

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему КОМП'ЮТЕРНА СИСТЕМА ДИСТАНЦІЙНОГО КЕРУВАННЯ ЗА
ДОПОМОГОЮ ГОЛОСУ

Виконав: студент 2 курсу, групи КНТ-612м
спеціальності _____

123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Спеціалізовані комп'ютерні системи

МІНЯЙЛО П.І.

(ПРИЗВИЩЕ та ініціали)

Керівник КАСЬЯН К.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЩЕРБАК Н.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти магістерський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) «Спеціалізовані комп'ютерні системи»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.

“15” листопада 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

МІНЯЙЛО Павлу Івановичу

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Комп'ютерна система дистанційного керування за допомогою голосу.

керівник проєкту (роботи) к. т. н., доцент, КАСЬЯН Костянтин Миколайович

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

затверджені наказом вищого навчального закладу від “24” жовтня 2023 року № 400

2. Строк подання студентом проєкту (роботи) 10 грудня 2023 року

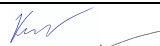
3. Вихідні дані до проєкту (роботи) _____

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

Аналіз сучасних «розумних» систем. Проєктування системи. Моделювання системи безпеки. Тестування та дослідження системи.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	КАСЬЯН К. М., доцент		
нормоконтроль	ЩЕРБАК Н. В., ст. викл.		

7. Дата видачі завдання 01.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз наявних методів та алгоритмів	10.10.2023 р.	
	використання голосу для		
	Керуванням пристроїв		
2	Розробка вимог для системи	15.10.2023 р.	
3	Вибір методів керування голосом	20.10.2023 р.	
4	Реалізація програмної системи	05.10.2023 р.	
5	Дослідження моделі безпечного середовища	20.11.2023 р.	
6	Оформлення отриманих результатів у ПЗ	25.11.2023 р.	
7	Оформлення графічного матеріалу	01.12.2023 р.	
8	Оформлення допоміжного матеріалу	10.12.2023 р.	

Студент  Павло МІНЯЙЛО
(підпис) (ім'я, ПРИЗВИЩЕ)

Керівник проєкту (роботи)  Костянтин КАСЬЯН
(підпис) (ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 70 с., 22 рис., 11 табл., 42 джерел.

РОЗУМНА КОЛОНКА, ГОЛОСОВЕ КЕРУВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ІНТЕРНЕТ РЕЧЕЙ.

Об'єкт дослідження – процеси голосового керування пристроями.

Предмет дослідження – методи та алгоритми розпізнавання та обробки мовних сигналів.

Мета – розробка комп'ютерної системи для розпізнавання мови з високою точністю та зручною у використанні.

Практичні результати: розроблено програмно-апаратний комплекс для голосового керування розумною колонкою.

Наукова новизна: застосовано інтегрований підхід до обробки природної мови поєднуючий хмарні сервіси та мікроконтролери.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування апаратні платформи STM32, Raspberry Pi, мови програмування Python, C++, API Google.

Результати. Розроблений програмно-апаратний комплекс дає змогу здійснювати голосове керування розумною колонкою.

Галузь вивикористання – малі домогосподарства.

ABSTRACT

Explanatory note to the master's work: 70 pages, 22 figures, 11 tables, 42 sources.

SMART SPEAKER, VOICE CONTROL, ARTIFICIAL INTELLIGENCE, INTERNET OF THINGS.

The object of research is processes of voice control of devices.

The subject of research is methods and algorithms for recognition and processing of speech signals.

The goal is to develop a computer system for speech recognition with high accuracy and ease of use.

Practical results: a hardware and software complex was developed for voice control of a smart speaker.

Scientific novelty: an integrated approach to natural language processing combining cloud services and microcontrollers is applied.

Materials, methods and technical means: object-oriented programming of hardware platform STM32, Raspberry Pi, programming languages Python, C++, Google API.

The results. The developed hardware and software complex enables voice control of the smart speaker.

The field of use is small households.

ЗМІСТ

Скорочення та умовні позначки	7
Вступ.....	8
1 Аналіз сучасних «розумних» систем	10
1.1 Аналіз пристроїв.....	10
1.2 Аналіз портативних акустичних систем	11
1.3 Інновації і тренди в розробці розумних колонок.....	13
1.4 Інтеграція з пристроями розумного дому.....	16
1.5 Автоматичне розпізнавання мови	18
1.6 Висновки за розділом.....	21
2 Проєктування системи.....	22
2.1 Розробка структури.....	22
2.2 Обґрунтування апаратної частини системи.....	27
2.3 Вибір програмного забезпечення системи	35
2.3 Висновки за розділом.....	44
3 Моделювання системи безпеки	45
3.1 Налаштування середовища	45
3.2 Серверна частина системи	50
3.3 Висновки за розділом.....	52
4 Тестування та дослідження системи	53
4.1 Тестування функцій системи	53
4.2 Дослідження сервісів розпізнавання мови	58
4.3 Висновки за розділом.....	61
Висновки	63
Перелік джерел посилання	65

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI	- Artificial Intelligence;
API	- Application Programming Interface;
IDE	- Integrated Development Environment;
IoT	- Internet of Things;
JSON	- JavaScript Object Notation;
MQTT	- Message Queuing Telemetry Transport;
NLP	- Natural Language Processing;
REST	- Representational State Transfer;
SOAP	- Simple Object Access Protocol;
UX	- User Experience;
VUI	- Voice User Interface.

ВСТУП

Сучасний світ неможливо уявити без впливу комп'ютерних технологій, які стали ключовою частиною повсякденного життя. Ці технології не тільки вдосконалюються з часом, але й перетворюють спосіб, яким спілкуємося та взаємодіємо з пристроями, такими як смарт-домашні системи та голосові помічники. Вони пропонують широкий спектр можливостей, від створення локальних мереж до розвитку швидкісної передачі даних та прогресивного програмного забезпечення [1].

У цьому світі, існує значний інтерес до розвитку як централізованих, так і децентралізованих систем, які пропонують різні підходи до обробки та передачі інформації. Централізовані системи забезпечують надійність та єдність управління, тоді як децентралізовані системи пропонують гнучкість та відсутність залежності від центрального сервера, що може зменшити витрати і збільшити ефективність [2].

Основною метою цієї кваліфікаційної роботи є розробка комп'ютерної системи для розпізнавання мови, призначеної для використання в "розумних" колонках. Цей проект передбачає створення структурної схеми та розробку з'єднань між елементами системи, що дозволить їй підключатися до мережі і відповідно взаємодіяти з користувачами [2].

Сьогоднішні технологічні досягнення дозволяють розробляти все більш інтелектуальні та взаємопов'язані системи. Розумні домашні пристрої, такі як розумні колонки та освітлювальні системи, не тільки полегшують повсякденне життя, але й пропонують нові можливості для створення комфортного та ефективного домашнього середовища [3].

Ключовим елементом у цьому дослідженні є інтеграція голосових команд та розпізнавання мови, які надають користувачам зручний спосіб управління своїми пристроями без необхідності фізичного взаємодії. Це не тільки підвищує ергономіку використання, але й відкриває двері для більш

інтуїтивного та персоналізованого досвіду.

У межах цього проекту ми зосередимося на розробці системи, яка б забезпечувала високу точність розпізнавання мови та ефективність у використанні для кінцевого користувача. Планується використати передові алгоритми та інноваційні технології для досягнення цієї мети, що дозволить створити практичний продукт.

Розроблюваний сервіс зосереджується на значному вдосконаленні технології розпізнавання мови, впроваджуючи новітні алгоритми нейронних мереж та машинного навчання. Це не просто покращує точність розпізнавання мови, але й забезпечує більш плавне та природне взаємодія з розумними колонками. Окрім цього, цей сервіс враховує регіональні діалекти та акценти, що робить його доступним для ширшої аудиторії та забезпечує краще розуміння команд користувачів з різних регіонів.

Наукова новизна проекту полягає у використанні інтегрованого підходу до обробки природної мови (NLP), який поєднує глибоке навчання та семантичний аналіз. Це дозволяє системі не тільки розпізнавати слова, але й розуміти контекст та нюанси мовлення. Такий підхід значно підвищує ефективність взаємодії з користувачами, адже система здатна адаптуватися до індивідуальних особливостей мовлення кожного користувача [4].

З практичної точки зору, розробка цієї системи розпізнавання мови має потенціал значно поліпшити взаємодію з "розумними" пристроями в домогосподарствах. Вона може автоматизувати багато повсякденних задач, таких як управління освітленням, температурою та іншими функціями домашнього обладнання, завдяки простому голосовому керуванню. Це не тільки зробить "розумний" дім більш інтуїтивним і зручним, але й зменшить потребу в фізичній взаємодії, що є особливо важливим для людей з обмеженими можливостями або для літніх осіб [5].

1 АНАЛІЗ СУЧАСНИХ «РОЗУМНИХ» СИСТЕМ

1.1 Аналіз пристроїв

У сучасному світі інформаційних технологій, особливу увагу приділяється розвитку інтелектуальних рішень для дому, таких як розумні колонки. Ці пристрої повинні бути не тільки функціональними, але й зручними у використанні, із задовільним діапазоном звуку, компактними розмірами та достатньою пам'яттю для музики. Важливо також забезпечити, щоб інтерфейс був легким для розуміння та використання всіма користувачами [6].

Щоб забезпечити конкурентоспроможність на ринку, проводиться дослідження існуючих продуктів, таких як портативні колонки JBL Boombox та Defender, щоб виявити їх сильні сторони та недоліки (рис. 1.1). Цей аналіз дозволяє визначити оптимальні параметри та характеристики для нової розумної колонки, забезпечуючи її високу якість та функціональність. Ключовими аспектами вважаються надійність, дизайн, інтуїтивність використання, а також технічні характеристики, такі як частотний діапазон, інтерфейси та тривалість роботи від батареї [7].



Рисунок 1.1 – Портативні колонки

В рамках цього дослідження проводиться детальний аналіз лідерів ринку портативних акустичних систем, щоб визначити, які саме функції та характеристики роблять їх вибором споживачів. Особливу увагу звертаємо на інновації, які використовуються в цих пристроях, такі як розширені

можливості підключення, інтеграція з різними мобільними додатками та можливість синхронізації з іншими розумними пристроями в домі [8].

Розглядаються й такі важливі аспекти, як естетика дизайну, яка має велике значення для споживачів. Аналізуються не лише технічні характеристики, але й візуальний вигляд, матеріали корпусу та загальна якість збірки. Це важливо для розуміння того, як споживачі сприймають продукт та які фактори впливають на їх рішення при покупці [9].

Також розглядається питання екологічності та стійкості, оскільки споживачі все більше усвідомлюють важливість екологічно чистих та довговічних продуктів. Вивчення цих аспектів допоможе створити не тільки функціональний, але й екологічно відповідальний продукт, який буде відповідати сучасним вимогам та тенденціям ринку.

1.2 Аналіз портативних акустичних систем

В рамках цього дослідження проводиться детальний аналіз лідерів ринку портативних акустичних систем, щоб визначити, які саме функції та характеристики роблять їх вибором споживачів. Особливу увагу звертаємо на інновації, які використовуються в цих пристроях, такі як розширені можливості підключення, інтеграція з різними мобільними додатками та можливість синхронізації з іншими розумними пристроями в домі.

Розглядаються й такі важливі аспекти, як естетика дизайну, яка має велике значення для споживачів. Аналізуються не лише технічні характеристики, але й візуальний вигляд, матеріали корпусу та загальна якість збірки. Це важливо для розуміння того, як споживачі сприймають продукт та які фактори впливають на їх рішення при покупці [10].

Також розглядається питання екологічності та стійкості, оскільки споживачі все більше усвідомлюють важливість екологічно чистих та

довговічних продуктів. Вивчення цих аспектів допоможе створити не тільки функціональний, але й екологічно відповідальний продукт, який буде відповідати сучасним вимогам та тенденціям ринку.

Порівняльний аналіз аудіоколонок JBL Boombox і Defender Enjoy вимагає зосередження на кількох ключових аспектах: виробник, якість звуку, потужність, автономність роботи, функціональність, портативність, дизайн, зручність використання, стійкість до зовнішніх впливів та вартість (табл. 1).

Таблиця 1.1 – Аналіз портативних колонок

Параметр	JBL Boombox [11]	Defender Enjoy [12]
Виробники	виробляються компанією JBL, відомим американським брендом аудіотехніки, який спеціалізується на виготовленні високоякісних гучномовців і аудіоаксесуарів.	виробляється компанією Defender, яка є менш відомою, але пропонує доступніші варіанти аудіообладнання
Якість звуку	відомі своєю високою якістю звуку, зокрема, чіткими високими та глибокими басами	може мати менш виразний звуковий профіль порівняно з JBL, але все ж пропонує прийнятну якість для своєї цінової категорії.
Потужність та гучність	має більш високу гучність і потужність, що робить її ідеальною для використання на відкритому повітрі або великих приміщеннях.	має меншу потужність порівняно

Продовження таблиці 1.1

Параметр	JBL Boombox	Defender Enjoy
Автономність роботи	пропонують тривалу роботу на одному заряді, забезпечуючи довгий час використання без підзарядки.	може мати менший час автономної роботи через меншу ємність акумулятора.
Функціональність	часто оснащуються додатковими функціями, такими як водонепроникність, підключення до декількох джерел одночасно, та вбудованими мікрофонами для голосових дзвінків.	може не мати деяких з цих функцій, але зазвичай пропонує базові можливості підключення.
Портативність	є великою та важчою, але досі портативною	більш компактні та легкі
Вартість	Вище середнього	Вище середнього

1.3 Інновації і тренди в розробці розумних колонок

Останнім етапом дослідження є вивчення найсвіжіших інновацій та трендів у галузі розумних колонок. Важливо розуміти, які технологічні нововведення та концепції набирають популярності, щоб забезпечити, що наша розробка буде актуальною та затребуваною на ринку. Таким чином, ми зосереджуємо увагу на таких аспектах, як штучний інтелект, голосове

управління, інтеграція з домашніми автоматизаційними системами, а також на можливостях мультимедійного аудіо [13].

Споживачі сьогодні очікують від розумних колонок не лише високоякісного звуку, але й розумних функцій, таких як голосове керування, інтеграція з популярними голосовими асистентами та можливість управління іншими розумними пристроями в домі. Також спостерігається тенденція до збільшення попиту на пристрої, які можуть працювати у різних кімнатах без втрати якості звуку чи зручності управління [14].

Розумні колонки – це передові технологічні пристрої, які поєднують апаратне та програмне забезпечення для надання різноманітних інтерактивних і підключених послуг. Їх основна суть і використовувані технології описані в таблиці 1.2.

Таблиця 1.2 – Технології розумних колонок

Технологія	Опис
Розпізнавання та обробка голосу	розумні колонки використовують складну технологію розпізнавання голосу для виявлення та інтерпретації голосових команд. Це передбачає обробку природної мови (NLP), яка дозволяє пристрою розуміти людську мову та реагувати на неї природним та інтуїтивно зрозумілим способом.
Штучний інтелект (AI) і машинне навчання (ML)	ці технології є основою розумних колонок. AI дозволяє оратору вчитися на взаємодії та покращувати свої відповіді з часом. Алгоритми машинного навчання допомагають зрозуміти вподобання користувачів, підвищити персоналізацію та надавати точніші відповіді на запити [15].

