп'ютерної системи локального мовлення та запропоновано її архітектуру. В якості виконавчого пристрою обрано IoT-вузол на базі мікроконтролера ESP32, що має вбудований Wi-Fi-модуль і підтримує цифровий аудіоінтерфейс I2S. Для підсилення аудіосигналу обрано модуль MAX98357A, який працює з I2S-сигналом і дозволяє безпосередньо підключити динамік. Як кінцевий звуковий елемент обрано динамік Adafruit Speaker 3" з опором 4 Ом і потужністю 3 Вт, що є достатнім для відтворення голосових повідомлень у невеликій зоні.

У третьому розділі реалізовано програмну складову системи. Розроблено архітектуру програмного забезпечення, яка включає веб-інтерфейс адміністратора, сервер керування повідомленнями, HTTP-сервер аудіо файлів, MQTT-брокер, MQTT-модуль передавання команд, модуль журналювання та програму IoT-вузла на базі ESP32.

Отже, у результаті виконання кваліфікаційної роботи розроблено технічно обґрунтований проєкт комп'ютерної системи локального мовлення, яка поєднує IoT-вузли, IP-мережу, MQTT-команди, HTTP-доступ до аудіо файлів і веб засоби адміністрування. Запропонована система є модульною, масштабованою та придатною для подальшого розвитку.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
2. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том другий. Львів: «Магнолія 2006», 2024. 204 с.
3. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том перший. Львів: «Магнолія 2006», 2024. 333 с.
4. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2. Львів: «Магнолія 2006», 2024. 328 с.
5. Дерягін В., Дрогобицький М., Луцик Н.. Методи моніторингу та оптимізації взаємодії мікросервісів в istio service mesh. Матеріали XIII науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року). Тернопіль: ТНТУ. 2025. с.111.
6. Osukhivska H., Tysh Ie., Lobur T., Shylinska I., Lupenko S. Method for estimating the convergence parameters of dynamic routing protocols in computer networks. IEEE 16th International Conference on Computer Science and Information Technologies (CSIT). Lviv, Ukraine. 2021. Proceedings Volume 2, Pp. 228–231.
7. Espressif Systems. ESP32-WROOM-32E & ESP32-WROOM-32UE Datasheet. URL: [https://documentation.espressif.com/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.html](https://documentation.espressif.com/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.html) (дата звернення: 27.05.2026 р.).
8. Analog Devices. MAX98357A: Tiny, Low-Cost, PCM Class D Amplifier with Class AB Performance. Product Datasheet. URL: <https://www.analog.com/en/products/max98357a.html> (дата звернення: 29.05.2026 р.).
9. Adafruit Industries. Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A. URL: <https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp> (дата звернення: 29.05.2026 р.).

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

10. Adafruit Industries. Speaker - 3" Diameter - 4 Ohm 3 Watt. Product ID: 1314. URL: <https://www.adafruit.com/product/1314> (дата звернення: 01.06.2026 р.).

11. Banks A., Briggs E., Borgendale K., Gupta R. MQTT Version 5.0. OASIS Standard. OASIS Open, 2019. URL: <https://www.oasis-open.org/standard/mqtt-v5-0/> (дата звернення: 01.06.2026 р.).

12. Eclipse Foundation. Eclipse Mosquitto: An Open Source MQTT Broker. URL: <https://mosquitto.org/documentation/> (дата звернення: 03.06.2026 р.).

13. Eclipse Foundation. Eclipse Paho MQTT Python Client Documentation. URL: <https://eclipse.dev/paho/files/paho.mqtt.python/html/index.html> (дата звернення: 06.06.2026 р.).

14. Fielding R., Nottingham M., Reschke J. RFC 9110. HTTP Semantics. Internet Engineering Task Force, 2022. URL: <https://www.rfc-editor.org/rfc/rfc9110> (дата звернення: 06.06.2026 р.).

15. Pallets Projects. Flask Documentation. Quickstart. URL: <https://flask.palletsprojects.com/en/stable/quickstart/> (дата звернення: 08.06.2026 р.).

16. Python Software Foundation. Python 3 Documentation. URL: <https://docs.python.org/3/> (дата звернення: 08.06.2026 р.).

