Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
магістр
(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ТА ЗАСОБІВ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS AND TOOLS FOR RECOGNIZING VOICE COMMANDS

Виконав: студент(ка)2 курсу, групи КНТ-121м
Спеціальності 121 Інженерія програмного забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Бережний О.Ю.
(прізвище та ініціали)
Керівник Пархоменко А.В.
(прізвище та ініціали)
Рецензент Зеленьова І.Я.
(прізвище та ініціали)

2022

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
_____С.О. Субботін
"____" _____2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Бережного Олександра Юрійовича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів та засобів розпізнавання голосових команд. Research and Software Implementation of Methods and Tools for Recognizing Voice Commands
керівник проєкту (роботи) Пархоменко Анжеліка Володимирівна, к.т.н., доцент
,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від "04" листопада 2022 року № 363
2. Строк подання студентом проєкту (роботи) 1 грудня 2022 року
3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування апаратно-програмного комплексу. 3. Конструювання апаратного-програмного комплексу. 4. Керівництво програміста. 5. Керівництво користувача.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Пархоменко А.В., доцент		
Нормоконтроль	Бєлова А.В., асистент		

7. Дата видачі завдання “05 ” вересня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітки
1	Постановка завдання роботи	1 тиждень	Завдання
2	Аналіз предметної області	2 - 4 тижні	Розділ 1
3	Розробка архітектури програми	5 - 6 тиждень	Розділ 2
4	Розробка програми	7 - 10 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	11 - 12 тиждень	Розділ 3
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	13 тиждень	Розділи 4-5, додатки
7	Захист роботи	14 тиждень	

Студент(ка) _____ Бережний О.Ю.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи) _____ Пархоменко А.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра: 112 с., 60 рис., 21 табл., 3 дод., 43 джерела.

ГОЛОСОВИЙ АСИСТЕНТ, МЕТОД РОЗПІЗНАВАННЯ, АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС, WI-FI МОДУЛЬ, ОБРОБКА ГОЛОСОВИХ КОМАНД.

Об’єкт дослідження – процес розпізнавання голосових команд та керування за їх допомогою.

Предмет дослідження – методи та засоби захоплення, аналізу та розпізнавання голосових команд для організації голосового керування в системі Розумний будинок (СРБ).

Мета роботи – дослідження та програмна реалізація методів та засобів розпізнавання голосових команд для розширення функціональних можливостей СРБ та організації роботи як в онлайн, так і в офлайн режимі.

Матеріали, методи та технічні засоби: структурне та об’єктно-орієнтоване програмування, модулі ESP-32, I2C мікрофон, мова програмування C++, платформа Platformio, персональний комп’ютер з процесором Intel Core 5 під управлінням операційної системи Windows 10.

Результати. Створено апаратний-програмний комплекс на основі SoC модуля ESP-32, який надає можливість керування процесами та об’єктами СРБ за допомогою голосових команд незалежно від наявності Інтернет з’єднання.

Висновки. Розроблений апаратно-програмний комплекс голосового керування дозволяє розширити функціональні можливості СРБ, оскільки надає можливість керування IoT приладами за допомогою голосу.

Галузь використання – системи типу Розумний будинок, навчальні лабораторії.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 112 pages, 60 figures, 21 tables, 3 appendices, 43 sources.

VOICE ASSISTANT, RECOGNITION METHOD, HARDWARE AND SOFTWARE COMPLEX, WI-FI MODULE, VOICE COMMAND PROCESSING.

The object of research is the process of recognizing voice commands and controlling with their help.

The subject of the research is methods and tools of capturing, analyzing and recognizing voice commands for the organization of voice control in the Smart House (SHS) system.

The purpose of the work is research and software implementation of methods and tools for recognizing voice commands to expand the functionality of SHS and organize work both online and offline.

Materials, methods and technical means: structural and object-oriented programming, ESP-32 modules, I2C microphone, C++ programming language, Platformio platform, personal computer with an Intel Core i5 processor running the Windows 10 operating system.

The results. A hardware and software complex based on the SoC of the ESP-32 module has been created, which provides the ability to control processes and objects of SHS using voice commands, regardless of the presence of an Internet connection.

Conclusions. The developed hardware and software complex of voice control allows you to expand the functionality of SHS, as it provides the ability to control IoT devices using voice.

Field of use – Smart home type systems, educational laboratories.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
BLE	– Bluetooth Low Energy;
CoAP	– Constrained Application Protocol;
EEPROM	– Electrically Erasable Programmable Read-only Memory
GND	– Ground;
GPIO	– General Purpose Input Output;
HTTP	– Hyper Text Transfer Protocol;
I2C	– Inter-IC Sound;
IDE	– Integrated Development Environment;
IoT	– Internet of Things;
IP	– Internet Protocol;
LoRaWAN	– Long Range Wide Area;
LPC	– Linear Predictive Coding;
MCM	– Multi Chip Module;
MFCCs	– Mel-frequency cepstral coefficients;
MP	– Micro Python;
MQTT	– Message Queuing Telemetry Transport;
OTA	– Over The Air;
SaaS	– Software as a Service;
SiP	– System in Package;
SoC	– System on a Chip;
TCP	– Transmission Control Protocol;
Wi-Fi	– Wireless Fidelity;
АПК	– апаратно-програмний комплекс;
ЗСК	– загальна система керування;
ОС	– операційна система;
ПЗ	– програмне забезпечення;
СРБ	– система Розумний будинок.

ВСТУП

Технології Інтернету речей (Internet of Things, IoT) невпинно розвиваються, що супроводжується вдосконаленням та покращенням протоколів та пристроїв IoT систем. Це дозволяє розширювати поточні можливості «розумних» речей, інтегрувати IoT системи у нові галузі, але при цьому поки що ще існують певні проблемні питання, зокрема інтерфейс з IoT системою.

На сьогоднішній день IoT пристрої зазвичай не охоплюються загальною стандартизацією і кінцевий користувач вимушений майже для кожного приладу опановувати новий інтерфейс керування. Кожний інтерфейс потребує додаткового часу на ознайомлення з функціоналом. Окрім цього, слід зазначити, що зазвичай керування відбувається через графічний інтерфейс, а це створює додаткове навантаження на зорову систему людини.

З метою зменшення витрат часу на опанування користувачем принципів керування кожним IoT пристроєм, розробники об’єднують їх у загальну систему керування (ЗСК), наприклад, Amazon Echo, Google Home, Samsung's SmartThings Hub. Метою ЗСК є забезпечення можливості управління IoT пристроями через єдиний інтерфейс, який надає однакові (або схожі за стилем) засоби керування, покращує сприйняття системи керування людиною та дозволяє легше масштабувати систему.

З метою зниження навантаження на зорову систему людини існує потреба залучення інших органів сприйняття. Зокрема, доволі розповсюдженим рішенням є використання голосового керування, що надає можливості управління системою шляхом голосових команд користувача. Провідні програмні компанії активно застосовують голосових асистентів такі як Siri (Apple), Alexsa (Amazon), Ok Google та ін. Зазначений підхід також надає можливість керування ЗСК людям з вадами зору.

Основним недоліком зазначених систем є потреба в з’єднанні з мережею Інтернет та дуже обмежений режим роботи офлайн, що ставить під загрозу здатність функціонування ЗСК у разі відсутності Інтернет з’єднання. Окрім цього, підключення ЗСК до мережі Інтернет підвищує ризики кібератаки та зараження вірусом, отже підвищує вимоги до безпеки системи, знижуючи корисну обчислювальну потужність ЗСК.

Отже, тема роботи, що пов’язана з технологіями розпізнавання голосових команд в офлайн та онлайн режимі є актуальною.

Мета роботи – дослідження та програмна реалізація методів та засобів розпізнавання голосових команд для розширення функціональних можливостей СРБ та організації роботи як в онлайн, так і в офлайн режимі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження особливостей IoT технології

Термін IoT (Інтернет речей), відноситься до підключених до мережі Інтернет пристроїв, що забезпечує зв'язок між ними та співпрацю. Замість мережі інтернет може бути будь-яка мережа, яка об’єднує пристрої: від малої домашньої, яка функціонує за допомогою роутера до великої мережі на виробничому підприємстві.

Завдяки появі відносно недорогих комп'ютерних мікросхем та телекомунікацій з високою пропускною спроможністю ми маємо тепер можливість автоматизації рутинних процесів, поліпшення взаємодії з ними. Звичайні побутові пристрої можуть збирати данні з датчиків, а отже, відповідно, реагувати на їх збудження [1]. Наприклад, датчик температури реєструє зменшення температури нижче мінімального рівня, система Розумний будинок фіксує цю подію та активує кліматичну установку до досягання рекомендованого значення тепла.

Метою інтеграції пристроїв за технологією IoT є їх поєднання у певну екосистему, в якій цифрові системи можуть записувати, контролювати та коригувати кожну взаємодію між з’єднаними приладами речами. Фізичний світ зустрічається з цифровим – і вони співпрацюють.

Перші принципи та підходи IoT технологій почали зароджуватись у 1990-х роках, а вже на початку XXI сторіччя інтернет речей почав набувати розголосу та починають з’являтись масові серійні побутові і виробничі прилади [2]. Через те, що «розумні» прилади автоматизують та відчутно полегшують побутові та виробничі процеси – сьогодні IoT продовжує активний розвиток, постійно з’являються нові рішення, які розширюють вже існуючі можливості та галузь застосування [3].

IoT технологіям притаманний високий потенціал: вони суттєво, якісно змінюють підходи взаємодії і використання технологій завдяки чому

спрощуються технологічні процеси, підвищують швидкість виконання завдань. Це сприяло задіянню IoT у багатьох сферах життя (рис. 1.1) [4].

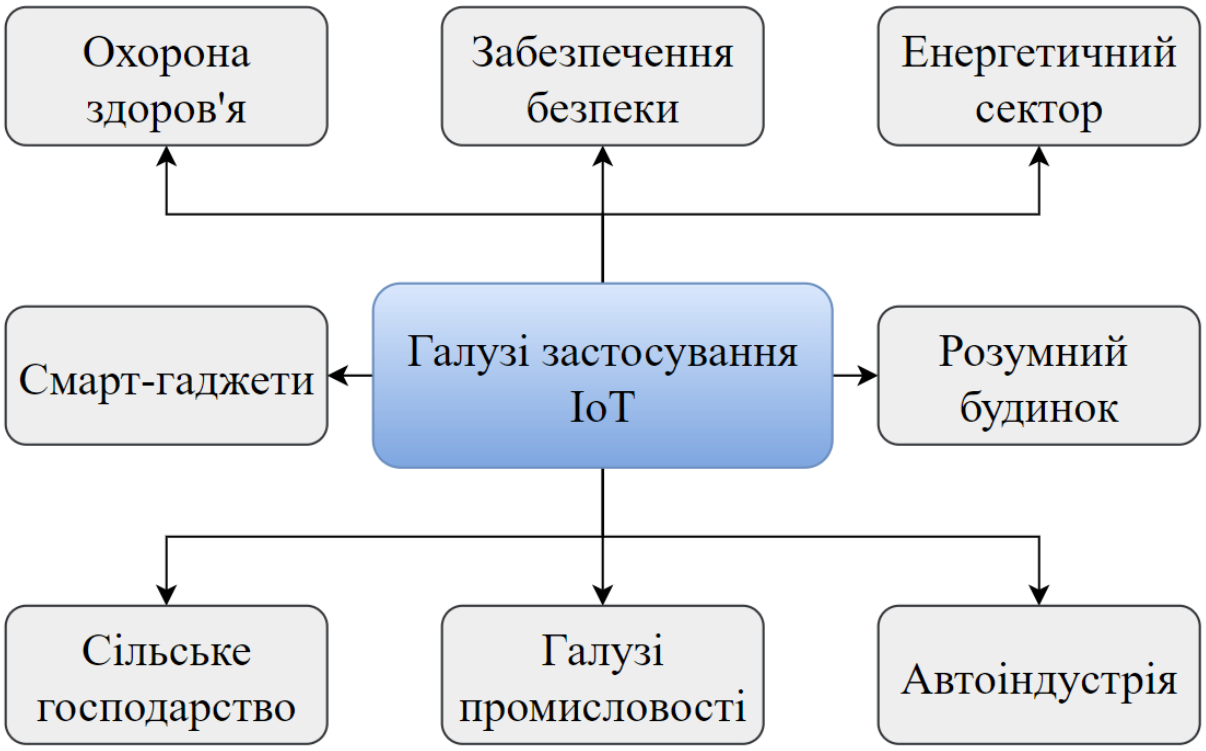


Рисунок 1.1 – Провідні галузі застосування IoT пристроїв [4]

В охороні здоров’я активно застосовується моніторинг за допомогою IoT датчиків. Дистанційно в автоматичному режимі вони здатні вимірювати показники пацієнта, що підвищує точність інформації, зменшуючи помилки в даних, заощаджує час на виміри та зменшує навантаження на медичний персонал. Лікарі можуть дистанційно в режимі реального часу цілодобово стежити за станом пацієнта та дистанційно призначати процедури [4].

Лікарі спираючись на дані дистанційного огляду за допомогою телемедицини (відеоконференції) можуть віддалено призначати лікування, оформлювати електронні рецепти. У багатьох випадках візити до лікарні не потрібні, що особливо актуально під час пандемії COVID-19 та інших обставин за яких масові скупчення людей наражають на потенційну небезпеку.

Для забезпечення безпеки, підвищення швидкості реагування на небезпечні події та надзвичайні ситуацій використовуються системи, які отримали загальну назву «Розумне місто». Це системи камер, датчиків, об’єднані в суцільну мережу обміну та аналізу даних дозволяють відслідковувати та досліджувати поведінку людей з метою попередження потенційно небезпечних ситуацій, розшуку зловмисників та розслідуванню злочинів.

Портативні комплекси охорони впроваджуються для виявлення спроб несанкціонованого доступу до об’єктів, миттєвого попередження про такі випадки охорону та активації програми захисту.

Будь-які надзвичайні ситуації, такі як спалах пожежі або затоплення, автоматично виявлятимуться за допомогою датчиків, і ця інформація передається відповідальним службам цілодобово у режимі реального часу. Команда ліквідації надзвичайних ситуацій може ефективно реагувати протягом декількох секунд, щоб почати операції з усування негативних наслідків [4].

В енергетичному секторі впроваджується концепція розумної електромережі. Вона уявляє вдосконалення існуючих електромереж за допомогою розумних датчиків, розміщених на лініях електропередачі та окремих споживачів. Ці датчики допомагають повідомити про будь-який збій, аномалію в лінії, зрозуміти її характер та дослідити причини. Якщо одна з ліній передачі не працює, інтелектуальні датчики автоматично спрацюють, щоб переключитися на іншу мережу, щоб забезпечити безперебійне постачання.

Завдяки датчикам на лічильниках споживачів запроваджено подвійні тарифи: день / ніч, мета яких є мотивація відповідального споживання енергоресурсів, розвантаження енергосистеми. Завдяки оптимізованому використанню та енергозбереженню можна значно збільшити загальну ефективність і зменшити витрати енергії [4].

У сільському господарстві та сільськогосподарській галузі існує багато проблем, пов’язаних із виробничим процесом. Молоде покоління все менше приваблює фермерство, через що виникає нестача робочих рук і, відповідно, потреба в оптимізації робочих процесів, підвищення ефективності всього сільського господарства з меншою кількістю працівників.

Технології розумних сенсорів допомагають покращити кожен етап сільського господарства, а автоматизація скорочує потребу в ручній праці [4]. Це такі рішення як: розумне зрошення (ефективне використання води для забезпечення сталої вологості ґрунтів), розумні теплиці (штучне підтримання сприятливих кліматичних умов для агрокультур); цифровізація скотарства (слідкування за станом здоров’я скота, дані про переміщення випасних тварин).

Промислова галузь є однією з перших, хто почав інтеграцію IoT, оскільки процеси який повністю змінив певну кількість етапів розробки і створення продукту [4]. Промисловий IoT допомагає оптимізувати різні етапи виробництва продукції, такі як: ланцюги поставок і управління складськими запасами, розробка продукту, автоматизація процесів конвеєрного виробництва, відслідковування якості продукції. IoT пристрої дозволяють зменшити експлуатаційні витрати, збільшити об’єми продукції та знизити собівартість виробництва.

На виробництвах розповсюдженим рішенням є використання не тільки окремих груп IoT, а й впровадження концепції об’єднаної фабрики – рішення для вдосконалення всіх сфер діяльності підприємства, де основні компоненти, такі як машини, інструменти та датчики, будуть підключені до об’єднаної мережі з метою полегшення керування та доступу [4]. Огляд перебігу процесу, відстеження часу простою, планове технічне обслуговування та паузу у робочому процесі можна зробити віддалено.

В автоіндустрії набирає популярності концепція автономного водіння (так званий автопілот), який у майбутньому має на меті позбавити авто від

необхідності мати водія. Автономне водіння розвивається завдяки використанню штучного інтелекту та технології розумних датчиків. Поточне покоління транспортних засобів обладнаних технологіями часткової автоматизації вже допомагає водіям у керуванні авто, підвищуючи безпеку та впроваджуючи систему попередження зіткнень і аварійних ситуацій: інтелектуальні датчики постійно збирають інформацію про стан транспортного засобу, якість дорожнього покриття, про оточуючі транспортні засоби, об'єкти на дорозі [4]. Транспортні засоби та розумні об'єкти можуть обмінюватися інформацією один з одним за допомогою радіочастотної технології.

Смарт гаджети, або носимі розумні пристрої такі як фітнес браслети або смартфони отримали значний розвиток, набули значного функціоналу та стали невід'ємними атрибутами повсякденного життя. Вони здатні відслідковувати стиль життя людини та надавати рекомендації щодо здоров'я, передбачати дії користувача да нагадувати йому про вірогідні справи та ін.

Окрему ланку серед направлень IoT займає система Розумний будинок. Ринок пропозицій насичено різноманіттям пристроїв і датчиків, які створені для полегшення рутинних процесів.

Будинки оснащують розумними замками, які керуються зі смартфону або магнітного ключа, відслідковують статистику відвідувачів, дистанційно дозволяють надавати та обмежувати доступ до оселі [4]. Розумне освітлення для дому крім енергозбереження, надає можливість автоматичного вмикання та вимикання у відповідь на зовнішнє збудження як то рух чи голосові команди. Автоматичне терморегулювання дозволяє підтримувати температурний режим максимально ефективно використовуючи енергоресурси, та оптимізуючи їх споживання.

Поширення IoT забезпечує зростаючий попит на ринку електрообладнання, та спонукає провідні компанії розроблювати та модернізовувати лінійки розумних приладів. Для розробки

використовуються напрацювання комп'ютерної інженерії та істотний технічний прогрес.

Серед чинників, що сприяли розвитку можна виділити (рис. 1.2) [5]:

– довіра користувачів до IoT технологій, оскільки вони вже витримали випробування часом і довели свою користь у повсякденні;

– популяризацію IoT серед населення, що породжує підвищений попит на продукцію, виробники IoT отримали впевненість що зможуть реалізувати створений товар;

– зниження вартості пристроїв зробили їх доступними для більш широкої кількості розробників і кінцевих споживачів;

– технологічний розвиток, який розширив галузь застосування IoT пристроїв та їх функціональні можливості, відбувається постійне вдосконалення продукції та підвищення зручності її використання;

– віддалені обчислення (хмарні технології) надали споживачам можливості взаємодіяти з даними за мінімальні витрати на апаратні пристрої, проводити розрахунки на віддалених серверах;

– через перспективність напряму останнім часом компанії залучають додаткове фінансування, яке стимулює розробників та науковців до винаходів.

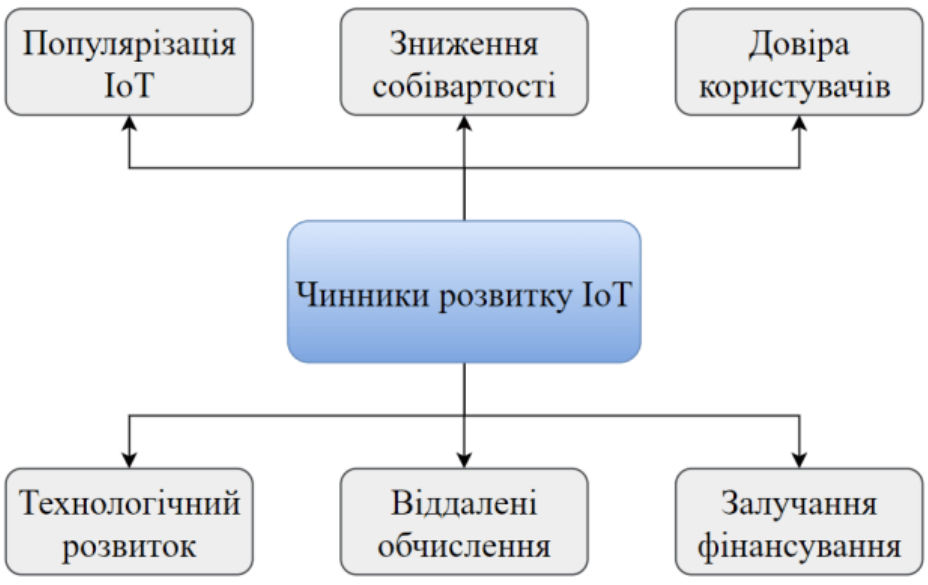


Рисунок 1.2 – Чинники розвитку IoT [5]

Однак, потрібно зазначити, що розвиток IoT ускладняється оскільки поки що ще зазнає впливу проблем недосконалості технологій (рис. 1.3) [6].

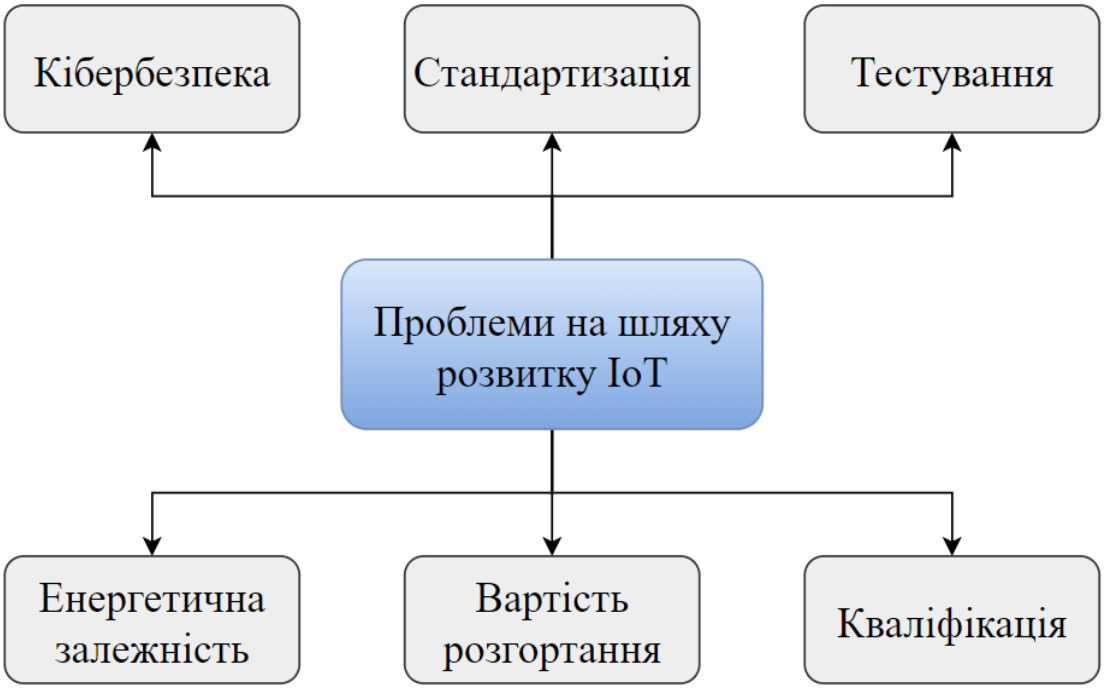


Рисунок 1.3 – Проблеми на шляху розвитку IoT [6]

Через відносно невелику обчислювальну потужність IoT пристроям складно забезпечити належний рівень стійкості від несанкціонованого доступу. Оскільки розумні пристрої збирають чималу інформацію про об’єкт дослідження хакерська атака потенційно може привести до великих проблем, адже може надати можливість слідкувати за станом об’єкту в режимі реального часу.

Відсутність стандартизації на початку активного розвитку технологій призвела до наявності великої кількості стандартів обміну даними, що ускладнює комунікацію розумних пристроїв різних моделей та поколінь, інтеграції їх у загальну систему. Це спричиняє певні обмеження на розробку розумних систем та розширення існуючих, підвищує складність проєктування, вартість розробки.

Із зростанням IoT що б встигнути в гонці виробництва, виробники намагаються скоротити час розробки пристрою, обмежуючи кількість тестувань та ігноруючи довгострокову підтримку існуючих пристроїв. Таким чином продукти не мають належного та регулярного тестування і оновлень програмного забезпечення [6].

Збільшення кількості розумних пристроїв призводить до підвищеного споживання електроенергії. Не всі будинки та офіси розраховані на додаткове навантаження мережі, то ж інколи це питання може стосуватись пожежної безпеки і стабільності енергозабезпечення. При розробці розумних систем треба окрему увагу приділяти її живленню.

Проєктування та впровадження великих IoT систем потребує залучання спеціалістів з відповідною профільною освітою, оскільки необхідно враховувати особливості розумних пристроїв та співвідносити їх функціональні можливості до потреб замовника. Це збільшує вартість розробки та монтажу системи.

Проте, навіть беручи до уваги недоліки IoT, інвестори готові вкладати кошти у розвиток галузі і впровадженні її у повсякденне життя. Це підтверджує статистика інвестицій провідних компаній у 2022 (табл. 1.1) [7].

Розмір глобального ринку IoT у 2021 році оцінювався в 384,70 мільярдів доларів США. За оцінками аналітичних агентств очікується, що ринок зростатиме з 478,36 мільярдів доларів США у 2022 році до 2465,2 мільярдів доларів США до 2029 року [8].

Таблиця 1.1 – Топ десять найбільших фінансових інвестицій у IoT сектор за 2022 рік [7]

Компанія	Напрям	Інвестиції млн. дол. США
Enable Injections	Розробка переносних систем доставки ліків	215
Axonius	Рішення для забезпечення безпеки	200
Helium Systems	Забезпечення доступу IoT датчиків до інтернет мережі	200
SiFive	Виробництво напівпровідників	175
Shenzhen Ou Ruibo Electronics	Технології IoT	157
Eigencomm	Виробництво чіпів IoT	156
Envision Digital	Поєднання штучного інтелекту з IoT	152
Salt Security	Платформа для захисту інтерфейсу з прикладного програмування портативних пристроїв	140
Kinexon	Розробка датчиків та сенсорів IoT	130
Leddar Tech	Технології визначення відстані до об’єктів	116

В Україні вже активно впроваджується IoT технології в проєктах як приватного (локального) так і регіонального рівнів (рис.1.4) [9]:

– провайдери стільникового зв’язку забезпечують можливість налагодження зв’язку шляхом випуску спеціальних тарифів для IoT пристроїв;

– комунальні підприємства дистанційно збирають дані з приладів обліку споживання енергоресурсів (Львів, Кропивницький);

– агросектор застосовує підходи розумного виробництва на полях та фермах (Харківська область);

– розгортуються мобільні станції моніторингу довкілля, якості повітря, радіаційного фону (Київ, Дніпро).

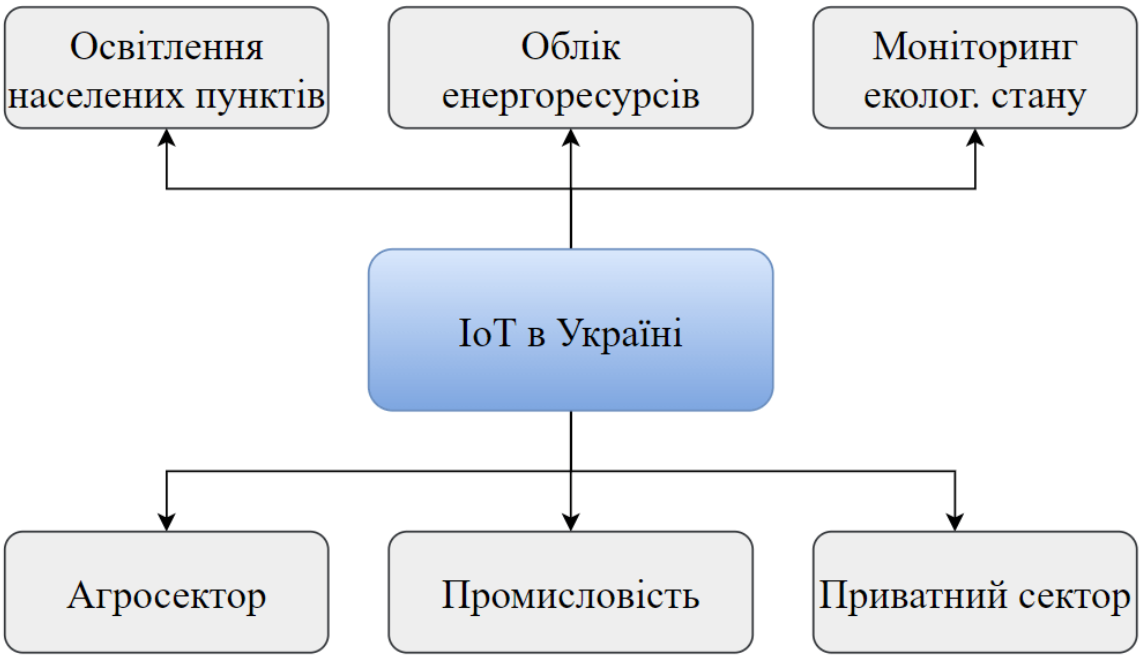


Рисунок 1.4 – IoT сектор в Україні [9]

За результатами проведеного дослідження можна стверджувати що IoT технології наразі стрімко розвиваються. Вже існує велика кількість реалізованих проєктів, включно й в Україні, але наразі наявні певні недоліки, які ускладнюють розвиток технологій.