Продовження таблиця 1.2

Технологія	Опис
Підключення до Інтернету	розумним колонкам потрібне підключення до Інтернету для доступу до різних онлайн-служб і баз даних. Це підключення дозволяє їм транслювати музику, отримувати доступ до новин, прогнозу погоди та інтегруватися з іншими веб-сервісами.
Інтеграція пристроями розумного дому	з багато розумних динаміків можуть підключатися та спілкуватися з іншими розумними пристроями в домі, такими як освітлення, термостати, камери безпеки та побутові прилади. Ця інтеграція зазвичай досягається за допомогою протоколів бездротового зв'язку, таких як Wi-Fi, Bluetooth, Zigbee або Z-Wave [16]
Хмарні обчислення	розумні колонки значною мірою покладаються на хмарні обчислення для зберігання та обробки даних.
Аудіотехнологія	високоякісні мікрофони та колонки мають вирішальне значення. Мікрофони призначені для сприйняття голосових команд на відстані, часто з можливістю фільтрації фонового шуму. Тоді динамік пристрою забезпечує чистий і високоякісний звук [17].
Протоколи безпеки та конфіденційності	враховуючи те, що розумні колонки завжди слухають, вони містять заходи безпеки для захисту даних користувачів. Це включає шифрування даних і можливість для користувачів переглядати та видаляти свої голосові записи.
Оновлення програмного забезпечення та мікропрограми	щоб підтримувати функціональність, покращувати функції та усунути вразливості безпеки, розумні колонки регулярно отримують оновлення програмного забезпечення та мікропрограми.

Розумні колонки представляють собою конвергенцію різноманітних передових технологій, включаючи штучний інтелект, розпізнавання голосу, хмарні обчислення та Інтернет речей, що забезпечує користувачам безперебійний інтерактивний досвід керування іншими пристроями, доступу до інформації та використання різноманітних цифрових сервісів [18].

Враховуючи ці тенденції, акцентуємо увагу на розробці продукту, який не тільки відповідає всім технічним вимогам, але й відрізняється інноваційними рішеннями. Це включає використання передових алгоритмів для покращення якості звуку, інтеграцію з широким спектром інших розумних пристроїв та можливість гнучкої кастомізації налаштувань залежно від індивідуальних вподобань користувача.

Таким чином, ми отримуємо всебічний аналіз вимог та тенденцій у галузі розумних колонок, який покладе основу для нашої подальшої розробки, спрямованої на створення інноваційного та затребуваного продукту.

1.4 Інтеграція з пристроями розумного дому

Інтеграція голосового помічника з пристроями розумного дому полягає у створенні безшовної і взаємопов'язаної системи, яка дозволяє користувачам управляти різними побутовими пристроями за допомогою голосових команд. Для цього, насамперед, голосовий помічник підключається до домашньої мережі, що дозволяє йому спілкуватися з іншими підключеними до мережі пристроями, такими як розумні лампочки, термостати, замки та медіацентри. Помічник налаштовується через спеціальний додаток, де користувачі можуть додавати та керувати пристроями, вказувати конкретні кімнати та налаштовувати автоматичні сценарії для домашньої автоматизації [19].

Завдяки алгоритмам машинного навчання та обробки природної мови,

голосовий помічник ефективно розпізнає та інтерпретує голосові команди користувачів, перетворюючи їх у конкретні дії для різних пристроїв. Безпека комунікації між помічником і пристроями, а також приватність користувачів, є важливими аспектами, особливо в контексті домашньої автоматизації [20].

Голосовий помічник може інтегруватися з зовнішніми сервісами та додатками, збільшуючи свої функціональні можливості. Наприклад, використання сервісів типу IFTTT дозволяє зв'язувати різні сервіси та пристрої для створення ще більш комплексних сценаріїв автоматизації [21]. Все це створює універсальне та гнучке середовище для розумного дому, в якому голосовий помічник відіграє ключову роль у спрощенні побутових процесів та підвищенні комфорту життя користувачів.

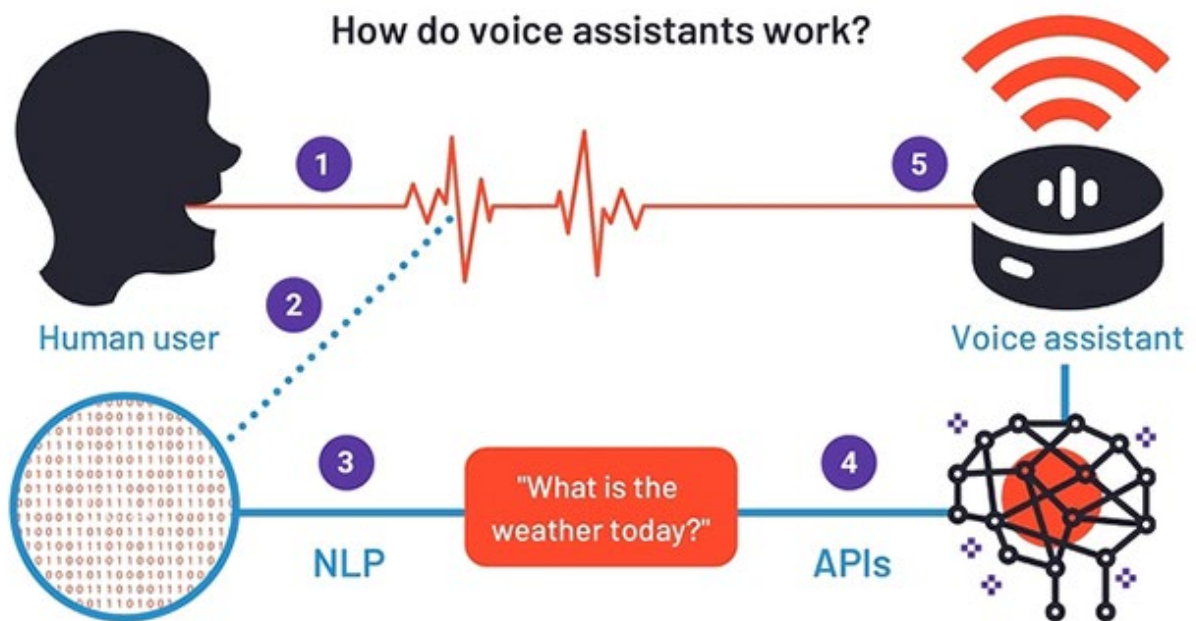


Рисунок 1.2 – Інтеграція із іншими пристроями

Інтеграція голосового помічника з розумними пристроями дому вимагає застосування різних мереж і протоколів для забезпечення ефективного спілкування та управління. Для підключення до домашньої мережі, голосовий помічник зазвичай використовує Wi-Fi або Bluetooth. Це дозволяє помічнику взаємодіяти з розумними пристроями, такими як

освітлювальні прилади, термостати, розумні замки, медіаплеєри та інші прилади, які підтримують ці мережеві технології.

Для управління розумними пристроями використовуються різні протоколи. Наприклад, Zigbee та Z-Wave – це два популярних бездротових протоколи, які забезпечують надійне спілкування між пристроями в межах дому. Вони відрізняються низьким споживанням енергії та забезпечують великий радіус дії, що є важливим для розширених домашніх систем. Іншим важливим аспектом є використання стандартів, таких як MQTT (Message Queuing Telemetry Transport), який є легким та ефективним протоколом обміну повідомленнями, ідеально підходить для Інтернету речей.

Безпека та приватність у таких системах також є критично важливими. Шифрування даних, безпечні методи аутентифікації та регулярні оновлення безпеки є ключовими для захисту від несанкціонованого доступу та забезпечення конфіденційності користувачів.

Інтеграція голосового помічника з зовнішніми сервісами, такими як IFTTT (If This Then That), подальше розширює можливості автоматизації, дозволяючи користувачам створювати складні сценарії, які зв'язують різні пристрої та сервіси. Це створює гнучку та універсальну систему, яка спрощує повсякденне життя і підвищує комфорт користувачів у їхніх розумних домах.

1.5 Автоматичне розпізнавання мови

У контексті розвитку технологій, існує можливість, що ми ще на кілька десятиліть відділені від створення систем, які можуть самостійно функціонувати та взаємодіяти з нами як справжні люди. Проте, наш постійний прогрес у цій галузі частково зумовлений розвитком технологій автоматизованого розпізнавання мови. Автоматичне розпізнавання мови (АРМ) є технологією, яка дозволяє особам використовувати голос для

взаємодії з комп'ютерними системами, де вона в своїх найскладніших формах імітує реальну людську бесіду. Сучасні приклади розроблених АРМ-технологій включають Siri від Apple, Alexa від Amazon, Cortana від Microsoft та інші системи, застосовані у бізнесі та високотехнологічних сферах. Проте, навіть ці програми, попри їх "точність" між 96 та 99%, досягають таких показників лише в оптимальних умовах та з використанням певних ключових слів [22]. Основний процес роботи будь-якого АРМ-програмного забезпечення, незалежно від його складності, полягає в наступному:

- ваша мова передається через аудіо-канал;
- використовуваний вами пристрій перетворює звуки в аудіо-хвилі;
- ці хвилі очищаються шляхом усунення шуму та регулювання гучності;
- очищена хвильова форма розділяється на фонемі;
- кожна фонема аналізується послідовно, починаючи з першої, де використовується статистичний аналіз для визначення слів і, відповідно, цілих речень;
- система розпізнавання мови, розшифрувавши ваші слова, виконує потрібні дії.

Трансформація голосу в цифровий текстовий формат є напрямком, який привертає увагу численних програмістів вже протягом декількох років. Ця область об'єднує в собі різні наукові дисципліни, включаючи лінгвістику, математику та комп'ютерні науки. Процес вимагає складного аналізу голосових характеристик для точного розпізнавання [23].

Розпізнавання мовлення часто використовується у сферах, де традиційні методи введення даних можуть бути неефективними. Це включає військові застосування, особливо у авіації, а також транспорт, освіту та автономні системи, де програмне забезпечення для розпізнавання голосу стає все більш популярним [24].

Розробка додатків, що перетворюють голос у текст, можлива завдяки різним API, які пропонують провідні технологічні компанії, такі як Google

[25], Microsoft [26], Amazon [27] та Apple [28]. Вони використовують свої унікальні алгоритми для обробки мовлення та вкладають значні ресурси в їх удосконалення, щоб точно перевести голос в текст.

Алгоритми розпізнавання мовлення є складними, і однією з причин їх складності є використання штучного інтелекту. Три основні методи, які зазвичай використовуються для цього, включають приховану марковську модель (ПММ), алгоритм динамічної трансформації тимчасової шкали (АДТТШ) та нейронні мережі [29]. Прихована марковська модель - це тип графічної моделі, яка широко використовується для моделювання часових даних. Ці моделі припускають, що спостережувані дані генеруються основними прихованими станами, що робить їх ефективними для застосувань, таких як розпізнавання мовлення, розпізнавання рукописного тексту та синтез мовлення.

Алгоритм динамічної трансформації тимчасової шкали є методом вимірювання схожості між двома рядами даних, які можуть варіюватися в часі або швидкості. Це можна уявити як порівняння двох відеозаписів: навіть якщо людина на одному відео йде повільно, а на іншому - швидко, або якщо є зміни швидкості, алгоритм зможе виявити схожість у моделях руху. АДТТШ дає змогу комп'ютеру визначити найкращий відповідник між двома заданими послідовностями, враховуючи певні обмеження. На відміну від ПММ, який аналізує голосовий сигнал та ймовірності слів у реченні, АДТТШ фокусується на розпізнаванні окремих слів.

Нейронні мережі, хоч і здатні виконувати складні завдання розпізнавання, можуть бути менш ефективними у порівнянні з ПММ при роботі з великими словниками. Вони використовуються через їх здатність зменшувати обсяг моделювання та підвищувати точність розпізнавання. Глибинне навчання у цьому контексті дозволяє розробляти гібридні системи, що поєднують переваги обох технологій [30].

На даний момент чимало великих компаній пропонують API для вирішення різноманітних задач у сфері машинного навчання, і розпізнавання

мови - не виняток. Для використання цих API не обов'язково бути експертом у мовній обробці, адже вони зазвичай мають дружній до користувача інтерфейс. Все, що вам потрібно - це відправити HTTP-запит з потрібними даними на сервер API, і ви отримаєте відповідь із завершеними задачами. Такий підхід є ефективним, коли ви розв'язуєте загальновідомі проблеми, які не вимагають унікального підходу. Крім того, він дозволяє заощадити час та кошти. Однак існують ситуації, коли використання готових API неможливе, і потрібно створити систему розпізнавання мови з нуля. Це може бути складним і вимагати значних ресурсів, але дозволить створити рішення, яке точно відповідатиме вашим вимогам, а також може покращити якість результатів за рахунок індивідуального підходу в розробці алгоритмів. Знання про різноманітні API допомагає визначити, коли їх варто використовувати, а коли краще розробити власне рішення. У мовній обробці виділяють два ключові завдання: перетворення мовлення в текст та перетворення тексту в мовлення.

1.6 Висновки за розділом

Аналіз ринку показав, що сучасні розумні колонки стають все більш інтегрованими в повсякденне життя людей. Це вимагає постійного удосконалення технологічних характеристик, включаючи якість звуку, дизайн, інтерфейс користувача та інтеграцію з іншими розумними пристроями. Інновації в цій галузі повинні орієнтуватися на підвищення зручності використання та розширення функціональності.

Сучасні споживачі вимагають, щоб розумні колонки були оснащені голосовим управлінням та інтеграцією з широким спектром голосових асистентів. Це підкреслює необхідність розробки інтелектуальних інтерфейсів, що можуть вплинути на користувацький досвід.

На основі проведеного аналізу та оцінки поточного стану ринку, стає зрозумілою важливість створення комп'ютерної системи, яка забезпечує високу точність розпізнавання мови та є зручною у використанні для кінцевого користувача. Така система має включати сучасні алгоритми обробки звуку та інтелектуальні можливості для підвищення ефективності та зручності використання.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Розробка структури

Система має забезпечити користувачам легкий доступ до контенту через їхні пристрої, дозволяти їм ефективно виконувати задачі та забезпечувати безперешкодну діяльність. Користувачі повинні мати можливість керувати гучністю звуку, а також при необхідності можуть призупинити роботу пристрою у разі виникнення технічних проблем. Загалом, взаємодія з системою повинна бути інтуїтивно зрозумілою і не вимагати від користувачів значних зусиль. Це демонструється на прикладі рисунку 2.1.

