

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

« __ » _____ 2023 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

**Програмно-апаратний комплекс розпізнавання
та інтеграції мови в побутові прилади**

Спеціальність 123 Комп'ютерна інженерія

123 – КМР.1 – 605.21710505

Студент

_____ В. Д. Веселовський
підпис

« __ » _____ 202__ р.

Керівник д-р техн. наук, проф.

_____ І. М. Журавська
підпис

« __ » _____ 202__ р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної магістерської роботи

Видано студенту групи 605м факультету комп'ютерних наук

Веселовський Владислав Дмитрович
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Програмно-апаратний комплекс розпізнавання та інтеграції мови в побутові прилади

Затверджена наказом по ЧНУ ім. Петра Могили від 03.11.2022 № 201.

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є: апаратне та програмне забезпечення розпізнавання голосових команд людини для здійснення керування побутовими приладами. Вхідними даними роботи є специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення.

4. Перелік питань, що підлягають розробці

- 1) огляд сучасних підходів та систем керування побутовими приладами;
- 2) аналіз переваг та недоліків існуючих систем;
- 3) розробка апаратної частини імплементації голосових команд до побутових приладів;
- 4) розробка програмної частини інтеграції голосових команд до побутових приладів;

5. Перелік графічних матеріалів

слайди презентації

6. Завдання до спеціальної частини

Проведення оцінки умов праці фахівців підприємства ФОП С. С. Гонтаренко, а саме аналіз виробничого приміщення, оцінка природного освітлення. Визначення рівня електричної та пожежної безпеки підприємства.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Д-р біол. наук, професор Григор'єва Л. І.	Кафедра екології Медичного інституту ЧНУ ім. Петра Могили	Спеціальна частина з охорони праці

Керівник роботи

д-р техн. наук, професор Журавська Ірина Миколаївна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Веселовський Владислав Дмитрович

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної магістерської роботи

Тема: Програмно-апаратний комплекс розпізнавання та інтеграції мови в побутові прилади

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	10.09.2022	24.09.2022	Виконано
2.	Огляд літератури за темою роботи	25.09.2022	08.10.2022	Виконано
3.	Складання календарного плану КМР	09.10.2022	23.10.2022	Виконано
4.	Аналіз предметної області	24.10.2022	07.11.2022	Виконано
5.	Розробка проєктних рішень	08.11.2022	22.11.2022	Виконано
6.	Моделювання	23.11.2022	09.12.2022	Виконано
7.	Конструювання АПЗ	10.12.2022	17.12.2022	Виконано
8.	Перевірка працездатності, тестування та апробація розробленого АПЗ,	18.12.2022	31.12.2022	Виконано
9.	Аналіз результатів тестування	01.01.2023	08.01.2023	Виконано
10.	Розробка керівництва користувача	09.01.2023	20.01.2023	Проведено
11.	Відгук керівника КМР	21.01.2023	01.02.2023	Виконано
12.	Оформлення КМР та презентації	02.02.2023	13.02.2023	Виконано
13.	Попередній захист	14.02.2023	14.02.2023	Проведено
14.	Рецензування	15.02.2023	16.02.2023	Виконано
15.	Завершення оформлення КМР та презентації	16.02.2023	24.02.2023	Виконано
16.	Захист кваліфікаційної роботи	25.02.2023	25.02.2023	Захищено

Розробив здобувач ВО Веселовський Владислав Дмитрович

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи зав. каф. КІ Журавська Ірина Миколаївна

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної магістерської роботи

«Програмно-апаратний комплекс розпізнавання та інтеграції мови в побутові прилади»

Студент гр. 605м Веселовський Владислав Дмитрович

Керівник: д-р техн. наук, професор Журавська Ірина Миколаївна

Актуальність теми кваліфікаційної роботи обумовлена швидкими темпами розвитку пристроїв, що працюють за концепцією «цифровий будинок». Ця концепція полягає в інтеграції пристроїв різного призначення та різних виробників у єдину систему. У більшості випадків таке об'єднання можливе за рахунок використання спільної системи керування, наприклад, на основі голосових команд.

Обробка голосових команд є складним процесом, що базується на методах аналізу аудіосигналу: амплітудно-часовий аналіз, спектральний аналіз тощо. Серед цих методів варто виділити найбільш новітній та сучасний нейронні мережі. При застосування цього методу необхідно мати готову, «навчену» модель нейронної мережі, що може виконувати ідентифікацію голосу, розпізнавання мови, аналізу емоцій тощо.

Об'єкт дослідження (розробки): процеси автоматизації керування побутовими приладами, що виконують свої функції через отримання голосових команд.

Предмет дослідження (розробки): методи та засоби створення систем автоматизованого керування побутовими приладами, що виконують свої функції тільки після ідентифікації голосових команд.

Мета: розробити програмно-апаратний комплекс розпізнавання мови та інтеграції голосових команд в побутові прилади.

Для досягнення поставленої мети було поставлено такі завдання:

- 1) оглянути сучасні системи керування побутовими приладами;
- 2) проаналізувати недоліки існуючих систем;

3) розробити апаратну частину імплементації голосових команд до побутових приладів;

4) розробити програмну частину відправки голосових команд до побутових приладів;

5) створити макетну модель для демонстрації роботи.

Кваліфікаційна робота містить: перелік скорочень, вступ, чотири розділи, висновок, перелік джерел посилання та два додатки.

Вступ містить основні обґрунтування актуальності розробки обраної теми, об'єкт, предмет дослідження, мету та завдання які необхідно виконати для досягнення поставленої мети.

В першому розділі проведено огляд засобів розпізнавання та інтеграції мови в побутові прилади. Розроблено специфікацію вимог до апаратно-програмного забезпечення.

Другий розділ містить опис процесу проектування апаратно-програмного забезпечення.

У третьому розділі наведено опис процесу розробки апаратного забезпечення та розробки програмного забезпечення для програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові прилади.

Четвертий розділ присвячено тестування розробленого програмного та апаратного забезпечення.

У висновку описано результати виконання кваліфікаційної роботи.

Додатки містять код програмного забезпечення та інформацію про апробацію кваліфікаційної роботи.

В спеціальній частині з охорони праці розглянуто забезпечення вимог охорони праці на робочому місці.

Кваліфікаційна робота містить 61 сторінку (без додатків), 49 рисунків, 31 джерело посилання, 2 додатки.

ABSTRACT

of the Master's Thesis

"Software and hardware complex of language recognition and integration in household appliances"

Student: Veselovskyi Vladyslav Dmytrovych

Supervisor: Dr.Sc. (Eng.), Professor Zhuravska Iryna Mykolaivna

The relevance of the topic of the qualification work is due to the rapid pace of development of devices working on the concept of "digital home". This concept is the integration of devices of different purposes and different manufacturers into a single system. In most cases, this integration is possible through the use of a common control system, for example, based on voice commands.

The processing of voice commands is a complex process based on methods of audio signal analysis: amplitude-time analysis, spectral analysis, etc. Among these methods, the most recent and modern neural networks should be highlighted. When applying this method, it is necessary to have a ready-made, "trained" neural network model, which can perform voice identification, speech recognition, emotion analysis, etc.

The object of the study: processes of automation of control of household appliances that perform their functions through the receipt of voice commands.

Subject of research: methods and ways of creating automated control systems of household appliances, which perform their functions only after the identification of voice commands.

Objective: to develop a hardware-software system for speech recognition and integration of voice commands into household appliances.

To achieve the objectives, the following tasks were set:

- 1) to analyze modern household appliance control systems;
- 2) to analyze the shortcomings of existing systems;
- 3) to develop a hardware part of voice commands implementation in household appliances;

4) develop the software part of sending voice commands to household appliances;

5) create a mock-up model to demonstrate the operation.

The qualification work contains a list of abbreviations, an introduction, four chapters, a conclusion, a list of references and two appendices.

The introduction contains the main justifications for the relevance of the chosen topic, the object, subject of research, the purpose and tasks to be performed to achieve the goal.

The first chapter provides an overview of speech recognition and integration tools in household appliances. A specification of hardware and software requirements is developed.

The second section describes the hardware and software design process.

The third section describes the process of hardware development and software development for the hardware-software complex for speech recognition and integration into household appliances.

The fourth section is devoted to testing the developed software and hardware.

The conclusion describes the results of the qualification work.

The appendices contain the software code and information about the approbation of the qualification work.

The special part on labor protection considers the provision of labor protection requirements at the workplace.

The qualification work contains 61 pages (without appendices), 49 figures, 31 references, 2 appendices.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 ОГЛЯД ЗАСОБІВ РОЗПІЗНАВАННЯ ТА ІНТЕГРАЦІЇ МОВИ В ПОБУТОВІ ПРИЛАДИ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО- ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
1.1 Огляд засобів розпізнавання мови.....	7
1.2 Інтеграція мови в побутові прилади	11
1.3 Обґрунтування та вибір підходів щодо керування побутовими пристроями за допомогою голосу людини.....	15
1.4 Специфікація вимог до апаратно-програмного комплексу розпізнавання та інтеграції мови в побутові прилади.....	17
Висновок до першого розділу	18
2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЯ РОЗПІЗНАВАННЯ ТА ІНТЕГРАЦІЇ МОВИ В ПОБУТОВІ ПРИЛАДИ.....	20
2.1 Проєктування апаратного забезпечення.....	20
2.2 Проєктування програмного забезпечення	22
2.3 Застосування моделі нейронної мережі для розпізнавання мови	25
Висновок до другого розділу	30
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	31
3.1 Апаратне забезпечення	31
3.2 Налаштування одноплатного комп'ютера.....	35
3.3 Реалізація контролю розумних пристроїв	40
3.4 Реалізація обробки отриманих команд	42
Висновок до третього розділу.....	44
4 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ГОЛОСОВОГО КЕРУВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ	45
4.1 Реалізація апаратного забезпечення.....	45

4.2 Розробка програмного забезпечення.....	47
4.3 Тестування розробленого комплексу.....	54
Висновок до четвертого розділу.....	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60
ДОДАТОК А Код програми.....	63
ДОДАТОК Б Публікації за темою роботи.....	68

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ	– апаратно-програмне забезпечення
БД	– база даних
ОС	– операційна система
ПЗ	– програмне забезпечення
ШІ	– штучний інтелект
API	– Application Programming Interface
ANN	– Artificial Neural Networks
AVS	– Alexa Voice Service
CASE	– Computer-Aided Software Engineering
CNNs	– Convolutional Neural Networks
CPU	– Central Processing Unit
DNNs	– Deep Neural Networks
ERD	– Entity-Relationship Diagram
GPU	– Graphics Processing Unit
IFTTT	– If This Then That
HTTPS	– Hyper Text Transfer Protocol Secure
UML	– Unified Modeling Language

ВСТУП

Починаючи з 2018 р., житлові та офісні приміщення інтенсивно насичуються смарт-приладами: керованими електричними розетками, outdoor та indoor IP-камерами, інтелектуальними клапанами для перекриття води і газу тощо. Більшість пристроїв керуються через мережі Wi-Fi або через Bluetooth-порти. Але найбільш зручним слід визнати керування побутовими приладами за допомогою голосових команд.

Існує декілька методів ідентифікації голосу: амплітудно-часовий аналіз, спектральний аналіз та ін. Новітнім підходом до обробки даних в аудіосигналі є нейронні мережі. В такому разі потрібно надавати на вхід мережі аудіосигнал і очікувати, що мережа навчиться самостійно виявляти закономірності в даних і вирішувати необхідні завдання: чи то вилучення ознак; ідентифікація того, хто говорить; розпізнавання мови; аналіз емоцій тощо.

Об'єкт дослідження (розробки): процеси автоматизації керування побутовими приладами, що виконують свої функції через отримання голосових команд.

Предмет дослідження (розробки): методи та засоби створення систем автоматизованого керування побутовими приладами, що виконують свої функції тільки після ідентифікації голосових команд.

Мета: розробити програмно-апаратний комплекс розпізнавання мови та інтеграції голосових команд в побутові прилади.

Для досягнення поставленої мети було поставлено такі **завдання:**

- 1) оглянути сучасні системи керування побутовими приладами;
- 2) проаналізувати недоліки існуючих систем;
- 3) розробити апаратну частину імплементації голосових команд до побутових приладів;

4) розробити програмну частину відправки голосових команд до побутових приладів;

5) розробити комп'ютерну модель для демонстрації роботи.

Методи дослідження: абстрагування, аналіз і синтез, спостереження, порівняння, комп'ютерне моделювання.

Наукова новизна роботи полягає в поєднанні методів ШІ, обробки природної мови (Natural Language Processing) та формальних граматики для розпізнавання голосових команд

Практичне значення роботи полягає в тому, що дана розробка дозволить розпізнавати мову керування та інтегрувати голосові команди в побутові пристрої.

Робота пройшла **апробацію** під час XXV Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 07–11 листопада 2022 р.).

Публікації. Основні положення та результати магістерської роботи опубліковані у збірнику матеріалів XXV Всеукраїнської науково-практичної конференції «Могилянські читання–2022» [12].

1 ОГЛЯД ЗАСОБІВ РОЗПІЗНАВАННЯ ТА ІНТЕГРАЦІЇ МОВИ В ПОБУТОВІ ПРИЛАДИ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Огляд засобів розпізнавання мови

В сучасному світі популярність набирає популярність пристроїв, що здатні працювати в системі «цифровий будинок». Цифровий будинок, на відміну від «розумного будинку», не ставить за мету об'єднувати та синхронізувати роботу всіх пристроїв у приміщенні, а також керувати ними як цілісною системою [1]. Цифровий будинок робить ставку на розвиток гаджетів різних компаній та різного призначення, нп., електричних пилососів, холодильників, прасок, пральних машин, сушарок, тостерів і т. п. (надалі – смарт-пристрої), якими можна керувати через мобільні застосунки та/або відповідні датчики (руху, вологості, температури й т. і.) тощо.

Засоби керування смарт-пристроями дуже швидко розвиваються. Популярний підхід на цей час – це використання мови або голосових помічників [2]. Цей підхід дозволяє власнику керувати пристроями цими пристроями без додаткового обладнання: сенсорних панелей, мобільних телефонів, пультів дистанційного керування тощо.

Голосове керування реалізуються на основі розподіленої архітектури [2]. Одним з таких засобів є «віртуальний асистент» Alexa Voice Service (AVS), який отримує голосові команди зі спеціалізованих пристроїв «колонок» або мобільних телефонів під керуванням сучасних операційних систем (ОС), а обробка отриманих команд відбувається за допомогою нейронної мережі, розміщеної на хмарному сервісі Amazon.

Актуальним побутовим обладнанням зі смарт-функціями, керування яким здійснюється AVS, є наступне: телевізори, програвачі, розетки з відповідним інтерфейсом та ін. Найбільш функціональними є засоби керування електричним живленням. Наприклад, Аоусосг 13А Alexa є

окремим пристроєм з мережевим інтерфейсом Wi-Fi 2,4 ГГц, що може отримувати сигнал керування від AVS (з підключенням до сервісу Popotan Smart) або власного програмного забезпечення (ПЗ) [3].

На практиці одна окрема розетка Аоусосг 13А Alexa використовується рідко, частіше застосовують набір з кількох розеток, підключених до сервісу Popotan Smart. В цьому випадку AVS налаштовується користувачем для можливості обробки голосових команди, наприклад, визначаються кімната розміщення та призначення кожної розетки (рис. 1.1).

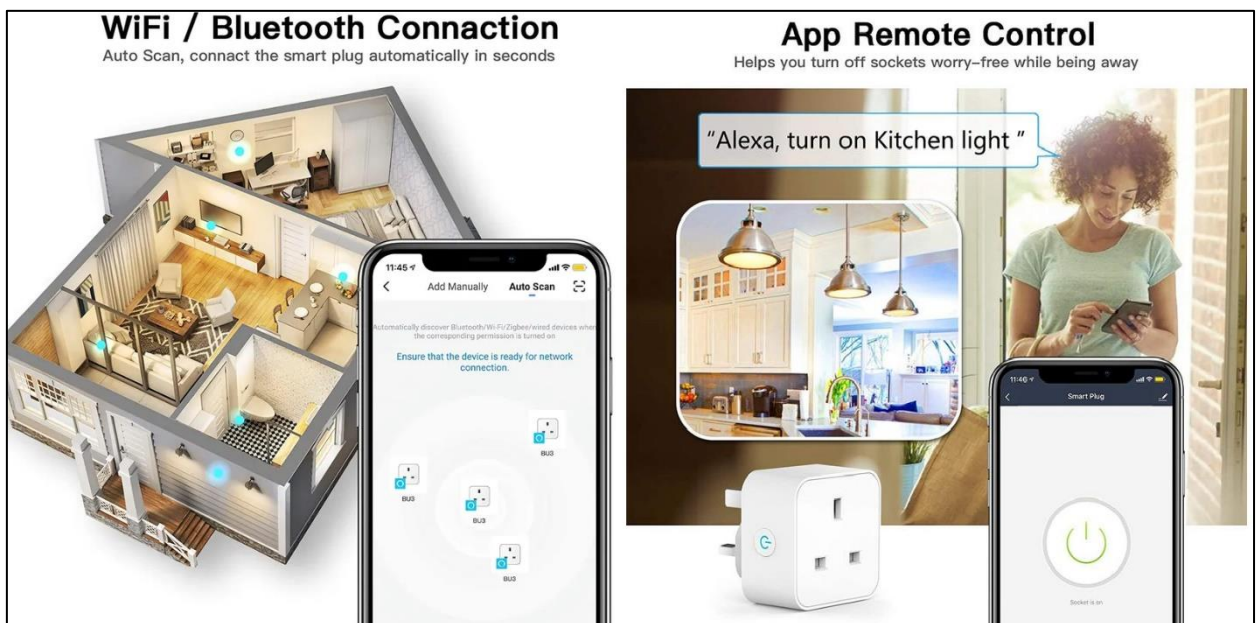


Рисунок 1.1 – Інтерфейс налаштування Аоусосг 13А Alexa для AVS

Варто зазначити, що AVS має значний недолік, а саме обмежений перелік мов, що можуть бути сприйняті та оброблені. Використання для керування пристроями Української мови неможливе.

Cortana – це програмне забезпечення для реалізації взаємодії з ОС Windows за допомогою голосу [2]. В основі цього програмного забезпечення також лежить використання елементів штучного інтелекту (ШІ), але нейронна мережа працює на серверах компанії Microsoft і прив'язана до акаунтів користувачів.

Варто відзначити, що Cortana, як і AVS, може працювати на спеціальному пристрої – Smart Speaker (смартколонці). Пристрої Smart Speaker являють собою інтерфейс голосового контролю, хоча і не мають достатньо обчислювальної потужності для підтримки роботи ШІ.

Віртуальна голосова помічниця з елементами ШІ від компанії Microsoft Cortana може керувати різними пристроями, наприклад, освітленням. Для цього необхідно застосувати пристрої Smart Bulb (розумна лампа). Компанія Philips пропонує PAR38 – E26 Smart Bulb (рис. 1.2), що за допомогою спеціалізованого сервісу може керуватись AVS або Cortana [4].



Рисунок 1.2 – Philips PAR38 – E26 Smart Bulb

У Cortana є більше недоліків ніж в Alexa Voice Service: проблеми з розпізнаванням мови, обов'язково потребує наявності акаунту Microsoft, а також підтримує дуже обмежену кількість мов. Окрім того, Cortana в нових версіях ОС Windows 10 (21H2) та Windows 11 представлена лише у вигляді функції для комерційних версій.

ПЗ для реалізації голосового керування Siri є також дуже розповсюдженим та інтегрованим в ОС iOS 6 та вище і може бути

використано лише в пристроях компанії Apple Inc. Побудоване це ПЗ також на основі ШІ й використовує для обробки голосу спеціалізований сервер (рис. 1.3).

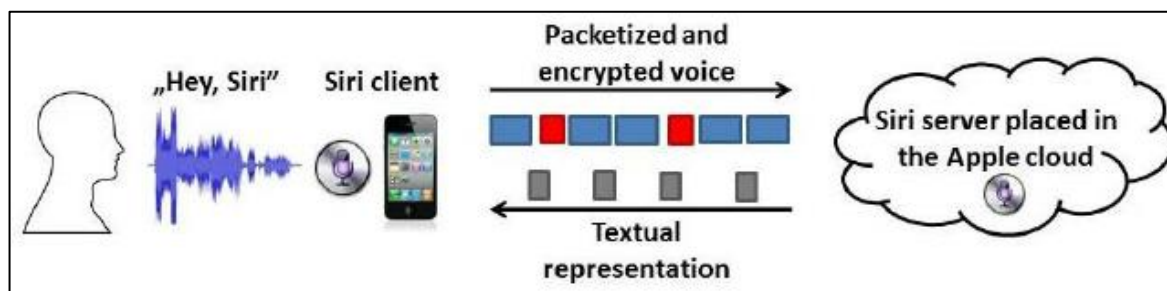


Рисунок 1.3 – Схематичне відображення архітектури Siri

Розумна помічниця для власників пристроїв від фірми Apple – Siri може керувати Philips PAR38 – E26 Smart Bulb, деякими типами розумних розеток, а також термостатами. На відміну від розглянутих вище пристроїв термостат має більший функціонал і приймає ширший спектр голосових команд.