1.2 Дослідження протоколів та пристроїв, які використовуються в IoT індустрії

1.2.1 Протоколи, що забезпечують взаємодію IoT пристроїв

Протокол – це набір правил, спільних для обох мережевих об’єктів, для забезпечення спілкування. Протокол керує тим, як виглядає спілкування об’єктів між собою: які значення та атрибути можуть бути передані та як вони приймаються й обробляються, містить данні про те, які методи безпеки використовуватимуться тощо [10].

Протоколи є невід’ємною частиною стеку технологій IoT. Без них була б неможливою реалізація взаємодії та комунікації IoT пристроїв.

IoT системи поділяються на дротові та бездротові. Дротові пристрої IoT підключаються через низьковольтні або звичайні лінії передачі даних. Кожен вузол має унікальний ідентифікатор і вбудоване програмне забезпечення, що працює на ньому. Такі пристрої є самодостатніми і не залежать від центрального хаба. Вони спілкуються за допомогою спеціальних протоколів, які розроблені спеціально для спілкування через загальну шину даних.

Дротове розгортання є надійним, але саме бездротове підключення стало рушійною силою поширення розгортання IoT. Це пов’язано з тим, що дротові рішення IoT не дозволяють оперативно додавати більше пристроїв до мережі оскільки потребує проведення додаткових ліній та інтеграцію конекторів до вже існуючої лінії, та окрім цього налаштування додаткових пристроїв може бути доволі складним [10].

Отже, бездротові протоколи зв’язку IoT мають наступні переваги:

– масштабованість: стандартні протоколи підтримують додавання нових пристроїв з мінімальними налаштуваннями конфігурації;

– відносно легка взаємодія: протоколи зв’язку IoT можна запрограмувати для роботи з різним обладнанням, вони підтримують пристрої від різних постачальників;

– надійність: стандартна технологія зв'язку забезпечує безпечну передачу даних і стійкість до перешкод.

Отже доречно буде розглядати протоколи, які пов’язані з бездротовою передачею даних.

Протоколи та стандарти IoT класифікують на дві категорії (рис. 1.5):

– протоколи мережевого рівня – використовують для підключення пристроїв у мережу в якій дозволяють наскрізний обмін даними;

– протоколи даних – забезпечують реалізацію взаємодії пристроїв на програмному рівні [11].

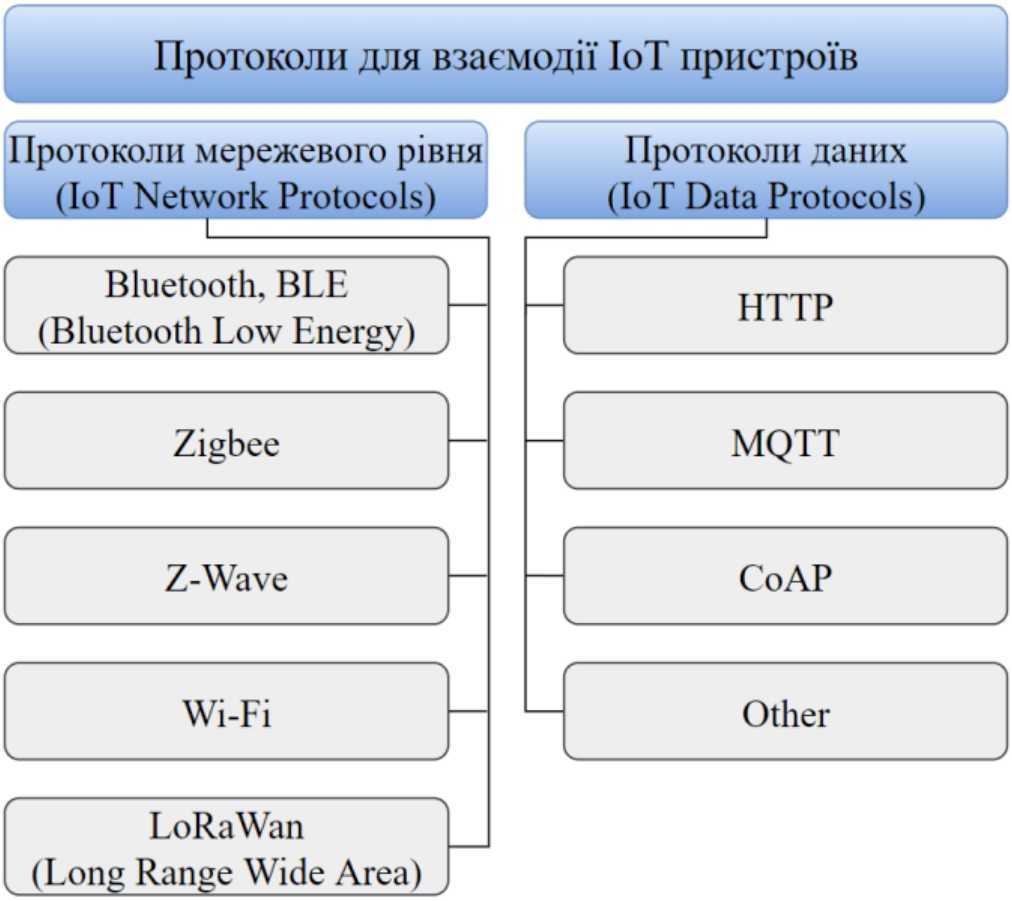


Рисунок 1.5 – Протоколи для взаємодії IoT пристроїв [11]

Дослідження показали, що існує декілька найпоширеніших протоколів мережевого рівня які використовуються залежно від завдань.

Bluetooth і Bluetooth Low Energy (BLE) – це бездротова технологія, яка використовується для обміну даними на короткій відстані. Часто використовується в персональних пристроях, таких як мобільні телефони, плеєри, планшети [12]. Bluetooth передає невеликі фрагменти даних пакетами, і може виникнуть проблеми при передачі великих файлів. Протокол широко використовується в система розумних будинків. Власники захоплюються можливістю керувати підключеними пристроями через смартфон, що досить зручно.

BLE є версією Bluetooth, оптимізованою для з’єднань IoT на малій відстані: споживає менше енергії, проте не може спілкуватися з класичним пристроєм Bluetooth, якщо на обох не встановлено відповідні протоколи. Є

достатньо безпечним, оскільки він шифрує дані, що передаються на рівні програми та мережі.

Zigbee – це надійний і масштабований комунікаційний протокол IoT, який використовується для збору даних з датчиків у домашній мережі або промислових додатках. Він передає невеликі обсяги даних на помірні відстані [12]. Zigbee працює за топологією сітки, що самовідновлюється, це робить його дуже надійним. Нові пристрої можуть приєднатися до мережі після виконання процесу «рукостискання», який займає зазвичай не більш ніж 30 мілісекунд. Але протокол потребує спеціального шлюзу для керування пристроями IoT, що відносно дорого коштує, особливо в порівнянні з Bluetooth.

Z-Wave – це малопотужний бездротовий протокол, який зазвичай використовується для рішень розумного дому та бізнес-додатків. Z-Wave пропонує найнижчу затримку серед розповсюджених мережевих протоколів зв’язку IoT. Важливо зазначити, що ця технологія працює з різною частотою в кожній країні, а це означає, що користувачам доведеться купувати адаптер пристрій, коли вони змінюють місце розташування. Z-Wave - це запатентована технологія, якою керує Z-Wave Альянс, також він контролює сертифікацію пристроїв . Це забезпечує що кожен гаджет Z-Wave буде сумісним з будь-яким контролером Z-Wave незалежно від виробника [12].

Wi-Fi використовує Інтернет-протокол (IP) для підключення пристроїв до локальної мережі (LAN). Він забезпечує надійний і безпечний зв'язок між близько розташованими пристроями. Цей протокол є відносно дешевим і простим у розгортанні, він підходить для застосування всередині приміщень, наприклад для домашньої автоматизації. Він добре працює з важкими файлами та може обробляти величезні обсяги даних з доволі високим рівнем швидкості [12]. Однак цей протокол зв’язку має обмеження діапазону, хоча останнім часом його розширюють додаючи другий діапазон.

LoRaWAN – радіомережа великого радіусу дії – це нестільникова технологія бездротової глобальної мережі, яка з’єднує пристрої на великій

відстані, що робить її придатною для розумних міст і промислових програм, які передають телеметричні дані на великі відстані. Одним із прикладів є інтелектуальні вуличні ліхтарі у країнах Європи, підключені до шлюзу LoRa, що працює за відповідним протоколом. Ця технологія може підключати мільйони пристроїв IoT і оптимізована для низького енергоспоживання. Нові пристрої можуть бути або жорстко закодовані, або організовані в бездротове з'єднання. Шлюз LoRa збирає дані з різних датчиків і передає їх на сервер або хмару через стандартний IP-протокол. Але цей протокол зв'язку IoT не підходить для програм, які потребують низької затримки або передають великий обсяг даних.

За результатами проведеного порівняльного аналізу було створено таблицю зведених характеристик протоколів мережевого рівня IoT пристроїв (табл. 1.2) [12].

Таблиця 1.2 – Зведена порівняльна характеристика протоколів мережевого рівня [12]

Протокол / параметр	Bluetooth, BLE	Zigbee	Z-Wave	Wi-Fi	LoRaWan
1	2	3	4	5	6
З'єднання	P2P Сітка Зірка	Сітка Зірка	Сітка	Зірка	Зірка
Швидкість передачі даних	1-3 Мбіт/с	250 Кбіт/с	До 100 Мбіт/с	До 1 Гбіт/с	50 Мбіт/с
Діапазон частот	2.4 ГГц	2.4 ГГц	908.42 МГц (США), ін. залежить від географічного положення	2.4 ГГц	Діапазон частот
Стандарт	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.15.4	IEEE802.il	LoRaWAN

Продовження таблиці 1.2

1	2	3	4	5	6
Покриття (пряма видимість)	До 100 м	До 200 м	До 100 м	До 70 м	До 15 км у приміських умовах
Споживання електроенергії	Bluetooth: помірне BLE: низьке	низьке	низьке	помірне	низьке
Вартість	Низька	Середня	Висока	Низька	Низька
Ліцензування	Безкоштовно	Комісія за кожен підключений пристрій	Комісія за кожен підключений пристрій	Безкоштовно	Безкоштовно
Підтримка	Bluetooth Special Interest Group (SIG)	Zigbee Alliance	Z-Wave Alliance	Wi-Fi Alliance	LoRa Alliance

Серед протоколів даних якісно відрізняються за популярністю протоколи HTTP, MQTT, CoAP (рис. 1.6) [13].

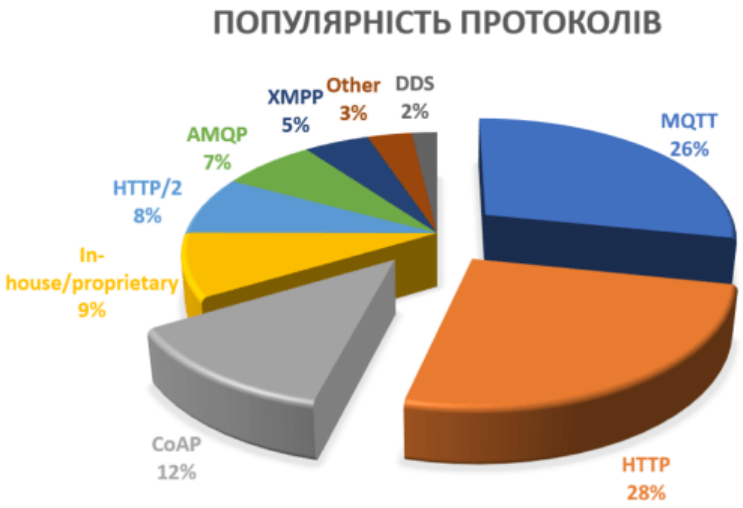


Рисунок 1.6 – Популярність мережевих протоколів IoT [13]

MQTT – це міжмашинний протокол підключення до IoT. Це система публікації та підписки, яка полегшує зв'язок між кількома пристроями, для якої не потрібно, щоб і клієнт, і сервер встановлювали з’єднання одночасно. Протокол дозволяє клієнтам підписатися на певні параметри, щоб вони могли отримати інформацію, яку вони очікують . MQTT був розроблений як простий і легкий протокол обміну повідомленнями в мережі з низькою пропускною здатністю [14].

Серед переваг протоколу MQTT виділяють: ефективну передачу даних і швидку реалізацію через відносну легкість протоколу, низьке використання мережі завдяки мінімізації пакетів даних, ефективний розподіл даних. Серед недоліків протоколу MQTT виділяють: повільні цикли передачі (порівняно з CoAP), малий вміст повідомлень, важку масштабованість.

HTTP – прикладний протокол для передачі документів гіпертекстового формату, наразі підтримує передачу даних у довільному форматі. Протокол являє собою комунікацію формату запитів / відповідей, коли клієнти посилають запит на сервер, а сервер відповідним чином відповідає на ці запити (кожен запит не залежить від іншого). Сьогодні протокол залишається одним із основних засобів комунікації у мережі. Дані у HTTP передаються за протоколом TCP, що гарантує надійність доставки та розбиває великі запити і відповіді на блоки, керовані мережею [13].

Як переваги HTTP виділяють відносну простоту реалізації, наочність даних, безпеку з’єднання та можливість передачі даних довільного об’єму і формату [14]. З поміж недоліків застосування протоколу HTTP наводять відносно велике енергоспоживання, в порівнянні до MQTT, що є «платою» за можливість передачі довільного розміру пакетів.

CoAP – це простий протокол із низькими накладними витратами, спеціально розроблений для обміну невеликим даними у мережах з низькою пропускною здатністю. Дуже схожий на HTTP, навіть якщо є важливі відмінності, які ми обговоримо пізніше. Інтерактивна модель CoAP схожа на

модель клієнт/сервер HTTP. CoAP використовує двошарову структуру обміну даними.

Серед конкурентних переваг протоколу CoAP властиві менші накладні витрати, короткий час пробудження, нижчу затримку порівняно з HTTP. Але слід зазначити що CoAP є відносно ненадійним протоколом через особливості архітектури: повідомлення CoAP досягають невпорядкованого порядку або можуть бути втрачені на шляху до місця призначення. Через підтвердження кожного повідомлення збільшується час обробки. Відсутня перевірка чи отримане повідомлення було розкодовано належним чином чи ні.

За результатами порівняльного аналізу було створено порівняльну таблицю характеристик протоколів даних (табл. 1.3) [14].

Таблиця 1.3 – Зведена порівняльна характеристика протоколів мережевого рівня [14]

Протокол / параметр	MQTT	HTTP	CoAP
Архітектура	Клієнт/Брокер	Клієнт/Сервер	Клієнт/Сервер
Розмір заголовку	2 байти	Невизначено	4 байти
Методи	Опублікувати / підписатись	Запит / відповідь	Запит / відповідь
Розмір повідомлення	Малий	Великий	Малий
Підтримка кешу та проксі	Частково	Так	Так
Транспортний протокол	TCP	TCP	UDP
Формат кодування	Бінарний	Текст	Бінарний
Модель ліцензування	Open Source	Безкоштовна	Open Source

Вибір протоколу даних повинен ґрунтуватися на характеристиках та особливостях типів даних що будуть передаватись. Необхідно звернути увагу на те, що звуковий потік є доволі об’ємними даними відносно показників більшості IoT датчиків. Слід враховувати що чим більша розповсюдженість та популярність протоколу що застосовується – тим ліпше буде налаштування взаємодії комплексу з зовнішніми IoT системами. Тобто це сприятиме покращенню масштабованості та зменшуватиме навантаження при удосконаленні АПК. Враховуючи вищезазначені критерії в якості протоколу обміну даними для АПК було обрано HTTP протокол.

1.2.2 Пристрої, що використовують в IoT системах

Пристрої що використовують в IoT системах поділяються на дві загальні групи: перетворювачі та контролери – так звані «розумові центри» (рис. 1.7) [15].

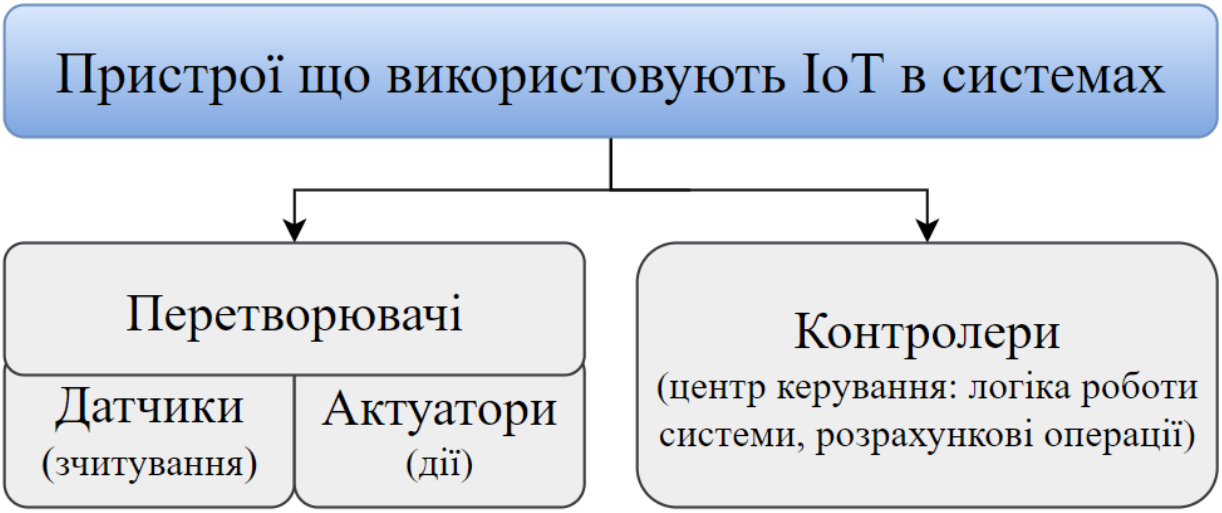


Рисунок 1.7 – Пристрої що використовують в IoT системах [15]

Перетворювачі – група пристроїв що забезпечують комунікацію та взаємодію IoT системи з зовнішнім середовищем. Зазвичай перетворювачі розділяють на дві підгрупи: датчики та актуатори (табл. 1.4) [16].

Таблиця 1.4 – Зведена порівняльна характеристика датчиків та актуаторів [16]

Перетворювач / параметр	Датчик	Актуатор
Сутність роботи	Вимірювання неперервних і дискретних змінних навколишнього середовища	Імпульс параметрів безперервних і дискретних процесів.
Порт	Вхідний	Вихідний
Результат роботи	Електричний сигнал	Тепло або рух
Приклад	Термометр, магнітометр, акселерометр, мікрофон	Світлодіод, гучномовець, соленоїд, сервопривід

Датчик – пристрій який відповідає за зчитування інформації з навколишнього середовища, здатний вимірювати фізичні величини та на їх основі генерувати електричні сигнали. Найчастіше це такі параметри як: температура, атмосферний тиск, загальна освітленість, вологість, звукові колихання, швидкість повітряних мас та ін. Незважаючи на різноманітність датчиків зазвичай їх поділяють саме на два типи за різновидом вихідного сигналу: аналогові та цифрові. Цифровий датчик на відміну від аналогового має в собі вбудований аналого-цифровий перетворювач, якій трансформує данні перед передачею [15].

Актуатор – це пристрій, який у результаті своєї роботи змінює фізичні величини, виконуючі певні дії (різновид яких залежить від функціоналу конкретного актуатору) після отримання деякого вхідного сигналу [16]. Тобто актуатор – це пристрій який у відповідь на керуючий сигнал генерує зміни у фізичній системі через вироблення тепла, здійснення руху тощо.

Контролер (розумовий пристрій) – пристрій, який побудовано з вбудованим мікроконтролером, призначення якого взаємодія з

перетворювачами, обробка даних, проведення обчислень, керування і контроль за процесами [15].

Розробка пристроїв для Інтернету речей вимагає оперативного створення прототипів різних апаратних конфігурацій у відповідь на швидкозмінні запити ринку. Через це розвиток в IoT розробці отримала модульна концепція, яка передбачає розробку з застосуванням готових модулів, що мають спільний інтерфейс взаємодії. Це дозволяє уникнути необхідності підбору другорядних компонентів (які потрібні для роботи того чи іншого модулю), оскільки вони вбудовані безпосередньо у сам модуль. Даний підхід має суттєві переваги в порівнянні з повноцінним процесом розробки: збільшена зручність, значна ефективність, гнучкість до змін, зменшення часових і матеріальних витрат [17].

Існує декілька концепцій модульних систем: система на чіпі (SoC), система в упаковці (SiP), багаточіпові модулі (MCM), дискретні чіпи на друкованій платі. Кожна реалізація системи має певні переваги та недоліки, однак, для розробки IoT системи кращим вибором стане SoC (табл. 1.5). SoC надає високий рівень гнучкості, зручність налаштування та багато іншого. Хоча реалізації SiP, MCM та дискретних чіпів можуть добре справлятися зі складними експлуатаційними вимогами, SoC розроблено для того, щоб робити це, економлячи енергію та витрати [18].

Таблиця 1.5 – Переваги SoC систем [18]

Система / параметр	SoC	Інші
Споживання енергії	Помірне, оптимізовано	Більше
Розмір	Компактне	Зазвичай більше
Вартість розробки	Відносно невеликі, легка масштабованість	Більші, але також легка масштабованість
Продуктивність	Висока, забезпечена комплексною оптимізацією компонентів	Залежить від налаштувань розробників

SoC – це тип модульної архітектури який уявляє собою інтегральну схему, що об’єднує необхідні для роботи компоненти електронного пристрою на одній платі для зменшення простору та підвищення ефективності. SoC модуль може включати:

– блок мікроконтролера – частина керування;

– радіочастотний модуль – відповідає за бездротовий зв’язок;

– енергонезалежна пам'ять – зберігає програмний код та іншу системну інформацію;

– енергозалежна пам'ять – зберігає тимчасові дані;

– цифровий сигнальний процесор – допомагає отримувати та розуміти сигнали від підключених пристроїв;

– керування живленням – керування та оптимізація енергоспоживання у різних режимах роботи.

Розповсюдженим втіленням SoC архітектури наразі є модулі на базі процесорів ESP: ESP-32, ESP-8266, та інші (рис. 1.8) [18].

Рисунко 1.8 – Модулі побудовані за SoC архітектурою

Проведений аналіз демонструє, що IoT система зазвичай складається з контролеру та перетворювачів (датчиків і актуаторів). В якості контролеру для IoT системи найоптимальнішим вибором є модулі, побудовані на SoC архітектурі.

1.3 Дослідження методів та засобів голосового керування

1.3.1 Голосове керування як метод взаємодії з IoT системами

Через спалахи епідеміологічних захворювань (COVID-19) зростає потреба у дотриманні соціальної дистанції та мінімізація дотику з поверхнями. Це надає новий імпульс розвитку технологіям сенсорної взаємодії, в тому числі й розпізнаванню голосу.

Перші пристрої з розпізнавання мови були розроблені у 50-х роках XX сторіччя, але були дуже обмеженими у своєму функціоналі (сприймали лише числа) та мали відносно невисоку точність розпізнавання [19].

У 10х роках 21-го сторіччя технологія розпізнавання досягла точності людського рівня завдяки залучанню алгоритмів штучного інтелекту. Після чого відбулося експоненціальний рівень використання технології в IoT пристроях. На підтвердження цього було проведене опитування компанії PwC, згідно до якого 90 відсотків респондентів були знайомі з можливостями голосових асистентів, а 72 відсотки мали досвід їх використання у повсякденному житті з різноманітними девайсами (табл. 1.6) [19].

Таблиця 1.6 – Розповсюдженість голосового керування за IoT пристроями [19]

Пристрій	Кількість респондентів (у %)
Смартфон	57
Планшет	29
Ноутбук	29
Персональний комп’ютер	29
Аудіосистема	27
Телевізор	21
Навігатор	20

Враховуючи тенденції росту та перспективності за оцінками дослідників Grand View Research у 2025 році обіг ринку розпізнавання мови

досягне 31.8 мільярдів доларів США. Очікується, що програмне забезпечення для розпізнавання голосу та мовлення на основі штучного інтелекту зростатиме з найшвидше з 2022 по 2030 рік [20].

Виділяють декілька причин, які сприяли росту популярності технології голосового керування:

– розповсюдження портативних пристроїв, і, відповідно, потреби забезпечення можливості керування «на ходу», що доволі складно зробити шляхом зорового контакту;

– збільшення різноманіття та кількості розумних колонок у сучасних домогосподарствах, що, потенційно, знижує вартість впровадження голосового керування у системи розумних будинків;

– технологічний прогрес у алгоритмах з обробки та аналізу людської мови, інтонації;

– поширення різноманіття варіантів штучного інтелекту та машинного навчання для напрацювання персоналізації досвіду, адаптації під конкретного користувача [20].

Системи розпізнавання мовлення продовжують розвиватися та стають все більш складними, підвищуються їх якісні характеристики. Експерти прогнозують, що в найближчі роки майже кожна програма буде мати інтегровану у себе голосову технологію.

Системи голосового керування вже інтегровані до багатьох IoT пристроїв та, відповідно у галузі, де застосовуються останні. У розумних домах широко розповсюджені SmartThings, Amazon Echo, Google Nest, які можуть виконувати нескладні голосові команди від користувачів для керування підключеними пристроями. У закладах охорони здоров’я голосове керування стає у нагоді при консультаціях за технологіями телемедицини.

У керуванні енергосистемою спостерігаються тенденції впровадження голосових асистентів для встановлення зворотнього зв’язку з споживачами. Як приклад провідна британська компанія Octopus Energy організувала спільний проєкт з віртуальнім помічником Amazon, Alexa, щоб допомогти

клієнтам контролювати їх споживання, зокрема дізнатися, у який час доби їхня електроенергія буде найдешевшою [21].

На виробництвах віртуальні помічники з голосовим керуванням допомагають співробітникам виконуючи тривіальні, але трудомісткі завдання, як планування зустрічей, налаштування нагадувань тощо. Існують голосові інтерфейси, які перетворюють мовлення в текст, можуть робити нотатки для зустрічей, дзвінків та ін. Інформаційне агентство Gartner прогнозує, що до 2023 року 25 відсотків взаємодії співробітників із програмами на виробництвах відбуватиметься за допомогою голосу [22].

Фінансовий сектор залучає технології голосового розпізнавання як для підвищення зручності взаємодії з клієнтом так і для покращення рівня безпеки. Bank of America, Privat Bank та багато інших використовують голосових асистентів щоб допомогти клієнтам перевірити їхній баланс, налаштувати сповіщення для відстежування витрат. Британський банк Atom використовує голос клієнтів як складову для авторизації у системі [23].

Незважаючи на прогрес використання голосових технологій у має свої труднощі. Щоб голосові контролери IoT гарно розпізнавали команди звук має бути чистим, без перешкод, голосові команди – зрозумілими та чіткими. Голосовий помічник не тільки повинен відокремлювати людський голос від фонових шумів, але також має підтримувати різні мови та людські акценти, щоб бути справді ефективним.

Окрім цього голосове керування в IoT стикається з проблемою обчислювальною потужністю пристроїв. Не всі пристрої IoT підключені до Інтернету або мережа не є завжди доступною. Щоб забезпечити надійну взаємодію між людиною та машиною, постає питання можливості розпізнавання голосу в автономному режимі.

Як показав проведений аналіз технологія голосового розпізнавання розвивається та активно застосовується для забезпечення керування IoT. Хоча на сьогодні існують певні труднощі з реалізацією, але переваги швидкості, зручності та взаємодії без використання тактичного дотику,

особливо в часи соціального дистанціювання, популяризують технологію та роблять перспективним її напрям.

1.3.2 Алгоритми, що використовуються для розпізнавання людського мовлення

Через особливості людської мови вважається, що її розпізнавання це одна з найскладніших областей. Технологія розпізнавання мовлення може бути оцінена за багатьма параметрами: за рівнем точності (частотою помилок у словах), швидкістю і ін. Існує доволі велика низка факторів, які можуть впливати на частоту помилок слів: вимова, гучність, тембр, акцент, гучність фоновий шум.

Для розпізнавання мовлення в тексті та підвищення точності транскрипції використовуються різні алгоритми та методи обчислення. Найпоширеніші з них є: NLP, HMM, N-грама, нейронні мережі [24].

NLP – сфера штучного інтелекту, яка зосереджена на взаємодії між людьми та машинами через мову через мову та текст. Багато мобільних пристроїв включають у свої системи розпізнавання мовлення для здійснення голосового пошуку, або забезпечують розпізнання контенту під час обміну текстовими повідомленнями.

HMM – так звані приховані моделі Маркова – методи побудовані на моделі ланцюга Маркова, який передбачає, що ймовірність певного стану залежить від поточного стану, а не від його попередніх станів. У той час як модель ланцюга Маркова корисна для подій спостереження, таких як введення тексту, приховані моделі Маркова дозволяють нам включати приховані події, в імовірнісну модель [24]. Вони використовуються як елементи послідовності під час розпізнавання мовлення, призначаючи мітки кожній одиниці, тобто словам, складам, реченням тощо. Ці мітки створюють відображення з наданими вхідними даними, що дозволяє визначити найбільш відповідну послідовність міток.

N-грама – це майже найпростіший тип мовної моделі, яка призначає ймовірності реченням або фразам. Уявляє собою послідовність N-слів. Правила граматики та ймовірності певних слів використовуються для покращення розпізнавання та точності.

Нейронні мережі – використовуються для алгоритмів глибокого навчання, обробляють навчальні дані, імітуючи взаємозв'язок людського мозку через рівні вузлів. Кожен вузол складається з входів, ваг, зміщення (або порогу) і виходу. Якщо це вихідне значення перевищує заданий поріг, воно «спрацьовує» або активує вузол, передаючи дані на наступний рівень у мережі. Нейронні мережі вивчають цю функцію відображення через контрольоване навчання, коригуючи на основі функції втрат через процес градієнтного спуску.