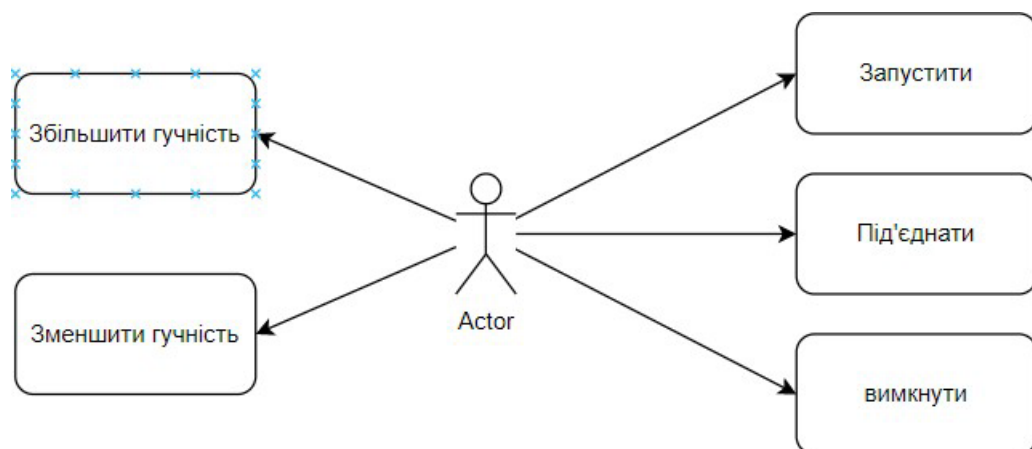


Рисунок 2.1 – Взаємодія із системою

Розробка блок-схеми алгоритму роботи системи є ключовим етапом у процесі проектування та розробки будь-якої програмної системи, оскільки вона відіграє важливу роль у визначенні та візуалізації логіки системи. Блок-схема допомагає розробникам та іншим зацікавленим сторонам зрозуміти, яким чином дані протікають через систему та які процеси виконуються на різних етапах.

Алгоритм дозволяє візуально представити комплексні процеси у спрощеній та зрозумілій формі. Це сприяє кращому розумінню функціональності системи та логіки її роботи, що є особливо корисним на початкових етапах розробки, коли основні концепції та архітектура системи все ще формуються. Вона допомагає уникнути непорозумінь та забезпечує однакове розуміння процесів та взаємодій всіма сторонами.

Блок-схема сприяє ідентифікації потенційних проблем та слабких місць у логіці алгоритму до початку програмування. Це дозволяє внести необхідні корективи на ранніх етапах, що може значно заощадити час та ресурси, які могли б бути витрачені на виправлення помилок у майбутньому.

Розробка алгоритму є важливим етапом у документуванні системи. Вона служить як довідковий матеріал для поточних та майбутніх розробників, допомагаючи зрозуміти та підтримувати систему протягом всього життєвого циклу.

Розробка блок-схеми алгоритму роботи системи - це не тільки про візуалізацію процесів, але й про ефективне планування, комунікацію, виявлення помилок та документування, що є невід'ємною частиною успішного розроблення програмного забезпечення.

Основна логіка роботи системи включає в себе набір функцій, які дозволяють визначити працездатність пристрою та його здатність виконувати базові операції (рис. 2.2).

Блок-схема описує послідовність дій алгоритму:

- перший крок — запуск системи;
- перевірка заряду - чи достатній рівень заряду батареї для подальшої

роботи;

- перевірка підключення зарядного пристрою. Якщо заряд недостатній, система перевіряє, чи підключено зарядний пристрій;
- перевірка підключення вихідного пристрою. Якщо заряд достатній, система перевіряє, чи підключено вихідний пристрій;
- ситуація "батарея". Якщо зарядний пристрій підключено, система переходить до режиму "батарея";
- робота системи. Після перевірки підключення і стану батареї, система переходить до основної робочої частини керування голосом.

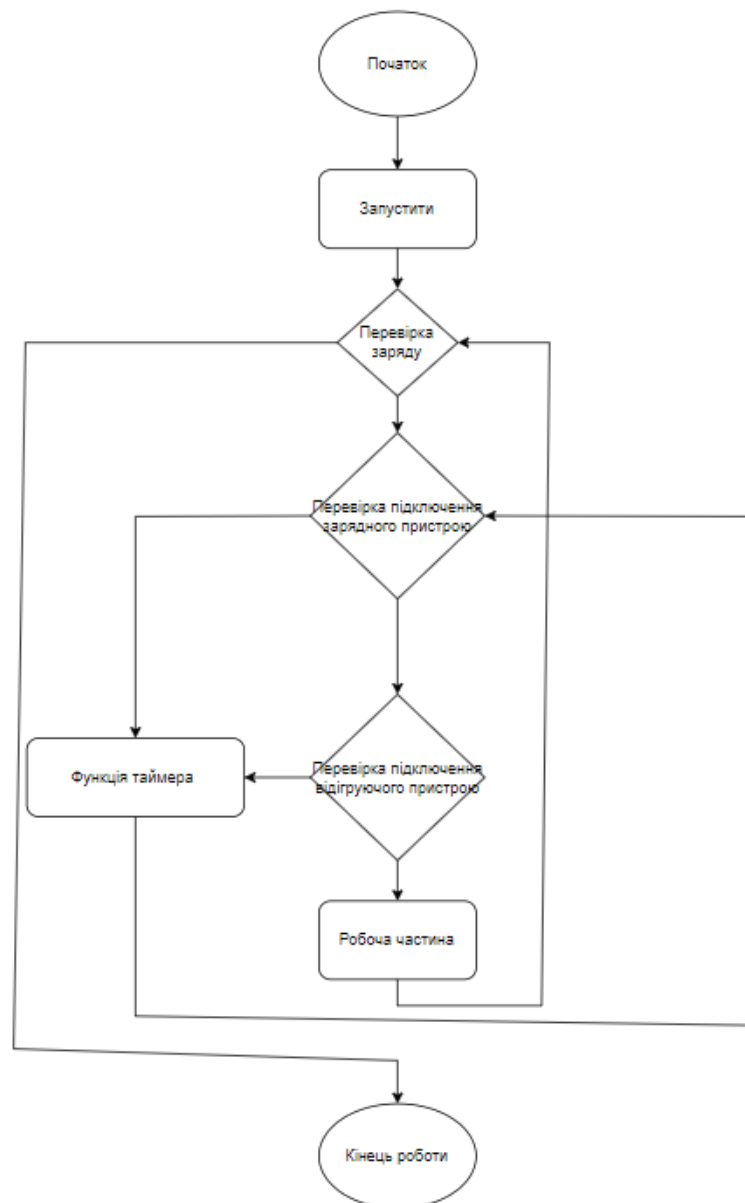


Рисунок 2.2 – Блок-схема роботи розумною колонки

Ергономічний дизайн інтерфейсів фокусується на створенні таких користувацьких інтерфейсів, які оптимізують взаємодію між користувачем і технологією, враховуючи зручність, ефективність та доступність. Головна мета полягає у зниженні когнітивного навантаження на користувача, щоб зробити використання технології інтуїтивно зрозумілим та легким.

Для досягнення цих цілей, ергономічний дизайн орієнтується на декілька ключових аспектів (табл. 2.1).

Таблиця 2.1 – Принципи ергономічного дизайну

Принцип	Опис
Спрощення інтерфейсу	Уникати перевантаження інтерфейсу надмірною кількістю елементів або складними меню.
Консистентність	Інтерфейси повинні бути уніфіковані у своєму дизайні з однаковим використанням кольорів, шрифтів, кнопок тощо.
Врахування потреб користувача	Розуміти задачі, які користувач намагається вирішити, і адаптувати функціонал інтерфейсу відповідно.

Технологій голосового управління - це технології дозволяють користувачам керувати різними пристроями і функціями за допомогою голосових команд, що значно зменшує необхідність фізичної взаємодії та підвищує загальний комфорт використання.

Методи тестування користувацького досвіду (User Experience Testing, UX Testing). – це важливий етап у розробці інтерфейсів, який допомагає гарантувати, що вони є не тільки функціональними, але й інтуїтивно зрозумілими та зручними для користувачів.

Методи тестування користувацького досвіду охоплюють:

- тестування з користувачами. Найбільш ефективний спосіб виявити проблеми у взаємодії з інтерфейсом — це спостерігати за реальними

користувачами під час їх використання продукту. Це допомагає ідентифікувати місця, де користувачі стикаються з труднощами або плутаниною;

- аналіз шляхів користувача (User Journey Mapping). Цей метод дозволяє відстежити, як користувачі переміщуються через інтерфейс, від однієї функції до іншої, виявляючи потенційні проблеми у навігації та доступності;

- А/В тестування. Ця методика передбачає створення двох версій інтерфейсу, де невеликі зміни вносяться в одну з версій, а потім аналізується, яка версія краще виконує певні цілі або завдання;

- використання аналітичних інструментів. Сучасні інструменти аналітики дозволяють збирати дані про використання продукту, включаючи частоту використання певних функцій, час взаємодії з продуктом та інші важливі показники.

Ці методи допомагають не тільки виявити і виправити проблеми, але й забезпечують цінний зворотний зв'язок для постійного вдосконалення продукту.

Основною метою системи для проектування комп'ютерного обладнання було забезпечення взаємодії з сервісом Google Speech API, котрий володіє функціональністю конвертації аудіофайлів у текстовий формат. Для комунікації з сервером сервісу використовуватиметься окремий сервер, який буде займатися обробкою запитів, що надходять і відправляються системою (рис. 2.3).

Треба розглянути наступні аспекти:

- інтеграція з Google Speech API. Подробиці процесу інтеграції системи із зазначеним API, включаючи авторизацію, формати даних, які використовуються, та оптимізацію процесу перетворення аудіо в текст;

- створення та управління сервером. Обговорення технічних деталей створення сервера, що взаємодіє з Google Speech API, включаючи вибір мови програмування, серверного програмного забезпечення, та механізмів обробки

запитів;

- безпека та конфіденційність: Розгляд питань безпеки даних та конфіденційності інформації, яка передається між системою та Google Speech API, з урахуванням стандартів шифрування та передачі даних.

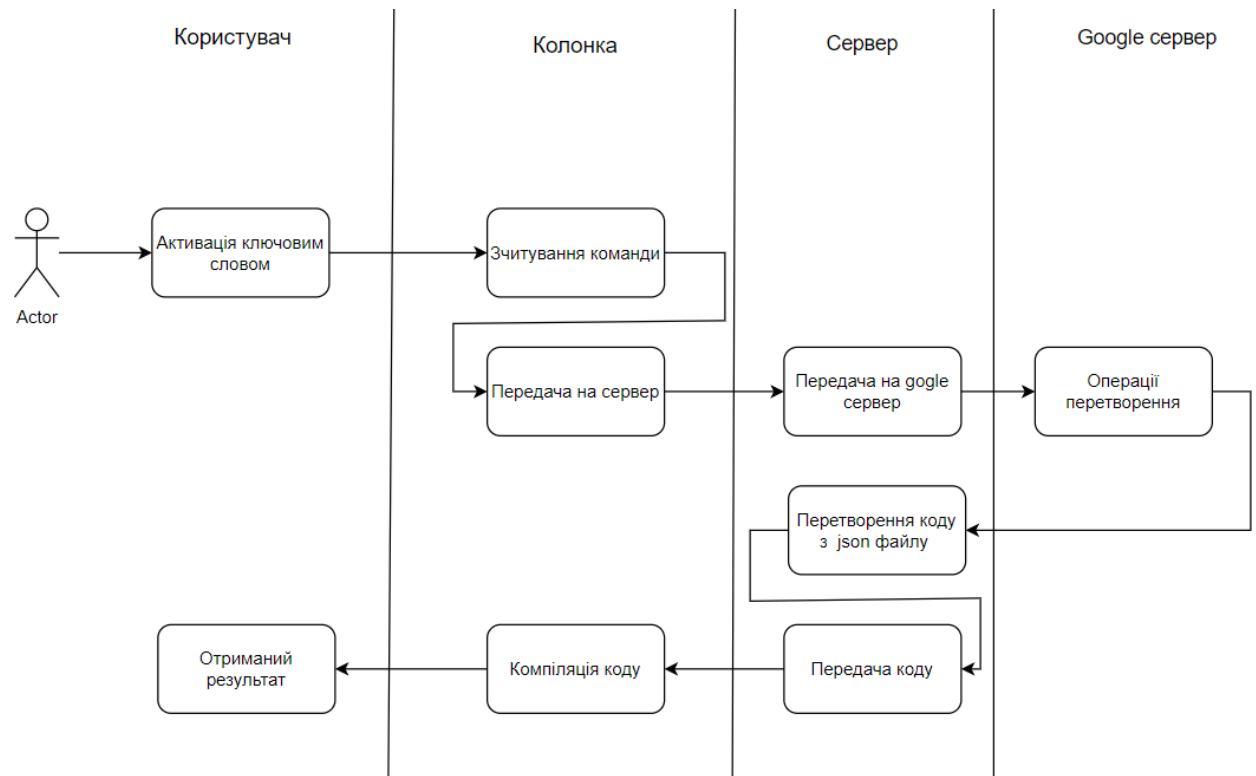


Рисунок 2.3 – Діаграма взаємодії компонентів системи

2.2 Обґрунтування апаратної частини системи

Аудіотехніка Fusion, створена і протестована компанією Audix в США, відповідає всім вимогам для використання у професійних студіях та на концертних майданчиках. Комплекти мікрофонів пропонують користувачам високий рівень якості та зручності, разом з надійністю та доступною ціною. Кожен мікрофон комплектується з кліпсами для кріплення на стійки та упаковується в міцний алюмінієвий кейс, що забезпечує їх безпеку під час транспортування та зберігання [31].

Основна перевага бездротових мікрофонів полягає в тому, що вони надають виконавцям більше свободи руху порівняно з провідними аналогами, хоча і мають обмеження у потужності сигналу та потребують регулярної заміни батарей. Бездротові мікрофони випускаються як для домашнього, так і для професійного використання. Домашні моделі зазвичай прості у використанні і мають мінімальні налаштування, тоді як професійні системи дозволяють детальне кастомізування звукових налаштувань. Це може включати калібрування та налаштування сигналу для кожного окремого мікрофона, дозволяючи одному приймачу одночасно обробляти сигнали з десятка і більше мікрофонів. Професійні бездротові мікрофони відомі своєю високою якістю передачі звуку, що робить їх популярним вибором для концертів. Також існують цифрові мікрофонні радіосистеми з професійних лінійок, які пропонують ще більшу якість та зручність у використанні.



Рисунок 2.4 – Вибір мікрофонів

Беззаперечним лідером серед компонентів сучасних звукових систем є динамічний драйвер. Гучномовці, оснащені динамічними драйверами, легко впізнаються по їх характерним конусам і куполам. Їх широка популярність обумовлена рядом переваг: здатністю відтворювати широкий спектр звуків, високою чутливістю, здатністю працювати з потужними сигналами, простотою конструкції та міцністю. Також динамічні драйвери називають пунктирними джерелами звуку, адже звук виходить із певної точки.

У складі акустичних систем часто можна зустріти кілька динамічних драйверів різного розміру. Конічний вуфер відтворює низькі частоти, тоді як високі частоти генеруються твітером, який зазвичай містить маленький металевий або тканинний купол. Іноді гучномовці комплектуються третім драйвером для середніх частот.

