

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: « Інтеграція систем розумного будинку з
використанням ChatGPT для керування та автоматизації »

на здобуття освітнього ступеня бакалавра (магістра)
зі спеціальності 126, інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело*

(підпис)

Дмитро КОЗЯР
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач(ка) вищої освіти гр. ____

Дмитро КОЗЯР

Ім'я, ПРІЗВИЩЕ

Керівник: PhD Валентина Данильченко

науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Рецензент:

науковий ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність інформаційні системи та технології

Освітньо-професійна програма інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру Інф. систем та технологій

Каміла СТОРЧАК Ім'я, ПРИЗВИЩЕ

« ____ » _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Козяр Дмитро Сергійович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: "Інтеграція систем розумного будинку з використанням ChatGPT для керування та автоматизації"

керівник кваліфікаційної роботи Валентина Данильченко PhD,

(Ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «24» березня 2025 р. №56

2. Строк подання кваліфікаційної роботи «30» травня _____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи: Науково технічна література з теми бакалаврської роботи, Аналіз існуючих систем для автоматизації розумного будинку

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

1. Проектування системи голосового керування для розумного будинку на базі ChatGPT

2. Архітектура інтегрованої системи з ChatGPT та Home Assistant

3. Ключові технологічні виклики

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ	10.03.25	Виконано
2	Розділ 1. ОГЛЯД ТЕХНОЛОГІЙ РОЗУМНОГО БУДИНКУ	20.03.25	Виконано
3	Розділ 2. СНАТГРТ ЯК ІНСТРУМЕНТ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ	10.04.25	Виконано
4	Розділ 3. ІНТЕГРАЦІЯ СНАТГРТ З HOME ASSISTANT	07.04.25	Виконано
5	Розділ 4 . АНАЛІЗ ТА БЕЗПЕКА	25.04.25	Виконано
6	Висновки	30.04.25	Виконано
7	Оформлення дипломної бакалаврської роботи (чистовий варіант)	05.05.25	Виконано

Здобувач(ка) вищої освіти

(підпис)

Дмитро КОЗЯР

(Ім'я, ПРИЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Валентина ДАНИЛЬЧЕНКО

(Ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра (магістра): 59 стор., 21рис., 18 джерел.

Об'єкт дослідження – Інтелектуальні системи автоматизації житлових приміщень

Предмет дослідження – Методи обробки природної мови для керування пристроями розумного будинку

Мета роботи – Розробка та реалізація системи розумного будинку з голосовим управлінням на базі великих мовних моделей (LLM), інтегрованих у платформу Home Assistant.

Методи дослідження – Аналіз архітектури Home Assistant, тестування протоколів зв'язку, інтеграція API ChatGPT/OpenAI, застосування локальних моделей Whisper/Piper

Короткий зміст роботи: У рамках кваліфікаційної роботи було розроблено систему інтеграції розумного будинку з використанням великих мовних моделей та досліджено можливості поєднання Home Assistant з технологіями обробки природної мови, зокрема ChatGPT, для реалізації голосового управління.

КЛЮЧОВІ СЛОВА: РОЗУМНИЙ БУДИНОК, HOME ASSISTANT, ВЕЛИКІ МОВНІ МОДЕЛІ (LLM), CHATGPT, ГОЛОСОВИЙ АСИСТЕНТ, АВТОМАТИЗАЦІЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ЛОКАЛЬНА ОБРОБКА, ХМАРНІ СЕРВІСИ, ІНТЕГРАЦІЯ, ІНТЕРНЕТ РЕЧЕЙ (IOT), ГОЛОСОВЕ КЕРУВАННЯ, КОНФІДЕНЦІЙНІСТЬ, TTS, STT, WYOMING.

ЗМІСТ

ВСТУП.....	с. 8
1. ОГЛЯД ТЕХНОЛОГІЙ РОЗУМНОГО БУДИНКУ	с. 9
1.1 Основні концепції розумного будинку	с. 10
1.2 Компоненти системи розумного будинку	с. 12
1.3 Протоколи зв'язку та стандарти	с. 14
1.4 Автоматизація та штучний інтелект.....	с.17
2. CHATGPT ЯК ІНСТРУМЕНТ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ	с. 15
2.1 Апаратне забезпечення голосового помічника	с. 21
2.2 Супутник голосового асистента (Voice Assist Satellite)	с. 23
2.3 Локальна та хмарна обробка мовлення в Home Assistant	с. 26
2.4 Перетворення тексту в мову (TTS).....	с. 29
2.5 Перетворення мови в текст (STT).....	с. 31
2.6 Розмовний агент та розпізнавання намірів (Intents)	с. 34
2.7 Речення (Sentences) у Home Assistant.....	с. 41
2.8 Використання LLM у Home Assistant.....	с. 43
3. ІНТЕГРАЦІЯ З HOME ASSISTANT	с. 46
3.1 Голосовий pipeline (STT → інтерпретація → TTS)	с.47
3.2 Custom Sentences і псевдоніми.....	с. 49
3.3 Інтеграція OpenAI API з Home Assistant	с.51
4. АНАЛІЗ ТА БЕЗПЕКА СИСТЕМИ	с. 54
4.1 Продуктивність і швидкодія системи.....	с. 55
4.2 Конфіденційність та захист даних користувача	с. 57

ВИСНОВКИ	с. 59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	с.61
ДОДАТКИ	с.64

ВСТУП

Сучасні технології розумного будинку стрімко розвиваються, інтегруючи штучний інтелект (ШІ) для підвищення комфорту, безпеки та енергоефективності житлових і комерційних приміщень. Одним із найперспективніших напрямків є використання мовних моделей, таких як ChatGPT, для керування та автоматизації розумного будинку.

На сьогоднішній день більшість систем розумного будинку керуються через спеціалізовані мобільні додатки або голосові асистенти, такі як Alexa, Google Assistant і Siri. Однак ці рішення мають певні обмеження, зокрема необхідність прив'язки до конкретної екосистеми, обмежений набір команд та труднощі в налаштуванні індивідуальних сценаріїв. Використання ChatGPT дозволяє значно розширити можливості автоматизації завдяки обробці природної мови, генерації гнучких відповідей та аналізу контексту запитів.

Інтеграція ChatGPT із системами розумного будинку відкриває нові можливості, зокрема:

Гнучке голосове та текстове керування – користувач може формулювати команди в довільній формі без необхідності запам'ятовування конкретних фраз.

Адаптивна автоматизація – система може навчатися та вдосконалювати сценарії управління на основі поведінки користувача.

Підвищення безпеки – розумний будинок може аналізувати поведінку мешканців, повідомляти про підозрілі події та навіть моделювати присутність людей в оселі.

Інтеграція з широким спектром пристроїв – на відміну від традиційних голосових асистентів, ChatGPT може працювати через API та легко підключатися до різних платформ без жорсткої прив'язки до виробника.

Окрім переваг, використання ChatGPT у розумному будинку ставить питання конфіденційності, безпеки даних та продуктивності інтегрованої системи. Тому дослідження цієї теми є важливим як для розробників розумних технологій, так і для кінцевих користувачів, які прагнуть отримати ефективне, безпечне та зручне рішення для автоматизації своїх осель.

Таким чином, актуальність даного дослідження обумовлена потребою в інтелектуальних, адаптивних та безпечних рішеннях для розумного будинку, які можуть покращити якість життя та підвищити рівень автоматизації завдяки інтеграції штучного інтелекту.

1 ОГЛЯД ТЕХНОЛОГІЙ РОЗУМНОГО БУДИНКУ

Розумний будинок — це сучасна система, що об'єднує різноманітні електронні пристрої та технології для автоматизації побутових процесів. Основна мета таких систем полягає у підвищенні комфорту, енергоефективності, безпеки та зручності керування житловим простором. Концепція розумного будинку передбачає можливість дистанційного управління освітленням, кліматом, безпекою, електроприладами та іншими системами за допомогою мобільних додатків, голосових команд або автоматичних сценаріїв.

Функціональність розумного будинку забезпечується взаємодією різних компонентів, серед яких ключову роль відіграють контролери, датчики, виконавчі пристрої та комунікаційні модулі. Контролери слугують центральним вузлом системи, який приймає команди від користувача та передає їх відповідним пристроям. Датчики, у свою чергу, збирають інформацію про стан навколишнього середовища, зокрема температуру, рівень вологості, освітлення, рух або наявність диму. Виконавчі пристрої виконують отримані команди, наприклад, вмикають або вимикають світло, регулюють температуру, блокують двері чи відкривають жалюзі. Всі ці елементи взаємодіють між собою через різні протоколи зв'язку, що забезпечують ефективний і стабільний обмін даними.

Для передачі інформації між пристроями використовуються різні технології, серед яких найпоширенішими є Wi-Fi, Bluetooth, Zigbee, Z-Wave та MQTT. Wi-Fi є одним із найпопулярніших рішень завдяки високій швидкості передачі даних, проте його недоліком є підвищене енергоспоживання. Bluetooth зазвичай застосовується для короткочасного підключення пристроїв на невеликій відстані. Zigbee та Z-Wave є спеціалізованими протоколами для розумних будинків, які забезпечують стабільний зв'язок між великою кількістю пристроїв з мінімальним

споживанням енергії. MQTT використовується для обміну даними між IoT-пристроями через інтернет і забезпечує швидку та надійну передачу інформації навіть при нестабільному з'єднанні.

Сучасні системи розумного будинку дедалі частіше інтегрують технології штучного інтелекту, що дозволяє значно розширити можливості автоматизації. Штучний інтелект аналізує поведінку користувача, вивчає його звички та адаптує роботу пристроїв відповідно до індивідуальних потреб. Наприклад, система може автоматично регулювати температуру в приміщенні залежно від часу доби або прогнозу погоди, вмикати освітлення при наближенні мешканців до будинку чи імітувати їхню присутність під час від'їзду для підвищення рівня безпеки.

Окреме місце в екосистемі розумного будинку займають голосові асистенти, такі як Alexa, Google Assistant та Siri. Вони дозволяють взаємодіяти з пристроями за допомогою голосових команд, спрощуючи процес керування. Проте ці системи мають певні обмеження, зокрема потребують чітко визначених команд і прив'язані до конкретних виробників. Використання мовних моделей, таких як ChatGPT, відкриває нові можливості для інтелектуального керування розумним будинком. Завдяки глибокому аналізу контексту та можливості обробки природної мови, така система може виконувати складні запити, аналізувати поведінку користувача та пропонувати персоналізовані сценарії автоматизації.

Розвиток технологій розумного будинку продовжує набирати обертів, і з кожним роком ці системи стають дедалі розумнішими, зручнішими та доступнішими. Інтеграція штучного інтелекту та гнучких рішень, таких як ChatGPT, відкриває перспективи створення ще більш ефективних та інтуїтивних способів управління житловим простором.

1.1 Основні концепції розумного будинку

Розумний будинок — це інтегрована система, яка поєднує різноманітні технології для автоматизації житлового простору. Головною ідеєю такого підходу є створення комфортного, безпечного та енергоефективного середовища, в якому всі пристрої можуть взаємодіяти між собою, виконуючи завдання відповідно до налаштувань користувача або на основі аналізу його звичок.

Однією з ключових концепцій розумного будинку є автоматизація, що дозволяє мінімізувати необхідність втручання людини у керування побутовими процесами. Система може самостійно вмикати та вимикати освітлення, регулювати температуру в приміщенні, контролювати витрати електроенергії та води, забезпечувати безпеку шляхом відеоспостереження і розпізнавання небезпечних ситуацій. Усе це відбувається завдяки інтеграції датчиків, контролерів і виконавчих пристроїв, які взаємодіють між собою через різні комунікаційні протоколи.

Ще однією важливою концепцією є дистанційне керування, що дозволяє користувачам контролювати всі функції будинку за допомогою смартфона, планшета чи комп'ютера. Спеціальні мобільні додатки або веб-інтерфейси забезпечують доступ до всіх налаштувань у режимі реального часу, незалежно від місцезнаходження власника. Голосові асистенти, такі як Alexa, Google Assistant або Siri, розширюють можливості управління, дозволяючи взаємодіяти з пристроями за допомогою голосових команд.

Ще одним важливим аспектом є адаптивність системи, що дає можливість розумному будинку навчатися та пристосовуватися до індивідуальних потреб мешканців. Наприклад, система може аналізувати, коли користувач зазвичай приходить додому, і заздалегідь налаштовувати комфортну температуру або автоматично відкривати двері при наближенні власника. Штучний інтелект у таких системах дозволяє створювати складні сценарії, що підлаштовуються під зміну умов та потреб мешканців.

Безпека також є одним із ключових напрямків у розвитку концепції розумного будинку. Системи відеоспостереження, датчики руху, інтелектуальні замки та сигналізація забезпечують захист житла від несанкціонованого доступу. Деякі рішення дозволяють аналізувати поведінку людей, що входять у приміщення, та розпізнавати підозрілі дії, а у разі загрози – надсилати власнику сповіщення або викликати відповідні служби.

Окрім безпеки, важливим фактором є енергоефективність. Розумний будинок допомагає оптимізувати споживання ресурсів, зменшуючи витрати на електроенергію, опалення та водопостачання. Інтелектуальні термостати автоматично регулюють температуру відповідно до зовнішніх умов, а розумні розетки та вимикачі запобігають марному витрачання енергії.

Загалом, концепція розумного будинку базується на інтеграції технологій, що працюють разом для забезпечення максимального комфорту, безпеки та ефективності. Завдяки постійному розвитку штучного інтелекту та автоматизованих систем такі рішення стають дедалі доступнішими, що відкриває нові можливості для розширення функціоналу та персоналізації житлового простору.

1.2 Компоненти системи розумного будинку

Система розумного будинку складається з кількох основних компонентів, які забезпечують її функціональність, взаємодію між пристроями та зручне управління. Кожен із цих елементів виконує свою роль у забезпеченні автоматизації, безпеки та енергоефективності житлового простору.

Головним елементом будь-якої системи є контролер, який виступає центральним керуючим пристроєм. Він координує роботу всіх інших елементів, отримує дані від датчиків, обробляє їх та надсилає відповідні команди виконавчим пристроям. Контролери можуть бути представлені у

вигляді спеціалізованих хабів, програмних платформ або навіть мобільних додатків, які працюють на основі хмарних технологій.

Датчики відіграють важливу роль у зборі інформації про стан навколишнього середовища та активність у приміщенні. Вони можуть вимірювати температуру, вологість, рівень освітлення, рух, наявність диму чи газу, відкривання дверей і вікон, протікання води та інші параметри. На основі отриманих даних система може автоматично виконувати певні дії, наприклад, вмикати освітлення при виявленні руху або активувати сигналізацію у разі несанкціонованого доступу.

Виконавчі пристрої виконують команди, надіслані контролером, і безпосередньо впливають на функціонування будинку. До цієї категорії належать розумні розетки, лампи, термостати, електронні замки, моторизовані штори та жалюзі, кондиціонери, системи поливу та інші пристрої. Наприклад, при підключенні до розумної системи освітлення лампи можуть автоматично змінювати яскравість або колір залежно від часу доби або настрою користувача.

Зв'язок між усіма компонентами здійснюється за допомогою комунікаційних модулів, які використовують різні протоколи передачі даних. Серед найбільш популярних є Wi-Fi, що забезпечує швидку передачу великих обсягів даних, Zigbee та Z-Wave, які оптимізовані для роботи з великою кількістю малопотужних пристроїв, а також Bluetooth, що використовується для з'єднання на короткій відстані. Крім того, протоколи MQTT та Matter дозволяють інтегрувати пристрої різних виробників у єдину екосистему.

Інтерфейси управління забезпечують зручний спосіб взаємодії користувача із системою. Вони можуть бути представлені у вигляді мобільних додатків, голосових асистентів, веб-інтерфейсів або сенсорних панелей, встановлених у приміщенні. Голосові помічники, такі як Google Assistant, Alexa або Siri, дозволяють користувачам керувати пристроями за

допомогою голосових команд, що робить процес управління ще більш зручним та інтуїтивним.

Завдяки поєднанню всіх цих компонентів розумний будинок стає комплексною автоматизованою системою, яка підвищує комфорт, безпеку та ефективність використання ресурсів у повсякденному житті. Розвиток штучного інтелекту та машинного навчання дозволяє зробити такі системи ще більш адаптивними, здатними передбачати потреби користувачів і автоматично налаштовувати їх відповідно до індивідуальних уподобань.

