

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)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система голосового керування IoT-пристроями для осіб з обмеженими можливостями*

Виконав: студент 4 курсу, групи СІ-41

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

*Бень О.Р.*

(прізвище та ініціали)

Керівник

(підпис)

*Луцик Н.С.*

(прізвище та ініціали)

Нормоконтроль

(підпис)

*Тили Є.В.*

(прізвище та ініціали)

Завідувач кафедри

(підпис)

*Осухівська Г.М.*

(прізвище та ініціали)

Рецензент

(підпис)

*Деркач М.В.*

(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

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

«24» 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 Бень Олександр Романовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система голосового керування IoT-пристроями для осіб з обмеженими можливостями

Керівник роботи Луцик Надія Степанівна, PhD, 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 15.06.2026 р.

3. Вихідні дані до роботи Технічне завдання

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

Вступ

1. Аналіз технічного завдання

2. Проєктна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорона праці

Висновки

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

1. Структурна схема системи

2. Схема електрична принципова

3. Блок-схема алгоритму роботи

4. Результати моделювання системи



## АНОТАЦІЯ

Бень О.Р. Комп'ютерна система голосового керування IoT-пристроями для осіб з обмеженими можливостями : робота на здобуття освітнього ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: голосове керування, IoT-технології, комп'ютерна система, мовні моделі, штучний інтелект, вбудовані системи, мікроконтролер, хмарні сервіси.

Кваліфікаційна робота присвячена розробці комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями. Метою роботи є проєктування та реалізація апаратно-програмного комплексу, здатного забезпечувати інтелектуальну голосову взаємодію користувача з розумними пристроями в мережі Інтернет речей.

У першому розділі проведено аналіз технічного завдання, сформульовано функціональні та експлуатаційні вимоги до системи, а також виконано огляд існуючих аналогів голосових асистентів і IoT-рішень. На основі порівняльного аналізу визначено основні переваги та недоліки відомих систем і обґрунтовано доцільність розробки власного рішення.

У другому розділі розроблено апаратне забезпечення системи, зокрема побудовано структурну схему та електричну принципову схему. Обґрунтовано вибір елементної бази, включно з мікроконтролером, модулями введення-виведення та засобами зв'язку, з урахуванням вимог до продуктивності, енергоефективності та масштабованості.

Третій розділ присвячено розробці алгоритму роботи системи та програмного забезпечення. Реалізовано інтеграцію з IoT-платформою та сучасними мовними моделями, а також проведено тестування розробленої системи з метою перевірки її працездатності та відповідності поставленим вимогам.

## ANNOTATION

Ben O.R. Computer System for Voice Control of IoT Devices for Disabled People. Bachelor's Graduation Thesis: speciality 123 — Computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: voice control, IoT technologies, computer system, language models, artificial intelligence, embedded systems, microcontroller, cloud services.

This thesis is devoted to the development of a computer system for voice control of IoT devices for disabled people. The aim of the work is to design and implement a hardware–software solution that enables intelligent voice interaction between the user and smart devices within the Internet of Things environment.

The first chapter analyzes the technical specifications, formulates functional and non-functional system requirements, and reviews existing voice assistant and IoT solutions. Based on a comparative analysis, the advantages and limitations of current systems are identified, and the relevance of developing a proprietary solution is justified.

The second chapter focuses on the design of the system hardware. A structural diagram and an electrical schematic diagram are developed, and the selection of the component base, including the microcontroller, input/output modules, and communication interfaces, is justified considering performance, energy efficiency, and scalability requirements.

The third chapter describes the development of the system algorithm and software. Integration with an IoT platform and modern language models is implemented, followed by testing of the developed system to verify its functionality and compliance with the defined requirements.

## ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	10
1.1 Огляд сфер застосування систем голосового керування .....	10
1.2 Аналіз вимог до система голосового керування IoT-пристроями .....	11
1.3 Огляд існуючих засобів для розробки комп'ютерної системи голосового керування.....	12
1.4 Аналіз можливих рішень поставленого завдання.....	17
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА .....	19
2.1 Структура системи голосового керування IoT-пристроями.....	19
2.2 Апаратне забезпечення системи голосового керування IoT-пристроями.....	22
2.2.1 Платформа ESP32-S3 DevKitC.....	22
2.2.2 Модуль MEMS-мікрофона INMP441 .....	25
2.2.3 Модуль аудіопідсилювача на MAX98357 .....	28
2.2.4 OLED дисплей 0.96".....	31
2.3 Опис електричної принципової схеми голосового керування пристроями .....	34
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА .....	37
3.1 Алгоритм роботи системи голосового керування IoT-пристроями для осіб з обмеженими можливостями .....	37
3.2 Розробка програмного забезпечення.....	40
3.2.1 Ініціалізація апаратних та програмних компонентів системи.....	40
3.2.2 Основний цикл обробки голосових запитів користувача .....	41
3.2.3 Формування та кодування аудіоданих для передачі у хмарні сервіси .....	43
3.2.4 Інтеграція з сервісом розпізнавання мовлення та формування текстового запиту.....	45
3.2.5 Інтеграція з мовною моделлю для інтелектуальної обробки запитів .....	46

					<i>КС КРБ 123.143.00.00 ПЗ</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розробив	Бень О.Р.				Літ.	Арк.	Акрушів
Перевірів	Луцик Н.С.				5	81	
Рецензент	Деркач М.В.				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
Н. Контр.	Тиш Е.В.						
Зав. каф.	Осунівська Г.М.						

*Комп'ютерна система голосового керування IoT-пристроями для осіб з обмеженими можливостями*

3.2.6 Синтез мовлення та озвучення відповіді користувачу.....	48
3.3 Реалізація взаємодії з сучасними мовними моделями для реалізації системи голосового керування IoT-пристроями.....	50
3.3.1 Налаштування Google AI Studio .....	50
3.3.2 Налаштування Google Speech .....	51
3.4 Результати моделювання та тестування системи .....	52
<b>РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....</b>	<b>56</b>
4.1 Характеристика життєдіяльності людини у системі «людина – машина – середовище існування».....	56
4.2 Шляхи збереження працездатності та підвищення продуктивності праці .....	59
<b>ВИСНОВКИ.....</b>	<b>62</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>63</b>
Додаток А Технічне завдання	
Додаток Б Перелік елементів	
Додаток В Лістинг програми	

					<i>КС КРБ 123.143.00.00 ПЗ</i>	Арк.
						6
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## СПИСОК СКОРОЧЕНЬ

API – Application Programming Interface

GPIO – General Purpose Input Output

I2C – Inter-Integrated Circuit

IoT – Internet of Things

I2S – Inter-IC Sound

JSON – JavaScript Object Notation

MEMS – Micro Electro Mechanical Systems

OLED – Organic Light Emitting Diode

RGB – Red Green Blue

UART – Universal Asynchronous Receiver Transmitter

АЦП – аналого-цифровий перетворювач

ГІ – голосовий інтерфейс

КСГК – комп'ютерна система голосового керування

МК – мікроконтролер

ММ – мовна модель

СРМ – система розпізнавання мовлення

ЦАП – цифро-аналоговий перетворювач

ЦОС – цифрова обробка сигналів

ХС – хмарний сервіс

					КС КРБ 123.143.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Стрімкий розвиток інформаційних технологій, зокрема Інтернету речей (IoT) та методів штучного інтелекту, зумовлює появу нових підходів до організації людино-машинної взаємодії. Одним із найбільш перспективних напрямів у цій галузі є використання голосових інтерфейсів, які забезпечують природну, зручну та інтуїтивно зрозумілу форму керування цифровими та вбудованими системами. Сучасні голосові асистенти знаходять широке застосування у побуті, промисловості, освіті та сфері «розумного дому», проте більшість існуючих рішень є закритими, орієнтованими на масового користувача та мають обмежені можливості персоналізації й інтеграції з користувацькими IoT-пристроями.

Актуальною проблемою є створення комп'ютерних систем голосового керування, які поєднують гнучкість IoT-архітектури з можливостями сучасних мовних моделей, здатних забезпечити більш точне розпізнавання та обробку природної мови з урахуванням індивідуальних потреб користувача. Особливо важливим є розроблення таких систем на основі доступних апаратних платформ, що дозволяє зменшити вартість, підвищити масштабованість і забезпечити адаптацію системи під конкретні сценарії використання. У зв'язку з цим задача розробки персоналізованого голосового асистента на базі IoT-технологій з інтеграцією сучасних мовних моделей є актуальною та практично значущою.

Метою кваліфікаційної роботи є проєктування та реалізація комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями, а також перевірка її працездатності та відповідності визначеним вимогам.

Для досягнення поставленої мети в роботі необхідно вирішити такі задачі:

- проаналізувати технічне завдання та сформулювати вимоги до комп'ютерної системи голосового керування IoT-пристроями;
- виконати огляд та аналіз існуючих аналогів голосових асистентів і IoT-рішень;

					КС КРБ 123.143.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- розробити структурну схему та електричну принципову схему апаратної частини системи;
- обґрунтувати вибір елементної бази з урахуванням функціональних і експлуатаційних вимог;
- розробити алгоритм функціонування системи;
- реалізувати ПЗ для мікроконтролерної платформи;
- виконати інтеграцію розробленої системи з IoT-платформою та сучасними мовними моделями;
- провести тестування системи та проаналізувати отримані результати.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		9

# РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

## 1.1 Огляд сфер застосування систем голосового керування

Системи голосового керування знаходять широке застосування в різних сферах діяльності людини, що зумовлено зростанням кількості підключених пристроїв та потребою у зручних і природних засобах керування ними. Використання голосового інтерфейсу дозволяє значно спростити взаємодію користувача з технічними системами, підвищити оперативність виконання команд та зменшити когнітивне навантаження, що є особливо актуальним у середовищах із великою кількістю функцій та пристроїв.

Однією з основних сфер застосування таких систем є побутова автоматизація та концепція «розумного дому». У даному контексті вони використовуються для керування освітленням, кліматичними системами, побутовою технікою, засобами безпеки та мультимедійними пристроями. Інтеграція з IoT-платформами дозволяє об'єднати різноманітні пристрої в єдину керовану систему, а застосування сучасних мовних моделей забезпечує більш точне розуміння команд і можливість персоналізації сценаріїв відповідно до звичок та потреб конкретного користувача.

Важливою сферою застосування є освітнє та науково-досліднє середовище. Персоналізовані голосові асистенти можуть використовуватися як інструменти підтримки навчального процесу, забезпечуючи доступ до інформації, нагадування про навчальні події, а також керування лабораторним обладнанням. У межах наукових досліджень такі системи слугують експериментальними платформами для вивчення методів обробки природної мови, взаємодії людини з комп'ютерними системами та інтеграції штучного інтелекту з вбудованими пристроями.

У сфері промисловості та офісної інфраструктури засоби голосового керування застосовуються для автоматизації рутинних операцій, моніторингу

					<i>КС КРБ 123.143.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Бень О.Р.</i>			<i>Аналіз технічного завдання</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевірив</i>		<i>Луцук Н.С.</i>					<i>10</i>	<i>9</i>
<i>Рецензент</i>		<i>Деркач М.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Тиш Е.В.</i>						
<i>Зав. каф.</i>		<i>Осужівська Г.М.</i>						

стану обладнання та керування інформаційними системами. Голосове керування дозволяє оператору взаємодіяти з системою без відволікання від основних завдань, що є критично важливим у виробничих умовах. Завдяки використанню IoT-технологій можливе централізоване управління розподіленими об'єктами, а персоналізація підвищує ефективність роботи окремих користувачів або груп.

Окремої уваги заслуговує застосування таких систем у сфері доступності та соціальної інклюзії. Системи голосового керування значно спрощують взаємодію з технічними засобами для людей з обмеженими можливостями, зокрема з порушеннями зору або моторики. Можливість налаштування поведінки системи та адаптації до індивідуальних особливостей користувача робить їх гарним інструментом підвищення якості життя.

Отже, комп'ютерні системи голосового керування є універсальними та актуальними. Широкий спектр можливих застосувань, поєднання IoT-технологій із сучасними мовними моделями та орієнтація на персоналізацію визначають доцільність розробки такої системи та формують вимоги до її функціональних і технічних характеристик, що враховуються під час подальшого проєктування.

## 1.2 Аналіз вимог до система голосового керування IoT-пристроями

Вимоги до комп'ютерної системи голосового керування формуються з урахуванням умов експлуатації, очікуваних сценаріїв використання, а також обмежень, пов'язаних із апаратними ресурсами та мережевою інфраструктурою.

Насамперед система повинна забезпечувати коректний прийом і обробку голосових команд користувача. Це передбачає вимоги до якості захоплення аудіосигналу, стійкості до зовнішніх шумів та швидкості реагування. Важливою є здатність системи працювати в реальному часі або з мінімально допустимими затримками, що безпосередньо впливає на зручність взаємодії користувача з голосовим асистентом. Окремо висувуються вимоги до точності розпізнавання мовлення та інтерпретації команд, що досягається за рахунок інтеграції сучасних мовних моделей.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

З огляду на використання IoT-технологій, система повинна підтримувати стабільний та захищений обмін даними між локальними пристроями та хмарними сервісами. До мережевих вимог належать підтримка стандартних протоколів передачі даних, надійність бездротового з'єднання та можливість роботи в умовах обмеженої пропускну здатності мережі. Важливою вимогою є масштабованість, яка забезпечує можливість підключення додаткових пристроїв без суттєвих змін архітектури системи.

Функціональні вимоги включають підтримку персоналізації, зокрема збереження індивідуальних налаштувань користувача, адаптацію поведінки асистента та формування відповідей з урахуванням контексту. Система повинна забезпечувати гнучке керування IoT-пристроями, виконання команд різної складності та можливість розширення набору підтримуваних функцій.

До апаратних вимог належать обмеження щодо енергоспоживання, продуктивності мікроконтролерної платформи та обсягу пам'яті. Апаратна частина повинна забезпечувати достатню обчислювальну потужність для попередньої обробки даних і стабільну роботу засобів зв'язку. Водночас система має бути компактною та економічно доцільною для використання в побутових умовах.

Окрему групу становлять вимоги до надійності та безпеки. Система повинна коректно обробляти помилки, забезпечувати захист персональних даних користувача та зберігати працездатність у разі короточасних збоїв зв'язку. Узагальнений аналіз вимог створює основу для обґрунтованого вибору апаратних і програмних засобів та визначає напрям подальшої розробки комп'ютерної системи.

### 1.3 Огляд існуючих засобів для розробки комп'ютерної системи голосового керування

Ринок голосових асистентів та інтегрованих IoT-систем сьогодні представлений широким спектром комерційних і відкритих рішень, що

					КС КРБ 123.143.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

демонструють різний рівень функціональності, інтегрованості та адаптації під конкретні сценарії використання. Серед найбільш відомих на глобальному рівні голосових помічників – Amazon Alexa, Google Assistant, Apple Siri, а також інші платформи, включно з локальними рішеннями (наприклад, AliGenie від Alibaba) та відкритими проєктами на зразок Microsoft. Однак, незважаючи на широке поширення та технічний прогрес у цій сфері, існуючі засоби мають низку суттєвих обмежень та недоліків, які обумовлюють необхідність розробки нового програмно-апаратного рішення з урахуванням сучасних вимог до персоналізації, надійності та інтеграції з IoT-екосистемою.

Одним із найбільш поширених і комерційно розвинених голосових асистентів є Amazon Alexa – цифровий помічник, що працює в екосистемі пристроїв Amazon Echo [1] та підтримує керування великою кількістю розумних пристроїв (рис. 1.1).