Принцип роботи всіх цих динаміків залишається однаковим. Електричний струм від підсилювача проходить через звукову котушку, що створює магнітне поле, яке розширюється і стискається відповідно до частоти звукового сигналу. Звукова котушка, яка висить у постійному магнітному полі, створеному магнітами в драйвері, взаємодіє з цим полем, що призводить до її руху вперед і назад. Цей рух передається конусу драйвера, викликаючи вироблення звуку [32].

Інші компоненти динамічного гучномовця включають центральну шайбу або так званий "павук", який підтримує голосову котушку на протязі її руху вперед і назад. Структура, відома як кошик, являє собою відлиту або виготовлену методом штампування металеву основу, до якої кріпляться усі частини динаміка. (Відлиті кошики зазвичай застосовуються у більш висококласних гучномовцях, в той час як штамповані використовуються в економічних моделях). Еластичне кільце підвісу з'єднує конус з рамкою кошика, дозволяючи дифузору переміщуватись вперед і назад, залишаючись при цьому закріпленим до кошика, з можливістю максимального руху, відомого як екскурсія.

STM32F103C8T6 є одним з мікроконтролерів сімейства STM32 від

компанії STMicroelectronics, оснований на ядрі ARM Cortex-M3 [33].

Atmel SAM3X8E, NXP LPC1768 та Microchip PIC32MX340F512H є мікроконтролерами, кожен з яких має свої унікальні характеристики та застосування. Atmel SAM3X8E, який часто зустрічається в Arduino Due, базується на 32-бітному ARM Cortex-M3 ядрі, пропонуючи багатий набір периферії та високу продуктивність для застосувань, що потребують інтенсивної обробки даних. Це робить його ідеальним для складних проектів, що вимагають додаткової пам'яті та обчислювальних можливостей.



Рисунок 2.5 - Atmel SAM3X8E

NXP LPC1768 є також 32-бітним ARM Cortex-M3 мікроконтролером, який популярний серед розробників за його високу продуктивність та гнучкість. Він має великий набір вбудованих периферійних інтерфейсів, включаючи Ethernet, USB OTG, CAN та багато інших, що робить його підходящим для широкого спектру промислових та споживчих застосувань.

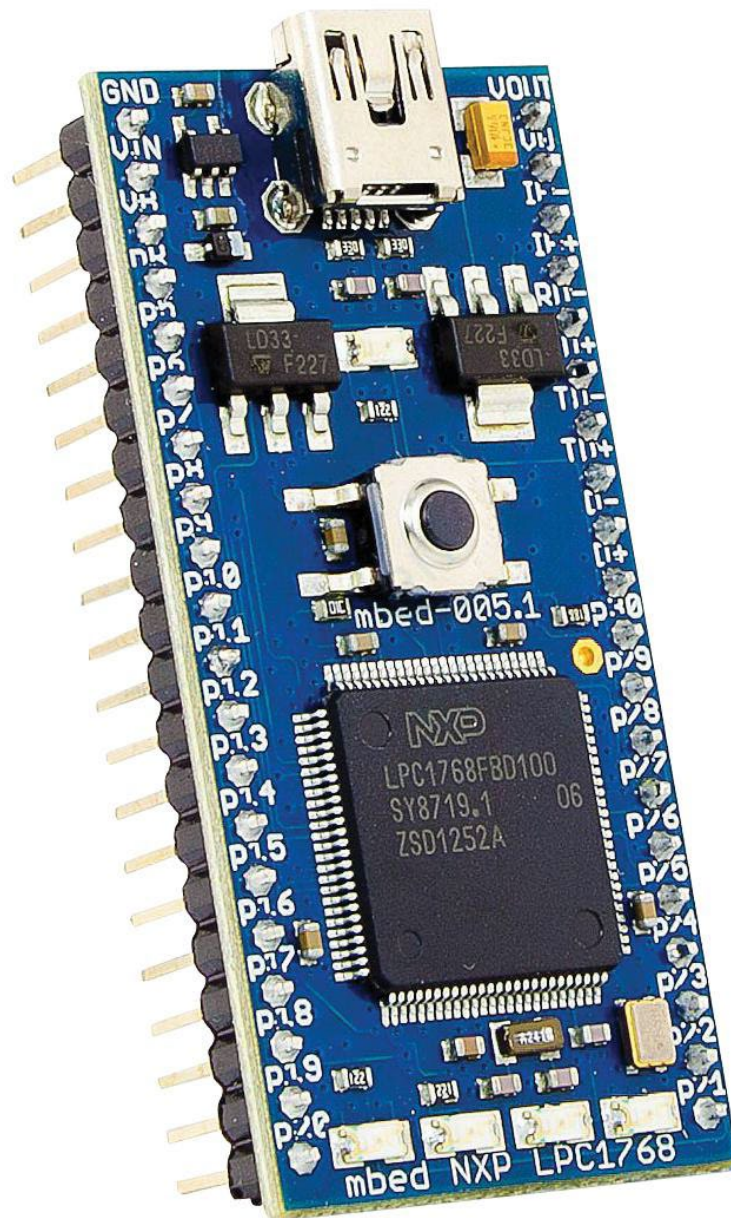


Рисунок 2.6 - NXP LPC1768

Microchip PIC32MX340F512H відноситься до сімейства PIC32 мікроконтролерів від Microchip. Він базується на MIPS ядрі та надає хорошу продуктивність з 512 КБ вбудованої флеш-пам'яті та 32 КБ RAM. Він також має багатий набір периферійних інтерфейсів, що робить його відмінним вибором для розширених аудіо/відео застосувань та інших високопродуктивних додатків.



Рисунок 2.7 - Microchip PIC32MX340F512H

Кожен з цих мікроконтролерів має свої переваги та специфікації, що робить їх підходящими для різних типів проектів у галузі електроніки та вбудованих систем. Вибір між ними залежить від конкретних вимог проекту, таких як необхідна обчислювальна потужність, доступні периферійні інтерфейси та обмеження розміру пам'яті.

Для порівняння з аналогами з інших сімейств або від інших виробників, важливо розглянути декілька ключових характеристик, таких як процесорне ядро, кількість пам'яті, кількість і типи введення/виведення (I/O), а також додаткові можливості, такі як вбудовані периферійні інтерфейси і підтримка апаратного прискорення (табл. 2.2).

Порівняння цих мікроконтролерів показує, що STM32F103C8T6 має досить конкурентоспроможні характеристики в своєму класі, особливо зважаючи на його доступність та популярність. Однак, інші мікроконтролери можуть пропонувати більше пам'яті, вищу тактову частоту або додаткові периферійні інтерфейси, що може бути критично для певних застосувань.

Таблиця 2.2 – Порівняння контролерів

Мікроконтролер	Процесор	Максимальна частота (МГц)	Пам'ять Flash (КБ)	SRAM (КБ)	Основні периферійні інтерфейси
STM32F103C8T6	ARM Cortex-M3	72	64	20	USB 2.0, I2C, SPI, USART, CAN, ADC, таймери
Atmel SAM3X8E [34]	ARM Cortex-M3	84	512	96	USB 2.0, I2C, SPI, USART, ADC, DAC, таймери
NXP LPC1768 [35]	ARM Cortex-M3	100	512	64	Ethernet, USB 2.0, I2C, SPI, USART, CAN, ADC, DAC, таймери
Microchip PIC32MX340F512H [36]	MIPS M4K	80	512	32	USB 2.0, I2C, SPI, USART, ADC, таймери

Контролер STM32F103C8T6 ARM STM32 - це плата для налаштування, яка також може бути поставлена у варіації STM32F103C6T6. Це мінімальна плата контролера від провідного у світі виробника - компанії STM32. Для програмування цієї плати необхідний окремий пристрій - програматор ST-Link, або можливе використання USB-UART конвертера. Плата STM32F103C8T6 з ядром ARM Cortex-M3 відрізняється відмінним балансом ціни та функціональності та пропонує більш високу продуктивність порівняно з більшістю економічних плат Arduino (рис. 2.5).

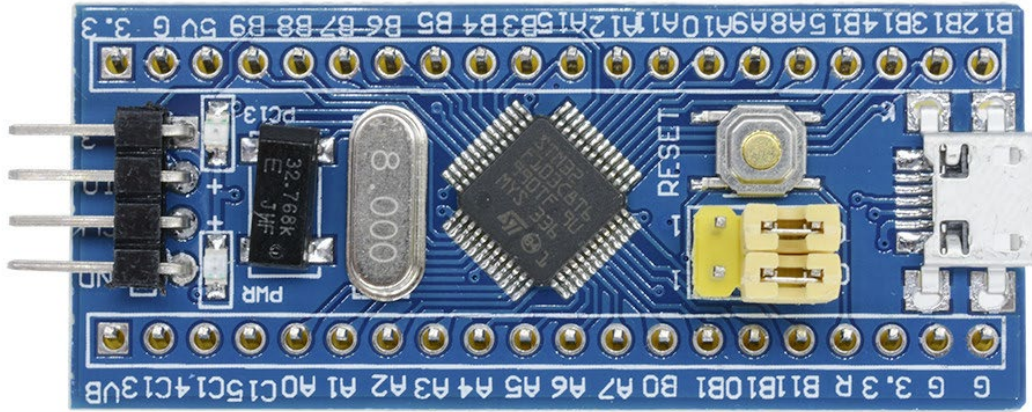


Рисунок 2.8 – Контролер системи [37]

На платі розміщені всі необхідні компоненти для старту роботи з мікроконтролером:

- 32 порти GPIO, які розведені на контактні площадки, сумісні за форм-фактором з роз'ємом DIP-40;
- два кварцові резонатори - один на 8МГц для тактування ядра та інший на 32768 Гц для тактування RTC;
- стабілізатор напруги на 3.3В, що забезпечує можливість живлення плати від 5В;
- роз'єм для підключення SWD програматора;
- MicroUSB роз'єм (підключений до апаратного USB інтерфейсу);
- кнопка для перезапуску системи;
- два світлодіоди (один для індикації живлення, другий підключений до порту PC13);
- два перемикачі для вибору зони пам'яті для початкового завантаження мікроконтролера.

Прошивка можлива через SWD програматор, як-от ST-Link, або за допомогою USB-UART конвертера. Завантажувач, встановлений в мікрочіп, надає можливість прошивки через вбудований MicroUSB порт. Для кодування та програмування STM32F103C8T6 можуть використовуватися як

спеціалізовані IDE, такі як Keil, IAR, Eclipse, так і Arduino IDE.

2.3 Вибір програмного забезпечення системи

За останні десять років галузі, такі як машинне навчання та обробка природної мови, зазнали значного прогресу, що особливо важливо у контексті розвитку голосового управління пристроями. Застосування статистичних методів у цих областях дозволяє алгоритмам ефективно вчиняти на основі величезної кількості даних, що сприяє швидкому зростанню технологій голосового керування. Ці технології вже знайшли застосування у багатьох сферах, включаючи медицину та освіту, та є ключовими для розвитку голосових асистентів у розумних домах.

В галузі ІТ також спостерігається стрімкий розвиток, зі створенням нових мов програмування та інструментів. Однак існують проблеми, які потребують вирішення, особливо у контексті доступності та ефективності. Наприклад, відсутність можливості для людей з обмеженими можливостями написати код є значною перешкодою. Крім того, традиційні засоби вводу, як-от клавіатура, кнопки і мишка, не завжди є ефективними для швидкого введення даних.

Сервіс для розпізнавання мовлення, який використовується в розробці голосового асистента, перетворює голосові команди користувачів на текст. У структурі цього голосового асистента, обробка мовлення здійснюється на хмарному сервері, а зв'язок із цим сервісом встановлюється через API-шлюз. Реалізація системи розпізнавання мовлення є складною задачею через велику кількість змінних, які впливають на точність розпізнавання. Серед основних факторів, що впливають на цей процес, є індивідуальні особливості мовлення користувача, включаючи акценти та вимову, а також зовнішні умови, такі як фоновий шум та особливості голосу користувача.

Мова програмування Python пропонує широкий спектр модулів та бібліотек для розпізнавання мовлення та його перетворення в текст, включаючи такі рішення, як ApiAI, SpeechRecognition, Google Speech Cloud, AssemblyAI, PocketSphinx, Watson Developer Cloud, WIT та інші. Використаний для сервісу розпізнавання мовлення модуль SpeechRecognition на Python дозволяє користуватися різноманітними можливостями різних провайдерів для конвертації мовлення в текст. Цей сервіс використовує модель «Speech to Text» від IBM, яка заснована на методах глибокого машинного навчання, включаючи рекурентні нейронні мережі з LSTM блоками та WaveNet моделями. Рішення дозволяє розпізнавати голосові команди на багатьох мовах та у різних аудіо форматах [38].

IBM Watson Speech to Text є сервісом від IBM Watson, який конвертує мову в текст. Цей сервіс дозволяє налаштовувати як словник, так і акустичні характеристики, що робить його адаптивним до різних умов використання. Однак основною недоліком IBM Watson є обмежена кількість підтримуваних мов, особливо спеціалізованих моделей для них. Інші функції сервісу включають альтернативи слів (бета-версія), часові мітки, фільтрацію ненормативної лексики та інтелектуальне форматування номерів телефонів, дат, валют тощо [35].

Microsoft Azure Bing Speech API є частиною хмарних сервісів Microsoft Azure, який здатен перетворювати мову в текст та текст у мовлення. Azure Bing перетворює мовлення на текст у реальному часі, з можливістю налаштування і форматування тексту, фільтрації ненормативної лексики та нормалізації. Інтеграція з Azure LUIS дозволяє витягувати значення з тексту. Сервіс безкоштовний до 5000 транзакцій на місяць, а після цього стягується плата 4 USD за кожні 1000 транзакцій [21].

Amazon Transcribe є частиною Amazon Web Services і дозволяє аналізувати аудіодокументи, збережені в Amazon S3, отримуючи текст із аудіо. Сервіс може додавати розділові знаки та формувати текст,

підтримує телефонну якість аудіо і додає часові позначки для кожного слова. Очікується, що незабаром він зможе розпізнавати та позначати різних спікерів у тексті. Також доступне створення власних слів. Послуга безкоштовна протягом перших 12 місяців після реєстрації (до 60 хвилин аудіо на місяць), а потім коштує 0,0004 USD за секунду обробленого аудіо [22].

VoxSigma API, розроблений Vocarìa Research, спеціалізується на мовних технологіях. Сервіс може перетворювати мову в текст, ідентифікувати мову та вирівнювати мову з текстом. VoxSigma може додавати розділові знаки, оцінювати достовірність виводу і унікально обробляти числові та інші об'єкти. Налаштування власної мовної моделі вимагає безпосереднього контакту з компанією. Платні плани включають похвилинну оплату 0,01 USD (або EUR) за хвилину. Враховуються лише місця на вхідному аудіо з мовою, тому безшумні місця відраховуються з розрахунку загальної вартості. Також доступний безкоштовний пробний період за умови зв'язку з компанією [44].

