

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Програмно-технічна реалізація голосового асистента для кіберфізичної системи
"Розумний будинок"

Назва теми

КВРКІ 180204.18.02.25 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія»

Назва

Виконав: студент IV курсу, група KI-18-2

Голо
Підпис

А.О. Головатюк

Ініціали, прізвище

Керівник

[Підпис]
Підпис, дата

К.Ю. Бобровнікова

Ініціали, прізвище

Нормоконтролер

С.М. Лисенко

До захисту допускаю:

Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

[Підпис]
Підпис

Т.О. Говорущенко

Ініціали, прізвище

« 1 » 06 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорушенко

“ 11 ” 01 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Головагюку Андрію Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок".

Керівник проекту (роботи) Бобровнікова К.Ю., к.т.н.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 02.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Дослідження поставленої задачі як складової кіберфізичної системи “розумний будинок”

Проектування голосового асистента для кіберфізичної системи «розумний будинок»

Програмно-апаратна реалізація і тестування голосового асистента для кіберфізичної системи «розумний будинок»


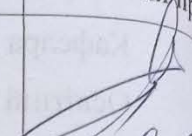
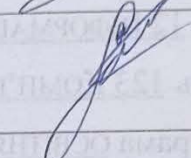
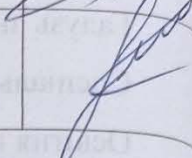
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Схеми з'єднань пристрою

Інтерфейси програмно-апаратного засобу

Блок-схеми програм

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання при
Нормоконтроль	Лисенко С.М., професор кафедри КІСП		
Антиплагиат	Нічепорук А.О., доцент кафедри КІСП		

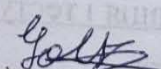
7. Дата видачі завдання « 11 » 01 2022 р.

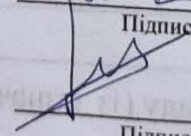
КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Пр
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2022	ВИ
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	ВИ
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	ВИ
4	Робота над розділом 2 – проектування підсистеми	01.04.2022	ВИ
5	Робота над розділом 3 – програмно-апаратна реалізація підсистеми	30.04.2022	ВИ
6	Оформлення пояснювальної записки згідно вимог	20.05.2022	ВИ
7	Попередній захист ВКР	24.05.2022	ВИ
8	Захист ВКР на засіданні ЕК	03.06.2022	

Студент

Керівник проекту (роботи)

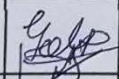
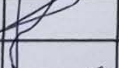
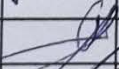
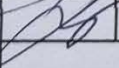

Підпис


Підпис

А.О. Головатюк
Ініціали, прізвище

К.Ю. Бобровнікова
Ініціали, прізвище

№ р я д к а	ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ екз	П р и м і т к а
			<u>Текстові документи</u>			
1		КВРКІ 180204.18.02.25 ПЗ	Пояснювальна записка	66		
			<u>Графічні матеріали</u>			
2		КВРКІ 180204.18.02.25 Е8	Схеми апаратних з'єднань	1		
3		КВРКІ 180204.18.02.25 Е8	Інтерфейси програмно-апаратного засобу	1		
4		КВРКІ 180204.18.02.25 Е8	Блок-схеми програм	1		

Зм	Арк	№ докум	Підпис	Дата	КВРКІ 180204.18.02.25 ВП			
Розробив		Головаток			Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Бобровнікова				У	1	1
Н. контр.		Лисенко				ХНУ, КІ-18-2		
Затв.		Говорушченко						

АНОТАЦІЯ

Тема кваліфікаційної роботи: Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок".

Автор роботи: Головатюк Андрій Олександрович.

Керівник роботи: Бобровнікова Кіра Юліївна.

Пояснювальна записка: 58 с., 32 рис., 3 табл., 4 дод., 34 джерела.

Графічна частина: 10 презентаційних слайдів.

КІБЕРФІЗИЧНА СИСТЕМА, РОЗУМНИЙ БУДИНОК, ARDUINO NANO, ГОЛОСОВИЙ АСИСТЕНТ, РОЗПІЗНАВАННЯ МОВИ.

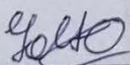
Метою кваліфікаційної роботи є проектування та програмно-технічна реалізація голосового асистента для підвищення комфорту керування кіберфізичною системою «Розумний будинок» за допомогою голосу.

Об'єктом дослідження є процес розпізнавання голосу та процес керування кіберфізичною системою «Розумний будинок» за допомогою голосу.

Предметом дослідження є програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок".

Для досягнення поставленої мети використовуються алгоритми розпізнавання голосу.

Практична цінність полягає в розробленні програмно-технічної реалізації голосового асистента для кіберфізичної системи "Розумний будинок", що надає можливість підвищити комфорт керування кіберфізичною системою «Розумний будинок».



Підпис студента

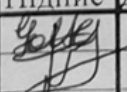
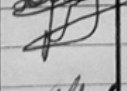
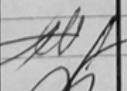
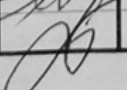
01.06.2022

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	4
ВСТУП.....	5
1. ДОСЛІДЖЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ЯК СКЛАДОВОЇ КІБЕРФІЗИЧНОЇ СИСТЕМИ “РОЗУМНИЙ БУДИНОК”	6
1.1 Інтернет речей та технології керування кіберфізичною системою "Розумний будинок"	6
1.2 Дослідження відомих реалізацій голосового асистента.....	9
1.2.1 Apple HomeKit Netatmo.....	9
1.2.2 Google Home.....	11
1.2.3 Amazon Echo	11
1.2.4 Висновки на базі досліджень відомих реалізацій голосового асистента для кіберфізичної системи «Розумний будинок»	13
1.3 Алгоритми розпізнавання голосу та мовлення.....	13
1.4 Огляд апаратних складових для реалізації голосового асистента для кіберфізичної системи «Розумний будинок»	16
1.4.1 Arduino Nano	16
1.4.2 Цифрові та аналогові фільтри. ПР-фільтр другого порядку.....	18
1.5 Постановка задачі	24
1.6 Висновки.....	25
2. ПРОЄКТУВАННЯ ГОЛОСОВОГО АСИСТЕНТА ДЛЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК».....	26
2.1 Обґрунтування вибору програмного забезпечення та апаратних ресурсів	26

КВРКІ 180204.18.02.25 ПЗ

М.	Арк	№докум.	Підпис	Дата		Літера	Аркуш	Аркушів
виконав.		Головатюк А.О.			Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок"		2	58
перевір.		Бобровникова К.Ю.						
і.контр.		Лисенко С.М.						
атвер.		Говорушенко Т.О.						

ХНУ, КІ-18-2

2.2	Програмна реалізація підключення мікрофону до плати та тестування його роботи.....	31
2.3	Взаємодія голосового асистента з кіберфізичною системою «Розумний будинок».....	33
2.5	Висновки	34
3.	ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ ГОЛОСОВОГО АСИСТЕНТА ДЛЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»....	35
3.1	Встановлення апаратних з'єднань для реалізації голосового асистента	35
3.2	Збір даних з АЦП Arduino Nano для подальшої обробки.....	38
3.3	Алгоритм роботи цифрового фільтру діапазону частот та розрахунок часу виконання обчислень	40
3.4	Розрахунок коефіцієнтів цифрового фільтру діапазону частот	42
3.5	Обробка шаблонів висловлювань.....	46
3.6	Використання алгоритму динамічного викривлення часу	47
3.7	Проведення навчання алгоритму розпізнавання шаблонів	49
3.8	Алгоритм розпізнавання мовлення користувача	53
3.9	Програмно-апаратна реалізація голосового асистента для кіберфізичної системи «Розумний будинок».....	54
3.10	Інтерпретація команд, отриманих голосовим асистентом.....	55
3.11	Висновки.....	61
	ВИСНОВКИ.....	62
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	64
	ДОДАТОК А Копія креслення «Схеми апаратних з'єднань».....	68
	ДОДАТОК Б Копія креслення «Алгоритми реалізації голосового асистента».....	69
	ДОДАТОК В Копія креслення «Інтерфейси програмно-апаратного засобу»	70
	ДОДАТОК Г Лістинг коду програми по обробці команд для кіберфізичної.....	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

AI – Artificial intelligence (Штучний інтелект)

IEEE – Electrical and Electronics Engineers (Інститут електротехніки та електроніки)

IETF – Internet Engineering Task Force (спеціальна група з розробки Інтернету)

IoT – Internet of things (Інтернет речей)

IoV – Internet of Vehicles (Інтернет транспортних засобів)

MFCC – Mel-frequency cepstral coefficients (Мелчастотні кепстральні коефіцієнти)

UID – unique identifier (Унікальний ідентифікатор)

АЦП – Аналогово-цифровий перетворювач

ОС – Операційна система

ПК – персональний комп'ютер

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		4

ВСТУП

За довгі роки еволюції люди весь час вдосконалювали своє життя, роблячи його більш комфортним та простим. Найбільше це проявляється у приміщеннях, в яких вони живуть. Середньостатистична людина проводить більшість часу вдома, відпочиваючи від роботи або працюючи прямо звідти. Сучасні будинки оснащені великою кількістю електроприладів, які роблять буденні справи більш простими та швидкими в використанні. Але і їх використання може бути спрощене та автоматизоване.

Це стає можливим за допомогою системи «Розумний будинок». Дана система підключається до більшості приладів в будинку і бере їх контроль на себе, автоматизуючи більшість повсякденних задач. Здебільшого кіберфізична система «Розумного будинку» включає в себе такі пристрої керування, як настінні термінали, мобільні телефони, комп'ютери, веб-інтерфейси, тощо. Але для ще більшого підвищення зручності керування даною кіберфізичною системою можна здійснювати за допомогою голосового асистента. Такий асистент сприймає та розпізнає команди власника, які він озвучує вголос. Далі, опираючись на дані команди, система посилає сигнали на прилади, які до неї підключені. Таким чином можна, наприклад, увімкнути електричний чайник на кухні не встаючи з дивану у вітальні. Для цього потрібно всього лише вголос сказати команду, яка закладена в програму розумного асистента.

Дана тема є дуже актуальною при значному підвищенні кількості домів, оснащених кіберфізичною системою «Розумний будинок».

Також у майбутньому дана тема може бути розширена за допомогою розробки системи штучного інтелекту та підключенні його до розумного асистента для більш глибокого розуміння намірів власника.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		5

1. ДОСЛІДЖЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ЯК СКЛАДОВОЇ КІБЕРФІЗИЧНОЇ СИСТЕМИ “РОЗУМНИЙ БУДИНОК”

1.1 Інтернет речей та технології керування кіберфізичною системою "Розумний будинок"

Розумний дім – зручне налаштування будинку, у якому приладами та пристроями можна керувати автоматично та віддалено із будь-якої точки, де є підключення до мережі Інтернет, за допомогою мобільного або будь-якого іншого мережевого пристрою. Пристрої у розумному будинку з’єднані між собою через мережу Інтернет, що дозволяє користувачам дистанційно керувати функціями, як освітлення, системою домашнього кінотеатру, побутовими приладами, вентиляцією, тощо[1].

Пристрої розумного будинку усі з’єднані між собою та доступні через одну центральну точку – ПК, планшет, смартфон або ігрову консоль. Це дає можливість керувати за допомогою однієї системи домашньої автоматизації такими системами, як телевізори, камери нагляду, освітлення, дверні замки, термостати і навіть холодильником. Система встановлюється на мережевий пристрій і користувач має змогу створити розклад часу у який певні зміни будуть вступати в силу.

Розумні побутові прилади мають здатність до самонавчання, тому вони можуть самостійно вивчати розклад домовласників та вносити зміни за такої потреби. Розумні будинки у яких є підтримка керування освітленням дозволяють власникам будинків знизити споживання електроенергії та отримати вигоду шляхом економії витрат на електроенергію[2]. Системи аутентифікації сповіщають власників у їх відсутність якщо у будинку буде виявлено будь-який рух. Також деякі з них можуть самостійно викликати поліцію або пожежну службу у разі необхідності.

Після підключення розумного будинку усі розумні прилади, які входять в його склад, стають частиною технології інтернету речей (IoT) – мережі фізичних об’єктів, які можуть збирати електронну інформацію та обмінюватися нею[3].

Нижче наведено кілька переваг та недоліків розумних будинків.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		6

Переваги:

- 1) встановлення технології розумного будинку забезпечує власникам будинків зручність;
- 2) користувачі можуть отримувати сповіщення щодо проблем у своїх домівках, оскільки вони усі підключені до портативного пристрою;
- 3) значна економія коштів через те що прилади вмикаються тільки тоді, коли це потрібно.

Недоліки:

- 4) через недосконалість безпеки існує ризик того, що досвідчені хакери зможуть отримати доступ до систем будинку через мережу Інтернет;
- 5) компоненти самої системи та їх встановлення іноді коштують дуже дорого.

Термін IoT, або Інтернет речей, відноситься до колективної мережі підключених пристроїв і технології, яка полегшує зв'язок між пристроями та хмарою, а також між самими пристроями. Усі обчислювальні пристрої у такій системі мають свої унікальні ідентифікатори (UID) та можуть передавати дані між собою без втручання людини[4].

Система IoT складається з розумних пристроїв, які підтримують Інтернет та використовують вбудовані системи, такі як датчики, процесори та комунікаційне обладнання призначене для збору, надсилання та маніпуляції з даними, які вони отримують з свого середовища.

Пристрої IoT обмінюються даними, отриманими з датчиків, які вони збирають, коли підключаються до шлюзу IoT або будь-якого іншого периферичного пристрою, дані з якого надсилаються у хмару задля аналізу. Іноді пристрої взаємодіють з спорідненими пристроями і діють на основі інформації, яку вони отримують один від одного[5]. Усі ці пристрої виконують більшу частину роботи навіть без участі людини, хоча люди мають можливість взаємодіяти з пристроями – наприклад, для їхнього налаштувати або надання їм інструкцій або отримання доступу до даних.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		7

IoT може використовувати штучний інтелект (AI) та машинне навчання для того, щоб допомогти зробити процес збору даних динамічнішим та простішим.

IoT найбільш поширений у транспортних, виробничих та комунальних організаціях, у яких використовуються датчики та інші пристрої що належать до Інтернету речей. Однак також він знайшов варіанти використання для організацій в галузі сільського господарства, інфраструктури і індустрії домашньої автоматизації[6].

Нижче наведені переваги та недоліки IoT.