Нейронні мережі, як правило, точніші та можуть приймати більше даних – це відбувається за рахунок ефективності продуктивності, оскільки вони, як правило, повільніше навчаються порівняно з традиційними мовними моделями [24].

Сьогодні існує декілька методів виділення ознак мовлення, які зазвичай використовуються в системах нейронного розпізнавання. Вони включають лінійні кепстральні коефіцієнти прогнозування (Linear Predictive Cepstral Coefficients), мел-частотні капстральні коефіцієнти (melfrequency cepstral coefficients), перцептивний лінійний прогнозний аналіз, диференціальні коефіцієнти першого та другого порядку та відносний спектральний аналіз.

На відміну від одновимірних характеристичних параметрів, спектрограма є більш інтуїтивно зрозумілим, компактним та ефективним представленням, яке несе інформацію про акустичну характеристику у формі двовимірної моделі та включає багаті акустичні характеристики, такі як енергія, висота тону, основна частота, і формант. Ці функції є цінними для автоматизованих систем розпізнавання мовлення, а також є широко використовуваними інструментами для аналізу мовлення, де

використовується спектрограма як акустична функція в поєднанні з методом штучної нейронної мережі для розпізнавання мови [25].

Серед нейронних мереж виділяють три найпопулярніші: штучну нейронну мережу (ANN), згорткову нейронну мережу (CNN) і рекурентну нейронну мережу (RNN) (табл. 1.7) [26].

Таблиця 1.7 – Порівняльне зіставлення ANN, CNN, RNN [26]

	ANN	CNN	RNN
Загальна характеристика	Один з найпростіших	Один з найпопулярніших	Найбільш складний
Структурна особливість	Інформація тече лише в одному напрямку	Базується на кількох шарах вузлів, вкл. один або більше згорткових рівнів	Інформація тече в різних напрямках, властиве самонавчання
Тип даних	Табличні і текстові данні	Зображення	Навчання з даними послідовності
Складність	Вважається простим	Вважається потужнішим	Вважається потужним через потенціал самонавчання
Особливості	Вміння роботи з неповними знаннями	Точність розпізнавання образів	Пам'ять і самонавчання
Головний недолік	Апаратна залежність	Потреба в великій кількості навчальних даних	Складне навчання
Використання	Вирішення комплексних проблем таких як прогнозний аналіз	Розпізнавання зображень	Високоточний аналіз мовлення, настроїв, швидкості та ін.

Беручи до уваги порівняльний аналіз нейронний мереж (табл. 1.7), а саме: націленість на обробку зображень, точність розпізнавання та потужні обчислювальні функції – для розпізнавання спектрограм голосу рекомендовано обирати саме згорткову нейронну мережу.

1.3.3 Проблематика існуючих апаратно-програмних рішень

Споживачі, які володіють IoT пристроями, поступово розширюють вже існуючий набір девайсів купуючи нові та поступово замислюються над застосуванням програми для керування ними. За даними дослідження Parks Associates, 50% власників розумних домашніх пристроїв мають пристрій, який контролює кілька розумних продуктів. Згідно до їх дослідження серед пристроїв для керування провідні позиції займають Amazon Echo та Google Home (рис. 1.9) [27].

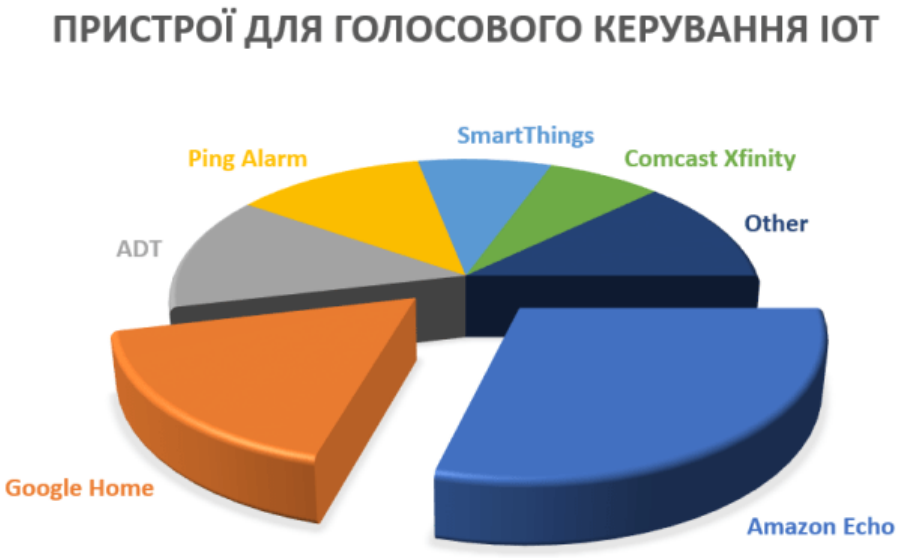


Рисунок 1.9 – Рейтинг пристроїв керування IoT пристроями [27]

Google Home і Amazon Echo пропонують власні інтелектуальні функції, через своїх цифрових помічників – Google Assistant або Amazon Alexa відповідно.

За своєю суттю вони мають дуже подібний між собою функціонал хоча і містять певні відмінності. Наприклад, Amazon містить можливість відстежувати замовлення на Amazon, виявляти роздратування у вашому голосі або – якщо у приміщенні тиша ніби всі сплять – шепотіти свої відповіді. Тим часом Google краще розуміє людську мову, може виконувати декілька наказів, які міститься в одному реченні як то вимкнути освітлення і музику без необхідності видавати обидві інструкції окремо.

Користувачі зазначають що найчастіше існує необхідність у декількох пристроях в квартирі адже він не здатних відслідковувати великі площини. Слід зазначити що хоча технічно існує можливість об’єднувати колонки Google і Alexa в одному приміщенні – за відгуками користувачів це здатне викликати певні непорозуміння, то ж краще віддавати перевагу конкретному типу помічника [28].

Широкі функціональні можливості здатні задовільнити багато користувацьких запитів, але незважаючи на доволі великий функціонал голосових асистентів користувачі виділяють дві провідні проблеми: відсутність інтернету та питання конфіденційності.

У разі відсутності інтернету зазначені голосові помічники втрачають свій функціонал адже вони розраховані на використання зовнішніх розрахункових потужностей, так званих «хмарних сервісов» для обробки голосових даних. Це призводить до того, що користувачі повинні вимкнути усі розумні у ручному режимі лише через те, що виникли перебої зі зв’язком. Питання доволі проблемне оскільки наприклад в Україні останнім часом перебої з інтернет мережею зустрічаються доволі часто про що свідчать числень статистичні дані: інколи падіння трафіку відбувається майже на половину – до 40 відсотків [29].

Окрім цього існує проблема щодо конфіденційності особистих біометричних даних. Деякі люди стурбовані тим, що, оскільки голосові помічники постійно прослуховують навколишнє середовище, вони представляють собою потенційну загрозу вторгнення в конфіденційність,

пристрій стеження за користувачем. Питання настільки гостро підіймається у суспільстві що виробники додали можливість видаляти файли історії голосового пошуку, але ця функціональність не вирішує того, що обробка даних відбувається віддалено [30].

1.4 Метод забезпечення стабільної роботи розпізнавання голосових команд

Беручи до уваги статистичні данні з використання кінцевими користувачами голосових команд для керування IoT пристроями (табл. 1.6), а, також, прогнозоване зростання попиту на програмне забезпечення для розпізнавання мовлення (п. 1.3.1) стає очевидним актуальність дослідження і питання з вдосконалення існуючих алгоритмів та програмно-апаратних засобів, які спрямовані на забезпечення розпізнавання голосових команд.

За результатами аналізу проблематики апаратно-програмних рішень (п.1.3.3), що випускаються можна стверджувати, що сьогодні існує невирішене питання забезпечення роботи пристрою голосового керування без доступу до мережі Інтернет. Зазначене питання набуває особливої актуальності на території Україні, оскільки останнім часом спостерігаються систематичні перебої зі зв’язком і доступом до мережі Інтернет (п. 1.3.3).

Пропонується удосконалити метод розпізнавання голосових команд за рахунок поєднання у програмному забезпеченні можливостей розпізнавання голосових команд як за допомогою хмарного сервісу так в офлайн режимі. Завдяки потужностями хмарного сервісу це дозволить зберегти доволі широкі функціональні можливості з розпізнавання, а офлайн режим буде гарантувати стабільність роботи базового функціоналу комплексу голосового керування не залежно від наявності Інтернет з’єднання, що буде якісно відрізняти АПК від провідних існуючих рішень.

Враховуючи проведене дослідження алгоритмів, які використовуються для розпізнавання людського мовлення (п. 1.3.2) з метою

забезпечення розпізнавання голосових команд в офлайн режимі пропонується використовувати згорткову нейронну мережу, в яку передавати аудіозаписи конвертовані у спектрограми (рис. 1.10).

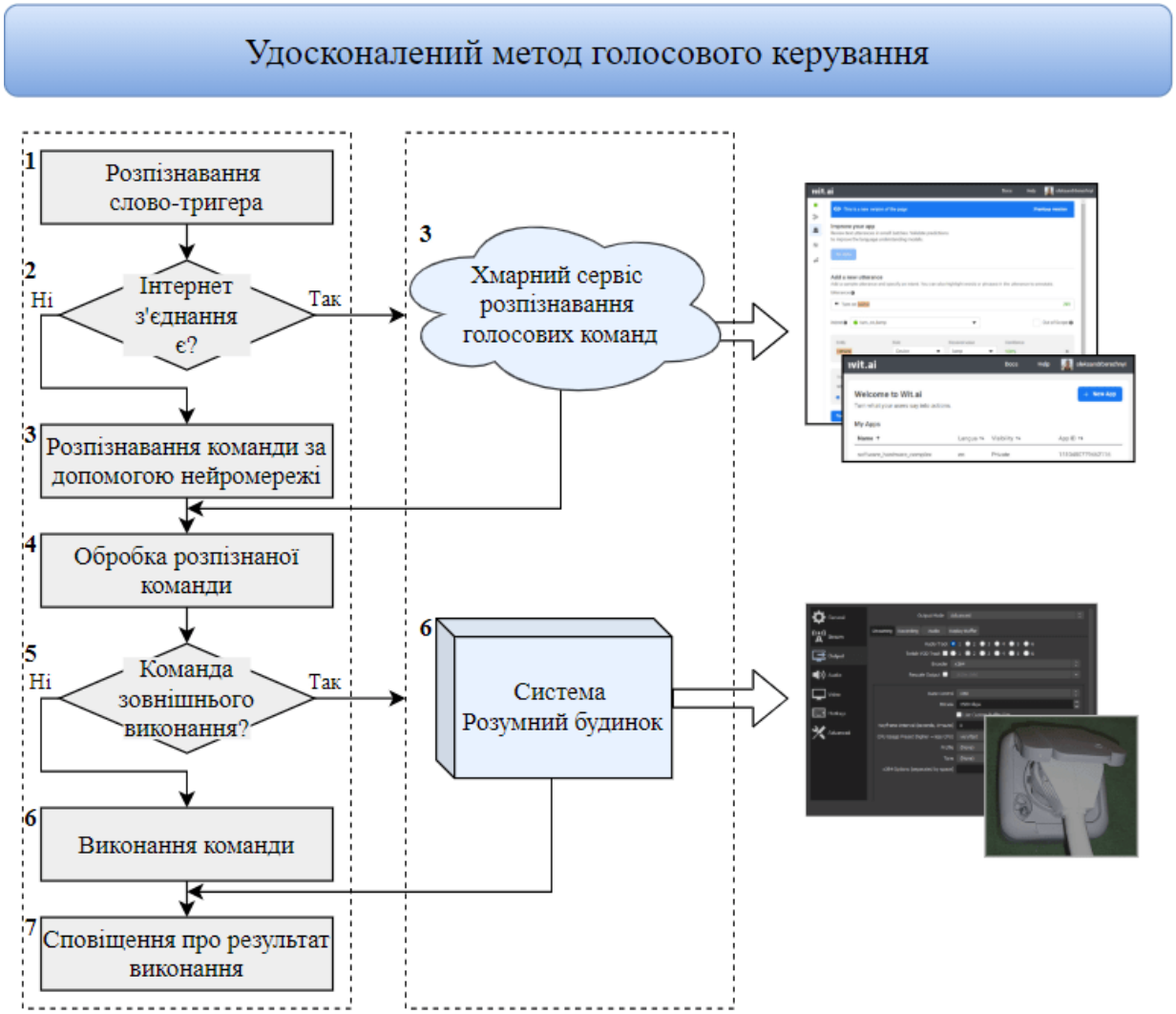


Рисунок 1.10 – Удосконалений метод керування СРБ на основі розпізнавання голосових команд

1.5 Постановка задач магістерської роботи

За результатами проведеного дослідження можна стверджувати, що IoT системи наразі зазнають стрімкого розвитку: в світі вже успішно створено та використовуються проєкти як приватного, так і державного рівня. Є приклади реалізації великих IoT проєктів і в Україні.

Збільшення кількості IoT пристроїв спонукає до впровадження нових методів та засобів керування ними. Одним з таких рішень є технології, які забезпечують можливість голосового керування. Вже існують певні проєкти голосових помічників, які отримали розповсюдження у суспільстві, але вони мають недоліки, що стримують їх розвиток і розповсюдження. Найбільший з них – неможливість роботи без підключення до мережі Інтернет.

Метою роботи є дослідження та програмна реалізація методів та засобів розпізнавання голосових команд для розширення функціональних можливостей СРБ та організації стабільної роботи як в онлайн, так і в офлайн режимі.

Для досягнення мети роботи необхідно вирішити такі завдання:

– удосконалити існуючий метод керування СРБ на основі розпізнавання голосових команд;
– виконати аналіз вимог та розробити архітектуру АПК;
– обрати апаратні та програмні засоби розробки;
– виконати програмну реалізацію;
– провести тестування АПК;
– розробити програмну документацію.

2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Аналіз вимог

Враховуючи результати проведеного аналізу проблематики інтерфейсів керування IoT системою, було визначено функціональні та нефункціональні вимоги до АПК (рис. 2.1), реалізація яких буде здатна суттєво полегшити взаємодію користувача з розумними IoT системами.

Рисунок 2.1 – Вимоги до АПК

З метою забезпечення можливості керування IoT системою за допомогою голосу АПК повинен мати функціонал розпізнавання голосових

команд. Враховуючи можливість втрати з’єднання IoT системи з мережею Інтернет слід забезпечити функціонал для розпізнавання базового набору команд в офлайн режимі, без з’єднання з зовнішніми сервісами. При наявності з’єднання з мережею Інтернет потрібно використати онлайн сервіс розпізнавання, оскільки це дозволить знизити інтенсивність навантаження на АПК та підвищити його ресурс. Також залучання зовнішнього сервісу збільшить кількість можливих команд та зменшить час на налаштування системи.

Оскільки люди використовують спілкування як засіб комунікації у соціумі постійне розпізнавання та аналіз навколишнього середовища може спричинити надмірне навантаження АПК. З метою уникання ситуації перенавантаження необхідно впровадження активації режиму розпізнавання команди лише після вживання слова-тригера.

Необхідно забезпечити можливість виконання розпізнаних команд як за допомогою вбудованих засобів керування, так і за допомогою запитів на зовнішні керуючі пристрої.

Враховуючи вірогідність того, що результат виконання голосових команд не завжди може бути одразу помітний для користувача (як то «ввімкнути обігрівач») необхідно впровадити зворотній зв’язок з користувачем, як то: «команда виконана», «команда не розпізнана».

Беручи до уваги можливу необхідність модернізації АПК чи зміни налаштувань, доречним буде інтеграція функції дистанційного оновлення програмного забезпечення, що здатне суттєво полегшити процес оновлення програмного забезпечення

З метою подальшого аналізу для покращення системи а, також, для відновлення ланки подій у випадку виникнення проблемної ситуації потрібно забезпечити ведення журналу історії голосових команд. Оскільки збої в енергомережі можуть залишити систему без зовнішнього живлення журнал повинен зберігатись в енергонезалежній пам’яті.

Спираючись на проведений аналіз мережевих протоколів та особливостей зв’язку IoT пристроїв, а також задля забезпечення мобільності системи з’єднання АПК з мережею повинно бути реалізовано бездротовим способом.

В АПК слід передбачити встановлення параметрів налаштувань за замовчуванням для безперешкодного відновлення роботи після перезавантаження чи зміни джерела живлення.

Окрім цього потрібно врахувати можливу зашумленість простору як то працюючий вентилятор чи кондиціонер, тобто наявність джерел шуму у приміщенні з АПК.

Враховуючи проаналізовані вимоги було складено діаграму прецедентів взаємодії користувача з АПК (рис. 2.2).

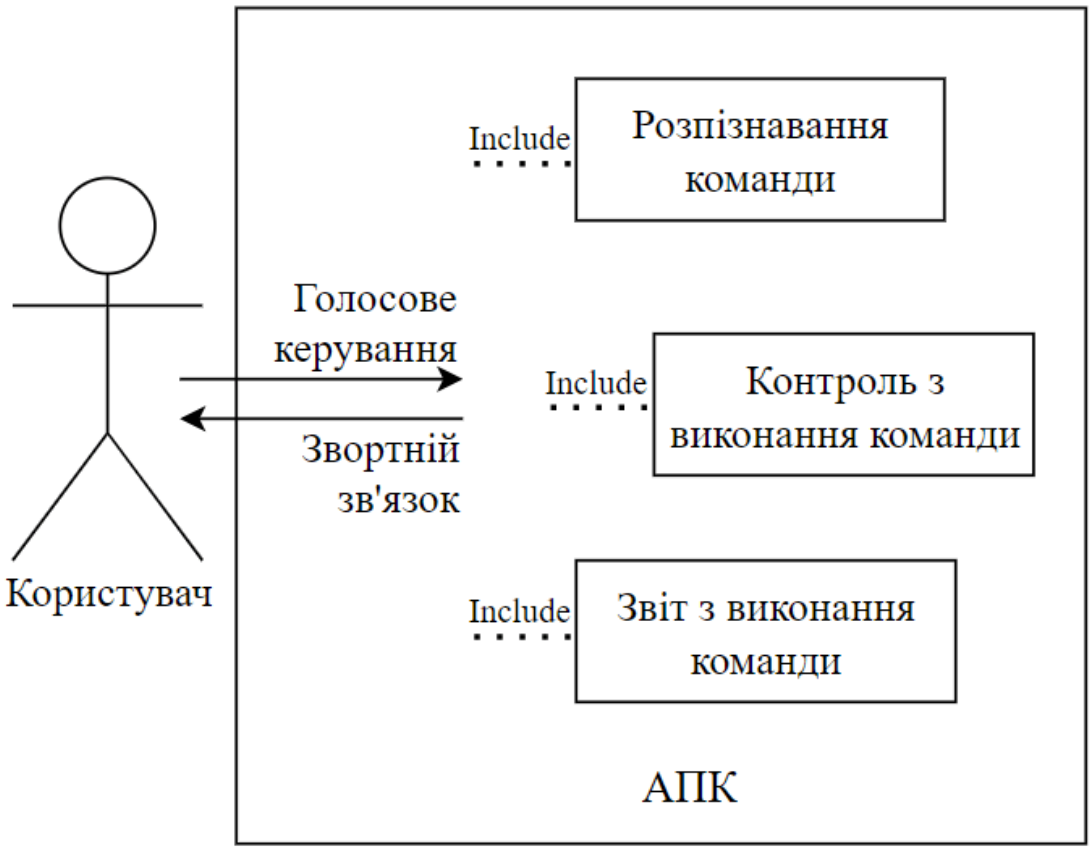


Рисунок 2.2 – Діаграма прецедентів взаємодії користувача з АПК

Засновуючись на проведеному аналізі вимог може бути спроєктовано архітектуру АПК.

2.2 Розробка архітектури

При розробці архітектури було враховано проведений аналіз взаємодії компонентів, особливості існуючих протоколів а також висунутих вимог до АПК. На рисунку 2.3 наведено розроблену архітектуру АПК.

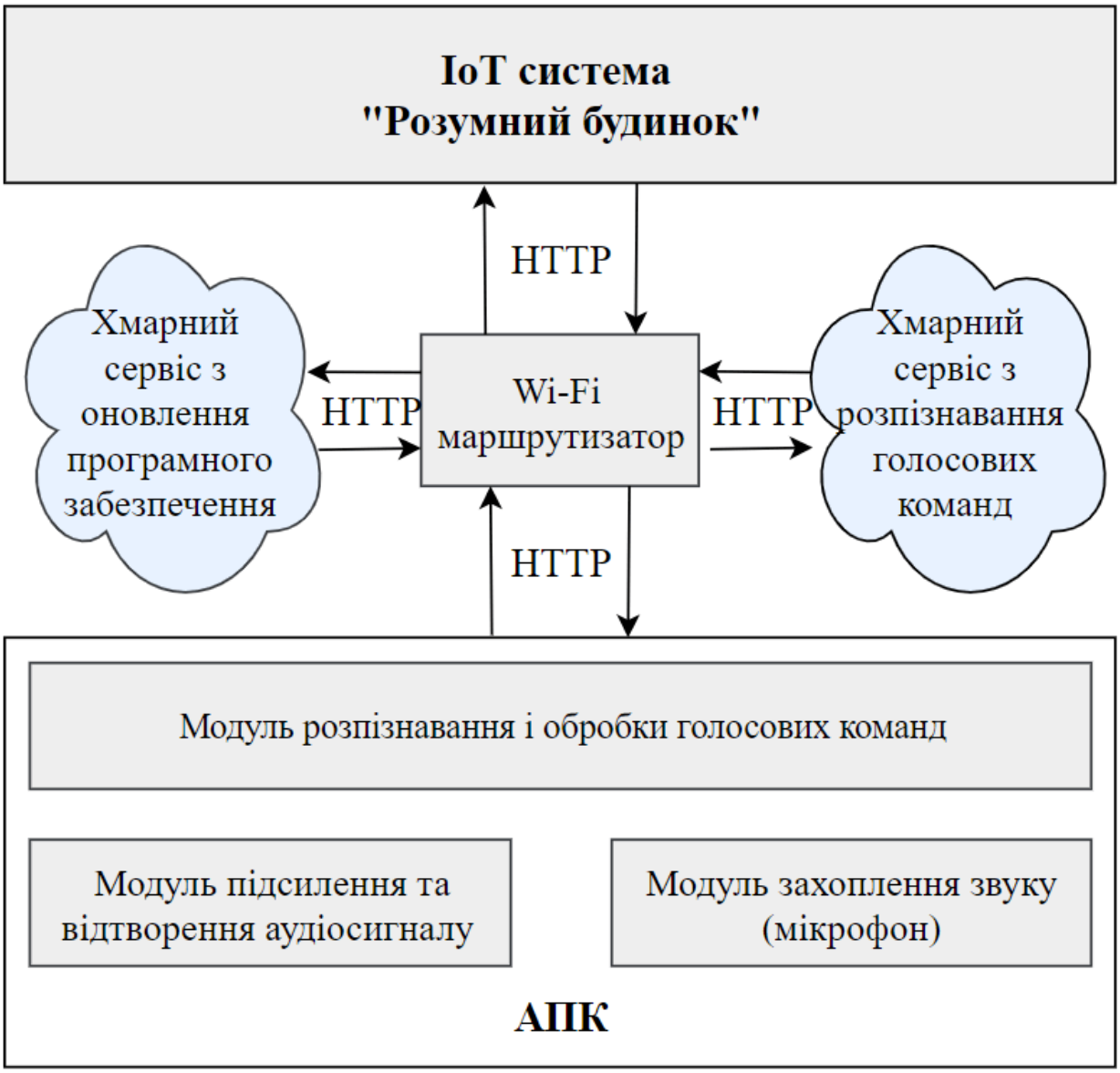


Рисунок 2.3 – Архітектура АПК

Комплекс складається з:

– модулю захоплення звуку;
– модулю розпізнавання, аналізу і обробки голосових команд;
– модулю підсилення та відтворення аудіосигналу.

Взаємодія з зовнішніми сервісами буде відбуватись за допомогою Wi-Fi маршрутизатору та мережі Інтернет.

2.3 Вибір засобів розробки

Процес розробки АПК необхідно починати з дослідження апаратних засобів, які буде використано. Спираючись на обрані апаратні засоби та їх технічні особливості буде визначено програмні засоби. Зазначена послідовність є обов’язковою оскільки функціональні можливості і розрахункова потужність апаратних засобів може накладати обмеження на перелік технологій та протоколів, що можуть бути сумісні, та, відповідно, задіяні при розробці.

2.3.1 Апаратні засоби

Головною складовою АПК є модуль розпізнавання і обробки голосових команд, тому його розглянемо першим.

Для забезпечення проведення розрахунків модуль повинен бути з інтегрованим мікроконтролером. Було проведено аналіз існуючих на ринку апаратних рішень: модулі з інтегрованими мікроконтролерами AVR, ESP та інші (табл. 2.1) [31].

За результатами порівняння для обробки голосових команд було обрано модуль з інтегрованим мікроконтролером ESP-32 (рис. 2.4) оскільки він забезпечує найліпше співвідношення розрахункової потужності, об’єму пам’яті та вартості, має вбудовану можливість роботи з Wi-Fi.

Рисунок 2.4 – Модуль ESP-32

Таблиця 2.1 – Зведена порівняльна характеристика модулів, які містять інтегровані мікроконтролери [31]

Мікроконтролер / Функціонал	ESP-32	ESP-8266	Arduino UNO	Arduino Mega 2560
1	2	3	4	5
Модель	Xtensa Dual Core 32-bit LX6	Tensilica 32-bit Xtensa LX106	ATMega328P	ATMega2560
Флеш-пам'ять	4 МБ	4 Мб	32 КБ	256 Кб
SRAM	520 Кб	128 Кб	2 Кб	8 Кб
EEPROM	512 Б	512 Б	1 КБ	4 Кб
Тактова частота	До 240 МГц	80 / 160 МГц	16 МГц	16 МГц
Робоча напруга	3,3 В	3,3 В	5 В	5 В
Wi-Fi	802.11 b/g/n	802.11 b/g/n	-	-
Bluetooth	Bluetooth v4.2, BLE	-	-	-

Продовження таблиці 2.1

1	2	3	4	5
Споживання	80 мА – 90 мА	15 мкА - 400 мА	45 мА – 80 мА	150 мА
ШІМ	16	4	6	14
UARTs	3	2	1	4
Аналогові піни	До 18	1	6	16
Цифрові піни вводу-виводу	36	17	14	54

Враховуючи функціональні вимоги, АПК необхідно забезпечити модулем захоплення звуку. Існує декілька варіантів отримання аналогових аудіоданих (захоплення звуку) для ESP-32:

– пряме зчитування з вбудованих аналого-цифрових перетворювачів (АЦП), добре підходе для одноразових зчитувань, але демонструє низькі результати для високих частот дискретизації;

– використання I2S (стандарт інтерфейсу електричної послідовної шини для з’єднання цифрових аудіопристроїв) з метою читання з вбудованих АЦП завдяки DMA (прямого доступу до пам’яті);

– використання I2S для безпосереднього читання з I2S-сумісних периферійних пристроїв.

АЦП ESP-32 має певні похибки, а калібрування передбачено відбувається на заводі-виробнику. Для кінцевого користувача калібрування є відносно складним процесом та потребує певних навичок. Як зазначалось вище безпосереднє використання АЦП підходить для низькочастотної та одноразової дискретизації. Використання I2S в комбінації з DMA показує дещо кращі результати, але теж містить певні недоліки, такі як підвищена зашумленість сигналу.

Для захоплення високоякісних аудіоданих кращим підходом є використання вбудованого I2S для зчитування зразків з зовнішнього MESM модулю [32].

Мікрофон MEMS типу – електроакустичний перетворювач, що містить датчик і спеціальну інтегральну схему (ASIC) в одному корпусі.

Датчик перетворює змінний вхідний звуковий тиск у зміну ємності, яку ASIC перетворює на аналоговий або цифровий вихід. Акустична хвиля потрапляє в мікрофон через невеличкий отвір у корпусі. Наша найкраща в своєму класі точка акустичного перевантаження (AOP), співвідношення сигнал/шум (SNR) роблять MEMS-мікрофони придатними для пристроїв, які вимагають дуже високого динамічного діапазону. [33].

Одним з найкращих MEMS мікрофонів для ESP-32 є INMP441 (рис. 2.5). Він має доволі високий SNR і чутливість, що робить його чудовим вибором для застосування в захопленні звуку у невеликих приміщеннях. INMP441 має рівну широкосмугову частотну характеристику, що забезпечує природний звук із високою розбірливістю [32].

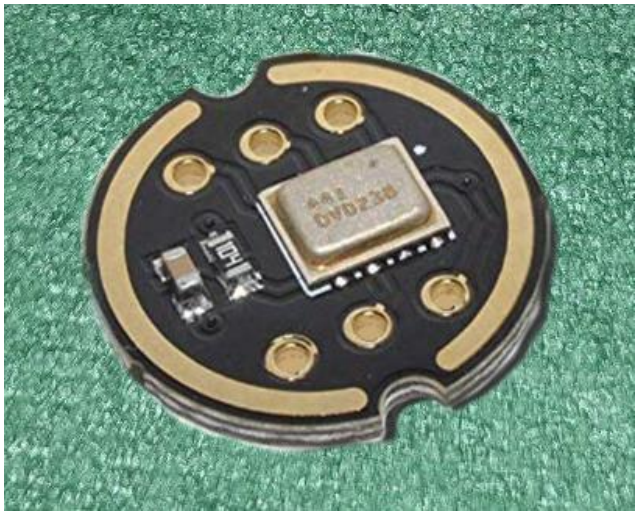


Рисунок 2.5 – Модуль захоплення аудіо

Для підвищення наочності на рисунку 2.6 зображено схему з’єднання ESP-32 з INMP441. Зіставлення пінів зазначено у таблиці 2.2.