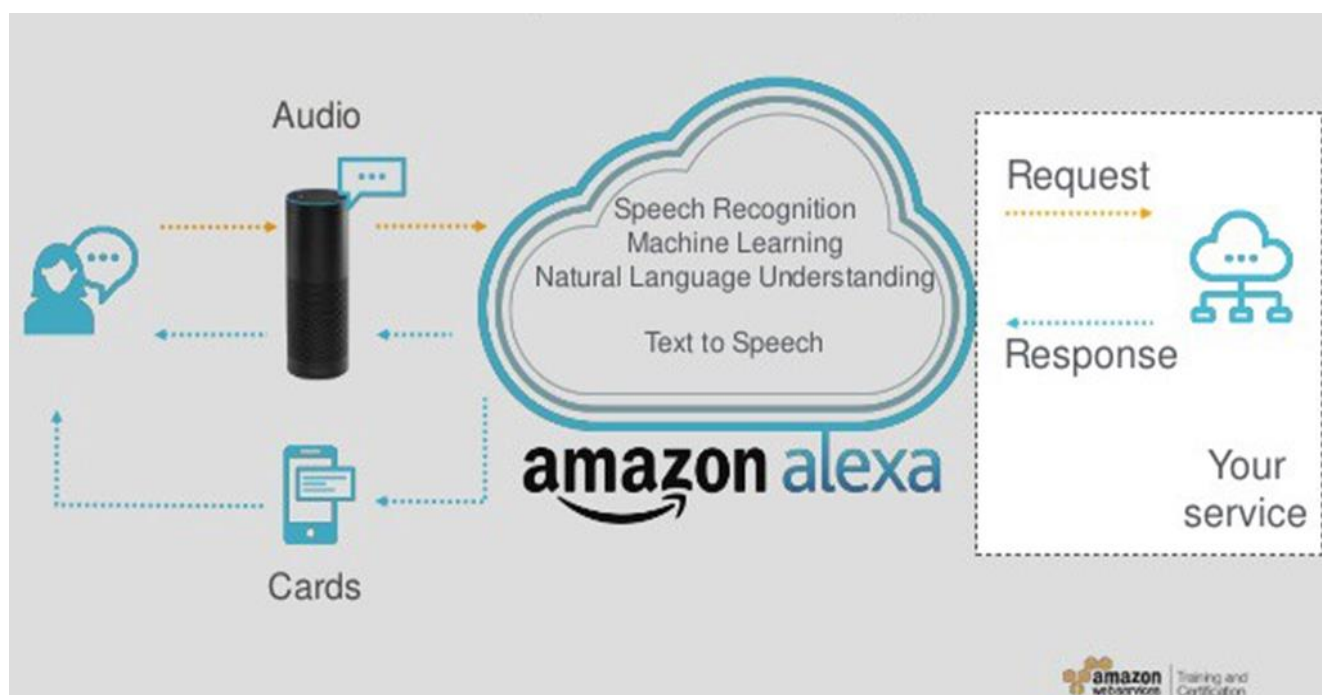


Рисунок 1.1 – Цифровий помічник Amazon Alexa

Alexa відзначається широкою інтеграцією у сфері «розумного дому» та підтримкою автоматизації сценаріїв за допомогою голосових команд. Однак Alexa має суттєві обмеження в адаптації під окремого користувача: персоналізація

					КС КРБ 123.143.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

команд є досить обмеженою, а поведінкові моделі сформовані в рамках глобальної екосистеми, що не завжди відповідає індивідуальним потребам користувачів. Крім того, існують технічні виклики, пов'язані з інтеграцією Alexa з передовими системами великих мовних моделей (LLM), включно з проблемами «галюцинацій» відповіді, затримками обробки та забезпеченням надійності на глобальному масштабі, що залишається відкритою проблемою у галузі корпоративної розробки таких асистентів

Google Assistant – ще одне провідне рішення, яке глибоко інтегроване в екосистему Android, Google Home та інших продуктів Google [2]. Цей асистент має сильні позиції щодо загальних інформаційних запитів, контекстної обробки та взаємодії з пошуковими сервісами (рис. 1.2).



Рисунок 1.2 – Цифровий асистент Google Assistant

Нещодавній реліз Gemini for Home демонструє спробу використання більш потужних моделей штучного інтелекту для більш природної та контекстно-усвідомленої взаємодії з користувачем. Проте і цей підхід має обмеження в персоналізації й прозорості обробки даних, а також часто залежить від хмарних

					КС КРБ 123.143.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

сервісів, що створює ризики з точки зору приватності та вимог до мережевої доступності. Також загальні комерційні рішення не завжди підтримують гнучкі сценарії локальної обробки природної мови або адаптацію алгоритмів під конкретні вимоги користувача чи IoT-пристрої сторонніх виробників.

Apple Siri успішно інтегрована в продукти Apple та відзначається високим рівнем захисту приватності [3]. Вона добре справляється із загальними командами на пристроях Apple та деякими сценаріями автоматизації «розумного дому». Однак Siri значно обмежена екосистемою Apple та не надає гнучких механізмів для розробників третіх сторін. Це означає, що її можливості як голосового асистента в IoT-середовищі значно поступаються галузевим лідерам, а також не підтримують розширюваних сценаріїв інтеграції сучасних мовних моделей.

Важливо також відзначити проекти вільного ПЗ, наприклад Microsoft [4], який пропонує відкритий код голосового асистента з можливістю роботи без використання сторонніх хмарних сервісів (рис. 1.3).



Рисунок 1.3 – Віртуальний асистент Microsoft

					КС КРБ 123.143.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Такий підхід може вирішувати деякі проблеми приватності та локального контролю даних, проте відсутність активної підтримки та розвитку з боку основної команди розробників значно обмежує актуальність і функціональні можливості цього рішення у порівнянні з комерційними платформами.

Крім цих комерційних платформ, на ринку також представлені локальні або регіональні рішення, наприклад AliGenie (Alibaba), які орієнтовані на внутрішній ринок певних країн [5]. Хоча вони мають певну локалізацію та підтримують специфічні мови, їх адаптація під широку екосистему IoT-пристроїв та інтеграцію з сучасними механізмами штучного інтелекту є обмеженою, а підтримка їхнього локального розвитку може залежати від політики компаній.

Огляд сучасних рішень також демонструє загальні проблеми та недоліки, що характерні для більшості існуючих голосових асистентів. Перше – обмежена персоналізація: багато асистентів орієнтовані на широкого користувача, а не на індивідуальний профіль, що знижує ефективність їх використання в специфічних сценаріях. Друге – залежність від хмарних сервісів та недостатня підтримка локального виконання команд, що створює виклики з точки зору затримок, стабільності та конфіденційності обробки даних. Третє – обмежена або складна інтеграція з IoT-пристроями сторонніх виробників, що зумовлює необхідність використання додаткових шлюзів, стандартів чи сумісностей. Четверте – питання безпеки та захисту даних, оскільки комерційні платформи часто обробляють велику кількість персональної інформації користувача, що підвищує ризики витоків чи профілювання.

Узагальнений аналіз існуючих засобів свідчить, що, незважаючи на значний прогрес у сфері голосових асистентів, доступні рішення мають суттєві обмеження щодо персоналізації, прозорості роботи, інтеграції з IoT-платформами та можливості локального розширення функціональності. Саме це обумовлює потребу в розробці власної комп'ютерної системи голосового керування, яка б поєднувала переваги існуючих підходів, але при цьому вирішувала їх ключові недоліки, забезпечуючи більш гнучку, надійну та налаштовану взаємодію користувача з IoT-пристроями середовища.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Аналіз можливих рішень поставленого завдання

Поставлене завдання розробки комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями може бути реалізоване з використанням різних архітектурних, апаратних і програмних підходів. Вибір конкретного рішення повинен базуватися на аналізі вимог технічного завдання, обмежень апаратних ресурсів, необхідного рівня персоналізації, а також можливостей масштабування та подальшої модернізації системи.

Одним із можливих підходів є використання повністю хмарного рішення, при якому основні функції розпізнавання мовлення, обробки природної мови та генерації відповідей виконуються на віддалених серверах. У такому випадку локальний пристрій виконує роль інтерфейсу введення та виведення, а також забезпечує передачу даних через мережу Інтернет. Перевагами цього підходу є висока точність обробки мовлення, доступ до потужних мовних моделей і мінімальні вимоги до обчислювальних ресурсів локального обладнання. Водночас основними недоліками є залежність від стабільності мережевого з'єднання, затримки при обробці команд, а також підвищені ризики, пов'язані з безпекою та конфіденційністю даних користувача.

Альтернативним рішенням є побудова системи з переважно локальною обробкою даних, коли значна частина функцій виконується безпосередньо на мікроконтролерній або одноплатній обчислювальній платформі. Такий підхід дозволяє зменшити затримки, підвищити автономність роботи та знизити залежність від хмарних сервісів. Проте обмежені обчислювальні ресурси вбудованих систем значно ускладнюють використання сучасних великих мовних моделей, що негативно впливає на гнучкість і точність системи голосового керування. Крім того, реалізація складних алгоритмів обробки мовлення на локальному рівні потребує оптимізації та може підвищити складність програмного забезпечення.

Найбільш доцільним з точки зору балансу між функціональністю та ресурсними обмеженнями є гібридний підхід, який поєднує локальну та хмарну

					КС КРБ 123.143.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

обробку даних. У межах такого рішення попередня обробка голосових сигналів, керування IoT-пристроями та виконання простих команд здійснюються на локальному рівні, тоді як складна семантична обробка природної мови та генерація інтелектуальних відповідей реалізуються з використанням хмарних мовних моделей. Даний підхід дозволяє зменшити навантаження на мережу, забезпечити прийнятний час відгуку та водночас використовувати переваги сучасних технологій штучного інтелекту.

З точки зору апаратної реалізації можливими є рішення на базі мікроконтролерів із вбудованими модулями бездротового зв'язку або використання більш продуктивних одноплатних комп'ютерів. Мікроконтролерні платформи відзначаються низьким енергоспоживанням, компактністю та достатньою продуктивністю для IoT-застосувань, що робить їх доцільними для розробки прототипу системи. Одноплатні комп'ютери, у свою чергу, забезпечують вищу обчислювальну потужність, але мають більші вимоги до енергоживлення та вартості.

Аналіз програмних рішень показує, що доцільним є використання стандартних IoT-платформ і мережевих протоколів, які забезпечують сумісність, масштабованість та спрощують інтеграцію з іншими системами. Застосування модульної архітектури програмного забезпечення дозволяє адаптувати систему до різних сценаріїв використання та розширювати її функціональність без суттєвих змін базової структури.

Оптимальним підходом до реалізації поставленого завдання є гібридна архітектура з використанням мікроконтролерної платформи, IoT-технологій та інтеграції сучасних мовних моделей через хмарні сервіси. Такий підхід забезпечує необхідний рівень персоналізації, гнучкості та надійності системи й відповідає вимогам, сформульованим у технічному завданні.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		18

## РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Структура системи голосового керування IoT-пристроями

Структура комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями формується з урахуванням необхідності реалізації функцій голосової взаємодії з користувачем, обробки запитів із застосуванням сучасних мовних моделей, отримання даних із сенсорів навколишнього середовища та керування виконавчими пристроями. Узагальнена структурна схема системи наведена на рисунку 2.1.

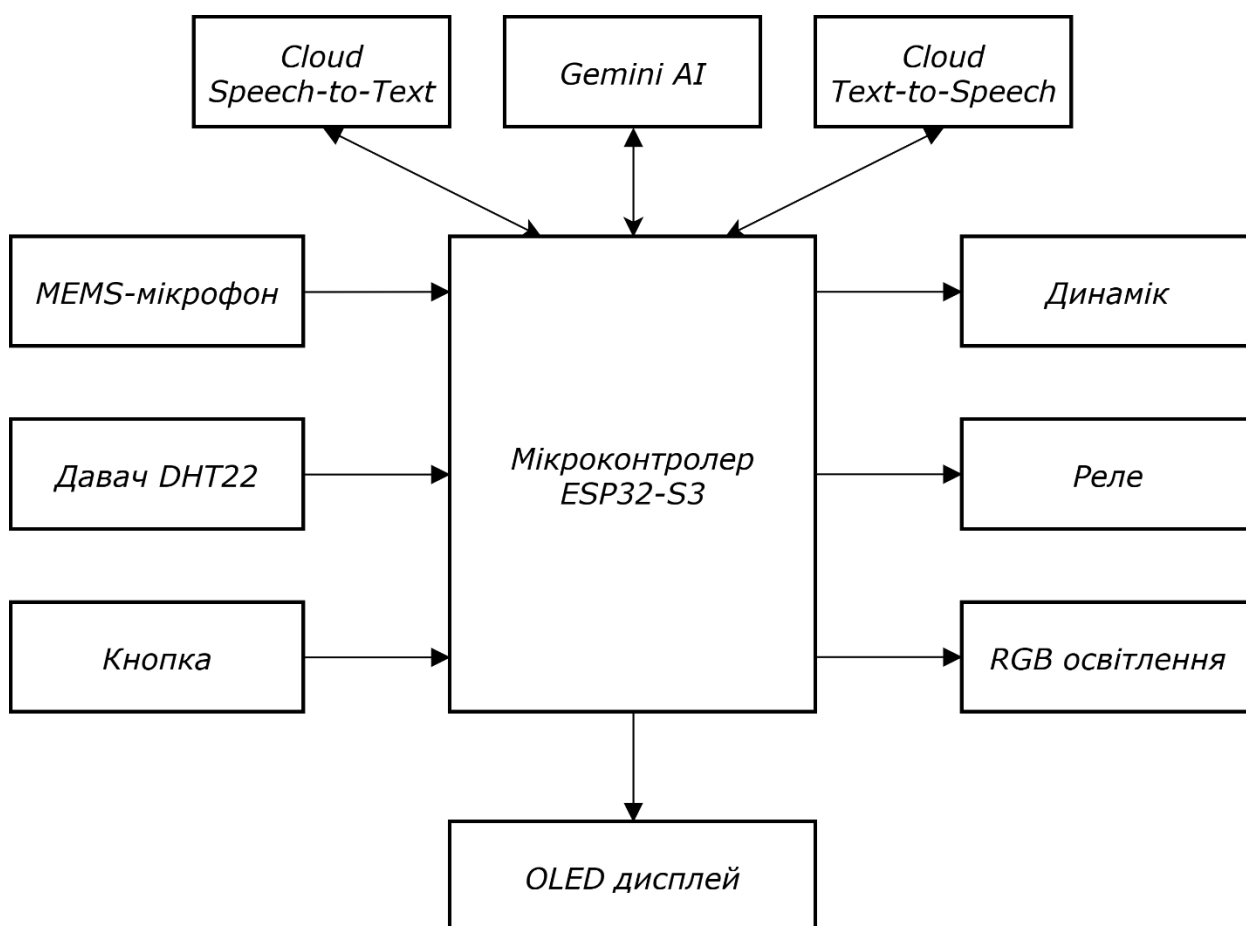


Рисунок 2.1 – Структурна схема системи голосового керування IoT-пристроями для осіб з обмеженими можливостями

					КС КРБ 123.143.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Бень О.Р.			Проектна частина	Літ.	Арк.	Акрушів
Перевірів		Луцик Н.С.					19	18
Рецензент		Деркач М.В.				ТНТУ, каф. КС, гр. СІ-41		
Н. Контр.		Тиш Е.В.						
Зав. каф.		Осужівська Г.М.						

На схемі відображено основні функціональні блоки системи та інформаційні зв'язки між ними. Центральним елементом системи є мікроконтролерний обчислювальний модуль, який виконує функції керування, обробки даних та координації взаємодії між усіма складовими системи. Саме цей модуль забезпечує збирання інформації з периферійних пристроїв, попередню обробку сигналів, організацію мережевої взаємодії та формування керуючих сигналів для виконавчих елементів. До мікроконтролерного модуля підключені пристрої введення, пристрої виведення інформації, сенсорні модулі та виконавчі пристрої.

Основним пристроєм введення голосової інформації у систему є цифровий мікрофонний модуль, який здійснює перетворення акустичних сигналів у цифровий аудіопотік. Сигнал від мікрофона передається до мікроконтролерного модуля через цифровий аудіоінтерфейс. Після отримання аудіосигналу мікроконтролер виконує його передачу через мережеве підключення до сервісу розпізнавання мовлення. У результаті роботи сервісу голосова команда перетворюється у текстову форму, яка надалі використовується для аналізу та інтерпретації.

Отриманий текстовий запит передається до сервісу обробки природної мови, що реалізований на основі сучасної мовної моделі. Цей сервіс виконує семантичний аналіз запиту користувача, визначає намір команди та формує відповідь або керуючу дію. У випадку інформаційного запиту формується текстова відповідь, яка повертається до мікроконтролерного модуля. Якщо ж запит містить команду керування пристроями, формується відповідний керуючий сигнал для виконавчих елементів системи.

Для відтворення голосових відповідей система використовує аудіовихідний тракт, що складається з цифрового аудіопідсилювального модуля та динамічного гучномовця. Мікроконтролер передає цифровий аудіосигнал на підсилювач через аудіоінтерфейс, після чого сигнал підсилюється і відтворюється через динамік. Це дозволяє реалізувати голосове спілкування користувача з асистентом та забезпечити природну форму взаємодії з системою.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Окрім голосового відтворення інформації, система передбачає також візуальне відображення даних. Для цього використовується графічний дисплей, підключений до мікроконтролера. На дисплеї можуть відображатися текстові повідомлення, результати обробки запитів, службова інформація про стан системи, а також дані, отримані від сенсорних модулів. Такий підхід забезпечує додатковий канал інформування користувача та підвищує зручність використання системи.

Для керування освітленням передбачено модуль світлодіодної індикації, який складається з кількох програмно керованих RGB-світлодіодів. До складу системи також входить сенсорний модуль вимірювання параметрів навколишнього середовища. Він призначений для отримання даних про температуру та відносну вологість повітря. Зчитування даних із сенсора здійснюється мікроконтролером у задані моменти часу або за запитом користувача. Отримані значення можуть використовуватися для формування інформаційних відповідей голосового асистента або відображатися на дисплеї.