Переваги IoT:

- 1) можливість доступу до інформації на будь-якому пристрої з будь-якого місця та у будь-який час;
- 2) автоматизація завдань, що зменшує потребу у людському втручанні;
- 3) технологія покращує взаємозв'язок між підключеними електронними приладами;
- 4) передача пакетів даних між пристроями через підключену мережу, що заощаджує час та гроші.

Недоліки IoT[8]:

- 1) оскільки кількість підключених пристроїв весь час збільшується, а між пристроями поширюється усе більше інформації, пропорційно збільшується ймовірність того, що хакер викраде конфіденційну інформацію;
- 2) якщо у системі з'явиться вірус або помилка то є велика ймовірність того, що усі підключені пристрої буде пошкоджено;
- 3) пристроям різних виробників важко взаємодіяти один з одним через відсутність міжнародних стандартів.

Існує кілька стандартів IoT, в числі яких наступні:

- 1) ZigBee – це бездротова малопотужна мережа з низькою швидкістю передачі даних. Використовується переважно в промислових умовах. ZigBee базується на стандарті 802.15.4 Інституту електротехніки та електроніки (IEEE);
- 2) IPv6 через малопотужні бездротові персональні мережі - це відкритий стандарт, визначений спеціальною групою з розробки Інтернету (IETF);

					КВРКІ 180204.18.02.25 ПЗ	Арк
						8
Зм.	Арк.	№докум.	Підпис	Дата		

3) LiteOS – це Unix-подібна ОС створена для бездротових сенсорних мереж. LiteOS підтримує смартфони, переносні пристрої, розумні будинки, інтелектуальні виробничі програми та Інтернет транспортних засобів (IoV). ОС також являється платформою для розробки розумних пристроїв.

1.2 Дослідження відомих реалізацій голосового асистента

Зараз є досить багато реалізацій голосового асистенту за допомогою яких можна керувати розумним будинком. У цьому розділі розглянуто декілька голосових асистентів різних компаній, їхні характеристики, переваги та недоліки. Серед розглянутих пристроїв є:

- 1) Apple HomeKit Netatmo[9];
- 2) Google Home[10];
- 3) Amazon Echo[11].

Під час розгляду цих пристроїв особливу увагу слід звернути на такі характеристики:

- 1) ціна пристрою;
- 2) сумісність з різними девайсами та операційними системами;
- 3) сприйняття різних мов;
- 4) підтримка інших засобів для управління.

1.2.1 Apple HomeKit Netatmo

Apple HomeKit[9] – це програмне середовище створене для розумного дому, яке Apple розробила для реалізації керування та взаємозв'язку різних підключених об'єктів за допомогою програми Home, яка може бути встановлена на iPhone та iPad з операційною системою iOS 10 та вище. За допомогою HomeKit користувач може керувати великою кількістю пристроїв домашньої автоматизації та взаємодіяти

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		9

з HomeKit. Також до неї можуть бути підключені аксесуари та динамік HomePod і декілька сумісно підключених об'єктів.

HomeKit також можна використовувати у режимі «вільні руки» завдяки голосовому помічнику Apple (Siri). Потім голос використовується як пристрій керування: користувач може дати Siri вказівки щодо увімкнення, вимкнення та налаштування будь-якого підключеного пристрою, який підключено та налаштовано в програмі Home.



Рисунок 1.1 – Вигляд пристрою Apple HomeKit Netatmo[9]

Переваги Apple HomeKit Netatmo:

- 1) великий перелік вбудованих функцій;
- 2) підтримка сучасних смартфонів;
- 3) сумісність з великою кількістю пристроїв автоматизації.

Недоліки:

- 1) немає підтримки операційної системи Android;
- 2) досить велика ціна.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						10
Зм.	Арк.	№докум.	Підпис	Дата		

1.2.2 Google Home

Google Home[10] – бездротовий динамік з вбудованим асистентом, розроблений в Google. Керування пристроєм відбувається за допомогою персонального асистента Google Assistant.



Рисунок 1.2 – Вигляд пристрою Google Home[10]

Переваги Google Home:

- 1) підтримує усі сучасні смартфони;
- 2) є можливість під'єднання сторонніх засобів автоматизації;
- 3) велика кількість готових функцій.

Недоліки:

- 1) повільна швидкість розпізнавання;
- 2) висока ціна.

1.2.3 Amazon Echo

Amazon Echo[11] – смарт-динамік розробки корпорації Amazon.com.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		11

Пристрій є 24-ох сантиметровий циліндричний динамік з вбудованим мікрофоном, який має вигляд семи-мікрофонного масиву. Пристрій керується голосом та реагує на ім'я «Алекса». Відразу після того як вимовлено це слово мова користувача записується і надсилається у «хмару» для аналізу та реакції, використовуючи функції проєкту Amazon Alexa – голосового помічника від Amazon.

Пристрій здатний до обмеженої взаємодії з користувачем, відтворення музики, створення списків завдань, встановлення будильників, відтворення аудіо книг та зачитування прогнозу погоди.



Рисунок 1.3 – Вигляд пристрою Amazon Echo[11]

Переваги пристрою Amazon Echo:

- 1) невелика ціна;
- 2) підтримка усіх сучасних смартфонів;
- 3) присутня можливість під'єднати сторонні засоби автоматизації.

Недоліки:

- 1) немає підтримки української та російської мов

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		12

1.2.4 Висновки на базі досліджень відомих реалізацій голосового асистента для кіберфізичної системи «Розумний будинок»

Проаналізувавши приклади реалізації відомих світових брендів було виявлено ряд недоліків, на які слід звернути уваги при реалізації проєкту. Найбільш вагомою проблемою є велика ціна пристрою та відсутність підтримки української та російської мов[12]. Також менш розповсюдженою але більш вагомою проблемою є відносно невелика швидкість розпізнавання та виконання команд.

1.3 Алгоритми розпізнавання голосу та мовлення

Розпізнавання мовлення (speech-to-text) – це здатність програми або машини визначати слова, які були вимовлені вголос, і перетворювати їх в текст, який можна прочитати. Просте програмне забезпечення для розпізнавання мовлення має досить обмежений словниковий запас та може розпізнавати слова чи фрази лише якщо вони промовляються чітко та повільно. Більш складне програмне забезпечення дає можливість обробляти природну мову, різні акценти та мови.

Розпізнавання голосу – це здатність машини або програми отримувати й інтерпретувати диктант або розуміти й виконувати голосні команди. Розпізнавання голосу набуло популярності та використання з появою AI та інтелектуальних помічників[13].

Розпізнавання мовлення та розпізнавання голосу – це дві різні технології.

Системи розпізнавання голосу дають змогу споживачам взаємодіяти з програмою, просто розмовляючи з нею. Це дозволяє забезпечити запити, нагадування та інші прості завдання не використовуючи рук.

Програмне забезпечення що реалізує розпізнавання голосу на комп'ютерах вимагає, щоб аналоговий аудіозапис був перетворений у цифрові сигнали. Ця технологія відома як аналого-цифрове перетворення. Щоб комп'ютер мав змогу розшифрувати сигнал, він повинен мати доступ до цифрової бази даних або словникового запасу слів або складів, а також достатньо швидкий засіб для

					КВРКІ 180204.18.02.25 ПЗ	Арк
						13
Зм.	Арк.	№докум.	Підпис	Дата		

порівняння цих даних та сигналів. Шаблони мовлення зберігаються на жорсткому диску та завантажуються в пам'ять під час запуску програми[14]. Компаратор перевіряє збережені шаблони на виході з аналого-цифрового перетворювача. Ця дія називається розпізнаванням шаблонів.

На практиці розмір словникового запасу програми розпізнавання голосу безпосередньо залежить від обсягу оперативної пам'яті комп'ютера, на якому вона встановлена. Програма для розпізнавання голосу працює у рази швидше, якщо весь її словниковий запас можна завантажити в оперативну пам'ять, у порівнянні з пошуком деяких збігів безпосередньо на жорсткому диску. Швидкість обробки також дуже важлива, оскільки вона впливає на те, наскільки швидко комп'ютер може шукати збіги в ОЗП.

Мова – послідовність звуків. Звук – накладання звукових хвиль різних частот. Приклад такого накладання зображено на рисунку 1.4. Хвиля характеризується двома атрибутами - частотою і амплітудою. Для збереження звукового сигналу на цифровому носії його необхідно розбити на велику кількість проміжків а потім взяти деяке усереднене значення на кожному із них. Графічне зображення цього етапу зображено на рисунку 1.5. Так механічні коливання перетворюються на набір чисел, який придатний для обробки на сучасних ЕОМ[15].

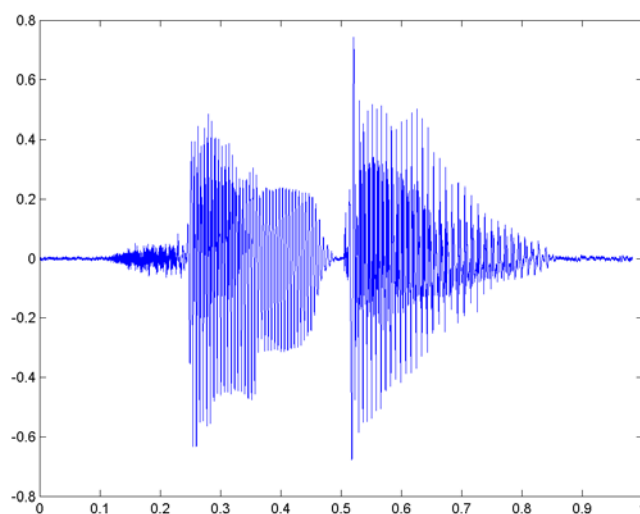


Рисунок 1.4 – Графічне зображення накладання звукових хвиль різних частот[12]

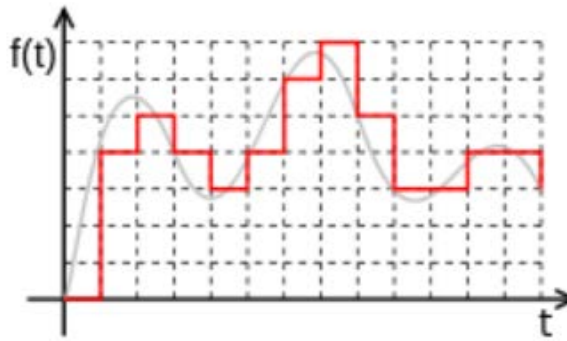


Рисунок 1.5 – Усереднені значення звукового сигналу на проміжках

Насамперед потрібно розбити дані на невеликі тимчасові проміжки – фрейми. Фрейми не повинні йти строго один за одним, а "внахлест". Це означає, що кінець одного фрейму повинен перетинатися із початком іншого.

При розпізнаванні мови першим завданням, яке доводиться вирішувати, є розбиття мови на окремі слова. Найпростіше це зробити якщо між словами є деякі паузи (проміжки тиші). Їх можна вважати розділювачами слів[16].

У такому разі потрібно знайти деяке значення, або поріг. Це значення вище за яке є слово а нижче – тиша. Є кілька варіантів реалізації:

- 1) кластеризувати значення і тим самим явно виділивши безліч значень які відповідають тиші. Це спрацює тільки якщо тиша займає досить велику частину вихідного сигналу;
- 2) задати константою. Спрацює лише тоді, коли вихідний сигнал завжди генерується за одних і тих самих умов та одним і тим самим способом;
- 3) проаналізувати ентропію. У цьому випадку ентропія означає те, з якою силою коливається сигнал у рамках заданого фрейму.

Для підрахунку ентропії конкретного фрейму потрібно знайти щільність розподілу значень сигналів даного кадру та розрахувати ентропію за формулою

$$E = \sum_{i=0}^{N-1} P(i) * \log_2 (P[i]). \quad (1.1)$$

MFCC – це своєрідне представлення енергії спектра сигналу. Плюси його використання полягають у наступному[17]:

- 1) використовується спектр сигналу, який дає можливість враховувати хвильову природу сигналу при подальшому аналізі;
- 2) спектр проектується на спеціальну mel-шкалу, що дозволяє виділити найбільш значущі частоти для сприйняття людиною;
- 3) кількість коефіцієнтів може бути обмежена будь-яким значенням що дає змогу стиснути кадр і кількість оброблюваної інформації.

1.4 Огляд апаратних складових для реалізації голосового асистента для кіберфізичної системи «Розумний будинок»

1.4.1 Arduino Nano

Arduino Nano [18] – це класична плата мінімальних розмірів від Arduino. Arduino Nano поставляється з штифтовими роз'ємами, які дозволяють легко прикріпити її до плати. Також вона оснащена роз'ємом Mini-B USB.

Зовнішній вигляд Arduino Nano зображено на рисунку 1.6, а її розпинівка на рисунку 1.7.

