

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ПОГОДЖЕНО

**Декан факультету Інформаційних
технологій**

_____ Ігор БОЛБОТ
(підпис)

«__» _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

**Завідувач кафедри Комп'ютерних систем,
мереж та кібербезпеки**

_____ Дмитро КАСАТКІН
(підпис)

«__» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

На тему: «Розробка голосової системи керування елементами розумного будинку»

Спеціальність F7 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерна інженерія»

Гарант освітньої програми

канд. фіз.-мат. наук, доцент

_____ **Євгеній НІКІТЕНКО**

**Керівник бакалаврської
кваліфікаційної роботи**

канд. фіз.-мат. наук, доцент

_____ **Євгеній НІКІТЕНКО**

Виконав

_____ **Андрій МАКСИМЕНКО**

КИЇВ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**
Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних систем, мереж та кібербезпеки
канд. пед. наук, доцент _____ Дмитро КАСАТКІН
« ____ » _____ 20 ____ р.

З А В Д А Н Н Я

**ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Максименку Андрію Олександровичу

(прізвище, ім'я, по батькові)

Спеціальність «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерна інженерія»

Тема кваліфікаційної бакалаврської роботи

«Розробка голосової системи керування елементами розумного будинку»

затверджена наказом ректора НУБіП України від «16» 12 2026 р. № 2250
«С»

Термін подання завершеної роботи на кафедру «16» 12 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи

Перелік питань, що підлягають дослідженню:

Перелік графічного матеріалу (за необхідності).

Дата видачі завдання “ 16 ” 12 2026 р.

**Керівник бакалаврської
кваліфікаційної роботи**
канд. фіз.-мат. наук, доцент

(підпис)

Євгеній НІКІТЕНКО

Завдання взяв до виконання

(підпис)

Андрій МАКСИМЕНКО

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області	04.02.2026 р.	Виконано
2	Проектування системи	15.03.2026 р.	Виконано
3	Реалізація системи	10.04.2026 р.	Виконано
4	Тестування системи	01.05.2026 р.	Виконано
5	Оформлення пояснювальної записки	25.05.2026 р.	Виконано
6	Оформлення графічного матеріалу	01.06.2026 р.	Виконано

Студент

_____ **А. О. Максименко**
(підпис) (ініціали та прізвище)

Керівник проекту (роботи)

_____ **Є. В. Нікітенко**
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Об'єкт дослідження: Розробка голосової системи керування елементами розумного будинку

Мета роботи: Дослідити і розробити прототип системи з голосовим керуванням елементами розумного будинку на базі одноплатного комп'ютера, що забезпечує прийом, обробку голосових команд та керування виконавчими пристроями.

Проект складається з п'ятих розділів.

Перший розділ присвячено аналізу проблемної області, зокрема розглядаються принципи функціонування систем розумного будинку та особливості використання голосового керування.

У другому розділі описано варіанти реалізації системи, проведено вибір апаратного та програмного забезпечення, а також визначено загальну структуру системи.

Третій розділ присвячено огляду й вибору апаратних та програмних компонентів: детально аналізуються обчислювальна платформа, периферійні пристрої, а також засоби реалізації голосового інтерфейсу (STT, TTS).

Четвертий розділ присвячено процесу проектування, розробки й реалізації системи, зокрема налаштуванню середовища, підключенню компонентів та реалізації алгоритмів опрацювання голосових команд.

П'ятий розділ присвячено тестуванню системи, аналізу результатів її роботи та формуванню рекомендацій щодо подальшого вдосконалення.

Результатом виконання дипломної роботи є розробка прототипу системи голосового керування елементами розумного будинку

ЗМІСТ

РЕФЕРАТ	4
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ	6
ВСТУП	7
1.2 Технології голосового керування та обробки мовлення (ASR, NLP)	9
1.3 Огляд існуючих систем голосового керування	10
1.3.1 Хмарні системи голосових асистентів.....	12
1.3.2 Локальні системи голосового керування.....	14
1.3.3 Гібридні підходи до реалізації.....	16
1.4 Аналіз переваг та недоліків існуючих рішень	18
1.5 Постановка задачі дослідження	21
РОЗДІЛ 2. ВАРІАНТИ РЕАЛІЗАЦІЇ СИСТЕМИ.....	22
2.1 Підходи до побудови систем голосового керування	22
2.2 Порівняння хмарних та локальних рішень	24
2.3 Вибір оптимального варіанту реалізації системи.....	27
2.4 Формування функціональних вимог до системи.....	29
РОЗДІЛ 3. ОГЛЯД ТА ВИБІР КОМПОНЕНТІВ	31
3.1 Огляд апаратних платформ.....	31
3.1.1 Одноплатні комп'ютери (Raspberry Pi).....	33
Таблиця 3.2 – Основні характеристики Raspberry Pi 4 Model B	34
3.1.2 Мікроконтролери (Arduino).....	36
3.2 Огляд периферійних пристроїв	39
3.2.1 Мікрофонні модулі та пристрої введення звуку	40
3.2.2 Виконавчі пристрої (сервоприводи, реле).....	42
3.3 Вибір програмного забезпечення.....	44
3.3.1 Операційні системи (OS)	46
3.3.2 Мови програмування та бібліотеки	49
3.4 Аналіз технологій голосового керування	50
3.4.1 Технології розпізнавання мовлення (STT, ASR).....	53
3.4.2 Технології синтезу мовлення (TTS).....	55
3.5 Формування загальної архітектури системи.....	56
РОЗДІЛ 4. РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ	58
4.1 Проєктування архітектури системи голосового керування	58
4.2 Налаштування програмного середовища	60
4.3 Підключення та конфігурація апаратних компонентів (GPIO, PWM).....	62
4.4 Розробка алгоритму обробки голосових команд.....	64
4.5 Реалізація програмного забезпечення системи.....	66
РОЗДІЛ 5. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	68
5.1 Методика тестування системи.....	68
5.2 Перевірка функціональності системи.....	69
5.3 Аналіз точності розпізнавання голосових команд	71
5.4 Оцінка продуктивності та стабільності роботи системи	73
5.5 Рекомендації щодо вдосконалення системи.....	75
ВИСНОВКИ.....	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

API (Application Programming Interface) — програмний інтерфейс взаємодії між компонентами системи.

GPIO (General Purpose Input/Output) — універсальні програмовані входи/виходи для підключення електронних компонентів.

IoT (Internet of Things) — концепція мережі фізичних пристроїв, підключених до Інтернету та здатних обмінюватися даними.

NLP (Natural Language Processing) — обробка природної мови, що використовується для аналізу та інтерпретації голосових команд.

STT (Speech-to-Text) — технологія перетворення мовлення у текст.

TTS (Text-to-Speech) — технологія синтезу мовлення з тексту.

OS (Operating System) — операційна система, що забезпечує роботу пристрою.

CPU (Central Processing Unit) — центральний процесор обчислювальної системи.

RAM (Random Access Memory) — оперативна пам'ять пристрою.

USB (Universal Serial Bus) — інтерфейс підключення периферійних пристроїв.

PWM (Pulse Width Modulation) — широтно-імпульсна модуляція, використовується для керування сервоприводами.

GUI (Graphical User Interface) — графічний інтерфейс користувача.

CLI (Command Line Interface) — інтерфейс командного рядка.

Wi-Fi (Wireless Fidelity) — технологія бездротової передачі даних.

JSON (JavaScript Object Notation) — формат обміну даними між компонентами системи.

ВСТУП

Розвиток інформаційних технологій на концепції Інтернету речей (IoT) привів до актуалізації систем розумного будинку. Завдяки цьому полегшується життя людей: автоматизується керування побутовими пристроями, економиться час і сили. Одним із найбільш перспективних напрямів розвитку таких систем є впровадження інтуїтивних інтерфейсів взаємодії, серед яких важливе місце займає голосове керування.

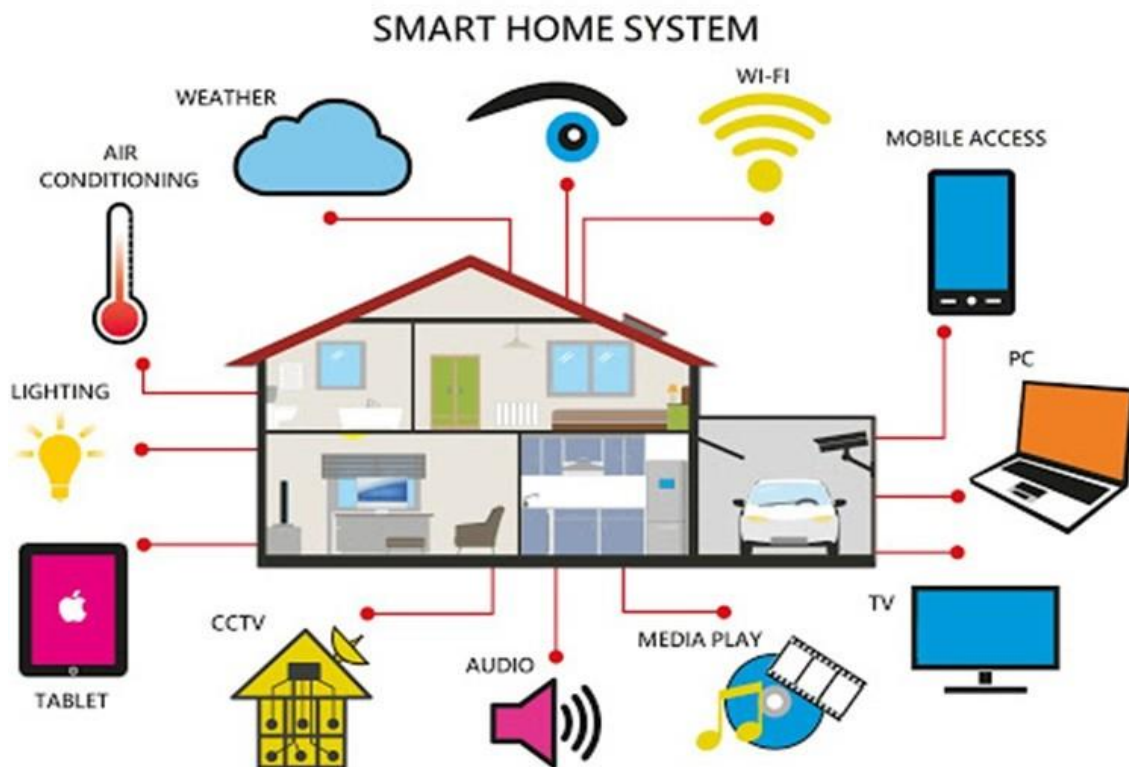


Рисунок 1 - Загальна архітектура системи «розумний будинок»

Завдяки голосовому керуванню не потрібно натискати кнопки чи користуватися пультом — достатньо просто сказати команду. Це робить техніку набагато зручнішою та доступнішою для всіх. Більше того, сучасні технології дозволяють навчити розпізнавати мову навіть невеликі та недорогі пристрої, як-от одноплатні комп'ютери.



Рисунок 2 - Приклад одноплатного комп'ютера



Рисунок 3 - Порівняльна схема хмарного та локального голосового керування

Такі відомі помічники, як Google Assistant, Alexa чи Siri, звісно, зручні, але вони потребують постійного інтернету, погано піддаються налаштуванню і можуть наражати на ризик особисті дані. Саме тому все більшої популярності набувають локальні голосові системи, які працюють повністю автономно, без хмар.

Рисунок 3 - Порівняльна схема хмарного та локального голосового керування

Актуальність теми — у швидкому розвитку технологій автоматизації та зростанні потреби в простих і зручних способах керування електронікою.

«Розумні будинки» дедалі частіше використовують у квартирах, приватних будинках та офісах. Вони автоматизують освітлення, температуру, безпеку й інші системи. Один із найперспективніших напрямків — голосове керування. Воно дає змогу взаємодіяти з технікою звичайною мовою, без додаткових пультів або сенсорних екранів.

Сучасні голосові системи працюють завдяки технологіям автоматичного розпізнавання мовлення (ASR) та обробки природної мови (NLP). Завдяки розвитку машинного навчання та зростанню потужності компактних пристроїв, голосове керування можна реалізувати навіть на одноплатних комп'ютерах (наприклад, Raspberry Pi). Це дозволяє створювати автономні локальні системи, яким не потрібне постійне підключення до Інтернету.

1.2 Технології голосового керування та обробки мовлення (ASR, NLP)

У сучасних «розумних будинках» дедалі популярнішим стає голосове керування. Воно дозволяє зручно взаємодіяти з технікою природною мовою – без кнопок, сенсорних панелей або додатків. Основа таких систем – технології автоматичного розпізнавання мовлення (ASR) та обробки природної мови (NLP).

ASR перетворює мову людини на текст. Процес включає отримання звуку з мікрофона, очищення від шумів, нормалізацію гучності й розпізнавання фонем, слів та речень. Раніше для цього використовували приховані марковські моделі (HMM), а тепер – глибокі нейронні мережі (DNN), рекурентні мережі (RNN) та трансформери. Завдяки цьому точність розпізнавання зросла навіть за наявності шуму чи різних акцентів. Важлива перевага сучасних ASR – робота в реальному часі. Наприклад, після команди «увімкнути світло» система має спрацювати майже миттєво. Для цього використовують оптимізовані алгоритми та компактні моделі розпізнавання.

NLP Після перетворення голосу на текст вступає в дію NLP – вона аналізує зміст команди й визначає наміри користувача. Система виділяє ключові слова: дію («відкрити») та об’єкт («двері»), після чого формує сигнал для виконавчого пристрою (сервоприводу, реле). Сучасні NLP-системи враховують контекст, аналізують структуру речень і працюють із різними мовами. У локальних системах часто використовують спрощені алгоритми, які потребують менше ресурсів і не потребують Інтернету.

Синтез мовлення (TTS) озвучує відповіді системи: підтверджує виконання команд, повідомляє про помилку чи стан пристроїв (наприклад: «Світло увімкнено»). Це створює зручний зворотний зв’язок.

Хмарні сервіси (Google Assistant, Amazon Alexa, Siri) забезпечують високу точність завдяки потужним серверам, але потребують постійного доступу до Інтернету й можуть нести ризики для конфіденційності.

Локальні системи (на базі Vosk, PocketSphinx) працюють без передачі даних на зовнішні сервери, навіть без Інтернету. Вони більш приватні й перспективні для «розумного будинку».

1.3 Огляд існуючих систем голосового керування

Голосові системи — один із головних напрямків розвитку «розумних будинків». Вони дають змогу керувати електронікою за допомогою голосу. Завдяки штучному інтелекту, ASR та NLP сучасні системи не просто виконують команди, а й підтримують діалог із користувачем.

Найпопулярніші серед них — Google Assistant, Amazon Alexa та Apple

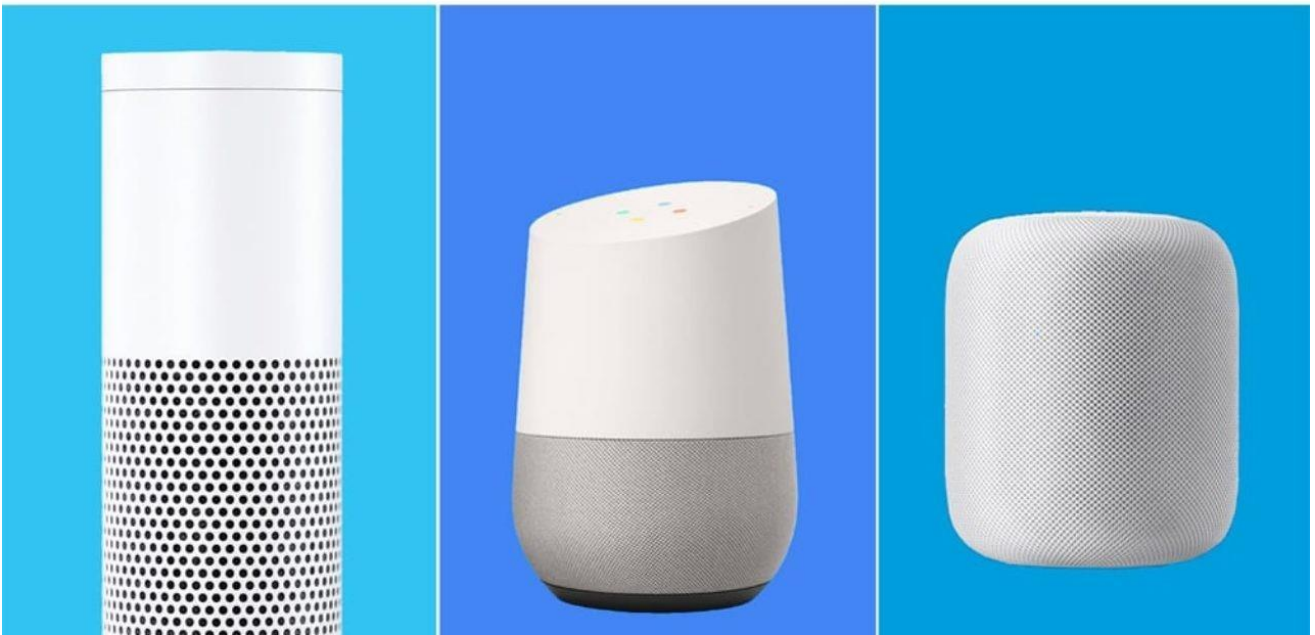


Рисунок 1.1 — Найпопулярніші віртуальні асистенти, що використовуються для керування пристроями

Siri. Вони інтегруються з багатьма пристроями: світло, обігрів, безпека, мультимедіа — все це можна контролювати централізовано.