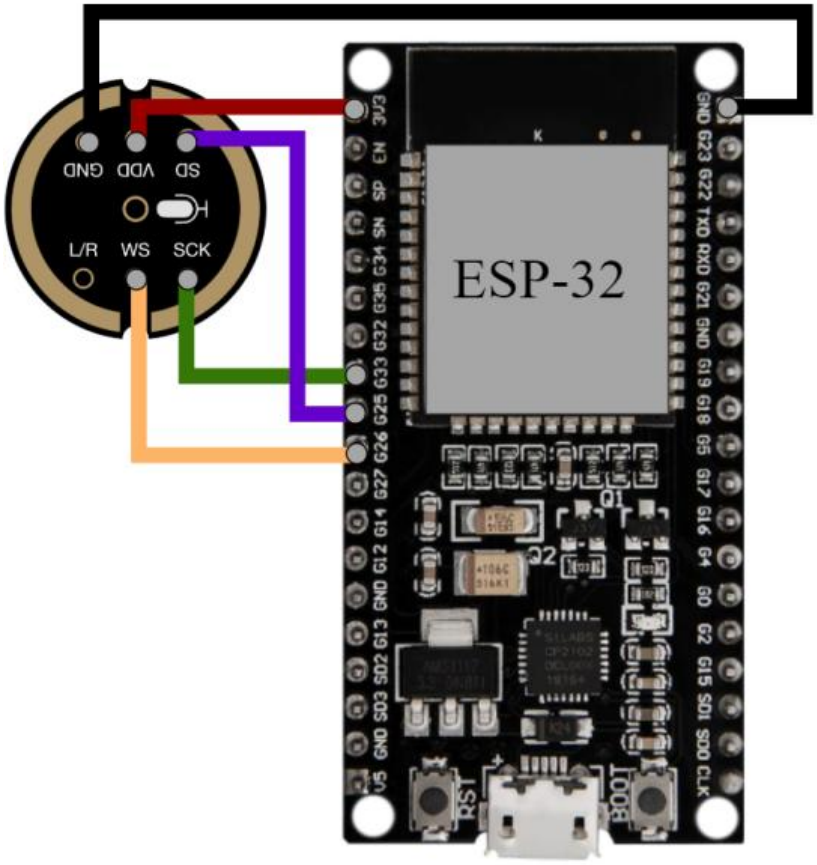


Рисунок 2.6 – Схема підключення ESP-32 до INMP441

Таблиця 2.2 – Зіставлення пінів для з’єднання ESP-32 з модулем MEMS мікрофону INMP441

Піни	
ESP-32	INMP441
3.3B	VDD
GND	GND
G33	SD
G25	SCK
G26	WS

Через архітектурні особливості ESP-32 аудіо вихід відбувається шляхом генерації сигналу за I2S протоколом. Отже необхідно залучити модуль, який буде декодувати I2S у аналоговий сигнал і підсилювати його.

Одним з кращих серед існуючих рішень для ESP-32 є модуль MAX98357A, який здатний забезпечити 3,2 Вт потужності в гучномовці з опором 4 Ом (рис. 2.7) [34].

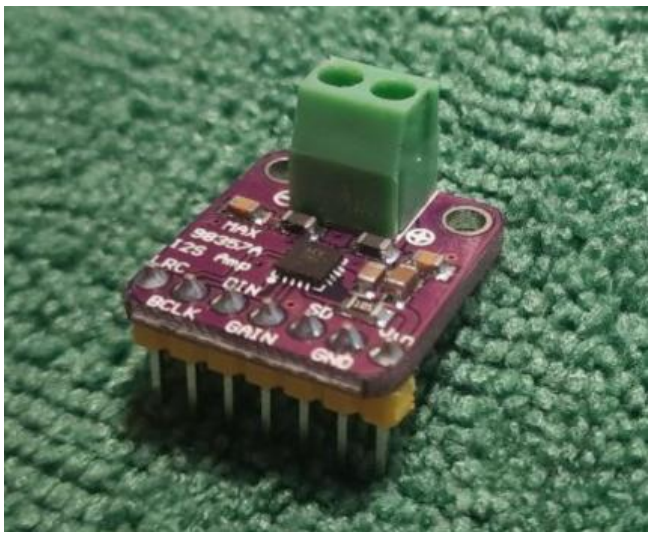


Рисунок 2.7 – Модуль декодування та підсилення аудіосигналу MAX98357A

Він містить контролер класу D, який може працювати від 2,7 В до 5,5 В постійного струму, має вбудований захист від перегріву та перевантаження по струму. Схема під’єднання MAX98357A до ESP-32 зображена на рисунку 2.8. Зіставлення пінів для з’єднання ESP-32 з модулем MAX98357A зазначено у таблиці 2.3.

Після під’єднання модулю підсилення аудіосигналу додається гучномовець. Загальна схема зображена на рисунку 2.9.

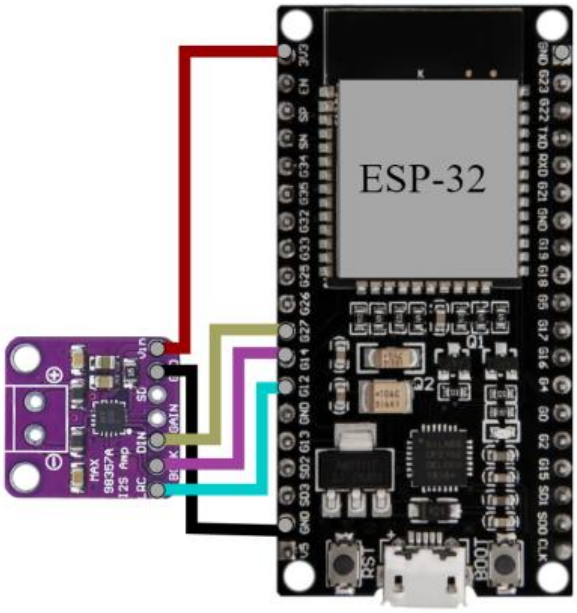


Рисунок 2.8 – Схема підключення ESP-32 до MAX98357A

Таблиця 2.3 – Зіставлення пінів для з’єднання ESP-32 з модулем MEMS мікрофону INMP441

Піни	
ESP-32	MAX98357A
3.3B	VDD
GND	GND
G27	DIN
G14	BCLK
G12	LRC

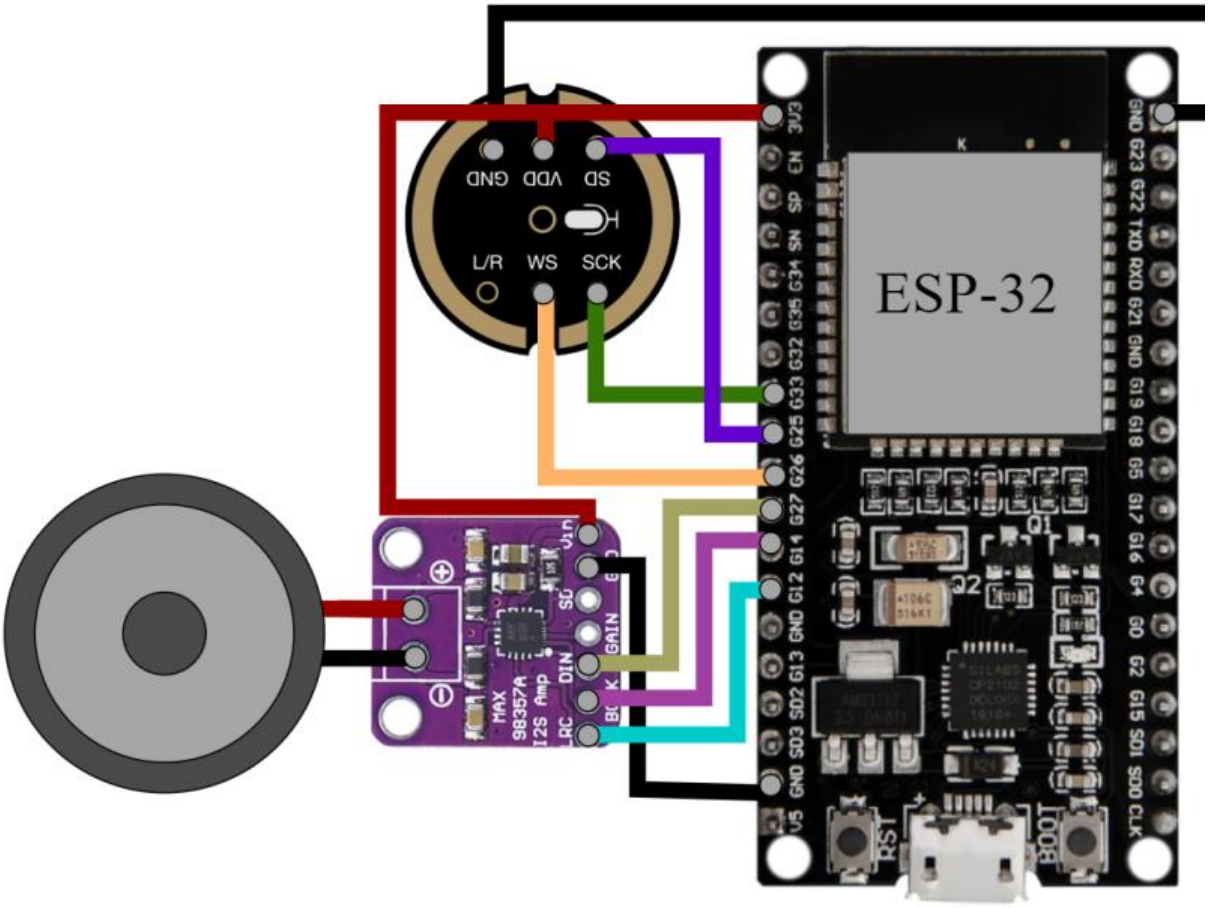


Рисунок 2.9 – Загальна схема підключення АПК

2.3.2 Програмні засоби

На основі проведеного аналізу апаратних засобів як програмований модуль було обрано SoC модуль ESP-32. Відповідно при виборі програмних

засобів необхідно враховувати особливості розробки для мікроконтролерів даної серії та платформи.

Оскільки платформа ESP була створена більш ніж декілька років назад і встигла отримати доволі широку розповсюдженість – сьогодні на ринку програмних рішень вже існує декілька варіантів мов програмування і програмних платформ які можна використати для написання програми.

На сьогодні активно застосовуються для програмування ESP модулів мови: Micropython, C, C++, JavaScript, LUA (рис. 2.10) [35].

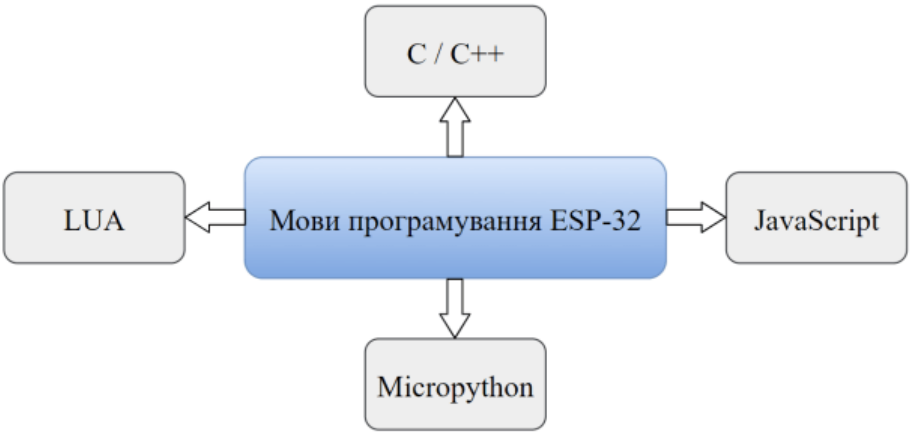


Рисунок 2.10 – Мови програмування для ESP-32 [35]

MicroPython – це мова об’єктно-орієнтованого сценарію, яка направлена на написання програм для мікроконтролерів та розроблена на основі ядра Python. Вона включає невеликий набіл стандартної бібліотеки Python. Як основну перевагу виділяють спрямованість на зручність написання коду для новачків та високу наочність. При необхідності у MicroPython може бути інтегровано більш потужні мови, як C або C++ з метою підвищення швидкості роботи або задіянню більш специфічного функціоналу. Через відносно простий синтаксис програма написана на MicroPython поступається у швидкодії та використаній пам’яті програмі що буде написана за C / C++.

LUA – асинхронна мова програмування сценаріїв, яка підтримує процедурне і функціональне програмування. LUA спирається на керування та

виконання завдань що пов’язані з певними подіями. Події, які вистроєні у чергу будуть оброблені по черзі та для кожної події буде виконано усі пов’язані завдання. Обробка та виконання відбувається за формулою FIFO. Серед недоліків, які властиві даній мові виділяють потребу в контролі за часом виконання завдань, адже якщо завдання виконується довго – це може призвести до порушення стабільності даних (як то підтримання Інтернет з’єднання, тощо).

JS – мультипарадигмена мова програмування з підтримкою об’єктно-орієнтованого програмування. Зазвичай використовується у веб застосунках та браузерах. Враховуючи потребу керування IoT приладами через інтернет – інколи буває задіяна для написання графічного інтерфейсу керування пристроїм. Мові властиві відносна легкість сприйняття синтаксису та висока розповсюдженість.

C++ – об’єктно орієнтована мова програмування, яка отримала широке розповсюдження в програмуванні мікроконтролерів через можливість високої оптимізації та безпосереднім керуванням розподілу пам’яті і даних. Це забезпечило високу популярність мови, а, отже, велику кількість бібліотек, посібників, прикладів використання, що здатне якісно поліпшити розробку програмного продукту, знизити витрати на процес розробки. Окрім цього наявність можливості оптимізації ресурсів забезпечує високу швидкодію та здатність проводити потужні розрахункові операції. Один з шляхів застосування мови C є її поєднання з операційною системою RTOS. Система мультизадачна, тобто дозволяє виконувати більш ніж одну задачу в режимі псевдоодночасного виконання. Завдяки планувальнику операційна система розподіляє завдання в визначеній області стеку, що дозволяє краще спрогнозувати використання ресурсів.

Неодноразово доведено чисельними тестами, що для програмування мікроконтролерів C++ має впевнену перевагу над іншими мовами високого рівня з таких параметрів як швидкодія та реалізація розрахункових потужностей [36]. Оскільки розпізнавання голосових команд буде

відбуватись за методом аналізу спектрограми показовим є порівняння швидкості обробки зображень на різних мовах програмування (табл. 2.4) [37].

Таблиця 2.4 – Порівняння швидкості обробки зображення на ESP платформі при використанні різних мов програмування [37]

Мова	Кількість кадрів оброблюємих в секунду
C++	35
MP	7 – 8
JS	5-7
Lua	4

Враховуючи проведений аналіз а також вимоги до системи що розробляється, а саме – необхідність в проведенні аналізу мовлення власними потужностями мікроконтролером в якості мови програмування обрано C++.

Провідними середовищами розробки програмного продукту мовою C++ для модулів на базі ESP є Arduino IDE і PlatformIO (рис 2.11).

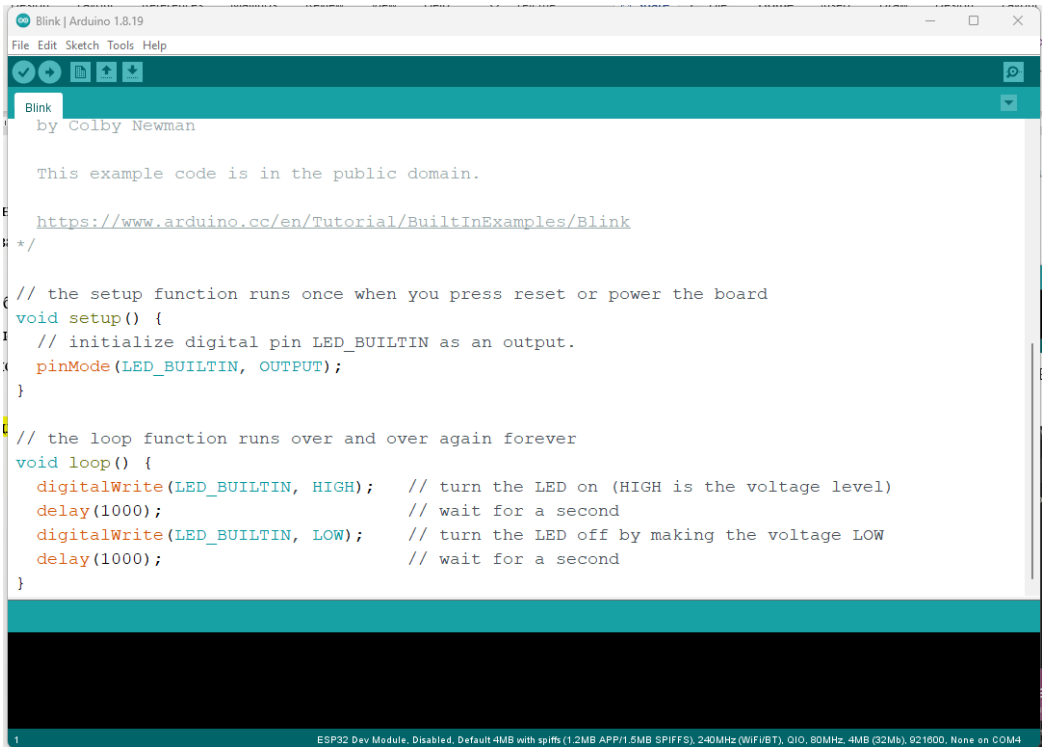


Рисунок 2.11 – Інтерфейс Arduino IDE

Arduino IDE – середовище розробки для написання та встановлення програмних продуктів до Arduino сумісних модулів, та модулів іших типів за умови наявності відповідних драйверів (рис. 2.11). Серед інших програм відрізняється відносною легкістю до опанування, зручністю, простим та інтуїтивно зрозумілим інтерфейсом. Arduino IDE містить інтегрований менеджер бібліотек, який дозволяє встановлювати додаткові розширення та застосунки. Має низьку недоліків, серед яких виділяють невисоку швидкодію та обмежений функціонал, що пов’язано з його простотою. Не має кодової навігації, менеджера залежностей, автодоповнення коду, системи керування версіями програмного продукту [38].

Можна сказати що Arduino IDE уявляє собою зручний інструмент для початку опанування роботи з ESP мікроконтролерами, проте в великих проєктах існує ризик нестачі функціональних можливостей.

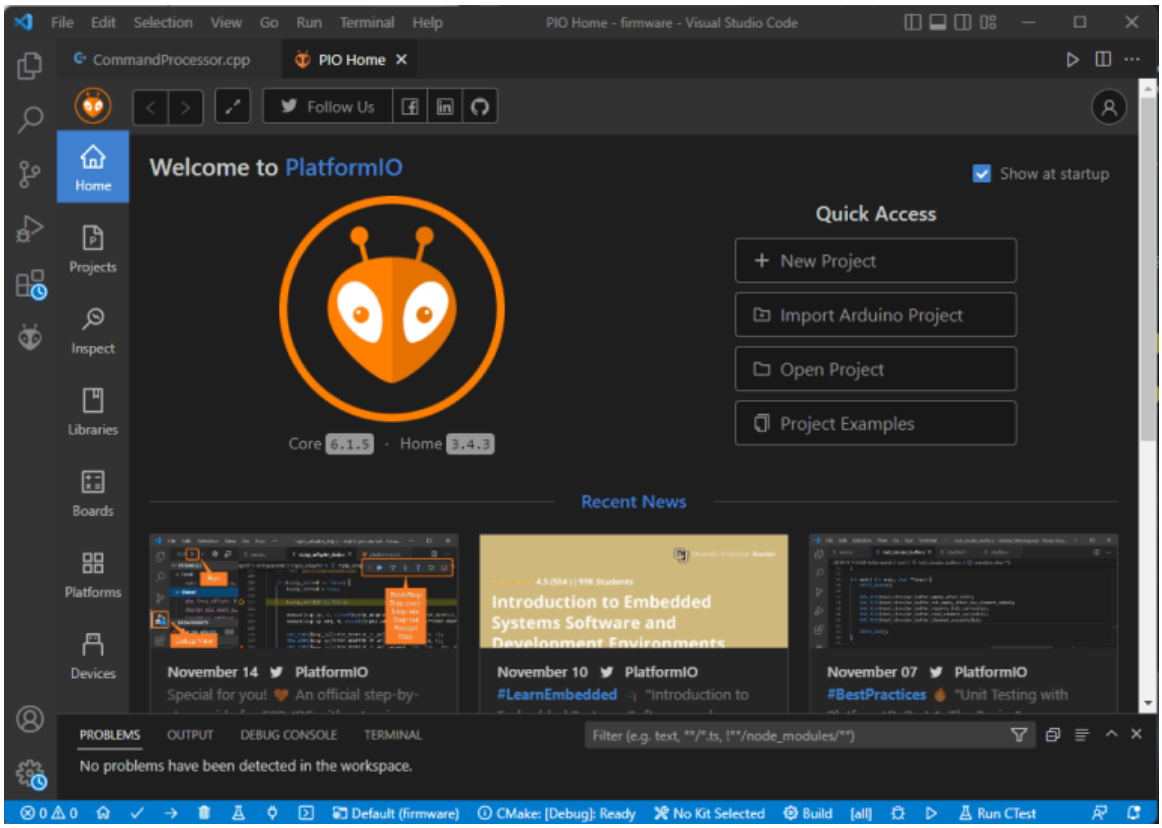


Рисунок 2.12 – Інтерфейс PlatformIO IDE у VS-Code

PlatformIO IDE – середовище розробки для написання та встановлення програмних продуктів на мікроконтролери IoT приладів (рис. 2.12). Містить уніфікований налагоджувач. Існує декілька редакторів коду до яких може бути встановлено PlatformIO IDE, проте розробники IDE радять встановлювати PlatformIO разом з редактором від Microsoft VS-Code. На їх думку така комбінація здатна найкраще реалізувати можливості середовища розробки, серед яких: індивідуальні налаштування графічного інтерфейсу, автодоповнення, навігація за кодом, інтелектуальні поради на основі аналізу коду, систему контроля версії Git, базу бібліотек [39].

В порівняння до Arduino IDE (табл. 2.5) основним недоліком є ускладнена взаємодія що пов’язано з доволі навантаженим графічним інтерфейсом, потреба у певному часі на опанування IDE.

Таблиця 2.5 – Порівняння PlatformIO та Arduino IDE

Критерій / Середовище	PlatformIO	Arduino IDE
Налаштування інтерфейсу	наявне	слабке
Складність опанування	середня	низька
Автодоповнення	наявне	відсутнє
Навігація за кодом	наявне	відсутнє
Система відслідковування версій	наявне	відсутнє

За результатами проведеного аналізу існуючих середовищ розробки а також беручи до уваги об’єм проєкту та кількість функціоналу що розробляється – для розробки АПК найліпшим є використання PlatformiO IDE.

Найпопулярнішими мовами для програмування алгоритмів машинного навчання є Python, C++, Lua, R (рис.2.13) [40]. При обранні мови програмування необхідно також звернути увагу на платформу, яка буде використовуватись з нею, оскільки це має безпосередній вплив на вихідні результати як то швидкодія, зручність використання та ін.

Серед існуючих рішень якісно відрізняється мова програмування Python в поєднання з платформою TensorFlow (табл. 2.6). Як непрямим підтвердженням правильності вибору може виступати статистика використання, а саме згідно до опитування Ideamotive 57% науковців використовують саме мову Python для машинного навчання [41].

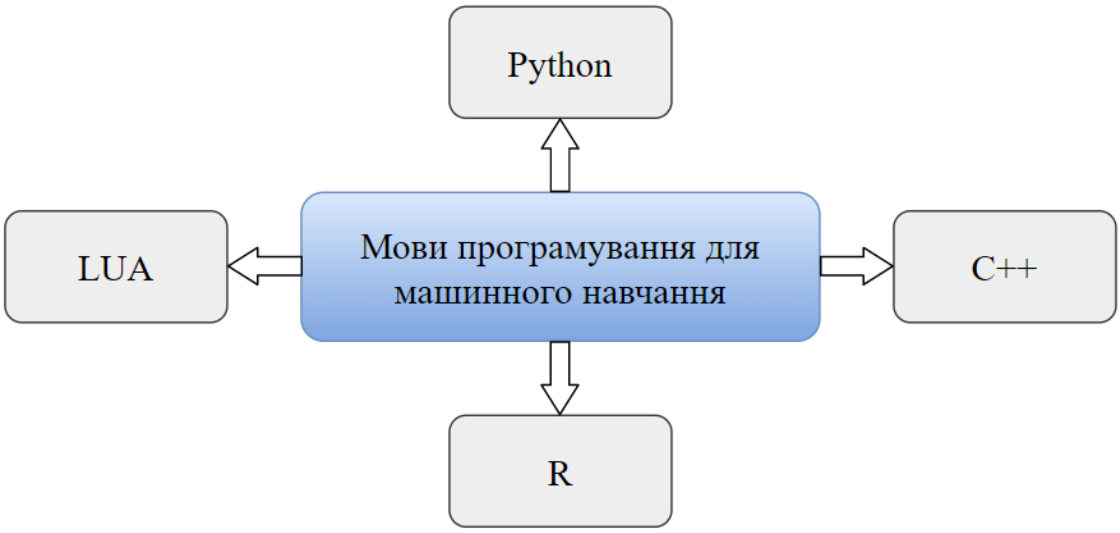


Рисунок 2.13 – Мови програмування для машинного навчання [41]

Таблиця 2.6 – Порівняння мов, платформ для машинного навчання[41]

Мова	Платформа	Підручники та навчальні матеріали	Архітектура: простота використання і модульність	Швидкість	Підтримка декількох GPU
Python	TensorFlow	+ + +	+ + +	+ +	+ +
Python	Theano	+ +	+	+ +	+
Lua	Torch	+	+ +	+ + +	+ +
C++	Caffe	+	+	+	+
C++	CNTK	+	+	+ +	+
R	MXNet	+ +	+ +	+ +	+ +

Оскільки для розробки розпізнавальної моделі голосових команд буде використано мову Python – необхідно обрати ще одне середовище розробки. Серед провідних середовищ виділяють Jupyter та Pycharm.

Jupyter – середовище розробки з відкритим вихідним кодом, використовується для інтерактивних обчислень з підтримкою декількох мов програмування: Juliad, Python, R. Запускається у веб-браузері (рис. 2.14).

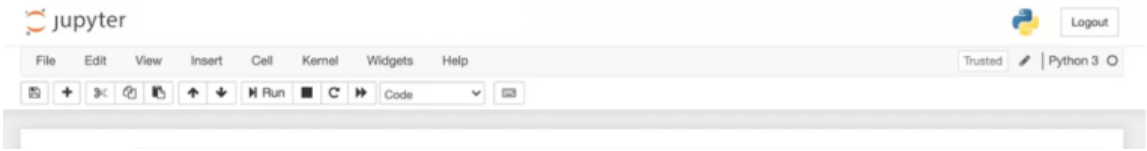


Рисунок 2.14 – Інтерфейс Jupyter

Pycharm – інтегроване середовище розробки для мови Python. Має доволі широкий функціонал, такий як аналіз коду, графічний налагоджувач, інтеграцію з системами контролю версій та інше (рис. 2.15).

Рисунок 2.15 – Інтерфейс Pycharm

Засновуючись на проведеному аналізі було створено порівняльну характеристику Jupyter та Pycharm (табл. 2.7) [42].

Таблиця 2.7 – Порівняння Jupyter та Pycharm [42]

	Jupyter	Pycharm
1	2	3
Визначення	Інтерактивна обчислювальна веб-платформа	Розумний редактор коду
Містить	Живий код, рівняння, описовий текст, візуалізації, інтерактивні інформаційні панелі та інші медіа	Забезпечує першокласну підтримку Python, JavaScript, CoffeeScript, TypeScript, популярної мови шаблонів тощо

Продовження таблиці 2.7

1	2	3
Класифікується як	Блокнот Data Science	Інтегроване середовище розробки (IDE)
Забезпечує	Виконання вбудованого коду за допомогою блоків Підтримку вбудованих графіків	Розумне автозавершення Інтелектуальний аналіз коду
Особливості	Блокова структура, підтримує kernel, а також latex	Потужний рефакторинг, інтеграція віртуальних середовищ та інтеграція Git
Гнучкість	Доволі гнучкий	Менш гнучкий, повільніший

За результатами порівняльного аналізу, враховуючи чисельні переваги, а також засновуючись на функціональних АПК було обрано Jupyter в якості середовища розробки для програмування моделі розпізнавання голосу.

Спираючись на результати проведеного аналізу з порівняння нейронних мереж (табл. 1.7) було обрано згорткову нейронну мережу для розпізнавання мовлення, оскільки вона має такі переваги як: націленість на аналіз зображень (у конкретному випадку - спектрограм голосу), відносно високою точністю розпізнавання та потужними обчислювальними функціями. Для підвищення точності даних для навчання моделі буде обрано базу голосових команд Google Speech Commands Dataset.

Окрім цього необхідно обрати хмарний сервіс для онлайн розпізнавання голосових команд. Найрозповсюдженішими сервісами сьогодні є Facebook’s Wit.ai, Google Dialogflow, Amazon Lex, IBM Watson Assistant, Microsoft Azure Bot Service. Серед яких якісно відрізняється Facebook’s Wit.ai [43].

Wit.ai – платформа для розпізнавання голосових команд від Facebook, яка надає фунціонал через взаємодію з IoT пристроями за допомогою HTTP протоколу (рис. 2.16). Від своїх аналогів відрізняється великою кількістю доступних для розпізнавання мов, орієнтованістю на технічно обізнаний персонал. Платформа має бібліотеки на декілька мов програмування. Безкоштовна як для особистого так і комерційного використання. Розробниками передбачено налаштування довільних команд для розпізнавання через графічний інтерфейс у веб-браузері.

Wit.ai цілком здатен задовільнити потреби АПК, а, отже, може бути використаний як сервіс для онлайн розпізнавання голосових команд.