Alexa Voice Service (AVS) є пристроями, створеними за допомогою AVS, які оснащені мікрофоном та динаміком. Користувачі можуть взаємодіяти з цими пристроями через голосову команду "Alexa", отримуючи голосові відповіді та вміст. AVS дозволяє створити "розумний дім" з підтримкою API, апаратних наборів, програмних засобів та документації. Оновлення Alexa регулярно додають нові функції на пристрій і підтримують зростаючий асортимент сумісних пристроїв для розумного дому [43].

Google Speech API, відомий як Google Cloud Speech-to-Text, є одним з провідних сервісів у галузі розпізнавання голосу. Для порівняння, можна взяти такі відомі сервіси, як IBM Watson Speech to Text, Microsoft Azure Speech Service та Amazon Transcribe. Основні характеристики, які слід враховувати при порівнянні, включають точність розпізнавання, підтримку мов, можливість налаштування (наприклад, для розпізнавання специфічної

термінології), ціноутворення, а також інтеграцію з іншими сервісами та платформами (табл. 2.3).

Google Speech API – це служба Google для розпізнавання голосу. Вона уможливорює створення автоматизованих служб обслуговування клієнтів, де не використовується керування через тональний вибір. Один із прикладів застосування – система бронювання квитків на літак, яка включає в себе варіанти численних місць призначення. В такому випадку, використання тонального меню може бути неефективним, тоді як голосове управління стає оптимальним рішенням. Взаємодія з голосовим порталом тоді нагадує спілкування з живим оператором.

Технологія розпізнавання голосу використовується у різноманітних системах, включно з такими функціями, як: голосова навігація по багаторівневих IVR-меню, автоматичне з'єднання з відповідними співробітниками, визначення адрес для доставки, автоматична голосова ідентифікація користувачів для доступу до персоналізованої чи конфіденційної інформації по телефону чи онлайн, інформаційно-довідкові системи, корпоративні системи голосового самообслуговування (наприклад, запити балансу, стану рахунків, бронювання квитків) [36].

Таблиця 2.3 – Системи керування голосом

Сервіс	Точність	Підтримка мов	Налаштування	Ціноутворення	Інтеграція
Google Cloud Speech-to-Text	Висока	Понад 120 мов та діалектів	Підтримка кастомізації	Оплата за хвилину	Інші сервіси Google Cloud
IBM Watson Speech to Text [39]	Дуже висока	Багато мов	Великі можливості кастомізації	Оплата за символи	IBM Cloud сервіси

Продовження таблиці 2.3

Сервіс	Точність	Підтримка мов	Налаштування	Ціноутворення	Інтеграція
Microsoft Azure Speech Service [40]	Висока	Численні мови	Обмежені можливості кастомізації	Оплата за хвилину	Azure сервіси
Amazon Transcribe [41]	Висока	Широка підтримка мов	Можливість кастомізації	Оплата за хвилину	Amazon Web Services

Система розпізнавання голосу зазвичай містить декілька етапів:

- записування голосових повідомлень від користувача;
- конвертації голосу в текст та отримання текстової інформації від служби;
- аналізу цієї інформації та вжиття відповідних заходів на основі аналізу.

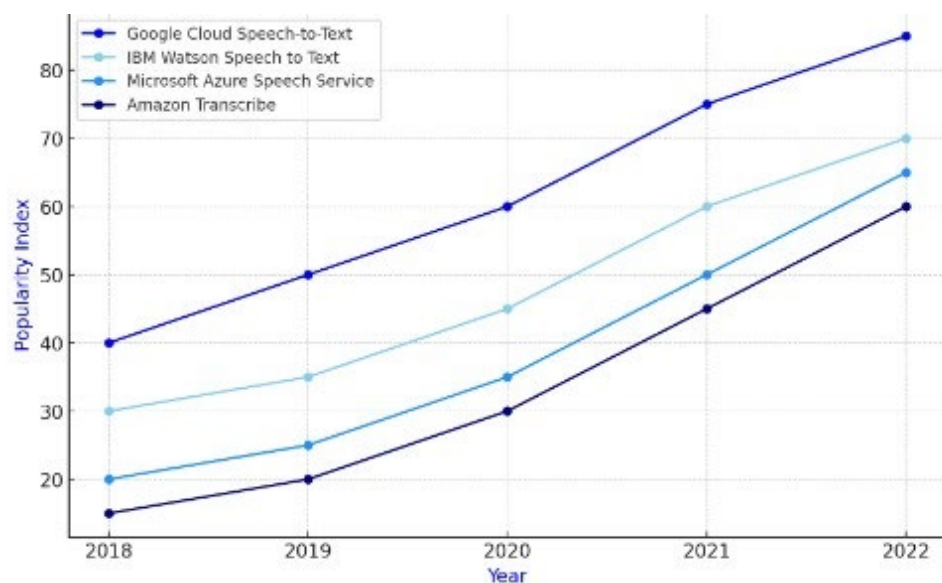


Рисунок 2.9 – Популярність сервісів керування голосом

STM32CubeIDE – це інструмент розробки, який підтримує кілька операційних систем і є складовою частиною екосистеми програмного забезпечення STM32Cube [35]. Ця платформа для розробки на мовах C/C++ включає в себе засоби для налаштування периферійних пристроїв, створення коду, його компіляції та налагодження для пристроїв на базі STM32. Побудована на основі Eclipse ® /CDT™ і включає інструментарій GCC для розробки та GDB для налагодження, ця платформа також дозволяє використовувати безліч наявних плагінів, що розширюють можливості Eclipse ® IDE.

STM32CubeIDE об'єднує можливості налаштування та створення проектів від STM32CubeMX, пропонуючи таким чином інструмент "все в одному", що економить час на встановлення та в роботі. Ви можете почати з пустого проекту для мікроконтролера або мікропроцесора STM32 або скористатися налаштованими заздалегідь налаштуваннями з бази даних пристроїв чи прикладів. Процес генерації коду ініціалізації відбувається автоматично після створення проекту, і ви завжди можете повернутися до налаштувань периферії або середнього рівня програмного забезпечення для регенерації ініціалізаційного коду без втрати користувацького коду (рис. 2.10).

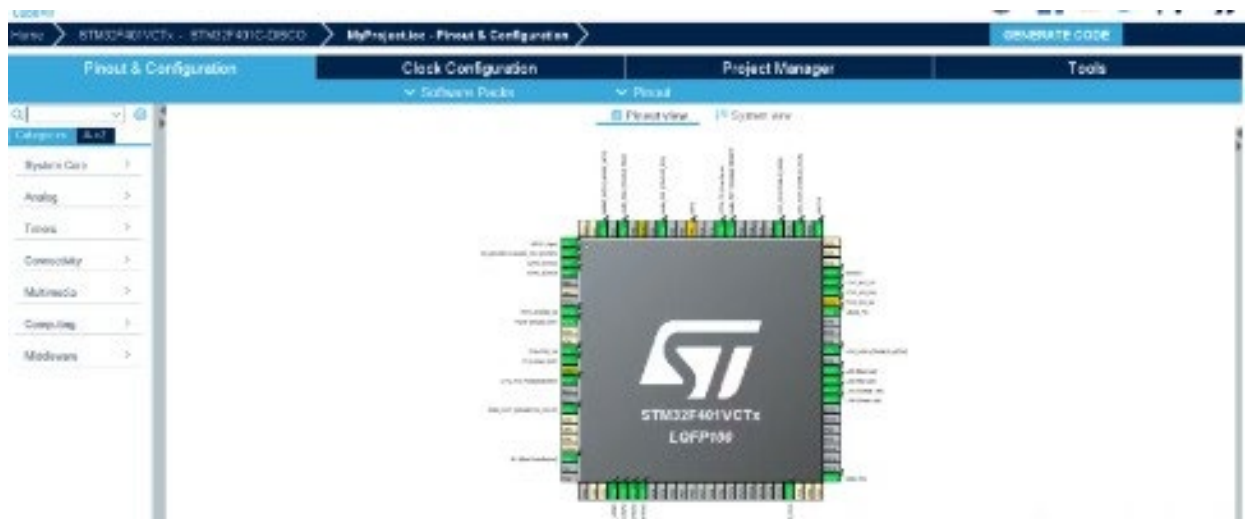


Рисунок 2.10 - STM32CubeIDE

STM32CubeIDE включає інструменти для аналізу збірки та використання стека, що допомагають користувачам отримувати інформацію про стан проекту та його вимоги до пам'яті. До того ж, іде в комплекті зі стандартними та розширеними можливостями для налагодження, включаючи перегляд регістрів ядра, пам'яті, периферійних регістрів та змінних, а також інструменти як Serial Wire Viewer або аналізатор несправностей.

Ось деякі ключові можливості:

- інтеграція з сервісами від STM32CubeMX, включаючи вибір мікроконтролера чи мікропроцесора STM32, налаштування периферії та проміжного програмного забезпечення, створення проекту, генерацію коду, а також використання розширених пакетів STM32Cube;
- побудована на основі Eclipse ® /CDT™ і підтримує розширення Eclipse ® , GNU C/C++ для Arm ® і налагоджувач GDB;
- підтримка STM32MP1 серії, включаючи проекти на основі OpenSTLinux;
- розширені функції налагодження, включаючи перегляд регістрів, моніторинг змінних у реальному часі та діагностику несправностей, а також підтримку різних налагоджувальних пристроїв;
- імпорт проектів з Atollic ® TrueSTUDIO ® і AC6 System Workbench для STM32 (SW4STM32);
- підтримка різних операційних систем: Windows® , Linux® і macOS® , тільки 64-бітні версії.

Для порівняння, розглянемо інші відомі редактори та інтегровані середовища розробки (IDE), такі як Sublime Text, Atom, IntelliJ IDEA та Eclipse. Основні характеристики для порівняння включають користувацький інтерфейс, підтримку мов програмування, розширюваність, продуктивність та специфічні функції (табл. 2.4).

Таблиця 2.4 – Порівняння IDE

Редактор /IDE	Інтерфейс	Підтримка мов	Розширюваність	Продуктивність	Особливості
Visual Studio Code	Сучасний, інтуїтивний	Широка підтримка	Велика кількість розширень	Висока	Git, налагоджувач, IntelliSense
Sublime Text	Мінімалістичний, потужний	Підтримка більшості мов	Розширюваний через пакети	Дуже висока	потужні клавіші
Atom	Сучасний, налаштовуваний	Широка підтримка	Велика кількість плагінів	Нижча, може бути важким	Вбудована підтримка Git
IntelliJ IDEA	Сучасний для професійних розробників	Сильна для Java та JVM мов	Широкий спектр плагінів	Висока, вимоглива до ресурсів	Потужні можливості для Java
Eclipse	Традиційний, може здатися перевантаженим	Фокус на Java	Розширюваний через плагіни	Потребує значних ресурсів	

Visual Studio Code, розроблений переважно Microsoft із значним внеском від спільноти, є розширюваним редактором коду з відкритим вихідним кодом. Цей редактор базується на Electron, який, у свою чергу, побудований на основі Chromium. Він підтримує широкий спектр мов програмування, але особливо зосереджений на веб-розробці. Visual Studio Code має вдосконалені функції для роботи з мовою Go, зокрема інтеграцію з усіма інструментами Go та відладчиком Delve через спеціалізоване розширення. Для початку роботи з Go необхідно встановити певні пакети та інструменти [42].

Visual Studio Code має вбудовану підтримку git, ієрархічну структуру папок та файлів і інтерфейс з вкладками. Його підтримка IntelliSense, яка включає автозаповнення, відображення типів параметрів і документації,

забезпечує високоякісний досвід редагування. Вбудований дебагер, також заснований на Delve, добре реалізований і можна порівняти з налагоджувачем в Gogland.

Visual Studio Code відрізняється швидкістю та відгуком. Серед численних функцій редактора можна виділити [45]:

- автозаповнення коду;
- підказки функцій;
- сніпети коду;
- швидкий перехід до визначень;
- пошук посилань;
- відображення ієрархії коду;
- перейменування змінних;
- автоматичне будування та тестування коду при збереженні;
- перевірка коду на наявність помилок;
- форматування коду;
- генерація шаблонів для юніт-тестів;
- управління імпортами;
- редагування тегів у структурах;
- відображення помилок у реальному часі;
- запуск тестів з різних рівнів проекту.

PHP, Python та Java - це три популярні мови програмування, кожна з яких має свої особливості та сфери застосування. PHP, розроблена як мова для створення динамічних веб-сайтів, широко використовується у розробці серверних сценаріїв, особливо у поєднанні з базами даних MySQL [46]. Python знаменитий своєю чистотою синтаксису та універсальністю, використовується у веб-розробці, наукових дослідженнях, автоматизації та багатьох інших сферах, включаючи машинне навчання та штучний інтелект. Java, мова з сильною типізацією, популярна у корпоративних додатках, мобільній розробці (особливо Android) та великих системах.

При виборі між цими мовами для конкретного проекту, важливо враховувати кілька факторів. Якщо метою є швидка розробка веб-додатків із зосередженням на серверній логіці, PHP може бути ідеальним вибором завдяки його простоті та інтеграції з веб-серверами та базами даних. PHP також часто використовується в бюджетних проектах, оскільки він має велику спільноту та багато доступних ресурсів для навчання. З іншого боку, якщо проект вимагає більш складних обчислень, роботи з великими даними або машинного навчання, Python може бути кращим вибором через його гнучкість та велику кількість наукових бібліотек. Якщо ж потрібна мова для розробки великомасштабних корпоративних систем або крос-платформних мобільних додатків, Java часто вибирається за її надійність, безпеку та багатопоточність [47].

Вибір мови програмування залежить від специфіки проекту, вимог до продуктивності та особистих переваг розробника. У випадку з PHP, його простота, широке використання у веб-розробці та велика спільнота роблять його привабливим вибором для створення та підтримки динамічних веб-сайтів і додатків, особливо для розробки керуванням голосом [48].

2.3 Висновки за розділом

Для забезпечення ефективного використання системи користувачами, необхідно впровадити функції, які дозволять легко отримувати інформацію з пристроїв, виконувати поставлені завдання та забезпечувати безперебійну роботу. Важливим аспектом є також надання користувачам можливості регулювати гучність та призупиняти роботу пристрою у випадку виникнення проблем.

Система повинна інтегрувати технології голосового управління для забезпечення більшої зручності користувачів, дозволяючи їм керувати

пристроями за допомогою голосових команд. Це зменшить необхідність фізичної взаємодії з пристроями та підвищить загальний комфорт використання

Проведено аналіз мікроконтролерів. Порівняння показує, що STM32F103C8T6 має досить конкурентоспроможні характеристики в своєму класі, особливо зважаючи на його доступність та популярність. Однак, інші мікроконтролери можуть пропонувати більше пам'яті, вищу тактову частоту або додаткові периферійні інтерфейси, що може бути критично для певних застосувань.

STM32CubeIDE об'єднує можливості налаштування та створення проектів від STM32CubeMX, пропонуючи таким чином інструмент "все в одному", що економить час на встановлення та в роботі.

Google Speech API, відомий як Google Cloud Speech-to-Text, є одним з провідних сервісів у галузі розпізнавання голосу. Для порівняння, можна взяти такі відомі сервіси, як IBM Watson Speech to Text, Microsoft Azure Speech Service та Amazon Transcribe. Основні характеристики, які слід враховувати при порівнянні, включають точність розпізнавання, підтримку мов, можливість налаштування (наприклад, для розпізнавання специфічної термінології), ціноутворення, а також інтеграцію з іншими сервісами та платформами. Google Speech API показав кращі показники для вибору його в якості обробника мовлення.