Google Assistant працює на Google Nest, Android тощо. Переваги: чудове розуміння природної мови, інтеграція з сервісами Google, підтримка контекстних команд та сценаріїв. Останні версії використовують моделі Gemini.

Amazon Alexa працює через пристрої Echo. Її особливість — навички (Skills), які дозволяють легко розширювати функції. Підходить для керування світлом, технікою, безпекою та складними сценаріями.

Apple Siri вбудована в екосистему HomeKit. Головні плюси: високий рівень захисту даних і тісна інтеграція з продуктами Apple (HomePod, iPhone тощо). Недолік — менше сумісних пристроїв, ніж у Alexa або Google.

Локальні системи — окремий перспективний напрям. Наприклад, **Home Assistant** (з відкритим кодом) дозволяє працювати без Інтернету, без передачі даних у хмару. Плюси: конфіденційність, автономність, гнучкість.

Для локального розпізнавання мовлення часто використовують **Vosk**, **PocketSphinx** або **Whisper**. Вони здатні працювати на одноплатних комп'ютерах (наприклад, Raspberry Pi) без постійного доступу до мережі.

Усі сучасні голосові системи поєднують ASR (розпізнавання), NLP (розуміння) та TTS (синтез мовлення). Це дозволяє не тільки віддавати команди, а й отримувати голосові відповіді, наприклад: «Світло увімкнено».

Недоліки хмарних систем (Google, Amazon, Apple): залежність від Інтернету, можливі затримки, ризики витоку даних, інколи — обмеження за мовами або платна підписка.

1.3.1 Хмарні системи голосових асистентів

Хмарні системи голосових асистентів — це найпопулярніший тип голосового керування на сьогодні. Головна їхня особливість: команди не обробляються на самому пристрої (наприклад, на колонці чи телефоні), а передаються через Інтернет на потужні віддалені сервери компанії-розробника. Там відбувається аналіз голосу, розуміння команди та формування відповіді. Завдяки цьому точність розпізнавання дуже висока, і можна використовувати складний штучний інтелект.

Як це працює. Ваш голос через мікрофон потрапляє в систему, потім — через Інтернет на сервер. Там технології ASR (розпізнавання мовлення) перетворюють голос на текст, а NLP (обробка природної мови) аналізує, що саме ви хотіли. Після цього сервер надсилає відповідь назад або подає команду на розумний пристрій (увімкнути світло, відкрити штори тощо).

Найвідоміші приклади — Google Assistant, Amazon Alexa та Apple Siri. Всі вони вміють керувати розумним будинком: світлом, обігрівом, музикою, камерами безпеки тощо.

Google Assistant працює на Android, пристроях Google Nest та багатьох інших. Його головна перевага — дуже добре розуміє живу мову, підтримує контекстні діалоги (можна ставити уточнюючі запитання) та вміє шукати інформацію в Інтернеті. Також він керує складними сценаріями розумного будинку.

Amazon Alexa працює через пристрої Echo. Її особливість — «навички» (Skills), тобто додаткові програми від сторонніх розробників, які легко розширюють функції асистента. Alexa популярна завдяки тому, що з нею сумісно дуже багато пристроїв різних виробників.

Apple Siri вбудована в iPhone, iPad, HomePod та інші продукти Apple. Вона тісно інтегрована з екосистемою HomeKit. Головна перевага — високий рівень безпеки та захисту особистих даних. Недолік — менше сумісних пристроїв, ніж у Google або Amazon.

Основні плюси хмарних систем:

- Висока точність розпізнавання, бо сервери дуже потужні й використовують великі мовні моделі.
- Можливість швидко оновлювати функції без змін у локальному пристрої.
- Підтримка багатьох мов і здатність навчатися під конкретного користувача.
- Централізоване керування: одна команда здатна запустити цілий сценарій (наприклад, «Вечірній режим» вимикає світло, закриває штори й вмикає фільм).

Головні недоліки:

- Повна залежність від Інтернету. Якщо з'єднання зникло або сервер тимчасово не працює, асистент стає марним. Це проблема для розумного будинку, де важлива надійність.

- **Конфіденційність:** всі ваші голосові команди передаються на зовнішні сервери. Існує ризик витоку даних або несанкціонованого доступу. Через це багато людей придивляються до локальних систем без передачі даних у хмару.
- **Обмежена гнучкість:** ви не можете вільно змінювати логіку роботи асистента — тільки те, що дозволяє розробник. Деякі функції можуть бути платними або вимагати конкретного обладнання.

Отже, хмарні голосові асистенти — це дуже зручне й потужне рішення для розумного будинку. Вони забезпечують чудову якість розпізнавання та широкі можливості. Але їхня головна вразливість — стабільний Інтернет та питання приватності. Це підштовхує багатьох до пошуку автономних локальних альтернатив.

1.3.2 Локальні системи голосового керування

Локальні системи голосового керування є альтернативою хмарним голосовим асистентам та набувають все більшої популярності у сфері розумного будинку. Їх головною особливістю є виконання обробки голосових команд безпосередньо на локальному пристрої без передачі даних на зовнішні сервери. Для цього використовуються одноплатні комп'ютери, мікроконтролери або локальні сервери.

Локальна система працює так: голосова команда користувача обробляється безпосередньо на самому пристрої або в межах домашньої мережі (без відправлення даних в Інтернет). Мікрофон записує голос, потім система за допомогою технології ASR (автоматичне розпізнавання мовлення) перетворює його на текст або зрозумілі команди. Далі програма аналізує цю команду й надсилає відповідний сигнал на пристрій, який має виконати дію (наприклад, увімкнути світло чи відкрити штори). приклад, увімкнути світло чи відкрити штори).



Рисунок 1.2 — Приклад використання локальної системи керування у вигляді домофонів

Основною перевагою локальних систем є конфіденційність даних. Оскільки голосові команди не передаються на віддалені сервери, значно зменшується ризик витоку особистої інформації користувача. Також важливою перевагою є автономність роботи — система може функціонувати навіть без підключення до Інтернету.

Для реалізації локальних систем часто використовуються Raspberry Pi 4 Model B або інші одноплатні комп'ютери. Дані пристрої мають достатню продуктивність для обробки голосових команд та керування периферійними пристроями через GPIO-інтерфейси. Завдяки компактності та невисокій вартості Raspberry Pi широко застосовується в системах домашньої автоматизації.

Для локального розпізнавання мовлення використовуються програмні рішення Vosk, PocketSphinx або Whisper. Дані системи дозволяють реалізовувати офлайн-розпізнавання мовлення без використання хмарних сервісів. Для програмної реалізації зазвичай застосовується мова Python та бібліотеки gpiozero, RPi.GPIO і SpeechRecognition.

Локальні системи добре справляються з простими командами, як-от: «увімкни світло», «відчини двері» або «вимкни вентилятор». Після того, як команду розпізнано, система подає сигнал на реле, сервопривод чи інший виконавчий пристрій.

Ще одна перевага — гнучкість у налаштуванні. Ви можете самі змінювати алгоритми, додавати нові команди або підключати додаткові пристрої. На відміну від хмарних платформ, тут немає жорстких обмежень від виробника.

Але є й мінуси:

- нижча точність розпізнавання, ніж у хмарних сервісах;
- потрібно більше часу та знань для налаштування програмного забезпечення;
- локальні системи краще працюють із простими окремими командами, а зі складними діалогами — гірше.

Отже, локальні системи голосового керування є перспективним напрямом розвитку розумного будинку. Вони забезпечують автономність, конфіденційність та можливість повного контролю над системою, що робить їх ефективним рішенням для реалізації власних систем автоматизації.

1.3.3 Гібридні підходи до реалізації.

Гібридні системи голосового керування поєднують у собі переваги локальних та хмарних рішень. Основна ідея такого підходу полягає у розподілі функцій між локальним пристроєм і хмарними сервісами залежно від складності задачі. Завдяки цьому система може забезпечувати як автономність роботи, так і високу точність розпізнавання мовлення.

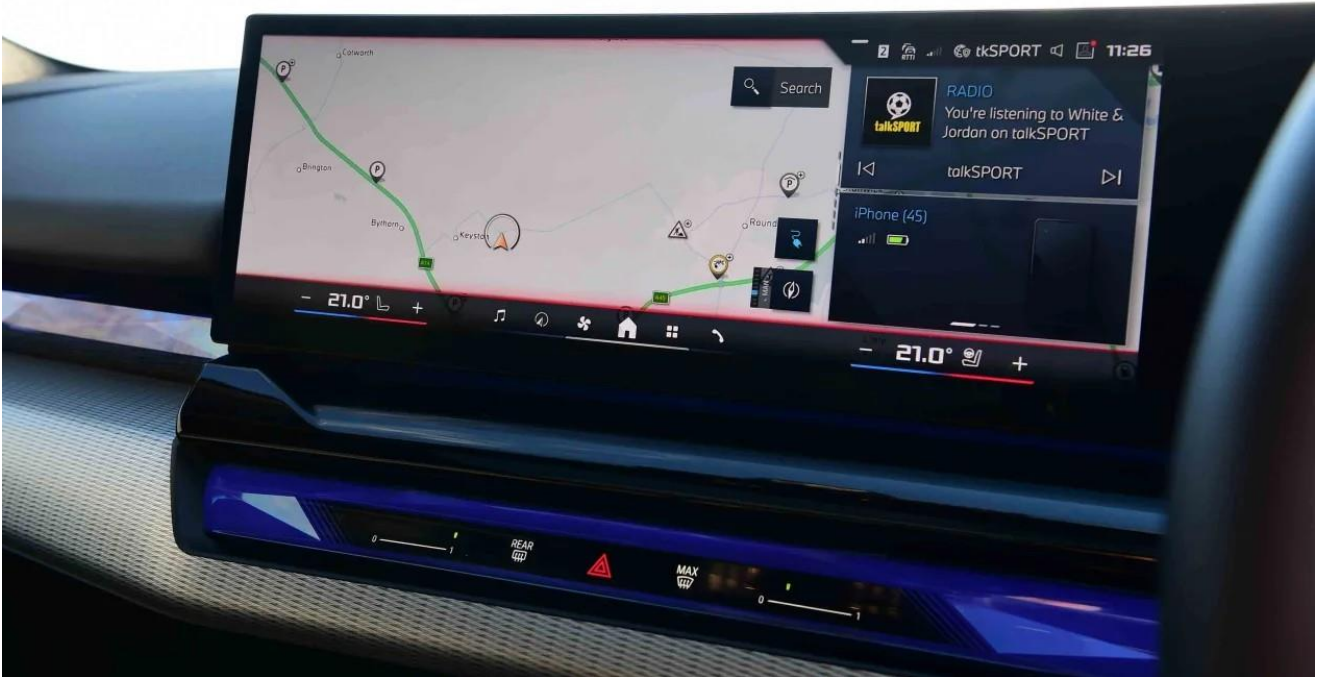


Рисунок 1.3 — Приклад використання гібридної системи в сучасних автомобілях

Принцип роботи гібридної системи полягає у тому, що прості голосові команди обробляються локально без підключення до Інтернету, а більш складні запити передаються на хмарні сервери. Наприклад, команди типу «увімкнути світло» або «відкрити двері» можуть виконуватись локально, тоді як складні інформаційні запити або діалогові сценарії обробляються за допомогою хмарних сервісів.

Однією з головних переваг гібридного підходу є підвищена надійність роботи системи. Навіть у разі втрати інтернет-з'єднання система зберігає можливість виконання базових функцій локально. Це особливо важливо для систем розумного будинку, де деякі елементи автоматизації повинні працювати безперервно.

Гібридні системи також дозволяють ефективніше використовувати ресурси локального пристрою. Наприклад, Raspberry Pi 4 Model B може виконувати базову обробку голосових команд та керування виконавчими пристроями, а складні обчислення передавати у хмару. Такий підхід дозволяє використовувати

компактне та недороге обладнання без значного зниження функціональності системи.

Ще однією перевагою є покращений рівень конфіденційності у порівнянні з повністю хмарними системами. Оскільки частина голосових команд обробляється локально, не всі дані передаються на зовнішні сервери. Це знижує ризик витоку персональної інформації користувача.

Для реалізації гібридних систем часто використовуються локальні модулі розпізнавання мовлення, наприклад Vosk або PocketSphinx, разом із хмарними сервісами Google Assistant чи Alexa. Локальна система забезпечує швидке виконання базових команд, а хмарні сервіси використовуються для складніших функцій та інтеграції з онлайн-сервісами.

Основним недоліком гібридного підходу є складність реалізації та налаштування. Розробнику необхідно забезпечити правильну взаємодію між локальними та хмарними компонентами системи. Крім того, навіть при частковій локальній обробці певна залежність від Інтернету все одно залишається.

Отже, гібридні системи голосового керування є перспективним напрямом розвитку розумного будинку, оскільки поєднують автономність локальних рішень із високою точністю та функціональністю сучасних хмарних технологій.

1.4 Аналіз переваг та недоліків існуючих рішень

Сучасні системи голосового керування для розумного будинку можна поділити на хмарні, локальні та гібридні рішення. Кожен із підходів має власні переваги та недоліки, які впливають на ефективність роботи, зручність використання та рівень безпеки системи.

Таблиця 1.1 - Переваги та недоліки популярних голосових асистентів.

Система	Переваги	Недоліки
Google Assistant	Висока точність розпізнавання, підтримка природної мови	Потребує Інтернету
Amazon Alexa	Велика кількість	Залежність від хмарних

	інтеграцій	сервісів
Apple Siri	Інтеграція з екосистемою Apple	Обмежене налаштування
Microsoft AI	Відкритий код, локальна робота	Нижча точність розпізнавання
Rhasspy	Автономність та локальна обробка	Складніше налаштування

Хмарні системи голосового керування, такі як Google Assistant, Alexa або Siri, є найбільш популярними завдяки високій точності розпізнавання мовлення та широким можливостям інтеграції. Вони використовують потужні сервери та сучасні алгоритми штучного інтелекту для обробки голосових команд. Основними перевагами таких систем є підтримка природної мови, можливість виконання складних команд та інтеграція з великою кількістю пристроїв розумного будинку.

Таблиця 1.2 - Порівняння хмарних та локальних систем голосового керування

Критерій	Хмарні системи	Локальні системи
Підключення до Інтернету	Обов'язкове	Необов'язкове
Швидкість роботи	Залежить від мережі	Висока при локальній обробці
Конфіденційність даних	Дані передаються на сервер	Дані залишаються локально
Точність розпізнавання	Висока	Середня або висока
Можливість автономної роботи	Обмежена	Повна
Гнучкість налаштування	Частково обмежена	Висока
Вартість використання	Можливі платні сервіси	Переважно безкоштовно
Залежність від сервісів	Висока	Мінімальна

Проте хмарні рішення мають і недоліки. Найважливішими з них є залежність від стабільного інтернет-з'єднання та ризики, пов'язані з

конфіденційністю даних. Оскільки голосові команди передаються на зовнішні сервери, існує ймовірність витоку персональної інформації користувача.

Локальні системи голосового керування працюють без передачі даних у хмару та виконують обробку команд безпосередньо на локальному пристрої. Основними перевагами таких рішень є автономність роботи, незалежність від Інтернету та підвищений рівень конфіденційності. Крім того, локальні системи дозволяють більш гнучко налаштовувати функціональність відповідно до потреб користувача.

Недоліками локальних систем є нижча точність розпізнавання мовлення у порівнянні з хмарними сервісами та складніше налаштування програмного забезпечення. Такі системи зазвичай краще працюють із простими голосовими командами та мають обмеження щодо складних діалогових сценаріїв.

Гібридні системи поєднують переваги локальних та хмарних рішень. В таких системах базові команди можуть оброблятися локально, а складніші запити — через хмарні сервіси. Це дозволяє забезпечити автономність роботи при виконанні простих функцій та одночасно використовувати можливості сучасних хмарних технологій.

Основною перевагою гібридних систем є баланс між продуктивністю, конфіденційністю та функціональністю. Проте реалізація таких систем є складнішою через необхідність організації взаємодії між локальними та хмарними компонентами.