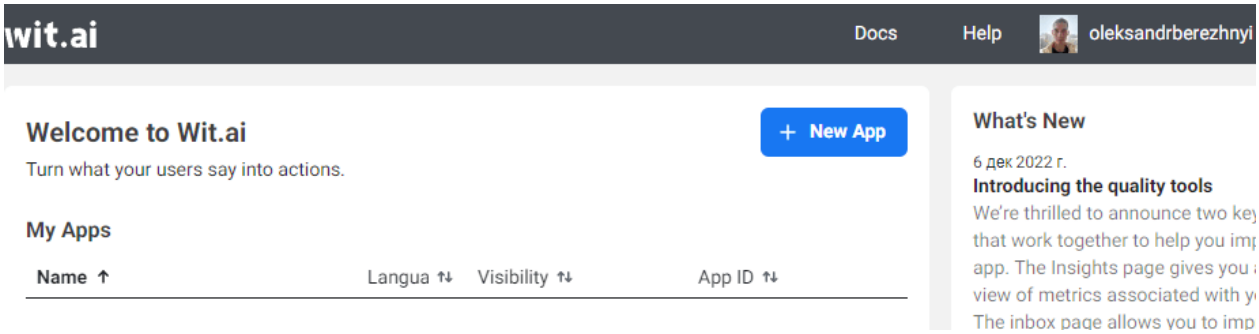


Рисунок 2.16 – Інтерфейс Wit.ai

При довгостроковій підтримці IoT продукту виникає потреба в оновленні програмного забезпечення з метою усунення виявлених програмних помилок та впровадженні нового функціоналу. Найпростіше мануальне оновлення потребує безпосередньої присутності спеціалісту поблизу IoT пристрою, що викликає надлишкові фінансові і часові витрати і є доволі складним при багатотиражному випуску продукції. Одним з шляхів оновлення програмного забезпечення з оптимальним використанням ресурсів є впровадження дистанційного оновлення.

Для забезпечення можливості дистанційного оновлення програмного забезпечення АПК доцільним є використання хмарного OTA сервісу – over-the-Air (рис. 2.17). Впровадження такого функціоналу призведе до

поліпшення роботи з IoT системою, пришвидшення процесу налагодження та зниження витрат на підтримку і обслуговування IoT пристрою.

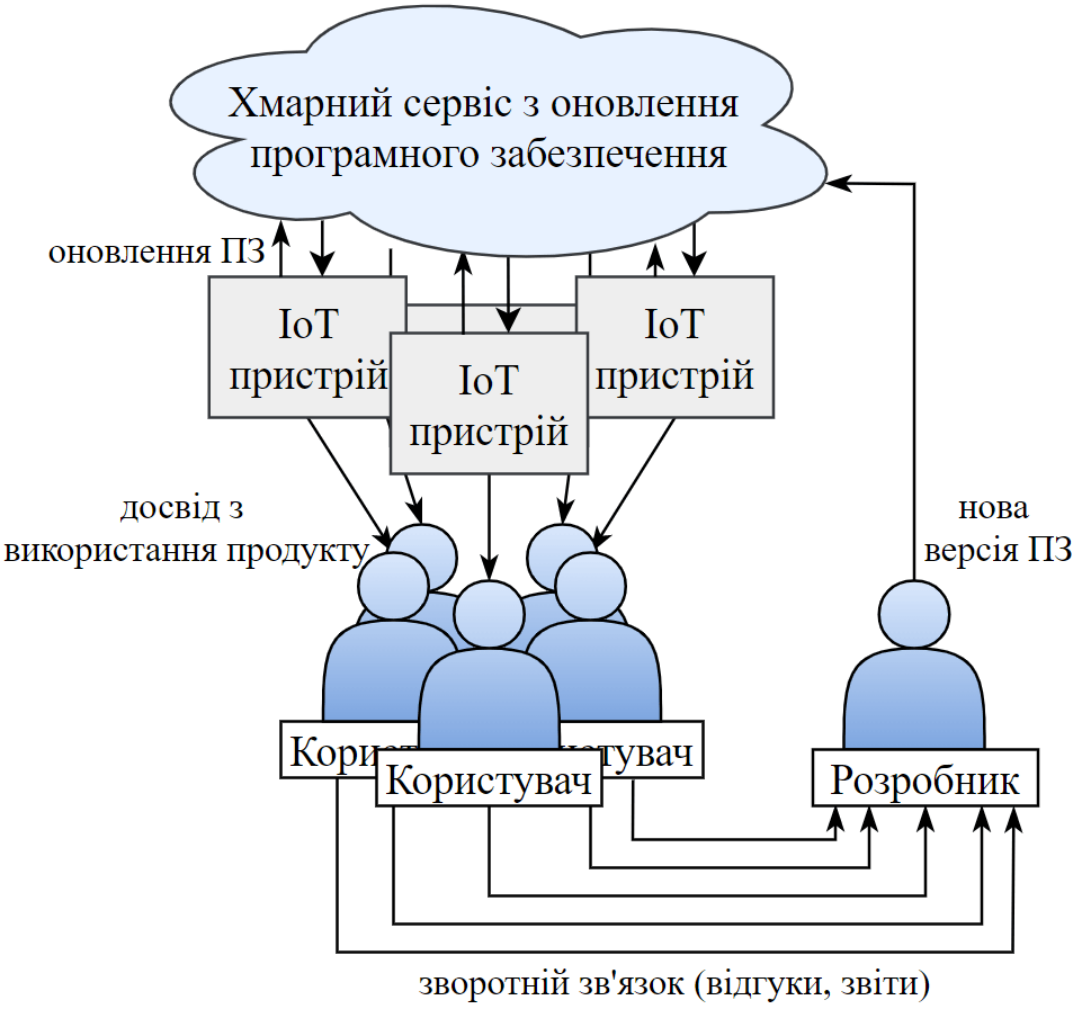


Рисунок 2.17 – Робота сервісу OTA оновлень

На сьогодні одними з найпоширеніших рішень серед дистанційних сервісів OTA оновлень є Google Cloud і OtaDrive.

Google Cloud – сервіс хмарного оновлення, який надає широкий спектр функціоналу включно з OTA оновленнями. Існує підтримка різних платформ, але через це дуже прискіпливий до налаштування, що значно ускладнює роботу з ним.

OtaDrive – сервіс для віддаленого оновлення програмного забезпечення апаратно програмних рішень на базі ESP модулів за технологіями OTA. Через відносно вузьку направленість має доволі простий і

зручний інтерфейс і потребує мінімальну кількість налаштувань. Присутня можливість оновлення як конкретного пристрою так і цілого набору. За для забезпечення розгалуження за рівнями доступу та підвищенням безпеки функціонування сервісу OTA drive має дворівневу авторизацію IoT пристроїв: за серійним номером та за ключем доступу (рис. 2.18).

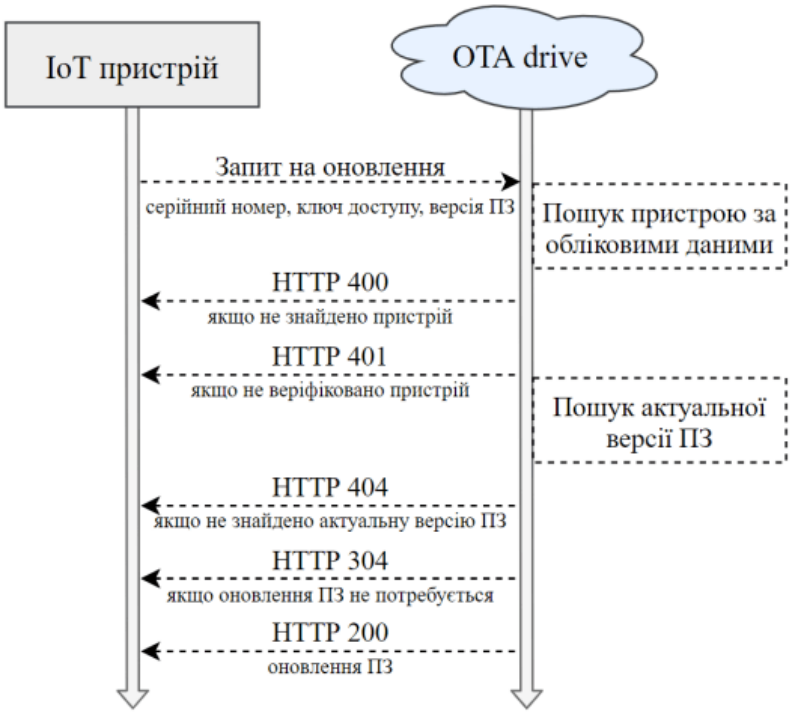


Рисунок 2.18 – Алгоритм роботи сервісу OtaDrive при оновленні програмного забезпечення

Оскільки АПК базується на ESP модулі і не потребує додаткового функціоналу доцільним буде використання OtaDrive (рис. 2.19) в якості сервісу для дистанційного оновлення програмного забезпечення.

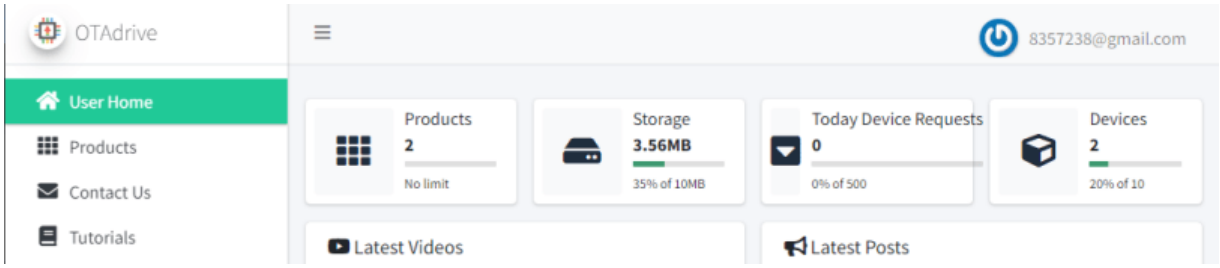


Рисунок 2.19 – Інтерфейс OtaDrive

2.4 Висновки до розділу 2

Згідно до мети роботи, виконано аналіз вимог до проєкту, розроблено архітектуру АПК, проведено аналіз та за його результатами обрано апаратні і програмні засоби для реалізації АПК.

Проаналізувавши існуючі рішення реалізації IoT систем взаємодію АПК, беручи до уваги функціональні можливості, а також розповсюдженість і популярність на ринку, в якості мережевого протоколу обрано Wi-Fi. Засновуючись на необхідності передачі даних великого об’єму (за мірою IoT пристроїв) як протокол обміну даними обрано HTTP.

Як розрахунковий центр обрано модуль з інтегрованим мікроконтролером ESP-32. Для захоплення звуку буде застосовано мікрофон MEMS типу, а саме INMP441. З метою забезпечення відтворення звуку АПК буде містити модуль відтворення та підсилення MAX98357A.

В якості мови програмування мікроконтролеру обрано C++ та бібліотеки для ESP платформ через те, що вони здатні забезпечити високий рівень швидкодії та обробки інформації. Середовищем розробки обрано PlatformIO IDE. Для програмування офлайн розпізнавальної моделі обрано мову Python та платформу Tensor-Flow, середовище написання Jupyter. Враховуючи потребу в спрямованості на розпізнавання спектрограм обрано згорткову нейронну мережу за двома слоями.

Для онлайн розпізнавання обрано сервіс Wit.ai. З метою забезпечення можливості дистанційного оновлення ПЗ хмарний сервіс OtaDrive.

3 КОНСТРУЮВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Алгоритм функціонування

Алгоритм роботи АПК представлено у вигляді UML діаграми (рис. 3.1).

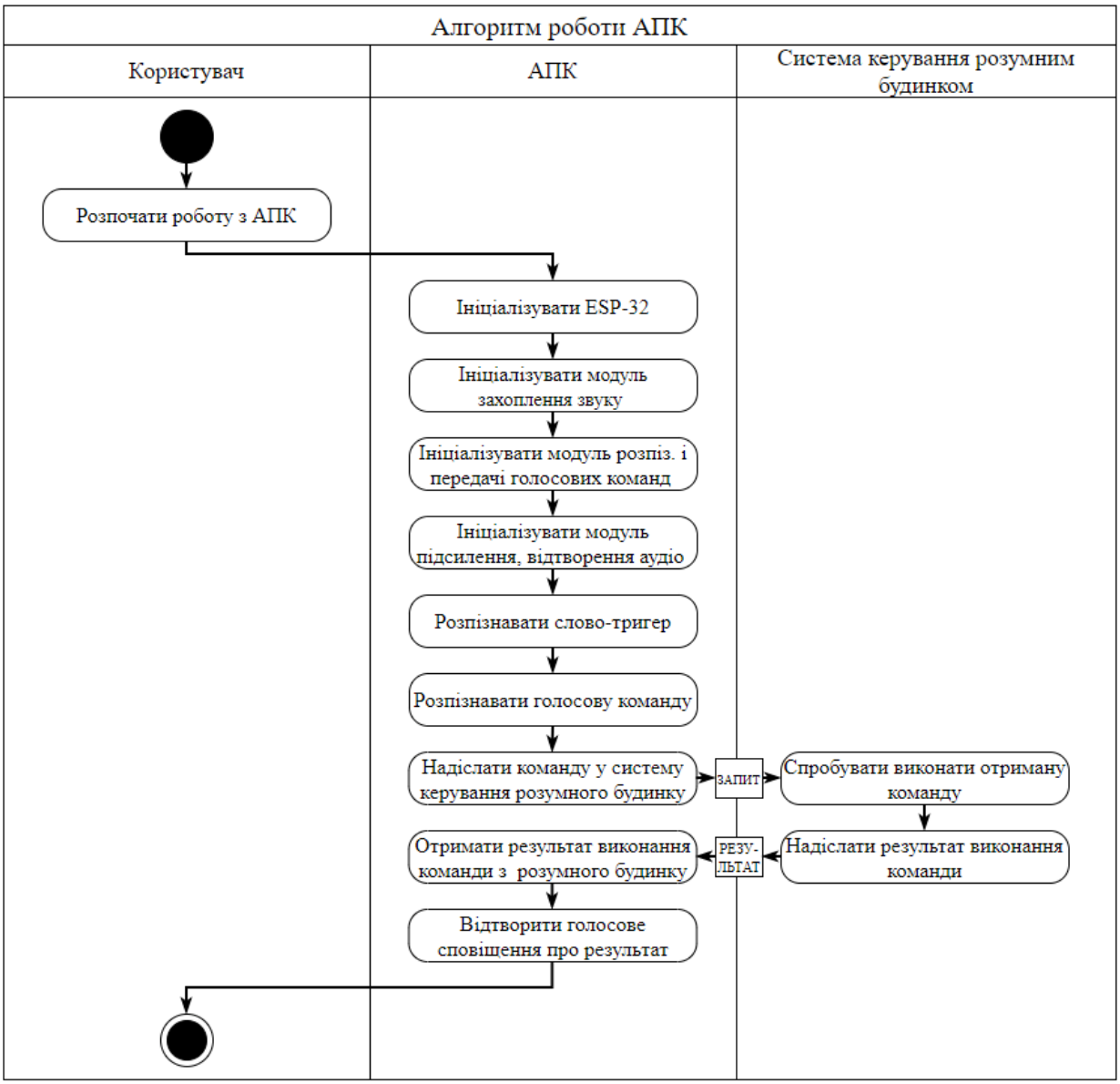


Рисунок 3.1 – Алгоритм роботи АПК

Робота АПК починається з ініціалізації модулів налаштуваннями за замовчуванням, встановленню зв’язку з розумним будинком і з мережею Інтернет. Якщо за якихось обставин не вдалося встановити зв’язок з мережею

Інтернет – передбачено циклічний алгоритм встановлення з’єднання. Між циклами передбачено затримку з метою уникання ситуації перенавантаження мережі.

Після ініціалізації АПК відслідковує звуки навколишнього середовища в очікуванні слова-тригера – збудника (рис. 3.2). Розпізнавання збудника завжди відбувається без задіяння мережі Інтернет з метою зменшення навантаження на сервер.

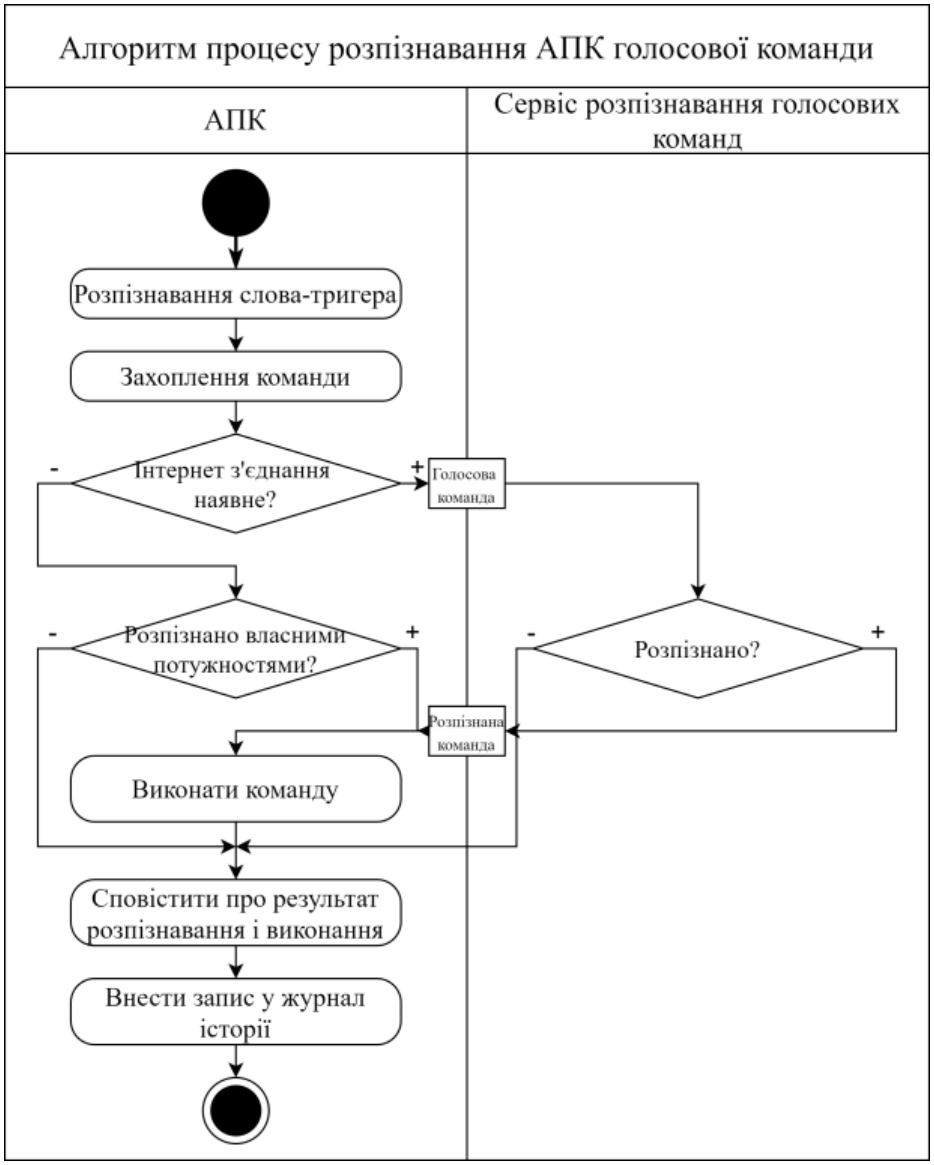


Рисунок 3.2 – Алгоритм розпізнавання голосової команди

При виявленні слова тригера відбувається фіксація всіх наступних слів для збереження цілісності команди. Зафіксувавши команду АПК намагається

через HTTP протокол зв’язатись з онлайн сервісом розпізнавання команд. У випадку успішного з’єднання АПК отримує у відповідь текстову стенограму команди. При відсутності доступу до мережі Інтернет або відсутності з’єднання з хмарним сервісом АПК робить спробу розпізнати команду власними розрахунковими потужностями за раніше завантаженою моделлю розпізнавання. У випадку якщо команду не було розпізнано користувачу надсилається відповідне повідомлення через гучномовець з вказанням режиму, через який відбувалось розпізнавання (онлайн або офлайн).

У випадку успішного розпізнавання команди вона у текстовому вигляді надсилається за допомогою HTTP протоколу в систему розумний дім, або, за наявності відповідних налаштувань, виконується модулем розпізнавання самотужки, після чого в обох випадках повідомляє користувачу прийнята чи відкинута була команда. Кожна дія від користувача записується у журнал команд, який зберігає данні в енергонезалежній пам’яті.

З урахуванням можливості подальшої модернізації АПК було запрограмовано алгоритм віддаленого оновлення програмного забезпечення за допомогою хмарного сховища (рис. 3.3). При ініціалізації АПК ESP надсилає запит на віддалений сервіс оновлення щодо наявності нової прошивки. Запит має чітко визначену структуру та складається з: номера поточної версії програмного забезпечення, коди доступу до сервісу та до акаунту пристрою, серійний номер пристрою. Віддалений сервіс перевіряє наявність відповідного акаунту. Якщо за вказаними кодами доступу існує обліковий запис – відбувається перевірка актуальності поточного програмного забезпечення. При наявності новішої версії – автоматично відбувається завантаження прошивки на пристрій, її встановлення та перезавантаження АПК.

У випадку якщо коди доступу помилкові – обліковий запис не було знайдено – віддалена перевірка актуальності програмного забезпечення АПК неможлива.

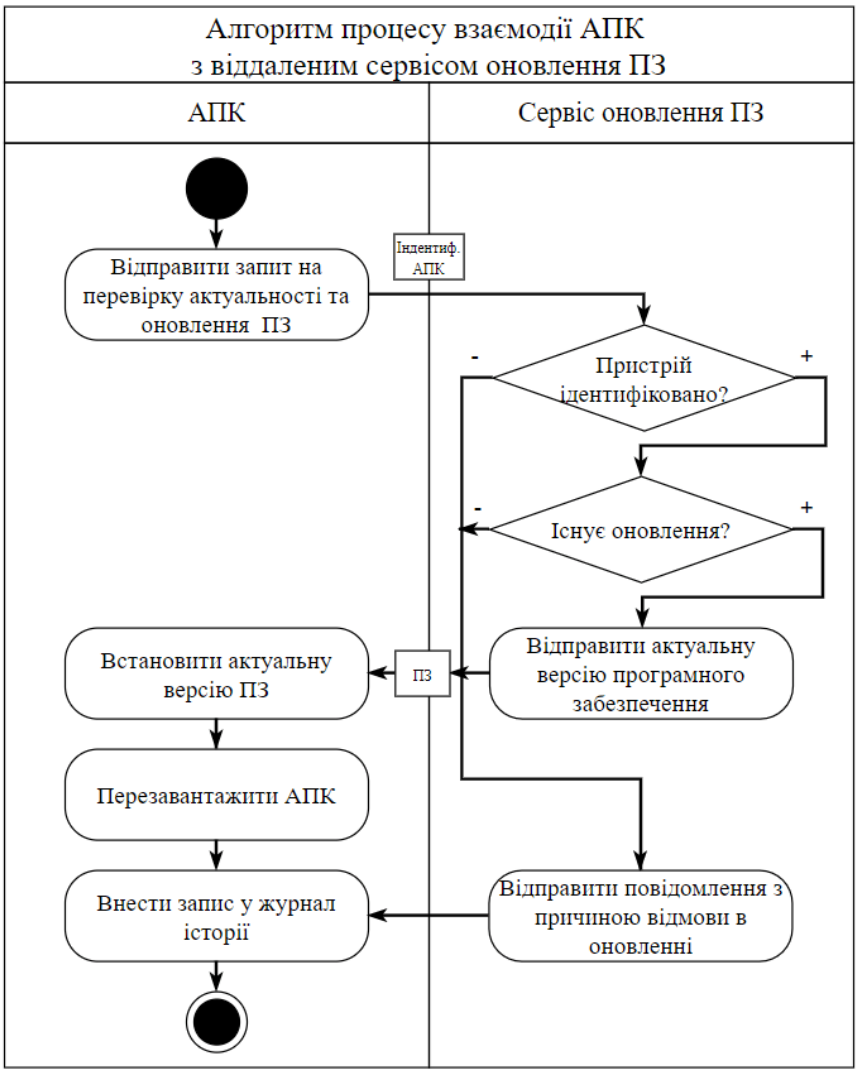


Рисунок 3.3 – Алгоритм віддаленого оновлення ПЗ АПК

3.2 Особливості програмної реалізації

При розробці програмного забезпечення АПК враховано рекомендації з документації до платформи та обраної мови програмування. А саме: програмне забезпечення структуровано за класами, які описані 2 файлами: безпосередньо файлом коду (має .cpp тип розширення) та заголовочного файлу (має .h тип розширення). Заголовочний файл містить перелік усіх змінних та функцій які мають публічний рівень доступу, а файл коду відповідає за безпосередню реалізацію логіки, так мовити «основний код». Для кожного класу в заголовочному файлі передбачено конструкцію

запобігання подвійного підключення у проєкт з меню уникнення можливих конфліктів.

Програмне забезпечення апаратно-програмного комплексу складається з декількох класів, окрім цього до проєкту входять бібліотеки що були задіяні, навчена модель розпізнавання, голосові команди у .waw форматі та інше (рис. 3.4, рис. 3.5).

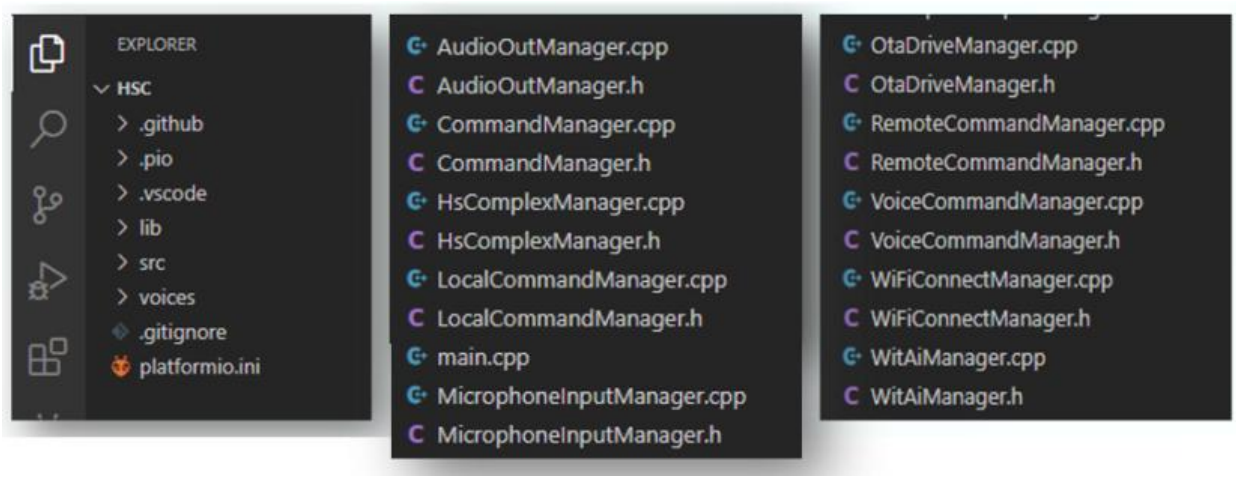


Рисунок 3.4 – Структура програмного забезпечення АПК

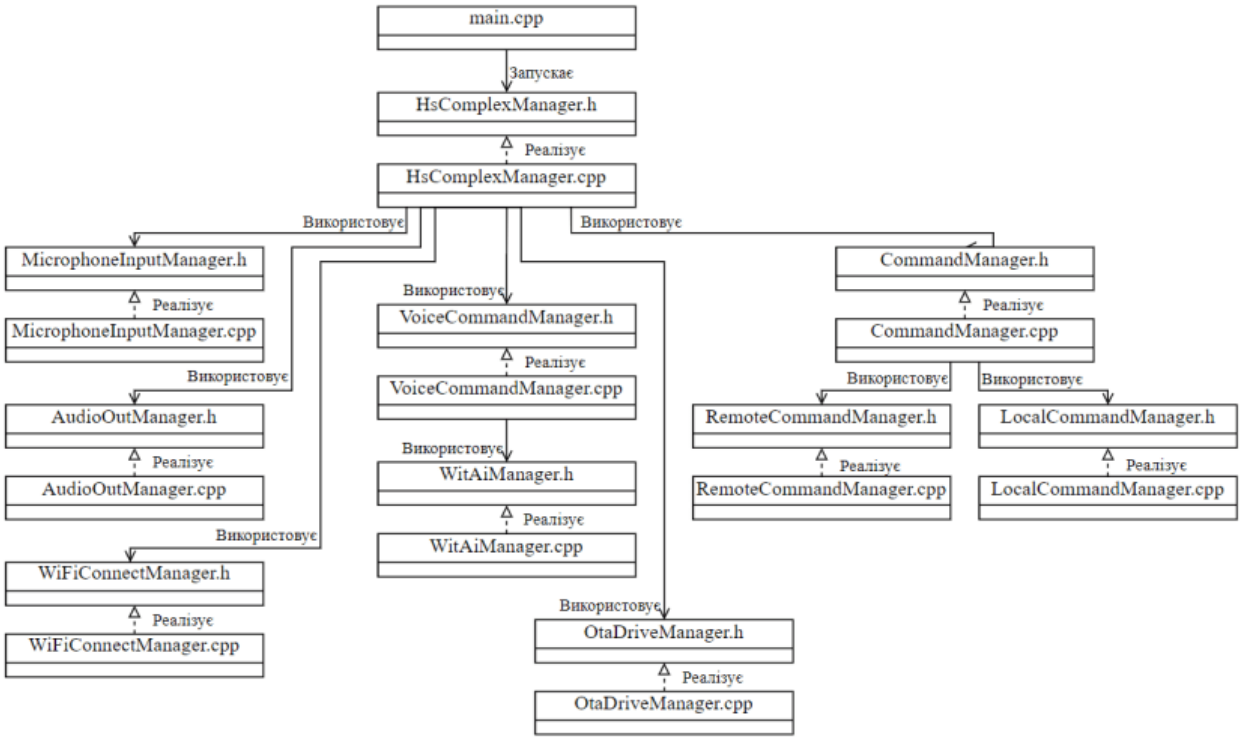


Рисунок 3.5 – Діаграма класів АПК

Клас WiFiConnectManager розроблено для забезпечення функціоналу з’єднання через Wi-Fi з інтернет мережею за допомогою вбудованого в ESP Wi-Fi адаптеру. Серед найбільш значимих функцій можна виділити InitConnect, яка для надає функціонал безпосередньо для з’єднання з мережею. У випадку якщо з’єднання з мережею неможливо встановити передбачено циклічний алгоритм з повтору спроби налаштувати з’єднання через заздалегідь визначену затримку. З метою поліпшення процесу розробки було використано бібліотеку «WiFiClient» від розробника Arduino IDE, яка забезпечує інтерфейс роботи з адаптером Wi-Fi ESP-32.