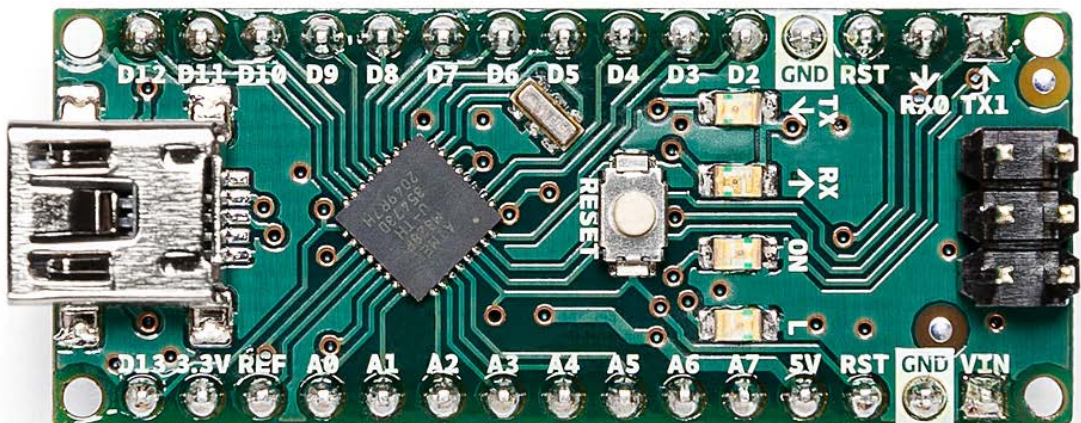
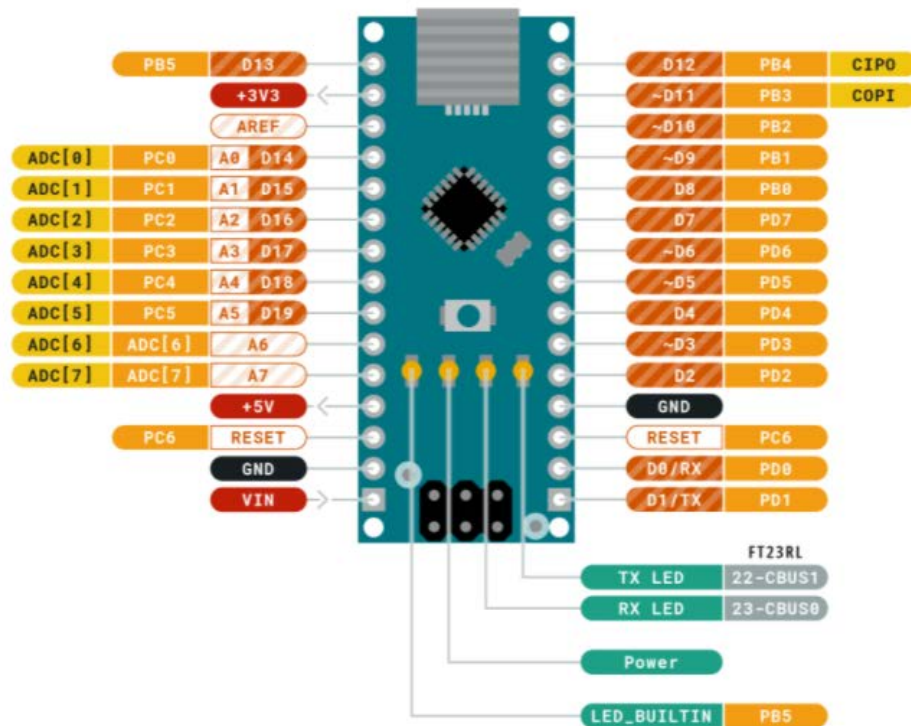


Рисунок 1.6 – Зовнішній вигляд плати Arduino Nano[19]

Основні характеристики плати Arduino Nano:

- 1) мікроконтролер ATmega328. Процесор ATmega328 працює з тактовою частотою 16 МГц і має 32 КБ флеш-пам'яті (з яких 2 КБ використовується завантажувачем);
- 2) маленький розмір плати. Маючи довжину 45 мм і ширину 18 мм, Nano є найменшою платою Arduino і важить всього 7 грам;
- 3) зроблено для макетної плати. Nano призначена для використання в макетній платі та має припаяні роз'єми для всіх контактів, що дозволяє легко прикріпити плату до будь-якої макетної плати[20-21].

Технічні характеристики плати Arduino Nano наведено в таблиці 1.1.



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO . CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1886, Mountain View, CA 94042, USA.

Рисунок 1.7 – Розпинівка плати Arduino Nano

Таблиця 1.1 – Технічні характеристики Arduino Nano

Плата	Ім'я	Arduino Nano
	Артикул	A000005
Мікроконтролер	ATmega328	
USB роз'єм	Mini-B USB	
Піни	Вбудований світлодіодний	13
	Цифрові вводу та виводу	14
	Виводи аналогового входу	8
	PWM	6
Комунікація	UART	+
	I2C	+
	SPI	+
Напруга	Введення і виведення	5В
	Вхідна (номінальна)	7-12В
	Постійний струм на контакт введення/виводу	20мА
Тактова частота	ATmega328	16МГц
Пам'ять	ATmega328P	2 КБ SRAM, 32 КБ flash 1 КБ EEPROM
Розміри	Вага	5гр
	Ширина	18мм
	Довжина	45мм

1.4.2 Цифрові та аналогові фільтри. IIR-фільтр другого порядку

Цифровий фільтр – будь який фільтр, який обробляє цифровий сигнал з метою відокремити та/або придушити певні частоти складових даного сигналу. На відміну від аналогового, цифровий фільтр має справу з цифровим а не аналоговим

сигналом. Також його властивості дискретні. У аналоговому фільтрі передаточна функція залежить від внутрішніх властивостей його складових елементів[23].

Цифрові фільтри зараз використовуються практично всюди, де потрібно обробити сигнал. Прикладом цього може стати спектральний аналіз, обробка зображень, відео, звуку та мови.

Перевагами цифрових фільтрів перед аналоговими можна вважати наступне:

- 1) висока точність фільтрації;
- 2) передавальна функція не залежить від дрейфу характеристик елементів;
- 3) легкість внесення змін та гнучкість налаштування;
- 4) компактність складових фільтру.

Натомість цифрові фільтри також мають декілька недоліків у порівнянні з аналоговими. А саме:

1) важкість роботи з високочастотними сигналами. Смуга частот є обмежена частотою Найквіста, яка рівна половині частоти дискретизації сигналу. Тому для високочастотних сигналів застосовуються аналогові фільтри, або, якщо на високих частотах немає корисного сигналу, спочатку знижують високочастотні складові за допомогою аналогового фільтру а потім обробляють сигнал цифровим фільтром[24];

2) важкість роботи в реальному часі. Обчислення повинні бути закінченими під час періоду дискретизації;

3) для більш високої швидкості обробки сигналів та точності потрібен потужний процесор та додаткове апаратне забезпечення у вигляді високоточних АЦП та ЦАП.

Однією із найбільш використовуваних форм фільтрів є біквэд. Біквэд - це ІІР-фільтр другого порядку (два полюси і два нулі). Цей фільтр має досить високий порядок, тому сам по собі може бути дуже корисним в більшості випадків. Але через коефіцієнт чутливості у фільтрах вищого порядку, біквэд часто використовують як базовий блок для більш складних фільтрів. Наприклад, двоквадратичний фільтр нижніх частот має нахил зрізу 12 дБ/октаву, що корисно для контролю тембру. Якщо в роботі потрібен нахил 24 дБ/октава, можна

					КВРКІ 180204.18.02.25 ПЗ	Арк
						19
Зм.	Арк.	№докум.	Підпис	Дата		

підключити два біквади, що матиме менше проблем з коефіцієнтом чутливості, ніж один проект четвертого порядку.

Біквади бувають декількох видів. Найбільше очевидною є пряма реалізація різницевого рівняння другого порядку ($y[n] = a_0*x[n] + a_1*x[n-1] + a_2*x[n-2] - b_1*y[n-1] - b_2*y[n-2]$), що називається прямою формою I. Приклад прямої форми I зображено на рисунку 1.8[25].

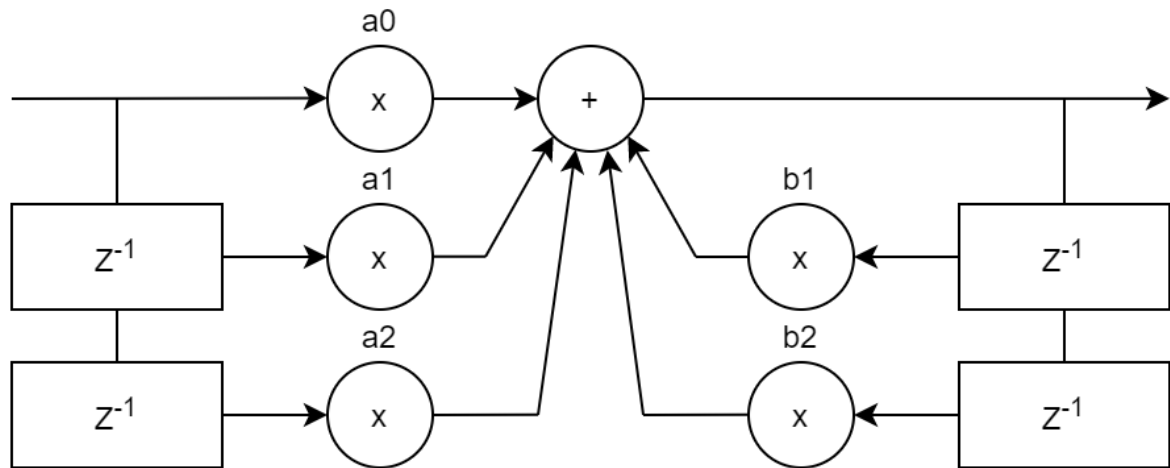


Рисунок 1.8 – Пряма форма I

Пряма форма I являється найкращим видом для реалізації в процесорі з фіксованою точкою, так як вона має одну точку підсумування (ЦСП з фіксованою точкою зазвичай оснащені розширеним акумулятором, який дозволяє здійснювати проміжні переповнення).

Пряму форму I можна розділити в точці її підсумування. Після цього слід узяти дві отримані половинки і поміняти їх місцями так, щоб половина зворотного зв'язку була першою.

Це зображено на рисунку 1.9.

Після проведених маніпуляцій одна пара затримок z являється надлишковою, зберігаючи ту саму інформацію що й інша.

Отже ці дві пари можна об'єднати в одну[27]. Після цього утвориться пряма форма II, яка зображена на рисунку 1.10.

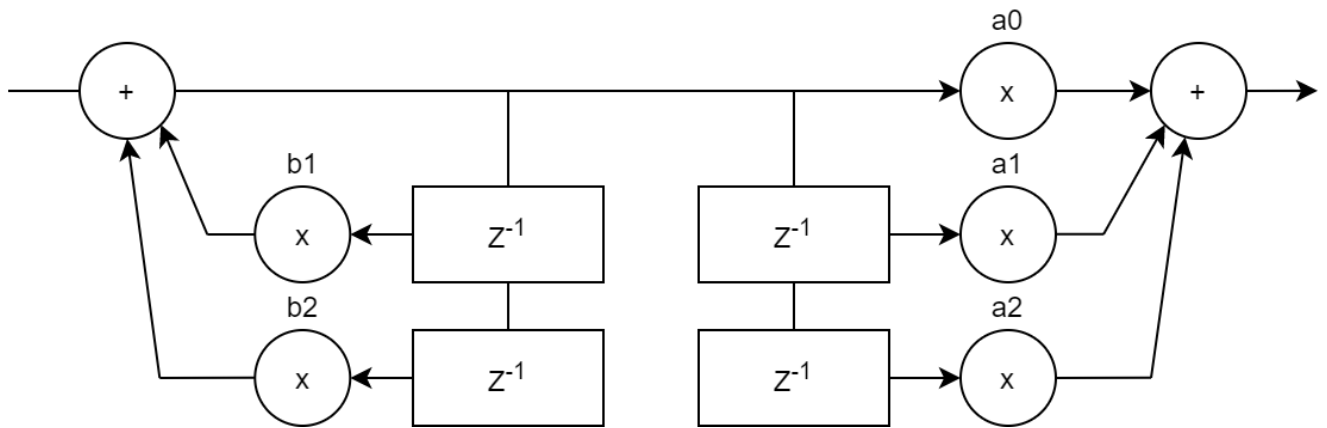


Рисунок 1.9 – Пряма форма I після розділення та розвороту

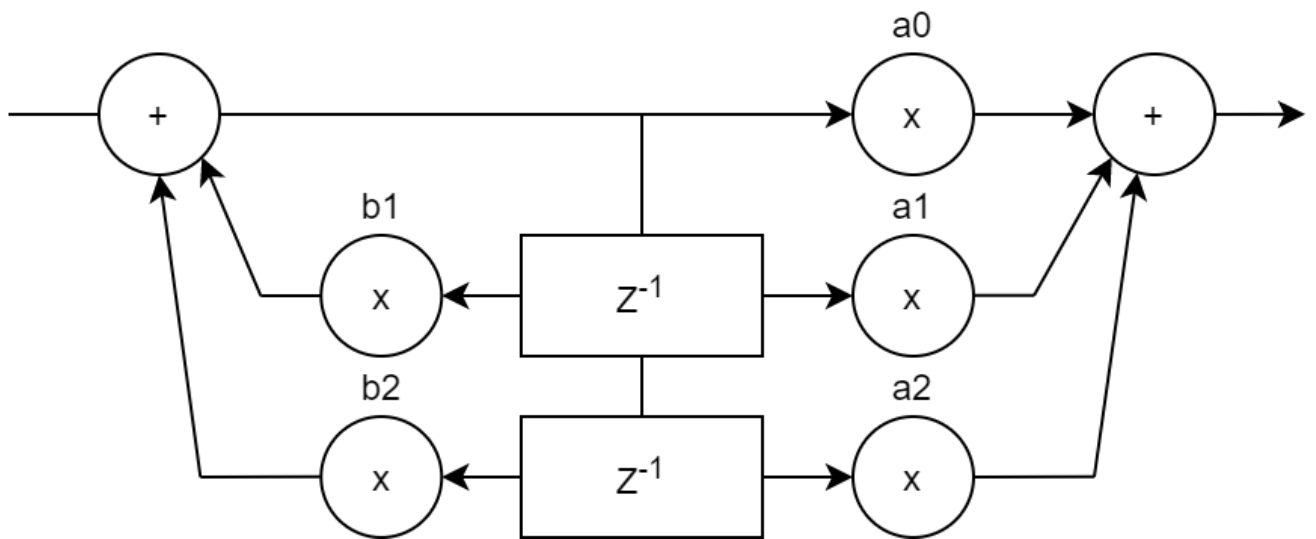


Рисунок 1.10 – Пряма форма II

II-га форма з плаваючою крапкою є кращою, так як вона зберігає два місця у пам'яті, а плаваюча крапка не є чутливою до переповнення, як математика з фіксованою крапкою.

Це може бути покращено за допомогою транспонування фільтру. Для того щоб це зробити потрібно повернути напрямок потоку сигналу на протилежний, вихід стає входом.

Характеристики фільтру при цьому не зміняться, але при такому розподілі іноді буває, що характеристики з плаваючою крапкою стають трохи кращими. Число з плаваючою крапкою має більш кращу точність, коли проміжні суми мають ближче значення (додавання малих чисел до числа з плаваючою крапкою являється

менш точним, ніж з подібними значеннями)[28]. Транспонована пряма форма II зображена на рисунку 1.11.

На низьких частотах біквади більш сприйнятливі до помилок квантування. В основному це відбувається через коефіцієнти зворотного зв'язку b_1 та b_2 та пам'ять затримки. Відсутність роздільної здатності в коефіцієнтах ускладнює точне розташування полюсів, що являється проблемою, коли полюси розташовані біля одиничного кола.