Таким чином, кожен із підходів має власні особливості. Хмарні системи забезпечують високу точність і широкий функціонал, локальні — автономність та захист даних, а гібридні рішення дозволяють поєднати переваги обох підходів. Саме тому гібридні системи вважаються одним із найбільш перспективних напрямів розвитку голосового керування для розумного будинку.

1.5 Постановка задачі дослідження

У сучасних системах розумного будинку широко застосовуються технології автоматизації, які підвищують рівень комфорту, безпеку й ефективність управління побутовою технікою. Особливо перспективним способом керування є голосові команди: вони дають змогу взаємодіяти з системою без пультів, кнопок чи сенсорних екранів, що робить її значно зручнішою у щоденному використанні.

Таблиця 1.3 - Основні функції системи

Функція	Опис
Розпізнавання голосових команд	Перетворення мовлення у текстовий формат
Аналіз команд	Визначення необхідної дії системи
Керування виконавчими пристроями	Передача сигналів на реле або сервоприводи
Голосовий зворотний зв'язок	Повідомлення користувача про результат роботи
Автономна робота	Функціонування без постійного доступу до Інтернету

Незважаючи на значний розвиток голосових асистентів, більшість сучасних рішень орієнтована на використання хмарних сервісів. Такі системи забезпечують високу точність розпізнавання мовлення та підтримують складні алгоритми обробки природної мови, проте мають низку недоліків. Основними проблемами є залежність від стабільного інтернет-з'єднання, обмеження у налаштуванні функціональності та ризики, пов'язані з конфіденційністю персональних даних користувача.

У зв'язку з цим актуальним є створення систем голосового керування, здатних працювати локально або в автономному режимі без постійного використання хмарних сервісів. Реалізація такої системи дозволить забезпечити

базове голосове керування елементами розумного будинку, підвищити рівень безпеки даних та зменшити залежність від зовнішніх сервісів.

Метою даного дослідження є розробка системи голосового керування елементами розумного будинку з використанням технологій розпізнавання мовлення та локальної обробки голосових команд.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз сучасних систем голосового керування та технологій обробки мовлення.
- дослідити особливості хмарних, локальних та гібридних рішень;
- виконати вибір апаратних та програмних компонентів системи;
- розробити алгоритм обробки голосових команд;
- реалізувати програмне забезпечення для керування виконавчими пристроями;
- провести тестування працездатності системи та оцінити ефективність її роботи.

Об'єктом дослідження є системи голосового керування для розумного будинку. Предметом дослідження є методи та засоби реалізації локальної системи голосового керування з використанням технологій розпізнавання мовлення.

Практичне значення роботи полягає у можливості створення автономної системи голосового керування, яка може бути використана для автоматизації окремих елементів розумного будинку та подальшого розширення функціональності системи.

РОЗДІЛ 2. ВАРІАНТИ РЕАЛІЗАЦІЇ СИСТЕМИ

2.1 Підходи до побудови систем голосового керування

Системи голосового керування є важливою складовою сучасних технологій автоматизації та розумного будинку. Їх основне призначення полягає у забезпеченні взаємодії користувача з електронними пристроями за допомогою голосових команд. Для реалізації таких систем використовуються різні підходи, які відрізняються способом обробки голосових даних, архітектурою побудови та принципом взаємодії між компонентами системи.

Одним із основних підходів є використання хмарної архітектури. У таких системах обробка голосових команд виконується на віддалених серверах. Локальний пристрій лише записує голос користувача та передає аудіодані через Інтернет до хмарного сервісу. Після обробки команда повертається у вигляді результату або сигналу для керування пристроями. Даний підхід забезпечує високу точність розпізнавання мовлення та підтримку складних алгоритмів штучного інтелекту. Проте такі системи повністю залежать від стабільного інтернет-з'єднання та можуть створювати ризики для конфіденційності даних.

Іншим підходом є побудова локальних систем голосового керування. У цьому випадку всі процеси обробки голосових команд виконуються безпосередньо на локальному пристрої без передачі інформації на зовнішні сервери. Основною перевагою такого підходу є автономність роботи та підвищений рівень безпеки даних. Локальні системи можуть функціонувати навіть при відсутності Інтернету, що є важливим для систем розумного будинку.

Недоліком локального підходу є нижча точність розпізнавання мовлення у порівнянні з хмарними сервісами та обмежені обчислювальні ресурси локальних пристроїв.

Окремо варто виділити гібридний підхід до побудови систем голосового керування. Він поєднує переваги локальних та хмарних рішень. У таких системах базові команди можуть оброблятися локально, а складніші запити — за допомогою хмарних сервісів. Це дозволяє забезпечити стабільну роботу системи навіть при нестабільному інтернет-з'єднанні та одночасно використовувати

можливості сучасних технологій штучного інтелекту.

За способом взаємодії з користувачем системи голосового керування також можна поділити на системи з фіксованими командами та системи з підтримкою природної мови. Системи з фіксованими командами працюють за принципом розпізнавання заздалегідь визначених фраз, наприклад «увімкнути світло» або «відкрити двері». Такі рішення простіші у реалізації та потребують менше обчислювальних ресурсів.

Системи з підтримкою природної мови використовують технології NLP (Natural Language Processing), які дозволяють аналізувати зміст речення та розуміти різні варіанти формулювання команди. Наприклад, команди «увімкни світло», «включи освітлення» або «зроби світло» можуть інтерпретуватись однаково. Подібні системи забезпечують більш природну взаємодію з користувачем, проте є складнішими у реалізації.

Також системи голосового керування можуть відрізнятися способом активації. У деяких рішеннях використовується ключова фраза активації, наприклад «Hey Siri» або «OK Google». Після виявлення ключової фрази система переходить в режим очікування команди. Інші системи можуть працювати у режимі постійного прослуховування або активуватись за допомогою окремої кнопки.

Таким чином, існує декілька основних підходів до побудови систем голосового керування, кожен із яких має власні переваги та особливості. Вибір конкретного підходу залежить від вимог до системи, необхідного рівня автономності, продуктивності, конфіденційності та складності реалізації.

2.2 Порівняння хмарних та локальних рішень

Системи голосового керування можуть реалізовуватись за двома основними підходами — хмарним та локальним. Кожен із них має власні особливості, переваги та недоліки, які впливають на продуктивність, зручність використання, безпеку та автономність роботи системи. Вибір підходу залежить від вимог до системи та умов її використання.

Таблиця 2.1 - Порівняння підходів до побудови систем голосового керування

Підхід	Принцип роботи	Переваги	Недоліки
Хмарний	Обробка голосових команд виконується на віддалених серверах	Висока точність розпізнавання, підтримка ШІ та NLP	Залежність від Інтернету, ризики конфіденційності
Локальний	Обробка команд відбувається безпосередньо на пристрої	Автономність роботи, безпека даних	Обмежені ресурси, нижча точність розпізнавання
Гібридний	Частина команд обробляється локально, частина — у хмарі	Поєднання автономності та високої точності	Складність налаштування та реалізації

Хмарні системи голосового керування базуються на використанні віддалених серверів для обробки голосових команд. У таких рішеннях аудіосигнал передається через Інтернет на сервери компанії-розробника, де виконується розпізнавання мовлення та аналіз природної мови. Після цього результат повертається користувачу або використовується для керування пристроями розумного будинку.

Основною перевагою хмарних рішень є висока точність розпізнавання мовлення. Це забезпечується використанням потужних серверів та сучасних алгоритмів штучного інтелекту. Хмарні системи добре працюють із природною мовою, підтримують складні голосові запити та можуть постійно вдосконалюватися завдяки оновленням програмного забезпечення.

Також хмарні сервіси мають широку інтеграцію з різними онлайн-платформами та пристроями розумного будинку. Користувач може створювати складні сценарії автоматизації та використовувати додаткові можливості, пов'язані з доступом до Інтернету.

Проте хмарні рішення мають суттєві недоліки. Найбільшою проблемою є залежність від стабільного інтернет-з'єднання. У разі втрати доступу до мережі система може частково або повністю втратити працездатність. Крім того, передача голосових даних на зовнішні сервери створює ризики для конфіденційності та безпеки інформації користувача.

Локальні системи голосового керування працюють без передачі даних у хмару та виконують обробку команд безпосередньо на локальному пристрої. Основною перевагою такого підходу є автономність роботи. Система може функціонувати навіть без доступу до Інтернету, що особливо важливо для систем автоматизації розумного будинку.



Рисунок 2.1 - Приклад локальної системи голосового керування на базі Raspberry Pi

Ще однією важливою перевагою локальних рішень є підвищений рівень конфіденційності. Голосові команди не передаються на зовнішні сервери, тому ризик витоку персональних даних значно зменшується. Також локальні системи дозволяють більш гнучко налаштовувати функціональність та адаптувати її під конкретні потреби користувача.

Недоліком локальних рішень є нижча точність розпізнавання мовлення у порівнянні з хмарними сервісами. Це пояснюється обмеженими обчислювальними ресурсами локальних пристроїв та використанням менш

складних моделей штучного інтелекту. Крім того, локальні системи можуть вимагати складнішого налаштування та технічних знань для їх реалізації.

Таблиця 2.2 – Порівняння систем з фіксованими командами та підтримкою природної мови

Характеристика	Системи з фіксованими командами	Системи з підтримкою природної мови
Принцип роботи	Розпізнавання заданих фраз	Аналіз змісту речення
Складність реалізації	Низька	Висока
Потреба в ресурсах	Мінімальна	Значна
Гнучкість взаємодії	Обмежена	Висока
Приклад команд	«Увімкнути світло»	«Зроби світло в кімнаті»

Порівнюючи обидва підходи, можна зробити висновок, що хмарні системи краще підходять для реалізації складних голосових асистентів із підтримкою природної мови та великої кількості онлайн-функцій. Локальні рішення є більш доцільними у випадках, коли важливими є автономність роботи, захист персональних даних та незалежність від зовнішніх сервісів.

Таким чином, вибір між хмарними та локальними системами залежить від поставлених задач та вимог до системи голосового керування. Для систем розумного будинку особливо важливими є стабільність роботи, конфіденційність та можливість локального керування, тому локальні або гібридні підходи є перспективним напрямом реалізації подібних систем.

2.3 Вибір оптимального варіанту реалізації системи .

Після аналізу існуючих підходів до побудови систем голосового керування було визначено, що для реалізації системи розумного будинку найбільш доцільним є використання локального або гібридного підходу. Такий вибір обумовлений необхідністю забезпечення автономності роботи, захисту

персональних даних користувача та можливості роботи системи без постійного підключення до мережі Інтернет.

Хмарні системи голосового керування забезпечують високу точність розпізнавання мовлення та підтримують складні алгоритми обробки природної мови. Проте їх використання має низку недоліків, серед яких залежність від стабільного інтернет-з'єднання, обмеження у налаштуванні функціональності та ризики, пов'язані з конфіденційністю даних. Для системи розумного будинку такі недоліки можуть негативно впливати на стабільність та безпеку роботи.

Локальні системи голосового керування дозволяють виконувати обробку команд без передачі даних на зовнішні сервери. Це забезпечує підвищений рівень безпеки та можливість автономної роботи системи. Крім того, локальний підхід дає можливість самостійно налаштовувати алгоритми роботи та адаптувати систему під конкретні потреби користувача.

З урахуванням особливостей дипломного проєкту оптимальним варіантом реалізації було обрано локальну систему голосового керування з можливістю подальшого розширення функціональності. Такий підхід дозволяє реалізувати базове голосове керування елементами розумного будинку без необхідності використання зовнішніх хмарних сервісів.

Для реалізації системи планується використання технологій автоматичного розпізнавання мовлення (ASR) та локальної обробки голосових команд. Голосова команда користувача буде прийматись через мікрофон, після чого система виконуватиме її аналіз та формуватиме керуючий сигнал для виконавчого пристрою.

У якості програмної платформи обрано мову програмування Python, оскільки вона має велику кількість бібліотек для роботи з голосовими командами, аудіо та керування периферійними пристроями. Для розпізнавання мовлення можуть використовуватись локальні модулі, наприклад Vosk або SpeechRecognition.

Обраний варіант реалізації дозволяє забезпечити:

- автономну роботу системи;
- зменшення залежності від Інтернету;
- підвищення рівня конфіденційності;
- можливість подальшого розширення функціональності;
- відносно невисоку вартість реалізації системи.

Таким чином, локальний підхід є найбільш доцільним для реалізації системи голосового керування елементами розумного будинку в межах даного дипломного проєкту, оскільки поєднує достатню функціональність, автономність роботи та простоту реалізації.

2.4 Формування функціональних вимог до системи

Для успішної реалізації системи голосового керування елементами розумного будинку необхідно сформувавши основні функціональні вимоги, які визначають принцип роботи системи, її можливості та особливості взаємодії з користувачем. Формування таких вимог дозволяє визначити необхідний функціонал системи та забезпечити її стабільну роботу.

Основною функцією системи є прийом та обробка голосових команд користувача. Система повинна забезпечувати розпізнавання голосу за допомогою мікрофона та перетворення мовлення у текстовий формат для подальшого аналізу. Після розпізнавання команда повинна оброблятися програмним забезпеченням, яке визначає необхідну дію та формує керуючий сигнал для виконавчого пристрою.

Система повинна підтримувати виконання базових команд керування елементами розумного будинку. До таких команд можуть належати:

- увімкнення або вимкнення освітлення;
- відкриття або закриття дверей;
- керування виконавчими механізмами;
- активація окремих режимів роботи системи.

Однією з важливих функціональних вимог є автономність роботи системи. Система повинна мати можливість функціонувати без постійного підключення до мережі Інтернет. Це дозволить забезпечити стабільну роботу навіть при відсутності доступу до зовнішніх сервісів.

Також система повинна забезпечувати достатній рівень швидкодії. Час між поданням голосової команди та виконанням відповідної дії має бути мінімальним для забезпечення комфортної взаємодії користувача із системою.

Таблиця 2.3 – Основні компоненти обраної системи

Компонент	Призначення
Raspberry Pi 4B	Центральний обчислювальний модуль системи
USB-мікрофон	Прийом голосових команд користувача
Серво-привід	Виконання керуючих дій
Python	Основна мова програмування системи
Vosk / SpeechRecognition (але можна і без нього)	Розпізнавання голосових команд
Raspberry Pi OS	Операційна система пристрою

Важливою вимогою є забезпечення коректного розпізнавання голосових команд. Система повинна підтримувати роботу з наперед визначеним набором команд та забезпечувати їх стабільне розпізнавання при різних умовах використання.

Для підвищення зручності використання система може підтримувати голосові повідомлення або текстові повідомлення про виконання команд. Це дозволяє забезпечити зворотний зв'язок між користувачем і системою.

Ще однією функціональною вимогою є можливість подальшого розширення системи. Архітектура системи повинна дозволити підключення нових виконавчих пристроїв, додавання нових голосових команд та розширення функціональності без значної зміни програмної частини.

Окрім функціональних вимог, важливими є вимоги до безпеки та надійності системи. Система повинна стабільно працювати при тривалому використанні, забезпечувати коректне виконання команд та мінімізувати ймовірність помилкових спрацювань.

Таким чином, сформовані функціональні вимоги визначають основні можливості системи голосового керування та створюють основу для подальшого проектування апаратної та програмної частини системи розумного будинку.

РОЗДІЛ 3. ОГЛЯД ТА ВИБІР КОМПОНЕНТІВ

3.1 Огляд апаратних платформ

Апаратна платформа є одним із ключових компонентів системи голосового керування розумним будинком, оскільки саме вона забезпечує виконання програмного забезпечення, обробку голосових команд та взаємодію з периферійними пристроями. Вибір апаратної платформи впливає на продуктивність системи, швидкодію, енергоспоживання, можливість підключення додаткових модулів та загальну вартість реалізації проєкту.

Для побудови систем голосового керування найчастіше використовуються одноплатні комп'ютери та мікроконтролери. Дані пристрої дозволяють реалізовувати як прості системи автоматизації, так і більш складні рішення з підтримкою технологій розпізнавання мовлення та обробки природної мови.

Одноплатні комп'ютери являють собою компактні обчислювальні пристрої, які мають процесор, оперативну пам'ять, інтерфейси введення-виведення та підтримують повноцінні операційні системи. Такі платформи мають достатню продуктивність для запуску програм обробки голосу, роботи з аудіопристроями та керування периферійними модулями.

Найбільш популярною платформою у сфері навчальних та дослідницьких проєктів є Raspberry Pi 4 Model B. Даний одноплатний комп'ютер підтримує

операційну систему Linux, має GPIO-інтерфейси для підключення зовнішніх пристроїв та достатню обчислювальну потужність для реалізації систем голосового керування. Завдяки компактним розмірам, доступній вартості та великій кількості програмних бібліотек Raspberry Pi широко використовується у системах розумного будинку.

Також для реалізації систем автоматизації часто використовуються мікроконтролери, наприклад Arduino. На відміну від одноплатних комп'ютерів, мікроконтролери мають меншу продуктивність та зазвичай не підтримують повноцінні операційні системи. Їх основною перевагою є низьке енергоспоживання, простота використання та можливість роботи у реальному часі.