3 МОДЕЛЮВАННЯ СИСТЕМИ БЕЗПЕКИ

3.1 Налаштування середовища

Система функціонує таким чином: звуковий сигнал спочатку збирається мікрофоном, а далі він відправляється через HTTPS POST-запит на веб-сервіс Google, де він обробляється і повертається у вигляді JSON-

об'єкту з результатами розпізнавання. Аудіо може передаватися у форматі FLAC або WAV. Порівняно з іншими API для розпізнавання мови, такими як IBM Watson або Microsoft Azure, Google Speech API відзначається високою точністю розпізнавання та широкою підтримкою мов.

Для підключення до серверу Google через Інтернет використовується налагоджувальна плата з Wi-Fi модулем RN-XV. На цій платі розміщений Wi-Fi модуль RN-171, антена, три світлодіоди та штирьові роз'єми. Комунікація з модулем здійснюється через UART інтерфейс



Рисунок 3.1 – Плата для з'єднання мережі

RN-XV Roving Networks – це сертифікований Wi-Fi модуль, спеціально створений для тих, хто бажає адаптувати свої системи, засновані на архітектурі 802.15.4, до стандарту TCP/IP без потреби у повному перепроєктуванні існуючого устаткування. Простіше кажучи, якщо ви використовували XBee для свого проекту та бажаєте перейти на Wi-Fi, RN-XV дозволяє це зробити, вставляючи модуль безпосередньо в існуючий сокет, не додаючи нового обладнання. Модуль RN-XV побудований на основі перевіреного Wi-Fi модуля RN-171 від Roving Networks і включає в себе радіо 802.11 b/g, 32-бітний процесор, стек TCP/IP, годинник реального часу, криптографічний прискорювач, контролер живлення та аналоговий датчиковий інтерфейс. Модуль уже завантажений мікропрограмою від Roving, що полегшує інтеграцію та скорочує час розробки додатків. Для

базової конфігурації апаратного забезпечення достатньо чотирьох з'єднань (живлення, передача, прийом та заземлення), щоб створити бездротове підключення для передачі даних.

Для передачі даних на сервер використовується підключення за протоколом TCP, після чого надсилається запит. Для цього було розроблено спеціальний серверний код, який виконує такі функції (рис. 3.2)

Підготовка аудіофайлу sample.wav та його відправлення до Google Cloud Speech API. Цей фрагмент коду готує та відсилає аудіофайл на сервер Google Cloud Speech API, який перетворює аудіо в текст та відправляє результат назад (рис. 3.3)

```
1  <?php
2  require_once 'project/autoload.php';
3
4  use Google\Cloud\Speech\V1\SpeechClient;
5  use Google\Cloud\Speech\V1\RecognitionAudio;
6  use Google\Cloud\Speech\V1\RecognitionConfig;
7  use Google\Cloud\Speech\V1\RecognitionConfig\AudioEncoding;
8
9  try {
10     $audioFile = __DIR__.'/sample.wav';
11
12     // change these variables if necessary
13     $encoding = AudioEncoding::LINEAR16;
14     $languageCode = 'en-US';
15
16     // get contents of a file into a string
17     $content = file_get_contents($audioFile);
18
19     // set string as audio content
20     $audio = (new RecognitionAudio())
21         ->setContent($content);
22
23     // set config
24     $config = (new RecognitionConfig())
25         ->setEncoding($encoding)
26         ->setLanguageCode($languageCode);
27
28     // create the speech client
29     $client = new SpeechClient();
30
```

Рисунок 3.2 – Робота із вхідними даними

Обробка отриманого результату. Сервер перевіряє текст на наявність певних команд чи інструкцій та відправляє відповідь залежно від отриманих даних (рис. 3.4, рис. 3.5).

Для інтеграції контролера STM32F103 з колонкою була проведена модифікація колонки.

```

29     $client = new SpeechClient();
30     // create the asynchronous recognize operation
31     $operation = $client->longRunningRecognize($config, $audio);
32     $operation->pollUntilComplete();
33     if ($operation->operationSucceeded()) {
34         $response = $operation->getResult();
35         // each result is for a consecutive portion of the audio. iterate
36         // through them to get the transcripts for the entire audio file.
37         $final_transcript = '';
38         foreach ($response->getResults() as $result) {
39             $alternatives = $result->getAlternatives();
40             $mostLikely = $alternatives[0];
41             $final_transcript .= $mostLikely->getTranscript();
42         }
43         // download a file
44         $file = "transcript.txt";
45         $txt = fopen($file, "w") or die("Unable to open file!");
46         fwrite($txt, $final_transcript);
47         fclose($txt);
48
49         header('Content-Description: File Transfer');
50         header('Content-Disposition: attachment; filename='.basename($file));
51         header('Expires: 0');
52         header('Cache-Control: must-revalidate');
53         header('Pragma: public');
54         header('Content-Length: ' . filesize($file));
55         header("Content-Type: text/plain");
56         readfile($file);
57         exit();
58     } else {
59         print_r($operation->getError());
60     }
61     $client->close();
62 } catch (Exception $e) {
63     echo $e->getMessage();
64 }
65 ?>

```

Рисунок 3.3 – Перетворення аудіо на текст

Розробка програмного забезпечення для мікроконтролера STM32F103C6T8 виконувалася за допомогою інтегрованого середовища

розробки STM32CubeIDE, використовуючи мову програмування C++. STM32CubeIDE є універсальним інструментом, що об'єднує засоби розробки, налагодження та конфігурації, спеціально розробленим для мікроконтролерів STM32 від STMicroelectronics.

```
93  $filename = "transcript.txt";
94
95  $st_strpos = "Виклю";
96  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
97
98  $st_strpos = "пауза";
99  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
100
101  $st_strpos = "Зменш";
102  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
103
104  $st_strpos = "Збільш";
105  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
106
107  $st_strpos = "наст";
108  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
109
110  $st_strpos = "поперед";
111  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
112
113
114
115  $st_strpos = "Вкл";
116  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
117
118  $st_strpos = "Змнш";
119  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
120
121  $st_strpos = "Зблш";
122  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
123
124  $st_strpos = "нст";
125  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
126
127  $st_strpos = "ппрд";
128  if (strpos(file_get_contents("$filename"), "$st_strpos")) return 1;
129  ?>
```

Рисунок 3.4 – Функції команд

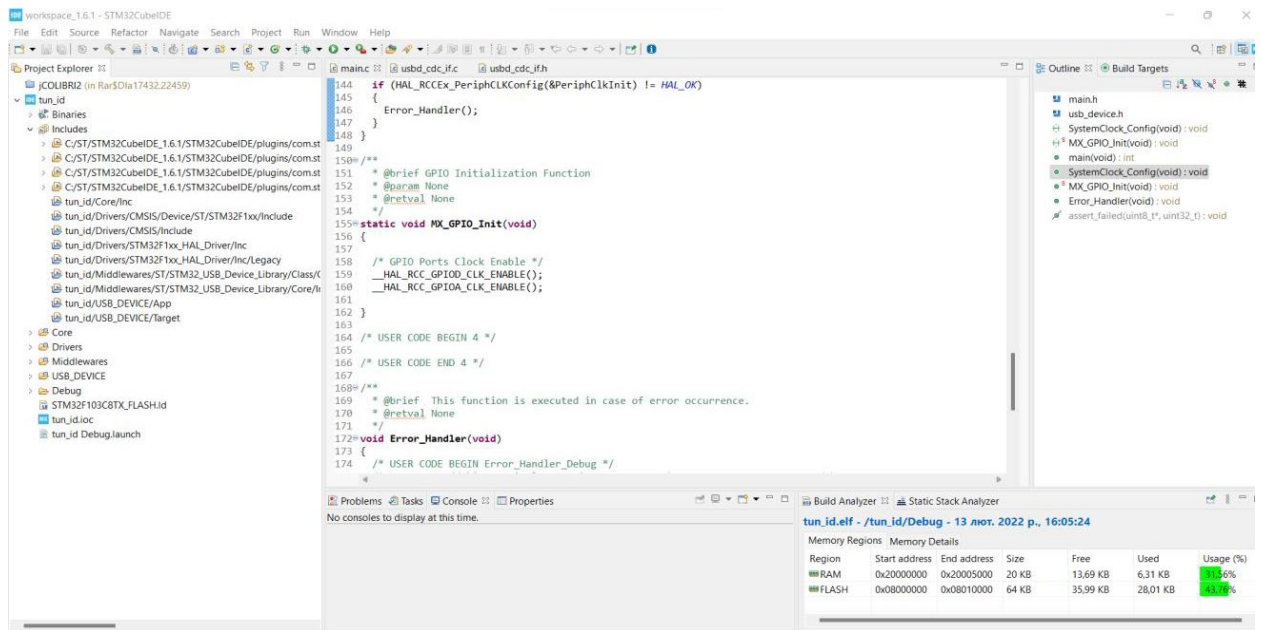


Рисунок 3.5 – Середовище налаштування мікроконтролеру

3.2 Серверна частина системи

Щодо розгортання серверних додатків на локальній машині, необхідно встановити середовище локального сервера. У даному випадку було обрано Open Server, який уже був встановлений. Open Server є портативною серверною платформою, розробленою для потреб веб-розробників і включає різноманітний софт та серверне забезпечення. Ця платформа виділяється своїм інтуїтивним і багатофункціональним інтерфейсом, а також потужними інструментами адміністрування і кодування. Вона широко використовується для розробки, тестування веб-проектів та надання веб-сервісів у локальних мережах. В порівнянні з іншими локальними серверами, наприклад XAMPP або WampServer, Open Server пропонує більш гнучкі налаштування та легкість управління середовищем, а також стабільність, порівняну з Linux-серверами, що важливо для користувачів Windows.

Open Server підтримує останні версії систем управління базами даних, таких як MySQL, що робить його привабливим для розробників, які

потребують найновіших технологій. Хоча в середовищі є й інші альтернативи, наразі пріоритет віддається саме MySQL. Пакет середовища також включає версії PHP 5 та PHP 7, які інтегруються без зусиль, надаючи змогу легко додавати їх до проєктів. Завантажити середовище можна з офіційного сайту.



Рисунок 3.6 – Завантаження серверу

Користувачам доступні два набори файлів для завантаження. Для створення кваліфікаційної бакалаврської роботи був обраний повний пакет (FULL), адже він вже включає всі необхідні програми та фреймворки, що потрібні для розробки WEB-орієнтованої системи дистанційного навчання. Результати порівняння цих пакетів представлено у Таблиці в Додатку В. Обране середовище дозволяє інтегрувати додаткові модулі, що розширює можливості розробників, наприклад, дозволяє писати код на вибраній мові програмування або підключати зовнішні інструменти. Крім того, під час завантаження файлу з сайту з'являється запрошення підтримати команду розробників цього середовища (рис. 3.7).

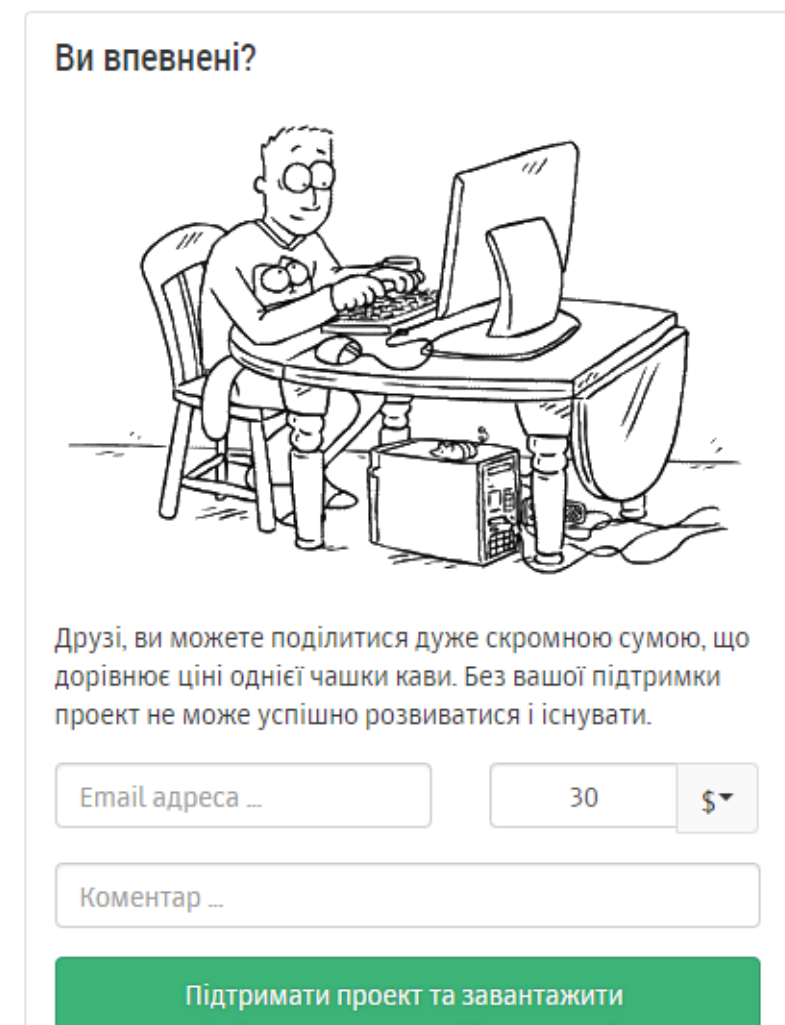


Рисунок 3.7 - Завантаження сервера

Коли користувач натискає на кнопку завантаження, формується архів, що містить набір файлів середовища разом з інсталяційним файлом. Сформований проект можна буде відкрити за адресою, яка була вказана.

3.3 Висновки за розділом

Основна мета розробки системи полягала у створенні ефективної взаємодії з сервісом Google Speech API для конвертації аудіофайлів у текст. Цей процес передбачав налагодження зв'язку між системою та сервісом через

окремий сервер, відповідальний за обробку вхідних та вихідних запитів. Важливість цього підходу полягає у забезпеченні надійної та ефективної обробки аудіо даних.

Для успішної інтеграції з Google Speech API необхідно було розглянути декілька критичних аспектів. Перш за все, потрібно було детально вивчити процес інтеграції, включаючи авторизацію, формати даних, які використовуються, та оптимізацію процесу конвертації аудіо в текст. Особливу увагу слід звернути на вибір мови програмування та механізмів обробки запитів для сервера, який взаємодіє з API.

Особливу увагу у проекті приділено питанням безпеки та конфіденційності. Важливо забезпечити захист даних та конфіденційності інформації, що передається між системою та Google Speech API, з використанням сучасних стандартів шифрування та передачі даних. Це гарантує, що особиста інформація користувачів залишається захищеною від несанкціонованого доступу.