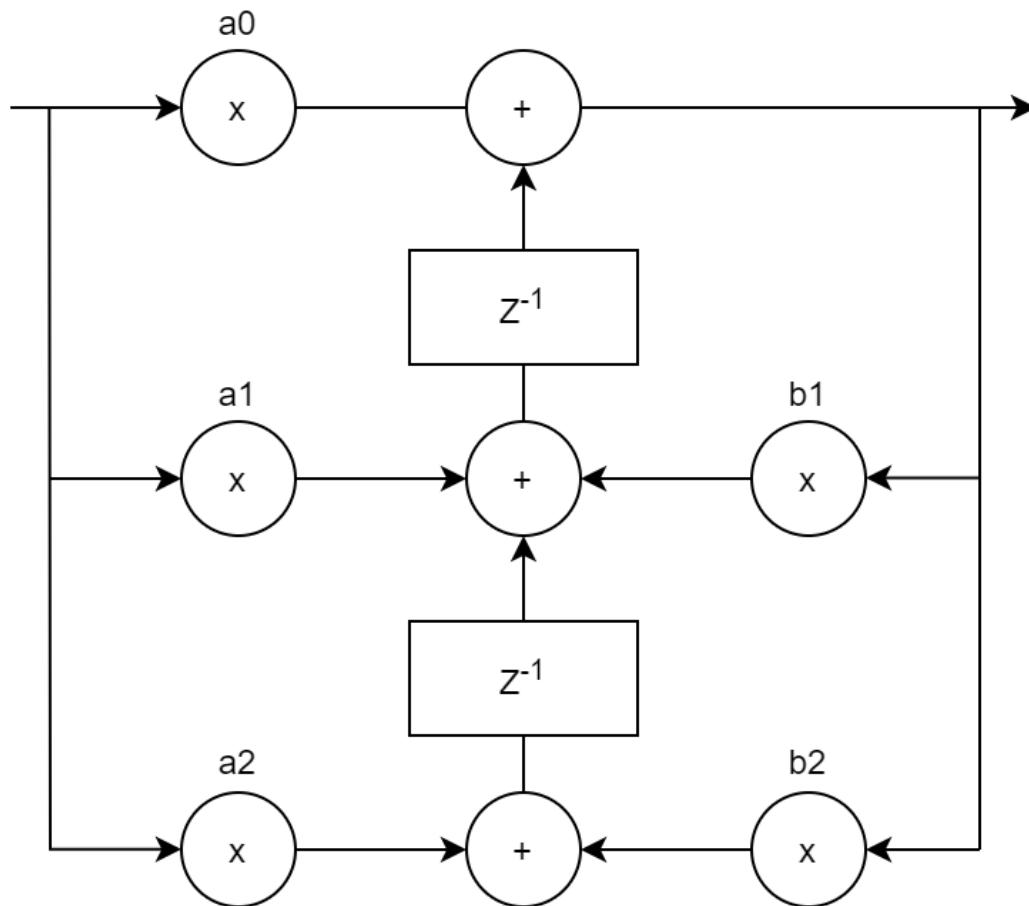


Рисунок 1.11 – Пряма форма II транспонована

Інша проблема полягає в пам'яті із затримкою, а саме в тому що множення генерує більше бітів, а біти обрізаються під час зберігання в пам'яті. Ця помилка квантування повертається до фільтру, викликаючи нестабільність. 32-бітової плаваючої крапки зазвичай достатньо для аудіофільтрів, але може знадобитися

використовувати подвійну точність, особливо на дуже низьких частотах (для контрольної фільтрації) і при високих частотах дискретизації.

Для фільтрів з фіксованою крапкою 24-бітові коефіцієнти та пам'ять добре працюють для більшості фільтрів, але починають ставати нестабільними нижче приблизно 300 Гц при частоті дискретизації 48 кГц (або вдвічі більше при частоті дискретизації 96 кГц)[30]. Подвійна точність завжди дорого коштує для процесора з фіксованою крапкою, але, існує проста техніка для підвищення стабільності. Квантування відбувається, коли акумулятор більш високої точності зберігається в пам'яті затримки з нижчою точністю з правого боку. Беручи помилку квантування (різницю між повним накопиченим значенням і його значенням після збереження його в пам'яті) і додаючи його назад для наступного вибіркового обчислення, фільтр працює майже так само добре, як і використовує обчислення з повною подвійною точністю, але зі значною менші обчислювальні витрати. Цей метод називається формуванням шуму першого порядку, його зображено на рисунку 1.12.

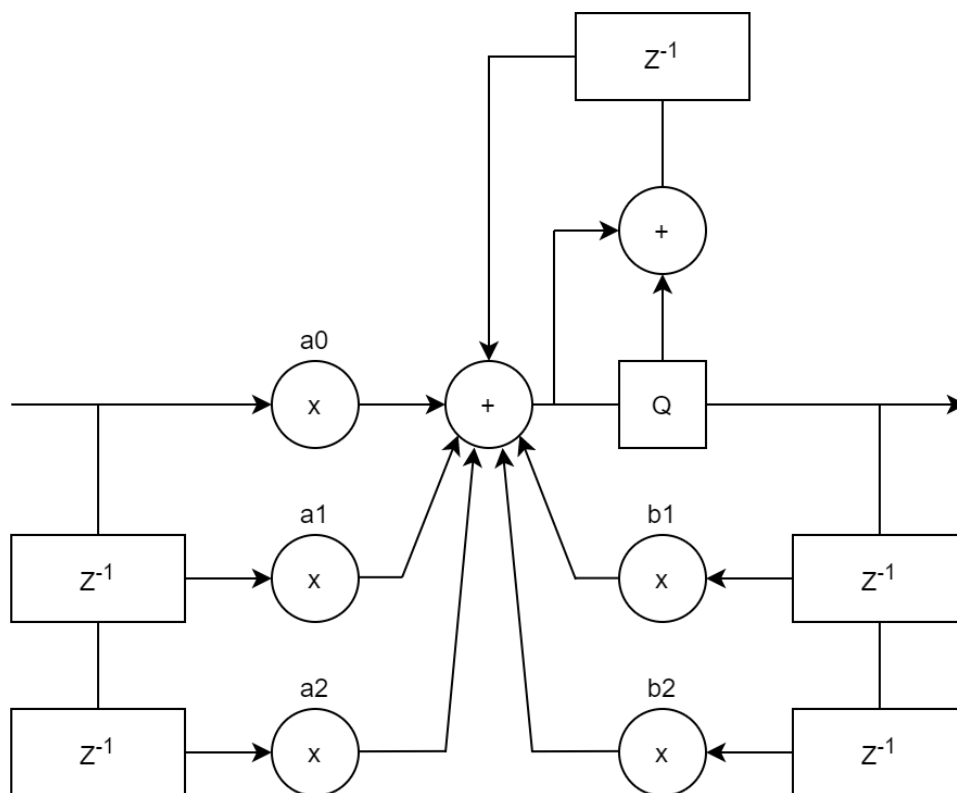


Рисунок 1.12 – Пряма форма I з формуванням шуму першого порядку

Зм.	Арк.	№докум.	Підпис	Дата

Загалом, 16-розрядна обробка з фіксованою крапкою не підходить для аудіо без подвійних коефіцієнтів точності та обчислень.

Нарешті, біквади — це лише один із інструментів програмістів DSP — вони не завжди є найкращою формою фільтра. Існують інші фільтри, які не поділяють низькочастотну чутливість біквдрату (загалом точність біквдрного коефіцієнта дуже хороша на високих частотах і погана на низьких[31]. Є інші форми фільтрів, які розподіляють точність більш рівномірно або обмінюються вимкнено знижена високочастотна продуктивність для кращої низькочастотної продуктивності). Тим не менш, біквади добре відомі, а інструментів для проектування багато, тому вони зазвичай є першим вибором для фільтра ІІР.

Ще одна форма фільтра – це фільтр змінної стану . Він має дуже чудову продуктивність на низьких частотах та обмеження на високих частотах, які необхідно обійти, але, що найголовніше, коефіцієнти частоти та добротності є окремими та їх легко змінити для динамічної фільтрації. Це також чудовий генератор синусоїди низької частоти.

1.5 Постановка задачі

Дослідивши велику кількість матеріалу на задану тему впливає чіткий порядок дій для програмно-технічної реалізації голосового асистента для кіберфізичної системи «Розумний будинок»:

- 1) підібрати апаратні складові для реалізації голосового асистента для кіберфізичної системи «Розумний будинок»
- 2) встановити апаратні з'єднання. На цьому етапі потрібно підключити модуль МАХ9814, мікрофони та динаміки до плати Arduino Nano;
- 3) написати алгоритм збору даних з АЦП Arduino Nano для подальшої обробки;
- 4) налаштувати алгоритм дії цифрового фільтру діапазону частот та розрахувати час виконання обчислень;
- 5) розрахувати коефіцієнти цифрового фільтру діапазону частот;

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		24

- 6) створити алгоритм обробки шаблонів висловлювань;
- 7) налаштувати алгоритм використання динамічного викривлення часу;
- 8) провести навчання розпізнавання шаблонів мовлення;
- 9) створити алгоритм розпізнавання мовлення на базі результатів попередніх кроків;
- 10) встановити з'єднання між розробленою програмно-технічною реалізацією голосового асистента та кіберфізичною системою «Розумний будинок»;
- 11) розробити програмне забезпечення для обробки команд та передачі їх до компонентів кіберфізичної системи «Розумний будинок»;
- 12) протестувати програмно-технічну реалізацію голосового асистента для кіберфізичної системи "Розумний будинок".

На основі даного переліку задач буде здійснено розробку голосового асистента для кіберфізичної системи «Розумний будинок».

1.6 Висновки

У першому розділі було розглянуто поняття інтернету речей та технології керування кіберфізичною системою «Розумний будинок». Також було досліджено відомі реалізації голосового асистента, такі як: Apple HomeKit, Google Home, Amazon Echo.

На основі досліджень було виявлено переваги та недоліки кожного з них та задано перспективи для майбутньої розробки. При цьому було розглянуто алгоритми розпізнавання голосу, які будуть використовуватись при створенні розумного асистента. Також було проведено огляд апаратних складових (зокрема їх характеристик), таких, як плата Arduino Nano та цифрових аналогових ПР фільтрів.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						25
Зм.	Арк.	№докум.	Підпис	Дата		

2. ПРОЄКТУВАННЯ ГОЛОСОВОГО АСИСТЕНТА ДЛЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

2.1 Обґрунтування вибору програмного забезпечення та апаратних ресурсів

Для нормального функціонування голосового асистента потрібно щоб у будь-якій точці будинку, де потрібно взаємодіяти з системою розумного будинку, був розміщений мікрофон та динамік. Мікрофон потрібен для запису голосу користувача і подальшого розпізнавання команди. Динаміки ж потрібні для того, щоб давати користувачеві зворотній зв'язок. Наприклад, якщо користувач скаже «увімкни кондиціонер» відбудеться наступний порядок дій:

- 1) мікрофон записує фразу, сказану користувачем;
- 2) відбувається розпізнавання голосу з запису;
- 3) якщо команда розпізнана, сигнал передається до пристрою, до якого вона має відношення (до кондиціонеру);
- 4) прилад (кондиціонер) виконує команду, яка поступила (вмикається);
- 5) сигнал того, як команда відпрацювала передається до системи;
- 6) через динаміки відтворюється звуковий запис, який дає змогу зрозуміти про стан системи. В даному випадку користувач почує «Кондиціонер увімкнено».

Для керування голосом часто використовують Arduino Voice Recognition. Він являється фірмовим пристроєм для розпізнавання голосових команд від компанії Arduino. Серед його недоліків є те, що є обмеження за кількістю команд та необхідність перепрошивати пристрій для додавання нових.

Схему голосового управління за допомогою Arduino Voice Recognition зображено на рисунку 2.1.

На рисунку 2.2 зображено схему підключення Arduino Voice Recognition на реальному прикладі.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		26

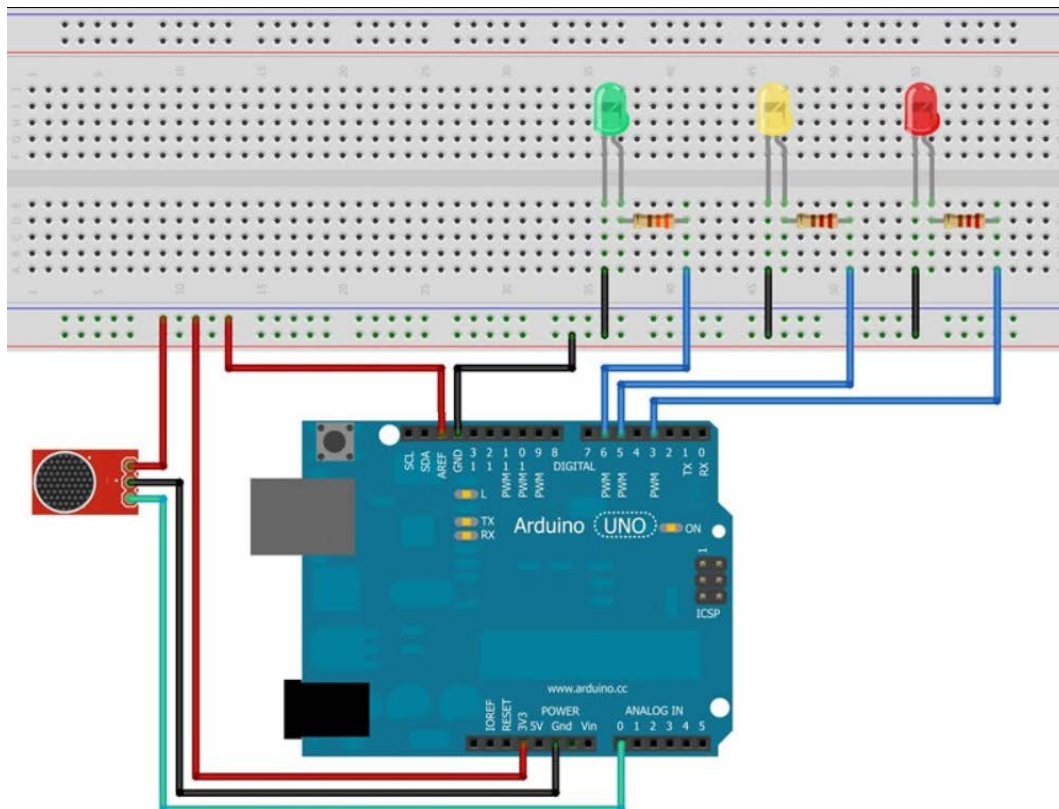


Рисунок 2.1 – Схема голосового управління розумним будинком за допомогою пристрою Arduino Voice Recognition