Клас WitAiManager розроблено для взаємодії з Wit.ai сервісом. Відповідно алгоритми класу містять функціонал для передачі голосового повідомлення і отримання результатів розпізнавання голосової команди. До класу було іпортовано бібліотеку «ArduinoJson» для обробки відповіді з сервісу оскільки відповідь надається в json форматі.

Взаємодію з моделлю розпізнавання голосових команд забезпечує клас VoiceCommandManager. Взаємодія з моделлю відбувається за допомогою бібліотеки «TfMicro», яка адаптує використання TensorFlow на мікроконтролерах, так звана TensorFlow Lite. Клас розраховано на використання з навченою моделлю, яку заздалегідь конвертовано у мову С.

Клас OtaDriveManager розроблено для забезпечення функціоналу перевірки і оновлення програмного забезпечення (у разі необхідності). Серед функціоналу можна виділити функції: GetDeviceID – яка повертає персональний номер ESP модулю, CheckAvailibleUpdate – функція перевірки на сервері актуальності програмного забезпечення, TryUpdate – функція для завантаження та встановлення програмного забезпечення. Функціонал оновлення передбачає захисний алгоритм який у разі виявлення помилок в процесі встановлення нового програмного забезпечення намагатиметься відновити останню робочу версію ПЗ. Для завантаження з ОТА сервісу двійкового файлу і процесу встановлення оновлення було задіяно бібліотеку від ESP платформи HttpUpdate.

Клас MicrophoneInputManager забезпечує функціонал налаштування та взаємодії ESP-32 з модулем мікрофону. Для налаштування коректної роботи мікрофону через I2C інтерфейс у класі інтегровано бібліотеку «Driver/I2s.h». Клас AudioOutManager створено для забезпечення функціоналу з налаштування взаємодії ESP-32 з гучномовцем і відтворенню .wav файлів.

Клас CommandManager здійснює загальне керування виконанням команд. LocalCommandManager забезпечує виконання голосових команд власними потужностями комплексу. Клас RemoteCommandManager забезпечує взаємодію з розумним будинком через HTTP протокол. Клас HsComplexManager містить логіку роботи комплексу. Клас main зумовлений особливостями ESP архітектури та реалізує інструкції для загальної роботи комплексу.

Файл налаштувань .config має загальні змінні параметри проєкту які забезпечують функціонування комплексу (табл. 3.1). Файл commads.cpp має перелік доступних команд.

Таблиця 3.1 – Перелік основних параметрів для налаштування ESP-32

Параметр	Опис
WiFiName	Ім’я Wi-Fi мережі
WiFiPassword	Пароль до Wi-Fi мережі
WitAiApiUrl	Адреса до сервісу Wit.at
WitAiApiKey	Ключ авторизації для Wit.ai сервісу
OtaDriveUrl	Адреса до сервісу OtaDrive
OtaDriveName	Ім’я для з’єднання до OtaDrive
OtaDriveKey	Ключ авторизації для OtaDrive сервісу
SmartHomeApiUrl	Адреса розумного будинку

Програмне рішення для навчання нейромережі міститься окремо з метою не перенавантажувати розміри програмного рішення АПК. Для побудови розпізнавальної моделі використовуються бібліотеки tensorflow, numpy, datetime та інші.

Програмне рішення складається з декількох файлів: GenTrainingDataManager, TrainDataManager, ConvertTrainModelManager.

GenTrainingDataManager – відповідає за генерацію даних для навчання, а саме для генерації спектрограм з аудіофайлів. Враховуючи достатню кількість даних з метою підвищення стресостікйості системи слова були розділені на дві групи «потрібні», та «ті, що не мають значення». Це потрібно для підвищення здатності системи розпізнавати саме «потрібні слова». З метою підвищення працездатності системи передбачено функціонал для генерації з існуючих аудіофайлів нові, з побутовим шумом на фоні.

TrainDataManager – файл, який відповідає за безпосередньо побудову моделі розпізнавання за допомогою зготкової нейронної мережі двох слоїв. Це забезпечує функція Conv2D з бібліотеки Tensorflow.Keras.

ConvertTrainModelManager – файл, який забезпечує конвертацію побудованої моделі для задіяння у TensorFlow Lite на ESP-32. На виході буде сконвертована у мову С модель.

Аналізуючи розроблене програмне рішення можна стверджувати що АПК має гарну структурованість, високу модульність. Це полегшує подальшу модернізацію апаратно-програмного комплексу, дозволяє, за необхідності, повторно використовувати розроблені програмні рішення у інших проєктах.

3.3 Тестування та експериментальне дослідження працездатності АПК

З метою перевірки розробленого АПК на працездатність було розроблено методику для проведення розробленого АПК, розроблено методику для проходження функціонального тестування.

Функціональне тестування – це один з видів тестування, процес забезпечення якості який є різновидом «чорної скриньки», що засновано на

перевірки функціоналу через зіставлення вхідних і вихідних даних. Воно проводиться для надання оцінки відповідності комплексу попередньо висунутим функціональним вимогам. Успішне проходження функціонального тестування надає можливість стверджувати що розроблений продукт реалізує заявлений функціонал.

Беручи до уваги вимоги що були визначені до АПК було розроблено тест-кейси та контрольний список для проведення функціонального тестування. Після цього згідно до складеного списку проведено безпосередньо саме тестування. Результати перевірки працездатності АПК за умови наявності та відсутності доступу до Інтернет-мережі наведено у таблицях 3.2 – 3.3.

Таблиця 3.2 – Тест-кейс перевірки працездатності розпізнавання за умови наявності доступу АПК до Інтернет-мережі

Передумови			
Увімкнути Wi-Fi роутер, через який налаштовано з’єднання і який забезпечує доступ до мережі Інтернет. Увімкнути живлення АПК			
№	Дія	Очікуваний результат	Статус
1	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
2	Промовити одну з визначених команд	Програно аудіосигнал, що інформує про успішне розпізнання команди. Програно аудіосигнал, що інформує про результат виконання команди	Пройдено
3	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
4	Промовити будь-яку невизначену команду	Програно аудіосигнал, що інформує про неможливість виконання команди	Пройдено
5	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
6	Не промовляти нічого	Через 10 секунд програно аудіосигнал, що інформує про відсутність команди	Пройдено

Таблиця 3.3 – Тест-кейс перевірки працездатності розпізнавання за умови відсутності доступу АПК до Інтернет-мережі

Передумови			
Увімкнути живлення АПК			
№	Дія	Очікуваний результат	Статус
1	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
2	Промовити одну з визначених команд	Програно аудіосигнал, що інформує про успішне розпізнання команди. Програно аудіосигнал, що інформує про результат виконання команди	Пройдено
3	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
4	Промовити будь-яку невизначену команду	Програно аудіосигнал, що інформує про неможливість виконання команди локально	Пройдено
5	Промовити слово-тригер	Програно аудіосигнал, що підтверджує початок сприйняття команди	Пройдено
6	Не промовляти нічого	Через 10 секунд програно аудіосигнал, що інформує про відсутність команди	Пройдено

Контрольний список функціонального тестування АПК наведено в таблиці 3.4.

Таблиця 3.4 – Контрольний список функціонального тестування АПК

Вимога	Статус
1	2
Активація за словом-тригером	Пройдено
Розпізнавання команди без з’єднання з мережею Інтернет	Пройдено
Розпізнавання команди зі з’єднанням з мережею Інтернет	Пройдено
Аудіоінформування щодо сприйняття команди	Пройдено

Продовження таблиці 3.4

1	2
Передача розпізнаної команди розумному будинку	Пройдено
Виконання розпізнаної команди власними потужностями	Пройдено
Аудіоінформування щодо виконання команди	Пройдено
Дистанційне оновлення програмного забезпечення	Пройдено
Ведення журналу виконаних команд	Пройдено
Збереження налаштувань при перезавантаженні	Пройдено

Результати проведеного тестування свідчать про стабільність роботи апаратно-програмного комплексу, повній відповідності висунутим функціональним вимогам.

3.4 Висновки до розділу 3

За результатами виконаної роботи розглянуто особливості й розроблено алгоритм роботи АПК. Беручи до уваги апаратні обмеження та особливості компонентів, розроблено схему взаємодії складових частин АПК.

Описано структурні частини програмного забезпечення, основні розроблені функції, змінні. Створена архітектура характеризується структурованістю та модульністю. Це сприяє поліпшенню та пришвидшенню модернізації програмного забезпечення (за необхідністю), дозволяє використовувати розроблені програмні рішення повторно у суміжних проєктах.

Для реалізації перевірки працездатності АПК та з метою переконання у повній працездатності АПК було розроблено тест-кейсі і контрольний список функціонального тестування АПК, за якими проведене відповідне випробування. Успішне виконання усіх пунктів проведених тестувань свідчить про стабільну роботу АПК.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови виконання програми

Призначення розробленого програмного продукту полягає у забезпеченні роботи апаратно-програмного комплексу з розпізнавання голосових команд. Він складається з двох програм:

– програма для забезпечення роботи безпосередньо самого апаратно-програмного комплексу на модулі ESP-32;

– програма для навчання розпізнавальної моделі з метою її подальшого застосування в апаратно-програмному комплексі.

Програмне забезпечення апаратно-програмного комплексу розроблено на мові C++ з застосуванням бібліотек платформи PlatformIO.

Програмне забезпечення навчання розпізнавальної моделі розроблено на мові Python з застосуванням бібліотеки TensorFlow.

Для експлуатації апаратно-програмного комплексу необхідні такі програмно-технічні засоби:

– джерело живлення 5В або 3.3В;

– Wi-Fi роутер який має з’єднання з мережею Інтернет (не обов’язково, але бажано).

Для налаштувань апаратно-програмного комплексу необхідні такі програмно-технічні засоби:

– персональний комп’ютер з встановленим PlatformIO IDE та мовою Python (версії 3.0 або вище);

– монітор;

– оперативна пам’ять 1 ГБ або більше;

– клавіатура;

– маніпулятор типу миша.

4.2 Характеристики програми

Розроблене програмне забезпечення дозволяє АПК виконувати такі функції:

– розпізнавання голосових команд за допомогою віддаленого сервіс;

– розпізнавання голосових команд за допомогою власних потужностей без застосування віддалених сервісов;

– надсилання розпізнаних команд до віддаленого серверу розумного будинку;

– виконання розпізнаних команд власними потужностями;

– встановлення налаштувань за замовчуванням;

– ведення журналу розпізнаних команд;

– дистанційне оновлення програмного забезпечення АПК за допомогою хмарного сервісу.

4.3 Звернення до програми

В програмному забезпеченні апаратно-програмного комплексу можна встановити налаштування для з’єднання з віддаленими сервісами, а саме:

– WiFiName – ім’я Wi-Fi мережі;

– WiFiPassword – пароль до Wi-Fi мережі;

– WitAiApiUrl – адреса до сервісу Wit.at;

– WitAiApiKey – ключ авторизації для Wit.ai сервісу;

– OtaDriveUrl– адреса до сервісу OtaDrive;

– OtaDriveName – ім’я для з’єднання до OtaDrive;

– OtaDriveKey – ключ авторизації для OtaDrive сервісу;

– SmartHomeApiUrl – адреса розумного будинку.

Переглядання і корегування списку доступних для розпізнавання віддаленим сервісом команд доступно у особистому кабінеті (рис. 4.1) за посиланням https://wit.ai/apps.

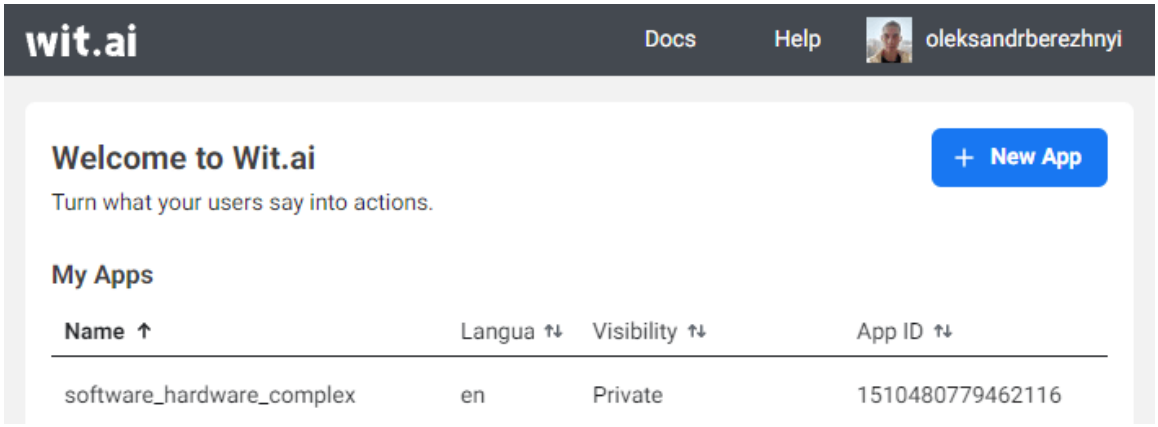


Рисунок 4.1 – Особистий кабінет розробника у сервісі Wit.ai

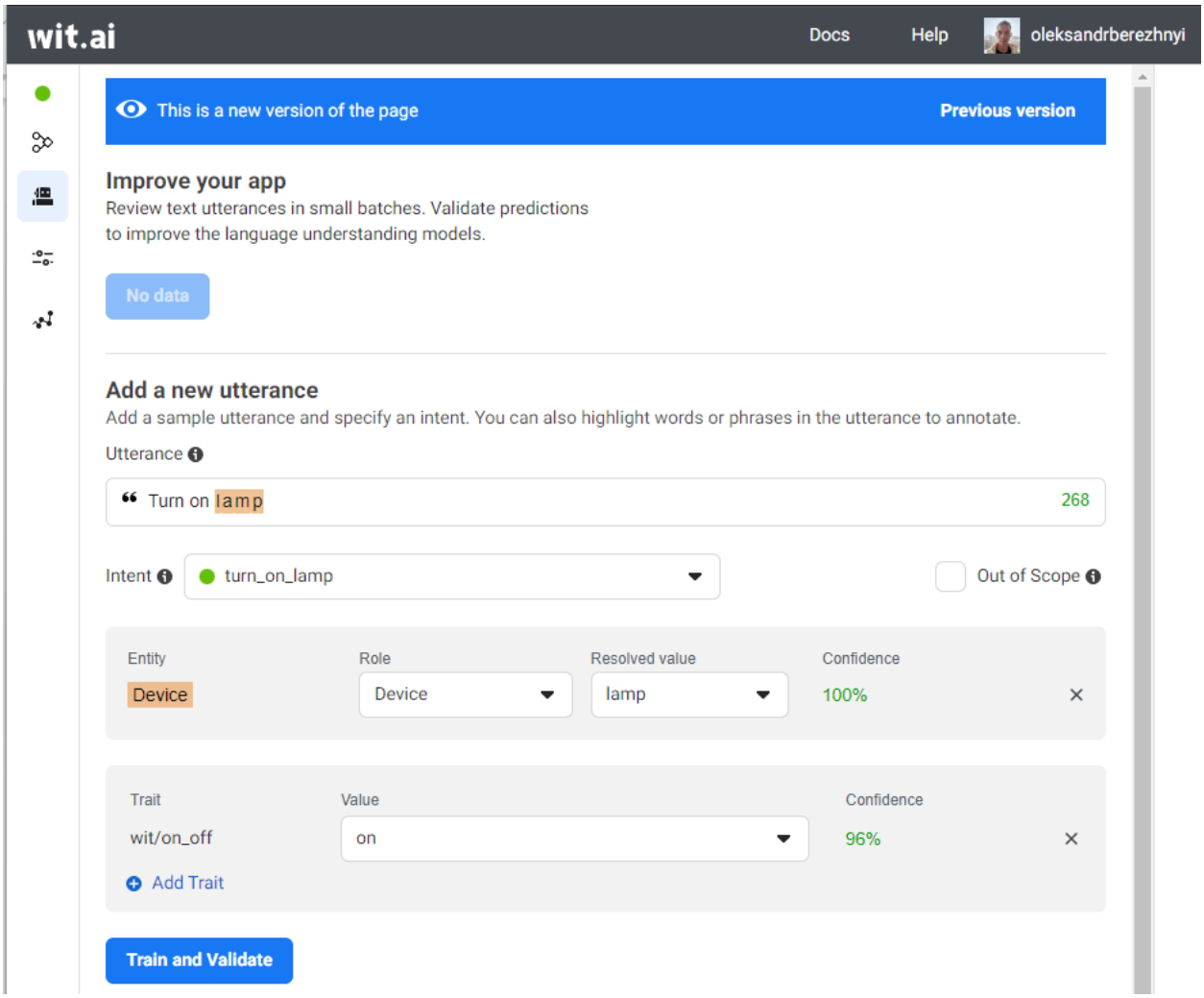


Рисунок 4.2 – Налаштування команди на розпізнавання у сервісі Wit.ai

Завдяки гнучкості Wit.ai можна змінити певні налаштування чи виправити помилки як в команді що розпізнається, так і в відповіді, що надсилається як результат розпізнавання (рис. 4.2). Зверніть увагу, після внесення корегувань необхідно натиснути «Train and validate» для адаптації системи розпізнавання, а, також, за необхідності внести зміни до програмного забезпечення АПК.

В апаратно-програмному комплексі реалізовано можливість для дистанційного оновлення програмного забезпечення через хмарний сервіс. Що б оновити програмне забезпечення необхідно авторизуватись у хмарному сервісі OtaDrive, обрати потрібну групу апаратно-програмних пристроїв і завантажити у неї актуальне програмне забезпечення у .bin форматі (рис. 4.3). Зверніть увагу що з метою уникнення помилок перед завантаженням на сайт бажано перевірити програмне забезпечення самотужки на тестовому пристрої.

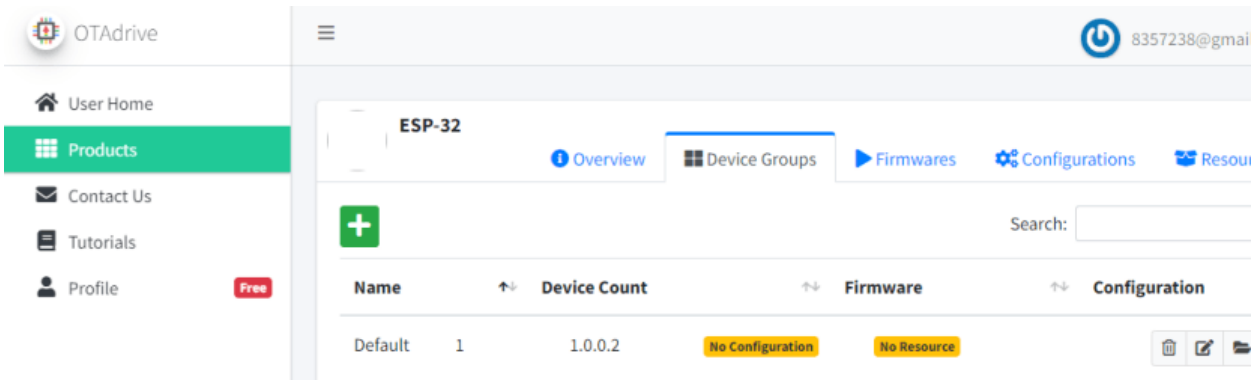


Рисунок 4.3 – Завантаження нової версії програмного забезпечення до хмарного сервісу OtaDrive

Зверніть увагу, що у випадку коли сервіс не зможе розпізнати в автоматичному режимі метадані нової версії програмного забезпечення Вам слід вказати їх у мануальному режимі (рис. 4.4).

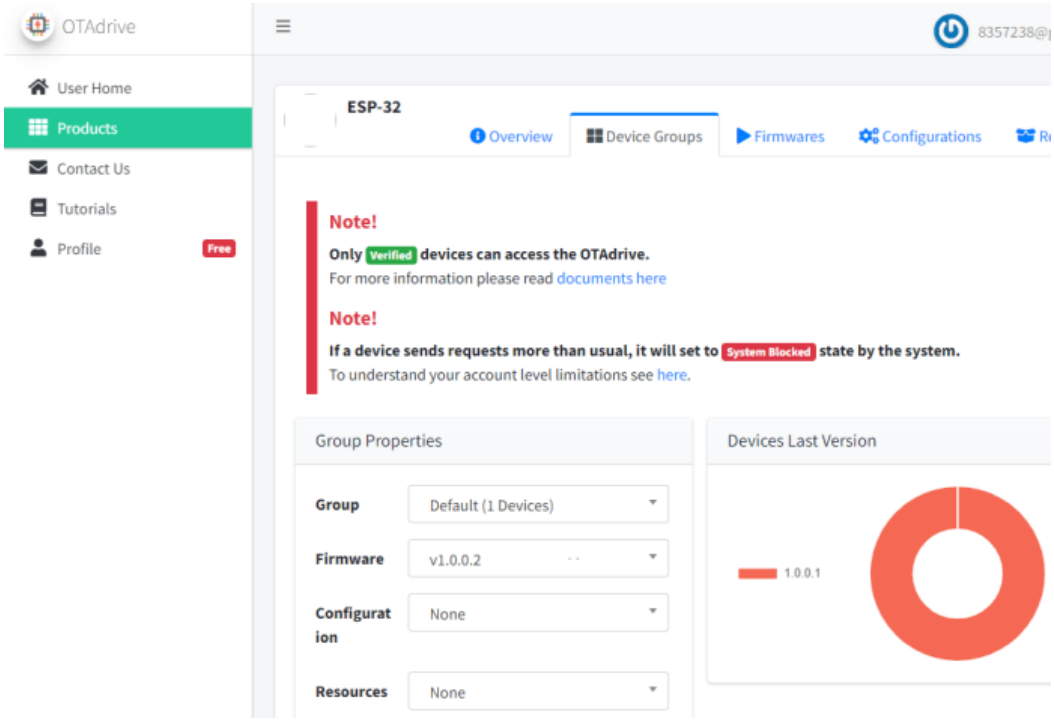


Рисунок 4.4 – Налаштування метаданих нової версії програмного забезпечення у сервісі OtaDrive

За умови успішного завантаження нової версії програмного забезпечення на хмарний сервіс (рис. 4.5) вона буде автоматично запропонована усім пристроям групи, які будуть під’єднуватись до хмарного сховища OtaDrive.

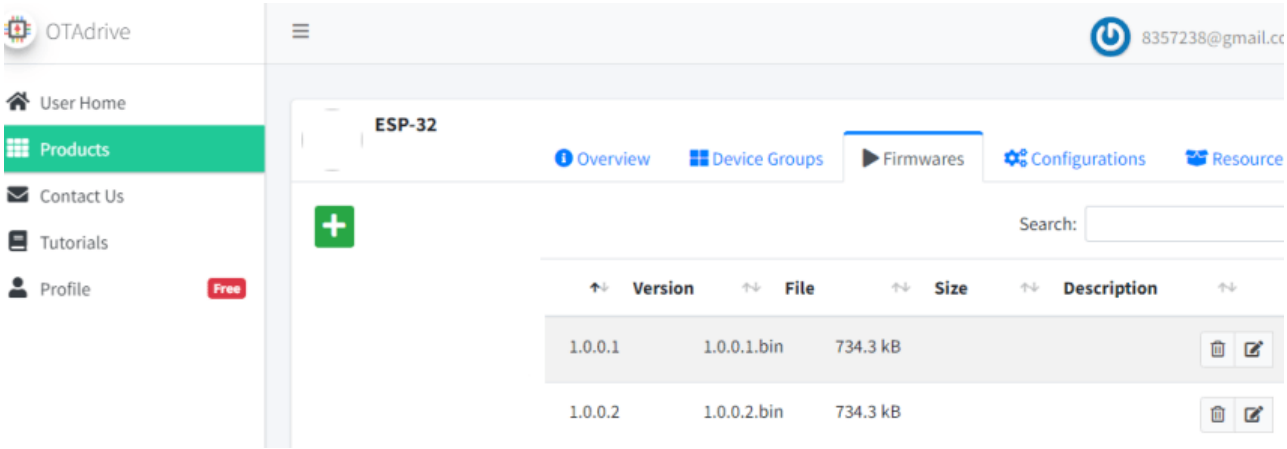


Рисунок 4.5 – Збереження декількох версій програмного продукту на хмарному сервісі OtaDrive

4.4 Вхідні та вихідні дані

Вхідними даними програми апаратно-програмного комплесу є:

– налаштування Wi-Fi мережі;

– налаштування доступу до віддалених сервісів.

Вихідними даними є:

– розпізнані голосові команди, які надаються відоповідно в залежності від результатів розпізнавання;

– журнал історії;

– аудіоівідгук (інформування) кінцевого користувача щодо перебігу розпізнавання і виконання команд.

4.5 Повідомлення і обробка виняткових ситуацій

Розроблений апаратно-програмний комплекс забезпечує стабільну роботу і функціонування. Виняткові ситуації можливі при неправильному налаштуванні компонентів. У випадку виникнення помилок передбачено відправлення повідомлень про тип помилки через послідовний порт.

У таблиці 4.1 перелічено повідомлення програмісту, які можуть бути надіслані апаратно програмним комплексом.

Таблиця 4.1 – Повідомлення програмісту від блоків АПК

Повідомлення	Опис	Дії програміста
1	2	3
Error. Microphone initialize failed.	Модуль мікрофону не ініціалізовано	Перевірити правильність параметрів налаштув. мікрофону
Error. Audio out initialize failed.	Модуль гучномовця не ініціалізовано	Перевірити правильність параметрів налаштування гучномовця

Продовження таблиці 4.1

1	2	3
Warning. Wi-Fi credentials are incorrect	При з’єднанні з Wi-Fi мережею виникла помилка некоректності даних авторизації	Перевірити правильність параметрів налаштувань Wi-Fi з’єднання
Warning. OtaDrive credentials are incorrect	При з’єднанні з хмарним сервісом OtaDrive виникла помилка некоректності даних авторизації	Перевірити правильність параметрів налаштувань OtaDrive з’єднання
Warning. Wit.ai credentials are incorrect	При з’єднанні з хмарним сервісом Wit.ai виникла помилка некоректності даних авторизації	Перевірити правильність параметрів налаштувань Wit.ai з’єднання

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Призначення програми

Програмний продукт призначено для організації керування розумними пристроями за допомогою голосових команд, а саме для:

– забезпечення можливості розпізнавання голосових команд за допомогою віддаленого сервісу;

– забезпечення можливості розпізнавання голосових команд за допомогою власних потужностей без застосування віддалених сервісів;

– надсилання розпізнаних команд до віддаленого серверу розумного будинку;

– виконання розпізнаних команд власними потужностями;

– ведення журналу розпізнаних команд.

5.2 Умови виконання програми

Для експлуатації апаратно-програмного комплексу необхідні такі програмно-технічні засоби:

– джерело живлення 5В або 3.3В;

– Wi-Fi роутер який має з’єднання з мережею Інтернет (не обов’язково, але бажано).

Живлення 5В здатне забезпечити більш стабільну роботу блоків та підвищити допустиму максимальну дальність від Wi-Fi роутера.

5.3 Повідомлення оператору

Розроблений апаратно-програмний комплекс забезпечує відправлення повідомлень через послідовний порт про поточний стан роботи а також надання аудіовідгуку щодо стану роботи.

У таблиці 5.1 зазначено повідомлення оператору, які може надсилати апаратно-програмний комплекс оператору через послідовний порт. У таблиці 5.2 зазначено опис аудіосигналів, які може відтворювати апаратно-програмний комплекс оператору.

Таблиця 5.1 – Повідомлення оператору від апаратно-програмного комплексу через послідовний порт

Повідомлення	Опис
Wi-Fi. Successfully configured	Wi-Fi адаптер було успішно сконфігуровано
Wi-Fi. Successfully connected	АПК успішно під’єднався до Wi-Fi мережі
Software. Update checked	Перевірено на наявність оновлення програмного забезпечення
Software. Update installed	Встановлено нове програмне забезпечення
Trigger word. Was recognized	Тригерне слово розпізнано
Command. Was recognized (remote)	Команда була розпізнана через віддалений сервіс
Command. Was not recognized (remote)	Команда не була розпізнана через віддалений сервіс
Command. Was recognized (esp)	Команда була розпізнана власними потужностями
Command. Was not recognized (esp)	Команда не була розпізнана власними потужностями
Command. Was executed successfully (remote)	Команда була виконана успішно через взаємодію з розумним будинком
Command. Was failed (remote)	Команда не була успішно виконана розумним будинком
Command. Was executed successfully (esp)	Команда була виконана успішно власними потужностями
Command. Was failed (esp)	Команда не була успішно виконана власними потужностями

Таблиця 5.2 – Повідомлення оператору від апаратно-програмного комплексу через гучномовець

Повідомлення	Опис
«Короткий звук на початку роботи»	АПК готовий до роботи
«Короткий звук»	Тригерне слово розпізнано
Recognized	Команда була розпізнана через віддалений сервіс або власними потужностями
Not recognized	Команда не була розпізнана через віддалений сервіс або власними потужностями
Successfully	Команда була виконана успішно через взаємодію з розумним будинком або власними потужностями
Failed	Команда не була успішно виконана розумним будинком або власними потужностями

5.4 Виконання програми

Для виконання програми необхідно ввімкнути блок живлення і під’єднати його до апаратно-програмного комплексу. АПК розпочниає роботу в автоматичному режимі (рис. 5.1). На початку роботи через гучномовець буде подано короткий звуковий сигнал, що свідчить про успішну ініціалізацію усіх складових.