17. O'Leary N. PubSubClient: A Client Library for MQTT Messaging on Arduino. GitHub Repository. URL: <https://github.com/knolleary/pubsubclient> (дата звернення: 10.06.2026 р.).

18. Arduino. Arduino Documentation. URL: <https://docs.arduino.cc/> (дата звернення: 10.06.2026 р.).

19. MDN Web Docs. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 27.05.2026 р.).

20. MDN Web Docs. CSS: Cascading Style Sheets. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 27.05.2026 р.).

21. Fette I., Melnikov A. RFC 6455. The WebSocket Protocol. Internet Engineering Task Force, 2011. URL: <https://www.rfc-editor.org/rfc/rfc6455> (дата звернення: 27.05.2026 р.).

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

22. Dizdarevic J., Michalke M., Jukan A. Engineering and Experimentally Benchmarking Open Source MQTT Broker Implementations. arXiv, 2023. URL: <https://arxiv.org/abs/2305.13893> (дата звернення: 12.06.2026 р.).

23. Saif D., Matrawy A. A Pure HTTP/3 Alternative to MQTT-over-QUIC in Resource-Constrained IoT. arXiv, 2021. URL: <https://arxiv.org/abs/2106.12684> (дата звернення: 12.06.2026 р.).

24. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для розроблені у відповідності з освітньою програмою «Комп'ютерна інженерія» першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології». Тернопіль, ТНТУ. 2024. 39 с.

25. Катренко Л.А., Катренко А.В. Охорона праці в галузі комп'ютерингу. Львів: Магнолія-2006. 2012. 544 с.

26. Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня „бакалавр”. Для студентів всіх форм навчання рівень вищої освіти перший (бакалаврський) / укл. : О. Я. Гурик , І. Б. Окіпний. Тернопіль : ТНТУ імені Івана Пулюя, 2021. 20 с.

					<b>КС КРБ 123.147.00.00 ПЗ</b>	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

« 02 » лютого 2026 р.

КОМП'ЮТЕРНА СИСТЕМА ЛОКАЛЬНОГО МОВЛЕННЯ З

ВИКОРИСТАННЯМ ІОТ-ВУЗЛІВ ТА ІР-МЕРЕЖІ

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 15 листках

**Вид робіт:**

Кваліфікаційна робота

**На здобуття освітнього ступеня «Бакалавр»**

**Спеціальність 123 «Комп'ютерна інженерія»**

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

\_\_\_\_\_ к.т.н., доц. Луцків А.М.

\_\_\_\_\_ Бойчук Н.В.

« 02 » лютого 2026 р.

« 02 » лютого 2026 р.

**Тернопіль 2026**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система локального мовлення з використанням IoT-вузлів та IP-мережі».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.147.00.00

### 1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Бойчук Назарій Віталійович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/9-188 від 24.04.2026 р.)

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи – 21.06.2026 р.

## 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Комп'ютерна система локального мовлення з використанням IoT-вузлів та IP-мережі розробляється як програмно-апаратне рішення для централізованого передавання аудіо повідомлень у межах навчального корпусу, адміністративної будівлі або іншого об'єкта з поділом приміщень на окремі зони мовлення. Система призначена для оперативного інформування користувачів, передавання службових оголошень, навчальних повідомлень, попереджувальних сигналів та екстрених аудіо повідомлень.

На відміну від традиційних аналогових систем локального мовлення, у яких передавання звуку здійснюється через окремі аудіолінії та централізовані підсилювачі, запропонована система використовує локальну IP-мережу та окремі IoT-вузли мовлення. Кожен вузол встановлюється у визначеній зоні, наприклад в аудиторії, коридорі, лабораторії або на поверсі, і виконує відтворення аудіоповідомлення через підключений динамік.

Система повинна забезпечувати централізоване керування аудіотрансляціями через вебінтерфейс адміністратора. Адміністратор повинен мати можливість вибрати зону мовлення, завантажити або вибрати аудіофайл, задати пріоритет повідомлення та запустити трансляцію. Передавання керуючих команд між сервером і IoT-вузлами повинно здійснюватися через MQTT-брокер, а доступ до аудіофайлів – через HTTP-сервер.