Окрім сенсорів, система містить виконавчі модулі, які забезпечують можливість керування зовнішніми електричними пристроями. Для цього використовуються модулі реле, підключені до цифрових виходів мікроконтролера. За допомогою реле система може вмикати або вимикати різні побутові пристрої. Керування реле здійснюється на основі голосових команд користувача, які попередньо аналізуються мовною моделлю.

Для забезпечення взаємодії користувача із системою також передбачено елемент ручного керування у вигляді кнопки. Кнопка може використовуватися для активації режиму прослуховування голосової команди, запуску певних функцій або перезапуску системи. Наявність такого елемента дозволяє підвищити зручність експлуатації системи та забезпечити додатковий спосіб керування.

За допомогою бездротового мережевого підключення забезпечується обмін даними з хмарними сервісами обробки мовлення та мовною моделлю. Саме через мережевий канал здійснюється передача аудіоданих для розпізнавання мовлення, а також передача текстових запитів до сервісу обробки природної мови та отримання сформованих відповідей.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

## 2.2 Апаратне забезпечення системи голосового керування IoT-пристроями

### 2.2.1 Платформа ESP32-S3 DevKitC

Плата розробника ESP32-S3 DevKitC є сучасною апаратною платформою для створення вбудованих систем, IoT пристроїв та систем автоматизації. Вона побудована на базі мікроконтролера ESP32-S3, який поєднує високопродуктивний обчислювальний модуль із вбудованими засобами бездротового зв'язку. Завдяки поєднанню обчислювальних можливостей, великої кількості периферійних інтерфейсів та підтримки Wi-Fi і Bluetooth, плата широко застосовується у системах автоматизації, розумного дому, робототехніки та IoT-рішеннях (рис. 2.2).

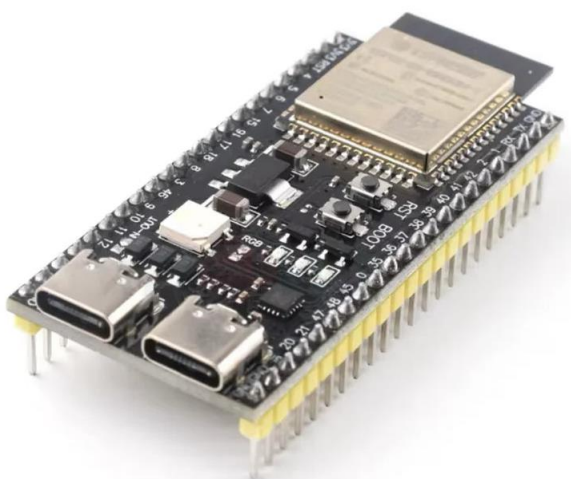


Рисунок 2.2 – Платформа ESP32-S3 DevKitC

Конструктивно платформа являє собою компактну друковану плату, на якій розміщено мікромодуль ESP32-S3, елементи живлення, інтерфейс USB, кварцовий резонатор, антену для бездротового зв'язку, а також виводи для підключення периферійних пристроїв. Мікромодуль містить основний мікроконтролер, оперативну та флеш-пам'ять, а також радіочастотну частину, що забезпечує роботу бездротових мереж. Для підключення до комп'ютера і програмування використовується інтерфейс USB Type-C, через який здійснюється передача даних і подача живлення. Наявність такого інтерфейсу значно спрощує процес розробки та налагодження програмного забезпечення.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Основою плати є мікроконтролер ESP32-S3 із двоядерною архітектурою. Завдяки двом обчислювальним ядрам система може одночасно виконувати декілька задач, наприклад обробку аудіосигналу, передачу даних через мережу та керування периферійними пристроями. У складі мікроконтролера також реалізовані апаратні прискорювачі для обробки сигналів і підтримки алгоритмів машинного навчання, що робить платформу придатною для застосування у системах із елементами штучного інтелекту.

Важливою складовою плати є підсистема пам'яті. Мікроконтролер містить внутрішню оперативну пам'ять SRAM, а також зовнішню флеш-пам'ять для зберігання програмного коду та даних. У деяких конфігураціях додатково використовується PSRAM, що дозволяє працювати з більшими обсягами даних. Наявність достатнього обсягу пам'яті дає змогу реалізувати складні алгоритми обробки інформації, зокрема роботу з аудіопотоками та обмін даними з хмарними сервісами. Характеристики платформи ESP32-S3 DevKitC наведено у таблиці 2.1.

Таблиця 2.1 – Характеристики платформи ESP32-S3 DevKitC

Характеристика	Значення
Мікроконтролер	ESP32-S3 (двоядерний Xtensa LX7)
Тактова частота	до 240 МГц
Оперативна пам'ять	близько 512 КБ SRAM
Flash-пам'ять	до 16 МБ
Додаткова пам'ять	до 8 МБ PSRAM
Бездротові інтерфейси	Wi-Fi 802.11 b/g/n, Bluetooth Low Energy
Кількість GPIO	до 45
Аналогові входи	12-бітні ADC (до 20 каналів)
Цифрові інтерфейси	UART, SPI, I <sup>2</sup> C, I <sup>2</sup> S, USB
Інтерфейси керування	PWM, RMT, таймери
Напруга живлення	5 В через USB або 3.3 В
Інтерфейс програмування	USB Type-C

Плата ESP32-S3 DevKitC має розвинений набір периферійних інтерфейсів, що забезпечує підключення різноманітних сенсорів, виконавчих пристроїв та інших електронних модулів. До них належать цифрові інтерфейси UART, SPI, I<sup>2</sup>C, I<sup>2</sup>S, а також USB-інтерфейс. Наявність інтерфейсу I<sup>2</sup>S є особливо важливою для систем, пов'язаних із обробкою аудіосигналів, оскільки він використовується для підключення цифрових мікрофонів та аудіопідсилювачів. Також у мікроконтролері передбачено аналогові входи ADC, модулі генерації широтно-імпульсної модуляції (ШІМ) та численні цифрові входи-виходи загального призначення (рис. 2.3).

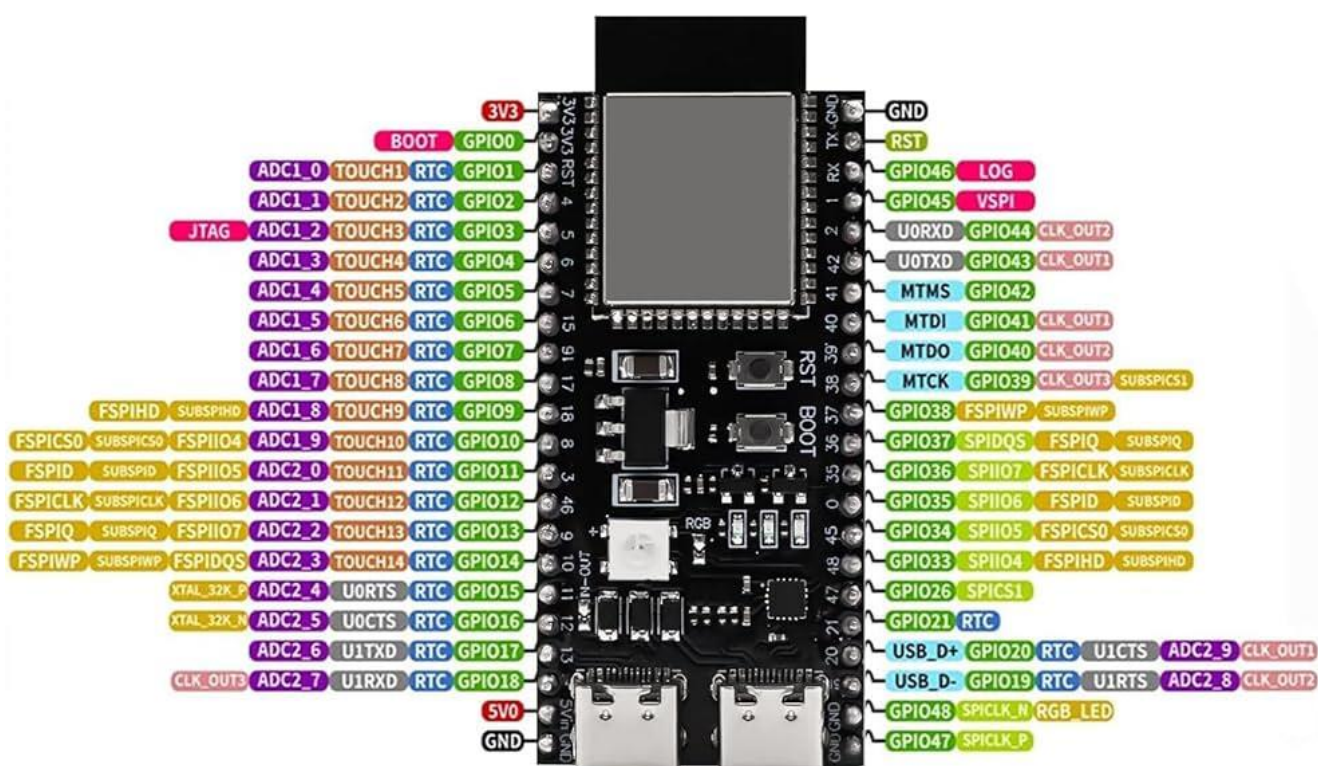


Рисунок 2.3 – Призначення виводів платформи ESP32-S3 DevKitC

Для забезпечення бездротового обміну даними у платі реалізовано підтримку стандартів Wi-Fi, а також BLE. Це дозволяє підключати пристрій до локальної мережі або безпосередньо до інших бездротових пристроїв. Завдяки цьому плата широко використовується у системах інтернету речей, де необхідний обмін даними з хмарними сервісами або мобільними пристроями.

Ще однією важливою особливістю ESP32-S3 DevKitC є підтримка режимів енергозбереження. Мікроконтролер може працювати у різних режимах, включаючи режим глибокого сну, у якому споживання струму може становити лише десятки мікроампер. Це дозволяє використовувати платформу у пристроях з автономним живленням або системах з обмеженим енергоспоживанням.

Вибір плати ESP32-S3 DevKitC для реалізації комп'ютерної системи голосового керування IoT-пристроями є технічно обґрунтованим. По-перше, мікроконтролер має достатню обчислювальну потужність для обробки аудіоданих, керування периферійними модулями та організації мережевого обміну з хмарними сервісами. По-друге, наявність інтерфейсу I<sup>2</sup>S дозволяє безпосередньо підключати цифровий MEMS-мікрофон і аудіопідсилювач, що необхідно для реалізації голосового інтерфейсу системи. По-третє, велика кількість цифрових входів-виходів дає можливість підключати сенсори, дисплеї, світлодіодні індикатори та релейні модулі без використання додаткових контролерів.

Крім того, підтримка бездротових технологій Wi-Fi забезпечує можливість взаємодії з хмарними сервісами обробки мовлення та мовними моделями, що є ключовим елементом функціонування системи голосового керування. Наявність широкої екосистеми програмних бібліотек і середовищ розробки також значно спрощує процес створення програмного забезпечення та інтеграції різних апаратних модулів.

Використання платформи ESP32-S3 DevKitC дозволяє створити функціонально завершену та масштабовану комп'ютерну систему голосового керування, що поєднує можливості локальної обробки даних із використанням хмарних сервісів штучного інтелекту, забезпечуючи при цьому достатню продуктивність, гнучкість та простоту інтеграції периферійних пристроїв.

### 2.2.2 Модуль MEMS-мікрофона INMP441

Модуль INMP441 є цифровим мікрофонним сенсором на основі технології MEMS, який призначений для високоякісного захоплення аудіосигналів у цифрових електронних системах. Завдяки компактним розмірам та наявності

					КС КРБ 123.143.00.00 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

цифрового інтерфейсу I2S цей модуль широко застосовується у системах розпізнавання мовлення, голосових асистентах, пристроях інтернету речей, робототехнічних системах та різноманітних аудіопристроях. Особливістю мікрофона є те, що він формує вже цифровий аудіопотік, що значно спрощує його інтеграцію з сучасними мікроконтролерами, такими як ESP32 (рис. 2.4).

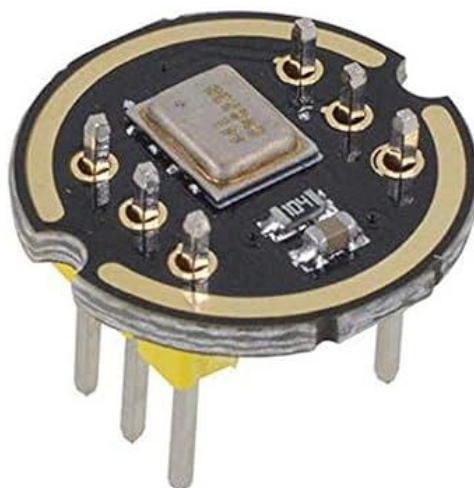


Рисунок 2.4 – Модуль MEMS-мікрофона INMP441

Конструктивно модуль складається з декількох основних частин: MEMS-сенсора, вбудованого аналого-цифрового перетворювача, цифрового інтерфейсу I2S та допоміжних електронних компонентів. Основним елементом є мікромеханічний сенсор, виконаний у вигляді мікроскопічної рухомої мембрани, розміщеної на кремнієвій підкладці. Під дією звукових хвиль мембрана здійснює коливання, що викликає зміну електричних параметрів сенсора. Ці зміни перетворюються у електричний сигнал, який надалі обробляється внутрішніми електронними схемами мікрофона.

У складі модуля також знаходиться вбудований АЦП, який перетворює аналоговий сигнал від сенсорної мембрани у цифровий аудіосигнал. Завдяки цьому на виході мікрофона формується 24-бітний цифровий потік даних, що передається через інтерфейс I2S. Використання цифрового формату значно підвищує стійкість системи до електромагнітних завад та зменшує вплив шумів, які можуть виникати під час передачі аналогового сигналу.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Принцип роботи модуля полягає у послідовному перетворенні акустичного сигналу у цифровий аудіопотік. Спочатку звукові хвилі потрапляють на чутливу мембрану MEMS-сенсора, що викликає її механічні коливання. Ці коливання перетворюються у електричний сигнал, який надалі оцифровується за допомогою внутрішнього АЦП. Отриманий цифровий сигнал передається до мікроконтролера через інтерфейс I2S у вигляді послідовного потоку даних.

Модуль має всенаправлену характеристику спрямованості, тобто він однаково добре сприймає звук з різних напрямків. Це є важливою перевагою для систем голосового керування, оскільки користувач може подавати голосові команди з різних положень відносно пристрою. Крім того, мікрофон характеризується високим відношенням сигнал/шум (SNR), що забезпечує якісне захоплення мовлення навіть у середовищах із помірним рівнем фонового шуму.

Ще однією важливою характеристикою є низьке енергоспоживання. Мікрофон споживає приблизно 1,4 мА струму, що дозволяє застосовувати його у портативних або енергоефективних пристроях. Модуль має широкий робочий температурний діапазон, що забезпечує стабільну роботу у різних умовах. Технічні характеристики модуля MEMS-мікрофона INMP441 наведено у таблиці 2.2.

Таблиця 2.2 – Характеристики модуля MEMS-мікрофона INMP441

Характеристика	Значення
Інтерфейс	I2S (Inter-IC Sound)
Напруга живлення	1,8 – 3,3 В
Споживаний струм	приблизно 1.4 мА
Формат вихідних даних	24-бітний цифровий аудіопотік
Відношення сигнал/шум (SNR)	близько 61 дБ
Чутливість	-26 dBFS
Частотний діапазон	60 Гц – 15 кГц
Максимальний рівень звуку	близько 120 dB SPL
Тип спрямованості	всенаправлений

Вибір модуля INMP441 для реалізації комп'ютерної системи голосового керування пристроями є технічно обґрунтованим. По-перше, мікрофон має цифровий інтерфейс I2S, який безпосередньо підтримується мікроконтролером ESP32-S3, що дозволяє підключати його без використання додаткових АЦП. По-друге, високий показник відношення сигнал/шум та достатньо широкий частотний діапазон забезпечують якісне захоплення мовлення користувача, що є критично важливим для систем розпізнавання голосових команд. По-третє, низьке енергоспоживання та компактні розміри модуля дозволяють легко інтегрувати його у компактні вбудовані пристрої. Крім того, модуль широко використовується у проєктах на базі ESP32 та інших мікроконтролерів, що забезпечує наявність великої кількості програмних бібліотек і прикладів реалізації.

Застосування MEMS-мікрофона INMP441 дозволяє реалізувати ефективну підсистему введення голосових команд у комп'ютерній системі голосового керування, забезпечуючи високу якість аудіозахоплення, простоту інтеграції з мікроконтролером та надійну роботу у складі IoT-пристрою.

### 2.2.3 Модуль аудіопідсилювача на MAX98357

Для реалізації підсистеми звукового виведення у комп'ютерній системі використовується модуль цифрового аудіопідсилювача на базі мікросхеми MAX98357, а також мініатюрний динамічний гучномовець з опором 8 Ом і потужністю 1 Вт. Така комбінація компонентів дозволяє реалізувати компактний та енергоефективний аудіовихід, необхідний для відтворення голосових повідомлень, відповідей асистента та інших звукових сигналів системи.