Мікроконтролери добре підходять для керування датчиками, реле, сервоприводами та іншими виконавчими механізмами. Проте для реалізації повноцінного голосового керування їх можливостей часто недостатньо, оскільки обробка мовлення потребує значних обчислювальних ресурсів.

При виборі апаратної платформи для системи голосового керування необхідно враховувати декілька основних факторів:

- продуктивність пристрою;
- можливість роботи з аудіо;
- підтримку операційної системи;
- наявність GPIO-інтерфейсів;
- можливість підключення периферійних пристроїв;
- енергоспоживання;
- вартість реалізації системи.

Для систем розумного будинку важливою є також можливість подальшого розширення функціональності. Апаратна платформа повинна підтримувати підключення нових датчиків, виконавчих механізмів та додаткових модулів автоматизації.

Таким чином, вибір апаратної платформи є важливим етапом проєктування системи голосового керування. Одноплатні комп'ютери забезпечують достатню продуктивність для обробки голосових команд та реалізації складних функцій автоматизації, тоді як мікроконтролери є ефективним рішенням для керування периферійними пристроями та виконання простих задач автоматизації.

3.1.1 Одноплатні комп'ютери (Raspberry Pi)

Таблиця 3.1 – Порівняння апаратних платформ для систем голосового керування

Характеристика	Raspberry Pi 4 Model B	Arduino Uno	ESP32
Тип пристрою	Одноплатний комп'ютер	Мікроконтролер	Мікроконтролер
Операційна система	Linux	Відсутня	Відсутня
Продуктивність	Висока	Низька	Середня
Робота з аудіо	Підтримується	Обмежена	Частково
GPIO-інтерфейси	Так	Так	Так
Підключення до мережі	Wi-Fi, Ethernet, Bluetooth	Через модулі	Wi-Fi, Bluetooth
Підходить для ASR	Так	Ні	Частково
Вартість	Середня	Низька	Низька

Одноплатні комп'ютери є важливою складовою сучасних систем автоматизації та розумного будинку. Вони поєднують компактні розміри, достатню продуктивність та можливість роботи з периферійними пристроями, що робить їх ефективним рішенням для реалізації систем голосового керування.

Однією з найбільш популярних платформ у даній сфері є Raspberry Pi 4 Model B. Даний одноплатний комп'ютер широко використовується у навчальних, дослідницьких та практичних проєктах завдяки доступній вартості, великій кількості програмного забезпечення та підтримці різних периферійних модулів.

Raspberry Pi 4 Model B побудований на базі процесора ARM та підтримує багатоядерну обробку даних. Плата має оперативну пам'ять, USB-порти, мережеві інтерфейси Wi-Fi та Bluetooth, а також GPIO-контакти для підключення

зовнішніх пристроїв. Це дозволяє використовувати її для роботи з датчиками,

сервоприводами, реле, мікрофонами та іншими компонентами системи розумного будинку.



Рисунок 3.1 — Одноплатний комп'ютер Raspberry Pi 4 Model B

Однією з головних переваг Raspberry Pi є підтримка повноцінної операційної системи. На пристрої може використовуватись Raspberry Pi OS або інші Linux-системи, що дозволяє запускати програми для розпізнавання мовлення, обробки голосових команд та керування периферійними пристроями. Завдяки цьому Raspberry Pi здатний виконувати функції локального центру керування системою розумного будинку.

Таблиця 3.2 – Основні характеристики Raspberry Pi 4 Model B

Параметр	Значення
Процесор	Broadcom BCM2711 ARM Cortex-A72
Кількість ядер	4
Тактова частота	1.5 ГГц
Оперативна пам'ять	2–8 ГБ
Операційна система	Raspberry Pi OS (Linux)
Бездротові інтерфейси	Wi-Fi, Bluetooth

Параметр	Значення
GPIO	40 контактів
Живлення	5 В через USB-C

Для реалізації голосового керування Raspberry Pi підтримує підключення USB-мікрофонів та аудіопристроїв. Це дозволяє приймати голосові команди користувача та виконувати їх подальшу обробку за допомогою програмних бібліотек. Також плата може використовуватись для генерації голосових повідомлень та звукового зворотного зв'язку.

Важливою особливістю Raspberry Pi є наявність GPIO-інтерфейсів (General Purpose Input/Output), які дозволяють взаємодіяти з виконавчими механізмами. За допомогою GPIO можна керувати сервоприводами, реле, світлодіодами та іншими елементами автоматизації. Це робить платформу зручною для створення систем керування елементами розумного будинку. Ще однією перевагою Raspberry Pi є велика спільнота користувачів та наявність великої кількості документації. Існує значна кількість готових бібліотек, прикладів програм та інструкцій, що спрощує процес розробки систем автоматизації.

Проте Raspberry Pi має і певні недоліки. У порівнянні з персональними комп'ютерами одноплатний комп'ютер має обмежені обчислювальні ресурси. Через це при використанні складних моделей розпізнавання мовлення можуть виникати затримки або підвищене навантаження на систему. Також для стабільної роботи плати необхідне якісне живлення та правильне охолодження.

Незважаючи на певні обмеження, Raspberry Pi є ефективною платформою для реалізації систем голосового керування. Поєднання компактності, функціональності, підтримки операційних систем та можливості роботи з периферійними пристроями робить даний одноплатний комп'ютер оптимальним рішенням для побудови систем розумного будинку.

3.1.2 Мікроконтролери (Arduino)

Мікроконтролери є важливим елементом сучасних систем автоматизації та широко використовуються у проєктах розумного будинку. На відміну від одноплатних комп'ютерів, мікроконтролери орієнтовані переважно на виконання конкретних задач керування та роботу з периферійними пристроями у режимі реального часу.

Однією з найпоширеніших платформ у даній сфері є Arduino. Платформи Arduino набули популярності завдяки простоті використання, доступній вартості та великій кількості готових модулів і бібліотек. Вони активно застосовуються у навчальних, дослідницьких та побутових проєктах автоматизації.



Рисунок 3.2 — Одноплатний комп'ютер Arduino Uno

Arduino являє собою мікроконтролерну платформу, яка дозволяє взаємодіяти з різними датчиками, реле, сервоприводами, світлодіодами та іншими електронними компонентами. Програмування виконується за допомогою середовища Arduino IDE та мови програмування, заснованої на C/C++.

Однією з головних переваг Arduino є низьке енергоспоживання та простота роботи з цифровими й аналоговими входами та виходами. Це робить платформу зручною для реалізації систем керування освітленням, датчиками руху, температурними сенсорами та іншими елементами розумного будинку.

Також Arduino підтримує роботу з PWM-сигналами (Pulse Width Modulation), що дозволяє керувати сервоприводами та регулювати роботу різних виконавчих механізмів. Завдяки цьому мікроконтролери можуть використовуватись для реалізації простих систем автоматичного керування.

Ще однією перевагою Arduino є велика кількість готових модулів та додаткових плат розширення. Існують модулі для роботи з Wi-Fi, Bluetooth, RFID, дисплеями, мікрофонами та іншими пристроями, що дозволяє легко розширювати функціональність системи.

Проте мікроконтролери мають і певні обмеження. Основним недоліком Arduino є невисока обчислювальна потужність. У порівнянні з одноплатними комп'ютерами мікроконтролери не підтримують повноцінні операційні системи та мають обмежений обсяг пам'яті. Через це їх можливостей недостатньо для реалізації складних алгоритмів розпізнавання мовлення або обробки природної мови.

Для систем голосового керування Arduino зазвичай використовується як допоміжний модуль керування виконавчими пристроями, тоді як основна обробка голосових команд виконується на більш продуктивних пристроях. У деяких випадках можуть використовуватись спеціальні модулі розпізнавання голосу, проте їх функціональність зазвичай обмежується набором простих команд.

Таким чином, мікроконтролери Arduino є ефективним рішенням для реалізації задач керування периферійними пристроями та елементами автоматизації. Простота використання, низьке енергоспоживання та велика кількість модулів роблять Arduino популярною платформою у системах розумного будинку, хоча для складної обробки голосових команд можливостей мікроконтролерів часто недостатньо.

Arduino Uno є однією з найпопулярніших плат платформи Arduino. Вона побудована на базі мікроконтролера ATmega328P та широко використовується у навчальних і практичних проєктах. Плата має 14 цифрових входів-виходів, з яких 6 підтримують PWM-сигнали, а також 6 аналогових входів.

Основною перевагою Arduino Uno є простота використання та велика кількість навчальних матеріалів. Плата підтримує підключення різних датчиків, сервоприводів, реле та інших електронних компонентів. Arduino Uno часто використовується для створення простих систем автоматизації, робототехнічних проєктів та елементів розумного будинку.

Плата підключається до комп'ютера через USB та програмується у середовищі Arduino IDE. Завдяки доступній вартості та стабільній роботі Arduino Uno є хорошим варіантом для початківців.

Arduino Nano є компактною версією Arduino Uno. Вона також побудована на базі мікроконтролера ATmega328P та має подібні технічні характеристики, проте відрізняється значно меншими розмірами.

Arduino Nano має 14 цифрових входів-виходів, 8 аналогових входів та підтримує PWM-сигнали. Завдяки компактності плата добре підходить для вбудованих систем, переносних пристроїв та проєктів із обмеженим простором.

Плата часто використовується у невеликих системах автоматизації, пристроях керування та IoT-проєктах. Arduino Nano можна встановлювати безпосередньо на макетну плату, що спрощує процес прототипування електронних схем.

Основними перевагами Arduino Nano є компактність, низьке енергоспоживання та зручність використання у малогабаритних пристроях.

Arduino Mega 2560 є більш потужною платою у порівнянні з Arduino Uno та Nano. Вона побудована на базі мікроконтролера ATmega2560 та має значно більшу кількість входів-виходів і пам'яті.

Arduino Mega має 54 цифрові входи-виходи, з яких 15 підтримують PWM-сигнали, а також 16 аналогових входів. Завдяки цьому плата дозволяє підключати велику кількість датчиків, модулів та виконавчих механізмів одночасно.

Плата використовується у складних системах автоматизації, робототехнічних комплексах та проєктах розумного будинку з великою кількістю

компонентів. Arduino Mega підходить для реалізації систем, де необхідно одночасно обробляти багато сигналів та керувати великою кількістю пристроїв.

Основними перевагами Arduino Mega є висока функціональність, велика кількість портів та більший обсяг пам'яті. Недоліком плати є більший розмір та вища вартість у порівнянні з Arduino Uno та Nano.

3.2 Огляд периферійних пристроїв

Периферійні пристрої є важливою складовою систем голосового керування розумним будинком, оскільки саме вони забезпечують взаємодію між користувачем та системою автоматизації. До периферійних пристроїв належать мікрофони, датчики, виконавчі механізми, реле, сервоприводи, аудіомодулі та інші компоненти, які використовуються для введення, обробки та виконання команд.

Для реалізації голосового керування одним із основних пристроїв є мікрофон. Він використовується для прийому голосових команд користувача та передачі аудіосигналу до системи обробки мовлення. У системах розумного будинку можуть використовуватись USB-мікрофони або спеціальні мікрофонні модулі. Важливими характеристиками мікрофона є чутливість, якість запису звуку та рівень шумозаглушення.

Окрім пристроїв введення звуку, важливу роль у системах автоматизації відіграють виконавчі пристрої. Вони забезпечують виконання команд, отриманих після обробки голосового сигналу. До таких пристроїв належать реле, сервоприводи, електродвигуни, світлодіоди та інші механізми керування.

Одним із найбільш поширених виконавчих пристроїв у навчальних та дослідницьких проєктах є сервопривід. Сервоприводи дозволяють виконувати точне позиціонування механізмів та широко використовуються для автоматичного відкриття дверей, керування шторами або іншими елементами розумного будинку. Керування сервоприводом здійснюється за допомогою PWM-сигналів.

Для комутації електричних пристроїв часто використовуються релейні модулі. Вони дозволяють вмикати або вимикати освітлення, вентилятори, побутову техніку та інші електроприлади. Реле забезпечують електричну ізоляцію між низьковольтною системою керування та силовими електричними колами.

У деяких системах можуть використовуватись датчики руху, температури, освітленості або вологості. Такі пристрої дозволяють автоматизувати роботу системи та реалізовувати додаткові сценарії керування. Наприклад, освітлення може автоматично вмикатись при виявленні руху або зміні рівня освітленості приміщення.

Для забезпечення зворотного зв'язку з користувачем можуть використовуватись динаміки або аудіомодулі. Вони дозволяють системі озвучувати повідомлення про виконання команд або поточний стан системи.

При виборі периферійних пристроїв необхідно враховувати сумісність із апаратною платформою, тип підключення, енергоспоживання, точність роботи та вартість компонентів. Також важливо забезпечити стабільність роботи пристроїв та можливість їх подальшого розширення у складі системи розумного будинку.

Таким чином, периферійні пристрої є невід'ємною частиною систем голосового керування та забезпечують реалізацію функцій введення, обробки та виконання команд. Від правильного вибору компонентів залежить ефективність, стабільність та функціональність усієї системи автоматизації.

3.2.1 Мікрофонні модулі та пристрої введення звуку

Мікрофонні модулі та пристрої введення звуку є важливими компонентами систем голосового керування, оскільки саме вони забезпечують прийом голосових команд користувача та передачу аудіосигналу до системи обробки мовлення. Від якості мікрофона залежить точність розпізнавання голосу, стабільність роботи системи та зручність взаємодії користувача з елементами розумного будинку.

Основним завданням мікрофона є перетворення звукових хвиль у електричний сигнал, який надалі обробляється програмним забезпеченням системи. Для систем голосового керування важливими характеристиками мікрофонів є чутливість, рівень шумозаглушення, якість запису звуку та підтримка роботи у режимі реального часу.

У системах автоматизації можуть використовуватись як окремі USB-мікрофони, так і спеціалізовані мікрофонні модулі. USB-мікрофони є простим та зручним рішенням, оскільки легко підключаються до одноплатних комп'ютерів і не потребують складного налаштування. Вони забезпечують достатню якість звуку для реалізації базового голосового керування.

Також поширеними є мікрофонні модулі на базі електретних мікрофонів. Такі модулі часто використовуються у системах на базі Arduino або інших мікроконтролерів. Вони мають компактні розміри та низьке енергоспоживання, проте зазвичай забезпечують нижчу якість запису звуку у порівнянні з USB-мікрофонами.

Для покращення якості розпізнавання мовлення у сучасних системах можуть використовуватись цифрові MEMS-мікрофони. Вони забезпечують кращу стійкість до шумів, компактність та високу точність передачі аудіосигналу. Такі мікрофони часто використовуються у голосових асистентах та системах розумного будинку.

При виборі пристрою введення звуку важливо враховувати умови експлуатації системи. У приміщеннях із високим рівнем шуму бажано використовувати мікрофони з підтримкою шумозаглушення або спрямовані мікрофони, які краще сприймають голос користувача та зменшують вплив сторонніх звуків.

Для коректної роботи системи голосового керування також важливою є частота дискретизації та якість цифрової обробки звуку. Чим вища якість аудіосигналу, тим точніше система зможе розпізнавати голосові команди.

У системах розумного будинку мікрофонні модулі можуть працювати у режимі постійного прослуховування або активуватись після вимовлення ключової фрази. Після отримання аудіосигналу система передає його до програмного модуля розпізнавання мовлення для подальшої обробки.

Таким чином, мікрофонні модулі та пристрої введення звуку є важливими елементами систем голосового керування. Від правильного вибору мікрофона залежить якість роботи системи, точність розпізнавання голосових команд та загальна ефективність взаємодії користувача з системою розумного будинку.

3.2.2 Виконавчі пристрої (сервоприводи, реле)

Виконавчі пристрої є важливою складовою системи голосового керування розумним будинком, оскільки саме вони забезпечують фізичне виконання команд користувача. Після обробки голосового сигналу система формує керуючий сигнал, який передається до виконавчого механізму. До найбільш поширених виконавчих пристроїв належать сервоприводи та релейні модулі.



Рисунок 3.3 — Сервопривід і реле

Сервоприводи використовуються для точного керування положенням механізмів. Вони дозволяють повертати вал на заданий кут та утримувати його у необхідному положенні. У системах розумного будинку сервоприводи можуть застосовуватись для автоматичного відкриття дверей, керування шторами, вікнами або іншими рухомими елементами.

Одним із найпоширеніших сервоприводів у навчальних та дослідницьких проєктах є SG90. Даний сервопривід має компактні розміри, низьке енергоспоживання та простий принцип керування. Керування сервоприводом здійснюється за допомогою PWM-сигналу (Pulse Width Modulation), який задає необхідний кут повороту.

Основними перевагами сервоприводів є точність позиціонування, простота підключення та можливість автоматичного керування механізмами. Проте сервоприводи мають і певні недоліки, серед яких обмежений кут повороту та невелика потужність у компактних моделях.