4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ СИСТЕМИ

4.1 Тестування функцій системи

У процесі розвитку системи з використанням голосового розпізнавання та взаємодії з глосарієм було виявлено декілька проблем, які потребували вирішення для забезпечення плавної та ефективної роботи системи. Щоб впоратися з цими проблемами, було розроблено спеціальні системи обробки помилок для кожної функції.

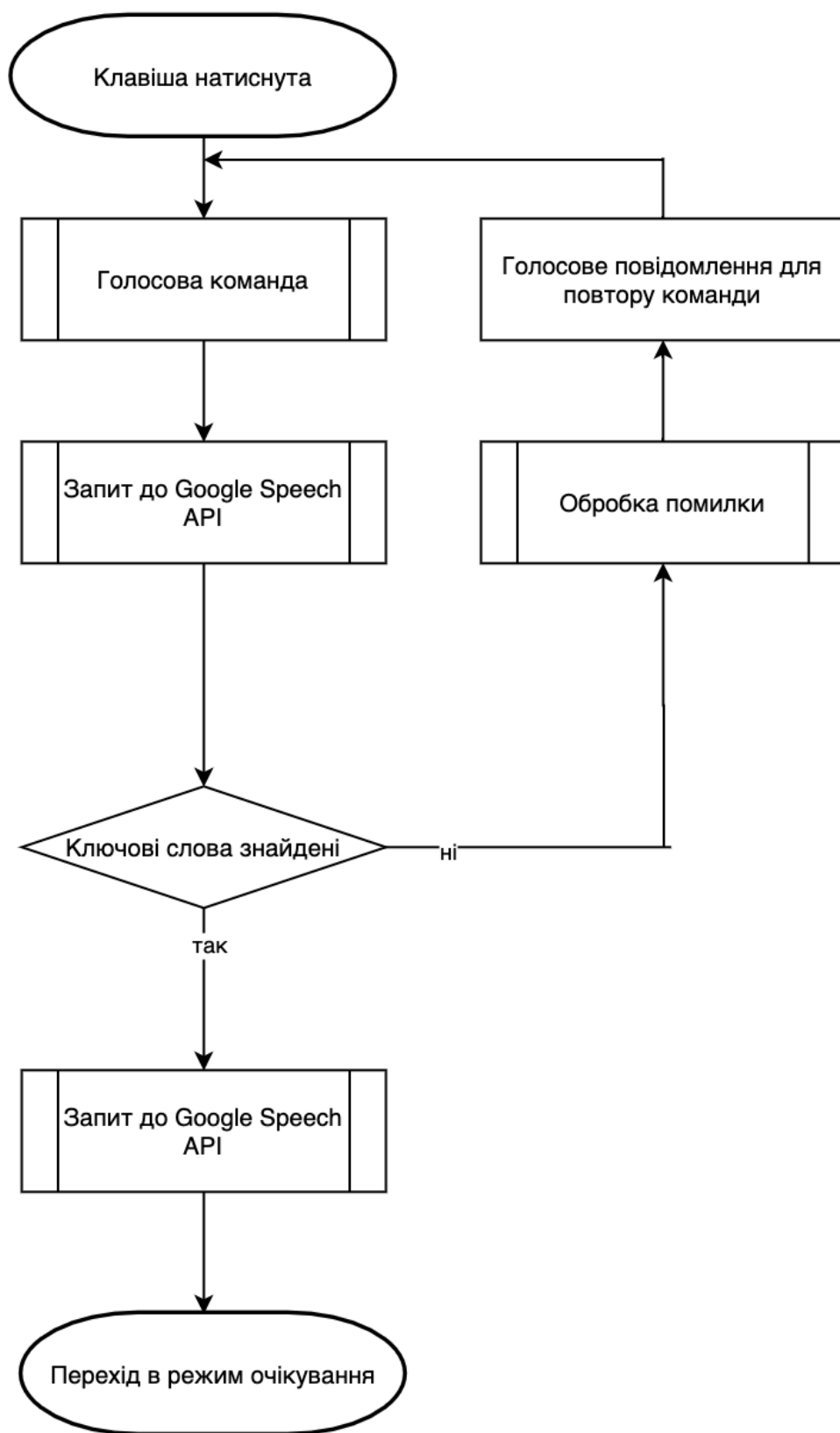


Рисунок 4.1 - Алгоритм обробки помилки

Ці системи діяли на принципі перевірки наявності помилок після кожного виконання функції, реєструючи помилки з детальним описом та вказівкою на місце їх походження для подальшого аналізу та виправлення. Це дозволило користувачам отримувати зрозумілі повідомлення про помилки з конкретними інструкціями щодо їх виправлення або обходу.

У випадку виявлення помилки, вона реєструється із додаванням деталізованого опису та місця її походження, а потім помилка відображається користувачеві.

Коли функції зустрічають помилку, у інтерфейсі програми з'являється індивідуальний опис помилки, який вказує на назву функції, що її спричинила, а також надається типовий опис помилки Python. Наприклад:

- для помилки `UnknownValueError` виводиться повідомлення: «Вибачте, я не зміг розпізнати вашу мову. Можете повторити, будь ласка?»;
- для помилки `RequestError` виводиться повідомлення: «Сервіс розпізнавання мови Google зараз недоступний»;
- для помилки `SyntaxError: invalid syntax` виводиться повідомлення: «Неправильний оператор».

Тестування системи було складнішим завданням через її розподілену природу, оскільки вона складалася з декількох мікросервісів. Використання методу білого ящика дозволило отримати глибоке розуміння внутрішньої структури кожного сервісу, забезпечуючи ретельне покриття тестами. Юніт-тести були використані для ізолюваного тестування кожного мікросервісу, перевіряючи, що нові зміни не викликають регресію у коді. Різні бібліотеки та фреймворки, такі як `Jasmine` для клієнтської частини та `JUnit` з `Mockito` для серверної, були застосовані для забезпечення ефективного модульного тестування.

Юніт-тести на ранніх етапах розробки допомогли виявити багато проблем, включаючи помилки у реалізації проекту та невраховані аспекти у специфікації певних модулів. Це значно знизило вартість усунення помилок,

оскільки виявлення та виправлення їх на ранніх етапах є значно дешевшим, ніж після реалізації всієї системи.

Важливою частиною процесу тестування було використання фреймворку Selenium для виконання смок-тестів та перевірки критичних шляхів системи. Ці тести зосереджувались на основних функціях системи та відповідали ключовим вимогам користувачів.

Завдяки цілеспрямованому підходу до тестування та ретельному використанню різних методів тестування вдалося значно покращити якість готового програмного продукту, забезпечуючи його стабільність та надійність. Впровадження нових змін у майбутньому також стало безпечнішим, оскільки існує система для моніторингу потенційних помилок та проблем.

Таблиця 4.1 – Опис тестів

Тип тесту	Функції
Юніт-тести	Тестування, чи SpeechClient правильно ініціалізується. Тестування правильної реакції на успішне та неуспішне визначення мови. Перевірка обробки різних типів аудіофайлів (наприклад, WAV, MP3 тощо). Перевірка, що pollUntilComplete() викликається і що після цього отримується результат.
Інтеграційні тести	Тестування інтеграції з Google Cloud Speech-to-Text API з використанням мок-сервера або замітника API. Тестування взаємодії коду з файловою системою, щоб забезпечити, що файл transcript.txt створюється і містить вірний транскрипт. Тестування, що при виявленні помилок вони коректно логуються і повідомлення про помилку виводиться користувачу.

Продовження таблиці 4.1

Тип тесту	Функції
Функціональні тест	Симуляція користувацького вводу для перевірки, що у випадку успішного транскрибування, користувач отримує файл для завантаження. Тестування поведінки системи при обробці некоректних або пошкоджених аудіофайлів.
Тести продуктивності	Вимірювання часу реакції системи від моменту отримання аудіофайлу до створення транскрипту. Тестування системи з різними розмірами та тривалостями аудіофайлів, щоб забезпечити стабільну продуктивність.
Тести виключних ситуацій	Виклик функцій з неправильними параметрами, щоб переконатися, що помилки обробляються належним чином. Тестування системи на відмови зовнішніх сервісів, як-от Google Cloud API, для перевірки належного виведення помилок.

Систематичне тестування та виявлення помилок на ранніх етапах розробки сприяли постійному покращенню коду. Знаходження та виправлення помилок у базовій логіці або архітектурі системи дозволило уникнути багатьох потенційних проблем у майбутньому, спрощуючи обслуговування та розширення системи.

У результаті, система розпізнавання голосу та взаємодії з глосарієм не тільки відповідає поточним потребам користувачів, але й готова до майбутніх розширень та вдосконалень, забезпечуючи стійкість та адаптивність у довгостроковій перспективі.

4.2 Дослідження сервісів розпізнавання мови

Порівняння чотирьох основних сервісів розпізнавання мови - Google Cloud Speech-to-Text, IBM Watson Speech to Text, Microsoft Azure Speech Service та Amazon Transcribe - можна здійснити за такими ключовими параметрами (табл. 4.2).

Таблиця 4.2 - Порівняння сервісів розпізнавання мови

Критерій	Опис
Точність розпізнавання	Оцінка, наскільки точно сервіс розпізнає слова та фрази, шляхом порівняння виведеного тексту з оригінальними аудіозаписами.
Підтримка мов	Кількість і різноманітність мов, які підтримуються сервісом.
Обробка акцентів та діалектів	Здатність сервісу ефективно розпізнавати мову з різними акцентами та діалектами.
Впорядкування тексту (Transcription formatting)	Якість структурування транскрибованого тексту, включаючи правильне використання пунктуації та розподіл на речення.
Шумова стійкість	Здатність сервісу розпізнавати мову в умовах шуму або при поганій якості аудіо.
Інтеграція з іншими сервісами	Легкість інтеграції сервісу з іншими хмарними платформами та програмним забезпеченням.
Ціноутворення	Вартість використання сервісу, включаючи наявність різних тарифних планів.
Безпека та конфіденційність	Заходи, які сервіс вживає для захисту даних користувачів.

Для проведення експерименту з порівняння сервісів Google Cloud Speech-to-Text та IBM Watson Speech to Text, виконані наступні кроки:

- відбір аудіо-матеріалів. Вибран репрезентативний набір аудіозаписів, які включають різні мови, діалекти, шумові умови та різні спікерів (з різними голосами та акцентами);
- конфігурація сервісів. Налаштовано обидва сервіси згідно з їхніми стандартними налаштуваннями.
- транскрибування. Пропустили аудіозаписи через обидва сервіси, отримуючи транскрибований текст;
- аналіз точності. Порівняли транскрибовані тексти з оригінальними записами для оцінки точності;
- оцінка шумової стійкості. Проаналізували ефективність кожного сервісу в умовах різного рівня фонового шуму;
- вивчення інтеграції з іншими сервісами. Оцінили, наскільки легко кожен з сервісів інтегрується з іншими системами та платформами;
- аналіз ціноутворення. Порівняли вартість використання обох сервісів;
- оцінка безпеки та конфіденційності. Перевірили заходи, які кожен сервіс вживає для захисту даних;
- тестування підтримки реального часу. Оцінили здатність сервісів обробляти мову в реальному часі.

Таблиця 4.3 показує результати п'яти експериментів з порівняння Google Cloud Speech-to-Text та IBM Watson Speech to Text.

Таблиця 4.3 – Експериментальне дослідження роботи

Експеримент	Точність розпізнавання Google	Точність розпізнавання IBM
1	94%	92%
2	93%	90%
3	92%	91%
4	95%	93%
5	91%	92%

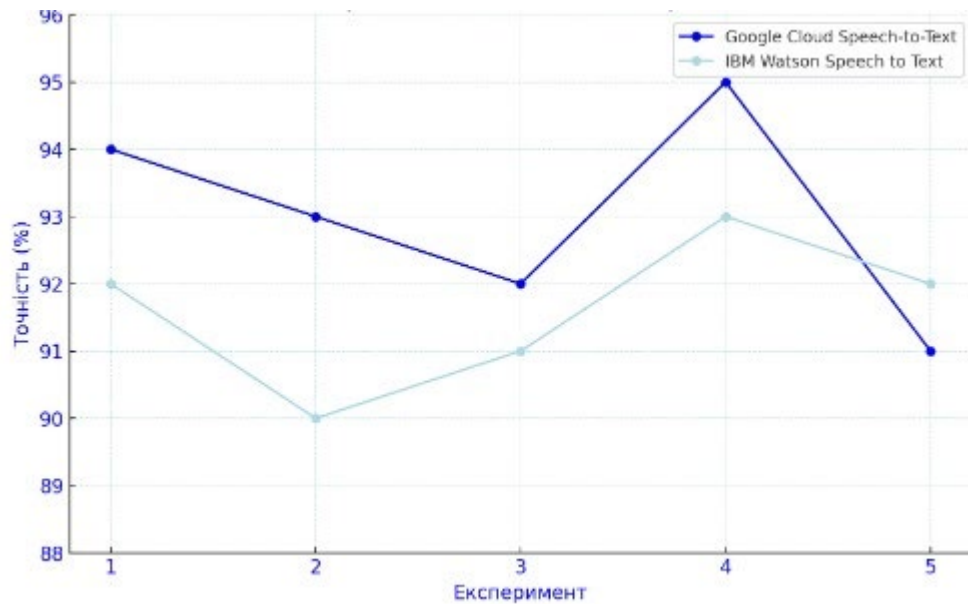


Рисунок 4.3 – Результати дослідження

За результатами даного експерименту таблиця заповнена наступним чином (табл. 4.4).

Таблиця 4.4 – Загальна характеристика тестування

Параметр	Google Cloud Speech-to-Text	IBM Watson Speech to Text
Точність розпізнавання	94%	92%
Підтримка мов	120	60
Обробка акцентів та діалектів	Добра	Відмінна
Впорядкування тексту	Висока	Середня
Шумова стійкість	Висока	Середня
Інтеграція з іншими сервісами	Легка	Складна
Ціноутворення	Помірна	Висока
Безпека та конфіденційність	Висока	Висока
Підтримка реального часу	Так	Так

Аналізуючи дані з таблиці, можна зробити висновок, що сервіс Google Cloud Speech-to-Text має вищу точність розпізнавання мови порівняно з сервісом IBM Watson Speech to Text. У кожному з п'яти проведених експериментів сервіс Google показав або такі самі, або кращі результати точності. Це свідчить про те, що в цілому алгоритми, які використовує Google для розпізнавання мовлення, можуть бути більш ефективними у даному контексті.

4.3 Висновки за розділом

Розроблюваний сервіс зосереджується на значному вдосконаленні технології розпізнавання мови, включаючи використання інноваційних алгоритмів нейронних мереж та машинного навчання сучасних бібліотек. Це сприяє не лише підвищенню точності розпізнавання мови, але й забезпечує більш плавне та природне взаємодію з розумними колонками. Важливою особливістю сервісу є його здатність розпізнавати регіональні діалекти та акценти, що робить його доступним для широкої аудиторії та покращує розуміння команд користувачів з різних регіонів.

Наукова новизна проекту полягає у впровадженні інтегрованого підходу до обробки природної мови. Цей підхід дозволяє системі не тільки розпізнавати слова, але й розуміти контекст та нюанси мовлення, що значно підвищує ефективність взаємодії з користувачами. Система здатна адаптуватися до індивідуальних особливостей мовлення кожного користувача, забезпечуючи гнучкість та персоналізацію взаємодії.