Модуль MAX98357 є цифровим монофонічним аудіопідсилювачем класу D, призначеним для роботи з цифровим аудіоінтерфейсом I2S. Його головною особливістю є те, що він одночасно виконує функції цифро-аналогового перетворювача (ЦАП) та підсилювача потужності, що дозволяє безпосередньо підключати його до мікроконтролера з інтерфейсом I2S без використання додаткових аудіокодеків (рис. 2.5).

					КС КРБ 123.143.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

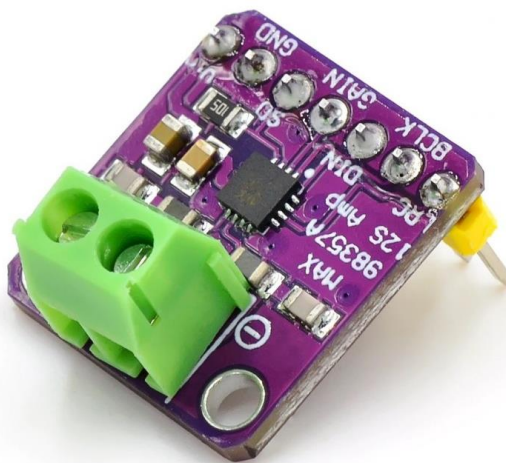


Рисунок 2.5 – Модуль аудіопідсилювача на MAX98357

Конструктивно модуль являє собою невелику друковану плату, на якій розміщено інтегральну мікросхему MAX98357, стабілізуючі та фільтрувальні конденсатори, резистори налаштування підсилення, а також роз'єми для підключення цифрових аудіосигналів і гучномовця. Зазвичай на платі передбачені контакти для підключення живлення, цифрових сигналів I2S та вихідні клеми для динаміка. Невеликі габарити модуля дозволяють легко інтегрувати його у компактні вбудовані системи.

Принцип роботи модуля базується на прийомі цифрового аудіосигналу через інтерфейс I2S. Мікроконтролер передає цифрові аудіодані у вигляді послідовного потоку через три основні сигнальні лінії: бітовий тактовий сигнал (BCLK), сигнал вибору слова або каналу (LRCLK) та лінію даних (DIN). Мікросхема MAX98357 приймає цей цифровий потік і виконує його декодування, після чого внутрішній ЦАП формує аналоговий аудіосигнал. Далі сигнал надходить на підсилювач класу D, який формує імпульсно-модульований сигнал великої потужності, придатний для безпосереднього підключення гучномовця.

Підсилювач належить до класу D, що означає використання імпульсного режиму роботи вихідного каскаду. У такому режимі транзистори працюють переважно у ключовому режимі, що дозволяє досягти високого коефіцієнта корисної дії та значно зменшити теплові втрати. Ефективність підсилювача може досягати приблизно 90 %.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Однією з важливих особливостей мікросхеми MAX98357 є можливість роботи без зовнішнього тактового сигналу MCLK, який часто використовується у цифрових аудіосистемах. Це значно спрощує підключення до мікроконтролерів, оскільки для передачі аудіоданих достатньо стандартних сигналів інтерфейсу I2S. Також мікросхема автоматично підтримує різні формати цифрового аудіо та частоти дискретизації від 8 кГц до 96 кГц. Технічні характеристики модуля аудіопідсилювача MAX98357 наведено у таблиці 2.3.

Таблиця 2.3 – Характеристики модуля аудіопідсилювача MAX98357

Характеристика	Значення
Тип аудіовходу	цифровий інтерфейс I2S
Кількість каналів	моно
Напруга живлення	2,5 – 5,5 В
Максимальна вихідна потужність	до 3,2 Вт (4 Ом, 5 В)
Потужність при навантаженні 8 Ом	приблизно 1,6 – 1,8 Вт
Частота дискретизації	8 – 96 кГц
Коефіцієнт гармонічних спотворень	близько 0,013 %
Ефективність	до 90 %
Відношення сигнал/шум	понад 89–100 дБ
Підтримувані коефіцієнти підсилення	3, 6, 9, 12, 15 дБ
Робочий температурний діапазон	-40...+80 °С

Вибір модуля MAX98357 для реалізації комп'ютерної системи голосового керування є технічно обґрунтованим. Він підтримує цифровий інтерфейс I2S, який безпосередньо реалізований у мікроконтролері ESP32-S3, що дозволяє передавати аудіосигнал без використання додаткових аудіокодеків або аналогових перетворювачів. Це значно спрощує апаратну структуру системи. Підсилювач класу D забезпечує високу енергоефективність і низький рівень тепловиділення. Модуль має невеликі габарити, просте підключення та підтримує стандартні напруги живлення 3,3–5 В.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання модуля аудіопідсилювача MAX98357 разом із компактним динаміком дозволяє реалізувати ефективну підсистему звукового виведення у комп'ютерній системі голосового керування, забезпечуючи достатню гучність, високу якість звуку та простоту інтеграції з мікроконтролерною платформою.

#### 2.2.4 OLED дисплей 0.96"

Для відображення текстової інформації, стану системи та результатів обробки голосових команд у комп'ютерній системі голосового керування використовується графічний OLED-дисплей. Такий дисплей є компактним модулем індикації, який широко застосовується у мікроконтролерних системах, пристроях інтернету речей, портативній електроніці та різноманітних вимірювальних приладах (рис. 2.6).



Рисунок 2.6 – OLED дисплей

Конструктивно модуль складається з декількох основних компонентів. До них належать OLED-матриця, драйвер дисплея, друкована плата з елементами керування та роз'єм для підключення до мікроконтролера. OLED-матриця формується з органічних світловипромінювальних діодів, розташованих у вигляді прямокутної сітки. Кожен елемент матриці відповідає окремому пікселю зображення, який може світитися при проходженні електричного струму через

					КС КРБ 123.143.00.00 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

органічний напівпровідниковий шар. Така структура дозволяє відображати текстову та графічну інформацію з високою контрастністю та чіткістю.

У більшості модулів OLED 0.96" використовується контролер SSD1306, який виконує функції керування матрицею дисплея та організовує взаємодію з мікроконтролером. Контролер містить внутрішню пам'ять кадру, у якій зберігається інформація про стан кожного пікселя. Мікроконтролер передає дані до дисплея у вигляді команд і графічних даних, після чого контролер формує сигнали керування для відповідних пікселів матриці. Завдяки цьому забезпечується відображення тексту, символів або простих графічних елементів.

Передача даних між мікроконтролером і дисплеєм у даному модулі здійснюється за допомогою інтерфейсу I2C. Такий підхід дозволяє значно зменшити кількість з'єднувальних проводів і спростити інтеграцію дисплея у мікроконтролерну систему.

Принцип роботи OLED-дисплея базується на електролюмінесценції органічних матеріалів. Коли через органічний шар проходить електричний струм, відбувається рекомбінація електронів і дірок, у результаті чого виділяється енергія у вигляді світла. Кожен піксель OLED-матриці може вмикатися або вимикатися незалежно від інших, що забезпечує високу контрастність зображення і можливість відображення чітких графічних елементів. Оскільки OLED-пікселі випромінюють світло самостійно, дисплей не потребує окремого джерела підсвітки, як це характерно для рідкокристалічних дисплеїв.

Дисплей має роздільну здатність 128×64 пікселі, що дозволяє відображати кілька рядків тексту, невеликі графічні елементи або індикатори стану системи. Завдяки невеликій діагоналі та компактним розмірам друкованої плати модуль легко інтегрується у портативні електронні пристрої. Крім того, OLED-дисплеї характеризуються дуже широким кутом огляду, який може перевищувати 160°, що забезпечує чітке зображення навіть при значному відхиленні від перпендикуляра до поверхні екрана.

Ще однією важливою перевагою OLED-дисплея є низьке енергоспоживання. Типовий робочий струм такого модуля становить приблизно 20–30 мА, що

					КС КРБ 123.143.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє застосовувати його у малопотужних мікроконтролерних системах. Завдяки цьому дисплей може працювати тривалий час без значного впливу на енергоспоживання пристрою. Основні технічні характеристики OLED-дисплея наведено у таблиці 2.4.

Таблиця 2.4 – Характеристики OLED-дисплея

Характеристика	Значення
Діагональ екрана	0.96"
Роздільна здатність	128 × 64 пікселі
Контролер	SSD1306
Інтерфейс зв'язку	I2C (ІС)
Кількість контактів	4 (VCC, GND, SDA, SCL)
Напруга живлення	3,3 – 5 В
Тип відображення	монохромний
Кут огляду	понад 160°
Активна область дисплея	21,7 × 10,8 мм
Споживаний струм	20–30 мА
Робоча температура	–30...+70 °С

Використання OLED-дисплея 0.96" є доцільним для реалізації комп'ютерної системи голосового керування. По-перше, дисплей має компактні розміри та невелике енергоспоживання, що дозволяє використовувати його у вбудованих пристроях. По-друге, інтерфейс I2C забезпечує просте підключення до мікроконтролера та потребує лише двох сигнальних ліній, що спрощує апаратну реалізацію системи.

Крім того, роздільна здатність 128×64 пікселі є достатньою для відображення текстових повідомлень, результатів обробки голосових команд, значень температури та вологості, а також службової інформації про стан системи. Висока контрастність забезпечує добру читабельність інформації у різних умовах освітлення.



Для приймання голосових команд користувача використовується цифровий MEMS-мікрофон INMP441, який працює через інтерфейс I2S. Лінія тактового сигналу SCK мікрофона підключається до порту GPIO4, який використовується як сигнал I2S\_SCK (BCLK). Контакт WS (LRCL) мікрофона підключається до GPIO5, що відповідає сигналу I2S\_WS, який визначає синхронізацію каналів. Лінія передачі даних SD підключається до GPIO6, що використовується як вхід I2S\_DATA. Контакт L/R підключається до GND, що визначає використання лівого каналу мікрофона.

Для відтворення звукових повідомлень використовується цифровий аудіопідсилювач MAX98357, який також працює через інтерфейс I2S. Сигнал BCLK підсилювача підключається до GPIO4, сигнал LRC — до GPIO5, а вхід цифрових аудіоданих DIN — до GPIO7 мікроконтролера, який використовується як вихід I2S\_DATA\_OUT. Вихід підсилювача SPK+ і SPK- підключається безпосередньо до мініатюрного динаміка, що забезпечує відтворення звукових відповідей голосового асистента.

Для відображення текстової інформації використовується OLED-дисплей. Лінія передачі даних SDA підключається до GPIO8, а тактова лінія SCL — до GPIO9 мікроконтролера ESP32-S3, які використовуються як шина I2C.

Для реалізації візуальної індикації режимів роботи системи використовується світлодіодне кільце з 16 RGB світлодіодами WS2812. Контакт DIN (вхід цифрового сигналу керування) підключається до GPIO45 мікроконтролера. Для забезпечення стабільної роботи між лінією даних та входом світлодіодного модуля встановлюється послідовний резистор номіналом 330 Ом, який зменшує рівень імпульсних завад.

Для контролю параметрів мікроклімату використовується цифровий давач температури і вологості DHT22. Контакт DATA підключається до GPIO37 мікроконтролера. Між контактами DATA та VCC встановлюється підтягувальний резистор номіналом 10 кОм, який забезпечує стабільність передачі цифрового сигналу.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Для ручного керування системою передбачено кнопку, яка використовується для активації режиму прослуховування голосової команди. Один контакт кнопки підключається до GPIO36, а інший — до GND.

Для керування зовнішніми електричними пристроями у системі використовуються два модулі реле на 5 В. Керуючі входи IN1 і IN2 підключаються відповідно до портів GPIO35 і GPIO0 мікроконтролера. За допомогою цих реле система може вмикати або вимикати зовнішні електричні пристрої, наприклад побутову техніку.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		36

## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Алгоритм роботи системи голосового керування IoT-пристроями для осіб з обмеженими можливостями

Алгоритм роботи комп'ютерної системи голосового керування IoT-пристроями орієнтований на забезпечення зручної, інтуїтивної та доступної взаємодії користувача з електронним середовищем за допомогою голосових команд. Основною особливістю алгоритму є поєднання локальної обробки сигналів на мікроконтролері з використанням хмарних сервісів для розпізнавання мовлення та інтелектуальної обробки запитів, що дозволяє реалізувати широкий функціонал системи.

Робота системи починається з етапу ініціалізації, під час якого мікроконтролер ESP32-S3 налаштовує всі необхідні апаратні та програмні модулі. Зокрема, виконується ініціалізація інтерфейсів I2S для роботи з цифровим MEMS-мікрофоном і аудіопідсилювачем, інтерфейсу I2C для взаємодії з OLED-дисплеєм, а також GPIO-портів для підключення релейних модулів, кнопки керування та світлодіодного кільця WS2812. Додатково здійснюється ініціалізація датчика температури і вологості DHT22. Після цього система встановлює з'єднання з мережею Wi-Fi, що є необхідною умовою для доступу до хмарних сервісів обробки мовлення та генерації відповідей.

Після завершення ініціалізації система переходить у режим очікування взаємодії з користувачем (рис. 3.1). У цьому режимі постійно виконується опитування стану кнопки активації, а також може здійснюватися аналіз вхідного аудіосигналу для виявлення ключової фрази активації, наприклад «ОК БОТ». Паралельно система періодично зчитує значення температури та вологості з датчика DHT22 і відображає ці дані на OLED-дисплеї.

					<i>КС КРБ 123.143.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Бень О.Р.</i>			<i>Практична частина</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевірив</i>		<i>Луцик Н.С.</i>					<i>37</i>	<i>19</i>
<i>Рецензент</i>		<i>Деркач М.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Тиш Е.В.</i>						
<i>Зав. каф.</i>		<i>Осуківська Г.М.</i>						

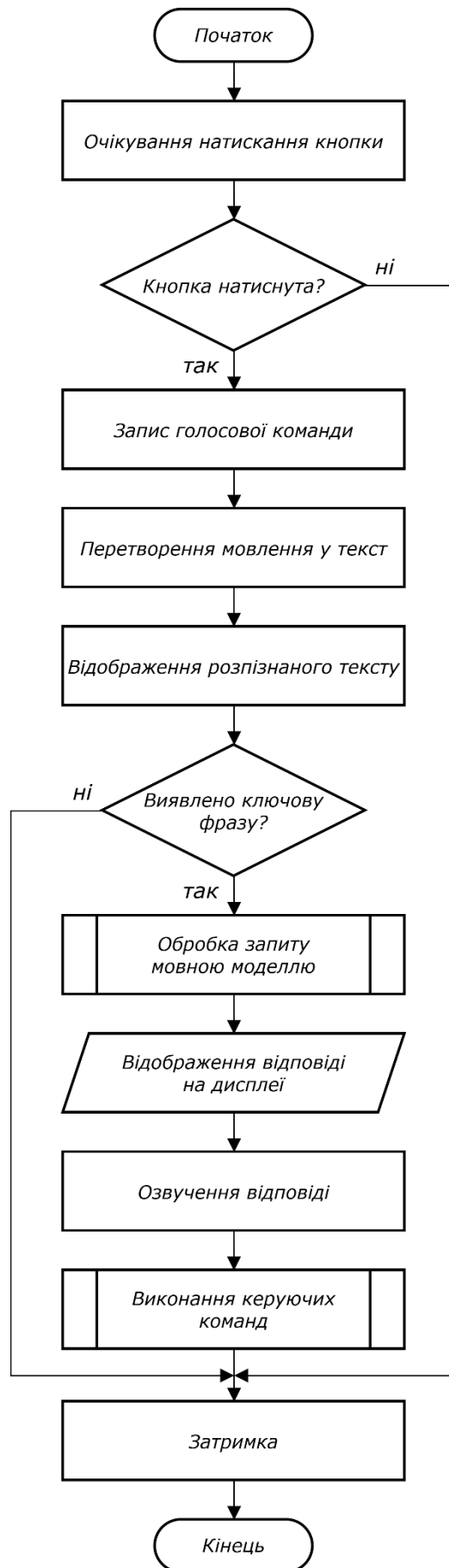


Рисунок 3.1 – Блок-схема алгоритму роботи системи голосового керування  
 IoT-пристроями

У разі спрацювання тригера, яким може бути натискання кнопки або розпізнавання ключової фрази, система переходить у режим обробки голосового запиту. На цьому етапі активується модуль запису аудіосигналу з MEMS-мікрофона, який здійснює перетворення звукових коливань у цифровий сигнал. Отримані аудіодані буферизуються та передаються до хмарного сервісу розпізнавання мовлення, де виконується їх перетворення у текстову форму.

Після отримання текстового представлення голосової команди система виконує попередній аналіз змісту запиту. На цьому етапі визначається, чи містить команда прямі інструкції для керування пристроями або отримання даних, чи потребує більш складної обробки. Якщо запит містить прості керуючі команди, наприклад увімкнення або вимкнення пристрою, система безпосередньо виконує відповідні дії шляхом зміни стану реле.