Термостат Ecobee3 Lite Smart (рис. 1.4) підтримує можливість інтеграції з Siri за допомогою використання стороннього сервісу Nest. У випадку використання цього термостату голосовий помічник може як керувати самим пристроєм, так і подавати користувачу зчитані дані (температура, вологість) за допомогою голосу.



Рисунок 1.4 – Термостат Ecobee3 Lite Smart

Siri має значно більший список мов, що підтримується. Проте варто зазначити, що керування локалізацію неможливо голосовими командами, а сам перелік смарт-пристроїв хоч і є більшим ніж в аналогів, але підтримується не все обладнання.

При дослідженні інструментів розпізнавання мови неможливо пропустити Google Assistant. Цей інструмент також є розповсюдженим в сучасному світі. ПЗ Google Assistant побудоване на основі інструментарію ШІ, але на відміну від попередніх засобів має додаткову функцію збереження діалогів з користувачем для подальшого вдосконалення нейронної мережі.

Google Assistant інтегрується з багатьма смарт-пристроями за допомогою використання сервісів: Philips Hue, SmartThings, Nest, Ring, WeMo [3-5]. Перевагою вказаних сервісів є те, що без створення наскрізної системи «розумний будинок», де все базується на центральному контролері, вони можуть бути інтегрованими, наприклад, зі службою IFTTT (аббревіатура від “If This Then That”, укр. ««Якщо це, тоді те»), яка керує онлайн-платформами цифрової автоматизації та поєднує безліч інших сервісів, що дозволяють автоматизувати комплекс певних завдань.

Наприклад, зазначене ПЗ може приймати команду на зразок «Гей, Google, увімкни світло». В такому разі за наявності спеціалізованого пристрою, наприклад, вже згаданого вище Philips PAR38 – E26 smart bulb, користувач увімкне освітлення. Обробку мови Google Assistant здійснює за допомогою хмарного сервісу.

Недоліком Google Assistant є невелика кількість мов, що підтримуються і режим роботи «слухати завжди», що може призвести до порушення конфіденційності.

1.2 Інтеграція мови в побутові прилади

В підрозділі 1.1 розглянуто можливості використання розумних розеток, керованого освітлення та термостатів, але це лише невелика частина обладнання, яке може бути інтегроване в єдину систему. Для визначення набору команд та особливостей роботи програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові прилади необхідно зазначити класи пристроїв, які є найбільш розповсюдженими.

Отже, для створення повноцінного апаратно-програмного комплексу інтеграції мови в побутові прилади необхідно визначити, які саме пристрої будуть працювати у комплексі та які функції вони можуть виконувати. Відповідно до сучасних досліджень [6] можна виділити 5 основних груп:

- 1) прилади керування мікрокліматом;
- 2) прилади керування освітленням;
- 3) системи безпеки;
- 4) мультимедійні системи;
- 5) пристрої керування іншою побутовою технікою та електромережею.

Прилади керування мікрокліматом можуть бути різними, в розділі 1.1 вже описано можливість інтеграції термостата Ecobee3 Lite Smart з керуванням голосом, але важливим є також контроль опалення та вентиляції.

Компанія TESH Sterowniki [7] пропонує спеціалізовані контролери (рис. 1.7), які можуть керувати котлами й радіаторам та інтегруються зі спеціалізованими сервісами. Ці сервіси дозволяють Google Assistant та AVS використовувати керування голосом.

Ряд контролерів опалення TESH Sterowniki містить декілька моделей, але вони побудовані на загальному принципі використання реле у якості основного засобу контролю.



Рисунок 1.7 – Контролер TECH Sterowniki

Підключення до мережі та спеціалізованих сервісів відбувається за допомогою інтерфейсу Wi-Fi. Може здійснюватися як централізовано для всієї системи, так і в окремих приміщеннях.

Прилади керування освітленням частково розглянуті в підрозділі 1.1, але вони мають багато різних варіацій: освітлення за таймером, освітлення при виявленні руху тощо. В контексті освітлення будинки поширеними є релейні вимикачі з функціями інтеграції до спеціалізованих сервісів та використанням голосового керування. Прикладом такого засобу є Fibaro Single Switch (рис. 1.8), що інтегрується до сервісу Apple HomeKit та може керуватись Siri [8]. Підключення цього пристрою до приладу керування

здійснюється за допомогою інтерфейсу Bluetooth, що обмежує дальність передачі сигналу.



Рисунок 1.8 – Fibaro Single Switch

Системи безпеки – окремий клас обладнання, що містить багато різних пристроїв, датчиків та функцій. В плані керування ці системи мають можливість вмикати або вимикати режими забезпечення безпеки й можуть керуватись в абсолютні різні способи:

- системи моніторингу витоку води або газу працюють постійно і не вимикаються;
- системи пожежної безпеки також працюють постійно і мають окремі канали зв'язку з пожежниками;
- системи відеоспостереження та охорони можуть вимикатись користувачем, але у якості елемента керування не використовують голосові команди, найчастіше функцію контролю виконують спеціалізовані ключі, пульти або застосунки.

Мультимедійні системи у більшості випадків підтримують можливість запуск спеціалізованого ПЗ для голосового керування, розглянутого у підрозділі 1.1.

Пристрої керування іншою побутовою технікою та електромережею у більшості випадків технічно реалізуються за допомогою реле. Вище розглянуто пристрій Aoусосr 13A Alexa, що керує подачею живлення за рахунок увімкнення/вимкнення живлення. Варто зауважити, що частіше застосовують пристрої, які мають набір реле для керування кількома пристроями одночасно. Прикладом такого модуля є AUBESS 2, який має можливість керувати чотирма електричними приладами одночасно [9].



Рисунок 1.9 – Модуль реле AUBESS 2

Зазначене обладнання зазвичай підключається для мережі за допомогою інтерфейсу Wi-Fi та інтегрується з Google Assistant за допомогою спеціалізованого сервісу.

1.3 Обґрунтування та вибір підходів щодо керування побутовими пристроями за допомогою голосу людини

Проведений огляд демонструє, що керування мовою або голосом є актуальною технологією у тренді швидкого зростання кількості смарт-пристроїв у сучасному будинку. Визначено, що керування мовою здійснюється за рахунок використання спеціалізованого ПЗ розпізнавання мови.

Варто підкреслити, що спільним недоліком більшості ПЗ для розпізнавання мови є підтримка обмеженої, невеликої кількості мов та можливість застосування виключно на певних операційних системах. Отже, актуальним є розробка програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові прилади, що матиме можливість додавання мовних пакетів та інтеграція з різними операційними системами.

Виконання кваліфікаційної роботи потребує використання певного апаратного забезпечення, що дозволить опрацьовувати вхідні голосові (мовні) команди різними мовами та керувати побутовими приладами. Сучасні одноплатні комп'ютери мають значні обчислювальні потужності, що забезпечуються процесорами архітектури ARM, інтерфейси для підключення великої кількості периферійного обладнання (датчиків, мікроконтролерів, модулів реле тощо), мережеві інтерфейси та ін. [10]. Зазначені характеристики роблять одноплатний комп'ютер найбільш оптимальним варіантом основи програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові прилади.

Методи розпізнавання та обробки голосових команд базуються на використанні методів ШІ, а точніше моделі нейронних мереж [2]. Існує декілька актуальних платформ і фреймворків, на яких цей механізм може бути реалізований: Tensor Flow, Whisper, Google Cloud Speech API тощо [11].

Керування побутовими приладами можливе шляхом застосування релейних контролерів, досліджених у підрозділах 1.1 та 1.2, або аналогічних з можливістю підключення до одноплатного комп'ютера. Наявність спеціалізованих інтерфейсів дозволяє підключати датчики для реалізації автоматизованого керування побутовими пристроями.

Отже, використання одноплатного комп'ютера у якості апаратного забезпечення дає змогу запустити програмне забезпечення для розпізнавання та обробки мови. Наявність інтерфейсів для підключення периферійного обладнання – датчиків, модулів реле тощо – дає змогу реалізувати керування

побутовими приладами, наприклад, модуль реле в поєднанні з датчиком світла можуть забезпечити керування освітленням. Реалізація розпізнавання мови можлива за рахунок ШІ й засобів його реалізації

1.4 Специфікація вимог до апаратно-програмного комплексу розпізнавання та інтеграції мови в побутові прилади

1 ПРИЗНАЧЕННЯ ТА МЕЖИ ПРОЄКТУ

1.1 *Призначення системи, для якої розробляється апаратно-програмне забезпечення:* апаратно-програмний комплекс розпізнавання та інтеграції мови в побутові прилади, призначений для використання в приватному будинку або квартирі. Обидва варіанти представляють собою місце проживання людей, що містить побутові пристрої, керування яких має здійснювати користувач за рахунок голосових (мовних) команд.

2 ЗАГАЛЬНИЙ ОПИС

2.1 *Сфера застосування.* Апаратно-програмний комплекс розпізнавання та інтеграції мови в побутові пристрої може бути застосований для спрощення керування побутовою технікою в житлових приміщеннях.

2.2 *Характеристики користувачів.* В проєкті передбачена всього одна роль користувачів. Розмежування прав доступу до різних функцій не передбачено. Користувачі, що працюватимуть з зазначеним комплексом, не матимуть поглиблених знань в галузі інформаційних технологій.

2.3 *Загальна структура і склад апаратно-програмного забезпечення (АПЗ). Системні вимоги.* До складу проєкту має входити декілька окремих компонентів: пристрій керування – одноплатний комп'ютер, модуль реле для керування побутовими приладами.

3 *ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ.* Для забезпечення роботи ПЗ з реалізацією методів ШІ необхідна наявність одноплатного комп'ютера, що обладнаний процесором з тактовою частотою не менше 1 ГГц, обсягом оперативної пам'яті не менше 1 Гбайт та мережевим

інтерфейсом для можливості застосування спеціалізованих АРІ розпізнавання мови.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Архітектура програмної системи. Програмна складова має бути реалізована на основі об'єктно-орієнтованого підходу. Реалізація методів ШІ має бути реалізована з використанням спеціалізованих сервісів.

4.2 Системне програмне забезпечення. Розроблений комплекс повинен коректно працювати на дистрибутивах ядра Linux. Для забезпечення переносимості необхідно використання інтерпретатора мови Python.

4.3 Програмне забезпечення ведення інформаційної бази. Зберігання словника команд має здійснюватися за рахунок системи керування базами даних SQLite.

4.4 Мова і технологія розробки ПЗ. У якості мови програмування необхідно застосувати мову Python, що забезпечую переносимість та сумісність з різними моделями одноплатних комп'ютерів.

5 ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

5.1 Апаратний інтерфейс для реалізації мережевого з'єднання має підтримувати бездротовий стандарт Wi-Fi 802.11ac.

5.2 Комунікаційний протокол HTTPS має бути використаний для підключення до спеціалізованого сервісу реалізації методів ШІ.

Висновок до першого розділу

В розділі проведено огляд сучасних засобів розпізнавання мови. Визначено переваги та недоліки цих засобів. Визначено основні технології розпізнавання голосових (мовних) команд. Наведено опис побутових приладів, керування якими здійснюється за допомогою мови.

Обґрунтовано вибір підходів для реалізації керування побутовими пристроями за допомогою голосу людини. На основі проведеного

дослідження розроблено специфікацію вимог до апаратно-програмного забезпечення.

2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЯ РОЗПІЗНАВАННЯ ТА ІНТЕГРАЦІЇ МОВИ В ПОБУТОВІ ПРИЛАДИ

2.1 Проєктування апаратного забезпечення

Розробка програмно-апаратного модуля розпізнавання та інтеграції мови в побутові прилади вимагає наявності значного обсягу обчислювальних ресурсів для можливості роботи з основним методом розпізнавання мови – штучним інтелектом. Окрім того для зручності розміщення в побутовому будинку апаратна складова повинна мати компактний формфактор. Зазначені характеристики притаманні одноплатним комп'ютерам [10].

Більшість моделей одноплатних комп'ютерів обладнані інтерфейсом general-purpose input/output. Цей інтерфейс дозволяє підключення периферійного обладнання до одноплатного комп'ютера. Контакти GPIO можуть виконувати роль входів або виходів для передачі даних в залежності від налаштувань. У більшості одноплатних комп'ютерів присутні спеціалізовані виходи GPIO, що призначені для живлення та не можуть бути налаштовані для передачі даних (рис. 2.1).

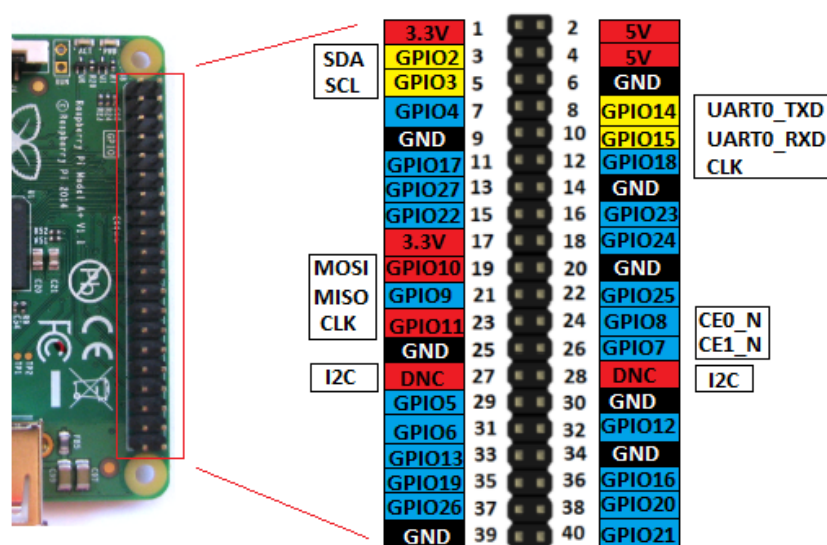


Рисунок 2.1 – Інтерфейс GPIO Raspberry Pi 3b

Змоделювати підключення компонентів апаратної частини комплексу можливо використавши спеціалізоване ПЗ: Fritzing. Цей продукт дозволяє будувати принципові схеми апаратних компонентів та моделювати процес їх роботи [13].

Найчастіше Fritzing застосовується для побудови моделей програмно-апаратних комплексів на основі плат Arduino, але вбудовані засоби цього інструменту дозволяють завантаження моделей різних одноплатних комп'ютерів.

Продемонструвати розроблений програмно-апаратний комплекс не можливо не відобразивши схему взаємозв'язків компонентів його модулів. Цей можливо зробити відобразивши його принципову схему. На принципових схемах електричні елементи позначають у вигляді умовних позначень, а також вказують зв'язки між ними. Відобразити таку схему можливо також за допомогою Fritzing (рис. 2.2).

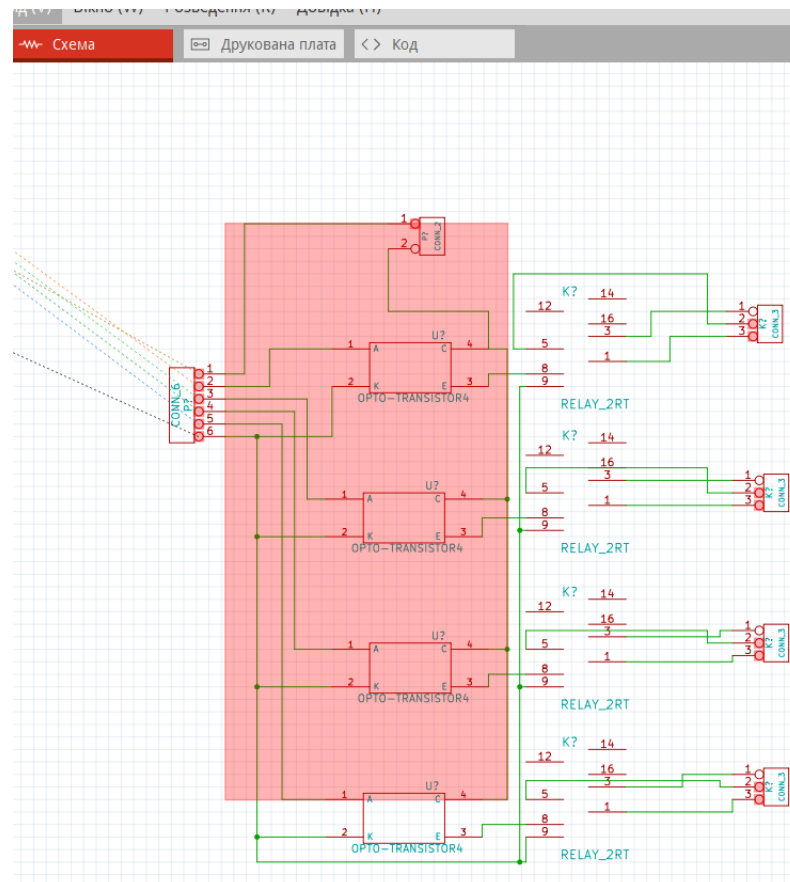


Рисунок 2.2 – Відображення принципової схеми засобами Fritzing

Наявність моделей програмного забезпечення дає можливість відтворення розробленого програмно-апаратного забезпечення та моделювання можливих його покращень.

2.2 Проектування програмного забезпечення

Розробка програмного забезпечення для програмно-апаратних комплексів є складним процесом. Перед початком розробки необхідно продумати та описати структуру і алгоритм роботи цього ПЗ.

Опис структури програмного забезпечення можливо виконати засобами Unified Modeling Language (UML; укр. «уніфікована мови моделювання»). UML являє собою графічний опис компонентів програмного забезпечення, що розробляється на основі парадигми об'єктно-орієнтованого програмування. Може використовуватись для схематичного опису бізнес-процесів, пов'язаних з ПЗ [14].

Мова UML включає різні моделі: діаграму станів, діаграму класів, діаграму прецедентів тощо. Всі ці діаграми використовуються комплексно та забезпечують розробникам комплексне уявлення про архітектуру ПЗ.

Діаграми (моделі) UML використовуюся комплексно для більш повного опису програмного забезпечення. В такому випадку рекомендовано використовувати наступну ієрархію ПЗ (рис. 2.3).

Найчастіше при проектуванні об'ємного програмного забезпечення використовується діаграма класів (Class diagram). Це статична структурна діаграма, що відображає: класи, методи, атрибути та їх взаємозв'язок. Є варіації цієї моделі:

- концептуальна діаграма класів. Варіація діаграми класів в якій відображені лише прикладні класи предметної області.

- діаграма класів етапу специфікації ПЗ. Діаграма класів, що будується на етапі побудови специфікації програмного забезпечення і є частиною процесу проектування ПЗ.

– діаграма класів етапу реалізації ПЗ. Відображаються елементи безпосередньо присутні в коді ПЗ.

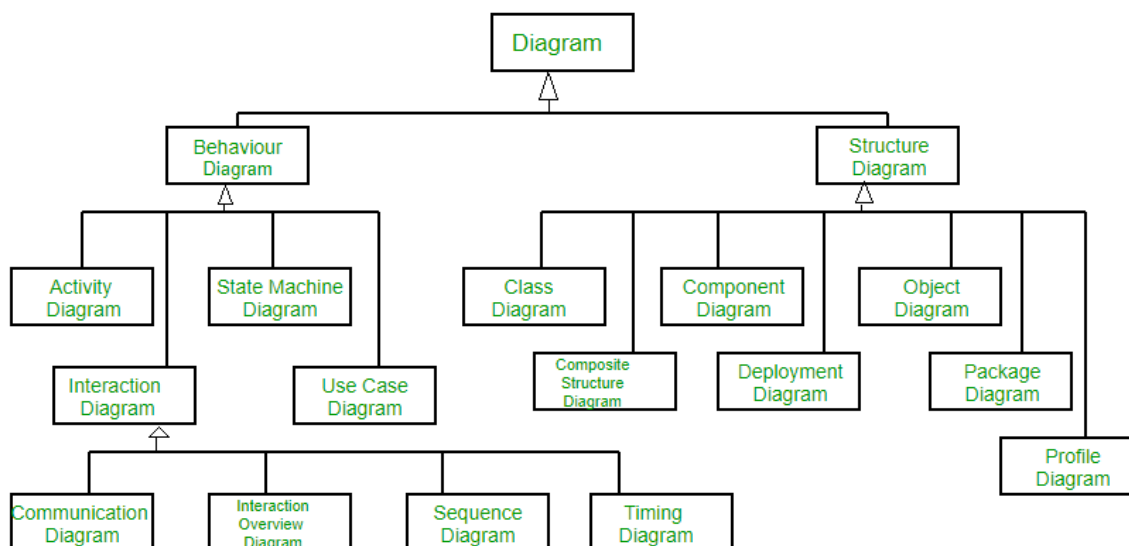


Рисунок 2.3 – Ієрархія діаграм UML

Описаний вище поділ типів діаграм класів не впливає на інструментарій її побудови. Програмне забезпечення для побудови цих моделей належить до класу computer-aided software engineering (CASE). Інструменти CASE дозволяє не тільки будувати різні моделі, а й часткового генерувати код на основі створених моделей. Перелік ПЗ, яке належить до цього класу доволі обширний, але варто виділити SoftwareIdeasModeler.