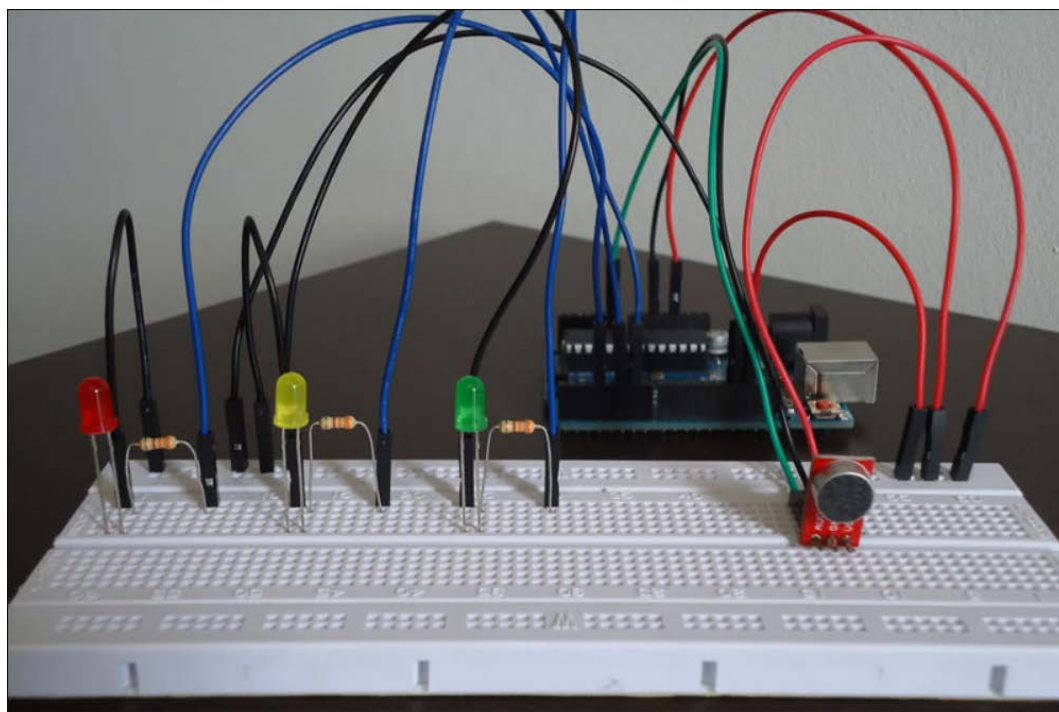


Рисунок 2.2 – Схема підключення Arduino Voice Recognition

Зм.	Арк.	№докум.	Підпис	Дата

КВРКІ 180204.18.02.25 ПЗ

Арк

27

Для реалізації розпізнавання голосу також може підійти плата Arduino Nano.

Також для реалізації знадобиться модуль MAX9814. Пристрій включає малошумний попередній підсилювач, підсилювач із змінним посиленням (VGA), вихідний підсилювач, мікрофонний генератор напруги зміщення та схему керування AGC. Попередній підсилювач із низьким рівнем шуму має фіксоване посилення 12 дБ, тоді як коефіцієнт посилення VGA автоматично регулюється від 0 дБ до 20 дБ, залежно від вихідної напруги та порогу AGC.

Вихідний підсилювач надає вибір посилення 8 дБ, 18 дБ та 28 дБ. Без стиснення каскад підсилювачів призводить до загального посилення 40 дБ, 50 дБ або навіть 60 дБ[31]. Трирівневий цифровий вхід програмує посилення вихідного підсилювача. На рисунку 2.3 зображено зовнішній вигляд цього модуля.

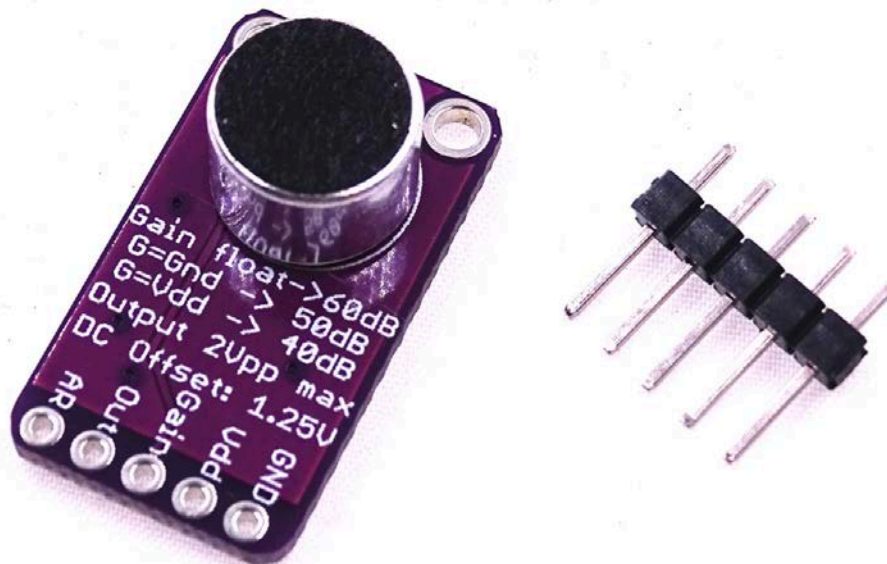


Рисунок 2.3 – Зовнішній вигляд модуля MAX9814

Основним у роботі голосового асистента є запис голосових команд користувача. Для цього до системи повинен бути під'єднаний хоча б один записуючий пристрій. Так, як робота голосового асистента передбачається в кіберфізичній системі «Розумний будинок», знадобиться декілька таких пристроїв. Принаймні по одному має розташовуватись в кожній кімнаті будинку.

Для запису голосу користувача був вибраний мікрофон Marshall Electronics MXL AC-404. Це конденсаторний мікрофон з дротовим підключенням добре підходить для поставленої задачі, оскільки має невеликі габарити (109 x 76 x 51 мм) та може записувати звук з будь-якої сторони однаково добре. Високопродуктивний три капсульний мікрофон забезпечує покриття на 180°. MXL AC-404 записує звук в діапазоні від 40 до 16 000 Гц, що також цілком задовольняє потреби проєкту. Також перевагою цього мікрофону над аналогами є характеристики АЦП: розрядність – 16 біт та частота дискретизації 48 кГц[33].

Даний мікрофон був створений для запису конференцій, тому має кріплення на стіл. Але це кріплення може бути використане для кріплення на стіну, що естетично впишеться в концепцію кіберфізичної системи «Розумний будинок». Зовнішній вигляд мікрофону зображено на рисунку 2.4.



Рисунок 2.4 – Зовнішній вигляд мікрофону Marshall Electronics
MXL AC-404

Також для роботи з голосовим асистентом потрібно розуміти чи була команда сприйнята правильно та мати так званий «зворотній зв’язок» з системою.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		29

Для цього потрібно підключити декілька динаміків, через які буде виводитись звукова інформація. Кількість динаміків має бути приблизно рівна кількості мікрофонів для обхвату усіх кімнат у будинку.

Для даних цілей була вибрана настінна акустична система SKY SOUND PM-2401TB, яка являється настінним двох смуговим гучномовцем. Даний гучномовець призначений для трансляції музичного супроводу та голосового оповіщення. Це є недорогим та стильним рішенням поставленої задачі, а такі характеристики, як частотний діапазон (40 – 20 000 Гц), чутливість (96 Дб), потужність (10-40 Вт) та опір (8 Ом) цілком підходять для використання у кіберфізичній системі «Розумний будинок»[34]. Зовнішній вигляд даної акустичної системи зображено на рисунку 2.5.

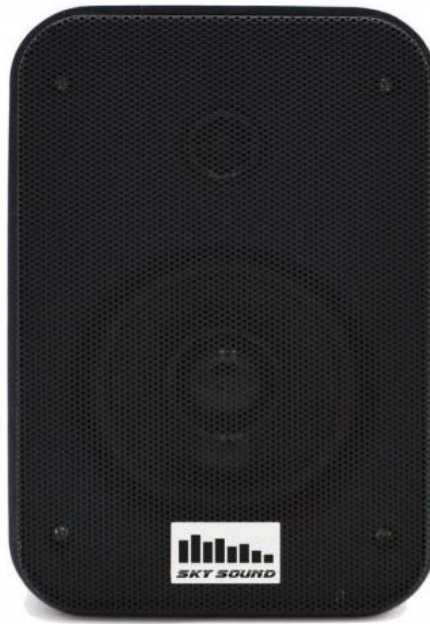


Рисунок 2.5 – Зовнішній вигляд акустичної системи SKY SOUND PM-2401TB

Для розпізнавання команд та виконання конкретних дій за інструкціями користувача потрібно створити окреме програмне забезпечення. Для цього була вибрана така мова програмування, як Python найактуальнішої на момент розробки версії 3.10.

Дана мова програмування була вибрана по декільком причинам:

					КВРКІ 180204.18.02.25 ПЗ	Арк
						30
Зм.	Арк.	№докум.	Підпис	Дата		

- 1) підвищена продуктивність. Завдяки простоті синтаксису не потрібно витрачати багато часу на його розуміння та можна зосередитися на роботі;
- 2) інтерпретована мова. Python буде виконувати код рядок за рядком та, при виникненні помилки, зупинить програму та повідомить про неї;
- 3) безкоштовна мова з відкритим кодом;
- 4) динамічна типізація. Не потрібно перейматися за призначення типів змінним, при їх оголошенні;
- 5) підтримка великої кількості бібліотек. Можна знайти бібліотеку з уже готовим рішенням певної задачі замість витрачання часу на її написання;
- 6) портативність. Не потрібно перейматися при переміщенні коду на різні платформи.

2.2 Програмна реалізація підключення мікрофону до плати та тестування його роботи

Для підключення мікрофону до плати та подальшого тестування потрібно знати декілька основних команд. Вони наведені нижче.

Слід використати `static const char channels = 1;` для встановлення кількості вхідних каналів.

`static const int frequency = 16000` встановлює частоту дискретизації на 20 кГц.
`short sampleBuffer[512]` – буфер для зчитування зразків, кожен з яких має 16 біт.

Для друку зразку на послідовний монітор слід використати `Serial.print(sampleBuffer[i])`

Після успішного завантаження коду на плату Arduino Nano потрібно відкрити послідовний монітор для ініціалізації програми.

Відкривши його, буде видно, що дані друкуються швидко. Це зразки аудіо, які записуються з мікрофона. Це можна побачити на рисунку 2.6.

Також можна відобразити ці дані у вигляді графіку. Даний графік зображено на рисунку 2.7.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		31

За допомогою зібраних даних при тестуванні роботи мікрофону в подальшому можна буде визначити рівень фонового шуму для встановлення границі, при досягненні гучності вище якої, звук буде рахуватись висловлюванням користувача.

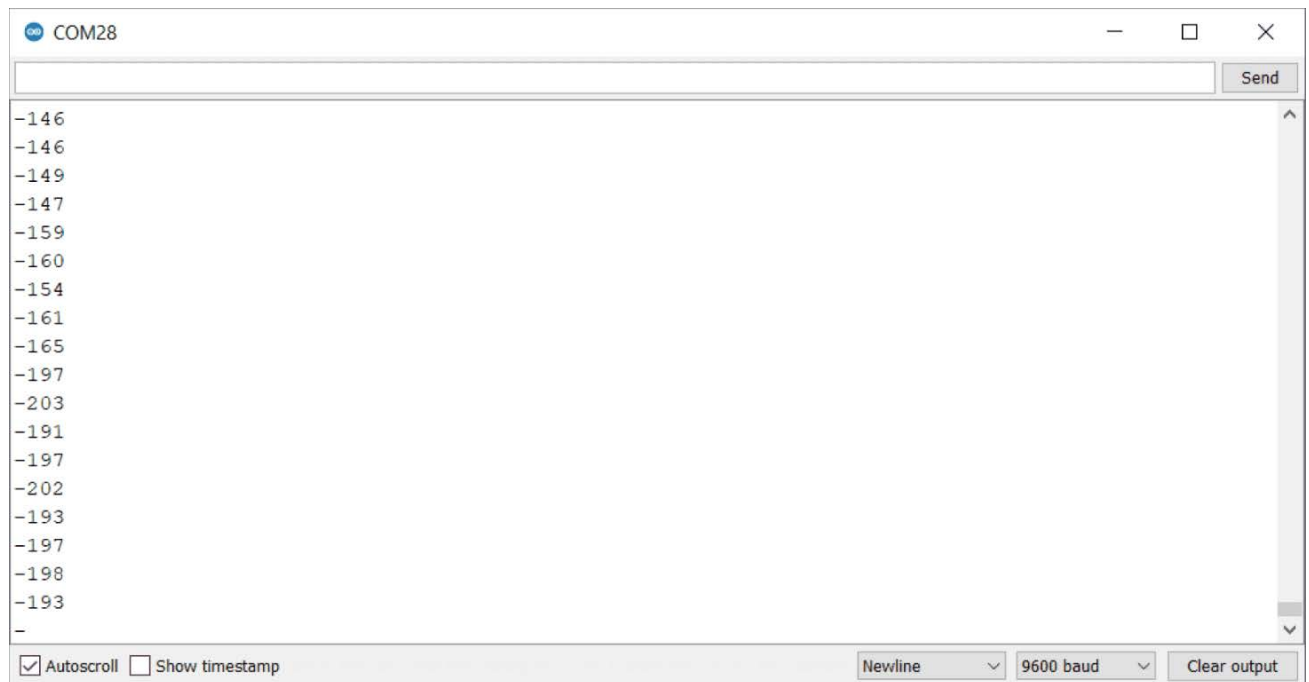


Рисунок 2.6 – Дані з мікрофону, підключеного до Arduino Nano

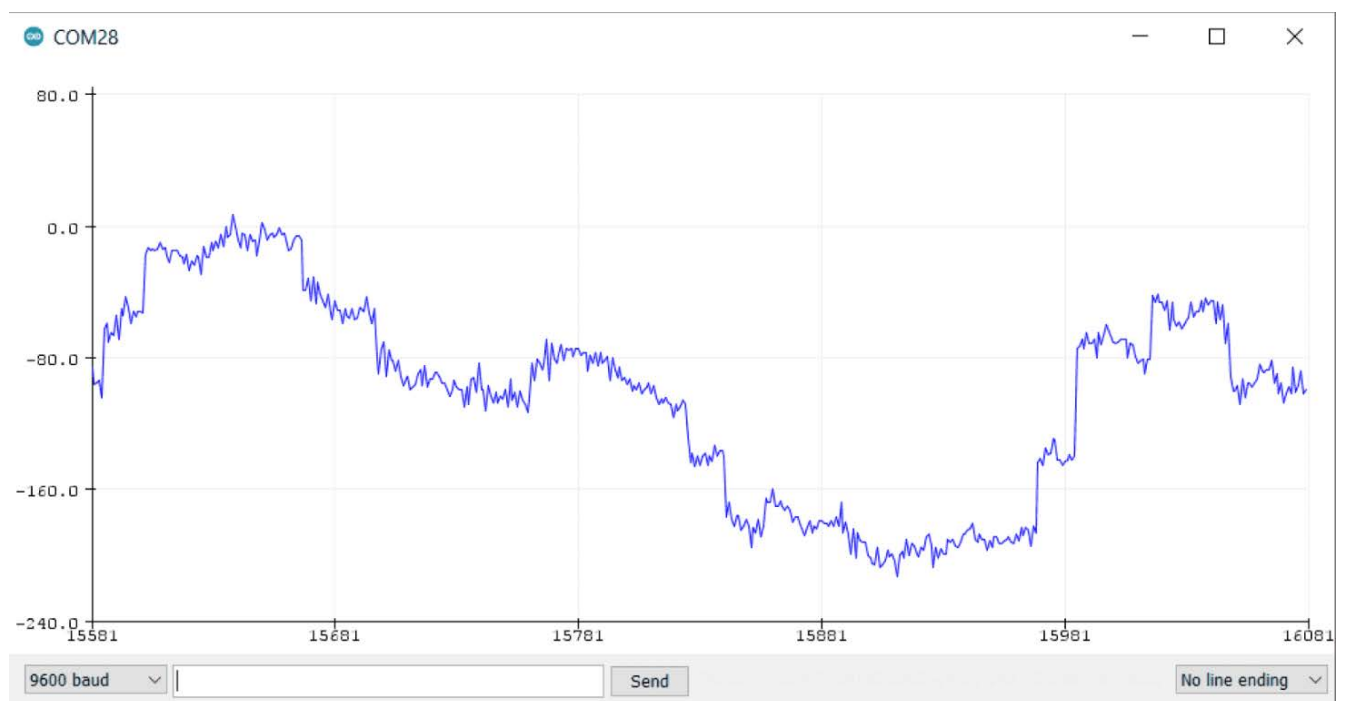


Рисунок 2.7 – Дані з мікрофону, підключеного до Arduino Nano, у вигляді графіку

Зм.	Арк.	№докум.	Підпис	Дата

КВРКІ 180204.18.02.25 ПЗ

Арк

32

2.3 Взаємодія голосового асистента з кіберфізичною системою «Розумний будинок»