Іншим важливим виконавчим пристроєм є релейні модулі. Реле використовуються для керування електричними пристроями шляхом замикання або розмикання електричного кола. За допомогою реле система може вмикати або вимикати освітлення, вентилятори, побутову техніку та інші електроприлади.

Релейні модулі забезпечують електричну ізоляцію між низьковольтною системою керування та силовими електричними колами. Це дозволяє безпечно керувати пристроями, які працюють від мережі 220 В. Для систем автоматизації часто використовуються одноканальні або багатоканальні релейні модулі.

Перевагами реле є простота використання, можливість керування потужними електроприладами та висока надійність роботи. Недоліком механічних реле є обмежений ресурс роботи контактів та невелика затримка при перемиканні.

У системах голосового керування виконавчі пристрої отримують сигнали від програмного забезпечення після успішного розпізнавання голосової команди.

Наприклад, після команди «відкрити двері» система може подати PWM-сигнал на сервопривід, а команда «увімкнути світло» може активувати відповідне реле.

Таким чином, сервоприводи та релейні модулі є основними виконавчими елементами систем розумного будинку. Вони забезпечують фізичне виконання голосових команд та дозволяють реалізувати автоматичне керування різними пристроями та механізмами.

3.3 Вибір програмного забезпечення

Програмне забезпечення є однією з основних складових системи голосового керування розумним будинком, оскільки саме воно забезпечує обробку голосових команд, взаємодію між компонентами системи та керування виконавчими пристроями. Від правильного вибору програмного забезпечення залежить стабільність роботи системи, швидкість обробки команд та можливість подальшого розширення функціональності.

Таблиця 3.3 – Основні компоненти програмного забезпечення системи

Компонент	Призначення
Raspberry Pi OS (Linux)	Операційна система для роботи платформи
Python	Основна мова програмування
Vosk / SpeechRecognition	Розпізнавання голосових команд
gpiozero / RPi.GPIO	Керування GPIO-пристроями
pyttsx3	Синтез голосових повідомлень
ALSA / PulseAudio	Робота з аудіопристроями

Для реалізації системи голосового керування необхідно використовувати програмне забезпечення, яке підтримує роботу з аудіопристроями, технологіями розпізнавання мовлення, GPIO-інтерфейсами та виконавчими механізмами. Також важливою вимогою є підтримка локальної обробки голосових команд без обов'язкового використання хмарних сервісів.

Одним із основних компонентів програмного забезпечення є операційна система. Для систем автоматизації та одноплатних комп'ютерів найчастіше

використовуються Linux-системи завдяки їх стабільності, відкритому коду та підтримці великої кількості програмних бібліотек. Linux забезпечує зручну роботу з периферійними пристроями, аудіомодулями та мережевими сервісами.

Для програмної реалізації системи голосового керування доцільно використовувати мову програмування Python. Вона є однією з найпоширеніших мов у сфері автоматизації та штучного інтелекту. Python має простий синтаксис, велику кількість бібліотек та підтримує роботу з аудіо, GPIO та технологіями розпізнавання мовлення.

Для реалізації розпізнавання голосових команд можуть використовуватись бібліотеки SpeechRecognition, Vosk або PocketSphinx. Дані програмні модулі дозволяють виконувати обробку голосу та перетворення мовлення у текстовий формат. Для локальних систем особливо важливою є підтримка офлайн-режиму роботи.

Для керування виконавчими пристроями використовуються бібліотеки gpiozero або RPi.GPIO. Вони дозволяють працювати з GPIO-контактами, керувати реле, сервоприводами та іншими периферійними пристроями.

Таблиця 3.4 – Порівняння бібліотек для розпізнавання мовлення

Бібліотека	Підтримка офлайн-режиму	Переваги	Недоліки
Vosk	Так	Працює локально, підтримка Python	Потребує налаштування моделей
SpeechRecognition	Частково	Простота використання	Часто залежить від зовнішніх сервісів
PocketSphinx	Так	Невелике навантаження на систему	Нижча точність розпізнавання

У деяких випадках система може використовувати модулі синтезу мовлення

(TTS — Text-to-Speech), які дозволяють генерувати голосові повідомлення та

забезпечувати зворотний зв'язок із користувачем. Для цього можуть використовуватись бібліотеки `pyttsx3` або інші програмні рішення.

При виборі програмного забезпечення важливо враховувати:

- стабільність роботи;
- підтримку локальної обробки голосу;
- сумісність із апаратною платформою;
- простоту налаштування;
- можливість подальшого розширення функціональності.

Таким чином, вибір програмного забезпечення є важливим етапом реалізації системи голосового керування. Використання Linux, Python та спеціалізованих бібліотек для роботи з голосовими командами дозволяє створити стабільну, гнучку та функціональну систему автоматизації розумного будинку.

3.3.1 Операційні системи (OS)

Операційна система є важливою складовою системи голосового керування, оскільки вона забезпечує взаємодію між апаратним та програмним забезпеченням. Саме операційна система відповідає за роботу програм, доступ до периферійних пристроїв, обробку даних та керування ресурсами системи.

Для реалізації систем автоматизації та розумного будинку найчастіше використовуються операційні системи на базі Linux. Вони є популярними завдяки стабільності роботи, відкритому вихідному коду, високому рівню безпеки та підтримці великої кількості програмних бібліотек.

Однією з найбільш поширених операційних систем для одноплатних комп'ютерів є Raspberry Pi OS. Вона спеціально адаптована для роботи з платформою Raspberry Pi та підтримує GPIO-інтерфейси, аудіопристрої, мережеві модулі та периферійне обладнання. Система має графічний інтерфейс та підтримує командний рядок, що дозволяє зручно налаштовувати програмне забезпечення та виконувати адміністрування системи.

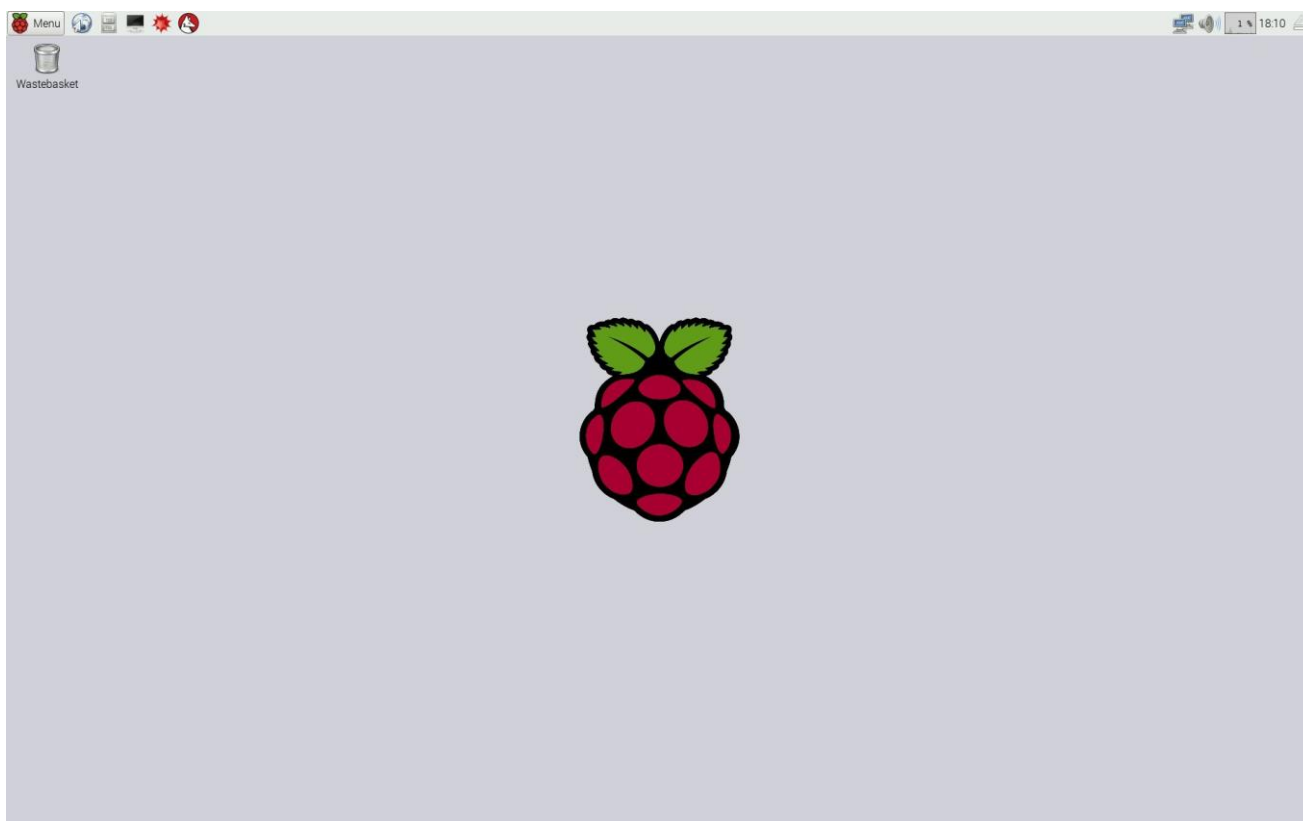


Рисунок — інтерфейс Raspberry Pi OS

Також для систем автоматизації можуть використовуватись Ubuntu Server або інші Linux-дистрибутиви. Ubuntu забезпечує стабільну роботу мережесервісів, підтримує велику кількість програмних пакетів та широко використовується у серверних системах і проєктах автоматизації.

Основними перевагами Linux-систем є:

- стабільність роботи;
- підтримка багатозадачності;
- висока безпека;
- підтримка роботи з мережами та периферійними пристроями;
- наявність великої кількості безкоштовного програмного забезпечення.

Для систем голосового керування важливою є підтримка роботи з аудіопристроями та бібліотеками розпізнавання мовлення. Linux забезпечує сумісність із програмними модулями для роботи з мікрофонами, аудіодрайверами та технологіями ASR і NLP.

Таблиця 3.5 – Операційні системи для систем голосового керування

Операційна система	Основне призначення	Переваги	Недоліки
Raspberry Pi OS	Робота з Raspberry Pi	Підтримка GPIO, простота налаштування	Обмежена продуктивність у складних задачах
Ubuntu Server	Серверні та мережеві системи	Стабільність, підтримка пакетів	Вищі вимоги до ресурсів
Debian Linux	Універсальні Linux-системи	Надійність та безпека	Складніше адміністрування
Arduino Firmware	Робота мікроконтролерів	Простота та швидкодія	Відсутність повноцінної ОС

Ще однією перевагою Linux є можливість віддаленого керування системою через SSH. Це дозволяє налаштовувати та обслуговувати систему без необхідності підключення монітора або клавіатури.

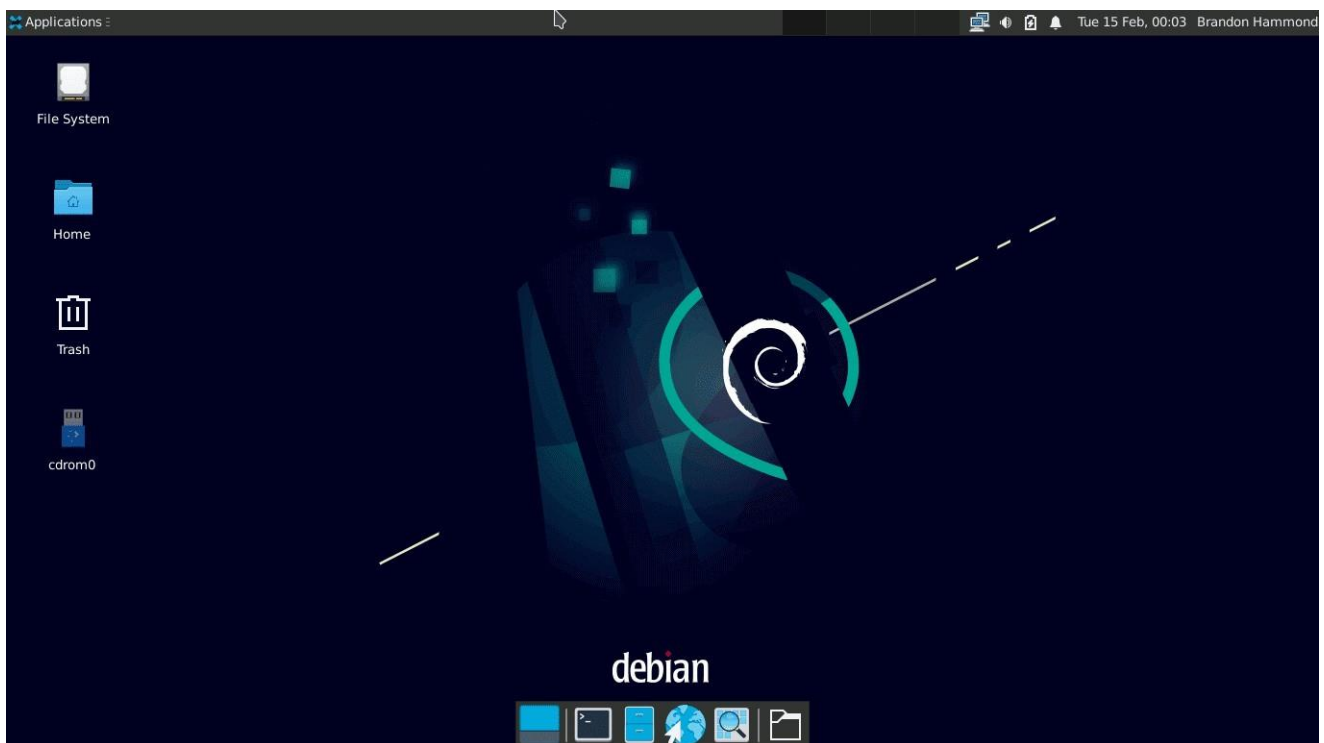


Рисунок — інтерфейс Debian Linux

Проте Linux-системи мають і певні недоліки. Для початківців налаштування операційної системи та встановлення необхідних бібліотек може бути складним. Крім того, деякі програми або драйвери можуть потребувати додаткового

налаштування.

У мікроконтролерних системах типу Arduino повноцінна операційна система зазвичай не використовується. Такі пристрої працюють із вбудованою прошивкою, яка безпосередньо виконує програмний код керування пристроями.

Таким чином, вибір операційної системи є важливим етапом реалізації системи голосового керування. Для систем розумного будинку найбільш доцільним є використання Linux-систем завдяки їх стабільності, гнучкості та підтримці необхідного програмного забезпечення для автоматизації та обробки голосових команд.

3.3.2 Мови програмування та бібліотеки

Для реалізації системи голосового керування важливу роль відіграє вибір мови програмування та програмних бібліотек. Саме програмне забезпечення забезпечує обробку голосових команд, взаємодію з периферійними пристроями та керування елементами розумного будинку.

Однією з найпоширеніших мов програмування у сфері автоматизації та систем розумного будинку є Python. Дана мова має простий синтаксис, велику кількість бібліотек та підтримує роботу з аудіо, мережею, GPIO-інтерфейсами та технологіями штучного інтелекту. Завдяки цьому Python широко використовується для створення систем голосового керування.

Основною перевагою Python є швидкість розробки та простота інтеграції різних програмних модулів. Крім того, мова підтримує багатоплатформеність та може працювати на Linux, Windows та інших операційних системах.

Для реалізації розпізнавання мовлення використовуються спеціалізовані бібліотеки ASR (Automatic Speech Recognition). Однією з найпоширеніших бібліотек є SpeechRecognition. Вона дозволяє працювати з мікрофоном, записувати аудіо та виконувати перетворення голосу у текстовий формат.

Для локального розпізнавання мовлення часто використовуються бібліотеки Vosk або PocketSphinx. Дані рішення дозволяють виконувати обробку

голосових команд без використання хмарних сервісів, що є важливим для автономних систем розумного будинку.

Для роботи з GPIO-контактами та керування виконавчими пристроями використовуються бібліотеки `gpiozero` та `RPi.GPIO`. Вони дозволяють керувати сервоприводами, реле, світлодіодами та іншими електронними компонентами за допомогою програмного коду.

У системах голосового керування також можуть використовуватись бібліотеки синтезу мовлення (TTS — Text-to-Speech). Однією з популярних бібліотек є `pyttsx3`, яка дозволяє генерувати голосові повідомлення та забезпечувати зворотний зв'язок із користувачем.

Для програмування мікроконтролерів Arduino зазвичай використовується мова C/C++ та середовище Arduino IDE. Такі рішення добре підходять для роботи з датчиками та виконавчими механізмами, проте мають обмежені можливості для складної обробки голосу.

При виборі мови програмування та бібліотек необхідно враховувати:

- простоту розробки;
- підтримку необхідних функцій;
- сумісність із апаратною платформою;
- можливість локальної обробки голосу;
- стабільність роботи системи.