У випадку запиту інформації про температуру та вологість система зчитує актуальні значення з давача DHT22, формує текстову відповідь і відображає її на дисплеї. Одночасно ця інформація може бути озвучена користувачеві за допомогою аудіопідсистеми.

Якщо команда стосується керування освітленням, алгоритм обробляє її параметри, зокрема зміну кольору або рівня яскравості світлодіодного модуля WS2812. У залежності від змісту команди система встановлює відповідний колір, змінює яскравість або повністю вимикає підсвічування.

У разі складних або довільних запитів, які не належать до попередньо визначених команд, текст передається до хмарної мовної моделі для обробки. Сервіс аналізує запит і формує змістовну текстову відповідь, яка повертається до мікроконтролера. Отримана відповідь відображається на OLED-дисплеї та додатково перетворюється у мовний сигнал за допомогою сервісу синтезу мовлення, після чого відтворюється через динамік. Після завершення обробки запиту система повертається у режим очікування, що забезпечує можливість безперервної взаємодії з користувачем.

Така організація алгоритму дозволяє ефективно поєднувати локальні можливості мікроконтролера з обчислювальними ресурсами хмарних сервісів.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

## 3.2 Розробка програмного забезпечення

### 3.2.1 Ініціалізація апаратних та програмних компонентів системи

Підпрограма `setup()` є ключовим елементом програмного забезпечення мікроконтролера, оскільки саме в ній виконується початкова ініціалізація всіх апаратних і програмних ресурсів системи голосового керування IoT-пристроями (рис. 3.2).

```
void setup() {  
  Serial.begin(115200);  
  pinMode(RELAY_BOILER, OUTPUT);  
  pinMode(RELAY_AC, OUTPUT);  
  pinMode(BUTTON_PIN, INPUT_PULLUP);  
  dht.begin();  
  leds.begin();  
  leds.setBrightness(brightness);  
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
  WiFi.begin(ssid,password);  
  while(WiFi.status() != WL_CONNECTED) delay(500);  
  setupI2S();  
  showText("Assistant ready");  
}
```

Рисунок 3.2 – Лістинг коду процедури `setup()`

На початку підпрограми здійснюється налаштування режимів роботи цифрових виводів мікроконтролера. Це дозволяє програмно змінювати їх логічний стан і, відповідно, керувати підключеними електричними пристроями.

Наступним кроком є ініціалізація датчика температури і вологості шляхом виклику методу `dht.begin()`. Це забезпечує підготовку сенсора до зчитування фізичних параметрів навколишнього середовища, які у подальшому можуть використовуватися як для відображення, так і для обробки голосових запитів користувача.

Після цього виконується ініціалізація світлодіодного модуля WS2812. За допомогою виклику `leds.begin()` активується керування світлодіодами, а функція `leds.setBrightness()` встановлює початковий рівень яскравості, який визначається

					КС КРБ 123.143.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

змінною brightness. Це дозволяє одразу задати комфортний рівень освітлення для користувача та забезпечити індикацію стану системи.

Ініціалізація OLED-дисплея здійснюється викликом `display.begin()`, де вказується режим живлення та I2C-адреса пристрою. Після цього дисплей стає готовим до відображення текстової інформації, зокрема службових повідомлень, результатів вимірювань та відповідей голосового асистента.

Важливим етапом є підключення до бездротової мережі Wi-Fi за допомогою функції `WiFi.begin()`. Після ініціалізації з'єднання виконується цикл очікування, у якому система перевіряє статус підключення до мережі. Цикл триває доти, доки з'єднання не буде встановлено, що є необхідною умовою для подальшої взаємодії з хмарними сервісами розпізнавання мовлення та обробки запитів.

Після успішного підключення до мережі викликається підпрограма `setupI2S()`, яка виконує налаштування цифрового аудіоінтерфейсу I2S. Це забезпечує можливість роботи з MEMS-мікрофоном для запису голосових команд користувача та передачі аудіоданих до мікроконтролера.

На завершальному етапі виконується виклик функції, яка виводить на OLED-дисплей повідомлення про готовність системи до роботи. Це слугує візуальним підтвердженням успішної ініціалізації всіх компонентів та переходу системи у режим очікування взаємодії з користувачем.

### 3.2.2 Основний цикл обробки голосових запитів користувача

Підпрограма `loop()` реалізує основний робочий цикл системи, у межах якого забезпечується безперервна взаємодія користувача з голосовим асистентом. Саме в цій частині ПЗ відбувається обробка подій, ініційованих користувачем, зокрема активація системи, запис голосового запиту, його розпізнавання, інтерпретація та виконання відповідних дій.

На початку кожної ітерації циклу виконується перевірка стану кнопки активації. Якщо кнопка натиснута, система переходить до режиму прийому голосової команди. Цей підхід дозволяє уникнути постійного аналізу аудіосигналу і зменшує обчислювальне навантаження на мікроконтролер (рис. 3.3).

					КС КРБ 123.143.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void loop() {
  if(digitalRead(BUTTON_PIN)==LOW) {
    showText("Listening...");
    String audio = recordAudioBase64();
    String text = speechToText(audio);
    showText(text);
    // Wake word
    if(text.indexOf("ок бот") >= 0) {
      showText("Processing...");
      String answer = askGemini(text);
      showText(answer);
      speak(answer);
      handleCommands(text);
    }
    delay(3000);
  }
}

```

Рисунок 3.3 – Лістинг коду процедури loop()

Після виявлення натискання кнопки на OLED-дисплеї відображається повідомлення «Listening...» за допомогою функції showText(), що інформує користувача про готовність системи до запису голосового запиту. Далі викликається підпрограма recordAudioBase64(), яка здійснює запис аудіосигналу з мікрофона та кодує його у формат Base64 для подальшої передачі через мережу.

Отримані аудіодані передаються до функції speechToText(), яка забезпечує їх обробку за допомогою хмарного сервісу розпізнавання мовлення. У результаті виконання цієї функції формується текстове представлення голосової команди користувача. Отриманий текст одразу відображається на дисплеї, що дозволяє користувачу контролювати правильність розпізнавання.

Наступним етапом є перевірка наявності ключової фрази активації, зокрема «ок бот», за допомогою методу text.indexOf(). Такий механізм реалізує додатковий рівень підтвердження наміру користувача взаємодіяти із системою та запобігає випадковому виконанню команд. У разі виявлення ключової фрази система переходить до етапу обробки запиту. На дисплеї відображається повідомлення «Processing...», після чого викликається функція askGemini(), яка передає розпізнаний текст до хмарної мовної моделі для інтелектуального аналізу. У відповідь система отримує сформований текст, який є результатом обробки запиту.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Отримана відповідь відображається на OLED-дисплеї та одночасно відтворюється у вигляді мовлення за допомогою функції `speak(answer)`. Це забезпечує мультимодальну взаємодію з користувачем, що є особливо важливим для осіб з обмеженими можливостями.

Паралельно викликається функція `handleCommands(text)`, яка аналізує зміст початкового запиту користувача та, у разі наявності відповідних інструкцій, виконує керування підключеними пристроями. Зокрема, можуть змінюватися стани релейних модулів, параметри освітлення або здійснюватися інші дії, передбачені функціоналом системи.

На завершення ітерації циклу використовується затримка, яка обмежує частоту обробки запитів і запобігає повторному спрацюванню системи одразу після виконання попередньої команди. Після цього керування знову повертається на початок циклу, забезпечуючи безперервну готовність системи до нових запитів користувача.

### 3.2.3 Формування та кодування аудіоданих для передачі у хмарні сервіси

Функція `recordAudioBase64()` призначена для зчитування аудіосигналу з цифрового мікрофона через інтерфейс I2S, його попередньої обробки та підготовки до передачі у хмарні сервіси розпізнавання мовлення. Вона є важливим елементом системи, оскільки забезпечує перетворення фізичного звукового сигналу у формат, придатний для подальшої цифрової обробки (рис. 3.4).

```
String recordAudioBase64() {
    const int samples = 2048;
    int32_t buffer[samples];
    size_t bytesRead;
    i2s_read(I2S_NUM_0, buffer, sizeof(buffer), &bytesRead, portMAX_DELAY);
    String audio = "";
    for(int i=0;i<samples;i++) {
        audio += String(buffer[i]) + ",";
    }
    return audio;
}
```

Рисунок 3.4 – Лістинг коду функції `recordAudioBase64()`

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

На початку підпрограми оголошується константа `samples`, яка визначає кількість аудіосемплів, що будуть зчитані за один цикл. У даному випадку це значення становить 2048, що забезпечує достатню тривалість аудіофрагмента для розпізнавання коротких голосових команд. Далі створюється буфер `buffer` типу `int32_t`, у якому зберігатимуться зчитані значення аудіосигналу. Використання 32-бітного типу даних узгоджується з попередньо налаштованою розрядністю I2S-інтерфейсу.

Змінна `bytesRead` використовується для збереження фактичної кількості байтів, зчитаних з інтерфейсу I2S. Зчитування даних виконується за допомогою функції `i2s_read()`, яка отримує аудіопотік з апаратного буфера I2S та записує його у масив `buffer`. Параметр `portMAX_DELAY` вказує на те, що функція буде очікувати завершення операції зчитування без обмеження за часом, що гарантує отримання повного набору семплів.

Після успішного зчитування аудіоданих виконується їх перетворення у рядковий формат. Для цього створюється об'єкт `String audio`, який поступово наповнюється значеннями з буфера. У циклі `for` кожен елемент масиву `buffer` перетворюється у текстове представлення та додається до рядка з розділенням комами. Такий підхід дозволяє сформувати послідовність числових значень, що описують амплітуду аудіосигналу у дискретні моменти часу.

Отриманий рядок, який містить закодовані аудіодані, повертається як результат виконання підпрограми. Незважаючи на назву функції, у представленій реалізації використовується не класичне кодування Base64, а текстове представлення числових значень семплів. Це спрощує реалізацію, однак може призводити до збільшення обсягу передаваних даних, що варто враховувати при інтеграції з мережевими сервісами.

Підпрограма `recordAudioBase64()` забезпечує зчитування аудіосигналу з мікрофона, його буферизацію та перетворення у формат, придатний для подальшої передачі і обробки, що є необхідним етапом функціонування системи голосового керування.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		44

### 3.2.4 Інтеграція з сервісом розпізнавання мовлення та формування текстового запиту

Функція `speechToText()` реалізує взаємодію мікроконтролера з хмарним сервісом розпізнавання мовлення, забезпечуючи перетворення аудіосигналу у текстову форму. Вона є одним із ключових компонентів системи голосового керування, оскільки саме на цьому етапі відбувається інтерпретація голосових команд користувача (рис. 3.5).

```
String speechToText(String audio) {
    HttpClient https;
    String url = "https://speech.googleapis.com/v1/speech:recognize?key="
    | | | | | | | + GOOGLE_SPEECH_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(4096);
    doc["config"]["encoding"] = "LINEAR16";
    doc["config"]["sampleRateHertz"] = 16000;
    doc["config"]["languageCode"] = "uk-UA";
    doc["audio"]["content"] = audio;
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
    return response;
}
```

Рисунок 3.5 – Лістинг коду функції `speechToText()`

На початку підпрограми створюється об'єкт `HttpClient https`, який використовується для організації HTTP-запитів до віддаленого сервера. Далі формується URL-адреса запиту, що включає кінцеву точку сервісу розпізнавання мовлення та унікальний API-ключ `GOOGLE_SPEECH_KEY`, необхідний для автентифікації. Виклик функції `https.begin()` ініціалізує з'єднання з відповідним сервером.

Після встановлення з'єднання до запиту додається заголовок `Content-Type` зі значенням `application/json` за допомогою методу `https.addHeader()`. Це вказує серверу, що тіло запиту буде передано у форматі JSON.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Наступним етапом є формування JSON-документа, який містить параметри розпізнавання та аудіодані. Для цього використовується об'єкт `doc` з виділеним обсягом пам'яті 4096 байт. У блоці `config` задаються основні характеристики аудіосигналу: формат кодування `LINEAR16`, частота дискретизації 16000 Гц та мовний код `uk-UA`, що відповідає українській мові. Це дозволяє сервісу коректно інтерпретувати переданий аудіопотік.

У блоці `audio` формується поле `content`, в яке передається змінна `audio`, що містить закодовані аудіодані, отримані з попередньої підпрограми. Після заповнення структури JSON вона перетворюється у рядок за допомогою функції `serializeJson()`, що формує готове тіло HTTP-запиту.

Передача даних на сервер здійснюється методом `https.POST()`, який відправляє сформований JSON-документ. Після цього виконується отримання відповіді від сервера за допомогою функції `https.getString()`, яка повертає результат у вигляді рядка. У відповіді міститься JSON-структура з результатами розпізнавання мовлення, включаючи розпізнаний текст та додаткові метадані.

Після завершення обміну даними викликається функція `https.end()`, яка закриває з'єднання та звільняє використані ресурси. Отриманий рядок відповіді повертається як результат виконання підпрограми і надалі може бути використаний для аналізу або передачі до мовної моделі.

Підпрограма `speechToText()` забезпечує повний цикл взаємодії з сервісом розпізнавання мовлення: від формування запиту до отримання результату, що є необхідною умовою реалізації голосового інтерфейсу в системі.

### 3.2.5 Інтеграція з мовною моделлю для інтелектуальної обробки запитів

Функція `askGemini()` призначена для організації взаємодії мікроконтролера з хмарною мовною моделлю, що використовується для інтелектуальної обробки текстових запитів користувача та генерації змістовних відповідей. Вона забезпечує реалізацію функцій голосового асистента, дозволяючи системі не лише виконувати прості команди, а й відповідати на довільні запитання (рис. 3.6).

					КС КРБ 123.143.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

String askGemini(String text) {
    HttpClient https;
    String url =
        "https://generativelanguage.googleapis.com/v1beta/models/gemini-pro:generateContent?key="
        + GEMINI_API_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(4096);
    doc["contents"][0]["parts"][0]["text"] = text;
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
    return response;
}

```

Рисунок 3.6 – Лістинг коду функції askGemini()

На початковому етапі підпрограми створюється об'єкт HttpClient https, який використовується для формування та відправлення HTTP-запитів. Далі формується рядок URL, що містить адресу API сервісу генерації контенту та ключ доступу GEMINI\_API\_KEY, необхідний для автентифікації запитів. Виклик функції https.begin() ініціалізує з'єднання з сервером хмарної платформи.

Після цього до HTTP-запиту додається заголовок Content-Type зі значенням application/json, що вказує на використання формату JSON для передачі даних. Наступним кроком є створення структури JSON-документа за допомогою об'єкта doc. У цьому документі формується вкладена структура відповідно до вимог API мовної моделі, де текстовий запит користувача передається у полі contents, що містить масив об'єктів з частинами, кожна з яких включає текст.

Після формування структури JSON виконується її перетворення у рядковий формат за допомогою функції serializeJson(). Отриманий рядок body містить повністю підготовлене тіло HTTP POST-запиту, яке відповідає специфікації API.

Передача запиту до сервера здійснюється методом https.POST(), який надсилає сформований JSON-документ до мовної моделі. У відповідь сервер повертає JSON-структуру, що містить згенеровану відповідь, а також додаткову інформацію про обробку запиту. Отримання цієї відповіді виконується за допомогою функції https.getString(), яка повертає результат у вигляді текстового рядка.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Після завершення обміну даними викликається функція `https.end()`, яка закриває HTTP-з'єднання та звільняє ресурси, використані під час виконання запиту. Отримана відповідь повертається як результат виконання підпрограми і надалі використовується для відображення на дисплеї та озвучення користувачеві.

Підпрограма `askGemini()` реалізує механізм інтеграції з хмарною мовною моделлю, що дозволяє значно розширити функціональні можливості системи, забезпечуючи інтелектуальну підтримку користувача та обробку складних запитів природною мовою.

### 3.2.6 Синтез мовлення та озвучення відповіді користувачу

Підпрограма `speak()` призначена для реалізації функції синтезу мовлення, тобто перетворення текстової відповіді системи у звуковий сигнал. Вона забезпечує голосову взаємодію з користувачем, що є критично важливим для підвищення доступності системи, зокрема для осіб з обмеженими можливостями зору або моторики (рис. 3.7).

```
void speak(String text) {
    HttpClient https;
    String url =
        "https://texttospeech.googleapis.com/v1/text:synthesize?key=" + GOOGLE_TTS_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(2048);
    doc["input"]["text"] = text;
    doc["voice"]["languageCode"] = "uk-UA";
    doc["audioConfig"]["audioEncoding"] = "MP3";
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
}
```

Рисунок 3.7 – Лістинг коду процедури `speak()`

На початковому етапі у підпрограмі створюється об'єкт `HttpClient https`, який використовується для організації HTTP-запиту до сервісу синтезу мовлення.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Далі формується URL-адреса запиту, що містить кінцеву точку API Google Text-to-Speech та ключ доступу GOOGLE\_TTS\_KEY. Виклик функції `https.begin()` ініціалізує з'єднання з віддаленим сервером.