SoftwareIdeasModeler реалізує можливість побудови моделей UML, ERD-діаграм, блок-схем та іншого [15]. Створені моделі можуть бути експортовані у вигляді графічних зображень для формування майбутньої нотації або у вигляді програмного коду для мови програмування (C++, C#, PHP, Python та ін.). Побудову та редагування моделей ПЗ цей засіб реалізує за допомогою графічного інтерфейсу користувача (рис. 2.4).

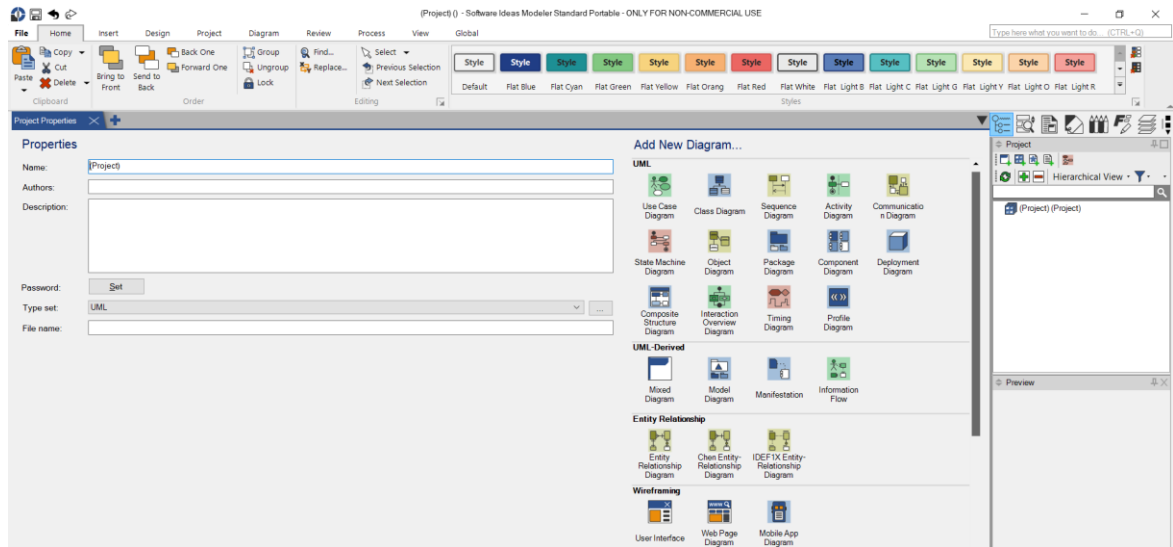


Рисунок 2.4 – Графічний інтерфейс користувача SoftwareIdeasModeler

Зберігання даних в програмно-апаратних комплексах вимагає застосування спеціалізованого сховища – бази даних (БД). В залежності від типу БД обираються засоби її проєктування.

Найбільш розповсюдженою для проєктів з одноплатними комп'ютерами застосовується реляційна БД. Проєктування такої бази здійснюється на основі Entity-Relationship Diagram (ERD). Ця модель має декілька варіантів нотації, але незалежно від типу дозволяє відобразити взаємозв'язок між таблицями в БД (рис. 2.5). Побудову бази можливо виконати інструментарієм SoftwareIdeasModeler.

Опис алгоритму роботи певного пристрою, також, є невід'ємною частиною проєктування програмно-апаратного комплексу. Виконати таке проєктування можливо за допомогою блок-схеми. Блок-схема – це відображення алгоритму розв'язання задачі за допомогою геометричних елементів, які позначають операції, цикли, умови тощо.

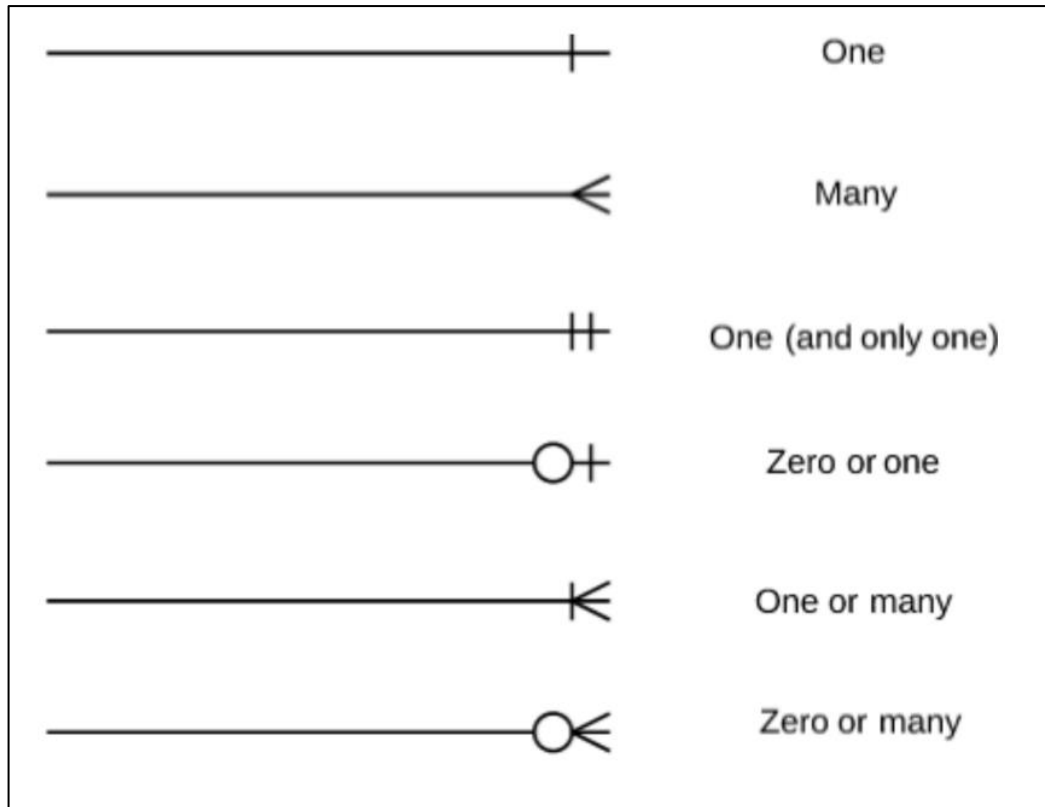


Рисунок 2.5 – Типи зв'язків діаграми ERD

Проектування графічного інтерфейсу користувача здійснюється за рахунок різних методів: побудови мокапів, діаграм компонентів, представлення карти сайту тощо.

2.3 Застосування моделі нейронної мережі для розпізнавання мови

Штучні нейронні мережі (НМ; англ. Artificial Neural Networks, ANN) – це обчислювальні системи, засновані на моделі біологічних нейронних мереж, з яких складається мозок тварин (рис. 2.6). Ці системи навчаються вирішенню задач, покращуючи свою продуктивність на них за рахунок дослідження прикладів, без спеціального програмування під задачу.

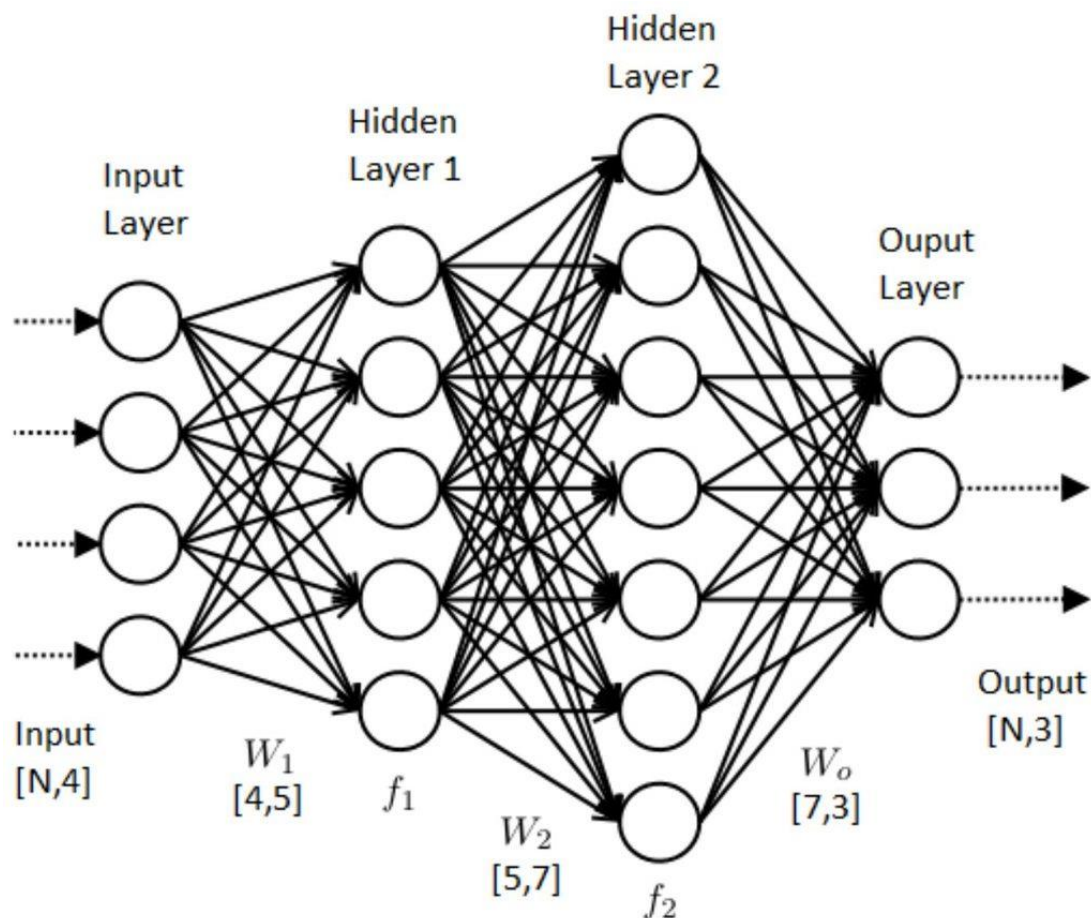


Рисунок 2.6 – Схематичне відображення моделі нейронної мережі

Важливою складовою програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові пристрої є модуль розпізнавання мови. Можливо реалізувати різними методами: застосувавши моделі Гауса або глибокі нейронні мережі (англ. Deep Neural Networks, DNNs).

На протязі довгого часу для аналізу звукового сигналу використовували модель Гаусової суміші, що дозволяло визначити побудувати акустичну модель мови. Однак використання DNNs дозволяють ефективніше визначати мову в акустичному сигналі, але потребують значно більше обчислювальних ресурсів [16].

Щоб зменшити обчислювальні витрати на етапі висновку, доцільно використовувати двійкові нейронні мережі для розпізнавання мовлення. За допомогою швидкої реалізацію множення двійкової матриці на сучасних архітектурах центрального процесора (англ. Central Processing Unit, CPU) і

графічного процесора (англ. Graphics Processing Unit, GPU) у реальних програмах можна досягти прискорення в 5–7 разів порівняно з множенням матриці з плаваючою комою з повною точністю [17–19].

Проте в незалежності від типу нейронної мережі для коректного розпізнавання мови необхідна «навчена» (натренована) модель. Натренувати модель можливо на великій кількості зразків мовних команд, але при розробці засобу керування побутовими пристроями необхідна наявність зразків вимови на багатьох різних мовах. Уникнути етапу навчання можна застосувавши вже натреновані моделі.

Мова програмування Python підтримує декілька фреймворків для роботи з ШІ: TensorFlow, PyTorch, Keras тощо. Ці фреймворки містять необхідних методів для створення, навчання та використання моделі нейронної мережі.

В контексті розпізнавання мови варто виділити фреймворки PyTorch. Цей інструмент ліг в основу Whisper, що є бібліотекою для обробки акустичного сигналу та визначення: визначення мови в акустичному сигналі, визначення якою саме мовою відбувається розмова, конвертація голосу в текст тощо[20].

Whisper дозволяє виконати кілька завдань одночасно використовуючи одну модель [20]. Проте для конвертації голосу в текст вимагає завантаження окремих моделей під кожну мову. Доступні моделі для понад 50 мов, але існує декілька варіації обсягів моделей під кожну мову:

- tiny обсягом 50 Мбайт дискового простору та 1 Гбайт оперативної пам'яті для коректної роботи;

- base обсягом 75 Мбайт дискового простору та 1 Гбайт оперативної пам'яті для коректної роботи;

- small обсягом 244 Мбайт дискового простору та 2 Гбайт оперативної пам'яті для коректної роботи;

– medium обсягом 769 Мбайт дискового простору та 5 Гбайт оперативної пам'яті для коректної роботи;

– large обсягом 1550 Мбайт дискового простору та 10 Гбайт оперативної пам'яті для коректної роботи.

Бібліотека Whisper здійснює обробку аудіосигналу за рахунок модулю PyAudio. Цей модуль дозволяє отримати аудіосигнал з мікрофону та розкласти його на частоти які згодом і будуть оброблені засобами PyTorch (рис. 2.7).

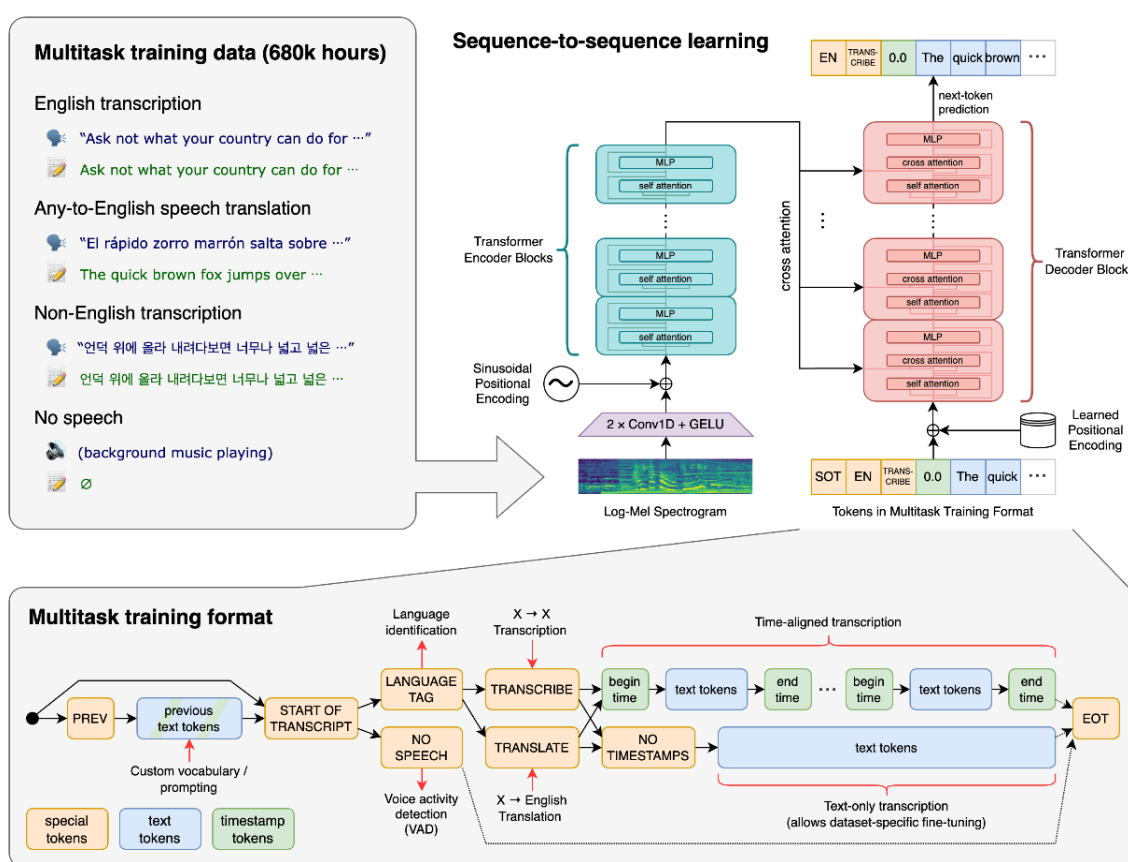


Рисунок 2.7 – Схематичне відображення процесу обробки аудіосигналу Whisper [20]

Натреновані моделі Whisper зберігаються у власному репозиторії та завантажуються динамічно за необхідності. Варто зазначити, що окрім наведених переваг Whisper має ряд недоліків: великий обсяг тренуваних

моделей, потреба в значних обсягах процесорного часу та оперативної пам'яті для використання моделі.

Інший варіант реалізації нейронної моделі для розпізнавання мови в аудіосигналі – це хмарні технології. Наприклад, Google Speech-to-Text API дозволяє за допомогою вебзапитів надіслати аудіосигналу до хмарного сервісу де за допомогою навченої моделі відбудеться конвертація в текст [21] (рис. 2.8).

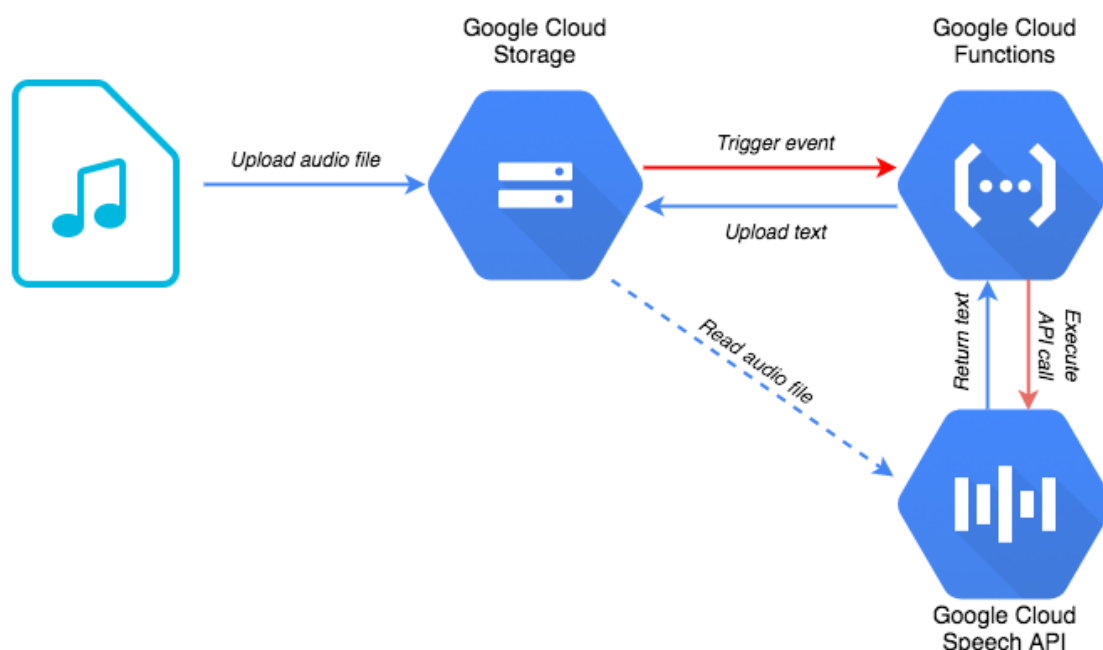


Рисунок 2.8 – Послідовність обробки аудіосигналу засобами Google [21]

Варто зазначити, що Whisper і Google Speech-to-Text API можуть використовуватись для реалізації розпізнавання мови у одному проєкті, але у різних режимах. Наприклад, Whisper завантажує модель та може працювати без мережі Інтернет, а Google Speech-to-Text API споживає менше ресурсів, але потребує стабільного підключення до серверів Google.

При наявності великої кількості прикладів для навчання штучної нейронної мережі можливе застосування інструментарію TensorFlow. TensorFlow – бібліотека призначена для спрощення реалізації машинного навчання в цілій низці задач: виявлення й розпізнавання образів, кореляцій зображень тощо.

TensorFlow потребує попередньої обробки вхідного звукового сигналу перед застосуванням нейронної мережі. Здійснити цю обробку (виділення частот та поділ на сегменти аудіодоріжки) можливо використавши бібліотеку PyAudio. Недоліком TensorFlow є відсутність репозиторію з навченими базами, що обумовлює необхідність проведення навчання нейронної мережі і тягне необхідність формування великої бази записів команду з різними інтонаціями на різних мовах.

Висновок до другого розділу

В другому розділі проаналізовано засоби проєктування програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові пристрої. Наведено опис інтерфейсу для підключення периферійного обладнання GPIO.

Досліджено засоби моделювання апаратного забезпечення: використання принципів схем. Розглянуто програмне забезпечення, що реалізує можливість цього проєктування.