Таким чином, Python є найбільш доцільною мовою програмування для реалізації системи голосового керування завдяки великій кількості бібліотек та підтримці технологій автоматизації. Використання спеціалізованих бібліотек для роботи з голосом та GPIO дозволяє створити функціональну та гнучку систему розумного будинку.

3.4 Аналіз технологій голосового керування

Технології голосового керування є основою сучасних систем автоматизації та розумного будинку. Вони забезпечують можливість взаємодії користувача з

електронними пристроями за допомогою голосових команд без використання фізичних елементів керування. Основними компонентами таких технологій є системи розпізнавання мовлення, обробки природної мови та синтезу мовлення.

Таблиця 3.6 – Основні мови програмування для систем автоматизації

Мова програмування	Основне призначення	Переваги	Недоліки
Python	Голосове керування, автоматизація	Простий синтаксис, багато бібліотек	Нижча швидкодія порівняно з C++
C/C++	Програмування мікроконтролерів	Висока швидкодія	Складніша розробка
JavaScript	Web-інтерфейси та IoT	Робота з вебтехнологіями	Менше бібліотек для GPIO
Java	Серверні системи	Багатоплатформеність	Вищі вимоги до ресурсів

Принцип роботи систем голосового керування полягає у перетворенні голосового сигналу користувача у текстовий формат з подальшим аналізом команди та виконанням відповідної дії. Для цього використовуються технології ASR (Automatic Speech Recognition) та NLP (Natural Language Processing).

Технологія ASR відповідає за автоматичне розпізнавання мовлення. Система отримує аудіосигнал через мікрофон, виконує його цифрову обробку та визначає текстовий зміст команди. Для підвищення точності розпізнавання використовуються алгоритми машинного навчання та нейронні мережі.

Сучасні системи розпізнавання мовлення можуть працювати як у хмарному, так і в локальному режимі. Хмарні сервіси забезпечують високу точність завдяки використанню потужних серверів та великих мовних моделей. Проте вони потребують постійного доступу до Інтернету та можуть створювати ризики для конфіденційності даних.

Локальні системи голосового керування виконують обробку команд безпосередньо на пристрої без передачі даних на зовнішні сервери. Для цього використовуються локальні модулі розпізнавання мовлення, наприклад Vosk або

PocketSphinx. Основними перевагами локального підходу є автономність роботи та захист персональної інформації користувача.

Після розпізнавання голосової команди система виконує аналіз її змісту за допомогою технологій NLP. Обробка природної мови дозволяє визначати наміри користувача та формувати відповідні дії системи. Наприклад, команда «увімкнути світло» після аналізу перетворюється у сигнал для активації виконавчого пристрою.

У системах голосового керування можуть використовуватись два основних підходи до аналізу команд:

- використання фіксованого набору команд;
- підтримка природної мови.

У першому випадку система працює із заздалегідь визначеними фразами. Такий підхід є простішим у реалізації та потребує менше обчислювальних ресурсів. У другому випадку система здатна розуміти різні варіанти формулювання команди, що забезпечує більш природну взаємодію з користувачем.

Для підвищення зручності роботи системи також можуть використовуватись технології синтезу мовлення (TTS — Text-to-Speech). Вони дозволяють системі генерувати голосові повідомлення та повідомляти користувача про виконання команд або стан системи.

Важливим аспектом технологій голосового керування є швидкодія та точність роботи системи. Якість розпізнавання мовлення залежить від характеристик мікрофона, рівня шуму, обчислювальної потужності пристрою та використовуваних алгоритмів обробки голосу.

Таким чином, технології голосового керування є важливою складовою сучасних систем розумного будинку. Використання ASR, NLP та TTS дозволяє створювати системи автоматизації з природною та зручною взаємодією між користувачем і електронними пристроями.

3.4.1 Технології розпізнавання мовлення (STT, ASR)

Технології розпізнавання мовлення є однією з основних складових систем голосового керування. Вони забезпечують перетворення голосу користувача у текстовий формат для подальшої обробки програмним забезпеченням системи. Найбільш поширеними технологіями у даній сфері є STT (Speech-to-Text) та ASR (Automatic Speech Recognition).

Технологія STT призначена для перетворення звукового сигналу у текст. Після отримання аудіосигналу через мікрофон система виконує його цифрову обробку та визначає текстовий зміст сказаних слів. Отриманий текст надалі використовується для аналізу команд та керування виконавчими пристроями.

ASR є більш широким поняттям та включає повний процес автоматичного розпізнавання мовлення. Дана технологія не лише перетворює голос у текст, а й аналізує мовні конструкції, окремі слова та особливості вимови. Сучасні ASR-системи використовують алгоритми машинного навчання, нейронні мережі та статистичні моделі для підвищення точності розпізнавання.

Таблиця 3.7 – Основні технології розпізнавання мовлення

Технологія	Призначення	Особливості
STT (Speech-to-Text)	Перетворення голосу у текст	Базова обробка мовлення
ASR (Automatic Speech Recognition)	Автоматичне розпізнавання мовлення	Аналіз слів, фраз та мовних конструкцій
NLP (Natural Language Processing)	Аналіз природної мови	Розуміння змісту команд
TTS (Text-to-Speech)	Генерація голосових повідомлень	Зворотний зв'язок із користувачем

Принцип роботи систем розпізнавання мовлення складається з декількох етапів. Спочатку мікрофон приймає голосовий сигнал користувача. Далі виконується попередня обробка аудіо: фільтрація шумів, нормалізація гучності та

виділення мовних характеристик. Після цього система аналізує звукові дані та визначає відповідні слова або фрази.

Для сучасних систем голосового керування важливою є можливість роботи у режимі реального часу. Чим швидше система виконує розпізнавання мовлення, тим комфортнішою є взаємодія користувача з розумним будинком.

Існують два основних підходи до реалізації технологій розпізнавання мовлення:

- хмарне розпізнавання;
- локальне розпізнавання.

Хмарні системи виконують обробку мовлення на віддалених серверах. Вони забезпечують високу точність роботи та підтримують складні алгоритми обробки природної мови. Проте такі системи потребують постійного доступу до Інтернету та можуть створювати ризики для конфіденційності даних.

Локальні системи розпізнавання мовлення працюють безпосередньо на пристрої без передачі даних у хмару. Для цього використовуються локальні модулі, наприклад Vosk або PocketSphinx. Основними перевагами такого підходу є автономність роботи та підвищений рівень захисту інформації.

Якість розпізнавання мовлення залежить від декількох факторів:

- якості мікрофона;
- рівня шуму у приміщенні;
- чіткості вимови користувача;
- продуктивності апаратної платформи;
- використовуваних алгоритмів обробки голосу.

Для реалізації систем голосового керування часто використовуються бібліотеки SpeechRecognition, Vosk API або інші програмні модулі, які забезпечують роботу з мікрофоном та обробку аудіосигналу.

Таким чином, технології STT та ASR є основою сучасних систем голосового керування. Вони дозволяють реалізувати природну взаємодію користувача з електронними пристроями та забезпечують ефективну роботу систем автоматизації розумного будинку.

3.4.2 Технології синтезу мовлення (TTS)

Технології синтезу мовлення (TTS — Text-to-Speech) використовуються у системах голосового керування для перетворення текстової інформації у звуковий сигнал. Завдяки TTS система може озвучувати повідомлення, підтверджувати виконання команд та забезпечувати зворотний зв'язок із користувачем.

Основне призначення технологій синтезу мовлення полягає у створенні природної голосової взаємодії між людиною та електронною системою. У системах розумного будинку TTS може використовуватись для повідомлень типу «світло увімкнено», «двері відкрито» або «команда не розпізнана».

Принцип роботи TTS полягає у перетворенні тексту у мовний сигнал за допомогою спеціальних алгоритмів. Система аналізує текст, визначає вимову слів, інтонацію та паузи, після чого генерує звукове повідомлення. Сучасні технології синтезу мовлення використовують методи машинного навчання та нейронні мережі для створення більш природного звучання голосу.

Існують два основних підходи до реалізації TTS:

- хмарний синтез мовлення;
- локальний синтез мовлення.

Хмарні системи синтезу мовлення використовують віддалені сервери для генерації голосу. Вони забезпечують високу якість звучання та підтримують велику кількість мов і голосів. Проте такі рішення потребують постійного доступу до Інтернету.

Локальні системи TTS працюють безпосередньо на пристрої без використання зовнішніх серверів. Основними перевагами такого підходу є автономність роботи та незалежність від інтернет-з'єднання. Локальний синтез мовлення часто використовується у системах розумного будинку, де важлива стабільність роботи системи.

Для реалізації синтезу мовлення у Python можуть використовуватись бібліотеки pyttsx3, gTTS або інші програмні модулі. Бібліотека pyttsx3 дозволяє виконувати локальний синтез мовлення без використання хмарних сервісів, що є важливим для автономних систем голосового керування.

Якість синтезу мовлення залежить від використовуваних алгоритмів, потужності системи та мовної моделі. Сучасні TTS-системи здатні генерувати голос, максимально наближений до природного мовлення людини.

У системах розумного будинку технології TTS підвищують зручність використання та покращують взаємодію користувача із системою. Озвучення результатів виконання команд дозволяє користувачу отримувати підтвердження роботи системи без необхідності використання екранів або додаткових індикаторів.

Таким чином, технології синтезу мовлення є важливою складовою систем голосового керування. Використання TTS дозволяє забезпечити голосовий зворотний зв'язок, підвищити комфорт використання системи та реалізувати більш природну взаємодію між користувачем і системою розумного будинку.

3.5 Формування загальної архітектури системи

Загальна архітектура системи голосового керування елементами розумного будинку визначає взаємозв'язок між апаратними та програмними компонентами, а також послідовність обробки голосової команди від моменту її подання користувачем до виконання відповідної дії. Правильно сформована архітектура дозволяє забезпечити стабільну роботу системи, простоту налаштування та можливість подальшого розширення функціональності.

Таблиця 3.7 – Основні компоненти архітектури розроблюваної системи

Компонент	Призначення в системі
Користувач	Подає голосову команду
USB-мікрофон	Приймає голосову команду та передає аудіосигнал
Raspberry Pi	Обробляє команду та керує виконавчим

Компонент	Призначення в системі
Модуль розпізнавання мовлення	пристроєм Перетворює голос у текст
Програмна логіка	Визначає, яку дію потрібно виконати
Сервопривід	Виконує фізичну дію, наприклад поворот механізму

Основу системи становить центральний керуючий пристрій, який виконує обробку голосових команд, приймає рішення щодо виконання дії та передає керуючі сигнали на виконавчі пристрої. До центрального пристрою підключаються мікрофон, програмні модулі розпізнавання мовлення, модуль аналізу команд та виконавчі механізми.

Робота системи починається з введення голосової команди користувачем. Мікрофон приймає звуковий сигнал і передає його до програмного модуля обробки аудіо. На цьому етапі виконується попередня обробка сигналу, зокрема зчитування звуку, фільтрація шумів та підготовка аудіоданих до розпізнавання.

Після цього використовується модуль розпізнавання мовлення, який перетворює голосову команду у текстовий формат. Отриманий текст передається до модуля аналізу команд. Даний модуль порівнює розпізнаний текст із заздалегідь визначеним набором команд та визначає, яку саме дію необхідно виконати.

Після визначення команди система формує керуючий сигнал для відповідного виконавчого пристрою. Наприклад, у разі команди «відкрити двері» система може передати PWM-сигнал на сервопривід, який виконує поворот на заданий кут. У разі команди «увімкнути світло» може бути активоване реле, яке замикає електричне коло.

Важливою складовою архітектури є модуль зворотного зв'язку. Він дозволяє повідомляти користувача про результат виконання команди. Зворотний зв'язок може реалізовуватись за допомогою голосового повідомлення, текстового

виводу у терміналі або світлової індикації. Наприклад, після успішного виконання дії система може повідомити: «Команду виконано».

Загальна архітектура системи може бути подана у вигляді такої послідовності: користувач подає голосову команду, мікрофон приймає аудіосигнал, програмний модуль виконує розпізнавання мовлення, модуль логіки визначає необхідну дію, після чого керуючий сигнал передається на виконавчий пристрій.

Перевагою такої архітектури є її модульність. Кожен компонент системи виконує окрему функцію, що спрощує тестування, налаштування та подальше вдосконалення системи. За необхідності можна додати нові голосові команди, підключити додаткові датчики або розширити кількість виконавчих пристроїв.

Таким чином, сформована архітектура системи голосового керування забезпечує логічну взаємодію між користувачем, програмним забезпеченням та апаратними компонентами. Такий підхід дозволяє реалізувати автономну систему керування елементами розумного будинку з можливістю подальшого розвитку та модернізації.

РОЗДІЛ 4. РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Проектування архітектури системи голосового керування

Проектування архітектури системи голосового керування є одним із основних етапів розробки системи розумного будинку. На даному етапі визначається структура системи, взаємодія між апаратними та програмними компонентами, а також принцип обробки голосових команд користувача.

Основною метою проектування є створення стабільної та функціональної системи, здатної виконувати голосове керування виконавчими пристроями у режимі реального часу. При розробці архітектури необхідно враховувати швидкодію системи, можливість локальної обробки команд, простоту масштабування та зручність подальшого налаштування.

Архітектура системи складається з декількох основних модулів:

- модуль введення звуку;
- модуль розпізнавання мовлення;
- модуль аналізу команд;
- модуль керування виконавчими пристроями;
- модуль зворотного зв'язку.

Модуль введення звуку відповідає за прийом голосових команд користувача. Для цього використовується мікрофон або USB-аудіопристрій, який передає аудіосигнал до системи обробки. На даному етапі також може виконуватись фільтрація шумів та попередня обробка аудіо.

Після отримання звукового сигналу роботу виконує модуль розпізнавання мовлення. Він аналізує аудіодані та перетворює голосову команду у текстовий формат. Для реалізації даного модуля можуть використовуватись технології ASR та спеціалізовані програмні бібліотеки для локального розпізнавання мовлення.

Далі текстова команда передається до модуля аналізу команд. Його основним завданням є визначення змісту команди та вибір необхідної дії. Модуль порівнює отриманий текст із набором підтримуваних команд та формує керуючий сигнал для відповідного виконавчого пристрою.

Модуль керування виконавчими пристроями забезпечує взаємодію системи з сервоприводами, реле та іншими елементами автоматизації. Для передачі сигналів використовуються GPIO-контакти та PWM-сигнали. Наприклад, команда «увімкнути світло» активує релейний модуль, а команда «відкрити двері» подає сигнал на сервопривід.

Для покращення взаємодії з користувачем у системі передбачений модуль зворотного зв'язку. Він може повідомляти про успішне виконання команд за допомогою текстових або голосових повідомлень. Це дозволяє користувачу контролювати стан системи та результат виконання дій.

Під час проєктування архітектури також враховувалась можливість подальшого розширення системи. Модульна структура дозволяє додавати нові голосові команди, підключати додаткові датчики або виконавчі пристрої без значної зміни основної логіки роботи системи.

Важливою особливістю розробленої архітектури є локальна обробка голосових команд. Це дозволяє забезпечити автономну роботу системи без постійного підключення до Інтернету та підвищити рівень конфіденційності даних користувача.

Таким чином, спроектована архітектура системи голосового керування забезпечує ефективну взаємодію між користувачем, програмними модулями та апаратними компонентами. Модульний підхід дозволяє реалізувати стабільну та гнучку систему автоматизації розумного будинку з можливістю подальшого розвитку функціональності.

4.2 Налаштування програмного середовища

Налаштування програмного середовища є важливим етапом реалізації системи голосового керування, оскільки саме на цьому етапі виконується підготовка операційної системи, встановлення необхідних бібліотек та конфігурація програмних компонентів для роботи системи.

Для реалізації системи було використано операційну систему Raspberry Pi OS на базі Linux. Дана система забезпечує підтримку роботи з GPIO-контактами,

аудіопристроями та програмними бібліотеками для обробки голосових команд. Встановлення операційної системи виконувалось на карту пам'яті microSD за допомогою спеціального програмного забезпечення Raspberry Pi Imager.

Після встановлення операційної системи було виконано початкове налаштування системи, зокрема:

- оновлення системних пакетів; налаштування мережевого підключення;
- активація SSH для віддаленого доступу;
- перевірка роботи USB-портів та аудіопристроїв.

Для оновлення системи використовувались стандартні Linux-команди оновлення пакетів та програмного забезпечення. Це дозволило забезпечити стабільну роботу системи та встановити актуальні версії бібліотек.

Наступним етапом стало встановлення мови програмування Python та необхідних програмних бібліотек. Для реалізації системи голосового керування були встановлені бібліотеки:

- SpeechRecognition — для роботи з мікрофоном та розпізнавання мовлення;
- Vosk — для локального розпізнавання голосових команд;
- PyAudio — для обробки аудіосигналу;
- gpiozero та RPi.GPIO — для керування GPIO-контактами;
- pyttsx3 — для синтезу мовлення та голосового зворотного зв'язку.