Після встановлення з'єднання до запиту додається заголовок `Content-Type` зі значенням `application/json`, що визначає формат передаваних даних. Далі створюється JSON-документ за допомогою об'єкта `doc`, у якому формується структура запиту відповідно до вимог API. У блоці `input` передається текст, який необхідно озвучити, тобто параметр `text`, отриманий як аргумент підпрограми.

У блоці `voice` задається мовна конфігурація синтезу мовлення. Зокрема, параметр `languageCode` встановлюється у значення `uk-UA`, що забезпечує генерацію мовлення українською мовою. У блоці `audioConfig` визначається формат вихідного аудіо, де параметр `audioEncoding` встановлюється як `MP3`. Це дозволяє отримати стиснений аудіофайл, придатний для подальшого відтворення.

Після формування JSON-структури виконується її перетворення у рядок за допомогою функції `serializeJson()`. Отримане тіло запиту передається на сервер методом `https.POST(body)`, який ініціює процес синтезу мовлення на стороні хмарного сервісу.

У відповідь сервер повертає JSON-об'єкт, що містить закодований аудіосигнал у форматі `Base64`. Ця відповідь зчитується за допомогою функції `https.getString()` та зберігається у змінній `response`. Після цього викликається функція `https.end()`, яка завершує HTTP-сеанс і звільняє ресурси.

Варто зазначити, що у представленій реалізації підпрограма виконує лише отримання аудіоданих, але не містить етапу їх декодування та відтворення через аудіовихід мікроконтролера. Для повноцінної реалізації функції озвучення необхідно додатково реалізувати обробку `Base64`-кодованого аудіо та його передачу до цифро-аналогового перетворювача або I2S-аудіовиходу.

Підпрограма `speak()` забезпечує інтеграцію системи з сервісом синтезу мовлення та формує основу для реалізації голосового зворотного зв'язку, що суттєво розширює функціональні можливості розробленої системи.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.3 Реалізація взаємодії з сучасними мовними моделями для реалізації системи голосового керування IoT-пристроями

#### 3.3.1 Налаштування Google AI Studio

Для реалізації функцій обробки природної мови та розпізнавання голосових команд у розроблюваній системі було використано хмарні сервіси штучного інтелекту компанії Google, зокрема платформу Google AI Studio та сервіс розпізнавання мовлення Google Cloud Speech-to-Text. Використання цих інструментів дозволило реалізувати перетворення голосових команд користувача у текстову форму, а також інтелектуальну обробку запитів за допомогою сучасної мовної моделі Gemini AI.

На початковому етапі було виконано налаштування середовища Google AI Studio, яке використовується для роботи з генеративними мовними моделями. У середовищі було створено новий проєкт для розробки та тестування взаємодії системи з мовною моделлю Gemini. У межах цього проєкту було активовано доступ до API мовної моделі, що дозволило виконувати запити до сервісу безпосередньо з мікроконтролерної системи через мережеве з'єднання. Після активації API було згенеровано персональний API-ключ, який використовується для автентифікації запитів, що надсилаються з пристрою ESP32-S3 до хмарного сервісу (рис. 3.8).

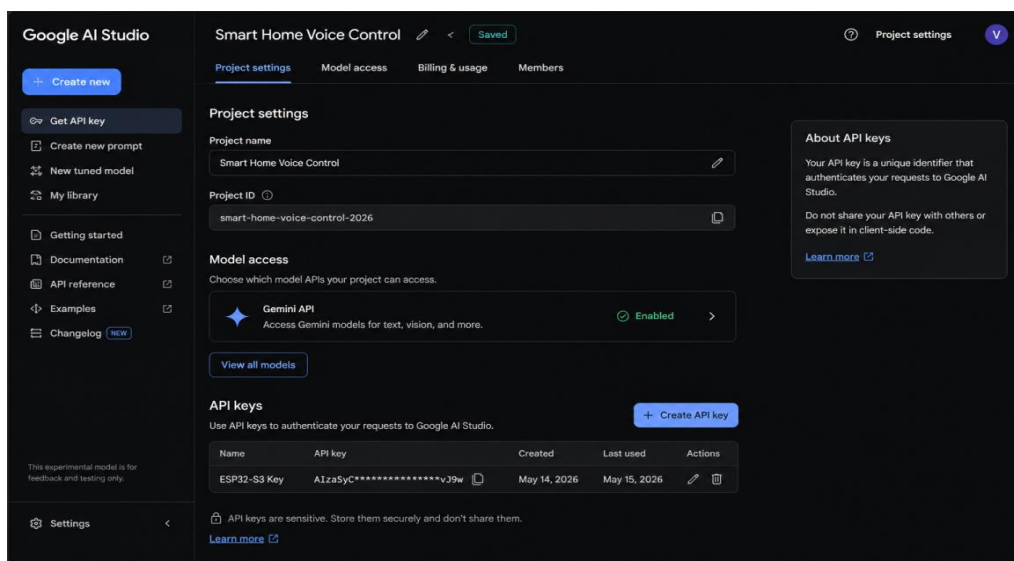


Рисунок 3.8 – Налаштування середовища Google AI Studio

					КС КРБ 123.143.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступним етапом було налаштування параметрів взаємодії з мовною моделлю. У середовищі Google AI Studio було протестовано декілька варіантів запитів для перевірки коректності обробки команд користувача. Під час тестування формувався текстовий запит, який передавався до моделі Gemini, після чого сервіс генерував відповідь у вигляді тексту. Отримані результати використовувалися для перевірки коректності формату JSON-запитів, а також для визначення оптимальної структури даних, які надсилатимуться з мікроконтролера.

Після успішного тестування у середовищі Google AI Studio було сформовано приклад HTTP-запиту для доступу до API мовної моделі. У цьому запиті використовувався метод POST, а у тілі запиту передавалася структура JSON, що містила текст команди користувача. API-ключ додавався до адреси запиту як параметр автентифікації. Отримана відповідь також мала формат JSON і містила текст, сформований мовною моделлю. Ця структура була використана під час розробки програмного забезпечення мікроконтролера ESP32-S3.

### 3.3.2 Налаштування Google Speech

Паралельно було виконано налаштування сервісу Google Cloud Speech-to-Text, який забезпечує перетворення голосових сигналів у текст. Для цього у хмарному середовищі Google Cloud було активовано відповідний API та налаштовано параметри доступу до нього. У налаштуваннях сервісу було вибрано режим обробки аудіоданих у форматі LINEAR16 з частотою дискретизації 16 кГц, що відповідає параметрам аудіосигналу, який отримується з цифрового MEMS-мікрофона INMP441 через інтерфейс I2S мікроконтролера ESP32-S3.

Для коректного розпізнавання голосових команд було також налаштовано мовні параметри сервісу. У конфігурації запиту було вказано код мови uk-UA, що дозволило системі розпізнавати українську мову. Крім того, було перевірено роботу системи для альтернативного варіанта en-US, що забезпечує можливість обробки англійськомовних голосових команд. Під час тестування використовувалися аудіофайли з прикладами голосових команд, які надсилалися до сервісу через HTTP-запити.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Після налаштування сервісу Google Speech-to-Text було сформовано структуру JSON-запиту, яка містить два основні блоки: конфігурацію розпізнавання та аудіодані. У блоці конфігурації задавалися параметри формату аудіо, частота дискретизації та мова розпізнавання. У блоці аудіоданих передавалася закодована аудіоінформація, отримана з мікрофона. У відповідь сервіс повертав JSON-структуру, що містила текстове представлення розпізаного мовлення.

Інтеграція обох сервісів у єдину систему здійснювалася таким чином: аудіосигнал, отриманий мікроконтролером від MEMS-мікрофона, передавався до сервісу Google Speech-to-Text, де виконувався процес розпізнавання мовлення. Отриманий текст команди надсилався до мовної моделі Gemini через API Google AI Studio. Після аналізу запиту мовна модель формувала текстову відповідь, яка поверталася до мікроконтролера та використовувалася для формування голосового повідомлення або виконання відповідної команди керування пристроями системи.

У результаті виконаних налаштувань було забезпечено повноцінну інтеграцію мікроконтролерної системи голосового керування IoT-пристроями з хмарними сервісами штучного інтелекту. Використання Google AI Studio та Google Speech-to-Text дозволило реалізувати функції розпізнавання голосових команд і інтелектуальної обробки природної мови, що значно розширило функціональні можливості розроблюваної системи та забезпечило високий рівень взаємодії користувача з пристроєм.

### 3.4 Результати моделювання та тестування системи

На завершальному етапі розробки комп'ютерної системи голосового керування IoT-пристроями було виконано моделювання її роботи, а також експериментальне тестування створеного фізичного прототипу. Проведення моделювання дозволило перевірити коректність побудови електричної схеми, взаємодію окремих апаратних компонентів та правильність обраних принципів підключення пристроїв до мікроконтролера. Для виконання цього етапу було

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

використано середовище Circuit Designer, яке призначене для розробки та моделювання електронних схем на базі мікроконтролерних платформ.

На початковому етапі в середовищі Circuit Designer було створено модель електричної принципової схеми розроблюваної системи. У схемі було розміщено основні компоненти системи. Після цього було виконано з'єднання між ними відповідно до розробленої електричної схеми (рис. 3.9).

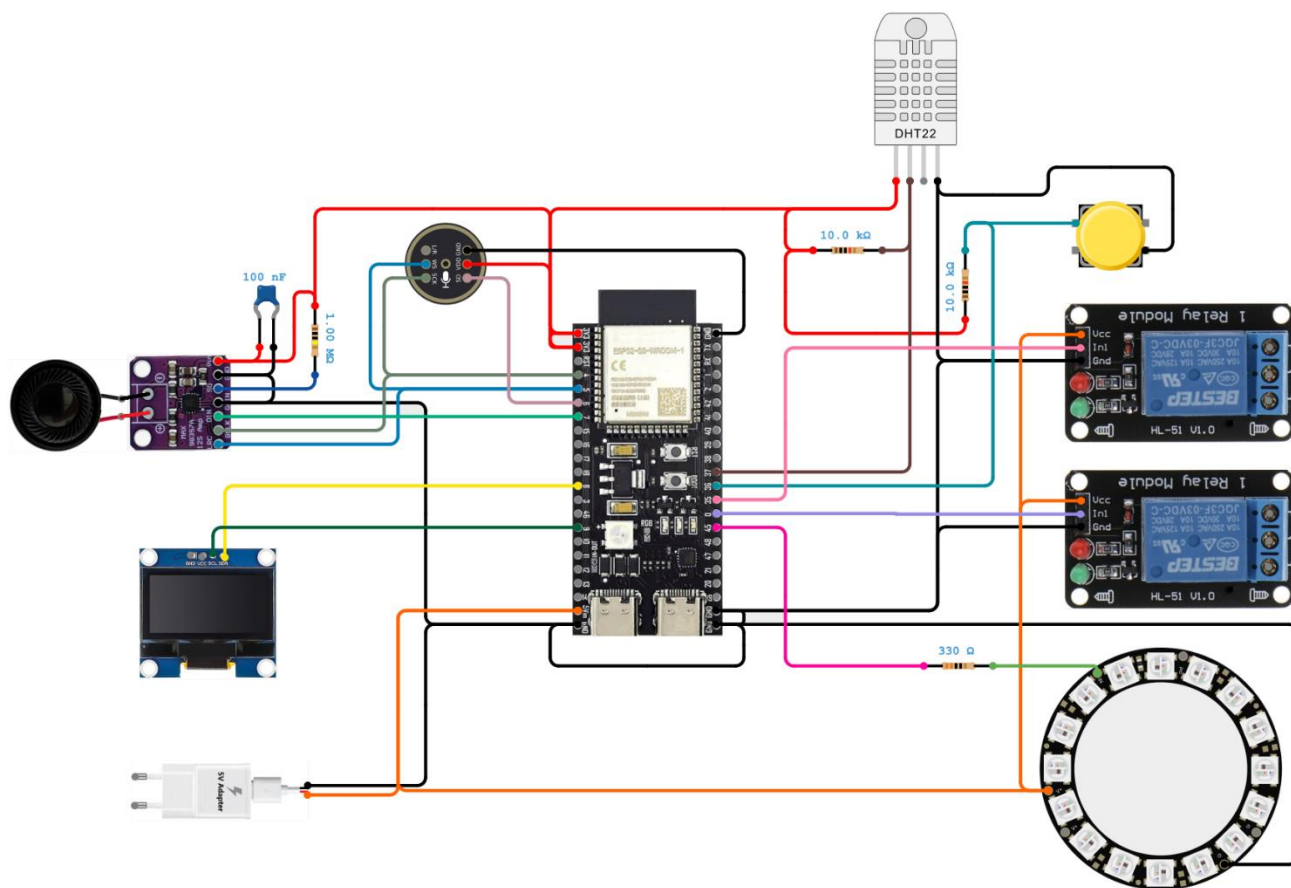


Рисунок 3.9 – Модель системи голосового керування IoT-пристроями

Після завершення побудови схеми було виконано її перевірку на наявність помилок підключення та конфліктів використання портів мікроконтролера. У процесі моделювання було встановлено, що обрана конфігурація підключення забезпечує коректну роботу всіх пристроїв та не викликає перевантаження окремих ліній живлення чи сигнальних каналів. Також було перевірено сумісність рівнів логічних сигналів між мікроконтролером ESP32-S3 та периферійними модулями.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Після завершення етапу моделювання було виконано складання фізичного прототипу системи на основі макетної плати. На цьому етапі всі апаратні компоненти були підключені до плати ESP32-S3 DevKitC відповідно до розробленої електричної схеми. Особлива увага приділялася правильності підключення інтерфейсу I2S для роботи з мікрофоном INMP441 та аудіопідсилювачем MAX98357A, оскільки ці модулі забезпечують основні функції голосової взаємодії користувача з системою.

Після складання апаратної частини було виконано завантаження розробленого ПЗ до пам'яті мікроконтролера. На початковому етапі тестування перевірялася коректність ініціалізації всіх підключених пристроїв. У ході перевірки було підтверджено стабільну роботу OLED-дисплея, який відображав службову інформацію про стан системи, а також коректне функціонування модуля світлодіодів WS2812.

Далі було виконано тестування сенсорної частини системи. Зокрема перевірялася робота датчика температури і вологості DHT22. Отримані дані коректно зчитувалися мікроконтролером та відображалися на дисплеї. Також було перевірено функціонування кнопки керування, яка використовувалася для активації режиму голосового запиту.

Наступним етапом стало тестування голосового інтерфейсу системи. Для цього користувач вимовляв голосову команду після натискання кнопки активації. Мікрофон INMP441 здійснював перетворення звукових коливань у цифровий аудіосигнал, який передавався до мікроконтролера через інтерфейс I2S. Отриманий аудіосигнал передавався через мережу Wi-Fi до сервісу розпізнавання мовлення Google Cloud Speech-to-Text, де виконувався процес перетворення мовлення у текст.

Після отримання текстової команди вона передавалася до мовної моделі Gemini AI через API платформи Google AI Studio. Мовна модель аналізувала зміст запиту та формувала текстову відповідь. Отримана відповідь передавалася назад до мікроконтролера та використовувалася для формування голосового повідомлення. Для відтворення звуку використовувався модуль аудіопідсилювача MAX98357A та мініатюрний динамік.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		54

Окремо було виконано тестування функцій керування зовнішніми пристроями. У межах цього тестування користувач подавав голосові команди для увімкнення або вимкнення пристроїв, підключених до релейних модулів. У результаті було підтверджено, що система коректно розпізнає команди керування та здійснює перемикання відповідних релейних виходів.

У процесі експериментального тестування також оцінювалася швидкість реакції системи на голосові запити. Було встановлено, що середній час обробки команди, включаючи передачу даних до хмарних сервісів і отримання відповіді, становить декілька секунд, що є прийнятним для систем такого класу.

Отримані результати моделювання та тестування підтвердили працездатність розробленої комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями. Система забезпечує коректне розпізнавання голосових команд, взаємодію з мовною моделлю, відтворення голосових відповідей та керування зовнішніми пристроями, що свідчить про досягнення поставленої мети кваліфікаційної роботи.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		55

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Характеристика життєдіяльності людини у системі «людина – машина – середовище існування»

Взаємодія людини з технікою й середовищем стає дедалі інтенсивнішою і складнішою, формуючи багатовимірну систему, де відбувається постійна передача інформації й енергії між її компонентами. Людина, машина (техніка) і навколишнє середовище розглядаються в ергономіці як складові й невід’ємні елементи єдиної системи «людина – машина – середовище існування».