Наведено опис засобів проєктування програмного забезпечення. Описано засоби опису архітектури програмного забезпечення. Досліджено основні моделі мови UML та область їх використання при створенні програмного забезпечення. Визначено особливості проєктування БД та засоби реалізації цього процесу.

Розглянуто можливість застосування методів ШІ – моделей нейронних мереж, для розпізнавання мови в аудіосигналі. Досліджено два варіанти інтеграції цього засобу в програмно-апаратний комплекс розпізнавання та інтеграції мови в побутові пристрої.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Апаратне забезпечення

Ринок одноплатних комп'ютерів дуже швидко розвивається, станом на 2023 рік представлена велика кількість різних виробників: Orange Pi, Raspberry Pi, ODROID тощо. Для визначення найбільш оптимальної моделі одноплатного комп'ютера необхідно проаналізувати найбільш розповсюджені моделі на сучасному ринку. Для забезпечення необхідного функціоналу одноплатний комп'ютер повинен підтримувати можливість роботи з засобами реалізації ШІ, мати мережевий інтерфейс, інтерфейс для підключення периферійного обладнання (мікрофон, модулі реле).

Виробник Orange Pi представляє на сучасному ринку декілька моделей: Orange Pi 5, Orange Pi 4, Orange Pi 3, Orange Pi Zero2, Orange Pi R1 Plus LTS. Модель Orange Pi 5 є найновішою і має значний обсяг обчислювальних ресурсів [22]: процесор Rockchip RK3588S побудований на архітектурі Cortex-A76 з тактовою частотою 2,4 ГГц, оперативна пам'ять стандарту LPDDR4 обсягом 8 Гбайт, мережевий інтерфейс Ethernet 10/100М/1000М, вбудована пам'ять відсутня, для роботи ОС доступний порт MicroSD (TF), підключення периферійного обладнання може бути підключення інтерфейсами USB3.0 або GPIO (26 входів/виходів). Розміри цього одноплатного комп'ютера складають 62 мм × 100 мм (рис. 3.1).

Характеристики Orange Pi 5 дозволяють його використання у якості апаратного забезпечення для реалізації апаратно-програмного модуля розпізнавання та інтеграції мови в побутові прилади, але ця модель має велику вартість, що перевищує ціну деяких Smart Speaker та деякі проблеми з підтримкою бібліотек реалізації ШІ, наприклад Pytorch.

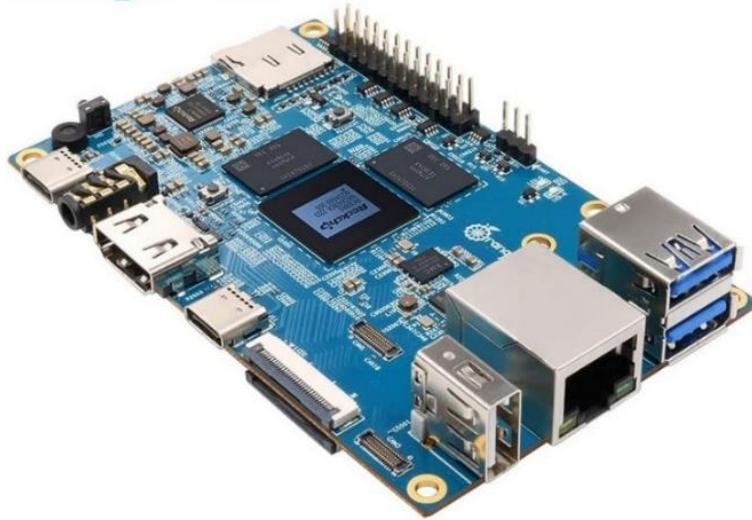


Рисунок 3.1 – Одноплатний комп'ютер Orange Pi 5

ODROID представлена на ринку багатьма моделями: ODROID-C4, ODROID-XU4, ODROID-N2+ та ін. На відміну від Orange Pi продукція ODROID має більш вузьку направленість на використання у якості різноманітних контролерів. Найновіша та розповсюджена модель ODROID-C4 підтримує більше різних спеціалізованих інтересів ніж аналоги.

Модель ODROID-C4 обладнана: процесором Amlogic S905X3 на архітектурі Cortex-A55 з тактовою частотою в 2 ГГц, оперативною пам'яттю обсягом 4 Гбайт стандарту DDR4, проводимим мережевим інтерфейсом Ethernet 10/100/1000M [23]. Підключення периферійного обладнання можливо інтерфейсами: GPIO, UART, SPI, USB, Micro USB. Розміри ODROID-C4 складають 62 мм × 100 мм.

Одноплатний комп'ютер Odroid-C4 від компанії Hardkernel, незважаючи на те, що використовує процесорний модуль Amlogic S905X3 з 4 трохи старішими ядрами Cortex A55 (проти чотирьох ядер Cortex A72 у Raspberri Pi4), працює на більших частотах ніж у конкурента.

Згідно з результатами тестів, проведених самим виробником Odroid-C4, компанії Hardkernel, їх одноплатний комп'ютер за продуктивністю обходить

еталонний Raspberry Pi4, при тому різниця в залежності від програми варіюється від 15 % до 60 % (рис. 3.2).

ODROID-C4 має значні обчислювальні потужності, але доступні дистрибутиви ОС підтримують застарілу версію Python, що не підтримує інструментарій для обробки засобів ШІ.

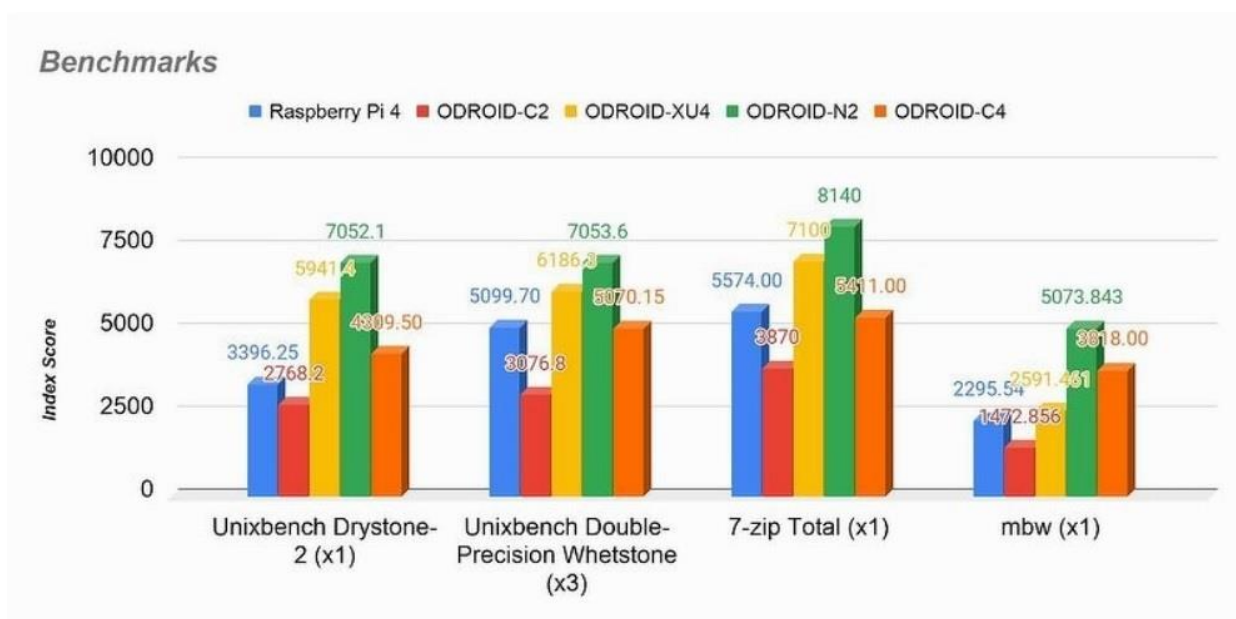


Рисунок 3.2 – Порівняльні характеристики одноплатного комп'ютера ODROID-C4 з конкурентами

Найбільш розповсюдженим та відомим є виробник Raspberry Pi. На сучасному ринку доступні декілька моделей: Raspberry Pi 4, Raspberry Pi 3b, Raspberry Pi Zero 2 та ін.. Raspberry Pi 4 має значний обсяг обчислювальних ресурсів, що спричиняє її попит серед користувачів. Станом на грудень 2022 року відчувається значний дефіцит цієї моделі на ринку України та її високу вартість. У той же час модель Raspberry Pi 3b коштує значно менше та має значний обсяг обчислювальних ресурсів.

Raspberry Pi 3b обладнана процесором Broadcom серії BCM2837B0 побудованим на архітектурі ARM Cortex-A53 з тактовою частотою 1,4 ГГц та 64-бітної архітектурою. Обсяг оперативної пам'яті складає 1 Гбайт стандарту

LPDDR2 [24]. Відсутня вбудована енергонезалежна пам'ять. Для передачі даних доступний бездротовий інтерфейс Wi-Fi (2,4 ГГц і 5 ГГц) 802.11ac. Підключення периферійного обладнання можливо інтерфейсами: GPIO, UART, SPI, USB. Розміри плати Raspberry Pi 3b складають 85 мм × 56 мм (рис. 3.3).



Рисунок 3.3 – Одноплатний комп'ютер Raspberry Pi 3b

Розповсюдженість та популярність Raspberry Pi призводить до постійного вдосконалення та оновлення драйверів та операційних систем. Завдяки цій особливості Raspberry Pi підтримує найбільш сучасні інструменти розробки ПЗ, в тому числі засоби ШІ.

Плата для роботи 4 модулів реле вже розглянута в розділі 2.1. Ця плата має доступні 6 контактів, що використовуються для підключення до одноплатного комп'ютера.

Отримати звуковий сигнал та обробки голосу застосовано USB камеру з вбудованим мікрофоном Logitech WebCam C270. Мікрофон цієї камери має функції шумопоглинання, що є перевагою при розпізнаванні мови. Драйвер для цієї камери підтримується більшістю ОС. Живлення камери відбувається інтерфейсом USB і забезпечується одноплатним комп'ютером.

Logitech WebCam C270 має розміри 72,91 мм × 31,91 мм (рис. 3.4), що разом з зазначеними характеристиками роботи камери оптимальним рішенням для використання в апаратно-програмному модулі розпізнавання та інтеграції мови в побутові прилади.



Рисунок 3.4 – Камера Logitech WebCam C270

3.2 Налаштування одноплатного комп'ютера

Першим елементом, що необхідний для роботи одноплатного комп'ютера, є операційна система (ОС). Для Raspberry Pi доступні різні ОС: Raspbian (зараз має назву Raspberry Pi OS), Ubuntu, LibreELEC, Windows 10 IoT тощо. Більшість базується на ядрі Linux та підтримує інструментарій необхідний для роботи з засобами ШІ [25].

Raspberry Pi OS може бути встановлена за допомогою спеціалізованого ПЗ, що доступний на офіційному сайті Raspberry Pi – «Raspberry Pi Imager». Цей засіб дає змогу користувачу обрати операційну систему та диск, на який відбудеться запис (рис. 3.5).



Рисунок 3.5 – Інтерфейс Raspberry Pi Imager

Коректна робота застосунку з використанням ШІ вимагає наявності підключення до мережі інтернет. Враховуючи наявність вбудованого бездротового інтерфейсу Wi-Fi налаштувати підключення можливо за допомогою вбудованого ПЗ Raspberry Pi OS – raspi-config (рис. 3.6–3.10).

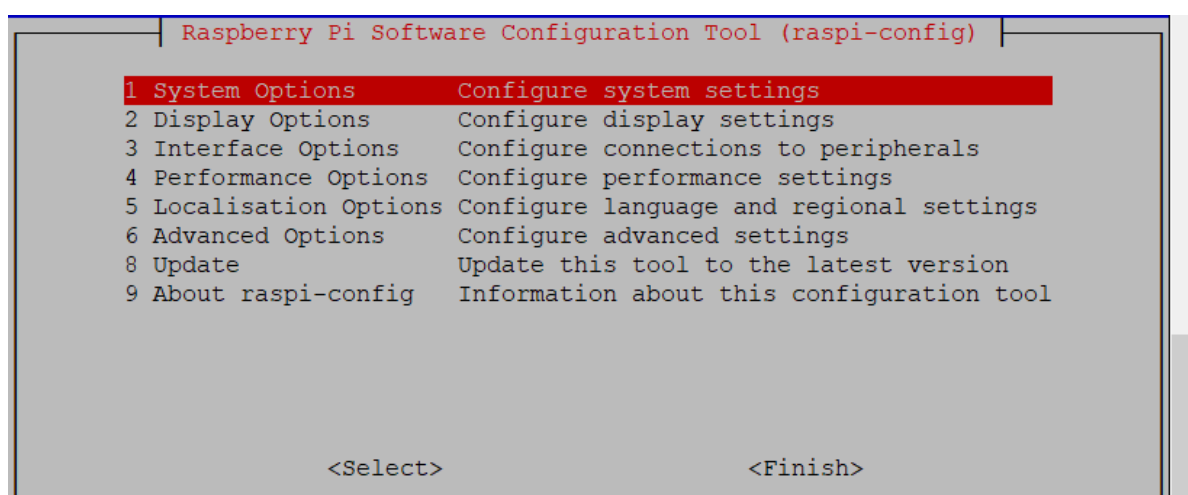
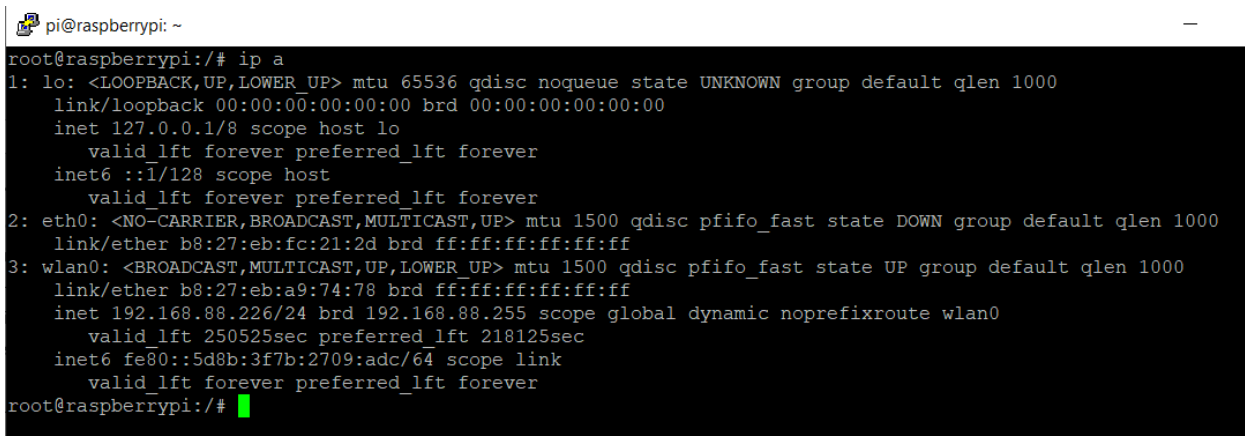


Рисунок 3.6 – Інтерфейс Raspi-config

Коректність налаштувань можливо перевірити переглянувши налаштування мережі. В Raspberry Pi OS це здійснюється командою «ip a». Вбудований бездротовий адаптер одноплатного комп'ютеру у переліку

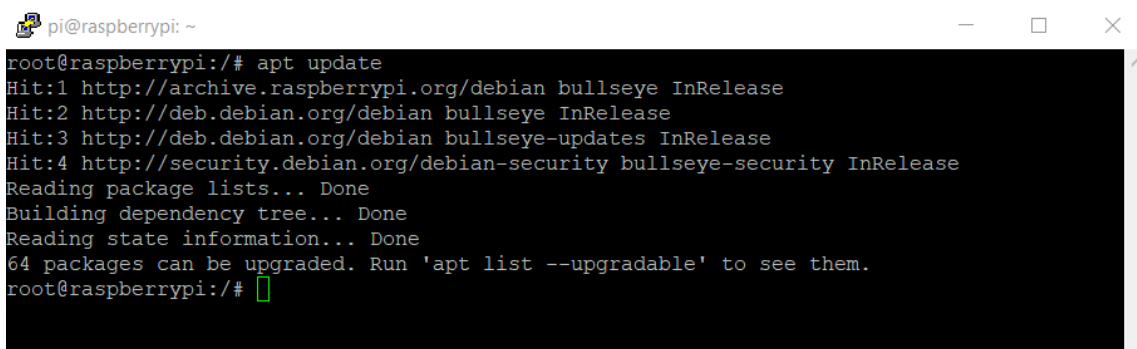
адаптерів позначається «wlan0», якщо цей адаптер отримує IP адресу налаштування встановлені коректно (рис. 3.7).



```
pi@raspberrypi: ~
root@raspberrypi:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether b8:27:eb:fc:21:2d brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether b8:27:eb:a9:74:78 brd ff:ff:ff:ff:ff:ff
    inet 192.168.88.226/24 brd 192.168.88.255 scope global dynamic noprefixroute wlan0
        valid_lft 250525sec preferred_lft 218125sec
    inet6 fe80::5d8b:3f7b:2709:adc/64 scope link
        valid_lft forever preferred_lft forever
root@raspberrypi:/#
```

Рисунок 3.7 – Налаштування мережевого підключення

Налаштування одноплатного комп'ютеру не обмежується виключно налаштуванням бездротової мережі. Реалізація програмного забезпечення з використанням інструментарію ШІ потребує додаткового завантаження необхідних бібліотек, що можливо здійснити за допомогою менеджера керування пакунками. В операційній системі Raspberry Pi OS використовується advanced packaging tool (APT).



```
pi@raspberrypi: ~
root@raspberrypi:/# apt update
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 http://security.debian.org/debian-security bullseye-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
64 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@raspberrypi:/#
```

Рисунок 3.8 – Застосування менеджера керування пакунками APT

Мова програмування Python забезпечує можливість зворотної сумісності версій за рахунок використання віртуальних оточень. Кожна встановлена версія цієї мови має власне віртуальне оточення, фізично це директорія з бібліотеками які використовуються при розробці.

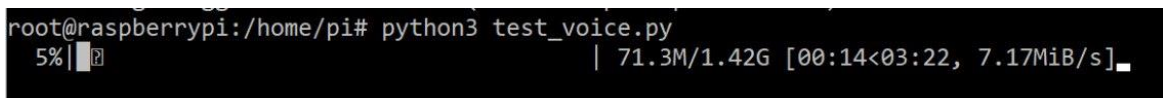
Додати до оточення нову бібліотеку можливо власним менеджером керування пакунками Python – pip3. Цей менеджер керування пакунками дозволяє встановлення вже згаданого фейворку Django [26]:

```
pip3 install Django
```

Встановлення засобів ШІ відбувається схожим чином, але розглянутий Whisper розміщений на репозиторії GitHub тому команда для завантаження бібліотеки виглядає наступним чином:

```
pip install git+https://github.com/openai/whisper.git
```

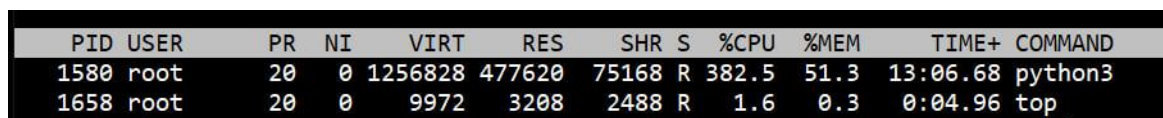
У разі вдалого встановлення можливий запуск застосунку з використанням Whisper. Запуск застосунку з використанням нейромережі зазначеного застосунку починається з завантаження моделі (рис.3.9).



```
root@raspberrypi:/home/pi# python3 test_voice.py
5% |██████████| 71.3M/1.42G [00:14<03:22, 7.17MiB/s]
```

Рисунок 3.9 – Інтерфейс завантаження навченої нейронної мережі Whisper

Бібліотека Whisper побудована на основі PyTorch і вимагає багато ресурсів (рис. 3.10), що призводить до дуже повільної обробки голосових команд однопалатним комп'ютером.



PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1580	root	20	0	1256828	477620	75168	R	382.5	51.3	13:06.68	python3
1658	root	20	0	9972	3208	2488	R	1.6	0.3	0:04.96	top

Рисунок 3.10 – Використання обчислювальних ресурсів Whisper

Продуктивнішим є використання бібліотеки SpeechRecognition, що містить реалізацію кількох інструментів CMU Sphinx, Google Speech Recognition, Whisper, Microsoft Azure Speech, IBM Speech to Text, Vosk API тощо. За допомогою цього засобу можливо використати Google Speech

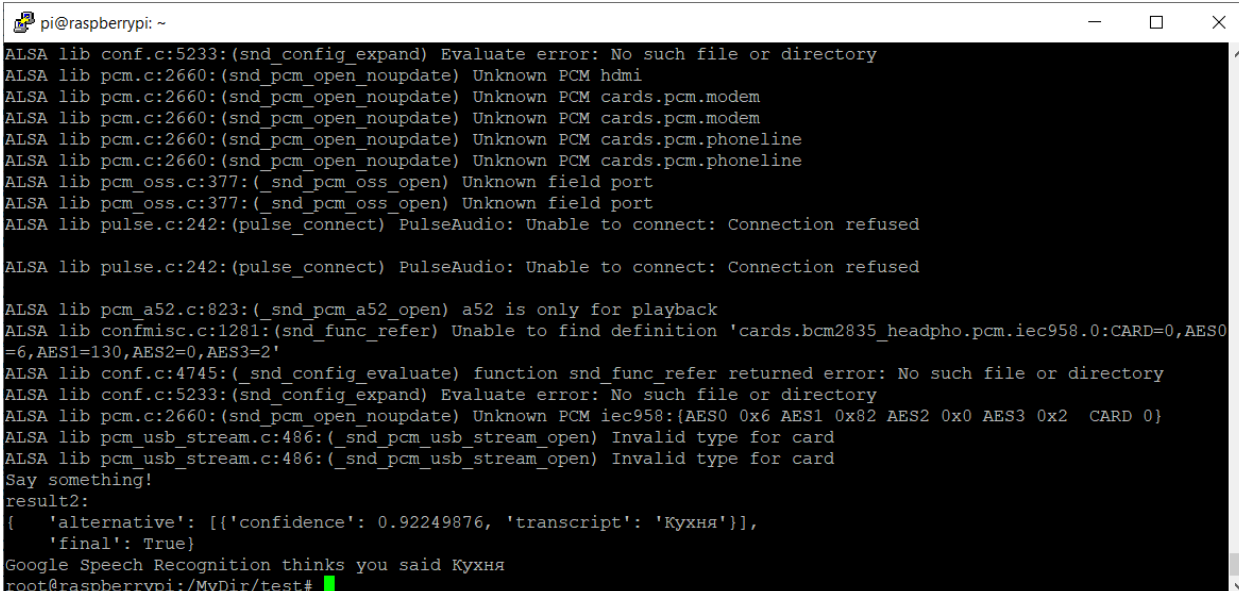
Recognition, який підтримує велику кількість мов та виконує обробку даних на серверах компанії Google і є безкоштовним.

Встановлюється SpeechRecognition також за допомогою pip:

pip install SpeechRecognition

При коректному налаштуванні параметрів для використання Google Speech Recognition бібліотека SpeechRecognition виводить на друк перелік знайдених пристроїв вводу після чого починає аналізувати рівень шуму на доступних.

Реалізація Google Speech Recognition в SpeechRecognition побудована на короткостроковому записі голосу (час задається користувачем) та надсиланні за допомогою API запису на сервер для конвертації в текст. Метод recognize_google(), приймає два параметри: вхідне джерело сигналу (ідентифікатор мікрофона) та мова якою розмовляють. У відповідь від серверу надходить рядок JSON формату з конвертованим текстом та можливими варіаціями (рис. 3.11).



```
pi@raspberrypi: ~
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM hdmi
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm_oss.c:377:(snd_pcm_oss_open) Unknown field port
ALSA lib pcm_oss.c:377:(snd_pcm_oss_open) Unknown field port
ALSA lib pulse.c:242:(pulse_connect) PulseAudio: Unable to connect: Connection refused
ALSA lib pulse.c:242:(pulse_connect) PulseAudio: Unable to connect: Connection refused

ALSA lib pcm_a52.c:823:(snd_pcm_a52_open) a52 is only for playback
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.bcm2835_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:4745:(snd_config_evaluate) function snd_func_refer returned error: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES1 0x82 AES2 0x0 AES3 0x2 CARD 0}
ALSA lib pcm_usb_stream.c:486:(snd_pcm_usb_stream_open) Invalid type for card
ALSA lib pcm_usb_stream.c:486:(snd_pcm_usb_stream_open) Invalid type for card
Say something!
result2:
{ 'alternative': [{'confidence': 0.92249876, 'transcript': 'Кухня'}],
  'final': True}
Google Speech Recognition thinks you said Кухня
root@raspberrypi:/MyDir/test#
```

Рисунок 3.11 – Відображення виводу результату роботи
Google Speech Recognition

Окрім меншого споживання ресурсів у Google Speech Recognition підтримує більшу кількість мов для розпізнавання голосових команд, не потребує завантаження моделі, але у той же час не може працювати без підключення до мережі інтернет.

3.3 Реалізація контролю розумних пристроїв

Розглянуті в першому розділі пристрої контролюються голосовими помічниками шляхом використання сторонніх сервісів. Зазвичай ці сервіси побудовані за архітектурою «If This Then That» (наприклад <https://ifttt.com/>) [27]. Такі сервіси виконують роль посередників, агрегаторів і поєднують API відповідного обладнання або сервісу (рис. 3.12).

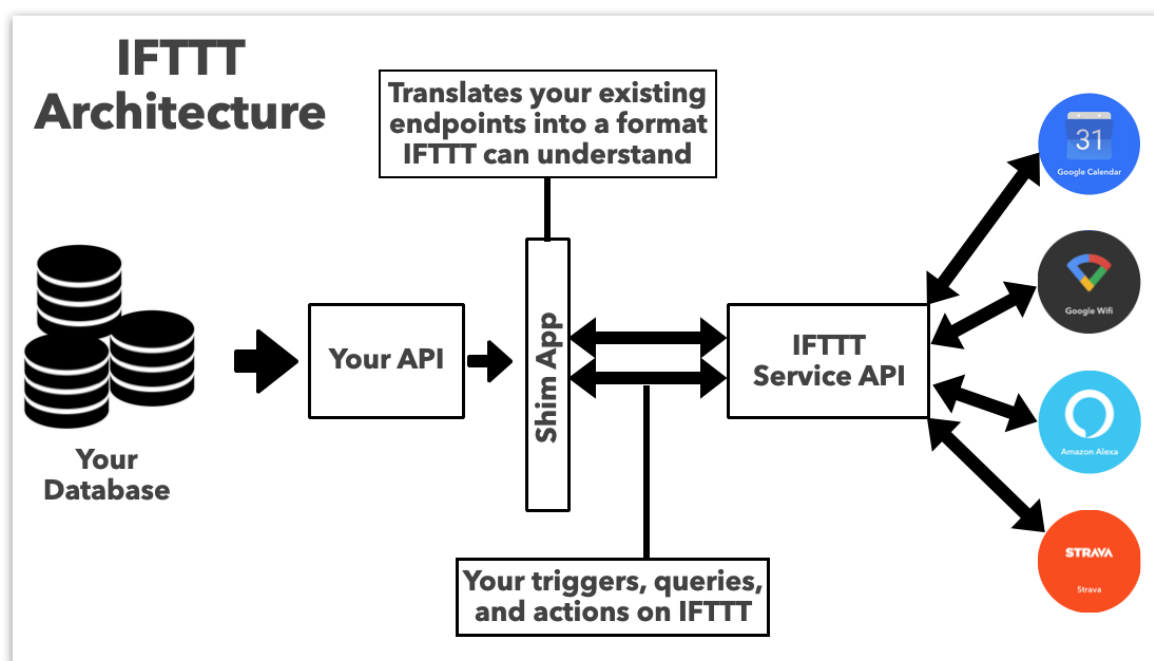


Рисунок 3.12 – Реалізація архітектури

Розглянутий в першому розділі Philips PAR38 має відкрите API представлене на офіційному сайті (Philips Hue), а також має набір шаблонних команд на сервісі <https://ifttt.com/> (рис. 3.13) і може бути інтегроване до програмно-апаратного модулю.

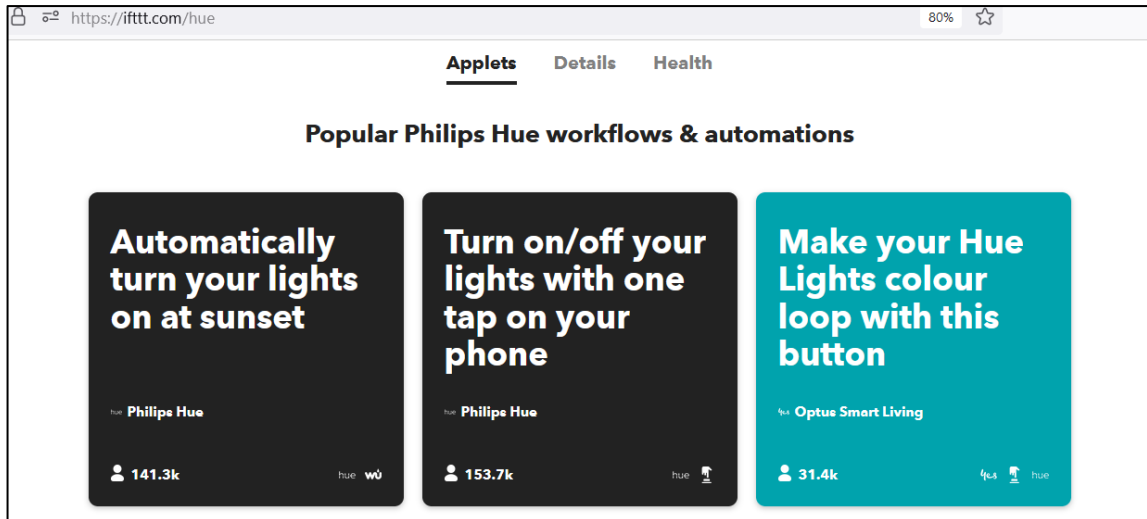


Рисунок 3.13 – Шаблони команд сервісу IFTTT для Philips PAR38

Прямий доступ до API виробника можливо отримати на сайті <https://developers.meethue.com/develop/>.

При наявності пристроїв та реєстрації розробник отримає спеціалізований токен безпеки і може використовувати запити (рис. 3.14).



Рисунок 3.14 – Приклад API Philips Hue

Ecobee3 Lite Smart проаналізований в першому розділі також може інтегруватись з <https://ifttt.com/>. Сервіс містить набір шаблонів «Google Nest Thermostat integrations» які сумісні з Ecobee3 і за наявності додаткових компонентів можуть контролювати мікроклімат певного приміщення.

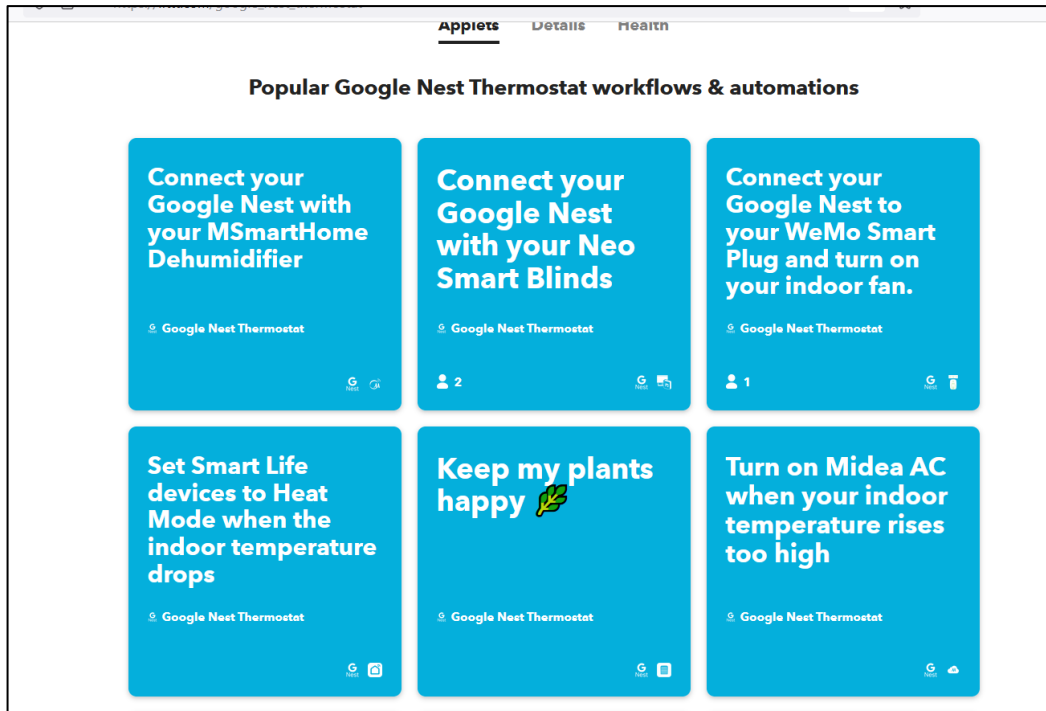


Рисунок 3.15 – Шаблони запитів Google Nest Thermostat Integrations

Запити, що представлені у <https://ifttt.com/> можуть бути реалізовані методом POST або GET. Незалежно від типу користувач має отримати відповідний токен безпеки на сайті виробника обладнання або на сервісі <https://ifttt.com/>.

3.4 Реалізація обробки отриманих команд

Описані вище засоби обробки голосових команд на основі нейронної мережі можуть бути застосовані для конвертації отриманої команди в текст, але після цього необхідна обробка отриманого тексту. Для цього можливо застосувати комбінацію двох підходів: обробки природної мови (Natural Language Processing) та формальних граматик.

Отримана команда при конвертуванні у текст може мати варіації, наприклад «Кухня увімкнути світло» і «Увімкни світло на кухні» з точки зору користувача є однаковими командами. Врахувати таку варіацію можливо розрахувавши редакційну відстань (відстань Левенштейна).

Відстань Левенштейна – це показник, який відображає різницю між двома послідовностями символів. Вона відображає яка мінімальна кількість операцій (видалення, заміни, вставки), необхідна для перетворення однієї послідовності символів в іншу.

Розрахунок редакційної відстані застосовується в широкому колі завдань тому в мові Python реалізовано окремий модуль, що реалізує підрахунок відстані Левенштейна – python-Levenshtein. Завантажити цей модуль можливо засобами вже гаданого pip3:

```
pip3 install python-Levenshtein
```

Модуль python-Levenshtein автоматично генерує матрицю для підрахунку та виводить користувачу вже розрахований результат (рис. 3.16).

```
root@raspberrypi:/# python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Levenshtein import distance
>>> print(distance("кухня", "кухні"))
1
>>> █
```

Рисунок 3.16 – Робота модулю python-Levenshtein

Команда, надана користувачем, повинна мати три обов'язкові елементи: приміщення в якому розміщено пристрій, назва пристрою, та дія, яку необхідно виконати. Така структура дозволяє класифікувати отримані команди як формальну мову. Формальна мова – це множина скінчених послідовностей символів, що описуються правилами певного виду, які називаються граматикою. У цьому контексті перераховані елементи є токенами.

У рамках реалізації обробки отриманих команд не обхідно провести лексичний аналіз та визначити наявні відповідних токенів у вхідному тексті (рис. 3.17). Подальша обробка токенів синтаксичним аналізатором немає

сенсу на даному етапі розробки, але може знадобитись при ускладненні команд.

```
Say something!
result2:
{  'alternative': [  {  'confidence': 0.92249876,
                      'transcript': 'Увімкни світло на кухні'},
                    { 'transcript': 'Увімкнути світло на кухні'},
                    { 'transcript': 'Увімкни світло на кухню'},
                    { 'transcript': 'Увімкни світло кухні'},
                    { 'transcript': 'Увімкнути світло на кухню'}],
  'final': True}
=====
action: Увімкни
device: світло
room: кухні
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'l
```

Рисунок 3.17 – Результат лексичного аналізу

Висновок до третього розділу

В розділі наведено огляд одноплатних комп'ютерів, що можуть виконувати роль основи програмно-апаратного комплексу. Визначено пристрій для запису вхідних голосових команд. Описано хід налаштування одноплатного комп'ютера для подальшого використання одноплатного комп'ютера для розпізнавання мовних команд та керування побутовими пристроями.

Розглянуто методи обробки отриманих команд, а саме визначення редакційної відстані, для забезпечення варіативності та лексичний аналіз для валідації отриманих команд.

4 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ГОЛОСОВОГО КЕРУВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ

4.1 Реалізація апаратного забезпечення

Здійснення керування побутовими приладами можливо з використанням модуля реле на основі GPIO. Серед доступних модулів реле варто виділити Songle, що постачається на платах по декілька модулів одночасно, наприклад, по 4 штуки (рис. 4.1).



Рисунок 4.1 – Плата з 4 модулів Songle

Такі плати містять не тільки модулі релей, а й елементи схеми живлення, світлодіодну індикацію і контакти для підключення ланцюгу живлення (рис. 4.2).

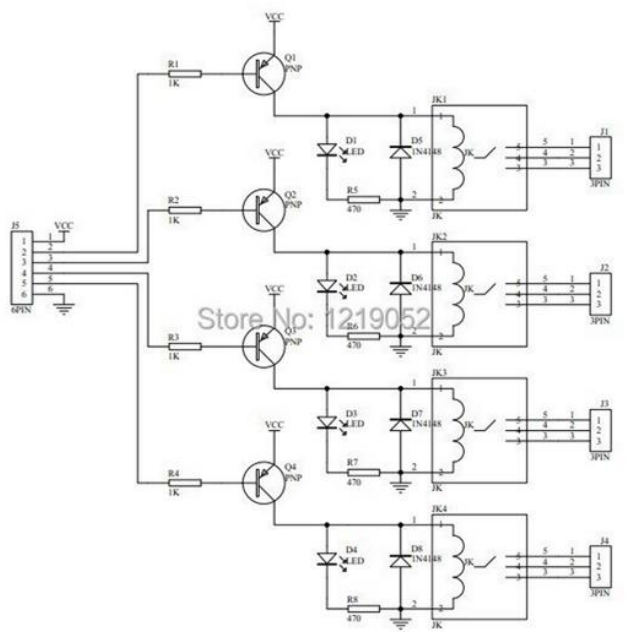


Рисунок 4.2 – Принципова схема плати з 4 модулями реле

Наявність схеми живлення на платі дозволяє її підключення до одноплатного комп'ютера без додаткових компонентів. Перед початком розробки необхідно розробити модель підключення за допомогою спеціалізованого ПЗ, наприклад Fritzing [14]. Підключити плату з 4 модулями реле можливо задіявши 6 входів/виходів GPIO: 1 – вихід живлення; 2 – вихід GND; 3, 4, 5, 6 – виходи для передачі даних: GPIO4, GPIO17, GPIO27, GPIO22 (рис. 4.3).

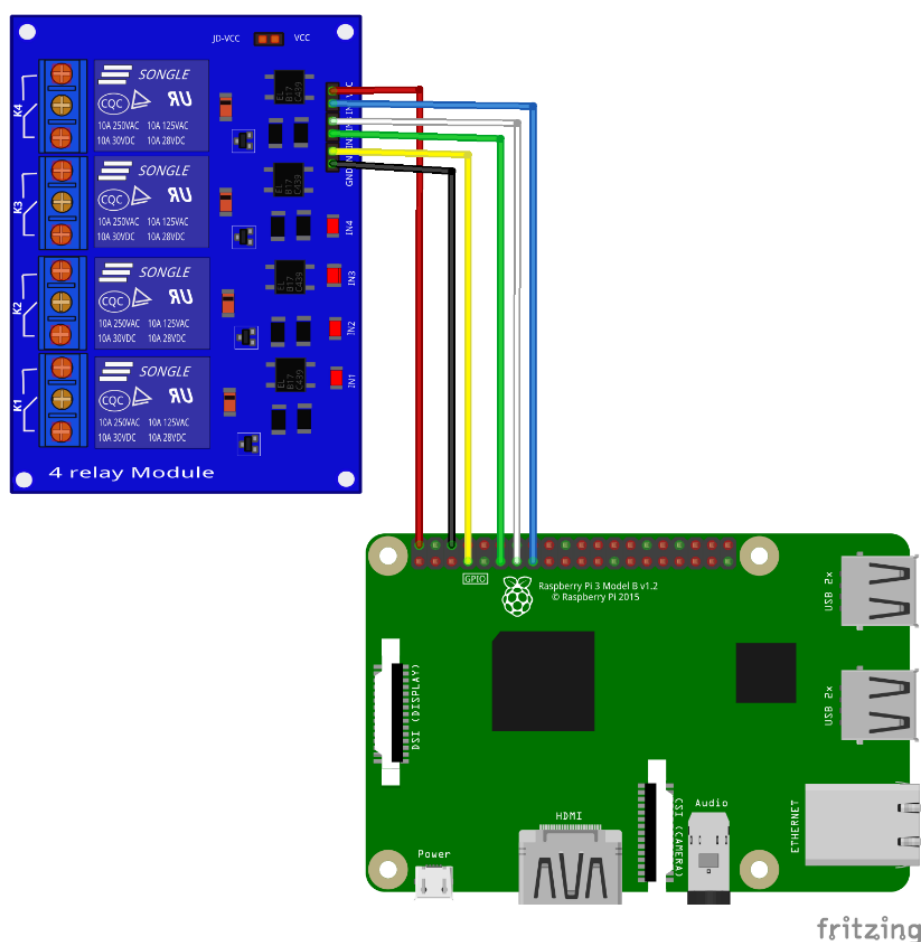


Рисунок 4.3 – Модель підключення модуля реле до одноплатного комп'ютера

Підключення мікрофона для розпізнавання мови можливо за допомогою інтерфейсу USB. Більшість дистрибутивів мають вбудований драйвер для роботи цього інтерфейсу. Вбудований інтерфейс audio 3.5 призначений виключно для виведення звуку та композитного відео.