Система призначена для підвищення гнучкості, масштабованості та зручності керування локальним мовленням. Вона дозволяє відмовитися від складної аналогової комутації аудіоліній, реалізувати логічне зонування, централізоване журналювання подій, контроль стану вузлів і підтримку повідомлень з різним рівнем пріоритету.

## 2.2 Мета створення системи

Метою створення комп'ютерної системи локального мовлення є розроблення доступного програмно-апаратного рішення, яке забезпечує централізоване керування аудіоповідомленнями, передавання команд до IoT-вузлів через локальну IP-мережу та відтворення повідомлень у визначених зонах.

Досягнення мети роботи можливе шляхом розв'язання таких задач:

- проведення аналізу традиційних аналогових та IP-орієнтованих систем локального мовлення;
- визначення функціональних і нефункціональних вимог до системи;
- розроблення загальної архітектури комп'ютерної системи локального мовлення;
- обґрунтування вибору апаратних компонентів IoT-вузла, зокрема ESP32, I2S-підсилювача MAX98357A та динаміка;
- розроблення схеми підключення компонентів IoT-вузла;
- проектування структури мережевого підключення та моделі зон мовлення;
- реалізація HTTP-сервера для зберігання та передавання аудіофайлів;
- реалізація MQTT-модуля для передавання керуючих команд;

- розроблення вебінтерфейсу адміністратора для запуску трансляцій;
- розроблення програмного забезпечення IoT-вузла на базі ESP32;
- забезпечення передавання статусних повідомлень від IoT-вузлів до сервера.

### 2.3 Характеристика об'єкту

Об'єктом проєктування у даній роботі є комп'ютерна система локального мовлення, яка функціонує в межах локальної IP-мережі та використовує розподілені IoT-вузли для відтворення аудіоповідомлень. Система складається із серверної частини, мережевої інфраструктури та виконавчих вузлів мовлення.

Серверна частина виконує функції централізованого керування. Вона повинна містити вебінтерфейс адміністратора, HTTP-сервер для роботи з аудіофайлами, MQTT-брокер або MQTT-модуль передавання команд, а також журнал подій. Сервер повинен забезпечувати обробку дій адміністратора, збереження аудіофайлів, формування керуючих команд, вибір MQTT-теми відповідно до зони мовлення та фіксацію результатів роботи системи.

IoT-вузол мовлення є кінцевим виконавчим пристроєм системи. Він повинен підключатися до локальної Wi-Fi-мережі, встановлювати з'єднання з MQTT-брокером, підписуватися на теми своєї зони, отримувати команди від сервера та запускати відтворення аудіофайлу. Для формування аудіосигналу вузол використовує інтерфейс I2S, через який ESP32 передає цифрові аудіодані на підсилювач MAX98357A. Підсилювач забезпечує подавання сигналу на динамік.

Система повинна підтримувати логічне зонування. Зоною мовлення може бути окрема аудиторія, коридор, поверх, корпус або група вузлів. Один IoT-вузол може належати до однієї або кількох логічних зон. Це дозволяє адміністратору запускати трансляцію як в окремому приміщенні, так і для групи вузлів або всієї системи.

У системі повинна бути передбачена підтримка пріоритетів повідомлень. Звичайні повідомлення використовуються для поточного інформування, а екстрені — для передавання термінових оголошень. У випадку надходження екстреного

повідомлення IoT-вузол повинен мати можливість перервати поточне відтворення звичайного повідомлення та перейти до відтворення повідомлення з вищим пріоритетом.

### 3 Вимоги до системи

#### 3.1 Вимоги до системи в цілому

Вимоги до комп'ютерної системи локального мовлення з використанням IoT-вузлів та IP-мережі поділяються на функціональні та нефункціональні. Функціональні вимоги визначають основні можливості системи, які повинні бути реалізовані на апаратному та програмному рівнях. Нефункціональні вимоги стосуються надійності, масштабованості, зручності використання, стабільності роботи та можливості подальшого розвитку системи.