АПК очікує слово-тригер. При промові користувачем слова-тригера буде через гучномовець подано короткий звуковий сигнал, який свідчить про готовність сприйняття голосової команди. Користувач повинен промовити команду.

У випадку якщо команда не була промовлена через 10 секунд через гучномовець буде передано повідомлення «Not recognized».

Якщо команда була промовлена АПК спробує розпізнати команду. У випадку проблем з розпізнаванням команди через гучномовець буде передано повідомлення «Not recognized». У випадку успішного розпізнавання команди через гучномовець буде передано «Recognized». За результатами виконання команди через гучномовець буде передано або «Successfully» (у випадку успішного виконання), або «Failed» (у випадку неможливості виконати команду).

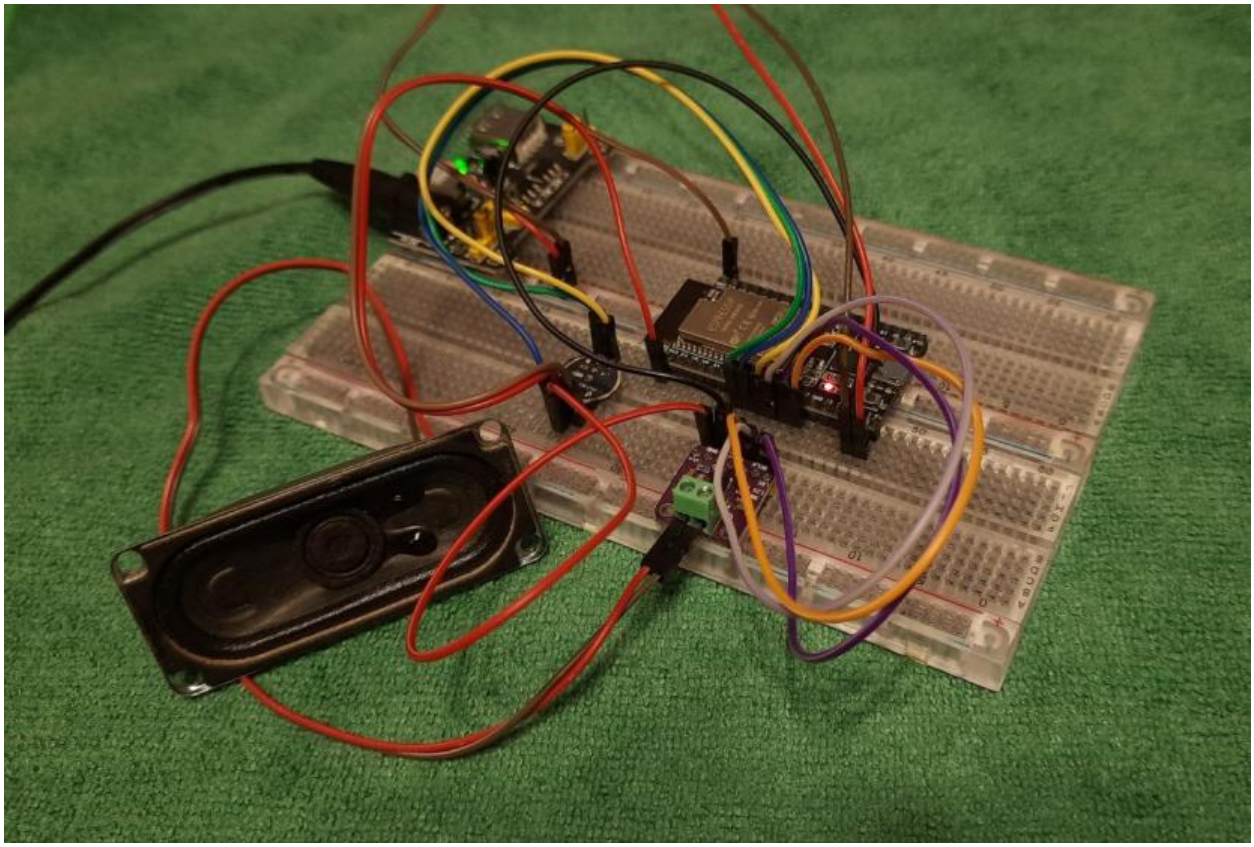


Рисунок 5.1 – Приклад роботи апаратно-програмного комплексу

ВИСНОВКИ

За результатами виконання дипломної кваліфікаційної роботи магістра вирішено актуальну задачу забезпечення можливості керування розумними пристроями через голосові команди як в онлайн так і в офлайн режимі.

Розроблений АПК має наступний функціонал: розпізнавання голосових команд; виконання розпізнаних команд; надання відгуку через гучномовець.

Наукова новизна роботи полягає у тому, що удосконалено принцип роботи голосового керування IoT системами за рахунок використання нейромережі, що на відміну від існуючих підходів дозволяє забезпечити стабільне розпізнавання голосових команд користувачів навіть у разі відсутності Інтернет з’єднання.

Практична цінність результатів роботи полягає у тому, розроблене апаратно-програмне рішення може бути використано для керування IoT системами типу Розумний будинок, що дозволяє покращити процес взаємодії з IoT пристроями людей з особливими потребами, оскільки використання голосового асистенту знизить навантаження на зорову систему користувача.

В подальшому заплановано розширення кількості доступних команд, підвищення зручності використання комплексу через розробку вебінтерфейсу для забезпечення можливості керування налаштуваннями за допомогою веббраузеру. Окрім цього заплановано покращення алгоритмів офлайн розпізнавання, а саме – забезпечення стійкості системи до зашумленого середовища, що розширить потенційну галузь застосування.

Основні положення та результати роботи були представлені в тезах доповіді НПК Тиждень науки – 2022 та на XI міжнародній науково-практичній конференції «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. AWS – What is IoT? [Electronic resource]. – Access mode: https://aws.amazon.com/what-is/iot.

2. ItOnlineLearning – History of the Internet of Things (IoT) [Electronic resource]. – Access mode: https://www.itonlinelearning.com/blog-history-iot.

3. Friendly Technologies – A Brief History of IoT [Electronic resource]. – Access mode: https://friendly-tech.com/a-brief-history-of-iot/

4. RF Page – Applications of Internet of Things (IoT) [Electronic resource]. – Access mode: https://www.rfpage.com/applications-of-internet-of-things-iot.

5. Deoras S. 10 reasons why IoT is rising today? [Electronic resource] / S..Deoras // Analytics India Magazine. – 2016. – Access mode: https://analyticsindiamag.com/category/research.

6. Chauhan N. Challenges in Internet of Things (IoT) [Electronic resource] / N. Chauhan // OpenGrowth. – 2022. – Access mode: https://www.opengrowth.com/resources/challenges-in-internet-of-things-iot.

7. Sladden O. The 10 biggest internet of things funding deals so far in 2022 [Electronic resource] / O. Sladden // Verdict. – 2022. – Access mode: https://www.verdict.co.uk/the-10-biggest-internet-of-things-funding-deals-so-far-in-2022.

8. Internet of Things (IoT) Market Size, Share & COVID-19 Impact Analysis [Electronic resource] // Market Research Report. – 2022. – С. 130. – Access mode: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307.

9. IOT – Наші проекти [Електронний ресурс]. – Режим доступу: https://iotukraine.com/category/projects-ua.

10. Rosencrance L. Exploring IoT Connectivity Options [Electronic resource] / L. Rosencrance // IoT World Today. – 2021. – Access mode: https://www.iotworldtoday.com/2021/09/13/exploring-iot-connectivity-options.

11. Libson S. Your Complete Guide to IoT Protocols and Standards in 2022 for Support Secure Data Exchange [Electronic resource] / S. Libson // Kellton. – 2022. – Access mode: https://www.kellton.com/kellton-tech-blog/your-complete-guide-to-iot-protocols-and-Standards-2022.

12. Expanice – A Rundown of IoT Communication Protocols: Use Cases, Advantages, and Limitations [Electronic resource]. – Access mode: https://expanice.com/article/iot-communication-protocols-comparison.

13. Khalid A. Secure IoT Resources with Access Control over RESTful Web Services / A. Khalid, A. Omar. // Jordan Journal of Electrical Engineering. – 2020. – №6. – С. 63–77.

14. Daviteq – What are the most popular IoT protocols? Characteristics and Applications of them [Electronic resource]. – Access mode: https://www.daviteq.com/blog/en/what-are-the-most-popular-iot-protocols.

15. Mone L. IoT Devices, Sensors, and Actuators Explained [Electronic resource] / L.-Mone // LeanIX. – 2018. – Access mode: https://www.leanix.net/en/blog/iot-devices-sensors-and-actuators-explained.

16. TechDifferences – Difference Between Sensors and Actuators [Electronic resource]. – Access mode: https://techdifferences.com/difference-between-sensors-and-actuators.html.

17. Fibocom – A Guide to Find the Right IoT Module for Your Project [Electronic resource]. – Access mode: https://www.fibocom.com/en/Blog/info_itemid_2133.html.

18. Telink – The Power of Using a System on Chip (SoC) Approach for IoT Development [Electronic resource]. – Access mode: https://www.telink-semi.com/system-on-chip.

19. PwC – Consumer Intelligence Series: Prepare for the voice revolution [Electronic resource]. – Access mode: https://www.pwc.com/us/en/advisory-services/publications/consumer-intelligence-series/voice-assistants.pdf.

20. Voice And Speech Recognition Market Worth $53.66 Billion By 2030. // Grand View Research. – 2022. – Access mode:

https://www.grandviewresearch.com/press-release/global-voice-recognition-industry.

21. Gausden G. Alexa, what time of day will my energy be cheapest? [Electronic resource] / G. Gausden // This is Money. – 2019. – Access mode: https://www.thisismoney.co.uk/money/bills/article-6825793/Octopus-Energy-teams-Amazon-Alexa-let-customers-use-voice-assistant-curb-energy-usage.html.

22. Omale G. Gartner Predicts 25 Percent of Digital Workers Will Use Virtual Employee Assistants Daily by 2021 [Electronic resource] / G..Omale // Gartner. – 2019. – Access mode: https://www.gartner.com/en/newsroom/press-releases/2019-01-09-gartner-predicts-25-percent-of-digital-workers-will-u

23. Atom – Find out how Atom plans to use biometrics as part of a multi-tiered security process [Electronic resource]. – Access mode: https://www.atombank.co.uk/blog/2015/11/why-is-atom-using-biometrics.

24. IBM – Speech Recognition [Electronic resource]. – Access mode: https://www.ibm.com/cloud/learn/speech-recognition.

25. Jia Y. Speaker recognition based on characteristic spectrograms and an improved self organizing feature map neural network / [Y. Jia, X. Chen, J.-Yu and others]. // Springer. – 2020. – Access mode: https://d-nb.info/1216479852/34.

26. Memon M. ANN vs CNN vs RNN: Neural Networks Guide [Electronic resource] / Memon M. – Access mode: https://levity.ai/blog/neural-networks-cnn-ann-rnn.

27. CEPro – Navigating the Role of Smart Home and Voice Assistant Platforms in 2021 [Electronic resource]. – Access mode: https://www.cepro.com/control/navigating-role-smart-home-voice-assistant-platforms-in-2021.

28. Cocks S. Amazon Alexa vs. Google Home: Which assistant should you pick in 2022? [Electronic resource] / S. Cocks // Good Housekeeping. – 2022. – Access mode: https://www.goodhousekeeping.com/uk/product-reviews/tech/a39384401/alexa-vs-google-home.

29. Миронович В. Netblocks повідомляє про збої в доступі до Інтернету в Україні [Електронний ресурс] / В. Миронович // Speka. – 2022. – Режим доступу: https://speka.media/netblocks-povidomlyaje-pro-zboyi-v-dostupi-do-internetu-v-ukrayini-p65lnv.

30. Carlton-Collins J. Amazon Echo vs. Google Home: Which is better? [Electronic resource] / J. Carlton-Collins // Journal of Accountancy. – 2017. – Access mode: https://www.journalofaccountancy.com/issues/2017/oct/amazon-echo-vs-google-home.html.

31. MakerGuids – ESP32 vs Arduino Speed Comparison [Electronic resource]. – Access mode: https://www.makerguides.com/esp32-vs-arduino-speed-comparison.

32. Atomic14 – ESP32 Audio Input - MAX4466, MAX9814, SPH0645LM4H, INMP441 [Electronic resource]. – Access mode: https://atomic14.com/2020/09/12/esp32-audio-input.html.

33. STMicroelectronics – MEMS Microphones [Electronic resource]. – Access mode: https://www.st.com/en/mems-and-sensors/mems-microphones.html.

34. Malyshenko A. Tests of MAX98357A module [Electronic resource] / A..Malyshenko // Hackaday. – 2020. – Access mode: https://hackaday.io/project/173620-perfect-i2s-audio-dac-for-esp8266esp32/log/180811-tests-of-max98357a-module.

35. Random Nerd Tutorials – Getting Started with the ESP32 Development Board [Electronic resource]. – Access mode: https://randomnerdtutorials.com/getting-started-with-esp32.

36. Couriol B. Programming Microcontrollers with JavaScript – Q&A with Peter Hoddie and Lizzie Prader [Electronic resource] / B. Couriol // InfoQ. – 2020. – Access mode: https://www.infoq.com/articles/iot-javascript-microcontrollers-Moddable-tc53.

37. Rak, F. Comparison of ESP programming platforms / Rak F., Wiora J. – Computer Science and Information Technologies, Poland, 2021. – Vol. 2. – No. 2.– pp. 77–86.

38. Arduino - What is Arduino? [Electronic resource]. – Access mode: https://www.arduino.cc/en/Guide/Introduction.

39. PlatformIO - What is PlatformIO? [Electronic resource]. – Access mode: https://docs.platformio.org/en/latest/what-is-platformio.html.

40. Getting Started with Deep Learning ? [Electronic resource]. – Access mode: https://www.kdnuggets.com/2017/03/getting-started-deep-learning.html.

41. Karczewski D. What Is The Best Language For Machine Learning In 2022? [Electronic resource] / Karczewski D. – Access mode: https://www.ideamotive.co/blog/what-is-the-best-language-for-machine-learning.

42. Geeksforgeeks - Difference Between Jupyter and Pycharm [Electronic resource]. – Access mode: https://www.geeksforgeeks.org/difference-between-jupyter-and-pycharm.

43. Dilmegani C. In-Depth Look Into Google Dialogflow vs Competitors in 2022 [Electronic resource] / Dilmegani C. – Access mode: https://research.aimultiple.com/dialogflow.

ДОДАТОК А

Текст програми

А.1 Файл WitAiManager.h

```
#ifndef WitAiManager_H
#define WitAiManager_H

#include <Arduino.h>
#include <WiFiClient.h>

class WitAiManager
{
  public:
    void  InitConnect(IPAddress  LocalIP,  IPAddress  GatewayIP,  IPAddress
Subnet, const char *ssid, const char *password);
};
#endif
```

А.2 Файл WitAiManager.cpp

```
#include "WitAiManager.h"

void  WitAiManager::InitConnect(IPAddress  LocalIP,  IPAddress  GatewayIP,
IPAddress Subnet, const char *ssid, const char *password)
{
    if (WiFi.config(LocalIP, GatewayIP, Subnet) == false)
    {
        Serial.print("SHC. Wi-Fi. Failed");
    }

    WiFi.begin(ssid, password);

    Serial.print("-");

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(200);
        Serial.print(".");
    }

    Serial.print("SHC. Wi-Fi. Configured");
}
```

А.4 Файл OtaDriveManager.h

```
#ifndef OtaDriveManager_H
#define OtaDriveManager_H

#include <Arduino.h>
#include <HTTPUpdate.h>
```

```
class OtaDriveManager  {

  private:
    String _key;
    String _firmwareVersion;
    String GetDeviceID();
  public:
    OtaDriveManager(String key, String firmwareVersion);
    void CheckAvailibleUpdate();
    void TryUpdate();
};
#endif
```

A.4 Файл OtaDriveManager.cpp

```
#include "OtaDriveManager.h"

OtaDrive::OtaDriveManager(String key, String firmwareVersion)
{
    this->_key = key;
    _firmwareVersion = firmwareVersion;
}

String OtaDriveManager::GetDeviceID()
{
    String chipIdHex = String((uint32_t)(ESP.getEfuseMac() >> 32), HEX);
    chipIdHex += String((uint32_t)ESP.getEfuseMac(), HEX);
    return chipIdHex;
}

void OtaDriveManager::TryUpdate()
{
    String url = "http://otadrive.com/deviceapi/update?";
    url += "&v=" + _firmwareVersion;
    url += "k=" + _key;
    url += "&s=" + getChipId();
    if(WiFiClient client)
    {
      httpUpdate.update(client, url, _firmwareVersion);
      Serial.print("SHC. Update checked");
    }
}
```

A.5 Файл WitAiManager.h

```
#ifndef WitAiManager_H
#define WitAiManager_H

#include <arduinoJson>
#include <stdint.h>
```

```
#include <string>

typedef struct
{
    float deviceConf;
    string IntenteName;
    float intenteConfidence;
    string text;
    string deviceNick;
    string traiteValue;
    float traitConfidence;
} Intent;

class ClientWiFi;

class WitAiManager
{
    private:
        ClientWiFi *m_wifi_client;

    public:
        WitAiManager(const char *access_key);
        ~WitAiManager();
        bool inited();
        void startCh(int size_in_bytes);
        void sendChData(const uint8_t *data, int size_in_bytes);
        void finishCh();
        Intent getAnswer();
};
```

A.6 Файл wifi_connect.cpp

```
#include <ArduinoJson.h>
#include "WiFiClientSecure.h"
#include "WitAiManager.h"

WitAiManager::WitAiManager(const char *access_key)
{
    wifiClient = new WiFiClientSecure();
    wifiClient->setInsecure();
    wifiClient->connect("api.wit.ai", 443);
    char authorization_header[100];
    snprintf(authorization_header,   100,   "authorization:   Bearer   %s",
access_key);
    wifiClient->printOut("POST /speech?v=20200927 HTTP/1.1");
    wifiClient->printOut("host: api.wit.ai");
    wifiClient->printOut(authorization_header);
    wifiClient->printOut("content-type:  audio/raw;  encoding=signed-integer;
bits=16; rate=16000; endian=little");
    wifiClient->printOut("transfer-encoding: chunked");
    wifiClient->printOut();
}
```

```
void WitAiManager::startCh(int size_in_bytes)
{
    wifiClient->printf("%X\r\n", size_in_bytes);
}

void WitAiManager::sendChData(const uint8_t *data, int size_in_bytes)
{
    wifiClient->write(data, size_in_bytes);
}

bool WitAiManager::inited()
{
    return wifiClient->inited();
}

void WitAiManager::finishCh()
{
    wifiClient->print("\r\n");
}

Intent WitAiManager::getAnswer()
{
    wifiClient->print("\r\n");
    wifiClient->print("0\r\n");
    int status = -1;
    int contentLeng = 0;
    while (wifiClient->inited())
    {
        char buffer[255];
        int read = wifiClient->readBytesUntil('\n', buffer, 255);
        if (read > 0)
        {
            buffer[read] = '\0';
            // blank line indicates the end of the headers
            if (buffer[0] == '\r')
            {
                break;
            }
            if (strncmp("HTTP", buffer, 4) == 0)
            {
                sscanf(buffer, "HTTP/1.1 %d", &status);
            }
            else if (strncmp("Content-Length:", buffer, 15) == 0)
            {
                sscanf(buffer, "Content-Length: %d", &contentLeng);
            }
        }
    }
    Serial.printf("Http status is %d with content length of %d\n", status,
contentLeng);
    if (status == 200)
    {
        filter["entities"]["device:device"][0]["value"] = true;
        filter["entities"]["device:device"][0]["confidence"] = true;
        filter["text"] = true;
```

```
        filter["intents"][0]["name"] = true;
        StaticJsonDocument<500> filter;
        filter["intents"][0]["confidence"] = true;
        filter["traits"]["wit$on_off"][0]["value"] = true;
        filter["traits"]["wit$on_off"][0]["confidence"] = true;
        StaticJsonDocument<500> doc;
        deserializeJson(doc,                                    *wifiClient,
DeserializationOption::Filter(filter));

        const char *text = doc["text"];
        const               char               *deviceNick               =
doc["entities"]["device:device"][0]["value"];
        const char *IntenteName = doc["intents"][0]["name"];
        float intenteConfidence = doc["intents"][0]["confidence"];
        float traitConfidence = doc["traits"]["wit$on_off"][0]["confidence"];
        float deviceConf = doc["entities"]["device:device"][0]["confidence"];
        const char *traiteValue = doc["traits"]["wit$on_off"][0]["value"];

        return Intent{
            .text = (text ? text : ""),
            .deviceNick = (deviceNick ? deviceNick : ""),
            .IntenteName = (IntenteName ? IntenteName : ""),
            .deviceConf = deviceConf,
            .traitConfidence = traitConfidence};
            .intenteConfidence = intenteConfidence,
            .traiteValue = (traiteValue ? traiteValue : ""),
    }
    return Intent{};
}

WitAiManager::~WitAiManager()
{
    delete wifiClient;
}
```

ДОДАТОК Б

Слайди презентації

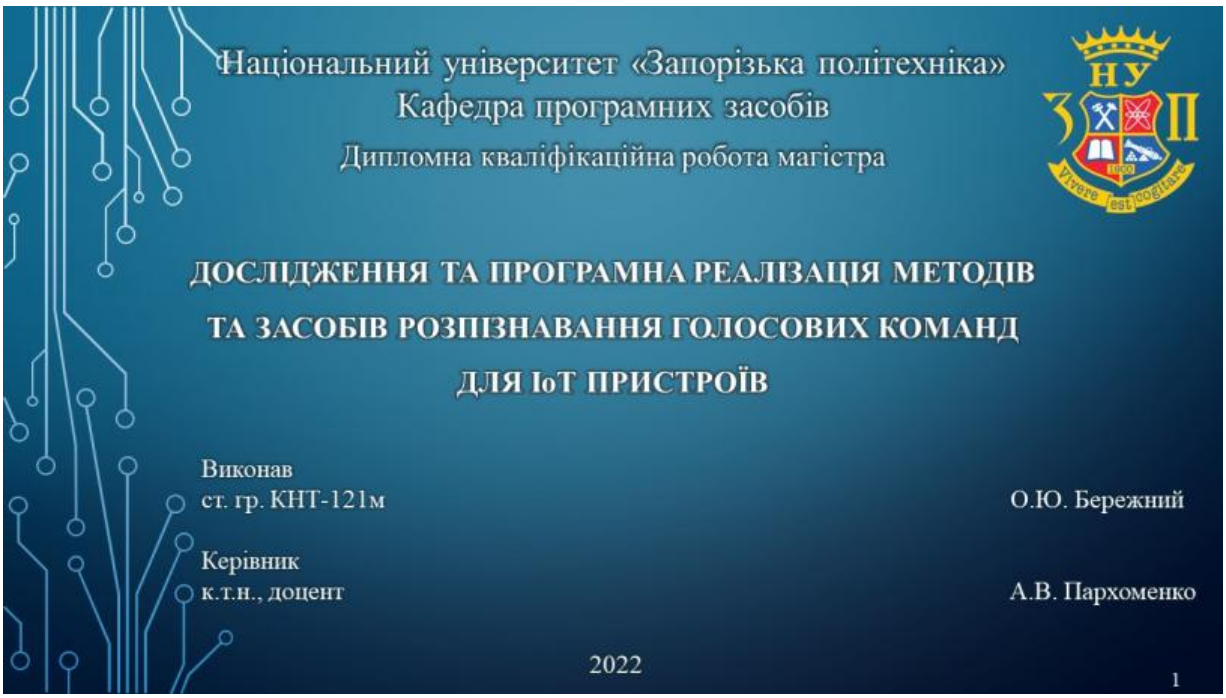


Рисунок Б.1 – Слайд № 1

Рисунок Б.2 – Слайд № 2

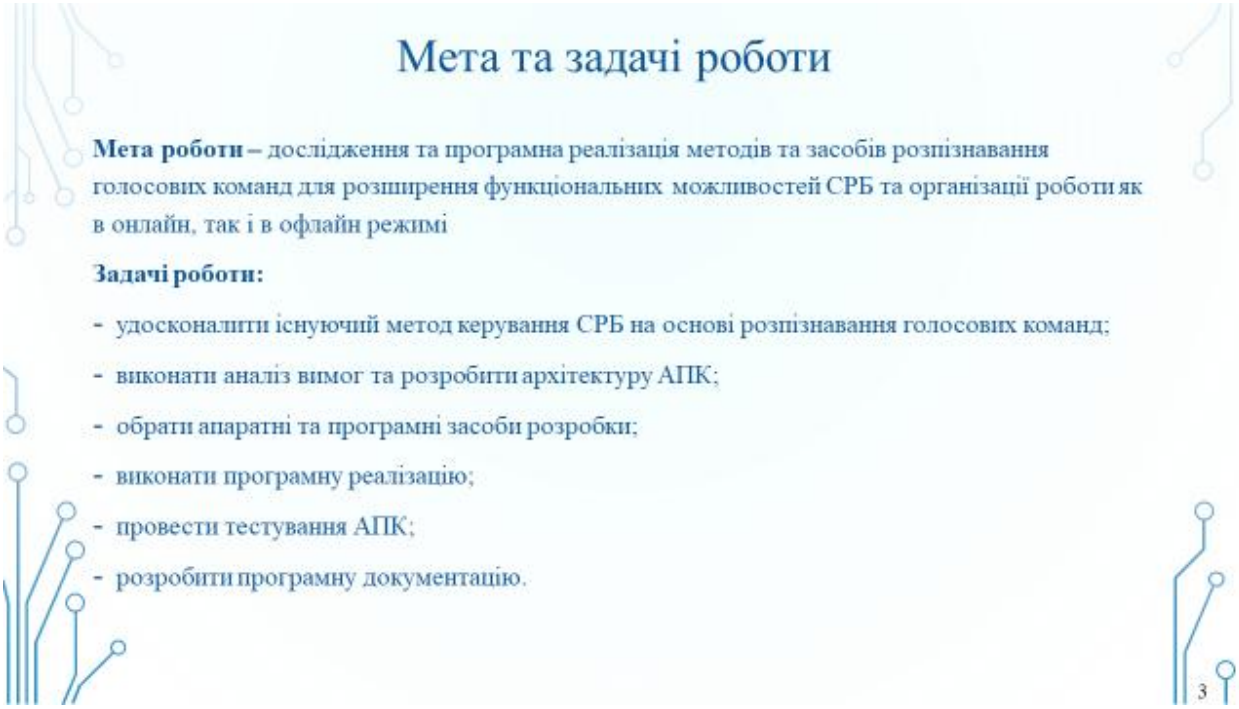


Рисунок Б.3 – Слайд № 3

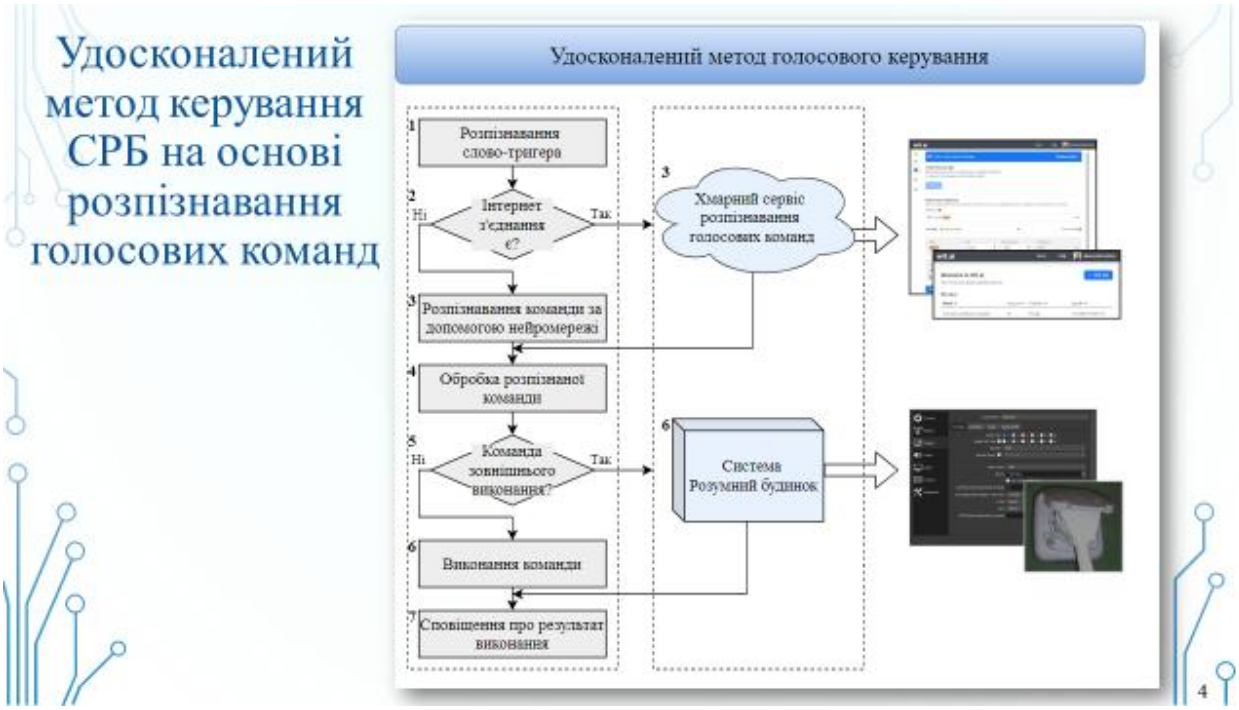


Рисунок Б.4 – Слайд № 4

Порівняння та вибір типу нейронної мережі

	ANN	CNN	RNN
Загальна характеристика	Один з найпростіших	Один з найпопулярніших	Найбільш складний
Структурна особливість	Інформація тече лише в одному напрямку	Базується на кількох шарах вузлів, вкл. один або більше згорткових рівнів	Інформація тече в різних напрямках, властиве самонавчання
Тип даних	Табличні і текстові данні	Зображення	Навчання з даними послідовності
Складність	Вважається простим	Вважається потужнішим	Вважається потужним через потенціал самонавчання
Особливості	Вміння роботи з неповними знаннями	Точність розпізнавання образів	Пам'ять і самонавчання
Головний недолік	Апаратна залежність	Потреба в великій кількості навчальних даних	Складне навчання
Використання	Вирішення комплексних проблем таких як прогнозний аналіз	Розпізнавання зображень	Високоточний аналіз мовлення, настроїв, швидкості та ін.