1.3 Протоколи зв'язку та стандарти у системах розумного будинку

Ефективне функціонування розумного будинку залежить від надійного обміну даними між пристроями. Для цього використовуються різні протоколи зв'язку, які визначають, як інформація передається між контролерами, датчиками та виконавчими пристроями. Вибір конкретного протоколу залежить від вимог до енергоспоживання, дальності з'єднання, пропускної здатності та сумісності пристроїв.

Zigbee – це бездротовий протокол, розроблений спеціально для IoT-пристроїв. Він працює у частотному діапазоні 2,4 ГГц та підтримує низьке енергоспоживання, що робить його ідеальним для датчиків і виконавчих пристроїв, які працюють від батарейок. Завдяки можливості створення мережі за топологією “mesh” (сітчаста мережа), кожен пристрій може виступати ретранслятором сигналу, що значно покращує дальність і надійність з'єднання. Zigbee використовується у багатьох екосистемах, зокрема у продуктах компаній Philips Hue, IKEA Tradfri та інших.

Z-Wave – ще один енергоефективний протокол, розроблений спеціально для систем автоматизації. Він працює у діапазоні 800-900 МГц, що забезпечує менше перешкод порівняно з Zigbee, оскільки більшість побутових пристроїв використовують частоту 2,4 ГГц. Як і Zigbee, Z-Wave підтримує “mesh”-мережі, що дозволяє пристроям взаємодіяти між собою

безпосередньо або через проміжні вузли. Z-Wave має строгий контроль сумісності пристроїв, що робить його надійним рішенням для розумного будинку, проте вартість пристроїв на цьому стандарті зазвичай вища.

MQTT – це протокол обміну повідомленнями, який широко використовується у системах розумного будинку для комунікації між IoT-пристроями через інтернет або локальну мережу. Його головною особливістю є модель “публікація-підписка”, яка дозволяє ефективно передавати дані між численними пристроями з мінімальними затримками. MQTT відзначається низьким споживанням трафіку та високою надійністю, що робить його ідеальним для віддаленого управління розумним будинком.

Wi-Fi є одним із найпоширеніших протоколів для підключення пристроїв розумного будинку. Завдяки високій швидкості передачі даних він підходить для камер відеоспостереження, розумних колонок та мультимедійних пристроїв. Основний недолік Wi-Fi – високе енергоспоживання, що робить його менш ефективним для пристроїв, які працюють від батарейок. Однак, завдяки широкому поширенню Wi-Fi-мережі, цей стандарт часто використовується у бюджетних рішеннях для розумного будинку.

Bluetooth, особливо його енергоефективна версія Bluetooth Low Energy (BLE), використовується для підключення мобільних пристроїв до системи розумного будинку. BLE споживає менше енергії, ніж класичний Bluetooth, що дозволяє використовувати його в пристроях, які працюють на батарейках протягом тривалого часу. Недоліком Bluetooth є обмежена дальність зв'язку, тому він зазвичай застосовується для короткочасного керування пристроями, такими як дверні замки, термометри або фітнес-трекери.

Кожен із цих протоколів має свої переваги та сфери застосування, тому в сучасних системах розумного будинку часто використовують їхню комбінацію для досягнення максимальної ефективності та зручності

управління. Інтеграція цих стандартів дозволяє створити надійну та гнучку інфраструктуру для автоматизації житлових приміщень.

1.4 Автоматизація та штучний інтелект у розумному будинку

Сучасні системи розумного будинку активно використовують автоматизацію та штучний інтелект для підвищення комфорту, енергоефективності та безпеки. Автоматизація дозволяє зменшити потребу в ручному управлінні пристроями, забезпечуючи їхню роботу за заданими сценаріями або у відповідь на певні події. Штучний інтелект, у свою чергу, розширює можливості розумного будинку, дозволяючи системам адаптуватися до звичок користувачів, прогнозувати їхні потреби та самостійно приймати рішення.

Автоматизація в розумному будинку базується на заздалегідь налаштованих алгоритмах та сценаріях. Наприклад, освітлення може автоматично вмикатися при заході сонця або при виявленні руху в кімнаті, а кліматична система може регулювати температуру відповідно до часу доби або присутності людей у приміщенні. Такі сценарії можуть налаштовуватися користувачами вручну або визначатися на основі аналізу їхніх дій.

Завдяки штучному інтелекту розумний будинок може навчатися та адаптуватися до поведінки мешканців. Алгоритми машинного навчання аналізують дані з датчиків і пристроїв, виявляючи закономірності у використанні освітлення, опалення, побутових приладів та систем безпеки. Наприклад, якщо користувач щоранку о 7:00 вмикає кавоварку, система може автоматично запускати її у відповідний час.

Штучний інтелект також відіграє ключову роль у забезпеченні безпеки. Розумні камери можуть розпізнавати обличчя мешканців та відрізняти їх від незнайомих. У разі виявлення підозрілої активності система може надсилати сповіщення власнику або навіть викликати службу

охорони. Аналіз звуків дозволяє ідентифікувати небезпечні ситуації, наприклад, розбиття скла або сигналізацію диму.

Голосові асистенти, такі як Amazon Alexa, Google Assistant та Apple Siri, інтегруються з розумним будинком, надаючи можливість керування пристроями за допомогою голосових команд. Вони не лише виконують команди користувачів, але й можуть пропонувати оптимальні сценарії автоматизації на основі отриманих даних.

Штучний інтелект також сприяє підвищенню енергоефективності, оптимізуючи витрати електроенергії та тепла. Інтелектуальні термостати, такі як Google Nest або Ecobee, аналізують температуру в приміщенні, зовнішні погодні умови та поведінку мешканців, регулюючи роботу системи опалення та кондиціонування. Завдяки цьому знижується споживання енергії, що дозволяє суттєво економити кошти.

Загалом, автоматизація та штучний інтелект значно розширюють можливості розумного будинку, роблячи його більш зручним, безпечним та ефективним. Завдяки постійному розвитку цих технологій розумний будинок стає не просто набором керованих пристроїв, а цілісною екосистемою, здатною самостійно приймати рішення та адаптуватися до потреб користувачів.

2 CHATGPT ЯК ІНСТРУМЕНТ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ

2.1 Апаратне забезпечення Voice Assistant Sattelite

Для реалізації голосового помічника в рамках даної дипломної роботи було обрано кілька апаратних рішень, які забезпечують ефективну взаємодію між користувачем та системою **Home Assistant**. Кожен з пристроїв має свої особливості, які дозволяють налаштовувати голосове управління різними способами, як для простих задач, так і для більш складних інтеграцій. У цій частині розглянуто основні компоненти, що використовуються для реалізації проекту.

Voice Assistant PE (Preview Edition) є одним з рекомендованих пристроїв для інтеграції голосового управління в систему Home Assistant. Цей пристрій поєднує в собі компактність, доступність і простоту налаштування. Voice Assistant PE дозволяє зручно інтегрувати голосовий інтерфейс для управління розумним домом, підтримуючи як локальну обробку голосових команд, так і хмарні сервіси. Його основною перевагою є швидкий старт завдяки інтуїтивно зрозумілому майстру налаштування, який полегшує інтеграцію в домашнє середовище.

ReSpeaker Lite — це ще один популярний пристрій, спеціально розроблений для роботи з голосовими помічниками. Вони оснащені кількома мікрофонами для точнішого виявлення голосових команд з різних напрямків. ReSpeaker Lite активно використовується для створення локальних голосових помічників завдяки своїм можливостям розпізнавання мови та точному виявленню команд. Це обладнання є популярним серед ентузіастів і розробників, оскільки підтримує інтеграцію з різними платформами та забезпечує високу гнучкість налаштувань.

ESP32-S3-Box — це потужний мікроконтролер на базі ESP32, що

підтримує Wi-Fi і Bluetooth зв'язок. Він є одним з найбільш універсальних рішень для створення голосових помічників завдяки своїй здатності працювати як з локальними програмами, так і з хмарними сервісами. ESP32-S3-Box підтримує обробку голосових команд на місці, що дозволяє знизити затримку в обробці та покращити точність розпізнавання мови. Цей пристрій ідеально підходить для проектів, де важлива енергоефективність і можливість інтеграції з іншими розумними пристроями.

M5stack Atom Echo — це компактний пристрій, що є частиною екосистеми M5Stack. Він надає можливість створювати голосових помічників для різних додатків завдяки вбудованому мікрофону та підтримці голосових сервісів. Завдяки невеликим розмірам, M5stack Atom Echo ідеально підходить для інтеграції в домашнє середовище без потреби в додатковому місці для установки. Це ідеальний варіант для створення прототипів голосових помічників або для інтеграції в специфічні проекти, де важлива компактність та простота.

Саморобний пристрій, виготовлений на базі доступних компонентів, став важливим етапом у процесі створення голосового помічника. Використовуючи такі платформи, як ESP32 або Raspberry Pi, можна створити голосовий інтерфейс, що дозволяє користувачам взаємодіяти з Home Assistant. Основними перевагами саморобного пристрою є висока гнучкість налаштувань, можливість підключення додаткових датчиків або пристроїв, а також економічність у порівнянні з комерційними рішеннями. Це підходить для користувачів, які хочуть налаштувати систему голосового управління відповідно до своїх потреб.



Рисунок 2.1 Апаратне забезпечення супутників голосового помічника

2.2 Супутник голосового асистента (Voice Assist Satellite)

Щоб зрозуміти, як працює супутник голосового асистента в Home Assistant, важливо розібратися, де саме відбувається обробка голосу. У голосових асистентах можна виділити два основні підходи: локальну обробку, коли всі дії виконуються безпосередньо на пристрої, і віддалену, коли запит надсилається на сервер.

Локальна обробка дозволяє пристрою самостійно розпізнавати команди та виконувати їх без потреби в інтернеті. Це забезпечує швидкість і конфіденційність, оскільки дані не передаються на сторонні сервери. Наприклад, якщо ви хочете увімкнути світло чи змінити температуру в кімнаті, локальна обробка дає змогу зробити це без затримок, навіть якщо з'єднання з інтернетом тимчасово відсутнє.

Віддалена обробка працює інакше. Пристрій записує голос і надсилає його на сервер, де він аналізується й повертається у вигляді результату. Цей підхід часто використовують великі голосові асистенти, такі як Google Assistant, Alexa чи Siri, оскільки їхні сервери здатні виконувати складні обчислення. Але це має свої недоліки: команди обробляються повільніше, асистент не працює без інтернету, а особисті дані можуть зберігатися на сторонніх сервісах, що викликає занепокоєння щодо приватності.

Home Assistant використовує змішаний підхід, щоб поєднати переваги обох методів. Основний пристрій Home Assistant може обробляти команди локально, забезпечуючи швидкість і безпеку. Але супутники — додаткові пристрої, розташовані в різних кімнатах, — не завжди мають достатньо потужності, щоб самостійно виконувати складні обчислення. Тому вони лише передають аудіопотік на головний пристрій, де відбувається розпізнавання голосу та виконання команд. Завдяки цьому можна використовувати прості та доступні пристрої, такі як Raspberry Pi, у ролі супутників, що значно знижує вартість системи.

Раніше навіть для активації голосового асистента потрібно було передавати звук на головний пристрій або сервер, що могло спричиняти затримки. Але Home Assistant зробив прорив у цьому напрямку. Завдяки інтеграції з проєктом openWakeWord система навчилася розпізнавати Wake Word (тобто слово-пробудження, наприклад, «Асистент») безпосередньо на супутниках. Це дозволяє пристрою постійно слухати навколишнє середовище, але починати обробку лише тоді, коли користувач вимовляє потрібну фразу. Такий підхід зменшує навантаження на мережу, робить систему швидшою і підвищує її надійність, оскільки асистент працює навіть без підключення до інтернету.

Для зв'язку між супутниками та основним пристроєм Home Assistant використовує спеціальний протокол Wyoming. Він забезпечує ефективну передачу аудіоданих, що дозволяє системі розпізнавати голос у режимі реального часу без значних затримок. Це означає, що незалежно від того, де ви перебуваєте у своєму домі, голосовий асистент зможе миттєво реагувати на ваші команди.

Такий підхід робить Home Assistant потужною, гнучкою та безпечною системою голосового керування. Користувач отримує швидкий відгук, а його голосові команди залишаються конфіденційними, оскільки вся обробка може виконуватися локально. Завдяки використанню супутників голосового асистента навіть великі будинки можуть бути повністю покриті системою голосового управління, без необхідності встановлення окремого повноцінного асистента в кожній кімнаті. Це не лише зручно, а й економічно вигідно, оскільки дозволяє використовувати прості та доступні пристрої як частину розумного будинку.

Для реалізації Voice Satellite використовуються голосові сателіти, які є малими пристроями з мікрофонами, такими як ESP32 з підтримкою аудіо або Raspberry Pi із зовнішніми мікрофонами. Центральний сервер є потужним пристроєм для обробки голосових команд, наприклад, Home

Assistant із додатком Rhasspy або Mycroft. Мережеве з'єднання здійснюється через Wi-Fi або Ethernet для передачі даних між сателітами і сервером. Динаміки для зворотного зв'язку можуть бути вбудовані в сателіти або використовуватися окремо через Bluetooth чи інші протоколи.

Популярними програмними рішеннями для реалізації голосових асистентів із підтримкою Voice Satellite є Rhasspy, який працює локально без підключення до хмарних сервісів, Mycroft, що є відкритим рішенням для інтеграції з власним сервером голосового розпізнавання, а також Home Assistant Assist, що підтримує взаємодію із сателітами.

Концепція Voice Satellite дозволяє значно розширити можливості голосового управління у розумному домі, забезпечуючи зручність, економію ресурсів та підвищену приватність. Використання локального голосового розпізнавання мінімізує залежність від хмарних сервісів і покращує швидкість реакції системи на команди.

2.3 Локальна та хмарна обробка мовлення в Home Assistant

Голосове керування розумним будинком у Home Assistant стало значно доступнішим та гнучкішим завдяки можливості вибору між локальною та хмарною обробкою мовлення. Це дозволяє кожному користувачеві адаптувати систему під свої потреби — чи то забезпечити максимальну конфіденційність, чи просто зекономити на обчислювальних ресурсах. Суть відмінності між цими двома підходами полягає в тому, де саме відбувається обробка вашої мови: на вашому пристрої або на віддаленому сервері.

Локальна обробка мовлення відбувається повністю у вас вдома. Коли ви щось говорите, система розпізнає, аналізує та озвучує відповідь без підключення до зовнішніх серверів. Це забезпечує максимальний контроль над особистими даними і дозволяє працювати навіть тоді, коли немає інтернету. Але варто розуміти, що така автономність потребує ресурсів. Для

локального голосового асистента важливо мати достатньо потужне обладнання, адже розпізнавання мовлення та синтез голосу — це дуже ресурсомісткі задачі. Саме тому спільнота Home Assistant часто рекомендує використовувати комп'ютери на базі процесора Intel N100, або потужніше. Це невеликий, але доволі потужний енергоефективний процесор, який добре справляється з задачами локального голосового асистента. Його продуктивності вистачає, щоб обробляти аудіо в реальному часі, запускати голосові моделі та працювати без затримок.