Основною метою розробки голосового асистента для кіберфізичної системи «Розумний будинок» є керування приладами, які входять до цієї системи. Усі висловлювання, розпізнані за допомогою Arduino Nano передаються на комп'ютер, на якому будуть знайдені команди для певних компонентів та передані безпосередньо до них. За допомогою голосових команд можна буде керувати такими компонентами системи, як:

- 1) вентиляція (відкриття та закриття вентиляційних шахт, ввімкнення вентилятора);
- 2) освітлення (ввімкнення та вимкнення світла в певній кімнаті або її частині);
- 3) музичні прилади (ввімкнення або вимкнення музики);
- 4) системи безпеки (управління камерами, розумними замками та сигналізацією);
- 5) телебачення (увімкнення та вимкнення телебачення або стрімінгових сервісів, перемотка та керування відео);
- 6) опалення (увімкнення або вимкнення опалення);
- 7) інтернет (пошук інформації в мережі Інтернет);
- 8) кухонне приладдя (управління чайником, плитою, холодильником, мікрохвильовою піччю, тощо);
- 9) кондиціонер (ввімкнення та вимкнення кондиціонеру);
- 10) розумні жалюзі (відкриття та закриття розумних жалюзів);
- 11) розумні розетки (управління приладами, які до них підключені).

Взаємодія голосового асистенту з компонентами кіберфізичної системи «Розумного будинку» зображено на рисунку 2.8.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		33

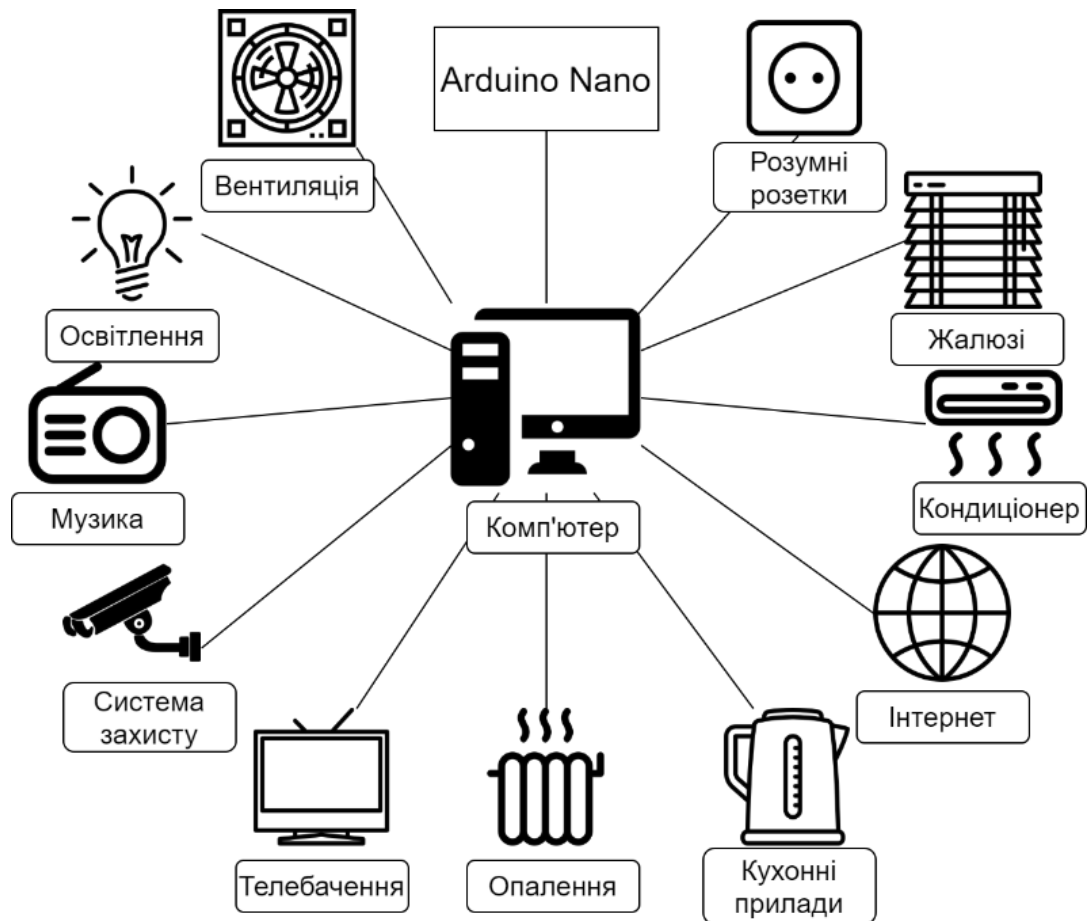


Рисунок 2.8 - Взаємодія голосового асистенту з компонентами кіберфізичної системи «Розумного будинку»

2.5 Висновки

В другому розділі проводилось обґрунтування вибору програмного забезпечення та апаратних ресурсів в залежності від поставленої задачі. Також на даному етапі було складено список дій, які потрібно зробити для реалізації голосового асистента для кіберфізичної системи «Розумний будинок».

Після цього було проведено підключення мікрофону до плати та його тестування. Також у другому розділі була побудована приблизна модель взаємодії голосового асистента з елементами системи «Розумний будинок».

Зм.	Арк.	№докум.	Підпис	Дата

3. ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ ГОЛОСОВОГО АСИСТЕНТА ДЛЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

3.1 Встановлення апаратних з'єднань для реалізації голосового асистента

Для початку потрібно підключити модуль MAX9814 до плати Arduino Nano.

Модуль MAX9814 має 5 контактів з маркуванням:

- 1) GND 0В;
- 2) VDD 5В;
- 3) GAIN;
- 4) OUT до аналогового контакту Nano;
- 5) AR.

Схематичне та реальне з'єднання модуля MAX9814 з платою Arduino Nano показано на рисунку 3.1 та 3.2 відповідно.

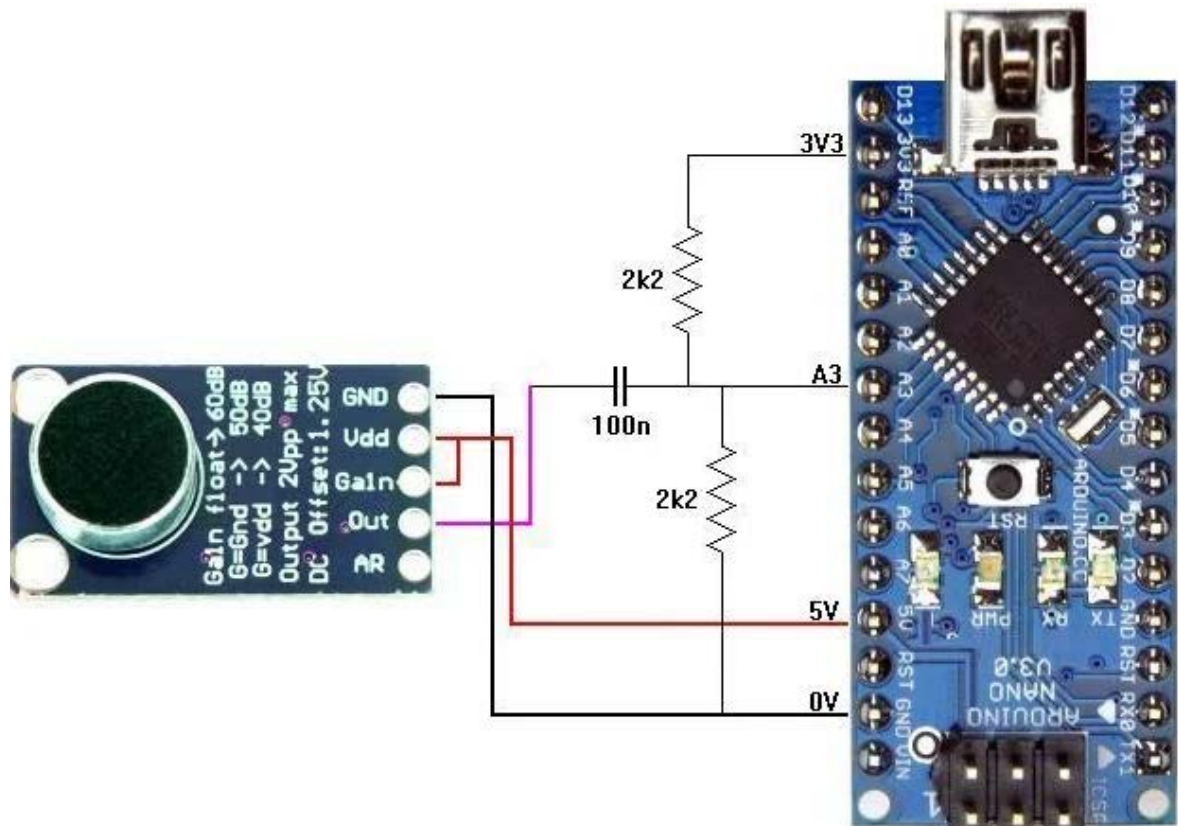


Рисунок 3.1 – Схематичне з'єднання модуля MAX9814 з платою Arduino Nano

Зм.	Арк.	№докум.	Підпис	Дата

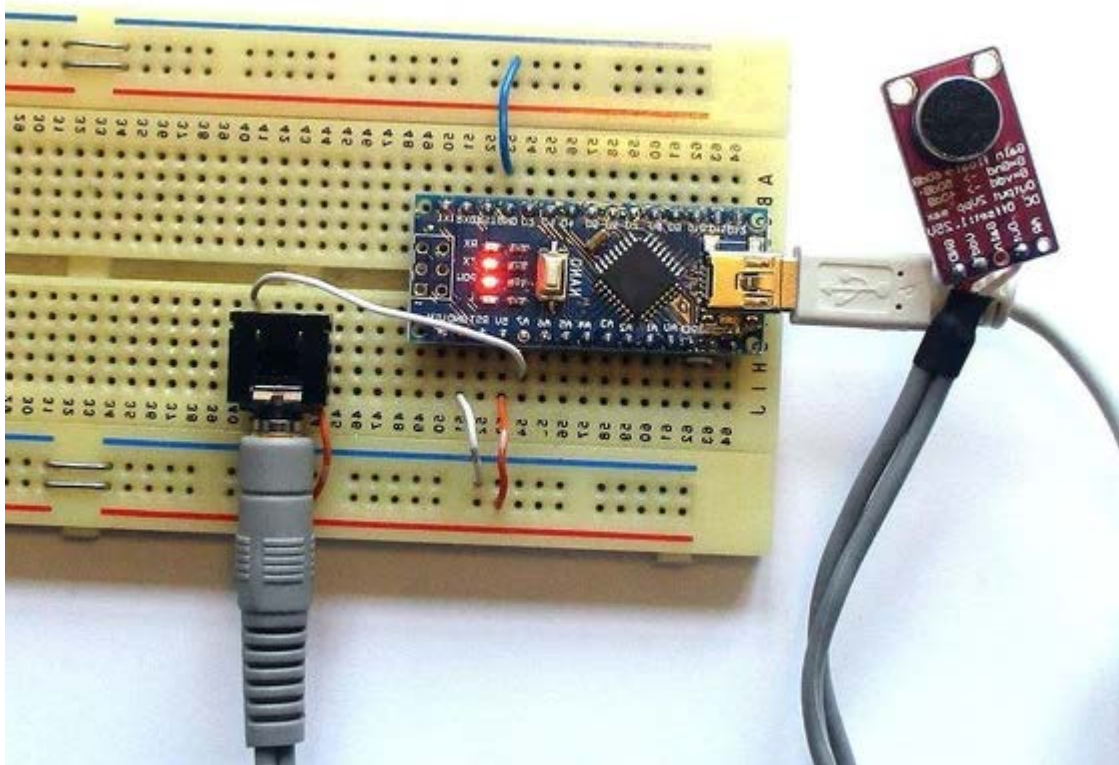


Рисунок 3.2 –З'єднання модуля MAX9814 з платою Arduino Nano

Пін A/R керує "Коефіцієнтом атаки та вивільнення" автоматичного регулювання посилення:

- 6) A/R = GND: співвідношення атаки до вивільнення становить 1:500;
- 7) A/R = VDD: співвідношення атаки до вивільнення становить 1:2000;
- 8) A/R = NonConnect: співвідношення атаки до вивільнення становить 1:4000.

Фактичний час атаки та звільнення встановлюється конденсатором на модулі.