Основними функціональними вимогами до системи є:

- можливість запуску аудіоповідомлень через вебінтерфейс адміністратора;
- можливість вибору окремої зони мовлення або групи зон;
- можливість запуску трансляції для всіх IoT-вузлів системи;
- можливість завантаження аудіофайлів на сервер;
- можливість зберігання аудіофайлів у каталозі HTTP-сервера;
- можливість передавання керуючих команд через MQTT;
- можливість отримання аудіофайлів IoT-вузлами через HTTP;
- можливість відтворення аудіоповідомлень через I2S-підсилювач і динамік;
- можливість передавання статусів IoT-вузлів на сервер;
- можливість перегляду стану вузлів у вебінтерфейсі адміністратора;
- можливість ведення журналу подій;
- можливість підтримки звичайних та екстрених повідомлень;
- можливість масштабування системи шляхом додавання нових вузлів.

На програмному рівні система повинна забезпечувати:

- стабільне підключення серверної частини до MQTT-брокера;
- формування JSON-команд для IoT-вузлів;
- вибір MQTT-теми відповідно до вибраної зони;
- перевірку коректності аудіофайлів, що завантажуються;
- формування URL-адреси аудіофайлу для передавання IoT-вузлу;
- обробку статусних повідомлень від вузлів;
- відображення поточного стану системи у вебінтерфейсі;
- логування дій адміністратора, команд і результатів виконання.

Основними нефункціональними вимогами є:

- система повинна мати зрозумілу модульну структуру;
- компоненти системи повинні бути доступними для реалізації навчального прототипу;
  - затримка між запуском команди та початком відтворення повинна бути прийнятною для службових повідомлень;
  - система повинна підтримувати подальше додавання нових IoT-вузлів;
  - програмна логіка повинна бути придатною для тестування та пояснення;
  - веб-інтерфейс повинен бути простим і зрозумілим для адміністратора;
  - система повинна забезпечувати стабільну роботу в межах локальної IP-мережі;
  - архітектура повинна дозволити подальшу модернізацію.

### 3.1.1 Вимоги до способів та засобів зв'язку між компонентами системи

У комп'ютерній системі локального мовлення необхідно передбачити взаємодію між серверною частиною, локальною IP-мережею та IoT-вузлами мовлення. Основним середовищем передавання даних повинна бути локальна IP-мережа, організована на основі Wi-Fi або Ethernet.

Серверна частина повинна бути підключена до локальної мережі та мати постійну IP-адресу. Це необхідно для того, щоб IoT-вузли могли підключатися до

MQTT-брокера та HTTP-сервера за сталою адресою. Для прототипу може використовуватися адреса сервера 192.168.1.10.

Передавання керуючих команд між сервером та IoT-вузлами повинно виконуватися через MQTT. MQTT-брокер повинен приймати повідомлення від серверного модуля та доставляти їх вузлам, які підписані на відповідні теми. Для адресації команд повинні використовуватися MQTT-теми, наприклад:

- campus/audio/zone1;
- campus/audio/zone2;
- campus/audio/all;
- campus/audio/emergency;
- campus/audio/status/zone1.

Для доступу до аудіофайлів повинен використовуватися HTTP. Сервер повинен зберігати аудіофайли у визначеному каталозі та надавати доступ до них за URL-адресою. IoT-вузол після отримання MQTT-команди повинен звернутися до HTTP-сервера, отримати аудіофайл або аудіодані та запустити відтворення.

Зв'язок між ESP32 та підсилювачем MAX98357A повинен здійснюватися через цифровий інтерфейс I2S. Для цього повинні використовуватися сигнальні лінії BCLK, LRC/WS та DIN. Для коректної роботи компонентів необхідно забезпечити спільну землю GND між ESP32 і MAX98357A.

Таким чином, основними способами зв'язку між компонентами системи є:

- Wi-Fi або Ethernet для підключення до локальної IP-мережі;
- MQTT для передавання керуючих команд і статусів;
- HTTP для доступу до аудіофайлів;
- I2S для передавання цифрового аудіосигналу від ESP32 до MAX98357A;
- провідникове підключення динаміка до виходу підсилювача.

### 3.1.2 Вимоги по діагностуванню системи

Діагностування комп'ютерної системи локального мовлення повинно передбачати перевірку працездатності серверної частини, мережевого з'єднання, MQTT-брокера, HTTP-сервера, вебінтерфейсу та IoT-вузлів.