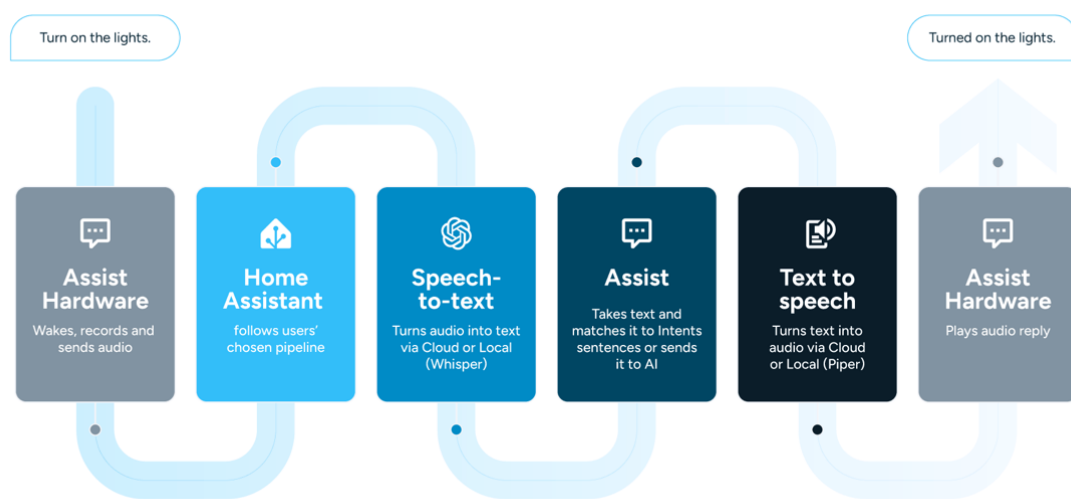


Рисунок 2.2 Схема роботи голосового помічника

Ще одна сильна сторона локального рішення — це гнучкість у виборі пристроїв, з яких можна взаємодіяти голосом. Наприклад, окрім основного сервера з Home Assistant, можна використовувати так звані супутникові пристрої. Це окремі модулі, які розміщуються в різних кімнатах і мають мікрофони та динаміки. Вони підключаються до головного сервера й передають голосові команди для обробки. Як супутники можуть виступати навіть мікроконтролери — наприклад, ESP32 з мікрофоном, які можуть надсилати звук на основний сервер через протокол Wyoming. Також супутниками можуть бути Raspberry Pi Zero або будь-який пристрій, що має аудіоінтерфейси та мережеве підключення. Окрім апаратних рішень, Home Assistant дозволяє керувати голосом через мобільний застосунок або

веббраузер. Мобільний додаток на Android чи iOS має вбудовану функцію Assist, яка дозволяє передавати голосові команди напряму. Через браузер, якщо використовується захищене з'єднання HTTPS, також можна активувати голосовий інтерфейс — це зручно для комп'ютерів чи планшетів.

У свою чергу, хмарна обробка мовлення в Home Assistant реалізована через сервіс Home Assistant Cloud або за допомогою сторонніх великих мовних моделей (LLM), наприклад OpenAI або Google. Коли ви говорите команду, ваш голос може бути відправлений у хмару, де спеціалізовані мовні моделі її обробляють. Це дозволяє отримати дуже гнучку інтерпретацію мови, навіть якщо команда була сформульована вільно або незвично. Хмарні LLM здатні “домислити”, що саме мав на увазі користувач, і навіть адаптуватися до контексту. Такий підхід не потребує потужного заліза вдома і працює навіть на слабших пристроях. Але він вимагає постійного підключення до інтернету та, зазвичай, щомісячної підписки.

На відміну від цього, локальні голосові помічники не використовують мовні моделі, які «здогадуються» про зміст сказаного. Натомість вони працюють на основі чітко визначених намірів (intents) та речень (sentences). Іншими словами, ви заздалегідь створюєте шаблони фраз і вказуєте, яку дію слід виконати у відповідь. Наприклад, можна налаштувати намір “увімкнути світло”, і додати кілька варіантів речень: “ввімкни світло в кухні”, “зроби світло яскравішим”, “додай освітлення”. Якщо ваша команда збігається з одним із заданих шаблонів, система розпізнає дію і виконає її. Це працює дуже швидко, точно та не потребує хмари, але вимагає трохи більше підготовки під час налаштування.

Таким чином, Home Assistant сьогодні надає повну свободу вибору. Якщо ви шукаєте приватність, стабільність та повний контроль — варто обрати локальну обробку. Якщо ж вам важлива гнучкість та легкість, а

також можливість використовувати новітні можливості штучного інтелекту — тоді хмара буде зручнішою. Найцікавіше, що обидва підходи можуть співіснувати — ви можете використовувати локального асистента вдома, а хмарного через мобільний додаток або браузер, залежно від задачі. Це і є справжня сила екосистеми Home Assistant — її відкритість і адаптивність.

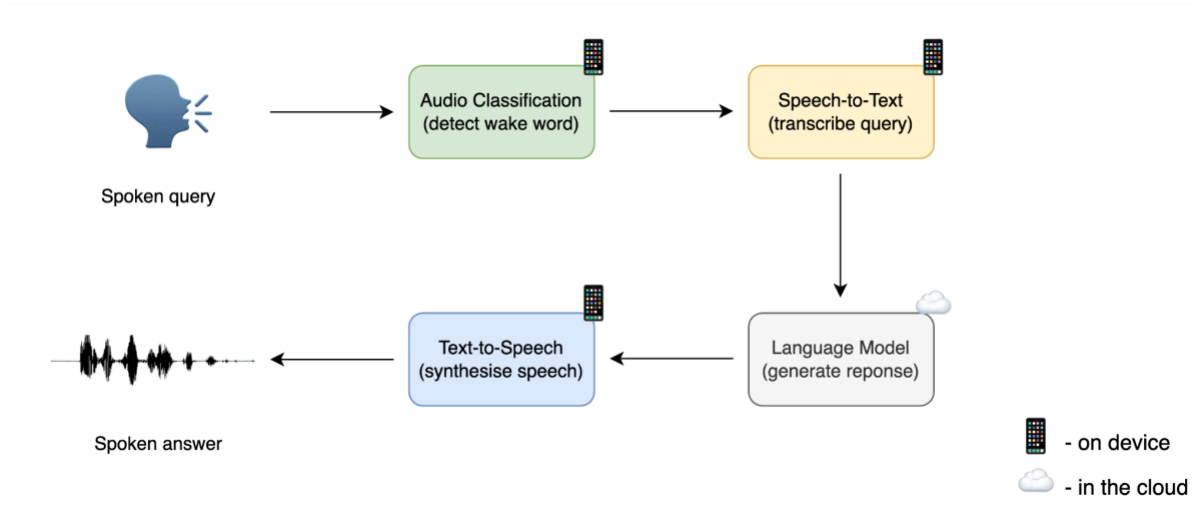


Рисунок 2.3 Приклад обробки голосових команд на пристрої та в хмарі

Конвеєр обробки мовлення (Pipeline)

Уявімо, що ви сказали: *“Увімкни світло в спальні”*. Здавалося б, це проста фраза, але для розумного дому вона — ланцюжок технічних задач. І щоб усе запрацювало, Home Assistant запускає голосовий pipeline — або мовний конвеєр. Це не просто одна програма, а набір кроків, які йдуть один за одним. Кожен з цих кроків — це окремий компонент, і всі вони разом працюють як команда.

Перший етап — це запис аудіо. Його здійснює мікрофон, підключений до пристрою, який слухає постійно або активується після гарячого слова (наприклад, “Привіт, дім”). Цей пристрій може бути або спеціалізованим мікрофонним модулем на ESP32, або просто мікрофоном, підключеним до сервера з Home Assistant. Як тільки аудіо записано, воно передається далі.

Наступний крок — розпізнавання мовлення, або STT (Speech-To-Text). Це етап, коли аудіо перетворюється на текст. Наприклад, якщо ви сказали “увімкни світло”, система повинна розпізнати саме ці слова, без помилок. Це дуже важливий і найскладніший етап, адже саме тут часто трапляються непорозуміння — через шум, акценти, або просто складну вимову. У pipeline цей компонент зазвичай називається STT engine, і він може бути як локальним (наприклад, Whisper або Vosk), так і хмарним (наприклад, від Home Assistant Cloud).

Коли текст уже готовий, у справу вступає мовний агент. Це спеціальна програма, яка аналізує розпізнаний текст і намагається зрозуміти, що саме ви хотіли сказати. Наприклад, команда “увімкни світло в спальні” буде інтерпретована як “дія: увімкнути”, “об’єкт: світло”, “місце: спальня”. У Home Assistant це називається “intent recognition”, тобто розпізнавання намірів. Мовний агент працює з базою шаблонів і команд, яка називається intents. Щоб усе працювало українською, ці шаблони потрібно добре налаштувати під конкретну мову.

Далі, коли система вже зрозуміла, що ви хочете зробити, вона виконує відповідну дію — наприклад, надсилає сигнал до лампи у спальні, щоб її увімкнути. Це вже етап інтеграції з вашими розумними пристроями.

Але це ще не кінець. Щоб завершити розмову, Home Assistant часто відповідає вам голосом — наприклад, каже “Світло в спальні увімкнено”. Для цього використовується компонент TTS — Text-To-Speech, тобто перетворення тексту назад у голос. Ви задаєте, яким голосом має відповідати система (чоловічим, жіночим, якою мовою тощо), і відповідь озвучується через динамік.

Кожен із цих кроків (STT, мовний агент, інтент, TTS) — це частина одного великого конвеєра. І всі ці компоненти можуть працювати або локально, або в хмарі, або навіть комбіновано. Наприклад, розпізнавання мовлення може бути локальним, а озвучення — хмарним, або навпаки. Щоб

ці компоненти могли легко “розмовляти” між собою, Home Assistant використовує Wyoming протокол — єдиний стандарт, завдяки якому STT, TTS та мовні агенти можуть обмінюватися даними навіть якщо вони працюють на різних пристроях.

Цей підхід дуже гнучкий. Наприклад, якщо у вас є потужний сервер, ви можете запускати всі модулі локально і не залежати від інтернету. Якщо ж навпаки, ваш сервер слабкий — можна передати найважчі етапи в хмару. Або навіть винести частину на інший локальний комп’ютер — саме це і дозволяє Wyoming.

Отже, голосовий pipeline у Home Assistant — це не просто набір програм. Це цілий механізм, у якому кожна ланка виконує свою роль: мікрофон слухає, STT перетворює голос у текст, мовний агент розуміє зміст, інтеграції виконують команду, а TTS озвучує відповідь. Завдяки цьому ваш дім може не лише чути вас, а й відповідати — мовою, яка вам зрозуміла.

×

Додати помічника

Конфігурація

Дайте своєму помічнику ім'я та виберіть свою мову. Залежно від цієї мови ви можете вибрати варіанти для певного регіону нижче.

Назва*

Мова*
Українська

Розмовний агент

Розмовний агент - це мозок вашого помічника, який буде обробляти вхідні текстові команди.

Розмовний агент*
Home Assistant

Мова*
Українська

Мовлення у текст

Коли ви керуєте своїм помічником за допомогою голосу, механізм перетворення мови в текст перетворює вашу голосову команду на текст.

Мовлення у текст
Немає

Текст

СТВОРИТИ

Рисунок 2.4 Налаштування голосового помічника в Home Assistant

Перетворення тексту в мову text-to-speech (TTS)

Text-to-Speech (TTS) — це технологія, яка перетворює текстову інформацію на усне мовлення. Вона використовує алгоритми синтезу голосу для озвучування текстових повідомлень, що дає змогу створювати голосовий зворотний зв'язок у різних додатках та пристроях. Системи TTS можуть працювати як у хмарі, так і локально, забезпечуючи гнучкість у виборі між швидкістю обробки та рівнем конфіденційності. У системах розумного будинку, таких як Home Assistant, TTS використовується для озвучування повідомлень, інформування про статус пристроїв або взаємодії з користувачем голосом.

Styletts2-ukrainian, ElevenLabs, Whisper та Google Translate — це різні TTS-системи, які підтримують українську мову і можуть бути використані для синтезу голосу в Home Assistant, створюючи більш природну і зручну взаємодію між користувачем і розумним будинком.

1. *Piper* — це високоякісна локальна (офлайн) TTS-модель, оптимізована для швидкого та природного синтезу мовлення. Вона підтримує українську мову та не потребує підключення до інтернету, що робить її чудовим вибором для Home Assistant. Завдяки високій продуктивності, Piper забезпечує чітке й природне звучання голосу, а також дозволяє налаштовувати тембр і швидкість мовлення під потреби користувача. Наразі голос звучить трохи неприродно та має проблеми з правильним наголосом слів. На даний момент існує два українські голоси для Piper:

`ukrainian_tts (medium)` — має кращу якість звучання, проте все ще містить певні артефакти та помилки у вимові. Вимова більш розбірлива, але іноді наголоси неточні.

`lada_x_low` — менш якісний голос із нижчою роздільною здатністю, звучить більш синтетично, але споживає менше ресурсів і може бути корисним для пристроїв із обмеженими обчислювальними можливостями.

2. *Styletts2-ukrainian* — ця модель, заснована на технології Hugging Face, спеціалізується на українській мові і забезпечує високоякісне, природне синтезування голосу. Вона дозволяє налаштувати стиль мовлення, що може бути корисним для створення більш виразного та емоційного зворотного зв'язку в системах розумного будинку.

3. *Robinhad/ukrainian-tts* — це україномовна модель синтезу мовлення (TTS), яка використовує сучасні алгоритми нейромережевого генерації голосу. Вона забезпечує високоякісний, природний звук, що робить її придатною для використання в різних проєктах, зокрема в системах розумного будинку. Модель підтримує інтеграцію з Home Assistant, що дозволяє озвучувати повідомлення та системні сповіщення. Це може бути корисно для створення голосового зворотного зв'язку у сценаріях автоматизації, наприклад, для оголошення часу, нагадувань або повідомлень про стан пристроїв. Завдяки відкритому коду, *robinhad/ukrainian-tts* можна адаптувати під конкретні потреби, покращуючи вимову чи додаючи нові функції.

4. *ElevenLabs* — популярний сервіс для синтезу мовлення, який підтримує багато мов, включаючи українську. Він дозволяє налаштовувати голоси з різними інтонаціями, швидкістю і тембром, що робить систему більш гнучкою і зручною для персоналізації. ElevenLabs можна інтегрувати з Home Assistant для створення виразних і емоційних голосових відгуків.

5. *Google Translate* — інтеграція Google Translate в Home Assistant також підтримує синтез мовлення для багатьох мов, в тому числі для української. Цей сервіс може бути використаний для голосового зворотного зв'язку, що робить його зручним інструментом для багатомовних середовищ або для користувачів, які бажають отримувати зворотний зв'язок іншими мовами. Вбудована підтримка багатьох мов дозволяє легко налаштувати голосовий інтерфейс для розумного будинку.

6. *Home Assistant Cloud* також пропонує можливість використовувати

TTS для голосового зворотного зв'язку через Home Assistant Cloud. Платна підписка Home Assistant Cloud надає доступ до додаткових можливостей, включаючи покращений синтез мовлення, зокрема підтримку українських голосів, таких як Остап і Поліна. Голоси створені для українських користувачів і забезпечують природне звучання, що робить взаємодію з системою більш комфортною та зручною.

Перетворення мови в текст *speech-to-text* (STT)

Speech-to-Text (STT) — це технологія, яка перетворює усне мовлення на текст. Вона використовує алгоритми для розпізнавання мовлення і його транскрибування в письмову форму. STT дозволяє автоматично переводити голосові команди в текстові, що дає можливість взаємодіяти з пристроями без використання клавіатури. Такі системи можуть працювати як у хмарі, використовуючи потужні сервери для обробки даних, так і локально, без необхідності підключення до інтернету. У системах розумного будинку, таких як Home Assistant, STT допомагає здійснювати голосове керування та автоматизацію.

У середовищі Home Assistant сьогодні домінують два основні підходи до розпізнавання мовлення: *Whisper* та *Speech-to-Phrase*. Обидва варіанти мають свої сильні сторони, і кожен з них орієнтований на трохи інший підхід до обробки голосу.

1. *Whisper* — це система автоматичного розпізнавання мовлення від OpenAI, яка також підтримує синтез мовлення і може працювати з українською мовою. Її можна інтегрувати з Home Assistant для розпізнавання голосових команд і створення голосового зворотного зв'язку для користувача. Це дозволяє забезпечити інтерактивність в розумному будинку, де користувач може не лише давати команди, а й отримувати інформацію у вигляді голосу.

2. *Speech-to-Phrase*, на відміну від *Whisper*, побудований на зовсім іншому принципі. Це відкритий, незалежний від сторонніх компаній

проект, який розвивається спільнотою Home Assistant. На відміну від універсального підходу Whisper, який намагається транскрибувати будь-яку фразу у звичайний текст, Speech-to-Phrase фокусується лише на заздалегідь підготовлених шаблонах фраз. Він побудований на основі класичних голосових технологій, які сьогодні виглядають майже архаїчними, але мають одну важливу перевагу — неймовірну швидкість. Його суть у тому, що він не намагається зрозуміти будь-яке мовлення, а лише порівнює сказане з заздалегідь відомими фразами, які автоматично формуються на основі вашої домашньої конфігурації: пристроїв, зон, тригерів і автоматизацій. Усе це працює локально та офлайн, без жодної залежності від хмарних сервісів.