Час атаки = $2,4 * C$ (час у мс, C в мкФ)

Вивід Gain керує підсиленням AGC:

- 9) GAIN = GND, посилення встановлено на 50 дБ;
- 10) GAIN = VDD, посилення встановлено на 40 дБ;
- 11) GAIN = NonConnect, нестиснений підсилення встановлено на 60 дБ.

Під час тестів було виявлено, що посилення 40дБ дає найкраще співвідношення сигналу до шуму. При більш високому посиленні фоновий шум занадто посилюється. Під час мовлення AGC зменшував розумний сигнал до

потрібного рівня, але коли користувач припиняв говорити шум повільно повертався.

Звуковий сигнал модуля зосереджений навколо 1.25 В, а сигнал змінювався в діапазоні від 0 В до 2.5 В. АЦП Arduino має 10 біт, тому числове значення може змінюватися від 0 до 1023. Так як було використано вихід 3.3 В Nano як опорну аналогову напругу, діапазон від 0 до 1023 означає від 0 В до 3.3 В.

Під час підключення можна приєднати модуль безпосередньо до одного з вхідних контактів АЦП, але на схемі було включено простий RC-фільтр високих частот. Це означає, що частоти нижче 1.4 кГц знижені. При цьому спектр є більш плоским, що дає можливість використовувати цілочисельну арифметику більш ефективно. Коли видаляються низькі частоти, АЦП та підсилювач менш схильні до кліпування. Оскільки модуль підключено до змінного струму, використовується два резистори для центрування входу АЦП близько до 1.65 В.

На рисунку 3.3 зображено схема підключення всіх компонентів до плати Arduino Nano.

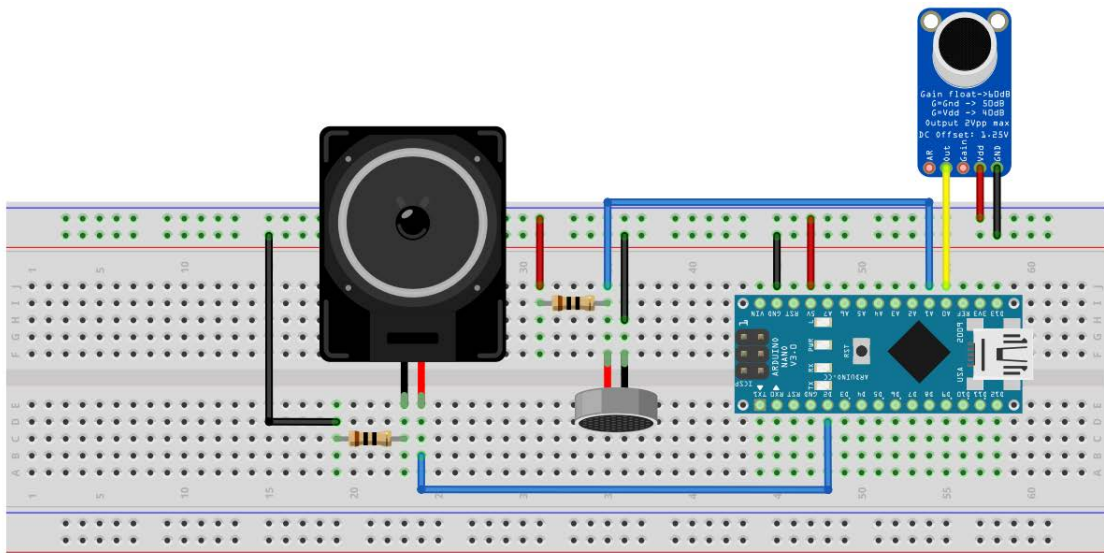


Рисунок 3.3 – Схема підключення компонентів до плати Arduino Nano

На рисунку 3.4 зображена ініціативна схема підключення компонентів до плати Arduino Nano

Зм.	Арк.	№докум.	Підпис	Дата

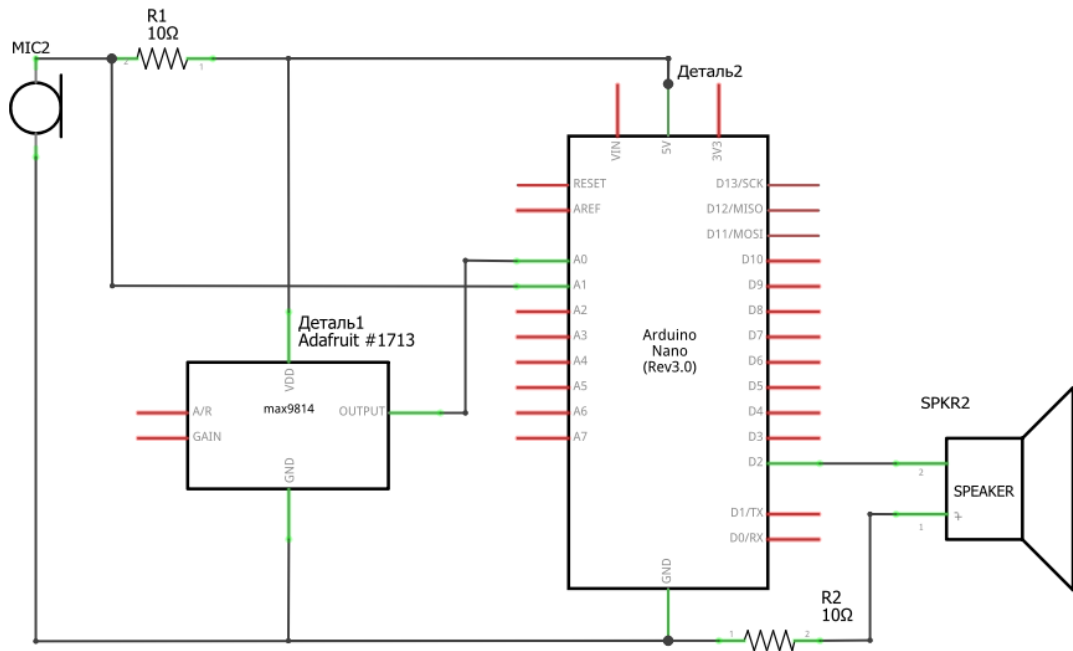


Рисунок 3.4 - Ініціативна схема підключення компонентів до плати Arduino Nano

3.2 Збір даних з АЦП Arduino Nano для подальшої обробки

Стандартним способом використання АЦП Arduino Nano є виклик `analogRead()`. Але `analogRead()` працює досить повільно. Вона ініціює АЦП та вибирає правильний вхідний контакт. Потім починається перетворення і очікується, доки перетворення не завершується. Це все займає приблизно 100 мкс.

У `Setup()` використовується стандартний код з бібліотеки Arduino, який використовується для ініціалізації АЦП:

```
analogReference(EXTERNAL);
```

```
analogRead(AUDIO_IN);
```

Опорна напруга для АЦП встановлюється на контакті `ARef`. Сам же `ARef` підключається до контакту 3.3 В. При першому виклику `analogRead()` буде отримано бібліотеку Arduino для конфігурації АЦП. У головному циклі для початку перетворення встановлюється біт `ADSC` (ADC Start Conversion). Це дає команду АЦП почати перетворення. Бібліотека Arduino перевела АЦП у режим єдиного перетворення, тому слід налаштувати `ADSC` для початку кожного перетворення.

Біт ADIF (прапорець переривання АЦП) встановлюється після того, як завершиться перетворення. Це означає, що можна робити щось інше, поки АЦП зайнятий. ADIF очищається шляхом встановлення значення 1. Біт ADIE (ADC Interrupt Enable) був очищений бібліотекою Arduino, тому фактично переривання не відбувається. Прапор перемикавання використовується для того, щоб перевірити, коли перетворення АЦП закінчується.

Десяти бітовий результат перетворення АЦП зчитується шляхом зчитування восьми розрядного регістру ADCL, а потім регістру ADCH. При зчитуванні даних з ADCL, значення ADCH заморожується, допоки його також не буде прочитано. Це зроблено для того, щоб гарантувати, що молодші та старші байти з різних зразків не будуть переплутані.

У циклі while потрібно викликати очікування байту АЦП за допомогою while (!getbit(ADCSRA, ADIF)). Далі в змінні value1 та value2 записуються ADCL та ADCH відповідно за допомогою строки byte value1 = ADCL. Після цього проводиться очистка прапору за допомогою команди bitSet() з аргументами ADCSRA та ADIF. Початок перетворення АЦП за допомогою тієї ж команди bitSet() з аргументами ADCSRA та ADSC. Далі в змінну value записується значення змінної value1 да плюсується значення value2 використовуючи value += value2 << 8.

Код «process the sample in val» виконується під час наступного перетворення АЦП.

Напруга на підсилювачі буде зосереджена в районі 512 В. Для обробки сигналу краще, щоб він був центрований навколо 0. Тому потрібно відняти середнє поточне значення вхідного значення від val.

Це можна зробити наступним чином:

```
static int zero = 512;
```

```
if (val < zero)
```

```
    zero--; else
```

```
    zero++;
```

```
val = val - zero;
```

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		39

Ескіз speakresog0.ino здійснює перевірку АЦП. Він може збирати зразки з швидкістю 9 тисяч вибірок за секунду. Скетч може надсилати значення на ПК за допомогою послідовної лінії, але послідовна передача сповільнює його приблизно до 1100 вибірок за секунду.

3.3 Алгоритм роботи цифрового фільтру діапазону частот та розрахунок часу виконання обчислень

Arduino nano має лише 2 Кбайт оперативної пам'яті, тому немає можливості зберігати усі зразки слова та повільно аналізувати їх. Через це потрібно робити більшу частину аналізу у режимі реального часу, під час того, як надходять зразки.

Розпізнавання мовлення найчастіше починається з вимірювання «енергії» у різних діапазонах частот, тобто амплітуди. Отже спочатку слід пропустити вхід через різні смугові фільтри. Arduino Nano не має достатньої обчислювальної потужності для того, щоб перетворення Фур'є під час надходження зразків. А навіть якщо і зможе то це буде досить повільно.

Більш вигідним способом поділу даних на діапазони у цій ситуації є використання цифрових фільтрів. Цифровий фільтр виконує просту математичну задачу для попередніх N вхідних вибірок а також попередніх N вибірок фільтра-входу для обчислення наступного вихідного значення фільтра. На рисунку 3.5 показано діаграму типового фільтру. На мові C це обчислення виглядало б як на формулі 2.1.

$$y[n] = i_0 * x[n] + i_1 * x[n - 1] + i_2 * x[n - 2] + j_1 * y[n - 1] + j_2 * y[n - 2], \quad (2.1)$$

де $x[n]$ – вхідне значення вибірки, а $y[n]$ – вихідне значення. $x[n-1]$, $y[n-2]$, тощо – попередні значення.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						40
Зм.	Арк.	№докум.	Підпис	Дата		

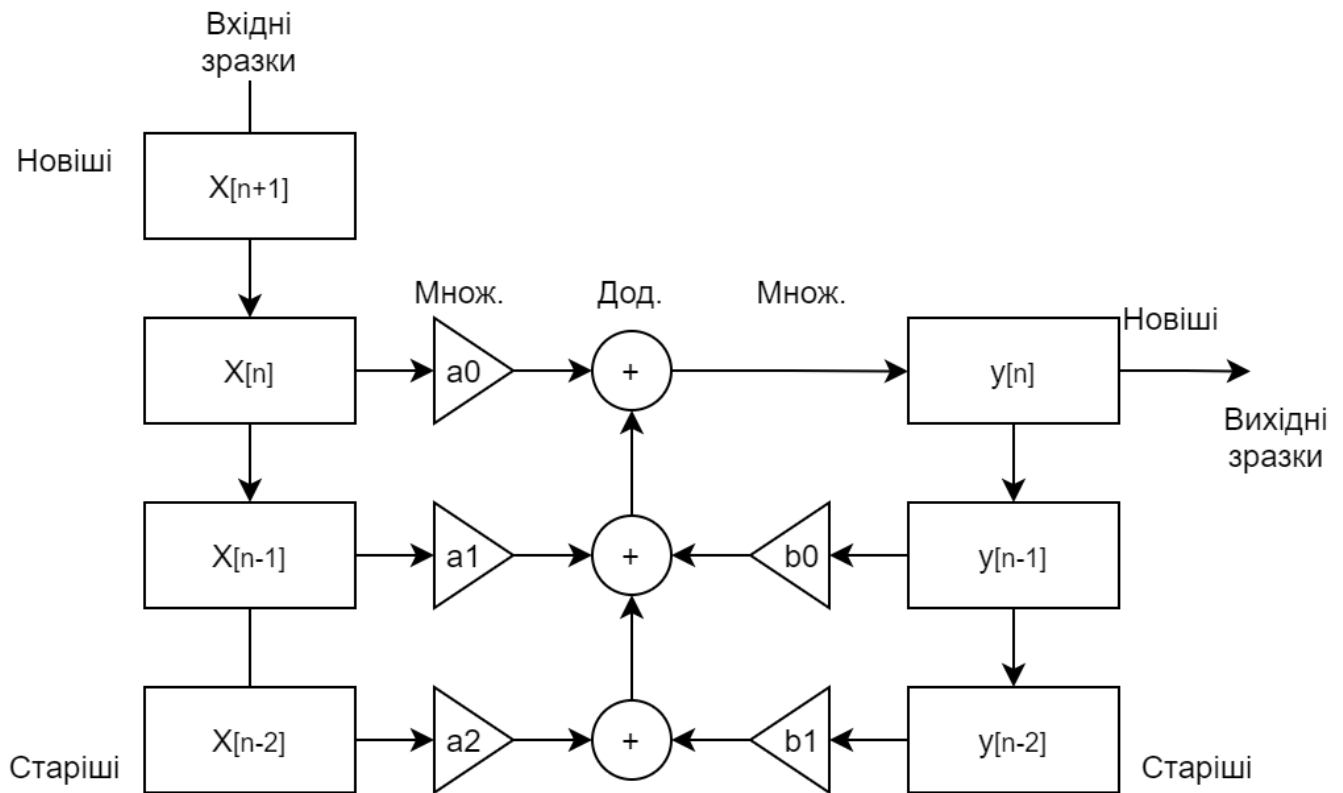


Рисунок 3.5 – Діаграма роботи ІІР фільтру

Програмою було збережено по два попередніх вхідних та вихідних значення. Оскільки зберігається по два файли, то цей фільтр називається фільтром другого порядку.

Якщо вихід залежить тільки від попередніх вхідних значень, він називається фільтром «кінцевої імпульсної характеристики», або «FIR» (b_0 і b_1 встановлені на нуль на діаграмі вище). Якщо вихід також залежить тільки від попередніх вихідних значень, то це фільтр «Нескінченної імпульсної характеристики», або «IIR».