Апаратно-програмний модуль розпізнавання та інтеграції мови в побутові прилади представлено на рис. 4.4.

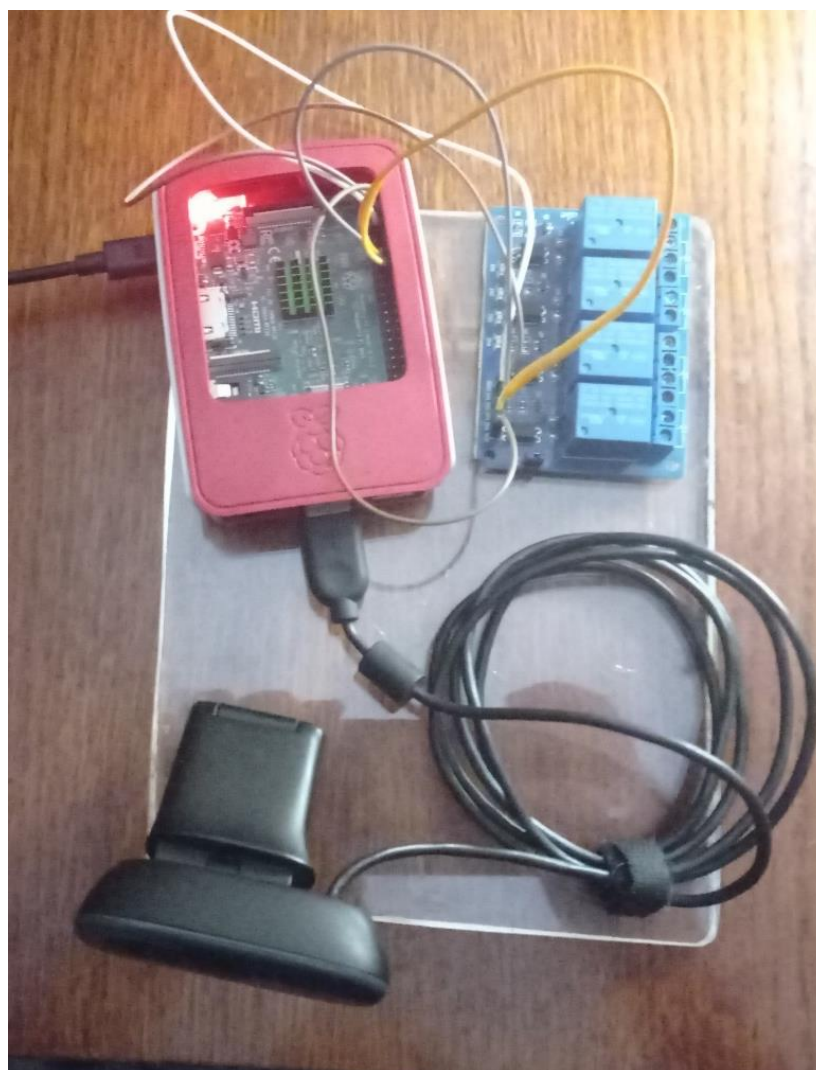


Рисунок 4.4 – Загальний вигляд розробленого апаратно-програмного модуля

4.2 Розробка програмного забезпечення

Програмна складова програмно-апаратного комплексу розпізнавання та інтеграції мови в побутові пристрої має містити кілька компонентів: клас обробки вхідної мови, клас для зчитування доступних команд, клас обробки запитів до API тощо.

Представити загальний алгоритм роботи застосунку можливо за допомогою блок-схеми (рис. 4.5).

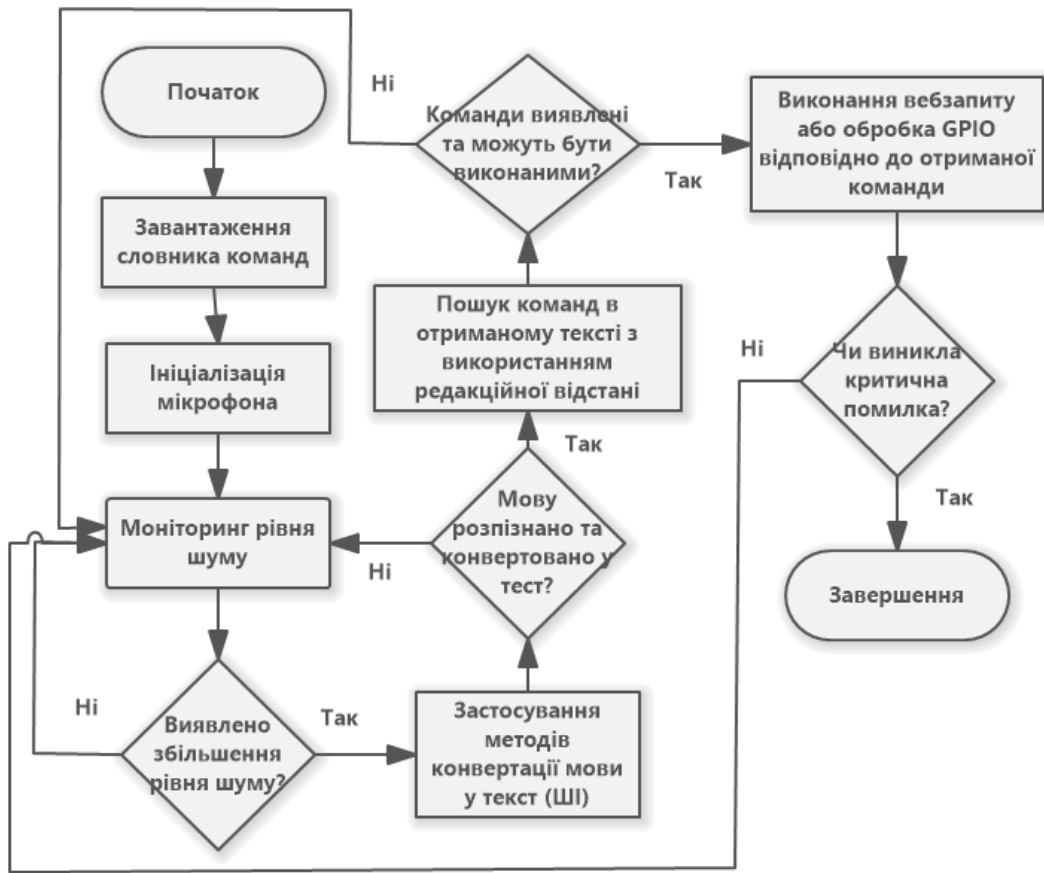


Рисунок 4.5 – Блок-схема алгоритму роботи апаратно-програмного модулю

Алгоритм розрахування редакційної відстані представлено на рис. 4.6.

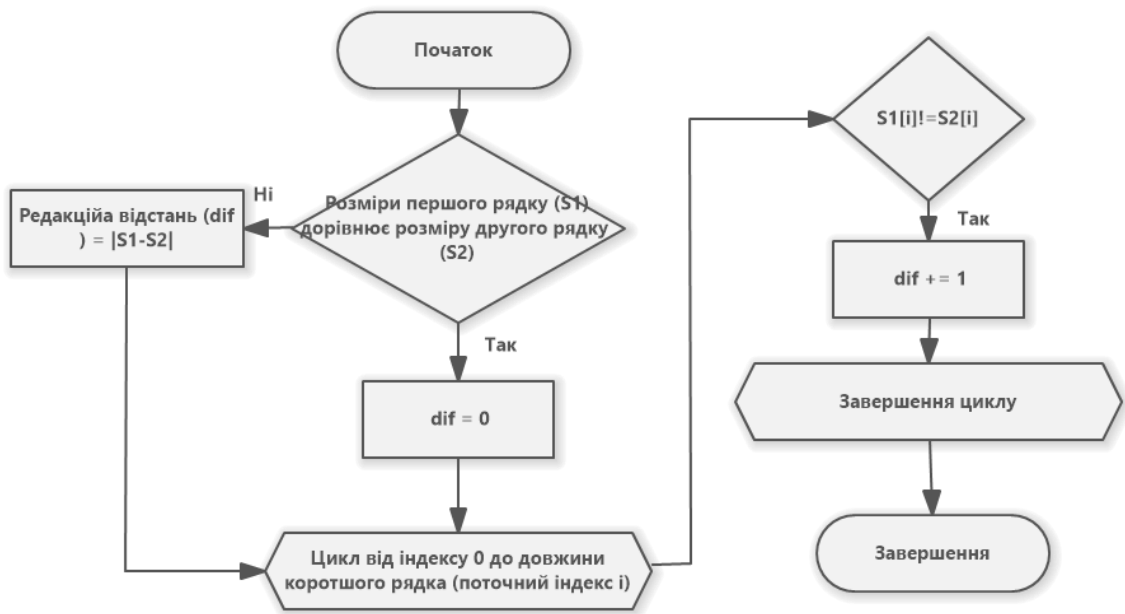


Рисунок 4.6 – Блок-схема визначення редакційної відстані

Корегування словнику з доступними командами потребує наявності додаткового інтерфейсу для взаємодії з користувачем. Наприклад, зручно додавати нові пристрої та прописувати елементи API за допомогою вебінтерфейсу. Перед розробкою цього інтерфейсу було побудовано Sitemap (рис. 4.7).

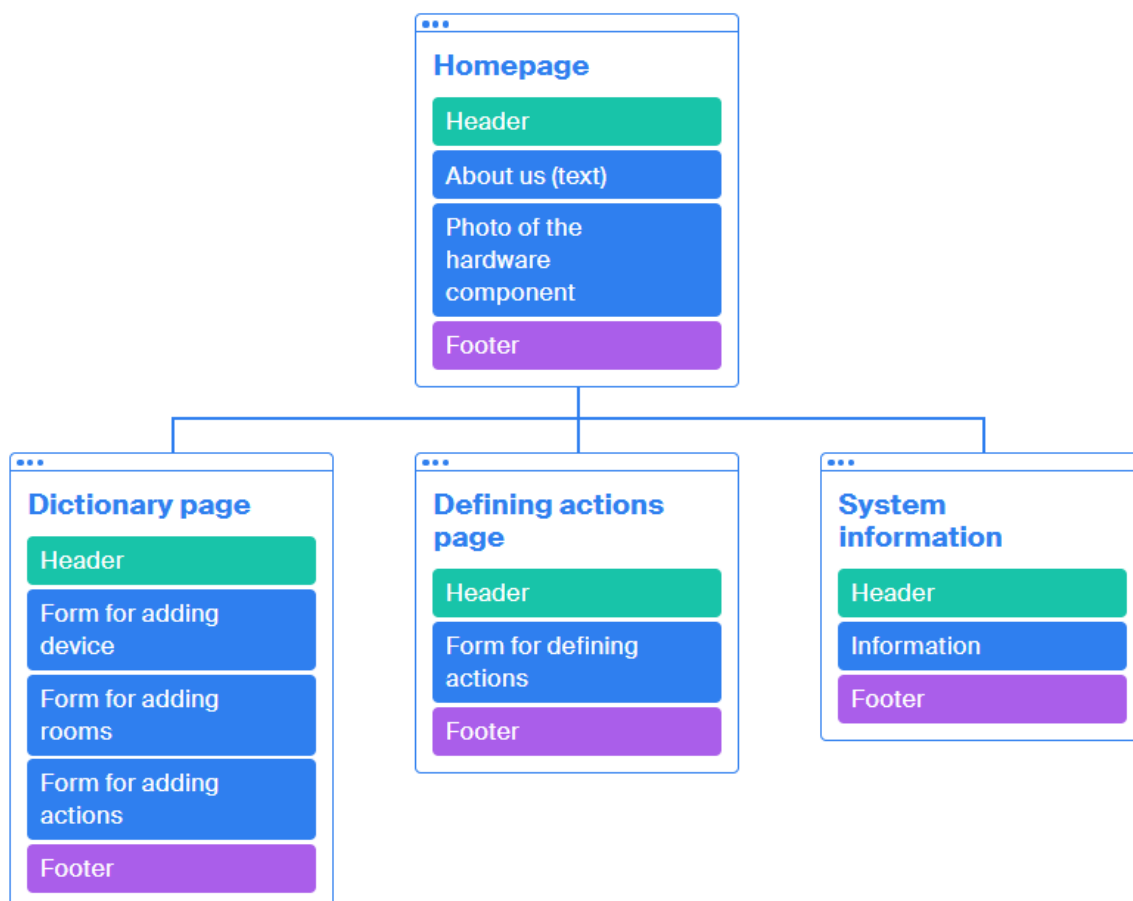


Рисунок 4.7 – Sitemap для вебінтерфейсу редагування словнику апаратно-програмного модулю

Вебінтерфейс має просту структуру з чотирьох сторінок: «Домашня» основна інформація про проєкт; «Сторінка словнику» сторінка з формами для додавання приміщень, приладів та команд; «Визначення команд» прив'язка певних приладів до кімнат та доступних команд (у якості команд можуть бути посилання API або дія з інтерфейсом GPIO); «Системна інформація» сторінка з відображенням поточної інформації про пристрій.

Діаграму класів для апаратно-програмного модулю розпізнавання та інтеграції мови в побутові прилади наведено на рис. 4.8.

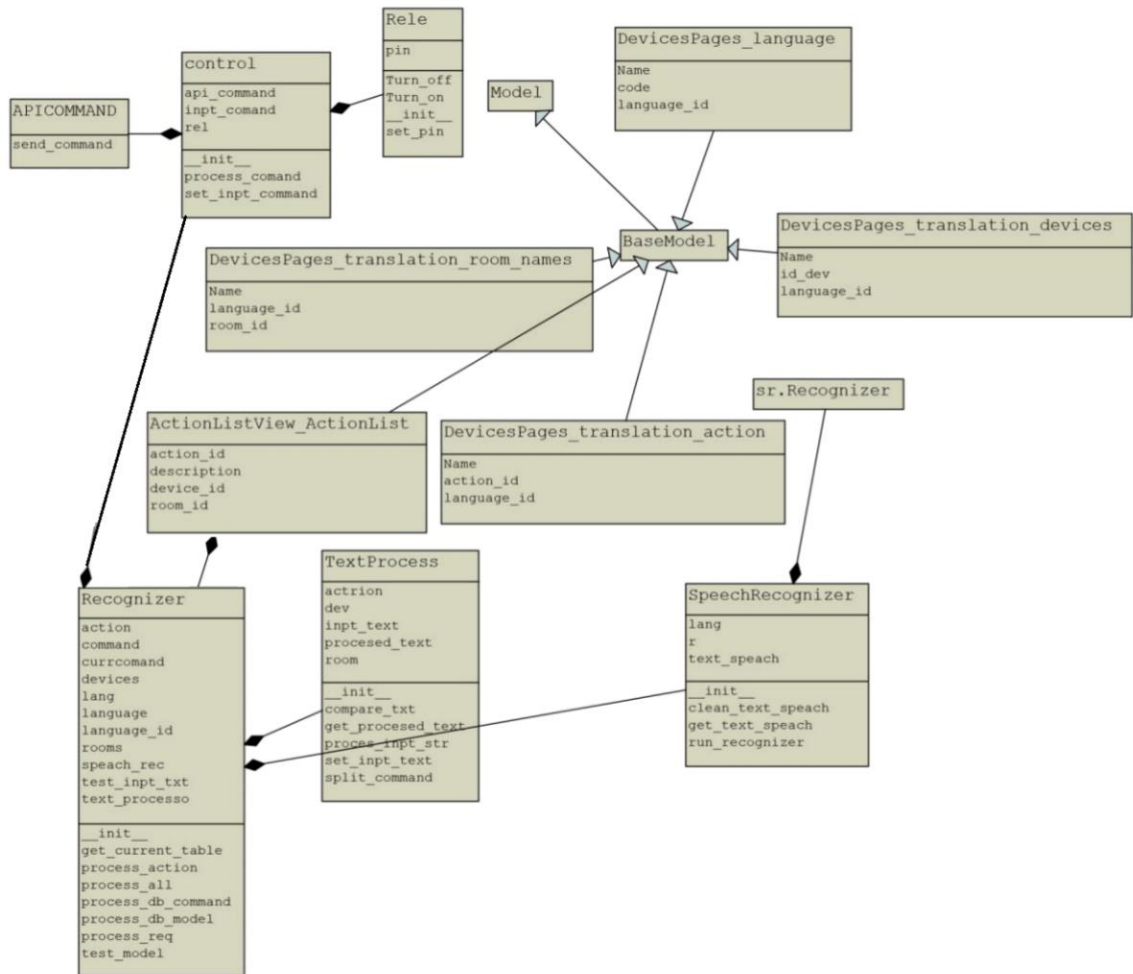


Рисунок 4.8 – Діаграма класів програмного забезпечення розпізнавання
голосу

Після перетворення мови в текст необхідно визначити, чи є отриманий текст командою та чи можливо виконати отриману команду. Реалізація цієї функції вимагає наявності словника, що може зберігатись у вигляді бази даних (БД). Перед створенням цієї БД необхідно спроектувати її за допомогою використання Entity-Relationship Diagram (ERD, рис. 4.9)

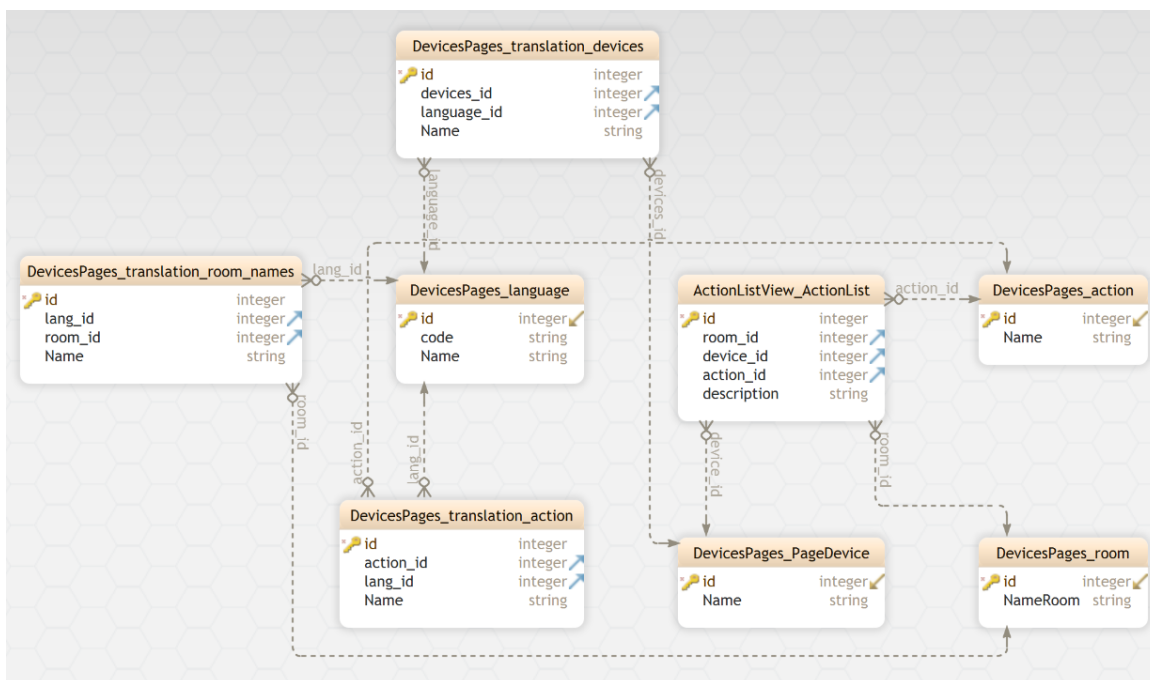


Рисунок 4.9 – ERD-діаграма словника команд

Особливість словника команд для апаратно-програмного модулю розпізнавання та інтеграції мови в побутові прилади полягає в необхідності обробки команд на кількох мовах. Для цього в представленій БД реалізовано таблиці: `DevicesPages_translation_action`, `DevicesPages_translation_devices`, `DevicesPages_translation_room_names`, де `DevicesPages` – ім'я застосунку, сутності фреймворку Django.

У якості системи керування базами даних (СКБД) необхідне використання SQLite, що застосовується у фреймворку Django для реалізації функцій об'єктно-реляційної моделі.

Фреймворк Django забезпечує можливість використання вебінтерфейсу для формування словника команд. В програмно-апаратному комплексі розпізнавання та інтеграції мови в побутові пристрої реалізовано декілька сторінок: авторизація, редагування словника та визначення команд.

Авторизація забезпечується вбудованими засобами Django на розробленій вебсторінці (рис. 4.10).