Після встановлення бібліотек було виконано перевірку роботи мікрофона та аудіосистеми. Для цього використовувались стандартні утиліти Linux та тестові Python-скрипти для запису і відтворення звуку. Коректна робота аудіопристроїв є важливою умовою стабільного розпізнавання голосових команд.

Також було налаштовано роботу GPIO-контактів для взаємодії з виконавчими пристроями. За допомогою тестових програм перевірялась можливість керування сервоприводами та релейними модулями через GPIO-інтерфейс.

Для спрощення розробки та тестування системи використовувався

віддалений доступ через SSH. Це дозволило виконувати налаштування та запуск програм без необхідності постійного підключення монітора та клавіатури до пристрою.

Після завершення налаштування програмного середовища було створено тестові програми для перевірки окремих компонентів системи. Зокрема, виконувалась перевірка:

- роботи мікрофона;
- розпізнавання голосових команд;
- передачі керуючих сигналів;
- роботи сервоприводів та реле.

Таким чином, налаштування програмного середовища дозволило підготувати систему до реалізації голосового керування та забезпечити стабільну взаємодію між програмними й апаратними компонентами системи розумного будинку.

4.3 Підключення та конфігурація апаратних компонентів (GPIO, PWM)

Після налаштування програмного середовища було виконано підключення та конфігурацію апаратних компонентів системи голосового керування. Основною метою даного етапу є забезпечення взаємодії між центральним пристроєм керування та виконавчими механізмами системи розумного будинку.

Для підключення периферійних пристроїв використовувались GPIO-контакти (General Purpose Input/Output). GPIO являють собою універсальні входи-виходи, які дозволяють передавати сигнали між одноплатним комп'ютером та зовнішніми електронними компонентами. За допомогою GPIO можна керувати реле, сервоприводами, світлодіодами та іншими виконавчими пристроями.

Одним із основних компонентів системи є сервопривід, який використовується для виконання механічних дій, наприклад відкриття або закриття дверей. Для підключення сервоприводу використовуються три контакти:

- живлення;
- земля (GND);

– сигнальний контакт.

Сигнальний контакт сервоприводу підключається до GPIO-виходу, який підтримує PWM-сигнали. PWM (Pulse Width Modulation) — це технологія широтно-імпульсної модуляції, яка дозволяє керувати положенням сервоприводу шляхом зміни тривалості імпульсів.

Принцип роботи PWM полягає у формуванні послідовності імпульсів із різною шириною. Залежно від параметрів сигналу сервопривід повертає вал на відповідний кут. Для системи голосового керування це дозволяє реалізувати автоматичне відкриття або закриття механізмів після отримання голосової команди.

Окрім сервоприводу, до системи було підключено релейний модуль. Реле використовується для керування електричними пристроями, наприклад освітленням або вентиляторами. Керування реле виконується через GPIO-вихід шляхом подачі логічного сигналу високого або низького рівня.

Підключення релейного модуля включає:

- підключення живлення;
- підключення контакту GND;
- підключення керуючого контакту до GPIO.

Для безпечної роботи реле забезпечує електричну ізоляцію між низьковольтною системою керування та силовими електричними колами.

Також було виконано підключення USB-мікрофона, який використовується для прийому голосових команд користувача. Після підключення мікрофона система автоматично визначила аудіопристрій, після чого було виконано тестування запису та відтворення звуку.

Для програмної конфігурації GPIO використовувались бібліотеки `gpiozero` та `RPi.GPIO`. За допомогою Python-програм було налаштовано режими роботи GPIO-контактів та реалізовано передачу сигналів на виконавчі пристрої.

Після завершення підключення апаратних компонентів було проведено

тестування роботи системи. Перевірялась коректність роботи PWM-сигналів, реакція сервоприводу на керуючі команди, а також вмикання та вимикання релейного модуля.

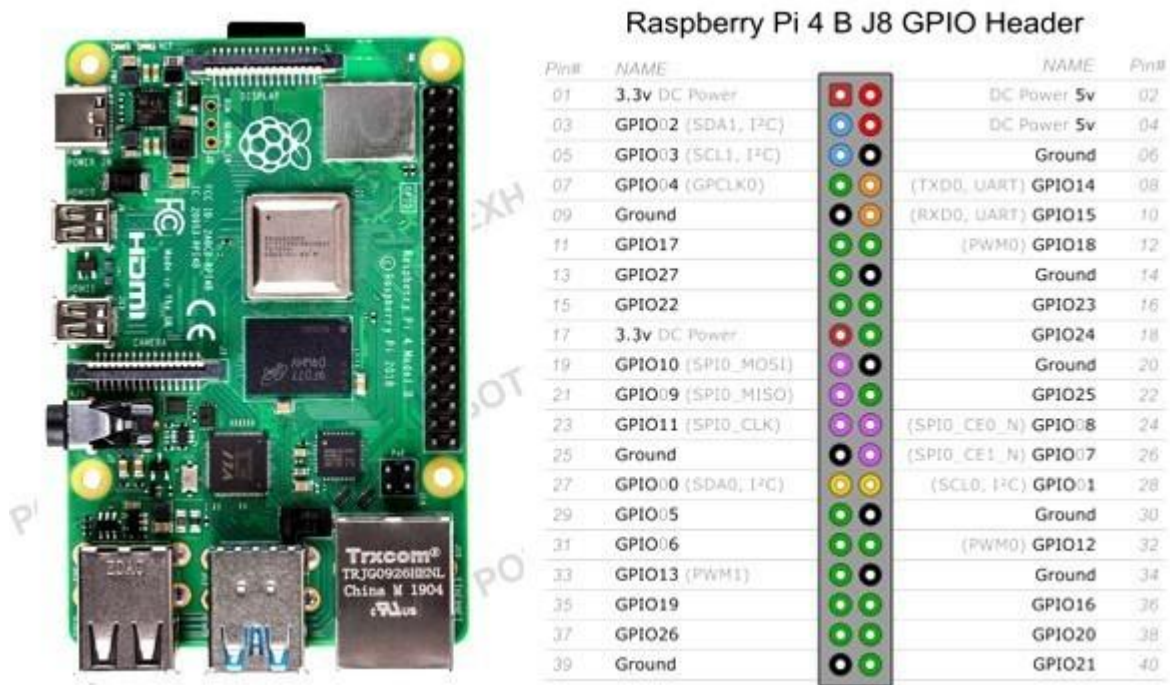


Рисунок 4.1 — Розпіновка одноплатного комп'ютера Raspberry Pi 4 B

Таким чином, підключення та конфігурація апаратних компонентів дозволили забезпечити взаємодію між програмною частиною системи та виконавчими пристроями. Використання GPIO та PWM забезпечує можливість реалізації автоматизованого голосового керування елементами розумного будинку.

4.4 Розробка алгоритму обробки голосових команд

Одним із ключових етапів реалізації системи голосового керування є розробка алгоритму обробки голосових команд. Даний алгоритм забезпечує взаємодію між користувачем та системою, виконує аналіз голосового сигналу та формує відповідні керуючі команди для виконавчих пристроїв.

Основною задачею алгоритму є прийом голосової команди, її розпізнавання, аналіз та виконання необхідної дії. Алгоритм повинен працювати у режимі реального часу та забезпечувати швидку реакцію системи на команди користувача.

Робота алгоритму починається з активації мікрофона та прослуховування навколишнього середовища. Після виявлення голосового сигналу система виконує запис аудіоданих та передає їх до модуля розпізнавання мовлення. На наступному етапі виконується попередня обробка аудіосигналу. Вона включає:

- фільтрацію сторонніх шумів;
- нормалізацію гучності;
- підготовку аудіоданих до розпізнавання.

Після цього система використовує технологію ASR для перетворення голосового сигналу у текстовий формат. Для реалізації локального розпізнавання мовлення можуть використовуватись бібліотеки Vosk або SpeechRecognition.

Отриманий текст передається до модуля аналізу команд. Даний модуль порівнює текст із набором підтримуваних голосових команд. Для спрощення реалізації у системі використовується набір заздалегідь визначених фраз, наприклад:

- «увімкнути світло»;
- «вимкнути світло»;
- «відкрити двері»;
- «закрити двері».

Після визначення команди алгоритм формує відповідний керуючий сигнал. Якщо команда пов'язана з керуванням освітленням, система активує релейний модуль через GPIO-контакт. Якщо команда стосується механічної дії, наприклад відкриття дверей, формується PWM-сигнал для керування сервоприводом.

У разі успішного виконання команди система може надсилати користувачу повідомлення про результат роботи. Для цього використовується текстовий або голосовий зворотний зв'язок за допомогою технологій TTS.

Також алгоритм передбачає обробку помилок. Якщо команда не була розпізнана або не входить до списку підтримуваних, система повідомляє користувача про помилку та переходить у режим очікування нової команди.



Рисунок 4.2 — Реалізація фізичного макету

Для підвищення стабільності роботи алгоритму було реалізовано циклічний режим прослуховування мікрофона. Це дозволяє системі постійно очікувати нові голосові команди без необхідності повторного запуску програми.

Загальна послідовність роботи алгоритму має такий вигляд:

1. Очікування голосової команди.
2. Запис аудіосигналу.
3. Попередня обробка звуку.
4. Розпізнавання мовлення.
5. Аналіз текстової команди.
6. Формування керуючого сигналу.
7. Виконання дії виконавчим пристроєм.
8. Виведення повідомлення про результат виконання.

Таким чином, розроблений алгоритм забезпечує повний цикл обробки голосових команд та дозволяє реалізувати автоматизоване керування елементами розумного будинку за допомогою голосового інтерфейсу.

4.5 Реалізація програмного забезпечення системи

Після проєктування архітектури та розробки алгоритму роботи системи було виконано реалізацію програмного забезпечення для голосового керування елементами розумного будинку. Основним завданням програмної частини є забезпечення взаємодії між користувачем, модулем розпізнавання мовлення та виконавчими пристроями системи.

Для реалізації програмного забезпечення була використана мова програмування Python. Даний вибір обумовлений простотою розробки, великою кількістю бібліотек для роботи з аудіо та GPIO, а також підтримкою технологій розпізнавання мовлення.

Програмне забезпечення системи складається з декількох основних модулів:

- модуль роботи з мікрофоном;
- модуль розпізнавання мовлення;
- модуль аналізу команд;
- модуль керування виконавчими пристроями;
- модуль зворотного зв'язку.

Модуль роботи з мікрофоном відповідає за запис аудіосигналу та передачу його до системи розпізнавання мовлення. Для реалізації даного модуля використовувалась бібліотека PyAudio, яка забезпечує роботу з аудіопристроями та запис звуку у режимі реального часу.

Для реалізації розпізнавання мовлення використовувалась бібліотека SpeechRecognition разом із локальним модулем Vosk. Після отримання голосової команди система виконує аналіз аудіосигналу та перетворює мовлення у текстовий формат.

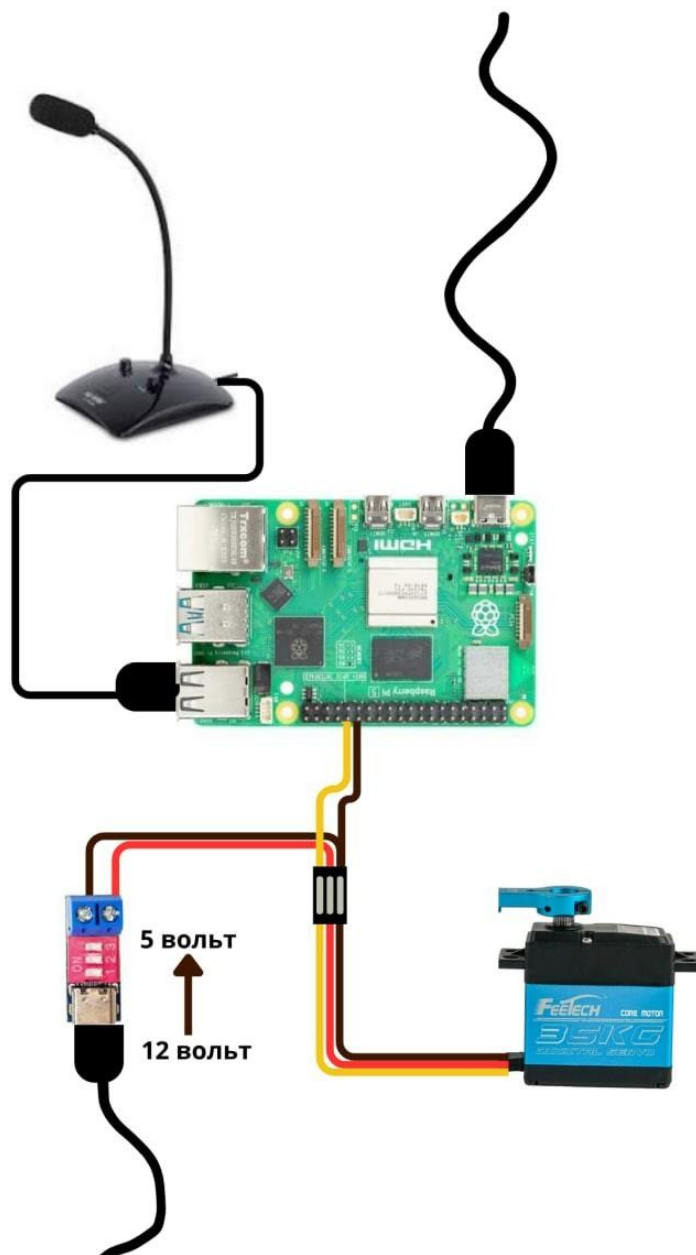


Рисунок 4.3 — Схема збірки

Отриманий текст передається до модуля аналізу команд. У програмі реалізовано набір умов, які порівнюють розпізнану команду із заздалегідь визначеним списком підтримуваних фраз. Наприклад, після розпізнавання команди «увімкнути світло» система активує релейний модуль, а команда «відкрити двері» запускає керування сервоприводом.

Для керування GPIO-контактами використовувались бібліотеки `gpiozero` та `RPi.GPIO`. За допомогою програмного коду реалізовано передачу сигналів на виконавчі пристрої та керування PWM-сигналами для сервоприводу.

Також у програмному забезпеченні реалізовано голосовий зворотний зв'язок. Для цього використовувалась бібліотека `pyttsx3`, яка дозволяє генерувати голосові повідомлення без використання хмарних сервісів. Після виконання команди система може повідомляти користувача про результат роботи, наприклад: «Світло увімкнено» або «Двері відкрито».

Для забезпечення стабільної роботи система працює у циклічному режимі. Програма постійно очікує нові голосові команди та виконує їх обробку у реальному часі. У разі помилки розпізнавання або отримання невідомої команди система повідомляє користувача про помилку та повертається у режим очікування.

Під час реалізації програмного забезпечення також було проведено тестування окремих модулів системи. Перевірялась коректність роботи мікрофона, точність розпізнавання мовлення, передача сигналів через GPIO та робота виконавчих пристроїв.

Таким чином, реалізоване програмне забезпечення забезпечує повний цикл роботи системи голосового керування — від прийому голосової команди до виконання відповідної дії виконавчим пристроєм. Використання Python та спеціалізованих бібліотек дозволило створити функціональну, гнучку та автономну систему розумного будинку.

Приклад кодуЮ що використовувався для написання програми:

```
import RPi.GPIO as GPIO

import time

import speech_recognition as sr

import pyttsx3
```

```
SERVO_PIN = 18
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(SERVO_PIN, GPIO.OUT)
```

```
servo = GPIO.PWM(SERVO_PIN, 50)
```

```
servo.start(0)
```

```
engine = pyttsx3.init()
```

```
recognizer = sr.Recognizer()
```

```
def speak(text):
```

```
    engine.say(text)
```

```
    engine.runAndWait()
```

```
def set_servo_angle(angle):
```

```
    duty = 2 + (angle / 18)
```

```
    GPIO.output(SERVO_PIN, True)
```

```
    servo.ChangeDutyCycle(duty)
```

```
    time.sleep(0.5)
```

```
    GPIO.output(SERVO_PIN, False)
```

```
servo.ChangeDutyCycle(0)
```

```
def listen_command():
```

```
    with sr.Microphone() as source:
```

```
        print("Слухаю команду...")
```

```
        recognizer.adjust_for_ambient_noise(source)
```

```
        audio = recognizer.listen(source)
```

```
    try:
```

```
        command = recognizer.recognize_google(audio, language="uk-UA")
```

```
        command = command.lower()
```

```
        print("Розпізнано:", command)
```

```
        return command
```

```
    except sr.UnknownValueError:
```

```
        print("Команду не розпізнано")
```

```
        return ""
```

```
    except sr.RequestError:
```

```
        print("Помилка сервісу розпізнавання")
```

```
        return ""
```

```
try:
```

```
    speak("Система голосового керування запущена")
```

```
while True:
```

```
    command = listen_command()
```

```
    if "відкрий" in command or "увімкни" in command:
```

```
        speak("Виконую команду відкриття")
```

```
        set_servo_angle(90)
```

```
    elif "закрий" in command or "вимкни" in command:
```

```
        speak("Виконую команду закриття")
```

```
        set_servo_angle(0)
```

```
    elif "стоп" in command or "завершити" in command:
```

```
        speak("Роботу системи завершено")
```

```
        break
```

```
    else:
```

```
        speak("Команду не розпізнано")
```

```
except KeyboardInterrupt:
```

```
    print("Програму зупинено користувачем")
```

```
finally:
```

```
    servo.stop()
```

```
    GPIO.cleanup()
```

РОЗДІЛ 5. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

5.1 Методика тестування системи

Після реалізації апаратної та програмної частини системи голосового керування було проведено тестування її працездатності. Основною метою тестування є перевірка коректності роботи всіх компонентів системи, оцінка точності розпізнавання голосових команд та визначення стабільності роботи системи у різних умовах використання.