У рамках системного підходу «людина – машина – середовище» ергономіка досліджує трудову діяльність людини у складі цієї системи й приділяє особливу увагу функціональній структурі системи «людина – машина». Такий підхід дозволяє не лише описати механіку взаємодії, але й виявити потенційні вузькі місця, де можуть виникати фізіологічні чи когнітивні навантаження.

Людина є одним з елементів зазначеної системи, в якій під терміном «людина» розуміється не лише одна істота, індивід, а й група людей, колектив, мешканці населеного пункту, регіону, країни, суспільство [33].

У центрі системи «людина – машина – середовище» знаходиться особа, яка сприймає інформацію через органи чуття, обробляє її в центральній нервовій системі й формує реакцію через дії. Саме цей процес охоплює фізіологічні, психологічні, когнітивні та соціальні аспекти життєдіяльності, що становлять основу безпеки й ефективності взаємодії. Інтерфейси й технічні засоби повинні бути ретельно спроектовані відповідно до фізичних можливостей людини – антропометричних, сенсорних, моторних, а також відповідно до її здатності обробляти складну інформацію під час роботи чи побуту.

					<i>КС КРБ 123.143.00.00 ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Бень О.Р.</i>			<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевірив</i>		<i>Луцик Н.С.</i>					56	6
<i>Консульт.</i>		<i>Сенчишин В.С.</i>				<i>ТНТУ, каф. КС, гр. СІ-41</i>		
<i>Н. Контр.</i>		<i>Тиш Е.В.</i>						
<i>Зав. каф.</i>		<i>Осуківська Г.М.</i>						

Принципи ергономіки, що охоплюють дизайн та модифікацію систем взаємодії, спрямовані на підвищення безпеки, комфорту і продуктивності. Ергономічні стандарти, закладені в ISO 9241 та інші нормативи, передбачають відповідність між можливостями людини та вимогами машинного інтерфейсу. Це гарантує, що структура, форма, розташування та інтуїтивність елементів управління відповідають потребам користувачів, зменшуючи ймовірність помилок чи травм.

Комплексні підходи до оцінки ризиків у таких системах дедалі частіше спираються на математичні моделі. Зокрема, застосування марковських процесів дозволяє моделювати динаміку небезпечних подій, враховуючи випадковість взаємодій людини з машиною і середовищем, що дає змогу передбачити критичні стани й упровадити превентивні заходи.

Механізм безпечної взаємодії включає врахування фізіологічних параметрів – зір, слух, моторика, втомлюваність, стрес. Сприймання звукових сигналів, зорових індикаторів, тактильного зворотного зв'язку повинно бути адекватно налаштоване: якщо індикатор надто яскравий або звук перевищує 80–85 дБА, це може викликати дискомфорт, втому або навіть слухову травму [33].

Під час розробки комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями особливої уваги потребує ергономіка сприймання звукових сигналів. Пристрій відтворює звуки в межах безпечного діапазону гучності, уникає високочастотних ультразвукових складових, а також забезпечує інтуїтивно зрозумілий інтерфейс управління. Надмірно гучні сигнали або погано продумана навігація спричиняють не лише дискомфорт, а й зростання когнітивного навантаження, що суперечить принципам безпечної взаємодії «людина – машина – середовище».

Центральна нервова система відповідає не лише за миттєву реакцію, але й за адаптацію – здатність пристосовуватися до змін навантаження, фільтрувати інформаційний шум і зберігати пильність навіть в умовах монотонної праці чи побуту. Зниження адаптаційних ресурсів, викликане дефіцитом сну, емоційним

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		57

перевантаженням або шкідливими чинниками середовища, збільшує ризик помилок, нещасних випадків або професійних захворювань.

Крім фізіології, важливими є психологічні аспекти: мотивація, увага, втома, рівень комфорту, взаємодія з іншими людьми. У високотехнологічному контексті, коли взаємодія з машинами відбувається через інтерфейси, кнопки, екрани, голосові помічники, виникає потреба враховувати когнітивну навантаженість і рівень довіри до системи. Помилки часто зумовлені не поганою технікою, а невідповідністю дизайну очікуванням користувача.

Необґрунтовано складний інтерфейс або незрозумілі інструкції можуть стати джерелом паніки у стресових умовах, особливо якщо система не передбачає чітких повідомлень про стан процесу або не підтримує оперативної допомоги. Сучасні стандарти HF/E спонукають вбудовувати кольорові коди, звукові сигнали, контекстну інструкцію, все для підтримки когнітивної безпеки [34].

Безпеку не забезпечує лише технічний дизайн – важливими є організаційні рішення. Вони включають навчання користувачів, моніторинг параметрів середовища (шум, освітлення, температура), періодичні огляди систем, оцінку ризиків і впровадження протоколів реагування у випадку надзвичайних ситуацій. Адекватна організаційна структура здатна зменшувати ймовірність виникнення аварій і підвищувати здатність системи до самовідновлення.

При цьому принцип системності «людина–машина–середовище» передбачає, що ризики мають розглядатися в комплексі: змінюється середовище – змінюється робота пристрою, людина може досі нормально реагувати – але умови можуть зумовити зрив взаємодії.

Система «людина – машина – середовище» – це динамічна модель, у якій безпека виходить за межі технічних характеристик і розширюється до врахування когнітивної, фізіологічної та емоційної взаємодії. Безпечна система – це така, що не лише мінімізує ризики, а й динамічно підтримує людину, спираючись на її потреби і стан у реальному часі.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

## 4.2 Шляхи збереження працездатності та підвищення продуктивності праці

Розумова праця об'єднує роботи пов'язані зі сприйняттям та опрацюванням інформації, необхідністю переважного навантаження сенсорного апарату, уваги, пам'яті, а також активації процесів мислення, емоційної сфери.

Виділяють такі різновиди розумової праці [34]:

- операторська;
- адміністративно-керівна;
- творча;
- праця викладачів і медичних працівників;
- праця учнів і студентів.

Вказані види роботи відрізняються щодо організації трудового процесу, рівномірності навантаження, ступеню емоційного напруження. При розумовій діяльності загострюється сприйняття, увага, пам'ять. Посилюється кровопостачання мозку, підвищується енергетичний обмін нервових клітин, змінюються показники біоелектричної активності мозку.

При інтенсивній інтелектуальній діяльності споживання кисню 100 г кори головного мозку в 5-6 разів більше, ніж споживання скелетного м'язу такої ж ваги при максимальному навантаженні. Розумова праця, а особливо, робота розробника комп'ютерних систем, супроводжується деякою нервово-емоційною напругою. Вона призводить до посилення серцево-судинної діяльності, дихання, енергообміну, підвищення м'язового тону.

Після закінчення розумової праці "робоча домінанта" повністю не згасає, зумовлюючи більш тривалу втому та виснаження ЦНС при розумовій праці, ніж при фізичній.

Практичне значення заходів щодо підвищення працездатності впливає із закономірностей її динаміки і зводиться до:

- збільшення фази стійкого стану в фонді робочого часу;
- прискорення процесу опрацювання;
- віддалення фази розвитку втоми;

					КС КРБ 123.143.00.00 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

– забезпечення високої продуктивності праці за нормальних фізіологічних затрат.

Комплекс заходів щодо підвищення і збереження працездатності працівників на оптимальному рівні реалізується на техніко-організаційному, соціально-економічному, санітарно-гігієнічному, медико-біологічному, психологічному напрямках.

Важливим фактором високої працездатності і продуктивності праці є оптимізація трудових навантажень на основі механізації і автоматизації виробничих процесів, удосконалення технології, скорочення і ліквідації важкої ручної праці. При правильній організації праці на легких роботах спостерігається найбільша тривалість фази стійкого стану, а на важких роботах вона нетривала [35].

Високий рівень працездатності безпосередньо залежить від умов праці, оскільки поліпшення їх супроводжується зменшенням енергетичних затрат організму на подолання несприятливого впливу факторів виробничого середовища.

Важливим напрямком підвищення працездатності працюючих є ритмізація трудових процесів, оптимізація темпу роботи, а також раціоналізація трудових рухів на фізіологічній основі, що сприяє формуванню і закріпленню робочих динамічних стереотипів, а отже зменшенню м'язових і вольових зусиль. Ритмічна робота підвищує функціональні можливості організму, сприяє його тренуваності і забезпечує економізацію енергетичних затрат.

Економізація функціональних затрат досягається завдяки стійкій домінанті і автоматизму дій, що виключає зайві рухи, розсіювання уваги. Особливе значення для підтримання працездатності працівників на високому рівні має раціональний режим праці і відпочинку.

Дослідження показують, що впровадження раціонального режиму праці і відпочинку на підприємствах забезпечує підвищення продуктивності праці на 8 – 10 %, сприяє поліпшенню фізіологічного стану працівників (зменшується частота

					КС КРБ 123.143.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

пульсу в процесі роботи, підвищується м'язова витривалість в кінці зміни, покращується координація рухів) [35].

Високій працездатності працівників сприяє і раціоналізація робочих місць на основі врахування антропометричних, біомеханічних і психофізіологічних вимог, що обумовлює раціональну робочу позу, зменшення статичних навантажень, оптимізацію робочої зони та інформаційних потоків.

Висока працездатність забезпечується за рахунок використання факторів естетичного впливу на працюючих. Такими факторами є колір, світло, музика. Слід підкреслити значення функціональної музики, яка впливає на емоційну сферу людини, підвищує збудливість і лабільність центральної нервової системи. На початку роботи вона прискорює процес праці, а в кінці робочого дня зменшує суб'єктивне відчуття стомленості.

Вплив функціональної музики посилюється, якщо вона поєднується з фізичними вправами. Останні підвищують лабільність органів, які безпосередньо беруть участь у виконанні роботи, активізують роботу органів дихання і кровообігу.

Під час розробки системи голосового керування IoT-пристроями для осіб з обмеженими можливостями дотримувалися заходи щодо збереження працездатності та підвищення продуктивності праці.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
						61
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено комп'ютерну систему голосового керування IoT-пристроями для осіб з обмеженими можливостями, яка забезпечує можливість голосової взаємодії користувача з електронним пристроєм, обробку запитів природною мовою та керування підключеними побутовими пристроями.

У ході виконання роботи проведено аналіз предметної області та досліджено сфери застосування голосових асистентів у сучасних інформаційно-керуючих системах. Також було сформульовано вимоги до розроблюваної системи та проаналізовано існуючі технічні рішення, що дозволило визначити оптимальну архітектуру системи та обґрунтувати доцільність використання мікроконтролерної платформи ESP32-S3 та хмарних сервісів обробки мовлення.

Розроблено структурну та електричну принципову схеми системи, які включають мікроконтролерний модуль, цифровий MEMS-мікрофон, аудіопідсилювач з динаміком, OLED-дисплей, світлодіодний модуль індикації, датчик температури і вологості, кнопку керування та реле для підключення зовнішніх пристроїв.

У програмній частині було розроблено алгоритм роботи системи та створено ПЗ для мікроконтролера, яке забезпечує зчитування аудіосигналу з мікрофона, передачу даних до сервісу розпізнавання мовлення, обробку текстових запитів за допомогою мовної моделі та формування голосових або текстових відповідей.

Також було виконано моделювання електронної схеми системи та проведено експериментальне тестування фізичного прототипу пристрою. Результати тестування підтвердили працездатність розробленої системи, коректність взаємодії її апаратних та програмних компонентів, а також можливість ефективного використання голосового інтерфейсу для отримання інформації та керування підключеними пристроями.

					КС КРБ 123.143.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lopatovska I., Rink K., Knight I., Raines K., Cosenza K., Williams H., Martinez A. Talk to me: Exploring user interactions with the Amazon Alexa. *Journal of Librarianship and Information Science*, 2019. 51(4). P. 984-997.
2. Kambale A. Home automation using google assistant. *UGC care approved journal*, 2023. 32(1). P. 1071-1077.
3. Reis A., Paulino D., Paredes H., Barroso I., Monteiro M.J., Rodrigues V., Barroso J. Using intelligent personal assistants to assist the elderlies An evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri. In 2018 2nd international conference on technology and innovation in sports, health and wellbeing (TISHW), 2018. P. 1-5.
4. Iannizzotto G., Bello L.L., Nucita A., Grasso G.M. A vision and speech enabled, customizable, virtual assistant for smart environments. In 2018 11th International Conference on Human System Interaction (HSI), 2018. P. 50-56.
5. Branham S.M., Mukkath A.R. Reading between the guidelines: How commercial voice assistant guidelines hinder accessibility for blind users. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 2019. P. 446-458.
6. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
7. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 354 с.
8. Буров Є., Митник М. Комп'ютерні мережі. (у 2-х томах). Львів, Магнолія, 2018. 740 с.
9. Strembitskyi M., Yavorska M., Palamar A., Kochan R., Yeromenko V. A comparative study of bug algorithms for robot navigation. *CEUR Workshop Proceedings, 3rd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2025)*, Ternopil, Ukraine, June 11-12, 2025. Vol. 4057 P. 312-321.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

10. Lupenko S., Lutsyk N., Yasniy O., Sobaszek Ł. Statistical analysis of human heart rhythm with increased informativeness. *Acta mechanica et automatica*. 2018. Vol. 12. P. 311-315.

11. Shabliy N., Lupenko S., Lutsyk N., Yasniy O., Malyshevska O. Keystroke dynamics analysis using machine learning methods. *Applied Computer Science*. 2021. Vol. 17, No. 4. P. 75-83.

12. Velychko D., Osukhivska H., Palaniza Y., Lutsyk N., Sobaszek L. Artificial Intelligence Based Emergency Identification Computer System. *Advances in Science and Technology Research Journal*, 18 no. 2, 2024, P. 296-304.

13. Zozulia A., Lytvynenko I., Lutsyk N., Lupenko S., Yasniy O. Method of vector rhythmcardiosignal automatic generation in computer-based systems of heart rhythm analysis. *Visnyk of TNTU*. 2020. Vol. 97, P. 122-132.

14. Yatsyshyn V., Pastukh O., Kukharska V., Palamar A., Kulikov S. Method and tool of detecting software architecture patterns in the process of computer systems development. *CEUR Workshop Proceedings, 4th International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2024)*, Ternopil, Ukraine, Opole, Poland, October 23-25, 2024. Vol. 3896. P. 12-24.

15. Palamar M., Yavorska M., Palamar A., Strembitskyi M. Modeling and Research of Satellite Antenna Adjustment Process for Earth Remote Sensing. *2022 IEEE 2nd Ukrainian Microwave Week (UkrMW)*, Kharkiv, Ukraine, November 14-18, 2022. P. 317-320.

16. Yasniy O., Lutsyk N., Demchyk V., Osukhivska H., Malyshevska O. The prediction of structural properties of Ni-Ti shape memory alloy by the supervised machine learning methods. *CEUR Workshop Proceedings, 3rd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2023)*, Ternopil, Ukraine, Opole, Poland, November 22–24, 2023. Vol. 3628. P. 73–78.

17. Palamar A., Voloskyi V., Kramar O., Kramar T., Stankevych O., Yatsyshyn V. Information computer system with a virtual tour for cultural heritage preservation of the Zbarazh Castle Museum's exhibition hall. *CEUR Workshop Proceedings, The 3rd International Workshop on Social Communication and Information Activity in Digital Humanities (SCIA 2024)*, Lviv, Ukraine, October 31, 2024. Vol. 3851.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		64

18. Palamar M., Nakonetchnyi Y., Palamar A., Strembitskyi M., Apostol Y. Modernization of the azimuth drive design for the antenna system. Scientific Journal of TNTU, Ternopil, Ukraine, 2025. Vol. 117, No 1, P. 54–61.

19. Palamar M., Horyn T., Palamar A., Batuk V. Method of calibration MEMS accelerometer and magnetometer for increasing the accuracy determination angular orientation of satellite antenna reflector. Scientific Journal of TNTU, Ternopil, Ukraine, 2022. Vol. 108, No 4. P. 79–88.

20. Palamar A., Stadnyk M., Palamar M. Adaptive PID regulation method of uninterruptible power supply battery charge current based on artificial neural network. Scientific Journal of TNTU, Ternopil, Ukraine, 2022. Vol. 107, No 3. P. 5–13.

21. Palamar M., Pasternak Y., Palamar A., Poikhalo A. Precision tracking of the trajectory LEO satellite by antenna with induction motors in the control system. Proceedings of the 2017 IEEE 9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2017), Bucharest, Romania, September 21–23, 2017. Vol. 2. P. 1051–1055.

22. Романов Д.В., Осухівська Г.М., Паламар А.М. Система управління зовнішнім освітленням на основі Інтернету речей. Актуальні задачі сучасних технологій : збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24-25 листопада 2021 року), Тернопіль: ТНТУ, 2021. С. 120.

23. Palamar A. Intelligent control and monitoring module for uninterruptible power supply system. II International Scientific and Practical Conference «Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs» (MC&FPGA-2020), Kharkiv, Ukraine. 2020. P. 12-13.