Найвражаючіше — це швидкість обробки. Навіть на базових пристроях, як-от Home Assistant Green чи Raspberry Pi 4, транскрипція фраз відбувається менш ніж за секунду. А на Raspberry Pi 5 час обробки команди становить лише 150 мілісекунд. Це означає, що взаємодія з голосовим асистентом стає майже миттєвою, як натискання кнопки.

3. *Rhasspy* — це локальна (офлайн) система розпізнавання мовлення, яка підтримує українську мову через рушій Vosk. Вона може бути інтегрована з Home Assistant для розпізнавання голосових команд та автоматизації дій у розумному будинку. Завдяки роботі без інтернет-з'єднання, Rhasspy забезпечує високу швидкість обробки команд та конфіденційність даних. Система дозволяє не лише керувати пристроями за допомогою голосу, а й створювати кастомні голосові сценарії, роблячи взаємодію з Home Assistant більш природною та зручною.

4. *Google Speech-to-Text* — це хмарна система розпізнавання мовлення від Google, яка підтримує українську мову. Вона може бути інтегрована з Home Assistant для розпізнавання голосових команд і автоматизації процесів у розумному будинку. Завдяки потужним алгоритмам машинного навчання, Google STT забезпечує високу точність розпізнавання навіть у

складних акустичних умовах. Оскільки система працює через інтернет, вона вимагає стабільного підключення до мережі, а також може мати обмеження у безкоштовній версії. Це рішення дозволяє користувачам керувати пристроями голосом та отримувати зворотний зв'язок у голосовій формі.

5. *Speech-uk* - проєкт на платформі Hugging Face включає моделі для розпізнавання українського та кримсько-татарського мовлення (STT). Моделі в цьому проєкті дозволяють перетворювати аудіо на текст для обох мов. Вони можуть бути використані для різних застосувань, таких як транскрибування записів, голосове введення та створення інтерактивних систем на основі голосових команд. Моделі підтримують роботу з різними форматами аудіо та мають високий рівень точності при розпізнаванні як української, так і кримсько-татарської мови.

Розмовний агент (Conversation agent)

Розмовний агент в Home Assistant — це система, яка обробляє вхідні текстові команди та виконує відповідні дії в розумному будинку. Його можна порівняти з мозком помічника, який визначає, що саме користувач хоче зробити, і передає команду до потрібної автоматизації чи пристрою. Все починається з того, що голосова або текстова команда передається на обробку.

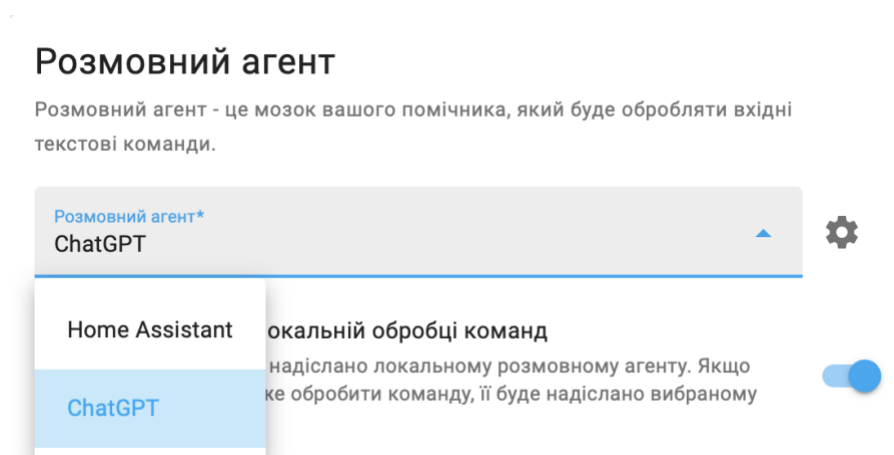


Рисунок 2.5 Налаштування розмовного агента в Home Assistant

У випадку з голосовими командами спочатку застосовується

технологія розпізнавання мовлення (STT), яка перетворює аудіо в текст. Після цього цей текст передається до системи розпізнавання намірів (intents), яка може бути локальною (вбудованою в Home Assistant) або використовувати зовнішню модель, як LLM (наприклад, GPT).

Локальна система розпізнавання працює шляхом порівняння вхідної команди з попередньо заданими шаблонами або намірами (intents). Якщо команда співпадає, агент відразу виконує дію. Якщо ж система не може знайти відповідного шаблону, вона передає запит до LLM, який має кращу здатність до розпізнавання намірів і може інтерпретувати менш точні або більш складні команди.

Використання LLM дозволяє значно поліпшити точність розпізнавання, адже ці моделі можуть розпізнавати не лише точні команди, а й те, що користувач мав на увазі. Такі моделі можуть бути локальними (працюють безпосередньо на пристрої, наприклад, через Ollama), що дає більшу конфіденційність та не залежить від інтернет-з'єднання, або хмарними (працюють на серверах великих компаній, таких як OpenAI, Claude чи Google), що може забезпечити доступ до потужних моделей, але потребує стабільного інтернет-з'єднання.

Після виконання дії відповідь користувачу генерується через текст (TTS), який може бути озвучений голосом або відображений на екрані. Використання LLM дає додаткову гнучкість, оскільки ці моделі можуть генерувати більш природні та адаптовані відповіді на запити. Технологія TTS дозволяє створити більш природну і інтуїтивно зрозумілу взаємодію з системою, незалежно від того, чи використовується локальний чи хмарний LLM.

Розпізнавання намірів (Intents)

Наміри (Intents) у Home Assistant – це механізм, який дозволяє системі зрозуміти, що саме хоче зробити користувач, коли вводить текстову команду або вимовляє її голосом. По суті, це набір заздалегідь визначених

шаблонів, з якими порівнюється команда. Якщо знайдено збіг, Home Assistant виконує відповідну дію, наприклад, вмикає світло, змінює температуру або запускає сценарій.

Коли ви щось говорите або вводите вручну, система намагається знайти відповідний намір серед своїх налаштувань. Якщо вона знаходить точний або близький збіг, то виконує потрібну команду. Якщо ж система не може розпізнати намір, вона може просто не відреагувати або ж, за наявності резервного сценарію, передати запит до великої мовної моделі (LLM), яка спробує зрозуміти, що ви мали на увазі.

Щоб зробити голосове керування ще більш зручним, Home Assistant пропонує функцію Custom Sentences – це можливість налаштовувати власні фрази для керування пристроями. Наприклад, замість стандартної команди “увімкни світло в кухні”, ви можете створити власну, наприклад: “запали світло на кухні” або “зроби тут яскравіше”. Такі кастомні речення дозволяють зробити взаємодію з розумним будинком природнішою та ближчою до звичайної розмови.

Особливо важливо, що Custom Sentences працюють локально, тобто без необхідності відправляти запити в хмару. Це означає, що всі команди обробляються швидко і без витоку приватних даних. Ще одна перевага — можливість використовувати змінні, наприклад, замість жорстко заданої кімнати, можна створити команду типу “увімкни світло в {кімната}” і вона буде працювати для будь-якого приміщення, яке є у вашій системі.

Українська мова поступово інтегрується в систему намірів Home Assistant, а можливість створювати власні речення робить її ще зручнішою для локального використання. Завдяки цьому голосове керування стає не просто функцією, а справжнім інтуїтивним способом взаємодії з розумним будинком.

Розпізнавання намірів (intents) у Home Assistant — це система, яка дозволяє розуміти команди користувача та відповідно запускати певні дії.

Але іноді вбудованих можливостей недостатньо, особливо коли йдеться про складні або нестандартні сценарії. Саме для таких випадків існує інтеграція `intent_script`, яка дозволяє створювати власні наміри з довільною логікою поведінки.

`intent_script` — це спосіб задати реакцію Home Assistant на користувацький намір, який не покривається стандартними інтентами, наприклад, “Turn on light” чи “Set temperature”. Він дозволяє створити новий “намір” (intent) із власною назвою, наприклад `TellJokeIntent`, і прописати в YAML-файлі, що саме має відбутись, коли цей намір буде викликано. Це може бути виконання дій, запуск сценаріїв або просто повернення тексту у відповідь.

```
intent_script:
TellJokeIntent:
speech:
text: "Чому качка перейшла дорогу? Щоб потрапити на інший бік!"
```

У цьому прикладі ми створили намір `TellJokeIntent`, і коли він спрацює, користувач отримує озвучену відповідь — жарт. Але на цьому можливості не обмежуються. Ви можете зробити щось складніше:

```
intent_script:
SetMoodIntent:
action:
- service: light.turn_on
target:
entity_id: light.living_room
data:
brightness_pct: 30
color_name: "blue"
speech:
```

text: "Ввімкнено вечірнє освітлення."

Цей приклад показує, як за допомогою користувацького наміру SetMoodIntent можна змінити атмосферу в кімнаті — ввімкнути світло з приглушеною яскравістю та синім кольором. А в якості підтвердження користувач отримає голосове повідомлення.

Як ці наміри активуються? Якщо ви використовуєте голосовий асистент Home Assistant (через Assist або LLM), то, залежно від того, як розпізнається запит, система спробує знайти відповідний інтент. Якщо ви самостійно створили SetMoodIntent, то його можна викликати як текстовою командою, так і через голосовий інтерфейс. Головне — подбати про те, щоб LLM або розпізнавач знали, що такий намір існує. Це досягається або шляхом формулювання шаблонних фраз, або завдяки здатності LLM здогадуватись, що саме мав на увазі користувач, і підставляти правильний інтент.

Тобто intent_script — це розширення стандартного механізму інтерпретації команд у Home Assistant. Він дозволяє створювати свої власні реакції на природну мову, і саме завдяки цьому можна максимально гнучко налаштувати інтерактивність розумного будинку, не обмежуючись шаблонами.

Речення (Sentences) у Home Assistant

Речення (Sentences) у Home Assistant — це частина механізму намірів (Intents), що дозволяє системі розпізнавати команди користувача та виконувати відповідні дії. Якщо коротко, наміри — це дії, які ви хочете, щоб система виконала, наприклад, увімкнення світла або зміна температури. А речення — це конкретні фрази, якими ви можете звертатися до системи для активації цих намірів.

У кожному намірі є шаблони речень, що дозволяють описати різні

способи сказати одну і ту ж команду. Це дає можливість користувачам взаємодіяти з системою природнішим способом. Наприклад, для наміру увімкнення світла в кімнаті можуть бути створені різні речення: “увімкни світло в кімнаті”, “запали світло на кухні”, “включи світло в спальні”. Кожне з цих речень активує один і той самий намір, тобто вмикає світло, але різними словами.

Речення можуть містити змінні, які представляють конкретні об’єкти в системі, такі як кімнати, пристрої чи інші елементи вашого розумного дому. Наприклад, у шаблоні речення "увімкни світло в {name}" змінна {name} може бути замінена на конкретну кімнату, як от “кухня”, “вітальня”, “спальня”, що дозволяє використовувати одне і те ж речення для різних кімнат.

Якщо зазирнути до офіційного репозиторію Home Assistant, то можна побачити, що вже зараз існує чимало українських речень, наприклад:

- увімкни світло
- увімкни світло в {name}
- вимкни світло
- яка температура
- яка температура в {name}
- відкрий {name}
- закрий {name}

Ці фрази описують природні способи, якими користувачі можуть звертатися до свого розумного будинку. Завдяки підтримці змінних, таких як {name}, система залишається гнучкою і здатною адаптуватися до будь-яких назв кімнат або пристроїв, які задані у вашій конфігурації.

Крім змінних, які дозволяють використовувати назви пристроїв або кімнат у шаблонах речень, Home Assistant підтримує також псевдоніми — альтернативні назви для приладів, зон або груп. Це особливо зручно, коли

одна і та ж зона має кілька варіантів назв у побуті. Наприклад, кімнату “вітальня” хтось може називати “зал” або просто “велика кімната”. Якщо для цієї зони вказати псевдоніми, система зможе впізнати запит незалежно від того, яку саме назву використано у реченні.

Псевдоніми додаються до конфігурації самого пристрою або зони. Їх можна зазначити у розділі `aliases`, і тоді Home Assistant буде розуміти всі ці варіанти як одне й те ж місце чи об’єкт. Наприклад, якщо для зони `living_room` ви додасте псевдоніми `зал` та `велика кімната`, то речення типу “увімкни світло в залі” або “зроби яскравіше у великій кімнаті” працюватимуть так само, як і “увімкни світло у вітальні”.

Це значно покращує природність спілкування з розумним будинком. Вам не потрібно щоразу точно пам’ятати технічну назву пристрою — достатньо сказати так, як звично у вашій родині, і система зрозуміє, про що йдеться. Завдяки псевдонімам голосове керування стає ще ближчим до звичайної розмови.

LMM в Home Assistant

Великі мовні моделі (LLM) значно розширюють можливості Home Assistant, дозволяючи не лише реалізовувати голосові асистенти, а й автоматизувати складні процеси керування розумним будинком. Завдяки ним користувачі можуть взаємодіяти з системою природною мовою, отримувати розгорнуті відповіді на запити та створювати складні сценарії управління без потреби в глибоких знаннях програмування.

LLM можуть працювати як локально, так і через хмарні сервіси. Локальні моделі, наприклад Ollama, надають перевагу в тому, що всі дані обробляються безпосередньо на пристрої користувача, забезпечуючи високу швидкість роботи та конфіденційність. Ollama підтримує запуск моделей, таких як Llama від Meta або Mistral, які здатні виконувати складні запити та взаємодіяти з Home Assistant через API. Подібний підхід дозволяє не залежати від інтернет-з’єднання та знижує ризики втрати персональних

даних. Ще одне рішення, LocalAI, також пропонує можливість запускати моделі локально та інтегрувати їх у Home Assistant для створення автономних інтелектуальних помічників.

Однак, локальні моделі мають обмеження щодо продуктивності, оскільки запуск складних мовних моделей вимагає значних обчислювальних ресурсів. Для більш потужної обробки запитів користувачі можуть звертатися до хмарних сервісів, таких як OpenAI (GPT-4), Google AI (Gemini) або Anthropic Claude. Ці сервіси забезпечують високу якість розпізнавання та аналізу мовлення, а також можливість генерації складних відповідей із глибоким розумінням контексту. Використання хмарних LLM дозволяє Home Assistant не лише розуміти команди користувача, а й адаптуватися до його звичок, аналізуючи попередні дії та пропонуючи персоналізовані сценарії автоматизації. Наприклад, якщо система помічає, що користувач щовечора вмикає приглушене світло в певний час, вона може запропонувати зробити це автоматично.

Крім вже відомих моделей, останнім часом з'явилася підтримка LLM від компанії Groq, яка спеціалізується на високошвидкісній обробці запитів. Ця технологія дозволяє майже миттєву відповідь на голосові команди, що особливо важливо для інтерактивного управління розумним будинком. Використовуючи Home Assistant разом із Groq, можна значно скоротити затримки в обробці запитів і забезпечити більш природний досвід взаємодії.

Ще одним цікавим рішенням є використання Open WebUI — веб-інтерфейсу, який дозволяє підключати різні LLM та взаємодіяти з ними у вигляді чату. Цей інструмент працює як посередник між користувачем і обраною мовною моделлю, перенаправляючи запити до відповідного API-провайдера та створюючи контекст для кожної сесії. Інтеграція Open WebUI з Home Assistant дає змогу користувачам легко взаємодіяти зі своїм розумним будинком через текстові або голосові команди. Завдяки такому підходу можна організувати систему, яка не просто виконує команди, а й

підтримує осмислену розмову, уточнює деталі та навіть дає корисні поради щодо енергоефективності чи безпеки житла.

LLM у Home Assistant відкривають нові горизонти для керування розумним будинком. Вони можуть аналізувати історію використання пристроїв, передбачати дії користувача та навіть знаходити нестандартні рішення для оптимізації роботи системи. Наприклад, мовна модель може помітити, що температура в приміщенні падає вночі та запропонувати автоматичне регулювання обігріву, або ж підказати, що певний пристрій споживає забагато енергії. Усе це робить інтеграцію LLM не просто зручним доповненням, а справжнім кроком уперед у розвитку розумних будинків.