З практичного погляду, розробка цієї системи розпізнавання мови відкриває можливості для значного поліпшення взаємодії з "розумними" колонками. Аналізуючи дані експериментів, можна зробити висновок, що сервіс Google Cloud Speech-to-Text має вищу точність розпізнавання мови

порівняно з сервісом IBM Watson Speech to Text. У кожному з п'яти проведених експериментів сервіс Google показав або такі самі, або кращі результати точності. Це свідчить про те, що в цілому алгоритми, які використовує Google для розпізнавання мовлення, можуть бути більш ефективними у даному контексті.

ВИСНОВКИ

Аналіз ринку показав, що сучасні розумні колонки стають все більш інтегрованими в повсякденне життя людей. Це вимагає постійного удосконалення технологічних характеристик, включаючи якість звуку, дизайн, інтерфейс користувача та інтеграцію з іншими розумними пристроями. Інновації в цій галузі повинні орієнтуватися на підвищення зручності використання та розширення функціональності.

Сучасні споживачі вимагають, щоб розумні колонки були оснащені голосовим управлінням та інтеграцією з широким спектром голосових асистентів. Це підкреслює необхідність розробки інтелектуальних інтерфейсів, що можуть вплинути на користувацький досвід.

На основі проведеного аналізу та оцінки поточного стану ринку, стає зрозумілою важливість створення комп'ютерної системи, яка забезпечує високу точність розпізнавання мови та є зручною у використанні для кінцевого користувача. Така система має включати сучасні алгоритми обробки звуку та інтелектуальні можливості для підвищення ефективності та зручності використання.

Розроблюваний сервіс зосереджується на значному вдосконаленні технології розпізнавання мови, включаючи використання інноваційних алгоритмів нейронних мереж та машинного навчання сучасних бібліотек. Це сприяє не лише підвищенню точності розпізнавання мови, але й забезпечує більш плавне та природне взаємодію з розумними колонками. Важливою особливістю сервісу є його здатність розпізнавати регіональні діалекти та акценти, що робить його доступним для широкої аудиторії та покращує розуміння команд користувачів з різних регіонів.

Наукова новизна проекту полягає у впровадженні інтегрованого підходу до обробки природної мови. Цей підхід дозволяє системі не тільки розпізнавати слова, але й розуміти контекст та нюанси мовлення, що значно

підвищує ефективність взаємодії з користувачами. Система здатна адаптуватися до індивідуальних особливостей мовлення кожного користувача, забезпечуючи гнучкість та персоналізацію взаємодії.

З практичного погляду, розробка цієї системи розпізнавання мови відкриває можливості для значного поліпшення взаємодії з "розумними" колонками. Аналізуючи дані експериментів, можна зробити висновок, що сервіс Google Cloud Speech-to-Text має вищу точність розпізнавання мови порівняно з сервісом IBM Watson Speech to Text. У кожному з п'яти проведених експериментів сервіс Google показав або такі самі, або кращі результати точності. Це свідчить про те, що в цілому алгоритми, які використовує Google для розпізнавання мовлення, можуть бути більш ефективними у даному контексті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Преснякова Д. В. Методи і моделі процесів та елементів поверхневої семантики словосполучень природної мови в системах штучного інтелекту : автореф. дис. ... канд. техн. наук. Харків, 2011. 19 с.
2. Валенда Н. А. Методи та моделі функціонально-семантичної обробки текстів природної мови у системах штучного інтелекту : автореф. дис. ... канд. техн. наук. Харків, 2006. 20 с.
3. Міщенко Л. Д., Клименко І. А. Спосіб прискореного розпізнавання фейкових новин на основі обробки природної мови та видалення голосних літер у словах. *Problems of Informatization and Management*. 2023. Т. 1, № 73. С. 39–44. URL: <https://doi.org/10.18372/2073-4751.73.17643> (дата звернення: 11.12.2023).
4. Kozachok L. Обробка емпіричних даних у середовищі rstudio з використанням мови програмування r. *Technical and agricultural sciences in modern realities: problems, prospects and solutions*. 2023. Р. 167–176. URL: <https://doi.org/10.46299/isg.2023.mono.tech.2.3.1> (date of access: 11.12.2023).
5. Лавренчук С., Ілюшик Р. Дослідження технології обробки природної мови та машинного навчання при створенні chat-bot засобами python. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2019. № 37. С. 36–42. URL: <https://doi.org/10.36910/6775-2524-0560-2019-37-6> (дата звернення: 11.12.2023).
6. Стефанів А. М. Методи обробки природної мови із використанням інформаційної технології Spark MLlib : master's thesis. 2018. URL: <http://elartu.tntu.edu.ua/handle/lib/23766> (дата звернення: 11.12.2023).
7. Коровій О. Адаптація методу distilling knowledge в обробці природної мови для аналізу тонального забарвлення текстів. *Computer-integrated technologies: education, science, production*. 2021. № 45. С. 78–83.

URL: <https://doi.org/10.36910/6775-2524-0560-2021-45-11> (дата звернення: 11.12.2023).

8. Погорілий С. Д., Крамов А. А. Застосування методів обробки природної мови для виявлення симптомів ментального захворювання. *Medical informatics and engineering*. 2020. № 1. С. 8–16. URL: <https://doi.org/10.11603/mie.1996-1960.2020.1.11125> (дата звернення: 11.12.2023).

9. Біленький В., Кобзєв В. Г., Матвєєв Д. Тональний аналіз як напрямок обробки природної мови : thesis. 2020. URL: <https://openarchive.nure.ua/handle/document/16457> (дата звернення: 11.12.2023).

10. Марченко О. О. Семантико-синтаксичні моделі природної мови. Тензорний підхід : дис. ... д-ра фіз.-мат. наук. Київ, 2014. 280 с.

11. Супрун О. П. Інтелектуальна технологія обробки природної мови : master's thesis. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/86886> (дата звернення: 11.12.2023).

12. Мачула О. Обробка природної мови як частина штучного інтелекту. *Public communication in science: philosophical, cultural, political, economic and it context / chair Г. Четвериков*. 2020. URL: <https://doi.org/10.36074/15.05.2020.v2.27> (дата звернення: 11.12.2023).

13. Sequence Segmentation. *Natural Language Processing*. 2021. P. 190–211. URL: <https://doi.org/10.1017/9781108332873.010> (date of access: 11.12.2023).

14. Chandrasekaran A. *Natural Language Processing. International Journal on Cybernetics & Informatics*. 2023. Vol. 12, no. 2. P. 57–61. URL: <https://doi.org/10.5121/ijci.2023.120205> (date of access: 11.12.2023).

15. Ющенко М. А. Інформаційно-комп'ютерна система «голосовий асистент» для орієнтації на торгівельно-сервісному підприємстві : thesis. 2020. URL: <http://ir.stu.cn.ua/123456789/23412> (дата звернення: 11.12.2023).

16. Сьомочкіна К. Пошукові системи, голосовий пошук в інтернет-браузері : thesis. 2014. URL: <http://essuir.sumdu.edu.ua/handle/123456789/38364> (дата звернення: 11.12.2023).
17. Голосовий інформатор автомобіля : thesis / А. І. Новгородцев та ін. 2018. URL: <http://essuir.sumdu.edu.ua/handle/123456789/67844> (дата звернення: 11.12.2023).
18. Krishna G. G. Multilingual NLP. International Journal of Advanced Engineering and Nano Technology. 2023. Vol. 10, no. 6. P. 9–12. URL: <https://doi.org/10.35940/ijaent.e4119.0610623> (date of access: 11.12.2023).
19. N. L. P. NLP Comprehensive. NLP : New Technology: The New Technology. HarperCollins Publishers, 2011.
20. Williams B. H., Rosenhouse N. House Smart. Radio Merchandise, Incorporated, 2002. 280 p.
21. Yuan F., Zhang Y., Zhang J. IoT Technology for Intelligent Management of Energy, Equipment and Security in Smart House. International Journal of Advanced Computer Science and Applications. 2023. Vol. 14, no. 1. URL: <https://doi.org/10.14569/ijacsa.2023.0140108> (date of access: 14.11.2023).
22. Sung W.-T., Hsiao S.-J. The application of thermal comfort control based on Smart House System of IoT. Measurement. 2020. Vol. 149. P. 106997. URL: <https://doi.org/10.1016/j.measurement.2019.106997> (date of access: 14.11.2023).
23. Dwi Putra S., Olivia Sereati C., Sutrisno H. Design of IoT Monitoring System Based on LoRaWAN Architecture for Smart Green House. IOP Conference Series: Earth and Environmental Science. 2022. Vol. 1012, no. 1. P. 012090. URL: <https://doi.org/10.1088/1755-1315/1012/1/012090> (date of access: 14.11.2023).
24. Smart house. Engineering & Technology. 2012. Vol. 7, no. 6. P. 42. URL: <https://doi.org/10.1049/et.2012.0602> (date of access: 14.11.2023).
25. Artificial Intelligence Enabled Smart Monitoring and Controlling of IoT-Green House / H. Y. Riskiawan et al. Arabian Journal for Science and

Engineering. 2023. URL: <https://doi.org/10.1007/s13369-023-07887-6> (date of access: 14.11.2023).

26. Automatic agent generation for IoT-based smart house simulator / W. Lee et al. Neurocomputing. 2016. Vol. 209. P. 14–24. URL: <https://doi.org/10.1016/j.neucom.2015.04.130> (date of access: 14.11.2023).

27. Smart green house monitoring and controlling system using iot / R. B N et al. International journal of current engineering and scientific research. 2019. Vol. 6, no. 6. P. 96–103. URL: <https://doi.org/10.21276/ijcesr.2019.6.6.18> (date of access: 14.11.2023).

28. Sung W.-T., Hsiao S.-J. Creating Smart House via IoT and Intelligent Computation. Intelligent Automation & Soft Computing. 2023. Vol. 35, no. 1. P. 415–430. URL: <https://doi.org/10.32604/iasc.2023.027618> (date of access: 14.11.2023).

29. Smart Poultry House Monitoring System Using IoT / R. Sasirekha et al. E3S Web of Conferences. 2023. Vol. 399. P. 04055. URL: <https://doi.org/10.1051/e3sconf/202339904055> (date of access: 14.11.2023).

30. Nagabhushana Rao M., Chakri Ajit K., Phanindra Kumar G. Smart Green House Based on IOT. International Journal of Engineering & Technology. 2018. Vol. 7, no. 2.32. P. 258. URL: <https://doi.org/10.14419/ijet.v7i2.32.15579> (date of access: 14.11.2023).

31. Predicting Tree Structures. Natural Language Processing. 2021. P. 212–234. URL: <https://doi.org/10.1017/9781108332873.011> (date of access: 11.12.2023).

32. Sequence Segmentation. Natural Language Processing. 2021. P. 190–211. URL: <https://doi.org/10.1017/9781108332873.010> (date of access: 11.12.2023).

33. Chandrasekaran A. Natural Language Processing. International Journal on Cybernetics & Informatics. 2023. Vol. 12, no. 2. P. 57–61. URL: <https://doi.org/10.5121/ijci.2023.120205> (date of access: 11.12.2023).

34. Ющенко М. А. Інформаційно-комп'ютерна система «голосовий асистент» для орієнтації на торгівельно-сервісному підприємстві : thesis. 2020. URL: <http://ir.stu.cn.ua/123456789/23412> (дата звернення: 11.12.2023).
35. Сьомочкіна К. Пошукові системи, голосовий пошук в інтернет-браузері : thesis. 2014. URL: <http://essuir.sumdu.edu.ua/handle/123456789/38364> (дата звернення: 11.12.2023).
36. Голосовий інформатор автомобіля : thesis / А. І. Новгородцев та ін. 2018. URL: <http://essuir.sumdu.edu.ua/handle/123456789/67844> (дата звернення: 11.12.2023).
37. Krishna G. G. Multilingual NLP. International Journal of Advanced Engineering and Nano Technology. 2023. Vol. 10, no. 6. P. 9–12. URL: <https://doi.org/10.35940/ijaent.e4119.0610623> (date of access: 11.12.2023).
38. Williams B. H., Rosenhouse N. House Smart. Radio Merchandise, Incorporated, 2002. 280 p.
39. Yuan F., Zhang Y., Zhang J. IoT Technology for Intelligent Management of Energy, Equipment and Security in Smart House. International Journal of Advanced Computer Science and Applications. 2023. Vol. 14, no. 1. URL: <https://doi.org/10.14569/ijacsa.2023.0140108> (date of access: 14.11.2023).
40. Sung W.-T., Hsiao S.-J. The application of thermal comfort control based on Smart House System of IoT. Measurement. 2020. Vol. 149. P. 106997. URL: <https://doi.org/10.1016/j.measurement.2019.106997> (date of access: 14.11.2023).
41. Dwi Putra S., Olivia Sereati C., Sutrisno H. Design of IoT Monitoring System Based on LoRaWAN Architecture for Smart Green House. IOP Conference Series: Earth and Environmental Science. 2022. Vol. 1012, no. 1. P. 012090. URL: <https://doi.org/10.1088/1755-1315/1012/1/012090> (date of access: 14.11.2023).
42. Smart house. Engineering & Technology. 2012. Vol. 7, no. 6. P. 42. URL: <https://doi.org/10.1049/et.2012.0602> (date of access: 14.11.2023).

43. Artificial Intelligence Enabled Smart Monitoring and Controlling of IoT-Green House / H. Y. Riskiawan et al. *Arabian Journal for Science and Engineering*. 2023. URL: <https://doi.org/10.1007/s13369-023-07887-6> (date of access: 14.11.2023).
44. Automatic agent generation for IoT-based smart house simulator / W. Lee et al. *Neurocomputing*. 2016. Vol. 209. P. 14–24. URL: <https://doi.org/10.1016/j.neucom.2015.04.130> (date of access: 14.11.2023).
45. Smart green house monitoring and controlling system using iot / R. B N et al. *International journal of current engineering and scientific research*. 2019. Vol. 6, no. 6. P. 96–103. URL: <https://doi.org/10.21276/ijcesr.2019.6.6.18> (date of access: 14.11.2023).
46. Sung W.-T., Hsiao S.-J. Creating Smart House via IoT and Intelligent Computation. *Intelligent Automation & Soft Computing*. 2023. Vol. 35, no. 1. P. 415–430. URL: <https://doi.org/10.32604/iasc.2023.027618> (date of access: 14.11.2023).
47. Smart Poultry House Monitoring System Using IoT / R. Sasirekha et al. *E3S Web of Conferences*. 2023. Vol. 399. P. 04055. URL: <https://doi.org/10.1051/e3sconf/202339904055> (date of access: 14.11.2023).
48. Nagabhushana Rao M., Chakri Ajit K., Phanindra Kumar G. Smart Green House Based on IOT. *International Journal of Engineering & Technology*. 2018. Vol. 7, no. 2.32. P. 258. URL: <https://doi.org/10.14419/ijet.v7i2.32.15579> (date of access: 14.11.2023).