Під час запуску системи повинна перевірятися доступність основних програмних компонентів. HTTP-сервер повинен мати службовий маршрут для перевірки працездатності, наприклад /health. MQTT-модуль повинен перевіряти підключення до брокера та можливість публікації повідомлень у задані теми.

У веб-інтерфейсі адміністратора повинна відображатися інформація про IP-адресу вузла, його зону та поточний стан. У разі відсутності статусних повідомлень від вузла протягом визначеного часу сервер повинен позначити його як недоступний.

До діагностичних заходів також належить перевірка коректності фізичних з'єднань між ESP32, MAX98357A, динаміком і джерелом живлення. Особливу увагу потрібно приділяти спільній землі GND, стабільності живлення 5 В та правильності підключення I2S-ліній.

### 3.1.3 Перспективи розвитку, модернізація системи

Перспективи розвитку комп'ютерної системи локального мовлення пов'язані з розширенням її функціональних можливостей, підвищенням надійності та інтеграцією з іншими інформаційними системами навчального закладу або підприємства.

Одним із напрямів модернізації є збільшення кількості IoT-вузлів та розширення моделі зон мовлення. Система може бути доповнена новими вузлами для окремих аудиторій, лабораторій, коридорів, поверхів або корпусів. При цьому загальна архітектура системи не потребує суттєвих змін, оскільки новий вузол може бути доданий шляхом налаштування MQTT-теми та реєстрації його у серверній частині.

Іншим напрямом розвитку є вдосконалення вебінтерфейсу адміністратора. У подальшому доцільно реалізувати авторизацію користувачів, розмежування прав

доступу, окрему сторінку керування зонами, автоматичне оновлення стану вузлів у реальному часі та перегляд архіву подій.

Також перспективним є впровадження підтримки потокового аудіо або трансляції голосу в реальному часі. У базовому варіанті система орієнтована на відтворення попередньо підготовлених аудіофайлів, однак у майбутньому вона може бути доповнена підтримкою live-трансляції з мікрофона.

Додатковим напрямом розвитку є інтеграція системи з іншими сервісами: системою розкладу занять, системою оповіщення про небезпеку, охоронною сигналізацією або внутрішньою інформаційною системою університету. Це дозволить автоматично запускати повідомлення відповідно до подій або сценаріїв.

#### 3.1.4 Вимоги до надійності системи

Комп'ютерна система локального мовлення повинна забезпечувати достатній рівень надійності, оскільки вона може використовуватися не лише для звичайних службових повідомлень, а й для передавання екстрених оголошень. Надійність системи повинна забезпечуватися як на апаратному, так і на програмному рівнях.

Серверна частина повинна стабільно обробляти запити адміністратора, зберігати аудіофайли, формувати MQTT-команди та записувати події у журнал. У разі виникнення помилки під час завантаження файлу або публікації MQTT-команди система повинна повідомити адміністратора про помилку.

MQTT-з'єднання між сервером і IoT-вузлами повинно підтримувати повторне підключення у випадку тимчасової втрати зв'язку. IoT-вузол повинен автоматично намагатися повторно підключитися до Wi-Fi-мережі та MQTT-брокера, якщо з'єднання було втрачено.

HTTP-сервер повинен коректно обробляти запити на отримання аудіофайлів. Якщо файл відсутній або має неправильний формат, сервер повинен повернути повідомлення про помилку. Це дозволить уникнути некоректного відтворення або зависання вузла.

IoT-вузол повинен забезпечувати стабільне живлення ESP32 та MAX98357A. Джерело живлення повинно мати достатній запас за струмом для роботи мікроконтролера та підсилювача під час відтворення аудіо. Для підвищення надійності необхідно забезпечити правильне підключення ліній живлення, спільної землі та сигнальних ліній I2S.

Система повинна забезпечувати:

- автоматичне повторне підключення ESP32 до Wi-Fi;
- автоматичне повторне підключення до MQTT-брокера;
- контроль доступності IoT-вузлів;
- журналювання помилок;
- відображення стану вузлів у вебінтерфейсі;
- обробку помилок під час завантаження аудіофайлів;
- коректну роботу під час запуску звичайних та екстрених повідомлень.