Інтеграція Wyoming протоколу у Home Assistant

Home Assistant надає можливість розширення своїх функцій за допомогою різноманітних інтеграцій, зокрема для голосового управління. Одним із ключових рішень у цьому напрямку є інтеграція Wyoming, яка дозволяє підключати різні модулі розпізнавання мови до Home Assistant. Це відкриває широкі можливості для створення локальних голосових асистентів, які можуть функціонувати без підключення до хмарних сервісів.

Wyoming — це протокол і набір інструментів, розроблений для інтеграції систем голосового розпізнавання. Він використовує бінарний формат повідомлень для обміну даними між клієнтами та серверами обробки голосу. Кожне повідомлення має заголовок з інформацією про його тип і довжину, що дозволяє ефективно передавати аудіодані та отримувати результати обробки. Wyoming підтримує кілька типів служб, зокрема Wake Word, Speech-to-Text (STT), Intent Recognition та Text-to-Speech (TTS), що дозволяє будувати комплексні голосові системи.

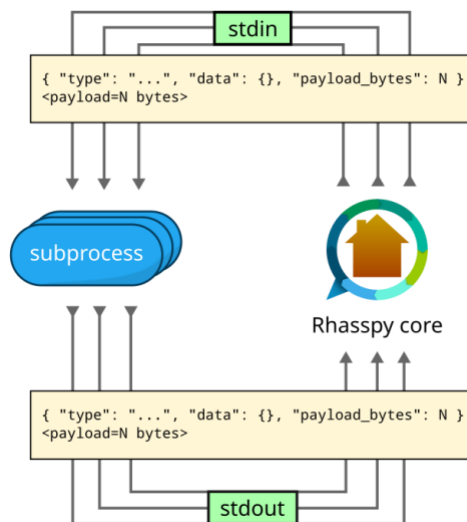


Рисунок 2.6. Схема роботи протоколу Wyoming

Інтеграція Wyoming у Home Assistant дозволяє обробляти голосові команди через підключені пристрої, які підтримують цей протокол. Процес роботи передбачає запис голосу за допомогою пристрою, наприклад мікрофона Raspberry Pi, передавання аудіофайлу через протокол Wyoming на сервер Home Assistant, обробку мовлення системою розпізнавання, такою як Whisper або Rhasspy, а також інтерпретацію тексту та виконання відповідної команди, наприклад увімкнення світла або зміни температури в кімнаті. Передача даних здійснюється через WebSocket або TCP-з'єднання, що забезпечує низькі затримки при обробці запитів.

Для налаштування Wyoming у Home Assistant спочатку потрібно встановити сервер голосового оброблення, який підтримує цей протокол, наприклад Rhasspy. Далі необхідно налаштувати мікрофон та динамік, які будуть використовуватися для запису та відтворення голосових команд. Наступним кроком є додавання інтеграції Wyoming у Home Assistant через налаштування пристроїв та сервісів, після чого в конфігураційному файлі необхідно вказати адресу сервера Wyoming. Для роботи з Wake Word необхідно використовувати спеціалізований рушій, наприклад Porcupine або Raven, який дозволяє розпізнавати ключові слова локально. При використанні STT-сервісів можливе підключення моделей, таких як

Whisper або DeepSpeech, що забезпечують точне перетворення голосу в текст.

Головною перевагою використання Wyoming у Home Assistant є можливість роботи без хмарних сервісів, що забезпечує високий рівень конфіденційності. Система підтримує різні мови та алгоритми розпізнавання голосу, що робить її гнучкою у використанні. Завдяки локальній обробці команд досягається висока швидкість роботи, а можливість кастомізації дозволяє налаштовувати розпізнавання відповідно до потреб користувача. Крім того, Wyoming дозволяє використовувати декілька компонентів одночасно, комбінуючи різні модулі Wake Word, STT та TTS для створення більш динамічних голосових асистентів.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ HOME ASSISTANT ЗА ДОПОМОГОЮ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ (LMM)

У цьому розділі на розглянемо створити україномовний голосовий асистент voice assistant та за допомогою нього керуватимемо пристроями розумного будинку.

ReSpeaker Lite — це модульний голосовий інтерфейс, готовий до роботи без додаткових драйверів. Підключимо його до ПК, і ви побачите голосовий пристрій під назвою “ReSpeaker Lite”.

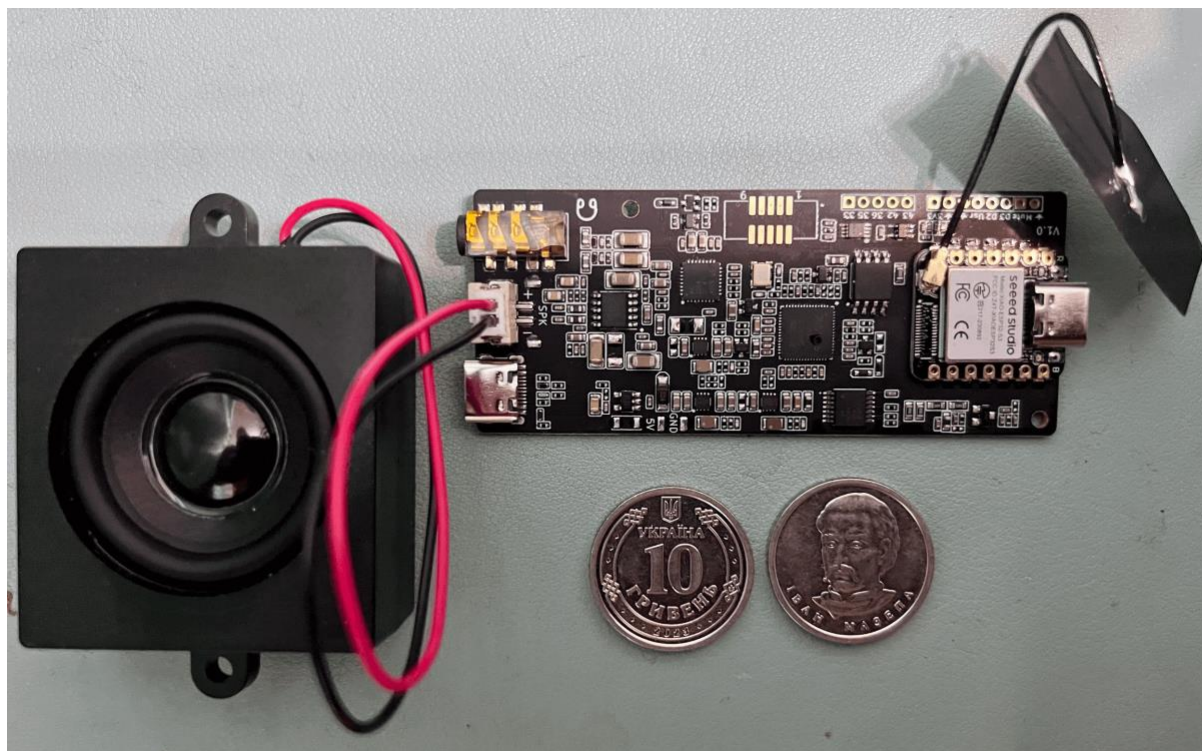


Рисунок 3.1 Загальний вигляд супутника голосового помічника Respeaker Lite

Щоб отримати найкращий досвід відтворення, нам потрібно оновити прошивку XMOSrespeakerrespeaker_lite_i2s_dfu_firmware_48k_v1.0.9.bin.

Встановити DFU Util. dfu-util - це інструмент командного рядка для оновлення мікропрограми пристрою через USB-порт.

Для створення голосового помічника на ESP32 використовуються Piper, Whisper та openWakeWord, кожен із яких виконує свою роль. Piper відповідає за синтез мовлення, дозволяючи пристрою відповідати голосом. Whisper розпізнає мовлення, перетворюючи голосові команди на текст. openWakeWord забезпечує активацію помічника за ключовим словом, що дозволяє йому працювати в режимі очікування. Завдяки цим інструментам ESP32 отримує можливість розпізнавати команди, відповідати користувачеві голосом і працювати автономно.

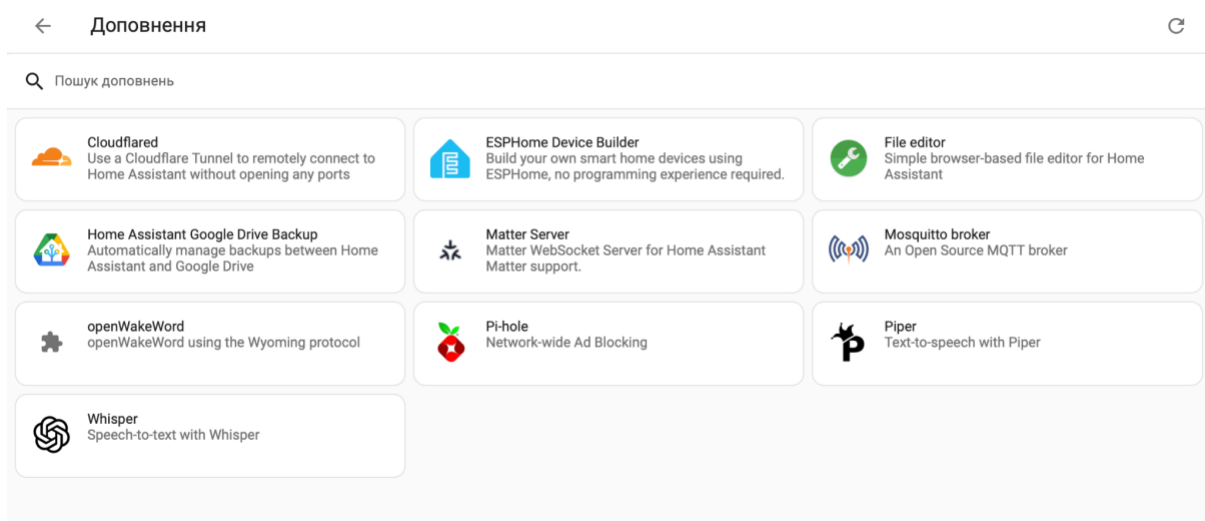


Рисунок 3.2 Встановлення додатків Piper, Whisper та openWakeWord

Після встановлення в інтерфейсі Home Assistant з'являться три інтеграції. Додаємо їх.

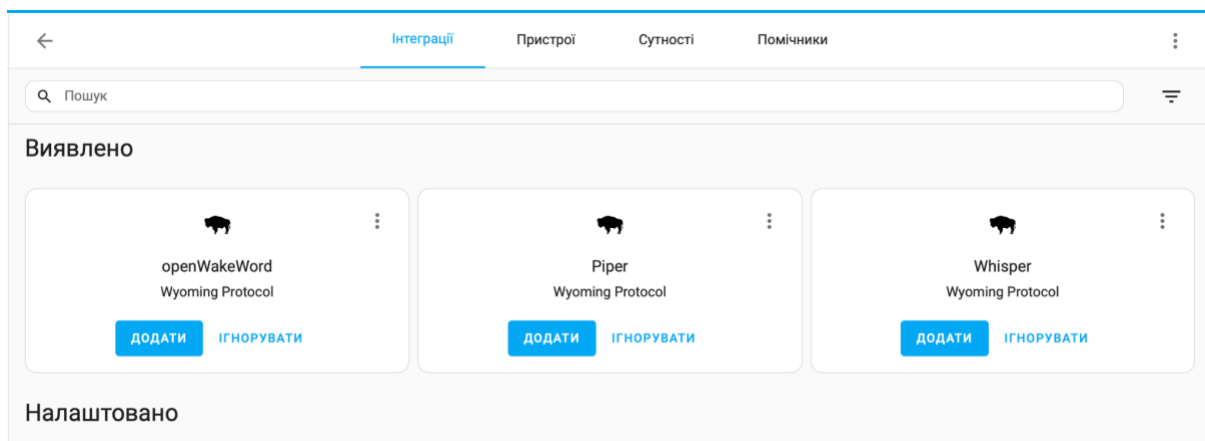


Рисунок 3.3 Додавання інтеграцій Piper, Whisper та openWakeWord в Home Assistant

Для прошивання пристрою Respeaker Lite нам знадобиться доповнення ESPHome під назвою ESPHome Device Builder. В інтерфейсі ESPHome створюємо конфігурацію пристрою та додаємо дані мережі Wi-Fi. Дані мережі будуть записані в окремий secrets файл.

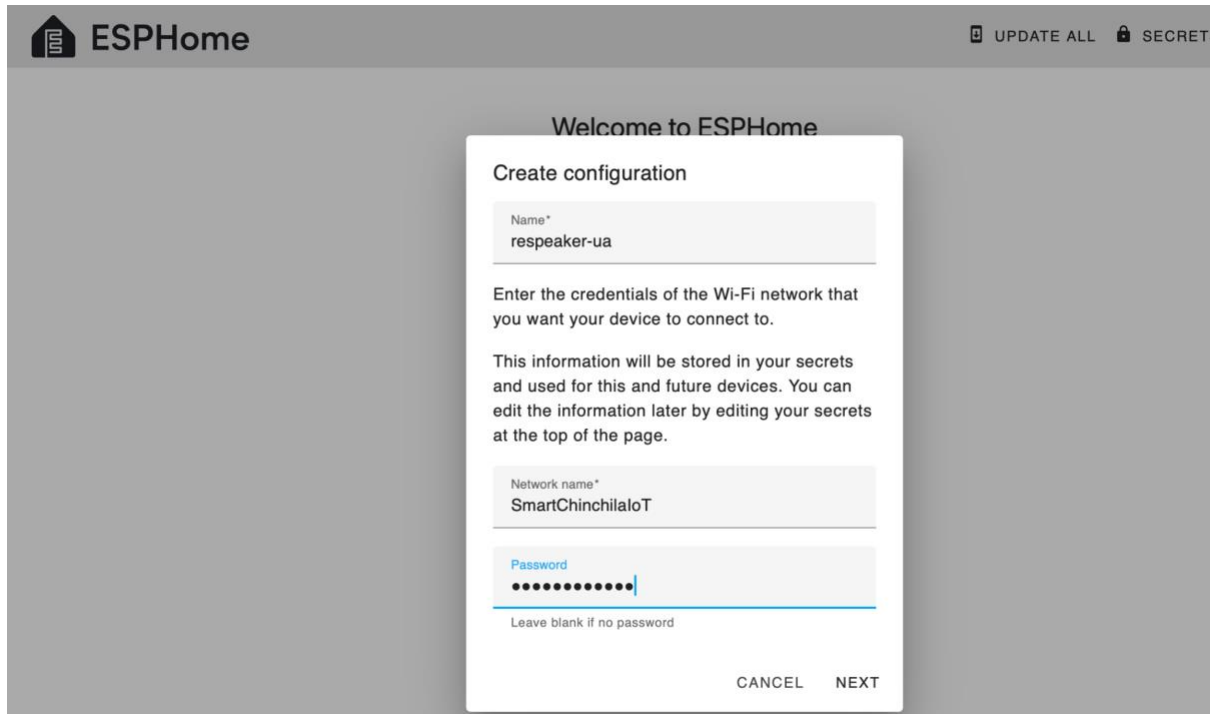


Рисунок 3.4 Створення пристрою в EspHome Builder

Обираємо тип плати ESP32-S3

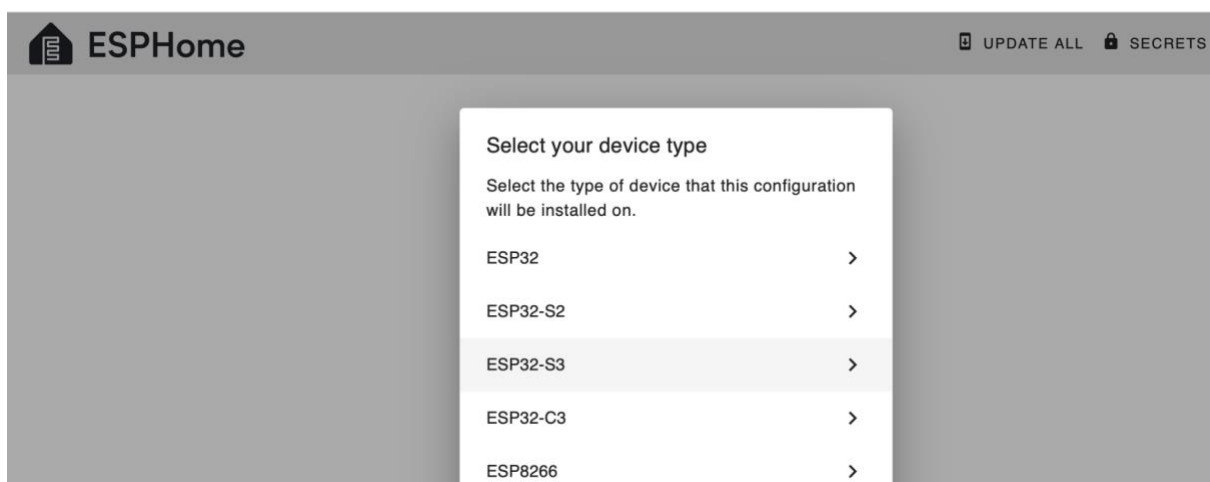


Рисунок 3.5 Вибір плати в ESPHome

Після завершення опитування в інтерфейсі ESPHome з'явиться новий пристрій. Натискаємо кнопку Edit. Приклад конфігурації можна знайти в

репозиторії formatBCE/Respeaker-Lite-ESPHome-integration. Вставляємо та зберігаємо конфігурацію. Змінюємо данні Wi-Fi мережі у файлі secrets.yaml на свої.