Тестування проводилось поетапно та охоплювало як окремі модулі системи, так і повний цикл її роботи. Під час перевірки оцінювались:

- коректність роботи мікрофона;
- точність розпізнавання голосових команд;
- швидкість реакції системи;
- робота GPIO-контактів;
- коректність керування виконавчими пристроями;
- стабільність роботи програмного забезпечення.

На першому етапі було виконано тестування аудіосистеми та мікрофона. Перевірялась якість запису звуку, стабільність роботи аудіодрайверів та можливість коректного прийому голосових команд. Також оцінювався вплив сторонніх шумів на якість розпізнавання мовлення.

Наступним етапом стало тестування модуля розпізнавання мовлення. Для цього використовувався набір заздалегідь визначених голосових команд, наприклад:

- «увімкнути світло»;
- «вимкнути світло»;

- «відкрити двері»;
- «закрити двері».

Кожна команда вимовлялась декілька разів у різних умовах: при нормальному рівні шуму, на різній відстані від мікрофона та з різною швидкістю вимови. Результати тестування дозволили оцінити точність роботи системи та визначити можливі помилки розпізнавання.

Після перевірки розпізнавання мовлення було проведено тестування роботи виконавчих пристроїв. Перевірялась коректність передачі сигналів через GPIO-контакти та робота PWM-керування сервоприводом. Також оцінювалась стабільність роботи релейного модуля при багаторазовому вмиканні та вимиканні.

Окремо було перевірено швидкість реакції системи. Вимірювався час між поданням голосової команди та виконанням відповідної дії виконавчим пристроєм. Даний параметр є важливим для забезпечення комфортної взаємодії користувача із системою.

Для перевірки стабільності роботи система тестувалась у режимі тривалого безперервного функціонування. Під час тестування оцінювалась робота програмного забезпечення, відсутність помилок у роботі GPIO та стабільність розпізнавання голосових команд.

Також було проведено тестування обробки помилок. Перевірялась реакція системи на невідомі або некоректні команди. У таких випадках система повинна повідомляти користувача про помилку та повертатись у режим очікування нової команди.

Таким чином, розроблена методика тестування дозволила комплексно перевірити працездатність системи голосового керування, оцінити ефективність роботи окремих модулів та визначити стабільність функціонування системи розумного будинку.

5.2 Перевірка функціональності системи

Після завершення розробки системи голосового керування було проведено перевірку її функціональності. Основною метою даного етапу є визначення коректності роботи всіх програмних та апаратних компонентів системи, а також перевірка виконання голосових команд і взаємодії між окремими модулями.

Перевірка функціональності проводилась у реальних умовах роботи системи. Під час тестування використовувався мікрофон для введення голосових команд, а також виконавчі пристрої — сервопривід та релейний модуль.

На першому етапі було перевірено запуск програмного забезпечення та коректність роботи всіх програмних модулів. Система успішно запускала після завантаження операційної системи та переходила у режим очікування голосових команд.

Далі проводилась перевірка роботи мікрофона та модуля розпізнавання мовлення. Для тестування використовувався набір команд:

- «увімкнути світло»;
- «вимкнути світло»;
- «відкрити двері»;
- «закрити двері».

Після вимовлення голосової команди система виконувала запис аудіосигналу, розпізнавання мовлення та аналіз текстової команди. У разі успішного розпізнавання команда передавалась до модуля керування виконавчими пристроями.

Під час перевірки роботи релейного модуля система успішно виконувала вмикання та вимикання підключеного навантаження. GPIO-контакти коректно передавали сигнали керування, а реле стабільно реагувало на отримані команди.

Також було проведено перевірку роботи сервоприводу. Після отримання голосової команди система формувала PWM-сигнал, який змінював положення

сервоприводу на заданий кут. Сервопривід коректно виконував механічні дії без помітних затримок.

Окремо перевірялась робота голосового зворотного зв'язку. Після виконання команди система генерувала повідомлення про результат роботи, наприклад «Світло увімкнено» або «Двері відкрито». Це дозволило покращити взаємодію користувача із системою.

Під час тестування також перевірялась робота системи у випадку помилкових або невідомих команд. Якщо команда не розпізнавалась або не входила до списку підтримуваних, система повідомляла користувача про помилку та продовжувала роботу у режимі очікування.

Для оцінки стабільності роботи система запускала у режимі тривалого функціонування. У процесі тестування не було виявлено критичних помилок або зависань програмного забезпечення. Усі основні функції системи працювали стабільно та коректно.

Таким чином, перевірка функціональності підтвердила працездатність розробленої системи голосового керування. Система успішно виконувала розпізнавання голосових команд, керування виконавчими пристроями та забезпечувала стабільну взаємодію між програмними й апаратними компонентами розумного будинку.

5.3 Аналіз точності розпізнавання голосових команд

Одним із найважливіших параметрів системи голосового керування є точність розпізнавання голосових команд. Від правильності роботи модуля розпізнавання мовлення залежить стабільність роботи всієї системи та зручність взаємодії користувача з елементами розумного будинку.

Для оцінки точності розпізнавання було проведено серію тестувань із використанням набору голосових команд. Під час тестування оцінювалась здатність системи правильно визначати команди користувача та виконувати відповідні дії без помилок.

Для перевірки використовувались такі команди:

- «увімкнути світло»;
- «вимкнути світло»;
- «відкрити двері»;
- «закрити двері».

Кожна команда вимовлялась декілька разів у різних умовах експлуатації.

Тестування проводилось:

- у тихому приміщенні;
- при наявності фонового шуму;
- на різній відстані від мікрофона;
- із різною швидкістю вимови команд.

Під час тестування у тихому приміщенні система демонструвала високий рівень точності розпізнавання. Більшість команд була коректно розпізнана з першої спроби, а виконавчі пристрої правильно реагували на отримані сигнали.

При збільшенні рівня фонового шуму спостерігалось незначне зниження точності роботи системи. У деяких випадках команди могли розпізнаватись некоректно або потребували повторного вимовлення. Основною причиною помилок був вплив сторонніх звуків на якість аудіосигналу.

Також було встановлено, що на точність розпізнавання впливає відстань між користувачем та мікрофоном. Найкращі результати система показувала при розташуванні користувача на невеликій відстані від пристрою введення звуку.

Під час тестування використовувалась локальна система розпізнавання мовлення, що дозволило забезпечити автономність роботи системи без використання хмарних сервісів. Проте локальна обробка голосу має дещо нижчу точність у порівнянні з сучасними хмарними рішеннями, які використовують потужні мовні моделі та великі обчислювальні ресурси.

Для підвищення точності розпізнавання у системі використовувався обмежений набір заздалегідь визначених команд. Такий підхід дозволив зменшити кількість помилок та спростити аналіз голосових даних.

Під час тестування також перевірялась швидкість розпізнавання команд. Система забезпечувала достатньо швидку обробку голосових запитів, а затримка між вимовленням команди та виконанням дії залишалась мінімальною.

У результаті проведеного аналізу було встановлено, що система забезпечує стабільне розпізнавання базових голосових команд та може ефективно використовуватись для керування елементами розумного будинку. Основними факторами, які впливають на точність роботи системи, є якість мікрофона, рівень шуму та чіткість вимови користувача.

Таким чином, результати тестування підтвердили працездатність системи голосового керування та достатній рівень точності розпізнавання команд для реалізації базових функцій автоматизації розумного будинку.

5.4 Оцінка продуктивності та стабільності роботи системи

Після завершення функціонального тестування було проведено оцінку продуктивності та стабільності роботи системи голосового керування. Основною метою даного етапу є визначення швидкодії системи, її здатності працювати у режимі реального часу та стабільно виконувати голосові команди протягом тривалого часу.

Під час оцінки продуктивності аналізувалась швидкість виконання основних операцій системи:

- запис та обробка аудіосигналу;
- розпізнавання голосових команд;
- аналіз текстових даних;
- передача сигналів на виконавчі пристрої;
- виконання механічних дій сервоприводом або реле.

Особлива увага приділялась часу реакції системи. Було визначено, що після вимовлення голосової команди система виконувала її обробку та запускала відповідну дію з мінімальною затримкою. Це забезпечувало комфортну взаємодію користувача із системою розумного будинку.

Під час тестування система стабільно працювала у циклічному режимі прослуховування мікрофона. Програмне забезпечення безперервно очікувало нові голосові команди та виконувало їх обробку без необхідності повторного запуску програми.

Для оцінки стабільності роботи було проведено тривале тестування системи. Протягом декількох годин система виконувала обробку голосових команд та керування виконавчими пристроями. У процесі роботи не було виявлено критичних помилок, зависань або аварійного завершення програмного забезпечення.

Також перевірялась стабільність роботи GPIO-контактів та PWM-сигналів. Сервопривід коректно реагував на сигнали керування та стабільно виконував механічні дії. Релейний модуль також працював без помилок та забезпечував надійне вмикання і вимикання підключених пристроїв.

Під час оцінки продуктивності було встановлено, що локальна обробка голосових команд потребує певних обчислювальних ресурсів. При використанні складніших моделей розпізнавання мовлення навантаження на процесор може збільшуватись, що впливає на швидкість роботи системи. Проте для реалізації базових функцій голосового керування продуктивності обраної апаратної платформи було достатньо.

Окремо оцінювався вплив зовнішніх факторів на стабільність роботи системи. Підвищений рівень шуму або нестабільне живлення могли частково впливати на точність розпізнавання мовлення, проте загальна працездатність системи залишалась стабільною.

Важливою перевагою розробленої системи є автономність роботи. Оскільки обробка голосових команд виконується локально, система не залежить від наявності інтернет-з'єднання та може стабільно функціонувати у автономному режимі.

У результаті проведеного тестування було встановлено, що система забезпечує достатній рівень продуктивності для реалізації голосового керування елементами розумного будинку. Програмне забезпечення та апаратні компоненти стабільно взаємодіють між собою та забезпечують надійну роботу системи у режимі реального часу.

Таким чином, оцінка продуктивності та стабільності підтвердила ефективність реалізованої системи голосового керування та можливість її використання у практичних системах автоматизації розумного будинку.

5.5 Рекомендації щодо вдосконалення системи

Розроблена система голосового керування продемонструвала стабільну роботу та можливість реалізації базових функцій автоматизації розумного будинку. Проте під час тестування було виявлено ряд аспектів, які можуть бути вдосконалені для підвищення продуктивності, точності розпізнавання мовлення та розширення функціональності системи.

Одним із основних напрямів удосконалення є покращення точності розпізнавання голосових команд. Для цього доцільно використовувати більш сучасні моделі ASR або інтегрувати потужніші алгоритми машинного навчання. Це дозволить системі краще працювати у шумному середовищі та коректніше розпізнавати команди різних користувачів.

Також перспективним напрямом є використання мікрофонів із системою шумозаглушення або масивів мікрофонів. Такі рішення дозволяють покращити якість аудіосигналу та зменшити вплив сторонніх шумів на роботу системи.

Для підвищення продуктивності системи можливе використання більш потужних апаратних платформ або оптимізація програмного коду. Це дозволить

зменшити затримку між вимовленням команди та виконанням відповідної дії, а також забезпечити підтримку складніших алгоритмів обробки мовлення.

Ще одним напрямом удосконалення є розширення функціональності системи. У майбутньому до системи можна додати:

- підтримку більшої кількості голосових команд;
- інтеграцію з датчиками температури, руху або освітленості;
- автоматичні сценарії роботи;
- мобільний або веб-інтерфейс для керування системою;
- підтримку декількох користувачів.

Для підвищення зручності використання доцільно реалізувати підтримку природної мови. Замість використання лише фіксованого набору команд система зможе аналізувати різні варіанти формулювання запитів користувача, що зробить взаємодію більш природною.

Окрему увагу варто приділити питанням безпеки. У майбутньому система може бути доповнена механізмами автентифікації користувача за голосом. Це дозволить обмежити доступ сторонніх осіб до керування елементами розумного будинку.

Перспективним також є використання резервного живлення для забезпечення стабільної роботи системи при відключенні електроенергії. Це підвищить надійність системи та дозволить уникнути втрати працездатності у критичних ситуаціях.

Таким чином, розроблена система має значний потенціал для подальшого розвитку та модернізації. Використання сучасних технологій розпізнавання мовлення, покращення апаратної частини та розширення функціональності дозволить створити більш ефективну, зручну та інтелектуальну систему голосового керування розумним будинком.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено систему голосового керування елементами розумного будинку на основі локальної обробки голосових команд. Під час виконання роботи було проведено аналіз сучасних технологій автоматизації, систем голосового керування та підходів до реалізації розумного будинку.

У першому розділі було досліджено сучасний стан розвитку систем розумного будинку, розглянуто технології голосового керування та обробки мовлення, а також виконано аналіз існуючих хмарних, локальних і гібридних систем голосових асистентів. У результаті аналізу було визначено переваги та недоліки різних підходів до реалізації систем голосового керування.

У другому розділі було розглянуто варіанти побудови системи та виконано порівняння локальних і хмарних рішень. На основі проведеного аналізу було обрано локальний підхід до реалізації системи, що дозволяє забезпечити автономність роботи, зменшити залежність від Інтернету та підвищити рівень конфіденційності даних користувача.

У третьому розділі було виконано огляд апаратних платформ, периферійних пристроїв та програмного забезпечення, необхідного для реалізації системи. Було проаналізовано можливості одноплатних комп'ютерів, мікроконтролерів Arduino, мікрофонних модулів, сервоприводів та релейних модулів. Також було досліджено технології розпізнавання та синтезу мовлення.

У четвертому розділі було реалізовано програмну та апаратну частину системи. Виконано налаштування програмного середовища, підключення апаратних компонентів через GPIO-інтерфейси та розроблено алгоритм обробки голосових команд. Для реалізації системи було використано мову програмування Python та спеціалізовані бібліотеки для роботи з голосом і виконавчими пристроями.

У п'ятому розділі було проведено тестування системи та оцінку її працездатності. Результати тестування показали, що система коректно виконує розпізнавання голосових команд, забезпечує керування виконавчими пристроями та стабільно працює у режимі реального часу. Також було проаналізовано фактори, які впливають на точність розпізнавання мовлення та продуктивність системи.

У результаті виконання дипломної роботи було досягнуто поставленої мети — створено працездатну систему голосового керування елементами розумного будинку з локальною обробкою голосових команд. Розроблена система може використовуватись як основа для подальшого розвитку та вдосконалення систем автоматизації розумного будинку.

Перспективами подальшого розвитку є покращення точності розпізнавання мовлення, інтеграція додаткових датчиків та IoT-пристроїв, підтримка природної мови та створення мобільного інтерфейсу для віддаленого керування системою.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Raspberry Pi Official Documentation](#)
2. [Arduino Official Documentation](#)
3. [Python Official Documentation](#)
4. [Vosk Speech Recognition Toolkit](#)
5. [SpeechRecognition Python Library](#)
6. [PyAudio Documentation](#)
7. [RPi.GPIO Documentation](#)
8. [gpiozero Documentation](#)
9. [pytttsx3 Documentation](#)
10. [Ubuntu Official Website](#)
11. [Raspberry Pi OS Documentation](#)
12. [Google Assistant Official Website](#)
13. [Amazon Alexa Official Website](#)
14. [Apple Siri Official Website](#)
15. [MQTT Official Website](#)
16. Буч Г. Об'єктно-орієнтований аналіз і проєктування з прикладами застосувань. – Київ : Діалектика, 2019. – 720 с.
17. Таненбаум Е. Комп'ютерні мережі. – Київ : Вільямс, 2020. – 960 с.
18. Соммервілл І. Інженерія програмного забезпечення. – Київ : Вільямс, 2018. – 912 с.
19. Stallings W. Operating Systems: Internals and Design Principles. – Pearson, 2018. – 800 p.
20. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. – Pearson, 2021. – 1152 p.