https://levity.ai/blog/neural-networks-cnn-ann-rnn

5

Рисунок Б.5 – Слайд № 5

Аналіз вимог до АПК

6

Рисунок Б.6 – Слайд № 6

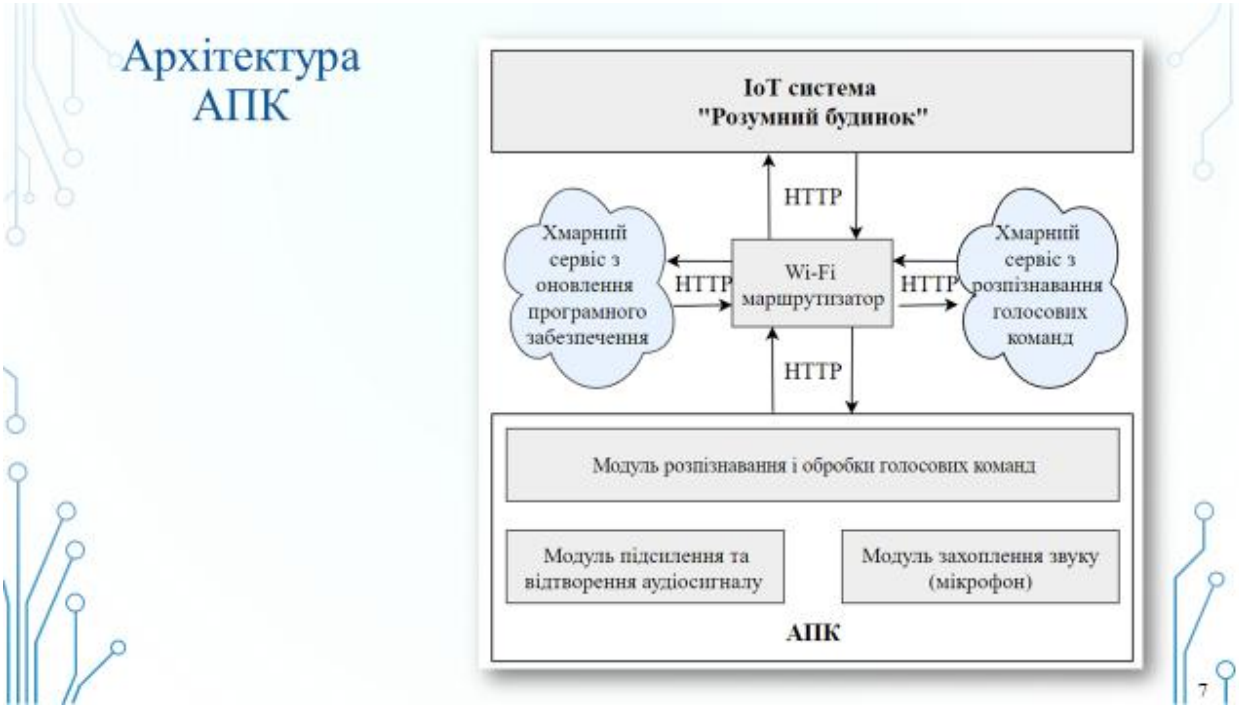


Рисунок Б.7 – Слайд № 7

	Bluetooth	Zigbee	Z-Wave	Wi-Fi	LoRaWan
Уєднання	Сітка, Зірка	Сітка, Зірка	Сітка	Зірка	Зірка
Швидкість	1-3 Мбіт/с	250 Кбіт/с	До100 Мбіт/с	До 1 Гбіт/с	50 Мбіт/с
Діапазон частот	2.4 ГГц	2.4 ГГц	Діапазон, залеж. від гео. полож	2.4 ГГц	Діапазон, залеж. від гео. полож.
Покриття	До 100 м	До 200 м	До 100 м	До 70 м	До 15 км
Споживн.	помірне	низьке	низьке	помірне	низьке
Вартість	Низька	Середня	Висока	Низька	Низька
Ліцензування	Безкоштовно	Комісія за пристрій	Комісія за пристрій	Безкоштовно	Безкоштовно

*https://www.daviteq.com/blog/en/what-are-the-most-popular-iot-protocols

Рисунок Б.8 – Слайд № 8

Вибір апаратних засобів

	ESP-32	ESP-8266	Arduino UNO	Arduino Mega 2560
Мікроконтро-лер	Xtensa Dual Core 32-bit	Tensilica 32-bit Xtensa	ATMega328P	ATMega2560
Флеш-пам'ять	4 МБ	4 Мб	32 КБ	256 Кб
SRAM	520 Кб	128 Кб	2 Кб	8 Кб
EEPROM	512 Б	512 Б	1 КБ	4 Кб
Тактова частота	До 240 МГц	80 / 160 МГц	16 МГц	16 МГц
Робоча напруга	3,3 В	3,3 В	5 В	5 В
Wi-Fi	802.11 b/g/n	802.11 b/g/n	-	-
Bluetooth	Bluetooth, BLE	-	-	-
Споживання	80-90 мА	15мкА-400мА	45–80 мА	150 мА
ШІМ	16	4	6	14
UARTs	3	2	1	4
Аналогові піни	До 18	1	6	16
Цифрові піни	36	17	14	54

Порівняння SoC модулів*

INMP441

MAX98357A

ESP-32

Загальна схема з'єднання

*https://www.makerguides.com/esp32-vs-arduino-speed-comparison/

Рисунок Б.9 – Слайд № 9

Аналіз та вибір інструментарію розробки

Мова	Платформа	Підручники навчальні матеріали	Зручність використання і модульність	Швидкість	Підтримка декількох GPU
Python	TensorFlow	+++	+++	++	++
Python	Theano	++	+	++	+
Lua	Torch	+	+ +	+++	++
C++	Caffe	+	+	+	+
C++	CNTK	+	+	++	+
R	MXNet	++	++	++	++

	Jupyter	Pycharm
Визначення	Інтерактивна обчислювальна платформа	Розумний редактор коду
Містить	Поєднує живий код, рівняння, описовий текст, візуалізації, інтерактивні інформаційні панелі та інші медіа	Поєднує підтримку декількох мов
Класифік.	Блокнот Data Science	IDE
Забезпечує	Виконання вбуд. коду блоками Підтримку вбуд. Графіків	Розумне автозавершення Інтелект. аналіз коду
Особливості	Блокова структура, підтримує kernel, а також latex	Потужний рефакторинг, інтеграція вірт. середовищ
Гнучкість	Доволі гнучкий	Менш гнучкий, повільніш

Критерій / Середовище	PlatformIO	Arduino IDE
Налаштування інтерфейсу	наявне	слабке
Складність опанування	середня	низька
Автодоповнення	наявне	відсутнє
Навігація за кодом	наявне	відсутнє
Система відслідковування версій	наявне	відсутнє

	ANN	CNN	RNN
Характер.	Один з найпростіших	Один з найпоширеніш.	Найбільш складний
Структурна особливість	Інформація тече лише в одному напрямку	Базується на кількох слоях вузлів, один або більше згорт. рівнів	Інформація тече в різних напрямках, самонавчання
Тип даних	Табличні і текстові	Зображення	Навчання з даними послідовності
Складність	Простий	Вважається потужнішим	Потужний завдяки потенціалу самонавчання
Особливості	Вміння роботи з неповними знаннями	Точність розпізнав. образів	Пам'ять і самонавчання
Недолік	Апаратна залежність	Великі навч. дані	Складне навчання
Використання	Вирішення комплексних проблем, таких як прогнозний аналіз	Розпізнавання зображень	Високоточний аналіз настроїв і розпізнавання швидкості мовлення

Мова	Кількість кадрів, оброблюваних в секунду
C++	35
MP	7 – 8
JS	5-7
Lua	4

http://www.deamotive.co/blog/what-is-the-best-language-for-machine-learning, https://www.geeksforgeeks.org/difference-between-jupyter-and-pycharm, https://levity.ai/blog/neural-networks-cnn-ann-rn

Рисунок Б.10 – Слайд № 10

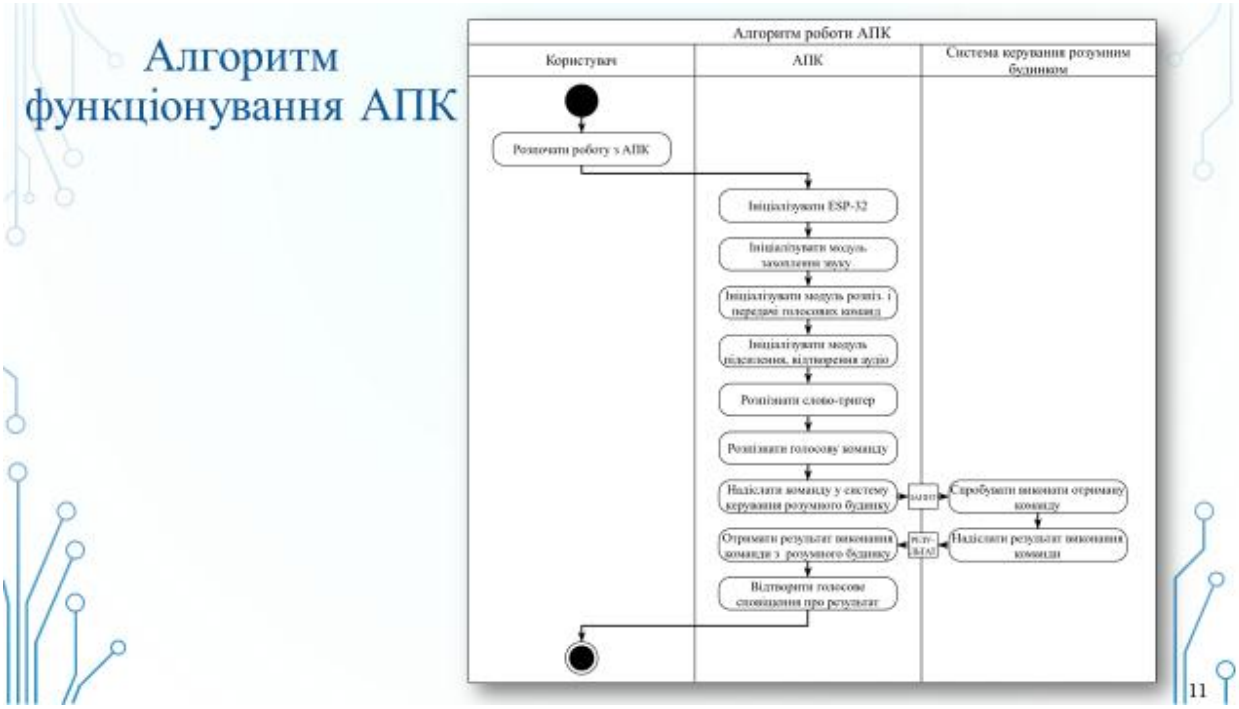


Рисунок Б.11 – Слайд № 11

Рисунок Б.12 – Слайд № 12

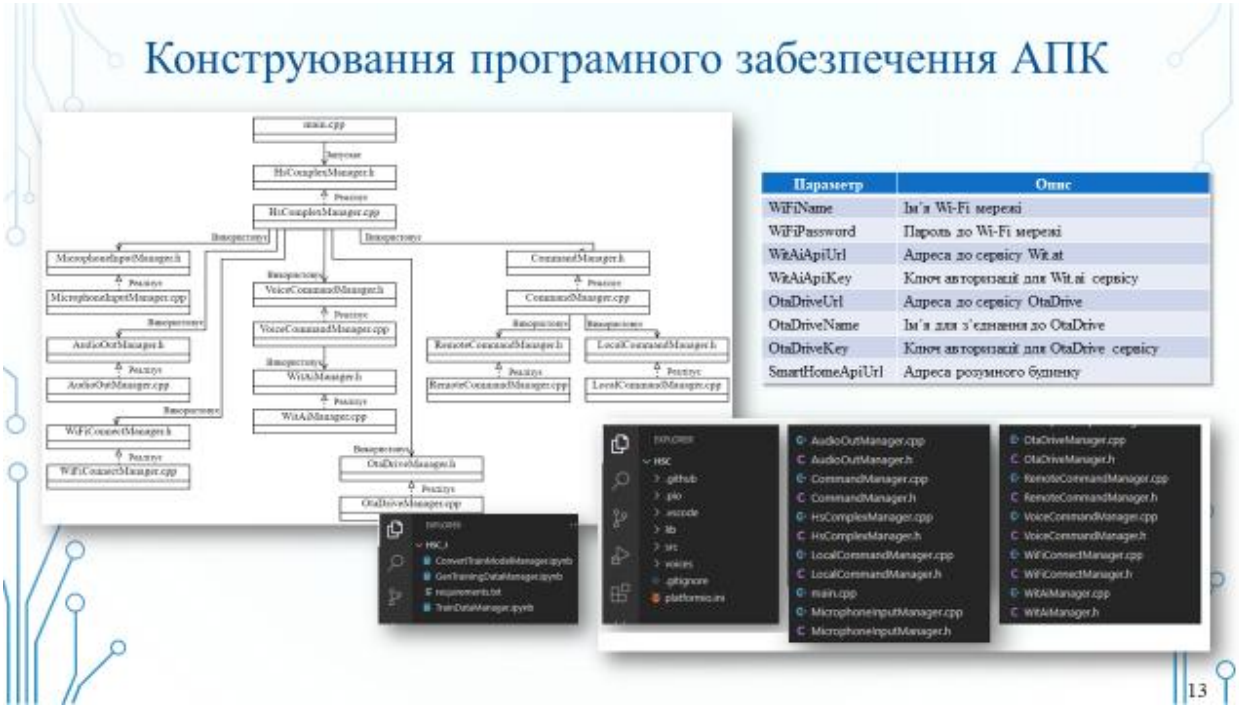


Рисунок Б.13 – Слайд № 13

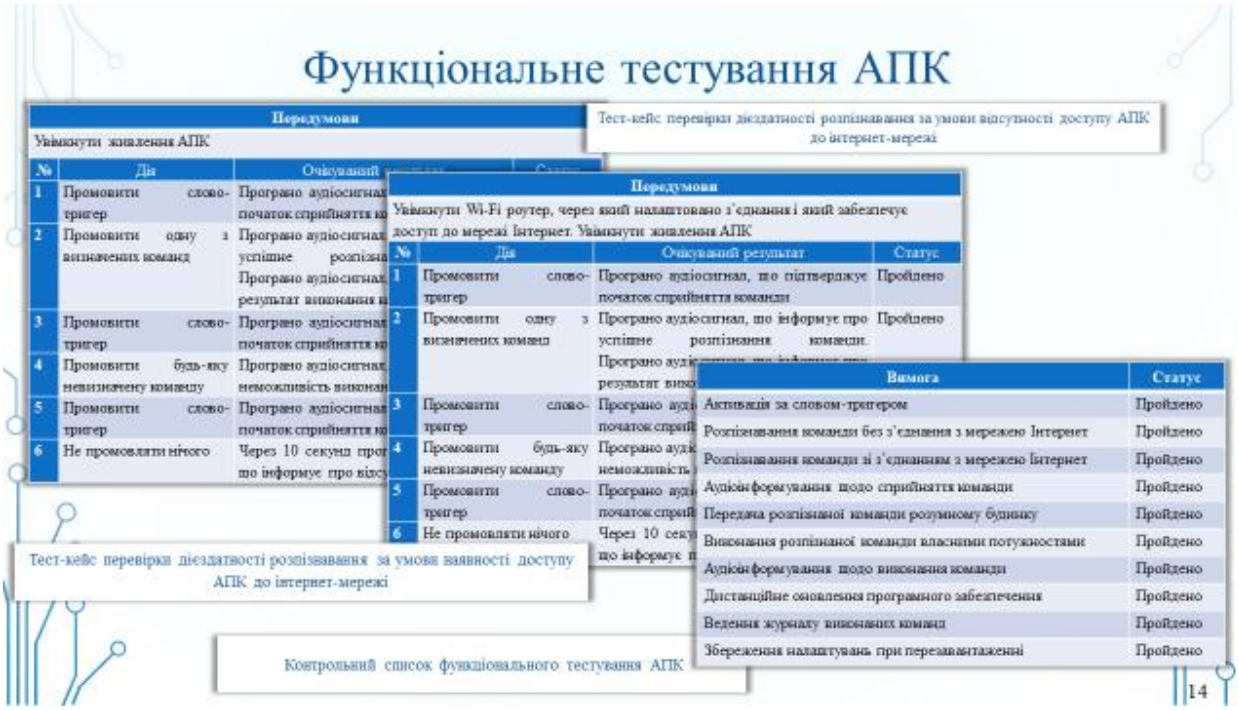
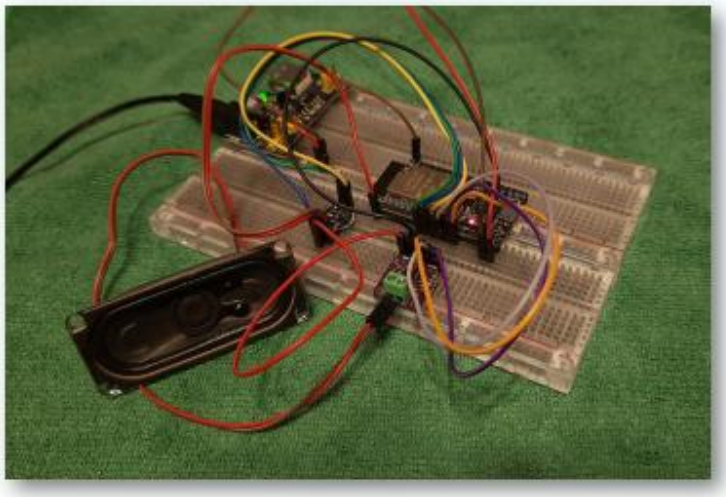


Рисунок Б.14 – Слайд № 14

Дослідження роботи прототипу АПК



15

Рисунок Б.15 – Слайд № 15

Висновки

Наукова новизна роботи полягає у тому, що удосконалено принцип роботи голосового керування IoT системами за рахунок використання нейромережі, що на відміну від існуючих підходів дозволяє забезпечити стабільне розпізнавання голосових команд користувачів навіть у разі відсутності Інтернет з'єднання.

Практична цінність результатів роботи полягає у тому, розроблене апаратно-програмне рішення може бути використано для керування IoT системами типу «Розумний будинок», також воно може покращити процес взаємодії з IoT пристроями людей з особливими потребами, оскільки використання голосового асистенту знизить навантаження на зорову систему користувача.

Як розвиток проєкту заплановано розширення кількості доступних команд, підвищення зручності використання комплексу через розробку вебінтерфейсу для забезпечення можливостями керування налаштуваннями за допомогою веббраузеру.

Основні положення та результати роботи були представлені в тезах доповіді НПК Тиждень науки – 2022 та на XI міжнародній науково-практичній конференції «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій».

16

Рисунок Б.16 – Слайд № 16

Рисунок Б.17 – Слайд № 17

ДОДАТОК В

Фотозвіт процесу розробки АПК

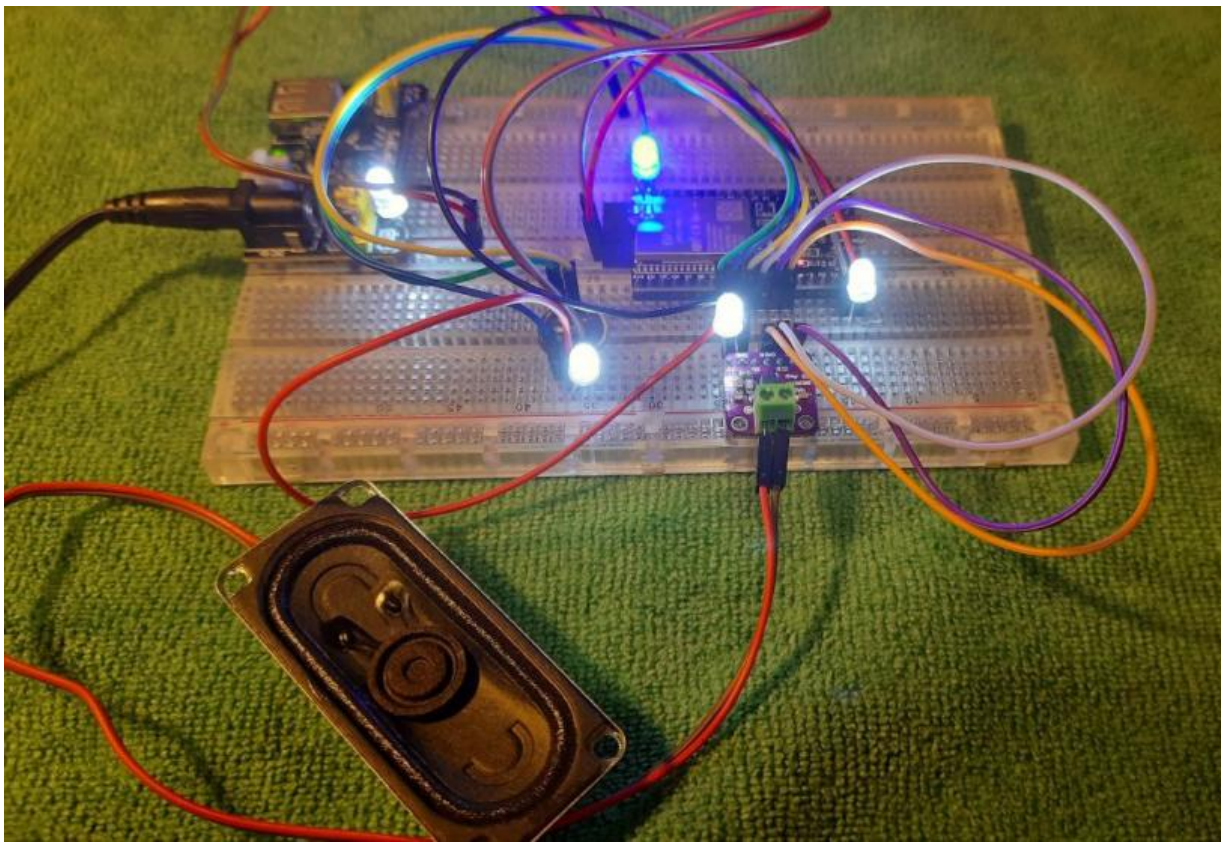


Рисунок В.1 – Налаштування з’єднання складових модулів АПК

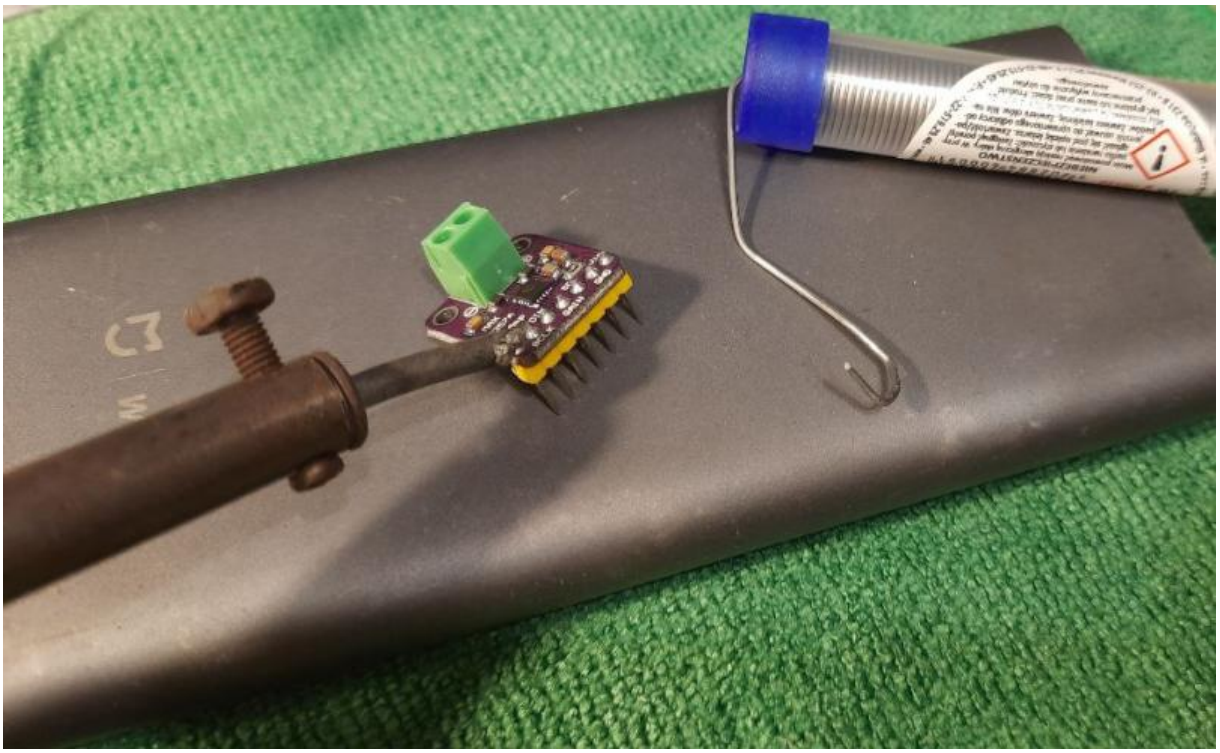


Рисунок В.2 – Пайка компонентів АПК

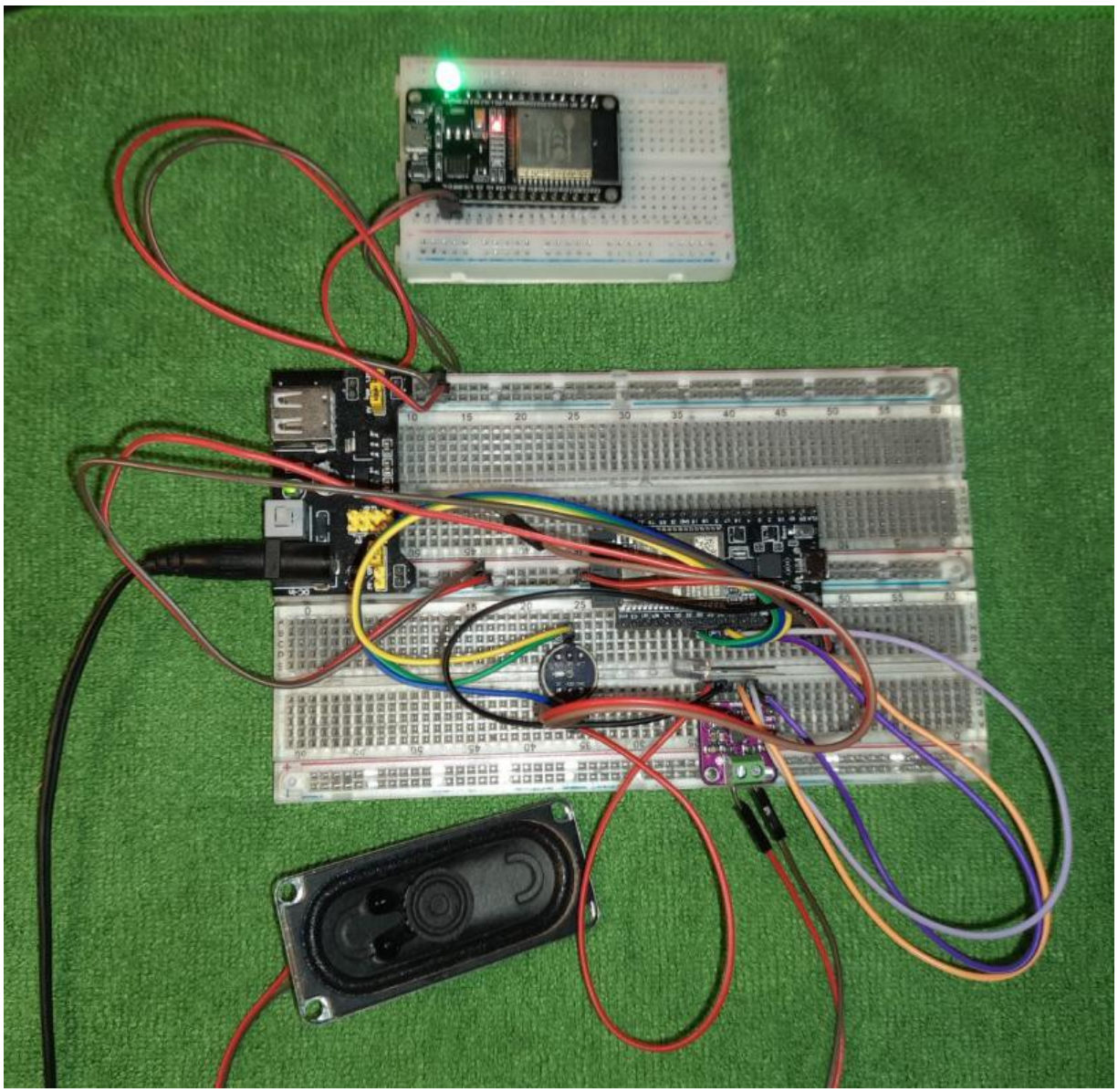


Рисунок В.3 – Тестування роботи АПК на базі ESP-32 в системі Розумний будинок