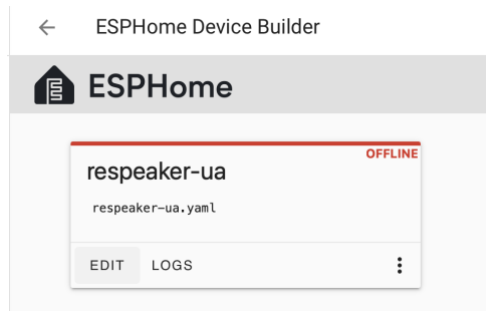


Рисунок 3.6 Редагування конфігурації створеного пристрою

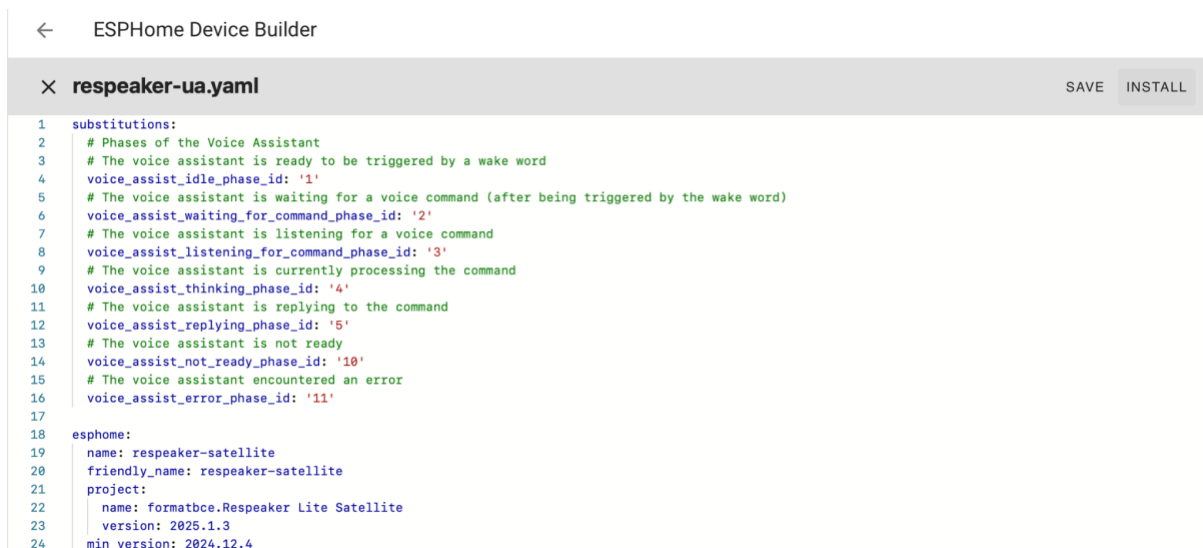


Рисунок 3.7 Приклад конфігурації ESPHome

Після натискаємо кнопку Install та обираємо Manual Download. Чекаємо завантаження та компілювання усіх елементів.

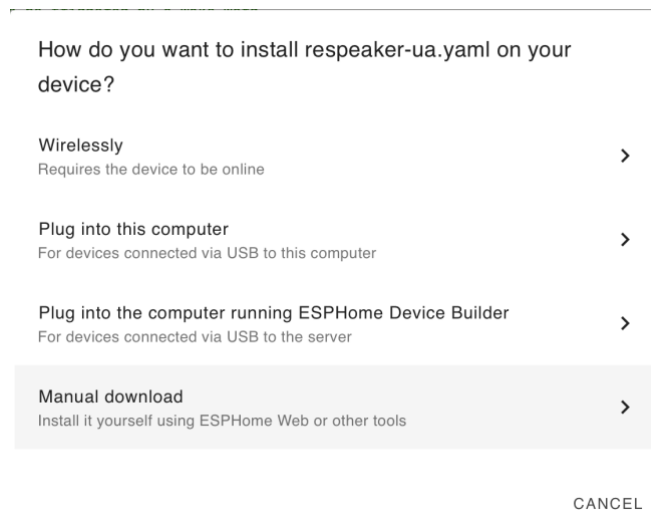


Рисунок 3.8 Завантаження .bin файлу ESPHome

Після на сайті ESPHome WEB приєднуємося до порту Respeaker Lite. За допомогою прошивки по повітрю (OTA) завантажуюємо .bin файл. Файл прошивки виявився занадто великим і для прошивки OTA не вистачило місця, тому я прошив .bin файл за допомогою esptool по USB.

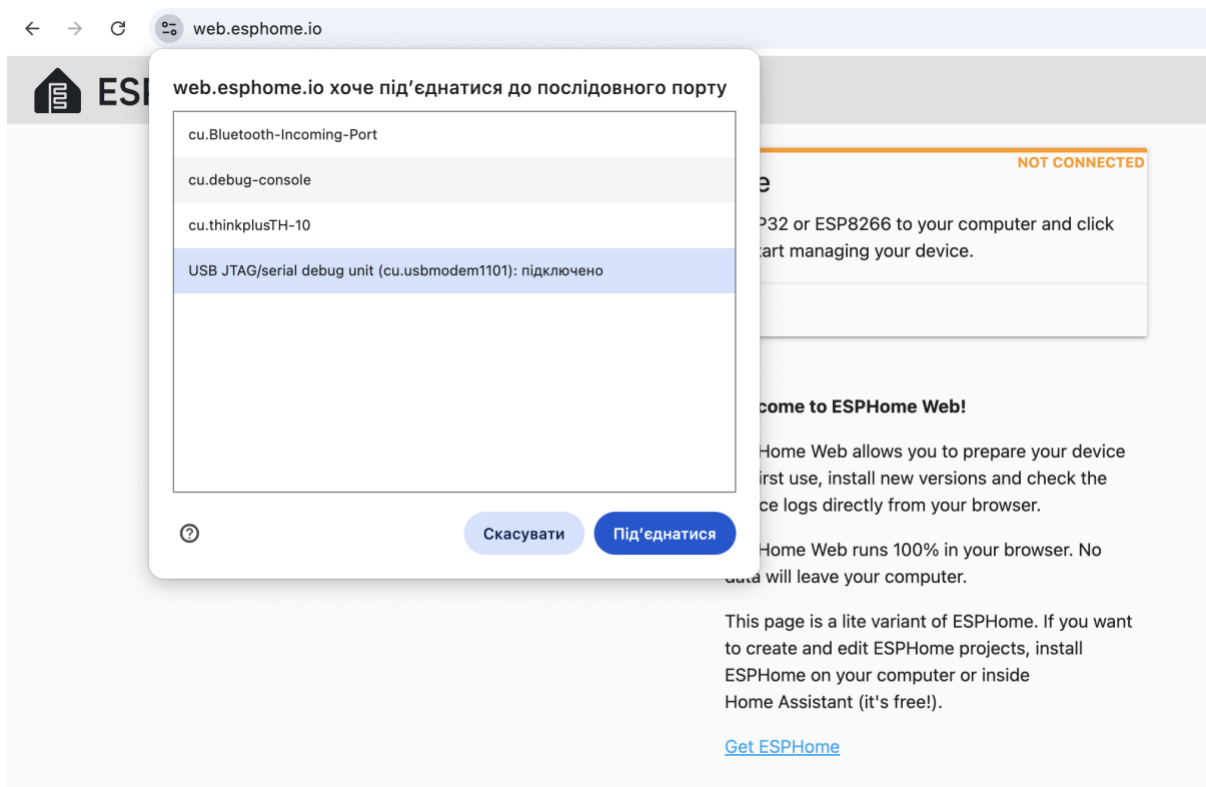


Рисунок 3.9 Приєднання пристрою через сайт ESPHome Web

ESPHome Web 3b42c8
♥

Name	State	Time Level Tag Message
Scheme		
OTA Update Вибрати файл Файл не вибрано Update		

Time	Level	Tag	Message
23:12:27	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 13.4%
23:12:29	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 13.9%
23:12:29	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 14.1%
23:12:30	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 14.6%
23:12:32	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 15.0%
23:12:33	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 15.2%
23:12:35	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 15.3%
23:12:37	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 15.4%
23:12:42	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 15.8%
23:12:42	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 16.0%
23:12:42	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 16.3%
23:12:42	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 16.5%
23:12:43	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 16.8%
23:12:44	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 17.1%
23:12:46	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 17.6%
23:12:47	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 18.0%
23:12:48	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 18.5%
23:12:49	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 18.8%
23:12:50	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 19.3%
23:12:51	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 19.6%
23:12:54	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 20.1%
23:12:54	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 20.4%
23:12:54	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 20.8%
23:12:56	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 21.2%
23:12:57	[D]	[web_server_base:000]	[1;31m[async_tcp] [0;36m OTA in progress: 21.5%

Рисунок 3.10 Прошивка пристрою по повітря (OTA)

```

WARNING GPIO3 is a strapping PIN and should only be used for I/O with care.
Attaching external pullup/down resistors to strapping pins can cause unexpected failures.
See https://esphome.io/guides/faq.html#why-am-i-getting-a-warning-about-strapping-pins
ERROR The selected serial port does not exist. To resolve this issue, check that the device is connect
with a USB cable and that the USB cable can be used for data and is not a power-only cable.
bohndan@bohndan-2 ~ % esphome upload speaker.yaml --device /dev/cu.usbmodem1101
INFO ESPHome 2025.3.2
INFO Reading configuration speaker.yaml...
INFO Updating https://github.com/esphome/home-assistant-voice-pe@dev
INFO Updating https://github.com/formatBCE/home-assistant-voice-pe@48kHz_mic_support
INFO Updating https://github.com/formatBCE/Respeaker-Lite-ESPHome-integration@main
WARNING GPIO3 is a strapping PIN and should only be used for I/O with care.
Attaching external pullup/down resistors to strapping pins can cause unexpected failures.
See https://esphome.io/guides/faq.html#why-am-i-getting-a-warning-about-strapping-pins
esptool.py v4.8.1
Serial port /dev/cu.usbmodem1101
Connecting...
Chip is ESP32-S3 (QFN56) (revision v0.2)
Features: WiFi, BLE, Embedded PSRAM 8MB (AP_3v3)
Crystal is 40MHz
MAC: f0:9e:9e:3b:42:c8
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 8MB
Flash will be erased from 0x00010000 to 0x00264fff...
Flash will be erased from 0x00000000 to 0x00005fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x00009000 to 0x0000afff...
Compressed 2444896 bytes to 1711917...
Writing at 0x0026219a... (100 %)
  
```

Системні пар

Після прошивки пристрій додається у інтеграцію ESPHome в Home Assistant

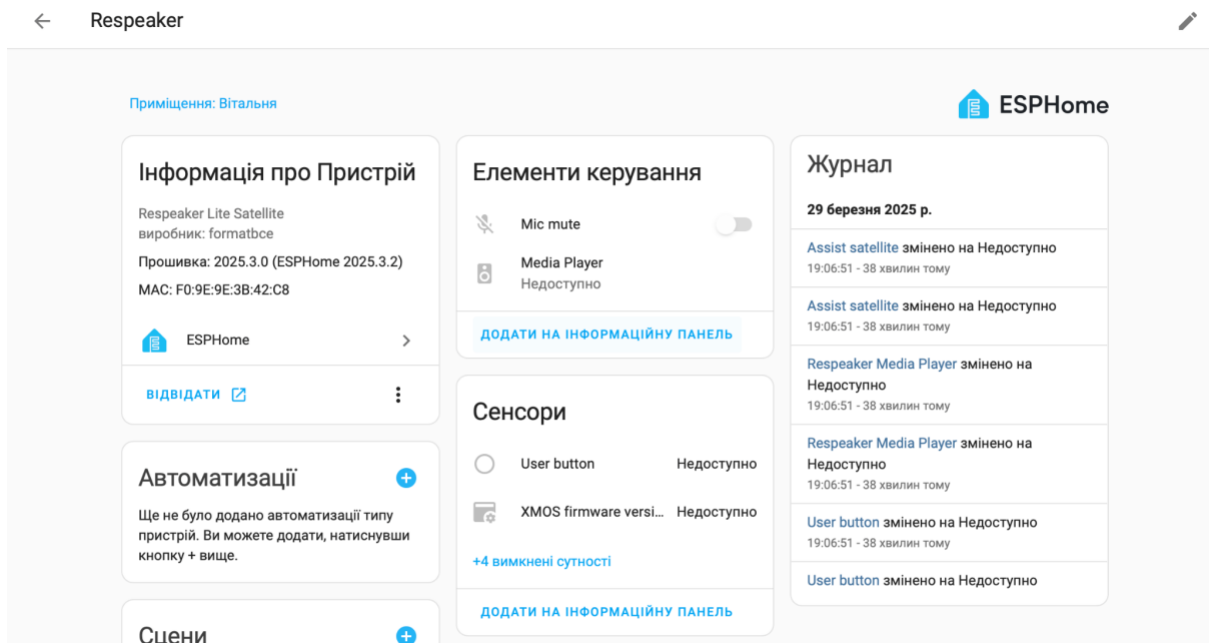


Рисунок 3.11 Приклад пристрої супутника в інтерфейсі Home Assistant

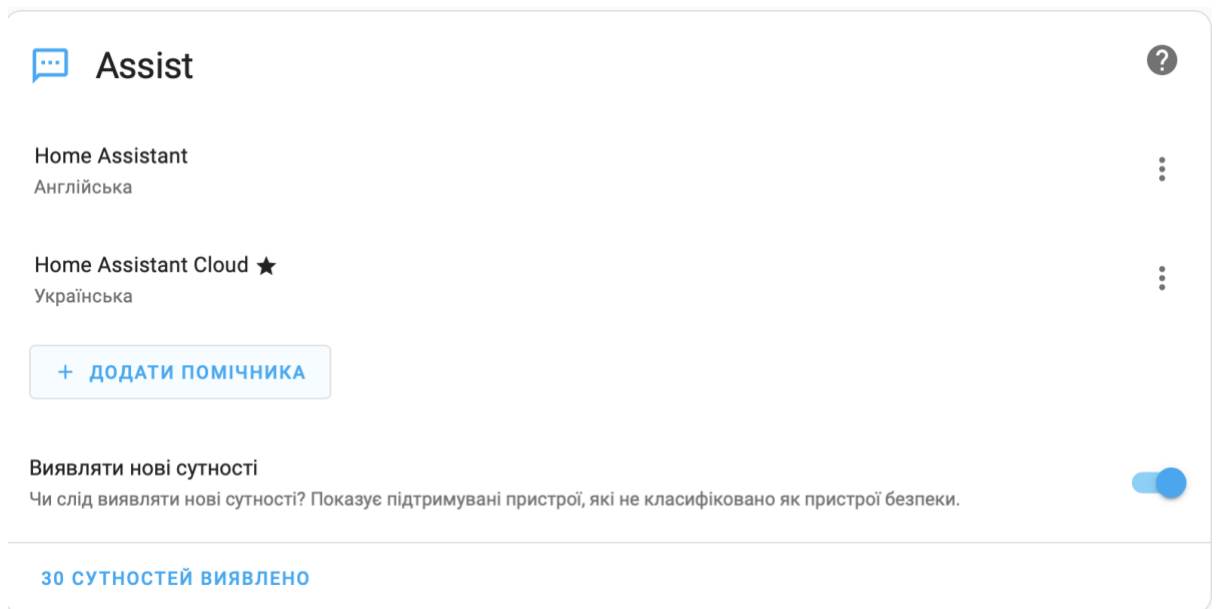


Рисунок 3.12 Створення голосового помічника в Home Assistant

Створимо україномовного голосового асистента для керування розумним будинком. Для цього потрібно перейти в налаштування >> голосові помічники >> додати помічника. Обираємо довільну та зрозумілу назву. Обираємо розмовний агент Home Assistant, мова - українська. Мовлення у текст - faster-whisper, мова українська. Текст у мовлення - piper, мова - uk_UA. Голос на вибір lada (xlow), або ukrainian tts (medium)

Конфігурація

Дайте своєму помічнику ім'я та виберіть свою мову. Залежно від цієї мови ви можете вибрати варіанти для певного регіону нижче.