24. Palamar A., Pettai E. Microgrid for the Department of Electrical Drives and Power Electronics. 8th International Symposium "Topical Problems in the Field of Electrical and Power Engineering" and "Doctoral School of Energy and Geotechnology II" (January 11-16, 2010), Pärnu, Estonia, 2010. P. 54-61.

25. Palamar A. Methods and means of increasing the reliability of computerized modular uninterruptible power supply system. Scientific Journal of TNTU, Ternopil, Ukraine, 2020. Vol. 99, No 3. P. 133–141.

					<i>КС КРБ 123.143.00.00 ПЗ</i>	<i>Арк.</i>
						65
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

26. Voloshchuk A., Velychko D., Osukhivska H., Palamar A. Computer system for energy distribution in conditions of electricity shortage using artificial intelligence. CEUR Workshop Proceedings, 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024), Ternopil, Ukraine, June 12-14, 2024. Vol. 3742 P. 66-75.

27. Palamar A., Palamar M. Fire Safety Monitoring System Based on Internet of Things. CEUR Workshop Proceedings, 2023. 1st International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2023), Ternopil, Ukraine, June 14-16, 2023. Vol. 3468. P. 164-172.

28. Palamar A., Karpinskyy M. Control of an Uninterruptible Power Supply in a DC Microgrid System. 10th International Symposium "Topical Problems in the Field of Electrical and Power Engineering" and "Doctoral School of Energy and Geotechnology II" (January 10-15, 2011), Pärnu, Estonia, 2011. P. 80-84.

29. Паламар М., Пастернак Ю., Паламар А. Дослідження динамічних похибок системи прецизійного керування антеною з асинхронним електроприводом. Вісник ТНТУ, Тернопіль: ТНТУ, 2014. Вип. 76, № 4. С. 164–173.

30. Stadnyk M., Palamar A. Project management features in the cybersecurity area. Scientific Journal of TNTU, Ternopil, Ukraine, 2022. Vol. 106, No 2. P. 54–62.

31. Palamar A., Karpinskyy M., Vodovozov V. Design and Implementation of a Digital Control and Monitoring System for an AC/DC UPS. 7th International Conference-Workshop «Compatibility and Power Electronics» (CPE 2011), June 1-3, 2011. P. 173–177.

32. Погребенник В.Д., Клим Г.І., Бордун І.М., Пташник В.В., Паламар А.М. Системи оперативного контролю інтегральних параметрів водного середовища. Т. 2. Елементи комп'ютерних систем оперативного контролю: колективна монографія. Житомир: Видавничий дім «Бук-Друк», 2021. 180 с.

33. Желібо Є.П. Безпека життєдіяльності : підручник / В. В. Зацарний. Київ : Каравела, 2023. 344 с.

34. Стищенко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. Безпека життєдіяльності: навч. посіб. Харків: ХНРУЕ, 2018. 336 с.

35. Андрейчук Н.І., Кіт Ю.В., Шибанов С.В., Шерстньова О.В. Охорона праці : навч. посіб. Львів : Видавництво Львівська політехніка, 2021. 276 с.

					КС КРБ 123.143.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

Додаток А  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Кафедра комп'ютерних систем та мереж

«ЗАТВЕРДЖУЮ»

Завідувач кафедру КС

\_\_\_\_\_ Осухівська Г.М.

“ 2 ” лютого 2026 р.

КОМП'ЮТЕРНА СИСТЕМА ГОЛОСОВОГО КЕРУВАННЯ  
ІОТ-ПРИСТРОЯМИ ДЛЯ ОСІБ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ

ТЕХНІЧНЕ ЗАВДАННЯ

на  8  листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

\_\_\_\_\_ PhD Луцик Н.С.

“ 2 ” лютого 2026 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

\_\_\_\_\_ Бень О.Р.

“ 2 ” лютого 2026 р.

Тернопіль 2026

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи бакалавра: «Комп'ютерна система голосового керування IoT-пристроями для осіб з обмеженими можливостями».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.143.00.00.

### 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, Бень Олександр Романович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи бакалавра є наказ по університету № 4/9-188 від «24» квітня 2026 року.

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи бакалавра – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи бакалавра – 21.06.2026 р.

## 1.5 Порядок оформлення та пред'явлення результатів роботи

Оформлення технічної документації до кваліфікаційної роботи бакалавра здійснюється згідно діючих вимог вітчизняних та міжнародних стандартів. Технічна документація до кваліфікаційної роботи бакалавра включає в себе текст пояснювальної записки та креслення, які максимально інформативно та стисло відображають основні результати розробки комп'ютерної системи голосового керування IoT-пристроями для осіб з обмеженими можливостями. Основними регламентними документами при оформленні та пред'явленні результатів проектування є групи діючих стандартів ДСТУ, ISO, ЄСКД та ЕСПД. Пред'явлення результатів кваліфікаційної роботи бакалавра відбувається шляхом захисту роботи на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Комп'ютерна система голосового керування IoT-пристроями, отримання інформаційних повідомлень, виконання користувацьких команд та автоматизації типових сценаріїв взаємодії з «розумним» середовищем. Система орієнтована на використання у побутових, навчальних та експериментальних умовах, а також може застосовуватися як прототип для подальших досліджень і розширень.

Система забезпечує обробку природної мови, інтерпретацію голосових команд та взаємодію з хмарними сервісами й мовними моделями з урахуванням індивідуальних налаштувань користувача.

## 2.2 Мета створення системи

Метою створення системи є розробка функціонального, гнучкого та масштабованого пристрою для голосового керування IoT-пристроями для осіб з обмеженими можливостями з інтеграцією сучасних мовних моделей, здатного забезпечити персоналізовану взаємодію користувача з мережею вбудованих пристроїв.

Досягнення поставленої мети передбачає поєднання апаратних засобів збору та передачі даних, програмних алгоритмів обробки мовлення та хмарних сервісів для інтелектуального аналізу голосових запитів.

## 2.3 Характеристика об'єкту

Об'єктом автоматизації є середовище взаємодії користувача з IoT-пристроями, яке включає мікроконтролерну платформу, периферійні сенсори та виконавчі модулі, мережеву інфраструктуру та хмарні сервіси обробки даних. Користувач взаємодіє із системою за допомогою голосових команд, які обробляються локально або з використанням віддалених мовних моделей.

## 3 Вимоги до системи

### 3.1 Вимоги до системи в цілому

Система повинна забезпечувати стабільну роботу в реальному часі, підтримувати персоналізацію параметрів користувача та бути здатною до інтеграції з зовнішніми сервісами. Архітектура системи має бути модульною, що дозволяє розширювати її функціональні можливості без суттєвих змін базових компонентів.

До загальних вимог належать:

- надійність і відмовостійкість;
- простота конфігурації та використання;
- можливість оновлення програмного забезпечення;
- захист даних користувача.

### 3.1.1 Вимоги до структури та функціонування системи

Структурно система повинна складатися з апаратного рівня, програмного рівня та хмарного рівня. Апаратний рівень включає мікроконтролер, модулі введення голосових команд та засоби зв'язку. Програмний рівень забезпечує попередню обробку сигналів, керування пристроями та взаємодію з хмарними сервісами. Хмарний рівень відповідає за обробку мовлення, аналіз природної мови та генерацію відповідей.

Функціонування системи передбачає безперервний цикл:

- прийом голосової команди;
- її розпізнавання та інтерпретацію;
- формування керуючих впливів або інформаційних відповідей;
- передачу результатів користувачу або виконавчим пристроям.

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

Зв'язок між компонентами системи повинен здійснюватися з використанням сучасних мережевих технологій. Основним каналом обміну даними є бездротовий зв'язок на базі Wi-Fi. Для локальної взаємодії між модулями допускається використання інтерфейсів UART, I<sup>2</sup>C або SPI.

Для обміну даними з IoT-платформою та мовними моделями повинні застосовуватися стандартні мережеві протоколи, зокрема HTTP або MQTT. Передача даних має бути захищеною та забезпечувати цілісність інформації.

### 3.1.3 Вимоги до режимів функціонування системи

Система повинна підтримувати кілька режимів роботи, зокрема режим очікування, активний режим обробки команд та режим налаштування. В активному режимі система безперервно обробляє голосові команди користувача, а в режимі налаштування забезпечує зміну параметрів персоналізації.

Передбачено також режим тестування, який використовується для перевірки працездатності окремих компонентів та всієї системи в цілому.

### 3.1.4 Перспективи розвитку та модернізації системи

Архітектура системи повинна передбачати можливість подальшої модернізації. Це включає підключення додаткових IoT-пристроїв, інтеграцію з новими мовними моделями та розширення набору підтримуваних голосових команд.

Перспективним напрямом розвитку є впровадження адаптивних алгоритмів навчання, які дозволяють системі підлаштовуватися під стиль мовлення конкретного користувача.

### 3.1.5 Вимоги до надійності системи

Система повинна забезпечувати стабільну роботу протягом тривалого часу без втрати функціональності. У разі виникнення помилок або збоїв має бути передбачене автоматичне відновлення роботи або повідомлення користувача про несправність.

Надійність системи досягається за рахунок:

- використання перевірених апаратних компонентів;
- програмної обробки помилок;
- контролю стану основних модулів.

Показники надійності системи голосового керування IoT-пристроями для осіб з обмеженими можливостями повинні відповідати вимогам ДСТУ 50136-1. Ймовірність безвідмовної роботи системи повинна складати не менше 99,8 %.

### 3.1.6 Вимоги до функцій та задач, які виконує система

Система повинна виконувати такі основні функції:

- прийом і розпізнавання голосових команд;
- обробку природної мови з використанням мовних моделей;
- керування IoT-пристроями;
- формування голосових або текстових відповідей;
- збереження персоналізованих налаштувань користувача.

Окрім основних функцій, система повинна підтримувати логування подій та базові механізми моніторингу стану.

### 3.1.7 Вимоги до апаратного забезпечення

Апаратна частина системи повинна базуватися на сучасній мікроконтролерній платформі з достатнім обсягом оперативної та флеш-пам'яті. До складу апаратного забезпечення входять модуль захоплення звуку, засоби бездротового зв'язку та елементи індикації.

Апаратні компоненти мають забезпечувати:

- низьке енергоспоживання;
- компактність;
- можливість розширення функціоналу.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ.

Комплект конструкторської документації повинен складатись з:

- пояснювальної записки;

- графічного матеріалу:
  1. структурна схема системи;
  2. схема електрична принципова;
  3. блок-схема алгоритму роботи;
  4. результати моделювання системи.

\*Примітка: В комплект конструкторської документації можуть вноситися зміни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання КРБ

№ етапу	Назва етапу виконання КРБ	Термін виконання
1.	Розробка технічного завдання	26.01 – 02.02
2.	Аналіз технічного завдання, вимог до комп'ютерної системи, та можливих рішень поставленого завдання	03.02 – 15.02
3.	Розроблення структури, вибір апаратного забезпечення, проектування комп'ютерної системи	20.04 – 25.04
4.	Реалізація алгоритму, написання програмного забезпечення, моделювання комп'ютерної системи	26.04 – 05.05
5.	Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»	07.05 – 25.05
6.	Оформлення пояснювальної записки та графічного матеріалу	26.05 – 07.06
7.	Перевірка на академічний плагіат, перевірка керівником та консультантами	08.06 – 14.06
8.	Попередній захист кваліфікаційної роботи бакалавра	15.06 – 21.06
9.	Захист кваліфікаційної роботи бакалавра	22.06.2026

## 6 Додаткові умови виконання кваліфікаційної роботи бакалавра

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

Додаток Б  
Перелік елементів



## Додаток В

### Лістинг програми

Лістинг В.1 – Код програми мікроконтролера для реалізації системи голосового керування IoT-пристроями для осіб з обмеженими можливостями.

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_NeoPixel.h>
#include "DHT.h"
#include "driver/i2s.h"
#include "secret.h"

#define I2S_WS 5
#define I2S_SD 6
#define I2S_SCK 4
#define LED_PIN 45
#define LED_COUNT 16
#define DHT_PIN 37
#define BUTTON_PIN 36
#define RELAY_BOILER 35
#define RELAY_AC 0

Adafruit_SSD1306 display(128, 64, &Wire, -1);
Adafruit_NeoPixel leds(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
DHT dht(DHT_PIN, DHT22);
WiFiClientSecure client;
int brightness = 50;
uint32_t currentColor;

void setup() {
  Serial.begin(115200);
  pinMode(RELAY_BOILER, OUTPUT);
  pinMode(RELAY_AC, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  dht.begin();
  leds.begin();
  leds.setBrightness(brightness);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  WiFi.begin(ssid, password);
  while(WiFi.status() != WL_CONNECTED) delay(500);
  setupI2S();
  showText("Assistant ready");
}

void loop() {
```

```

if(digitalRead(BUTTON_PIN)==LOW) {
    showText("Listening...");
    String audio = recordAudioBase64();
    String text = speechToText(audio);
    showText(text);
    // Wake word
    if(text.indexOf("ок бог") >= 0) {
        showText("Processing...");
        String answer = askGemini(text);
        showText(answer);
        speak(answer);
        handleCommands(text);
    }
    delay(3000);
}
}

void setupI2S() {
    i2s_config_t config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
        .sample_rate = 16000,
        .bits_per_sample = I2S_BITS_PER_SAMPLE_32BIT,
        .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_I2S,
        .dma_buf_count = 8,
        .dma_buf_len = 128
    };
    i2s_pin_config_t pin_config = {
        .bck_io_num = I2S_SCK,
        .ws_io_num = I2S_WS,
        .data_out_num = -1,
        .data_in_num = I2S_SD
    };
    i2s_driver_install(I2S_NUM_0, &config, 0, NULL);
    i2s_set_pin(I2S_NUM_0, &pin_config);
}

String recordAudioBase64() {
    const int samples = 2048;
    int32_t buffer[samples];
    size_t bytesRead;
    i2s_read(I2S_NUM_0, buffer, sizeof(buffer), &bytesRead,
portMAX_DELAY);
    String audio = "";
    for(int i=0;i<samples;i++) {
        audio += String(buffer[i]) + ",";
    }
    return audio;
}

String speechToText(String audio) {
    HTTPClient https;
    String url =

```

```

"https://speech.googleapis.com/v1/speech:recognize?key=" +
GOOGLE_SPEECH_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(4096);
    doc["config"]["encoding"] = "LINEAR16";
    doc["config"]["sampleRateHertz"] = 16000;
    doc["config"]["languageCode"] = "uk-UA";
    doc["audio"]["content"] = audio;
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
    return response;
}

String askGemini(String text) {
    HTTPClient https;
    String url =
        "https://generativelanguage.googleapis.com/v1beta/models/gemini-
pro:generateContent?key="
    + GEMINI_API_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(4096);
    doc["contents"][0]["parts"][0]["text"] = text;
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
    return response;
}

void speak(String text) {
    HTTPClient https;
    String url =
        "https://texttospeech.googleapis.com/v1/text:synthesize?key=" +
GOOGLE_TTS_KEY;
    https.begin(url);
    https.addHeader("Content-Type", "application/json");
    DynamicJsonDocument doc(2048);
    doc["input"]["text"] = text;
    doc["voice"]["languageCode"] = "uk-UA";
    doc["audioConfig"]["audioEncoding"] = "MP3";
    String body;
    serializeJson(doc, body);
    https.POST(body);
    String response = https.getString();
    https.end();
}

```

```

void handleCommands(String text) {
  text.toLowerCase();
  // котел
  if(text.indexOf("увімкни котел") >= 0)
    digitalWrite(RELAY_BOILER, HIGH);
  if(text.indexOf("вимкни котел") >= 0)
    digitalWrite(RELAY_BOILER, LOW);
  // кондиціонер
  if(text.indexOf("увімкни кондиціонер") >= 0)
    digitalWrite(RELAY_AC, HIGH);
  if(text.indexOf("вимкни кондиціонер") >= 0)
    digitalWrite(RELAY_AC, LOW);
  // температура
  if(text.indexOf("температура") >= 0) {
    float t = dht.readTemperature();
    float h = dht.readHumidity();
    String msg = "Температура " + String(t) + " градусів, вологість
" + String(h);
    showText(msg);
    speak(msg);
  }
  // яскравість
  if(text.indexOf("яскравість") >= 0) {
    brightness += 20;
    if(brightness > 255) brightness = 255;
    leds.setBrightness(brightness);
    leds.show();
  }
  // вимкнути світло
  if(text.indexOf("вимкни світло") >= 0) {
    leds.clear();
    leds.show();
  }
  // кольори
  if(text.indexOf("червоний") >= 0) currentColor =
leds.Color(255,0,0);
  if(text.indexOf("зелений") >= 0) currentColor =
leds.Color(0,255,0);
  if(text.indexOf("синій") >= 0) currentColor = leds.Color(0,0,255);
  for(int i=0;i<LED_COUNT;i++)
    leds.setPixelColor(i,currentColor);
  leds.show();
}

void showText(String text) {
  display.clearDisplay();
  display.setCursor(0,0);
  display.setTextSize(1);
  display.println(text);
  display.display();
}

```