### 3.1.5 Вимоги до надійності системи

Комп'ютерна система локального мовлення повинна забезпечувати виконання базових функцій керування аудіотрансляціями в межах локальної IP-мережі.

До основних функцій системи належать:

- запуск аудіоповідомлення адміністратором;
- вибір зони мовлення;
- вибір пріоритету повідомлення;
- передавання команди до IoT-вузла;
- отримання аудіофайлу IoT-вузлом;
- відтворення аудіоповідомлення через динамік;
- передавання статусу вузла на сервер;
- відображення стану вузлів;
- ведення журналу подій.

Сервер керування повинен виконувати такі задачі:

- приймати запити від вебінтерфейсу;

- перевіряти коректність параметрів трансляції;
- зберігати аудіофайли;
- формувати URL-адресу аудіофайлу;
- формувати MQTT-команду;
- визначати MQTT-тему відповідно до вибраної зони.

IoT-вузол повинен виконувати такі задачі:

- підключатися до Wi-Fi-мережі;
- отримувати IP-адресу;
- підключатися до MQTT-брокера;
- підписуватися на теми своєї зони, загальної трансляції та екстрених повідомлень;
- приймати MQTT-команди;
- аналізувати параметри команди.

Таким чином, система повинна забезпечувати повний цикл роботи: від вибору аудіоповідомлення адміністратором до його відтворення у визначеній зоні та фіксації результату виконання.

### 3.1.6 Вимоги до апаратного забезпечення

Апаратне забезпечення комп'ютерної системи локального мовлення повинно відповідати вимогам доступності, надійності, простоти підключення та достатньої продуктивності для реалізації навчального прототипу.

Основними апаратними компонентами системи повинні бути:

- сервер або персональний комп'ютер для роботи вебінтерфейсу, HTTP-сервера та MQTT-брокера;
- Wi-Fi-маршрутизатор або локальний комутатор для організації IP-мережі;
- мікроконтролер ESP32 для реалізації IoT-вузла;
- цифровий I2S-підсилювач MAX98357A;
- динамік 4 Ом / 3 Вт;
- джерело живлення 5 В;

- з'єднувальні провідники;
- макетна плата або монтажний модуль.

### 3.1.7 Вимоги до програмного забезпечення

Програмне забезпечення комп'ютерної системи локального мовлення повинно складатися із серверної частини та програмного забезпечення IoT-вузлів.

Серверна частина повинна бути реалізована з використанням мови програмування Python. Для створення вебінтерфейсу та HTTP-сервера доцільно використовувати Flask. Для роботи з MQTT доцільно використовувати бібліотеку `raho-mqtt`. MQTT-брокером може бути Mosquitto.

Програмне забезпечення IoT-вузла повинно бути реалізоване мовою C++ у середовищі Arduino IDE або сумісному середовищі для ESP32. Програмне забезпечення повинно мати модульну структуру, а веб-інтерфейс має забезпечувати просту взаємодію адміністратора із системою.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

1 Структура аналогових і цифрових систем локального мовлення.

2 Архітектура системи локального мовлення з використанням IoT-вузлів та IP-мережі.

3 Функціональна схема IoT-вузла в системі локального мовлення.

4 Блок схема алгоритму роботи системи локального мовлення.

\*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	26.01 – 02.02
2	Робота над першим розділом «Аналіз вимог до систем локального мовлення»	03.02 – 15.02
3	Робота над другим розділом «Проектування системи локального мовлення з використанням IoT-вузлів та IP-мережі»	20.04 – 25.04
4	Робота над третім розділом «Реалізація програмної складової комп'ютерної системи локального мовлення»	26.04 – 05.05
5	Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»	07.05 – 25.05
6	Оформлення пояснювальної записки і графічного матеріалу	26.05 – 7.06
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	8.06 – 14.06
8	Попередній захист кваліфікаційної роботи бакалавра	15.06 – 21.06
9	Захист кваліфікаційної роботи бакалавра	26.06.2026

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

## Додаток Б Блок схема алгоритму роботи IoT-вузла