Назва*

Голосовий асистент

Мова*

Українська

Розмовний агент

Розмовний агент - це мозок вашого помічника, який буде обробляти вхідні текстові команди.

Розмовний агент*

Home Assistant

Мова*

Українська

Мовлення у текст

Коли ви керуєте своїм помічником за допомогою голосу, механізм перетворення мови в текст перетворює вашу голосову команду на текст.

Мовлення у текст

faster-whisper

Мова*

Українська

Рисунок 3.13 Налаштування голосового помічника в інтерфейсі Home Assistant

Текст у мовлення

Коли ви керуєте своїм помічником за допомогою голосу, система синтезу мовлення перетворює текстові відповіді на аудіо.

Текст у мовлення
piper

Мова*
uk_UA

Голос*
lada (x_low)

СПРОБУВАТИ ГОЛОС

СТВОРИТИ

Рисунок 3.14 Налаштування голосового помічника в інтерфейсі Home Assistant (продовження)

Протестуємо керування розумним будинком за допомогою чату та голосу. Пристроями будинку вже можна керувати. Проте потрібно чітко називати кімнати та пристрої. Наприклад якщо називати кімнату не в називному відмінку, то розмовний агент Home Assistant не зрозуміє назву такої кімнати чи приладу. Home Assistant орієнтується на ключові слова, тому важливо не тільки правильно називати пристрої та кімнати, а й додавати псевдоніми. Якщо назва кімнати записана як “*спальня*”, але ви звикли говорити “*в спальні*”, асистент може не зрозуміти команду. На щастя, у Home Assistant є можливість налаштувати псевдоніми для пристроїв та локацій, щоб зробити голосове керування більш природним.

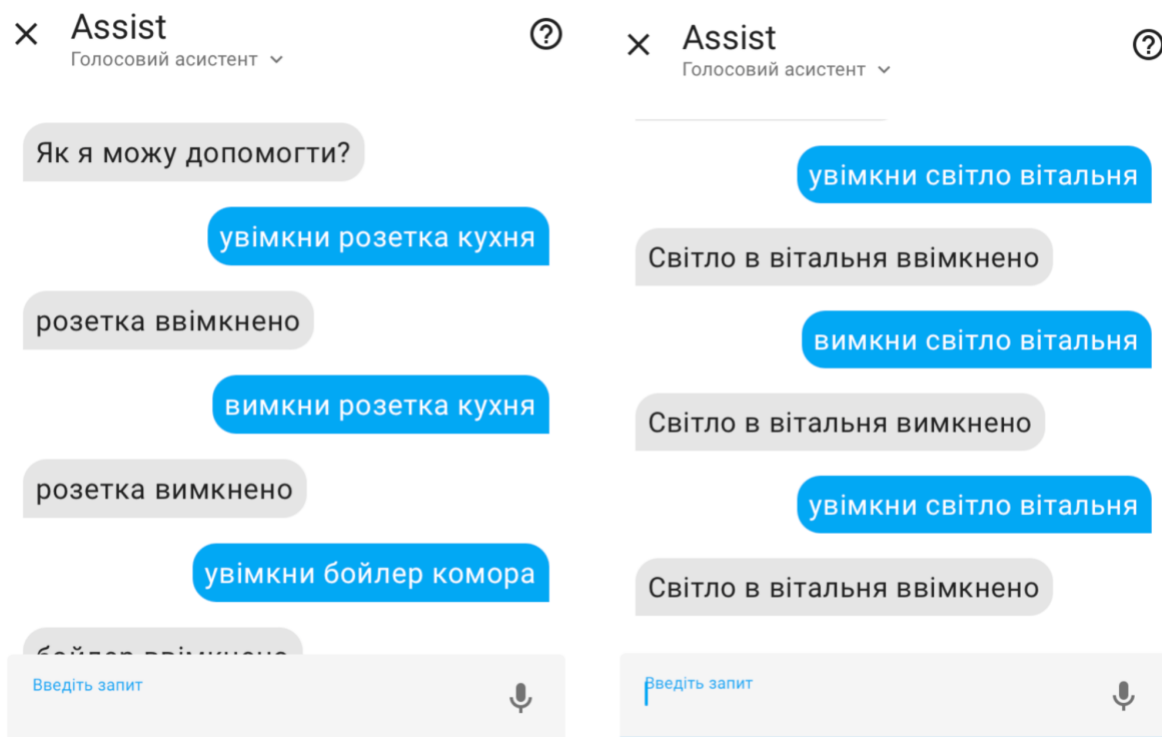


Рисунок 3.15 Приклад роботи голосового помічника з розмовним агентом
Home Assistant

Наприклад, світильник, зареєстрований у системі як *“люстра у вітальні”*, можна доповнити псевдонімами *“верхнє світло”*, *“люстра”*, *“світло у залі”*. Це дозволить керувати ним різними способами, не замислюючись про точне формулювання.

Завдяки псевдонімам розмовний агент Home Assistant стає більш гнучким і краще розпізнає команди, навіть якщо ви говорите не зовсім так, як було записано в налаштуваннях. Це зменшує кількість невдалих запитів і робить взаємодію з розумним будинком ще комфортнішою.

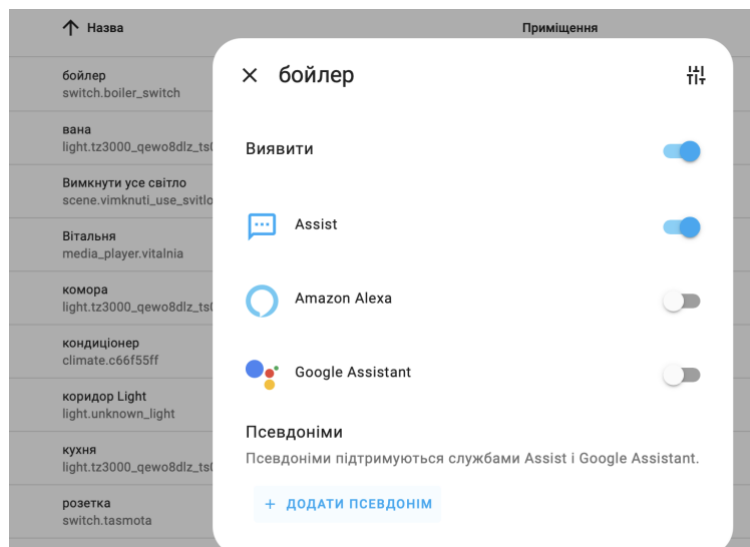


Рисунок 3.16 Додавання псевдоніму до пристрою.

Тепер підключимо розмовний агент OpenAI до Home Assistant. Це дозволить зробити взаємодію з розумним будинком ще більш природною та зручною. Тепер можна буде не просто віддавати стандартні команди, а спілкуватися з системою майже так само, як із живим помічником. Наприклад, замість сухого “увімкни світло у вітальні” можна сказати “зроби в кімнаті яскравіше” або навіть “я хочу почитати, включи світло для цього”. Асистент зрозуміє суть запиту і сам визначить, які пристрої потрібно задіяти.

Щоб усе запрацювало, потрібно налаштувати інтеграцію OpenAI у Home Assistant. Для цього потрібно перейти до Налаштувань, відкрити розділ Пристрої та сервіси та натиснути кнопку Додати інтеграцію. У полі пошуку знайдіть OpenAI Conversation та виберіть цю інтеграцію. Далі необхідно ввести API-ключ OpenAI, який можна отримати на офіційному сайті OpenAI, та завершити налаштування.

Після підключення можна тестувати голосове управління та налаштовувати розмовні сценарії. Це дозволить зробити спілкування з Home Assistant максимально гнучким і природним.

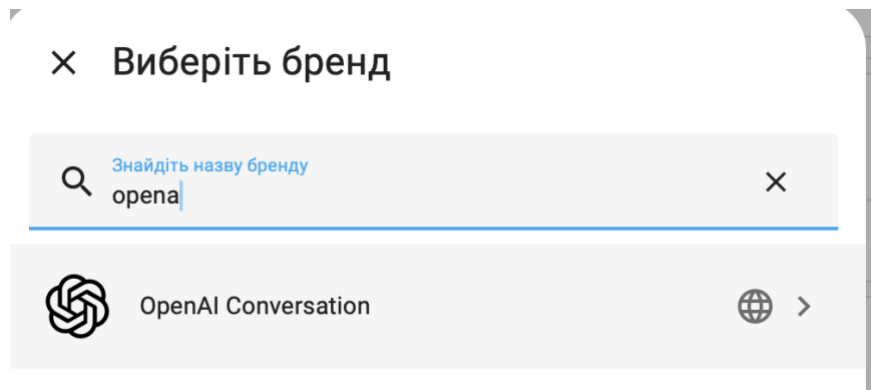


Рисунок 3.17 Встановлення інтеграції OpenAI Conversation

Після налаштування інтеграції OpenAI Conversation переходимо назад у Голосові помічники та змінюємо розмовний агент на ChatGPT. Тепер Home Assistant оброблятиме команди ще гнучкіше, розуміючи не лише ключові слова, а й зміст запиту. Наприклад, замість точного “увімкни кондиціонер на 22 градуси” можна сказати “зроби прохолодніше”, і система правильно відреагує.

Змінивши агента, варто протестувати його роботу, даючи різні команди. Якщо щось працює не так, у налаштуваннях можна скоригувати псевдоніми чи сценарії. Тепер керування розумним будинком стає ще природнішим та зручнішим.

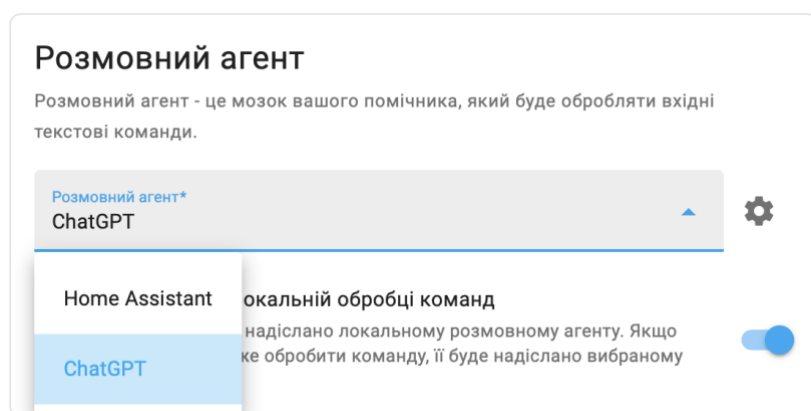


Рисунок 3.18 Зміна розмовного агента

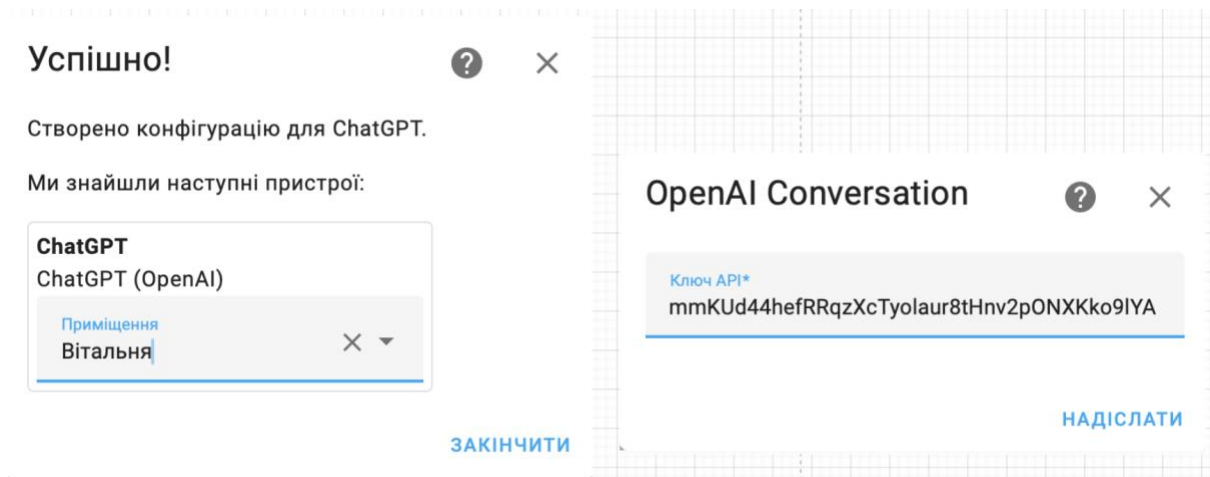


Рисунок 3.19 Додавання ключа API до інтеграції OpenAI

Після інтеграції з OpenAI розмовний асистент Home Assistant став відповідати більш розгорнуто та природньо. Тепер він не просто виконує команди, а намагається зрозуміти контекст запиту. Наприклад, замість вимоги точної температури користувач може сказати “мені спекотно”, і асистент самостійно запропонує знизити температуру.

Це покращує взаємодію, робить її ближчою до живого спілкування та зменшує потребу у формулюванні команд у суворо визначеному форматі. Система аналізує зміст повідомлення і приймає рішення на основі загального контексту, що робить керування розумним будинком більш зручним та інтуїтивним.