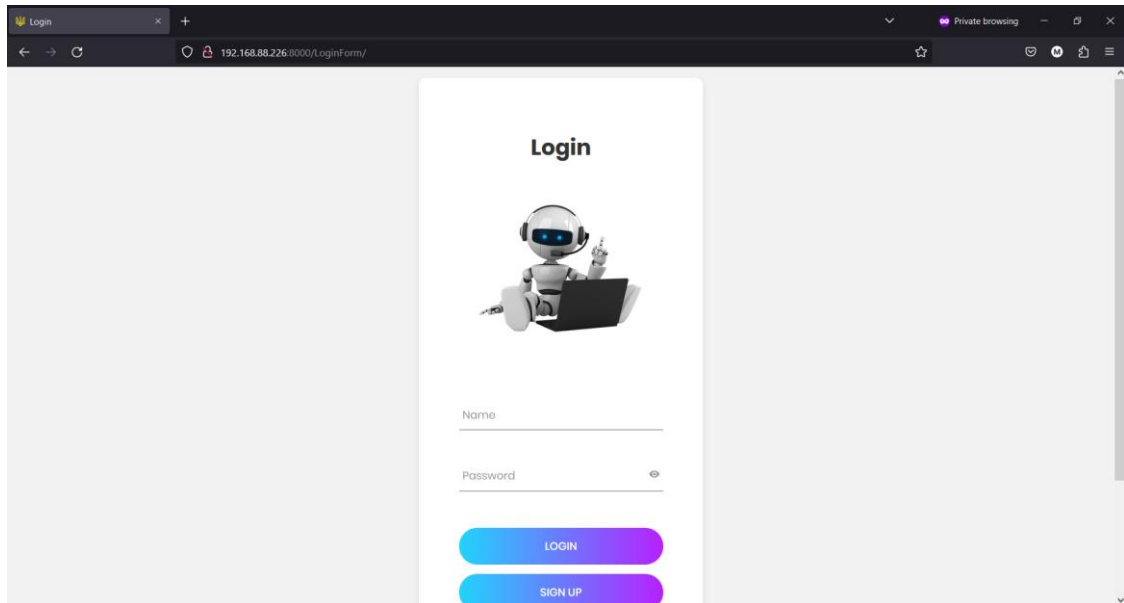


Рисунок 4.10 – Сторінка авторизації для редагування словника команд

Користувач має визначити яку саме сутність являє собою текст: приміщення де розташований побутовий пристрій, назва побутового пристрою, команда. Клас сутності є класом токена при обробці отриманої команди на основі методів формальних граматик.

Розроблений вебінтерфейс відображає наявні в словнику сутності на двох мовах: українській та англійській. Це забезпечує можливість введення можливість обробки отриманих команд українською та англійською мовами. Розроблена БД для зберігання словнику (рис. 4.9) дозволяє додавати більше мов і відповідно розширяти наявний інтерфейс.

Після конвертації голосових команди відбувається перевірка належності отриманого тексту до переліку доступних команд з врахуванням редакційної відстані. Користувач може розширити перелік доступних команд додавши їх тестове значення у словник за допомогою вебінтерфейсу (рис. 4.11).

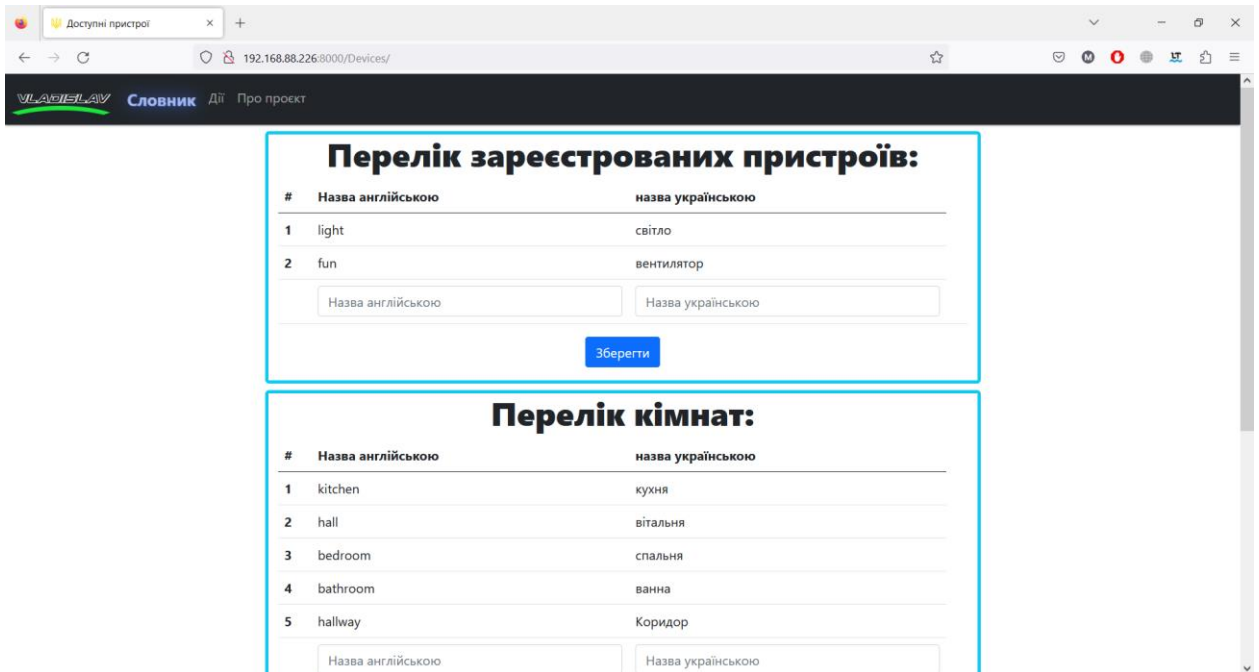


Рисунок 4.11 – Сторінка редагування словнику команд

Введення тесту команди дозволяє її розпізнати, але обробка голосового запиту потребує комбінування сутностей: визначення приміщення, визначення пристрою, визначення дії та короткий опис коду обробки цієї дії (запит API або звернення до GPIO). Реалізувати перевірку правильності введених коду для обробки дій можливо шляхом застосування регулярного виразу: $(pin\ [d]\ \{1,2\}(on|off))|(?:https?):\backslash\backslash(\w+:\w*)?(\S+)(:\d+)?$.

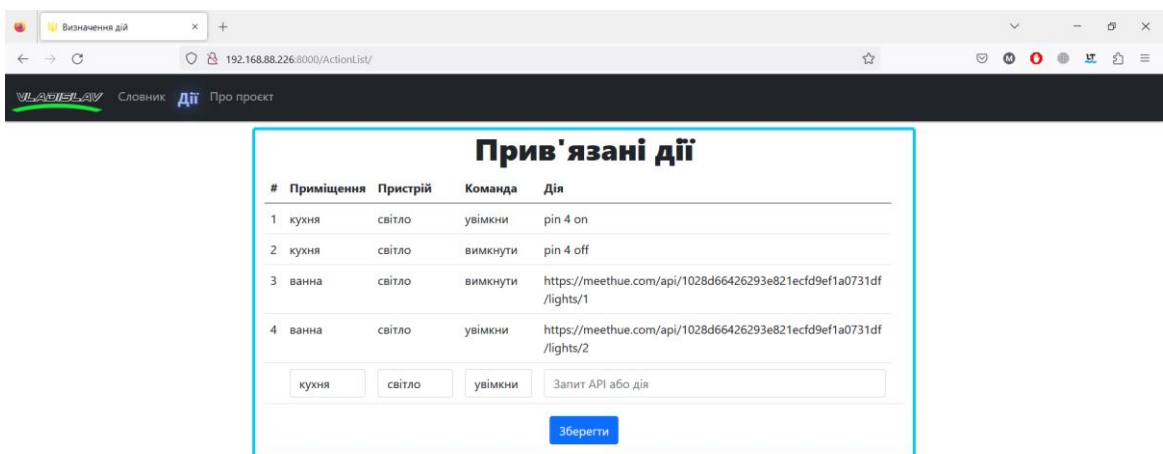


Рисунок 4.12 – Сторінка визначення команд

Розроблений інтерфейс може виконати надсилання на заданий вихід GPIO «0» або «1» в залежності від заданої користувачем команди. Модуль реле замикає ланцюг у випадку отримання 0, тому відповідає команді «on», а у випадку «off» надходить сигнал «1». Мова програмування Python дозволяє працювати з інтерфейсом GPIO засобами бібліотеки RPi.GPIO. У випадку використання у якості коду обробки адреси API використовується модуль urllib.

Код програми для керування апаратно-програмним модулем розпізнавання та інтеграції мови в побутові пристрої наведений у додатку А.

4.3 Тестування розробленого комплексу

Почати тестування розробленого програмного та апаратного забезпечення необхідно з тестування класу реле. Код для роботи з інтерфейсом GPIO доволі простий і включає метод GPIO.output(). Протестувати працездатність підключених виводів та коректність написаного коду можна встановивши статичні параметри «GPIO.output(4, 0)» і відслідкувавши індикацію модулю реле. У разі вдалого підключення відповідно до представлені вище моделі має спрацювати індикація першого виходу (рис. 4.13).

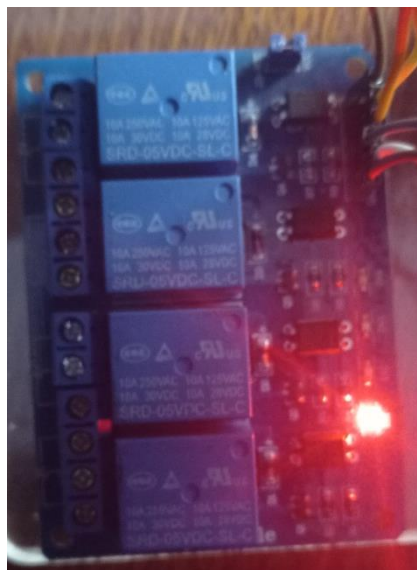
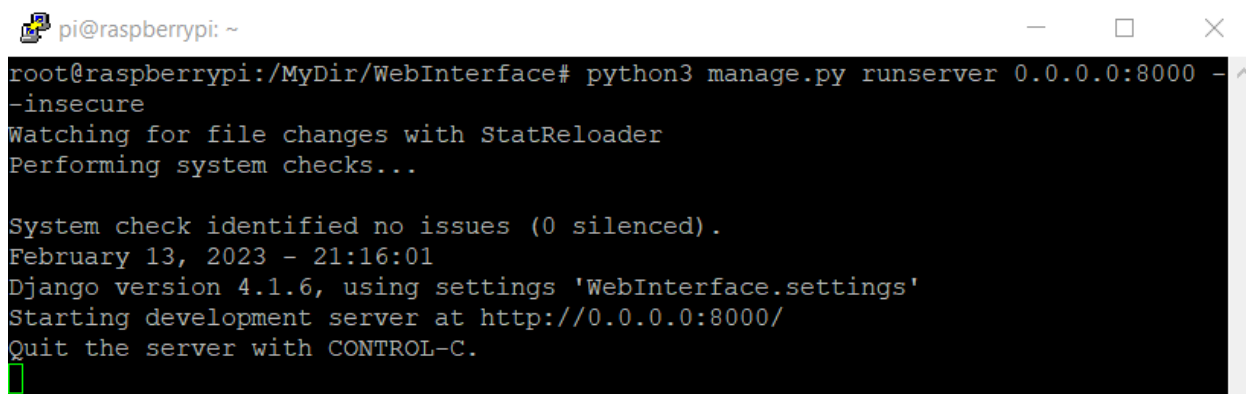


Рисунок 4.13 – Індикація увімкненого модулю реле

Перевірити працездатність вебсерверу можливо, запустивши застосунок вручну з консолі командою: `python3 manage.py runserver 0.0.0.0:8000 --insecure`. У разі коректного встановлення фреймворку та налаштувань серверу `python` виведе статус серверу (рис. 4.14)



```
pi@raspberrypi: ~  
root@raspberrypi:/MyDir/WebInterface# python3 manage.py runserver 0.0.0.0:8000 --insecure  
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
February 13, 2023 - 21:16:01  
Django version 4.1.6, using settings 'WebInterface.settings'  
Starting development server at http://0.0.0.0:8000/  
Quit the server with CONTROL-C.  
█
```

Рисунок 4.14 – Вивід даних запуску вебсерверу

Наступним важливим кроком є ручне тестування розробленого вебінтерфейсу. Цей процес вимагає додавання в словник нового приміщення, обладнання та дії. Наприклад, нове приміщення: гараж; нове обладнання: підіймач; дія: підняти (рис 4.15).

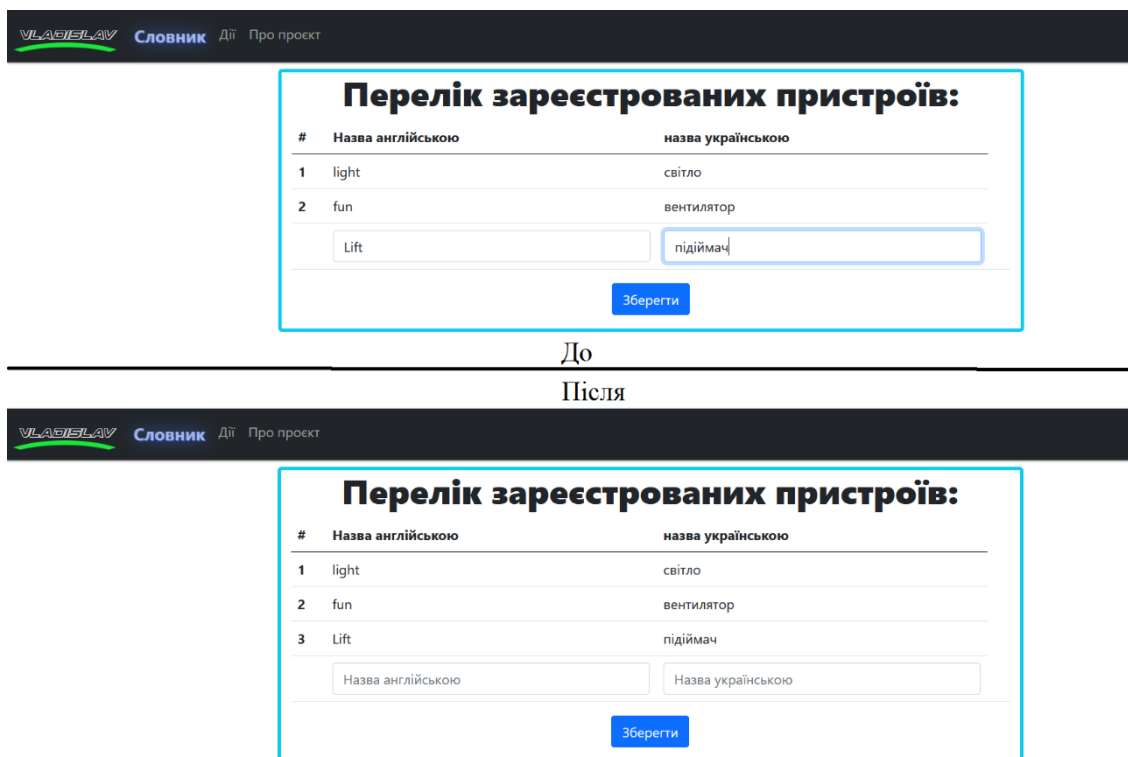


Рисунок 4.15 – Тестування додавання сутності в словник

Останнім кроком редагування словнику є створення взаємозв'язку між пристроєм, кімнатою та дією. У разі коректного додавання нових сутностей вони відобразатимуться на відповідній сторінці (рис. 4.16).

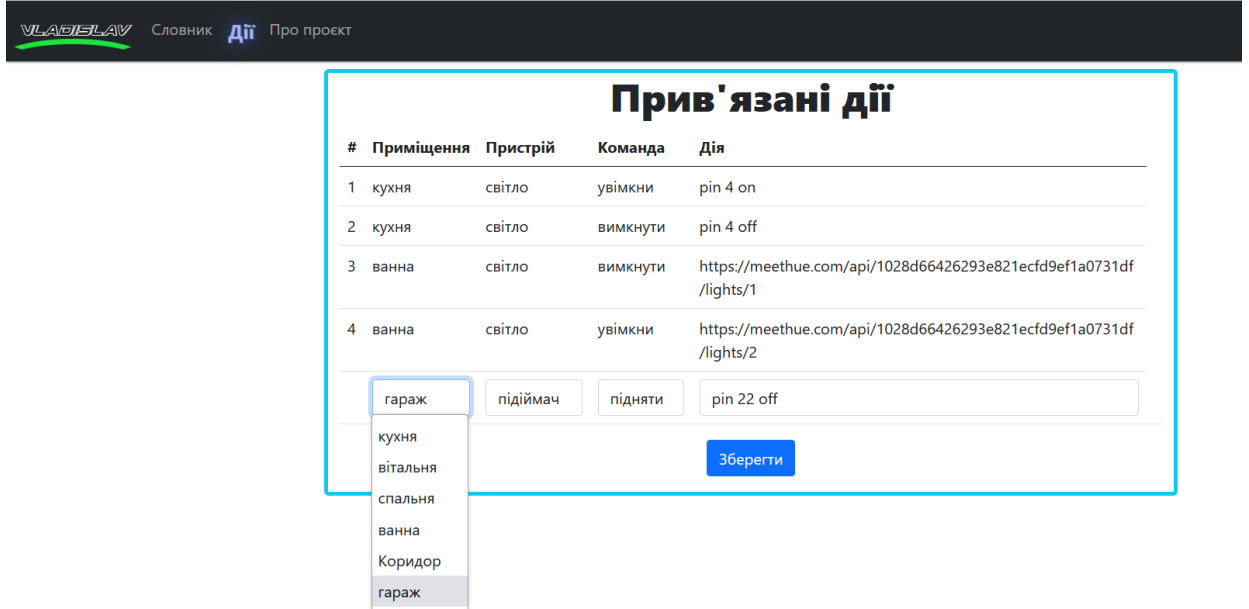


Рисунок 4.16 – Інтерфейс прив'язки дії

Додавши відповідні сутності в словник перевірити працездатність ШІ, модулю підрахунку редакційної відстані та лексичного аналізу можливо давши голосову команду «Підняти підіймач в гаражі» розроблено програмно-апаратному комплексу. У результаті коректної роботи можна побачити розпізнаний текст та виділені токени (рис. 4.17)

```
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES1 0x82 AES2 0x0 AES3 0x2 CARD 0}
ALSA lib pcm_usb_stream.c:486:( _snd_pcm_usb_stream_open) Invalid type for card
ALSA lib pcm_usb_stream.c:486:( _snd_pcm_usb_stream_open) Invalid type for card
Say something!
result2:
{ 'alternative': [ { 'confidence': 0.92249876,
                    'transcript': 'підняти підіймач в гаражі'},
                  {'transcript': 'підняти підіймач в Парижі'},
                  {'transcript': 'підняти підіймач в грязі'},
                  {'transcript': 'підняти підніме в гаражі'},
                  {'transcript': 'підняти підніматись в гаражі'}],
  'final': True}
=====
action: підняти
device: підіймач
room: гаражі
```

Рисунок 4.17 – Тестування роботи нейронної мережі, розрахунку редакційної відстані та лексичного аналізу

Висновок до четвертого розділу

В розділі наведено опис процесу проектування апаратного та програмного забезпечення. Представлено модель апаратного забезпечення розроблену засобами Fritzing. Продемонстровано розроблене апаратне забезпечення.

Виконано проектування програмного забезпечення. Наведено опис алгоритмів роботи програмно-апаратного комплексу та окремо відображено алгоритм визначення редакційної відстані для забезпечення варіативності команд.

Наведено архітектуру розробленого програмного забезпечення засобами UML. Представлено модель БД для зберігання словника команд. Описано особливості реалізації вебінтерфейсу.

Представлено результати тестування розробленого програмно-апаратного забезпечення. Визначено можливість перевірки роботи модулю реле, форм для додавання елементів у словник та прив'язки їх з кодом виконання дій.

ВИСНОВКИ

В результаті роботи створена система автоматизованого керування побутовими приладами, що виконують свої функції тільки після ідентифікації голосових команд.

Можна зробити висновок, що дизайн і розробка, яку було створено за допомогою голосової команди з додаванням функції розпізнавання голосу та мови розпізнавання, може полегшити повсякденну життєдіяльність людини. Автоматизоване управління можна застосувати до «цифрового будинку» для керування електронними пристроями, такими як освітлення, вентилятори, телевізори, кондиціонери та ін., керування якими базується на голосових командах. Його також можна застосовувати для автоматичного керування дверима. Крім того, також можна застосовувати голосові команди в інтелектуальних лікарняних палатах для контролю лежачих пацієнтів у лікарні та для допомоги у виклику медперсоналу. Це також може допомогти користувачам з обмеженими можливостями, наприклад, такими, хто пересувається у інвалідному візку.

Для досягнення поставленої мети розробки зазначеного комплексу було виконано такі завдання:

- 1) розглянуто сучасні системи керування побутовими приладами;
- 2) проаналізовано недоліки існуючих систем;
- 3) розроблено апаратну частину імплементації голосових команд до побутових приладів;
- 4) розроблено програмну частину відправки голосових команд до побутових приладів;
- 5) розроблено комп'ютерну модель для демонстрації роботи.

Практичне значення роботи полягає в тому, що дана розробка дозволяє розпізнавати мову керування та інтегрувати голосові команди в побутові пристрої.

Робота пройшла апробацію під час XXV Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 07–11 листопада 2022 р.).

В майбутньому слід очікувати пристроїв з можливістю управління з підтримкою стандарту LTE. У перспективі до керування побутовими приладами доцільно розвивати штучний інтелект, додавати машинне навчання, розпізнавання осіб та обробку великих даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Глибовець А. М., Моголівський В. О. Аналіз систем підтримки розумного будинку. *Control systems & computers*. Міжнародний науково-навчальний центр інформаційних технологій і систем НАН. 2019. Вип. 5. С. 30–37.
2. Ной М. В. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*. 2018. Vol. 37, № 1. P. 81–88.
3. Розумна розетка Аоусоср 13А Alexa : вебсайт. URL: <https://izi.ua/uk/p-48350996-umnaya-rozetka-aousocr-13a-alexa-dlya-britanskikh-rozetok> (дата звернення: 04.01.2023).
4. Philips PAR38 – E26 smart bulb : вебсайт. URL: <https://www.philips-hue.com/en-us/p/hue-white-par38---e26-smart-bulb/046677476816> (Last accessed: 04.01.2023).
5. Ecobee3 lite Smart: вебсайт. URL: <https://z-wave.com.ua/ua/p1410633795-umnyj-termostat-ecobee.html> (Last accessed: 04.01.2023).
6. Полякова О. В. Класифікація функціональних складових елементів системи інтелектуального керування середовищем при проектуванні житла. *Вісник Київського національного університету технологій та дизайну. Технічні науки*. 2016. Вип. 4, № 100. С. 133–143.
7. Контролери TECH Sterowniki. Вебсайт виробника. URL: <https://www.techsterowniki.pl/> (дата звернення: 05.01.2023).
8. Fibaro Single Switch. URL: <https://worldvision.com.ua/ru/fibaro-single-switch-dlya-apple-homekit-fgbhs-213/> (Last accessed: 05.01.2023).
9. AUBESS 2. URL: <https://www.joom.com/uk/products/6216fa4a6ddeaf0156ced2b2> (Last accessed: 05.01.2023).

10. Single Board Computer Market : Press Release. Publ. 2022, Nov. 11. URL: https://www.marketwatch.com/press-release/single-board-computer-market-size-trends-growth-status-share-research-and-forecast-2030-2022-11-11?mod=search_headline (Last accessed: 05.01.2023).
11. Засоби розпізнавання мови URL: <https://pypi.org/project/SpeechRecognition/> (дата звернення: 05.01.2023).
12. Веселовський В. Д., Журавська І. М. Методи ідентифікації голосу. Могилянські читання – 2022 : тези доп. XXV Всеукр. наук.-метод. конф. Миколаїв, 7–11 листоп. 2022 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2022. С. 60–62.
13. Fritzing: electronic made easy. URL: <https://fritzing.org/> (Last accessed: 05.01.2023).
14. Shcherban S., Liang P., Li Z., Yang C. Multiclass classification of four types of UML diagrams from images using deep learning. 2021. p. 55–62.
15. Інструментарій SoftwareIdeasModeler URL: <https://www.softwareideas.net/> (дата звернення: 10.01.2023).
16. Ваврук Є., Лашко О., Попович Р. Алгоритми та засоби обробки сигналів : навч. посіб. Львів : СПОЛОМ, 2021. 240 с.
17. Qian Y.-M., Xiang X. Binary neural networks for speech recognition. May *Frontiers of Information Technology & Electronic Engineering* 2019. Vol. 20, Is. 5. Pp. 701–715. DOI: 10.1631/FITEE.1800469.
18. Wingfield C., Zhang C., Devereux B. et al. On the similarities of representations in artificial and brain neural networks for speech recognition : Preprint. June 2022. DOI: 10.1101/2022.06.27.497678.
19. Patil J., Shewale A., Bhushan E. A voice based assistant using Google dialogflow and machine learning. *International Journal of Scientific Research in Science and Technology*. 2021, May. DOI: 10.32628/IJSRST218311.
20. Open AI whisper URL: <https://github.com/openai/whisper>(Last accessed: 05.01.2023).

21. Google Speech-to-Text API. URL: <https://cloud.google.com/speech-to-text> (Last accessed: 05.01.2023).
22. Orange Pi 5. URL: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html> (Last accessed: 05.01.2023).
23. ODROID-C4. URL: <https://www.hardkernel.com/shop/odroid-c4/> (Last accessed: 05.01.2023).
24. Однопалатний комп'ютер Raspberry Pi 3b. URL: <https://arduino.ua/prod1449-raspberry-pi-3-b> (Last accessed: 09.01.2023).
25. Howser G., Howser G. Raspberry Pi operating system. *In book: Computer Networks and the Internet: A Hands-On Approach*. 2020. P. 119–149.
26. Vincent W. S. Django for Beginners: Build websites with Python and Django. WelcomeToCode, 2021. 296 p.
27. If This Then That. URL: <https://ifttt.com/> (Last accessed: 22.02.2023).
28. Voice control circuit breaker: Intelligent voice control circuit breaker for household appliances: Patent 202011037758 India. Publ. 09/10/2020. The Patent Office Journal No. 41/2020.
29. Ma'arif M., Rijanto T., Suprianto B. Validation of voice recognition in various Google voice languages using voice recognition module V3 based on microcontroller. *Proc. of the 2020 the Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*. October 2020. DOI: 10.1109/ICVEE50212.2020.9243184.
30. Chan Z. Yu., Shum P. Smart office: A voice-controlled workplace for everyone. *Proc. of the 2nd International Symposium*. September 2018. DOI: 10.1145/3284557.3284712.
31. Gao T., Pan Q., Zhou J. A novel attention-guided generative adversarial network for Whisper-to-Normal speech conversion. Springer, 2023. Cognitive Computation. DOI: 10.1007/s12559-023-10108-9.

ДОДАТОК А

Код програми

Class SpeechRecognizer

```
import speech_recognition as sr

class SpeechRecognizer:
    def __init__(self):
        #self.lang="en-US"
        self.text_speech=""
        self.lang="uk-UA"
        self.r = sr.Recognizer()
    def run_recognizer(self):
        while 1:
            with sr.Microphone() as source:
                print("Say something!")
                audio = self.r.listen(source,5,5)
            try:
                txt_answer=self.r.recognize_google(audio,language=self.lang)
                self.text_speech=txt_answer
            except sr.UnknownValueError:
                print("Google Speech Recognition could not understand audio")
            except sr.RequestError as e:
                print("Could not request results from Google Speech Recognition
service; {0}".format(e))
        def get_text_speech(self):
            return self.text_speech
        def clean_text_speech(self):
            self.text_speech=""
```

Class Rele

```
import RPi.GPIO as GPIO

class Rele:

    def __init__(self):
        self.pin=None
        GPIO.setmode(GPIO.BCM)

    def set_pin(self,pin):
        self.pin=pin
        GPIO.setup(self.pin, GPIO.OUT, initial=1)

    def Turn_on(self):
        GPIO.output(self.pin, 0)

    def Turn_off(self):
        GPIO.output(self.pin, 1)

from peewee import *

class BaseModel(Model):
    class Meta:
        database = SqliteDatabase("/WebInterface/db.sqlite3") # соединение с
        базой, из шаблона выше

# Определяем модель исполнителя

class DevicesPages_language(BaseModel):
    language_id = AutoField(column_name='id')
    code = TextField(column_name='code')
    Name= TextField(column_name='Name')
    class Meta:
        table_name = 'DevicesPages_language'

class DevicesPages_translation_devices(BaseModel):
    id_dev = IntegerField(column_name='devices_id')
    language_id = IntegerField(column_name='language_id')
```

```
Name = TextField(column_name='Name')

class Meta:
    table_name = 'DevicesPages_translation_devices'

class DevicesPages_translation_room_names(BaseModel):
    language_id = IntegerField(column_name='language_id')
    rooms_id = IntegerField(column_name='rooms_id')
    Name = TextField(column_name='Name')

class Meta:
    table_name = 'DevicesPages_translation_room_names'

class DevicesPages_translation_action(BaseModel):
    action_id = IntegerField(column_name='actions_id')
    language_id = IntegerField(column_name='language_id')
    Name = TextField(column_name='Name')

class Meta:
    table_name = 'DevicesPages_translation_action'

class ActionListView_ActionList(BaseModel):
    rooms_id = IntegerField(column_name='rooms_id')
    device_id = IntegerField(column_name='device_id')
    action_id = IntegerField(column_name='action_id')
    description = TextField(column_name='description')

class Meta:
    table_name = 'ActionListView_ActionList'

from Levenshtein import distance

class TextProcess:
    def __init__(self,dev,room,action):
        self.procesed_text={ }
        self.inpt_text=""
        self.dev=dev
```



```
self.room=room
self.actrion=action
def split_command(self):
    splited_command=[]
    temp_list_command=self.inpt_text.split(" ")
    for comand in temp_list_command:
        splited_command.append(comand.replace(" ",""))
    return splited_command
def set_inpt_text(self,inpt_text):
    self.inpt_text=inpt_text
def compare_txt(self,inpt_list,txt):
    for element_inpt_list in inpt_list.keys():
        #print(distance(element_inpt_list, txt),element_inpt_list)
        if len(txt)>3 and distance(element_inpt_list, txt)<=2 :
            return inpt_list[element_inpt_list]
        elif len(txt)<=3 and distance(element_inpt_list, txt)<=1 :
            return inpt_list[element_inpt_list]
    return False
def proces_inpt_str(self):
    for inpt_part in self.split_command():
        dev=self.compare_txt(self.dev,inpt_part)
        room=self.compare_txt(self.room,inpt_part)
        action=self.compare_txt(self.actrion,inpt_part)
        if dev:
            self.procesed_text['dev']=dev
        if room:
            self.procesed_text['room']=room
        if action:
```

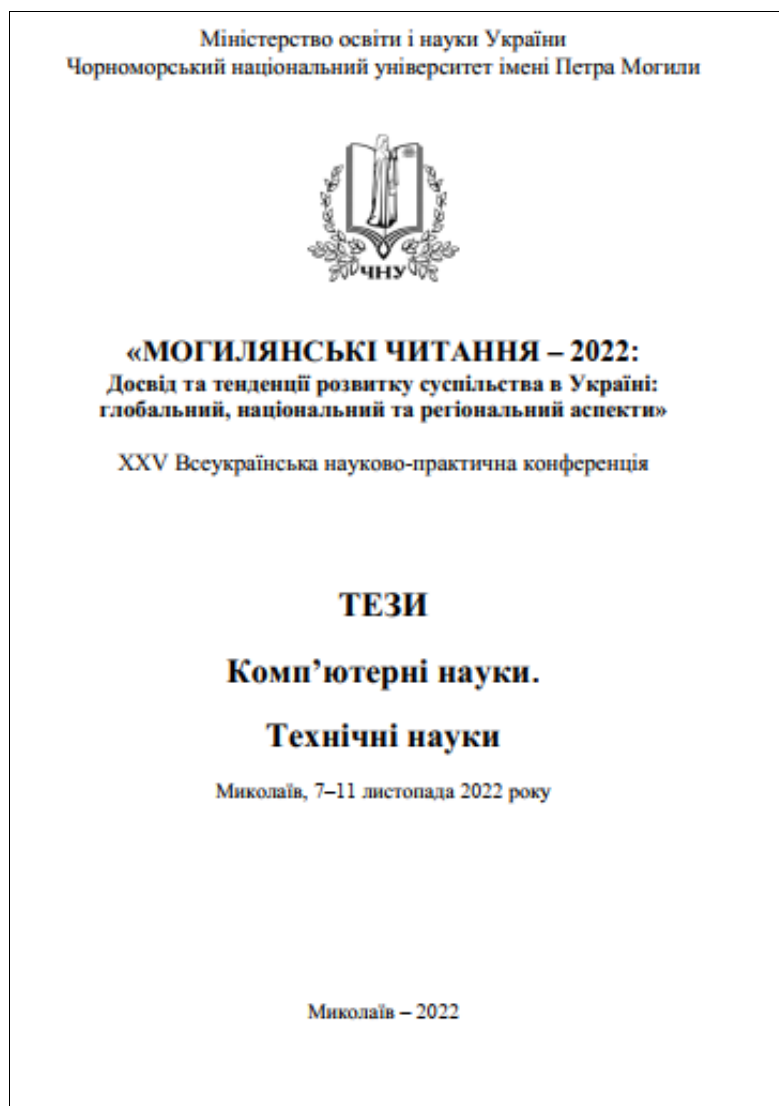
```
        self.procesed_text['action']=action

def get_procesed_text(self):
    self.proces_inpt_str()
    return self.procesed_text

from Rele import*
from WebRequest import *
class control:
    def __init__(self):
        self.inpt_comand=""
        self.api_command=APICOMMAND()
    def set_inpt_command(self,inpt_comand):
        self.inpt_comand=inpt_comand
    def process_comand(self):
        if re.fullmatch(r'^(http|https)', self.inpt_comand.lower()):
            self.api_command.send_command(self.inpt_comand)
        else:
            rele_command=self.inpt_comand.lower().split(" ")
            self.rel=Rele()
            self.rel.set_pin(int(rele_command[1]))
            if rele_command[2]=='on':
                self.rel.Turn_on()
            else:
                self.rel.Turn_off()
```

ДОДАТОК Б

Публікації за темою роботи



ПІДСЕКЦІЯ: Комп'ютерна інженерія

УДК 007:343.304

Веселовський В. Д.,
 магістрант,
Журавська І. М.,
 д-р техн. наук, професор,
 ЧНУ ім. Петра Могили, м. Миколаїв, Україна

МЕТОДИ ІДЕНТИФІКАЦІЇ ГОЛОСУ

Останнім часом швидко поширюється використання технології «дліпфейк» (англ. Deepfake), яка на замовлення кіберзлочинця зробить голос/відео молодшим (з урахуванням інтонації), або навіть «змінить стать» спікера. На жаль, результати цієї технології неминуче використовуватимуть у протизаконних цілях, наприклад, для захоплення керування роботизованими пристроями.

Об'єкт дослідження (розробки): процес автоматизації роботизованих систем, що виконують свої функції через отримання голосових команд.

Предмет дослідження (розробки): методи та засоби створення роботизованих систем, що виконують свої функції тільки після ідентифікації голосу власника.

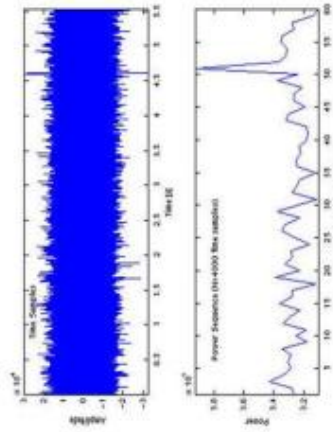


Рисунок 1 – Амплітудно-часовий аналіз

60

Для створення фейкового клоноу голосової команди з аудіозапису потрібен лише аудіофайл з голосом людини.
 Існує декілька методів ідентифікації голосу. Найпростішим підходом у обробці звуку є амплітудно-часовий аналіз. У цьому підході звуковий сигнал розглядається як одновишне уявлення коливань звукової хвилі з певною частотою семплювання (рис. 1).

Однією з основних переваг цього підходу є збереження повної інформації про сигнал, тобто, сигнал аналізується «як є», без відмови від важливої інформації. На жаль, це одночасно є і недоліком підходу – сигнал складно розділити на корисний і шум, а висока розмірність даних ускладнює їхню швидку та якісну обробку. Витягти велику кількість корисних властивостей із сигналу при подібному підході, на жаль, складно.

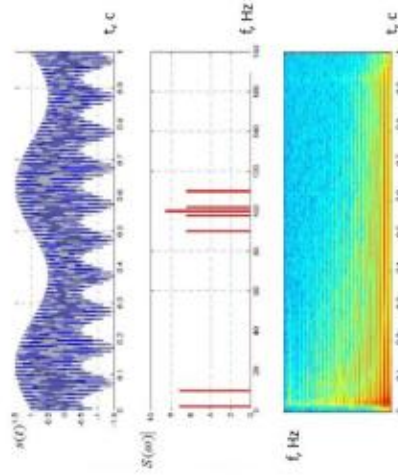


Рисунок 2 – Спектральний аналіз

Альтернативним підходом є спектральний аналіз. Його суть полягає в тому, що вихідний сигнал з якого завгодно необхідною точністю можна декомпонувати на складові. Іншими словами, будь-який складний сигнал може бути представлений у вигляді його синусоїд з певними частотами і амплітудами (рис. 2). Таким чином, спектр сигналу це вже двовишне уявлення сигналу, що дозволяє судити, як саме в часі розподіляється енергія сигналу по частотах.

61

<i>Москальський Б. А., Козлов О. В.</i> Оптимізація логістичних операцій підприємств малого бізнесу на основі інтелектуальних технологій.....	36
<i>Нечахін В. В.</i> Застосування нейронмережевої архітектури LSTM в системі керування сонячною електростанцією.....	39
<i>Обухова К. О.</i> Кіберзлочини та захист інтелектуальної власності.....	40
<i>Пилипчук Б. В., Журавська І. М.</i> Моніторинг фізичних навантажень велоспортсмена.....	47
<i>Сіденко Є. В., Кондратенко Г. В.</i> Рекомендаційна система для вибору мобільних пристроїв на основі методів прийняття рішень.....	49
<i>Скаковуб О. С.</i> Децентралізоване нелінійне керування групою БПЛА на основі нечітких алгоритмів.....	53
<i>Смолєнський М. М., Сіденко Є. В.</i> Дослідження архітектур нейронних мереж для фільтрації контенту.....	55
<i>Шани С. І.</i> Аналіз методів і засобів вимірювання параметрів нафтопродуктів.....	57
ПІДСЕКЦІЯ: Комп'ютерна інженерія	
<i>Веселовський В. Д., Журавська І. М.</i> Методи ідентифікації голосу.....	60
<i>Волощук С. І., Савінов В. Ю.</i> Застосування UWB-модуля DW1000 у робототехнічних системах.....	63
<i>Гончаров Д. С., Чуйко Г. П.</i> Застосування високочутливого оптичного датчика MAX30105 у медицині.....	65
<i>Ковальчук М. В., Обухова К. О.</i> Інерційні датчики та системи позиціонування.....	68
<i>Королюк Р. В., Данилова О. М., Бурлаченко І. С.</i> Автомобільна модель для дистанційного пошуку та картографування вибухонебезпечних пристроїв.....	70
<i>Медвідський С. В.</i> Використання динамічних біометричних показників для авторизації користувачів.....	73

При проходженні повітря через голосові зв'язки виникають вібрації, які у вигляді хвилі поширюються в середовищі. Кожен звук, якщо він не штучний, є цілим набором таких хвиль. Визначаючи основний тон звуку, його обертони та форманти можна успішно вирішувати ті чи інші завдання, пов'язані з аналізом аудіо.

Так, наприклад, частота основного тону (найнижча частота в сигналі) часто використовується в задачах визначення статі, так як середнє значення основного тону чоловіків і жінок відрізняється і становить в середньому 130 Гц у чоловіків проти 235 Гц у жінок. Аналіз набору обертонів голосу найчастіше корисний у завданні ідентифікації спікера, т. к. цей набір залежить від мовного апарату, який у кожного індивідуальний. І нарешті аналіз формант (областей посилення певних частот) активно використовується у завданнях, пов'язаних із перекладом мови в текст.

Спектрограма дозволяє успішно аналізувати всі вищезгадані складові, і багато в чому завдяки цьому моделі створені на основі даних про розподіл енергії за частотами в сигналі мають досить непогану точність.

Але є третій підхід до обробки даних в аудіо сигналі – нейронні мережі. Ідея проста – потрібно надавати на вхід мережі аудіо і очікувати, що мережа навчиться самостійно виявляти закономірності в даних і вирішувати необхідні завдання, чи то виділення ознак, ідентифікація того, хто говорить, розпізнавання мови, аналіз емоцій тощо.

В якості архітектури для отримання первинних ознак із сигналу ефективно використовують згорткові нейронні мережі, які показують хороші результати не тільки в задачах комп'ютерного зору, але і в комп'ютерного слуху. У процесі навчання нейронні мережі вивчать такі згортки, які, на думку мережі найкраще підходять для опису даних. При цьому самі дані не обов'язково представляти як вихідний сигнал.

Спільнота розробників повинна використовувати технічні та юридичні інструменти, які дозволять утримати використання зазначеної інноваційної технології «дідфейк» у легітимних межах. Наприклад, вбудовувати в звукові доріжки, що генеруються, своєрідні «водяні знаки», які вкажуть на їх штучний характер. Також необхідно розробляти інструменти з використанням форензики для розпізнавання підроблених аудіо- та голосових команд.