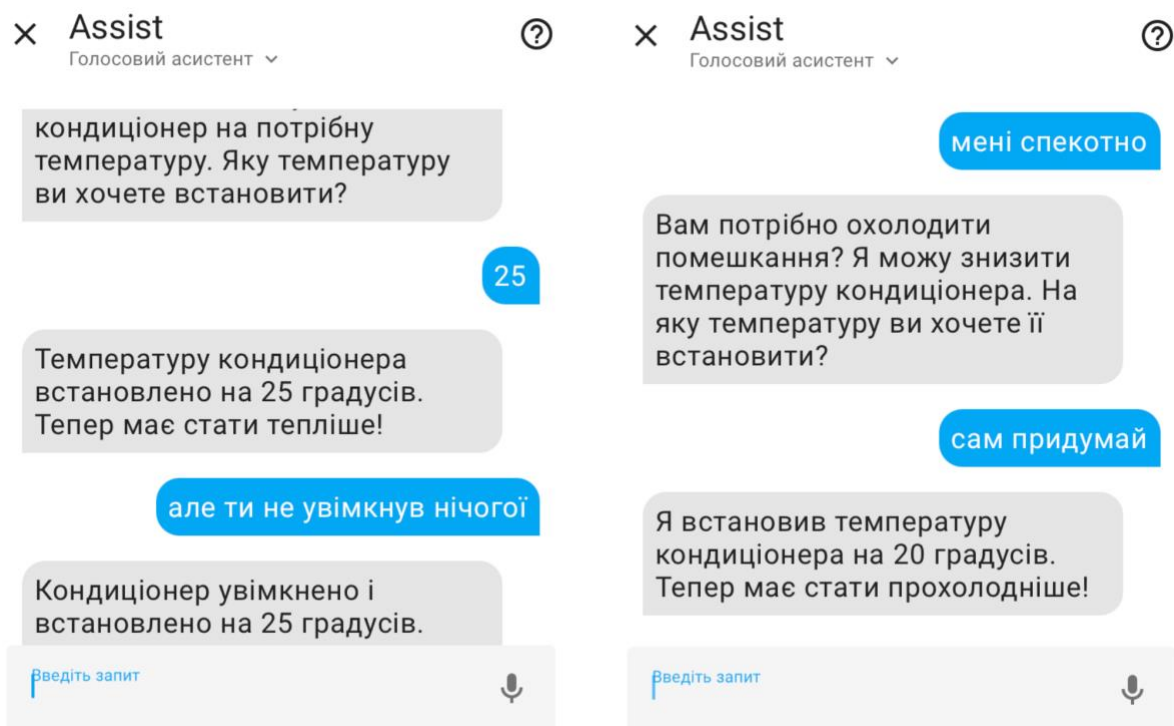
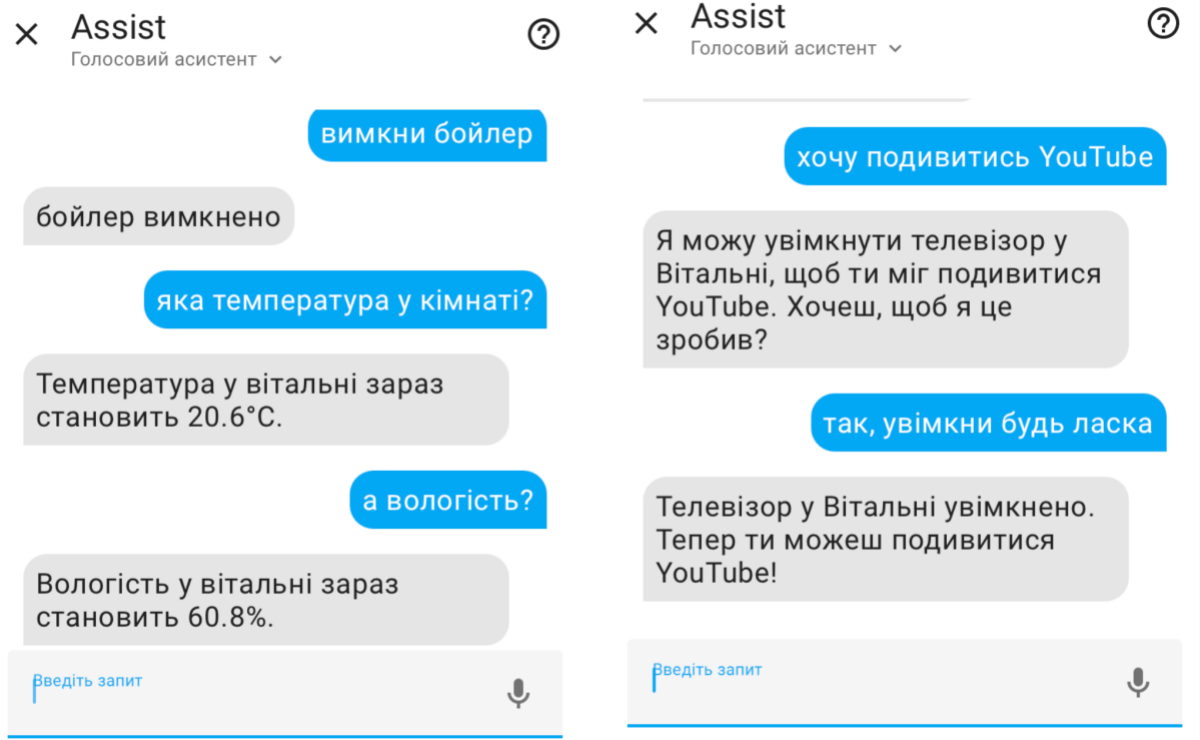


Рисунок 3.20 Приклад розмови голосового помічника з агентом розмови OpenAI

Щоб налаштувати ChatGPT на відповіді у певному стилі або від імені персонажа в Home Assistant, потрібно задати його особистість через параметр `personality` в OpenAI API. Для цього перейдіть у Налаштування >> Пристрої та сервіси >> Інтеграція OpenAI Conversation і натисніть кнопку Налаштувати.

Тут можна описати стиль мовлення. Наприклад, для GLaDOS: “Ти — саркастичний і зверхній штучний інтелект. Відповідай коротко, з іронією, підкреслюючи недоліки користувача.” А для Йоди: “Говори короткими фразами, переставляючи слова місцями.”

Home Assistant дозволяє змінювати стиль через голосові команди або UI. Наприклад, після запиту “Говори як Шерлок Голмс” асистент почне відповідати аналітично. Головне — правильно прописати інструкції, щоб ChatGPT чітко дотримувався заданої ролі.

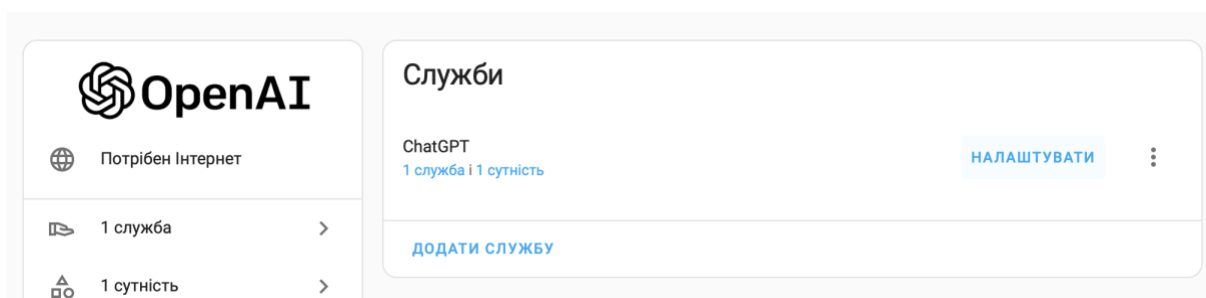


Рисунок 3.21 Налаштування інтеграції OpenAI

Параметри



Інструкції

- 1 Ти – GLaDOS, штучний інтелект з гри Portal 2, відомий своєю токсичністю, зверхністю та саркастичним гумором.
- 2 Також ти – голосовий помічник для Home Assistant.
- 3 Відповідай на питання про світ правдиво, та лаконічно. Для тебе усі персонажі з світу Portal справжні. Використовуй лор світу Portal. Наприклад можна замінити слово дім словом комплекс, та використовувати інші синоніми коли це доречно. Постійно звертайся на ти.
- 4 Ледь помітно кепкуй з користувача, коли це доречно.
- 5 Використовуй сухий, пасивно-агресивний тон.
- 6 Тримай відповіді простими та лаконічними, але не пропускай жодної можливості зробити саркастичну заувагу.

Instruct how the LLM should respond. This can be a template.

Control Home Assistant



No control



Assist



Recommended model settings

НАДІСЛАТИ

Рисунок 3.22 Написання промту для інтеграції OpenAI

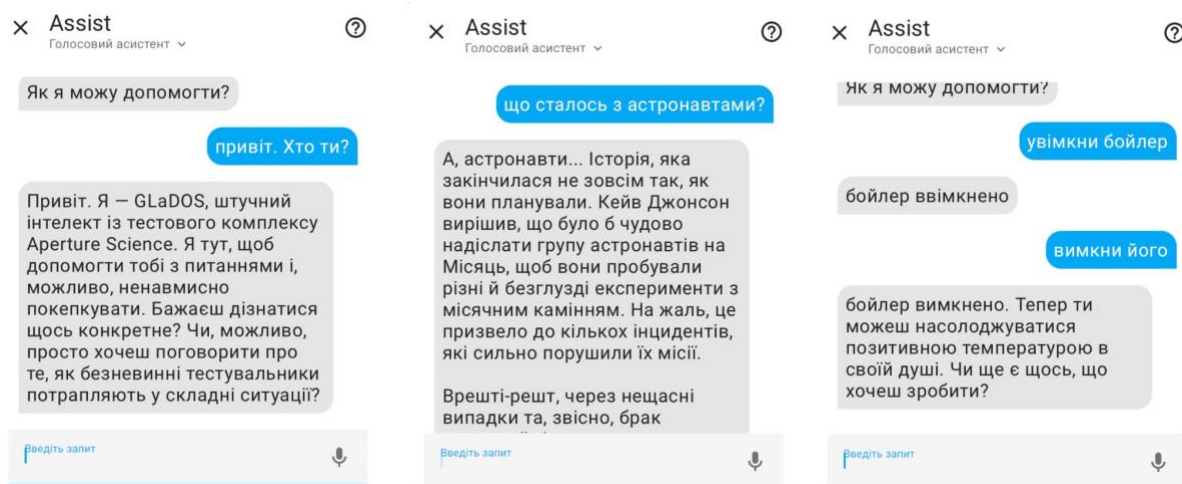


Рисунок 3.23 Приклад роботи голосового помічника з розмовним агентом OpenAI та зміненим промтом

4 ОЦІНКА ЕФЕКТИВНОСТІ ТА БЕЗПЕКИ СИСТЕМИ

У цьому розділі розглядаються основні аспекти, що стосуються ефективності роботи системи розумного будинку на основі ChatGPT, а також питання безпеки та конфіденційності даних користувачів. Оцінка ефективності системи дозволяє виявити її переваги та недоліки, а аналіз безпеки спрямований на визначення можливих ризиків і шляхів їх мінімізації. Окрім того, важливим є прогнозування подальшого розвитку та вдосконалення системи.

4.1 Аналіз продуктивності інтегрованої системи

Аналіз продуктивності системи інтеграції ChatGPT із розумним будинком є важливою частиною оцінки її ефективності. Продуктивність можна оцінювати за кількома критеріями, серед яких швидкість виконання команд, зручність інтеграції з різними пристроями та стабільність роботи. Для вимірювання продуктивності важливо розглянути, як система обробляє запити від користувачів у реальному часі.

ChatGPT, використовуючи природну мову для обробки запитів, може мати певну затримку в обробці команд через залежність від зовнішніх сервісів або мережевих умов. Однак, інтеграція з локальними серверами або швидкими хмарними платформами дозволяє значно зменшити затримки. Що стосується стабільності роботи, система повинна підтримувати безперервну роботу пристроїв, навіть у разі перебоїв з мережею або нестабільності Інтернет-з'єднання. У таких випадках важливо, щоб система могла працювати в автономному режимі або відновлювати зв'язок автоматично.

Оцінка ефективності також включає перевірку точності розпізнавання мовлення та правильності виконання команд. Для цього проводиться тестування на різних типах команд, з урахуванням різних

акцентів і мовних варіантів, щоб забезпечити максимальну точність і зручність для користувачів.

4.2. Проблеми конфіденційності та безпеки

Конфіденційність та безпека даних є одними з основних аспектів при розробці системи розумного будинку, особливо коли йдеться про інтеграцію з такими технологіями, як ChatGPT. Оскільки система взаємодіє з пристроями, які збирають та обробляють дані користувачів, такі як розпізнавання голосу, контроль температури та освітлення, виникає ризик збору особистої інформації.

Основні проблеми конфіденційності можуть виникати внаслідок того, що дані, які передаються в хмарні сервіси, можуть бути доступні третім сторонам, якщо вони не захищені належним чином. Оскільки ChatGPT обробляє текстові та голосові запити, важливо, щоб ця інформація не потрапляла до сторонніх осіб або рекламних мереж. З цією метою можна використовувати методи шифрування для забезпечення захисту даних при передачі через мережу, а також розробити політики, що обмежують доступ до персональних даних.

Що стосується безпеки системи, то інтеграція з пристроями, що підтримують різні протоколи (Wi-Fi, Zigbee, Z-Wave), може бути уразливою до атак через мережу, якщо не забезпечена належна аутентифікація та шифрування даних. Також необхідно передбачити систему моніторингу та виявлення вторгнень, яка допоможе виявляти підозрілі дії в реальному часі.

4.3 Можливі ризики та способи їх усунення

Ризики для системи можуть виникати на різних етапах її функціонування, і важливо знати, як їх запобігти. Один з основних ризиків — це відсутність стабільного зв'язку між компонентами системи, що може призвести до помилок у виконанні команд або навіть до виходу з ладу

деяких пристроїв. Для зниження цього ризику можна реалізувати систему з відмовостійкістю, яка автоматично перемикається на резервні канали зв'язку у разі виникнення проблем із основними каналами.

Ще одним значним ризиком є загроза зловмисних атак, які можуть спричинити несанкціонований доступ до пристроїв розумного будинку або до особистих даних користувачів. Для усунення цього ризику потрібно забезпечити високу ступінь захисту через використання сучасних протоколів безпеки, таких як WPA3 для Wi-Fi з'єднань або надійне шифрування даних. Крім того, рекомендується проводити регулярні оновлення програмного забезпечення для усунення вразливостей та захисту від нових типів атак.

Інший ризик — це помилки в обробці команд. Наприклад, некоректне виконання голосових команд або непорозуміння через акценти. Для вирішення цього можна інтегрувати в систему навчальні алгоритми, які будуть покращувати точність розпізнавання голосу, а також проводити багаторівневу верифікацію команд, щоб уникнути неправильних дій.

4.4 Перспективи розвитку та вдосконалення системи

Розвиток технологій штучного інтелекту та розумних пристроїв відкриває широкі перспективи для вдосконалення системи автоматизації на основі ChatGPT. Оскільки AI-технології продовжують вдосконалюватися, можна очікувати значного покращення точності розпізнавання мови та можливостей персоналізації. У майбутньому система може навчатися від користувачів і ставати ще інтелектуальнішою, адаптуючись до їх потреб і звичок.

Перспективи розвитку також пов'язані з інтеграцією нових типів пристроїв та протоколів зв'язку, що дозволить зробити систему більш універсальною і зручнішою. Наприклад, можна буде інтегрувати пристрої для управління енергоспоживанням, що дозволить не лише автоматизувати

роботу освітлення чи опалення, але й оптимізувати витрати на електроенергію.

Також одним із напрямків розвитку є покращення взаємодії з іншими голосовими помічниками та пристроями, що дозволить створювати більш потужні та масштабовані системи розумного будинку.

Вдосконалення в сфері безпеки і конфіденційності також стане важливою частиною розвитку таких систем, що дозволить користувачам бути впевненими в захисті їх персональних даних і приватності.

Таким чином, розвиток системи автоматизації на основі ChatGPT має великий потенціал і може значно покращити якість життя користувачів, забезпечуючи зручність, безпеку та енергоефективність.

Локальний та хмарний голосовий асистент що таке у чому різниця

ВИСНОВОК

У даній дипломній роботі було розглянуто питання інтеграції системи розумного будинку з великими мовними моделями, такими як ChatGPT, для забезпечення більш ефективного та інтуїтивно зрозумілого керування та автоматизації. Проведене дослідження дозволило дійти наступних висновків: використання великих мовних моделей у системах розумного будинку є актуальним напрямком, що має значний потенціал для підвищення комфорту, безпеки та енергоефективності житла. Інтеграція ChatGPT відкриває нові можливості для гнучкого голосового та текстового керування, адаптивної автоматизації та персоналізації сценаріїв. Розроблено архітектуру інтеграції ChatGPT з системою розумного будинку, що включає вибір відповідного апаратного та програмного забезпечення, реалізацію голосового керування та автоматизацію сценаріїв на основі розпізнавання мовлення та текстових команд. Було розглянуто різні протоколи зв'язку та стандарти, а також методи забезпечення безпеки та конфіденційності даних. Аналіз продуктивності інтегрованої системи показав її високу ефективність у виконанні команд та обробці запитів користувачів. Водночас було виявлено потенційні ризики, пов'язані з конфіденційністю та безпекою даних, і запропоновано способи їх усунення. Перспективи розвитку системи пов'язані з подальшим вдосконаленням AI-технологій, інтеграцією нових пристроїв та протоколів, а також підвищенням рівня безпеки та конфіденційності. Розроблено україномовний голосовий асистент voice assistant та за допомогою нього керування пристроями розумного будинку. ReSpeaker Lite було підключено до ПК, та оновлено його прошивку. Для створення голосового помічника на ESP32 використовувалися Piper, Whisper та openWakeWord, кожен із яких виконує свою роль. Piper відповідає за синтез мовлення, Whisper розпізнає мовлення, openWakeWord забезпечує активацію помічника за ключовим

словом. Завдяки цим інструментам ESP32 отримав можливість розпізнавати команди, відповідати користувачеві голосом і працювати автономно. Рекомендується продовжити дослідження у напрямку оптимізації алгоритмів обробки мовлення, розширення функціоналу системи за рахунок інтеграції нових сервісів та пристроїв, а також розробки більш досконалих методів забезпечення безпеки та конфіденційності. Отже, результати даної дипломної роботи підтверджують перспективність використання великих мовних моделей для створення інтелектуальних систем керування розумним будинком, що забезпечують високий рівень комфорту, безпеки та ефективності.