Для смугового фільтру порядок фільтра визначає, наскільки круто фільтр спадає нижче і вище частоти пропускання. Чим вище порядок, тим більше контролю над кривою відгуку фільтра. З цього стає зрозуміло, що чим вище порядок, тим більше коефіцієнтів потрібно і тим більше математичних задач потрібно винести на вибірку.

FIR-фільтр потребує більше коефіцієнтів і більше обчислень, щоб отримати ту ж саму криву відгуку, що і фільтр IIR. Але IIR-фільтр являється менш стабільним. Фільтр IIR може легше помилитися. Через це його вихід може дати

несподіваний результат або зависнути. Це особливо ймовірно, коли використовується ціла арифметика, яка буде використовуватись на Nano.

Важко знайти значення того, наскільки швидко Nano може виконувати додавання та множення. Це залежить від того, як це вимірюється: чи вмикається, наприклад, отримання та збереження значень. Восьми бітове додавання займає від 0,4 до 0,9 мкс. Множення займає приблизно на 50% більше часу. Шістнадцяти бітове додавання або множення займає приблизно вдвічі більше (як і очікувалося). А тридцяти двох розрядне додавання або множення займає приблизно в 5 разів більше часу за одnobайтове.

Арифметика з поділом і плаваючою комою займає набагато більше часу, оскільки вони виконуються в програмному забезпеченні.

Якщо потрібна частота дискретизації 8000 вибірок за секунду, це 125 мкс на вибірку. З 4 діапазонами частот частота становить 31 мкс на вибірку на діапазон. Також потрібно зібрати дані з АЦП, обчислити амплітуду смуг і зберегти результати у масиві. Тому не можливо дозволити собі більше, ніж ІІР-фільтр другого порядку.

3.4 Розрахунок коефіцієнтів цифрового фільтру діапазону частот

Для коректної роботи будь-якого цифрового фільтру потрібно спершу знайти правильні коефіцієнти. Популярним цифровим фільтром є фільтр ІІР другого порядку (біквэд).

Для реалізації проєкту потрібно не менше чотирьох смугових фільтрів. Коефіцієнти для смугового двоквадратного фільтру розраховуються за допомогою формули 2.2 та 2.3.

$$\text{Coef1} = 2 * (K * (K - 1)) * \text{normal}. \quad (2.2)$$

$$\text{Coef2} = \left(1 - \frac{K}{Q} + K^2\right) * \text{normal}. \quad (2.3)$$

Коефіцієнти K та $normal$ розраховуються за формулами 2.4 та 2.5.

$$normal = \frac{1}{1 + \frac{K}{Q} + K^2}. \quad (2.4)$$

$$K = \operatorname{tg}\left(\pi * \frac{f}{sps}\right), \quad (2.3)$$

де f – центральна частота смуги, sps – частота дискретизації, Q – « Q -фактор», який розраховується як одиниця поділена на ширину смуги.

Під час розрахунків можна ігнорувати a_1 , так як воно нульове. a_2 являється від'ємним a_0 , що також спрощує обчислення.

Значення Q залежить від того, на скільки віддалені між собою смуги. У ситуації, коли є багато смуг, які розташовані близько одна до одної, і не потрібно щоб вони перетиналися доцільно зробити коефіцієнт Q більшим. Але якщо смуги розташовані далеко одна від одної то коефіцієнт Q не повинен бути на стільки великим, щоб між смугами були проміжки.

Для смугового фільтру $Q = f_{center} / (f_{max} - f_{min})$. Під час експериментів було виявлено, що Q повинно бути трохи меншим ніж 2. Переупорядкувавши рівняння, фільтр можна обчислити як:

$$L1 = -B1 * PD - B2 * PPD + input;$$

$$output = A0 * (L1 - PPD);$$

$$PPD = PD;$$

$$PD = L1;$$

Голосні розрізняються за формантами. Формант – це пік енергії спектра, а голосний можна розпізнати за відносними розмірами та частотами перших двох, або інколи трьох формантів. Частота першої чоловічої форманти може коливатись від 250 Гц до 850 Гц. Друга форманта в свою чергу – від 600 Гц до 2500 Гц. Форманти у жінок вищі приблизно на 15%, а у дітей на 35%. Звичайно, при цьому є великі індивідуальні відмінності.

Маючи тільки 4 діапазони частот, не можна сподіватись на обчислення форматних частот, але вони будуть впливати на енергію в різних діапазонах.

Смути, які будуть вибрані, залежатимуть від динаміку. Оскільки система призначена для розпізнавання голосу з декількох динаміків, її слід налаштувати зважаючи на них усіх.

Коефіцієнти можна розрахувати на ПК, але сам фактичний фільтр буде працювати на Arduino за допомогою цілочисельної арифметики. Самі коефіцієнти являються дійсними числами у діапазоні від 0 до 2, тому їх множить на 0x10000 та перетворюють на цілі числа. Вихід з АЦП знаходиться у діапазоні від 0 до 1023, тому Arduino виконує відповідне зміщення, щоб повернути вихід фільтру у той самий діапазон. Вхідні та вихідні значення фільтру зберігаються у вигляді 16-ти розрядні цілі, але деякі обчислення необхідно виконувати за допомогою 32-ох розрядних цілих, щоб уникнути проблем зв'язаних з переповненням.

На даному етапі відомо «енергію» (амплітуду) 4-ох діапазонів частот. Після цього було додано п'яту «смугу» для нульової швидкості переходу – «ZCR».

ZCR показує як часто сигнал перетинає нуль вольт. Це є дуже хорошим способом помітити фрикативні та неголосні губні м'язи, такі як s, sh, th, t, k, тощо. При цьому слід використовувати невеликий гістерезис, який обчислює ZCR, щоб не вловлювати шуми низького рівня.

Далі в роботі усі 5 смуг будуть сприйматись однаково та називатимуться «діапазонами», хоч ZCR не являється діапазоном частот.

Обчислення коефіцієнтів виконується за допомогою розробленої програми для Windows «SpeechRecog1.exe». Також програма експортує коефіцієнти як файл Coeffs.h. Приклад роботи програми «SpeechRecog1.exe» зображено на рисунку 3.4.

Для обчислення коефіцієнтів потрібно перейти на вкладку «Розрахувати коефіцієнти» та ввести частоти верхнього та нижнього діапазонів. Отримані коефіцієнти будуть відображатися як оголошення int у пам'ятці. Далі потрібно зайти у File/ExportCoefficients, для того щоб зберегти константи як файл Coeffs.h, який готовий до включення в Arduino.

Після цього потрібно скопіювати отриманий файл Coeffs.h в той самий каталог, у якому знаходяться ескізи speakrecog1.ino та voicerecog2.ino. Далі слід

перекомпілювати ці ескізи для того, щоб вони мали змогу виконувати смугову фільтрацію на Arduino.

SpeechRecog1.exe робить смугові фільтри «рівномірними» у логарифмічному масштабі. На рисунку 3.6 вісь частоти (вісь x) є лінійною.

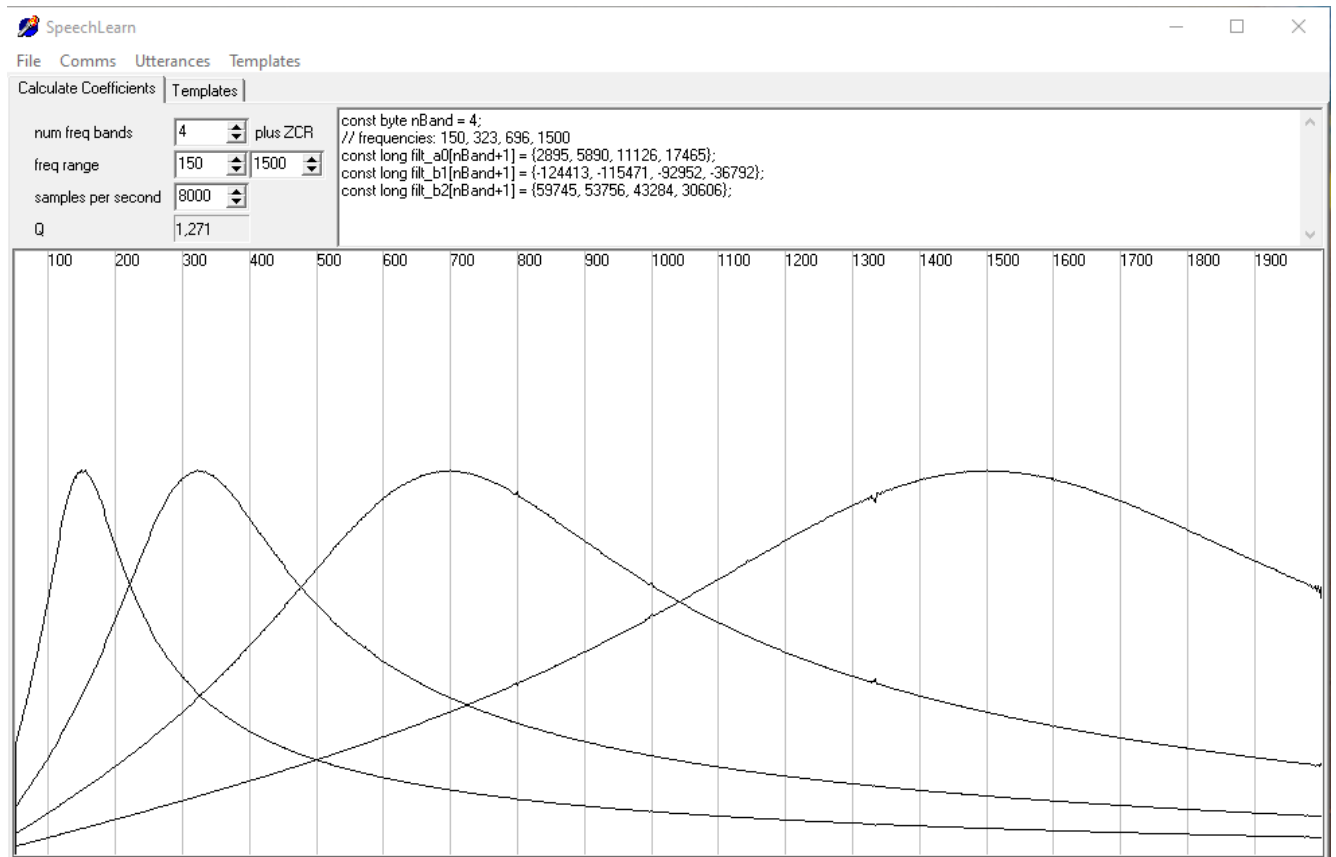


Рисунок 3.6 – Розрахунок коефіцієнтів за допомогою програми «SpeechRecog1.exe»

Коефіцієнт добротності повинен бути однаковим для усіх смуг, що означає, що усі вони повинні рівномірно розташовуватись у логарифмічному масштабі.

Після того, як ескіз speakrecog1.ino буде перекомпільований, він отримує зразки висловлювань та надсилає їх на ПК для того, щоб ПК міг обчислити «шаблони».

Ескіз speakrecog2.ino використовує шаблони для розпізнавання висловлювань.

3.5 Обробка шаблонів висловлювань

Все висловлювання ділиться Arduino на сегменти (кадри) довжиною 50 мс. У кожному сегменті вимірюється амплітуда з 5-ти діапазонів. Якщо висловлювання має 13 сегментів, то загалом виходить 65 16-бітових цілих, що охоплюють 0.65 секунд. Деякі висловлювання будуть більш короткими, тому останні кілька сегментів будуть близькими до нуля. Деякі висловлювання навпаки можуть бути довгими, тому остання частина може бути втрачена.

Вважається, що висловлювання починається тоді, коли загальна енергія в смугах перевищує поріг.

Після того, як ці 13 сегментів будуть збережені, потрібно вибрати, які з зразкових висловлювань найбільше нагадують і 65 чисел.

Як приклад можна використати висловлювання з цифрами від нуля до дев'яти.

Першим кроком є «нормалізація» вхідних даних. Це означає переконатися, що усі (наприклад три) висловлювання мають приблизно однакову гучність. Отже, дані сегментної смуги множаться на констант, щоб вони мали середню енергію (наприклад 100).

Спочатку ідеєю розв'язання задачі було використання методу аналізу головних компонентів, факторний або кластерний аналіз. Найочевиднішим був лінійний дискримінантний аналіз. Але за цим методом було складно відрізнити один тип висловлювання від іншого.

Наразі популярним рішенням є використання відстеження формантів. Зазвичай найбільш корисними є два найбільших піки у спектрі. Формантне відстеження спостерігає за тим, як змінюється частота цих піків від час висловлювання. Запустити відстеження формантів легко, якщо є повний спектр від перетворення Фур'є або якщо використовується LPC. Але цей метод не працює з 4-ма діапазонами частот.

Тому було прийнято рішення повернутися до абсолютно найпростішої схеми. Для цього просто потрібно створити десять шаблонів (типових прикладів

					КВРКІ 180204.18.02.25 ПЗ	Арк
						46
Зм.	Арк.	№докум.	Підпис	Дата		

висловлювань) для десяти різних цифр і виміряти різницю між вхідними даними та кожним із шаблонів.

Кожен шаблон містить по 65 значень цілих. Загальну різницю можна обчислити як:

for T = template

diff[T] = 0

for seg = segment

for band = band

diff [T] = diff [T] + abs(template[t,seg,band] -
incoming[seg,band])

Але деякі групи та сегменти являються більш важливими, ніж інші. Наприклад, частина «th» слова «three» досить різноманітна у порівнянні з частиною «ee». Отже до кожного числа у шаблоні додається «важливість» яка обчислюється так:

diff [t] = diff [t] + importance[t,seg,band] * abs(template[t,seg,band] -
incoming[seg,band])

Якщо значення для (t,seg,band) занадто відрізняється для даного класу висловлювання, значення шаблону менш важливе, аніж якщо значення завжди майже однако. Отже, «важливість» дорівнює $1 / (50 + \text{стандартне відхилення})$.

Вигляд висловлювання після сегментації зображено на рисунку 3.7.

3.6 Використання алгоритму динамічного викривлення часу

У розпізнаванні мовлення до записаних висловлювань часто застосовують «динамічне викривлення часу» (DTW, dynamic time warping).

Якщо порівнювати два приклади одного речення то, можливо таке, що перша половина в одному з прикладів була сказана трохи швидше. Отже при цьому можна розглянути першу половину, або розтягнути усе речення і змістити трохи вліво.

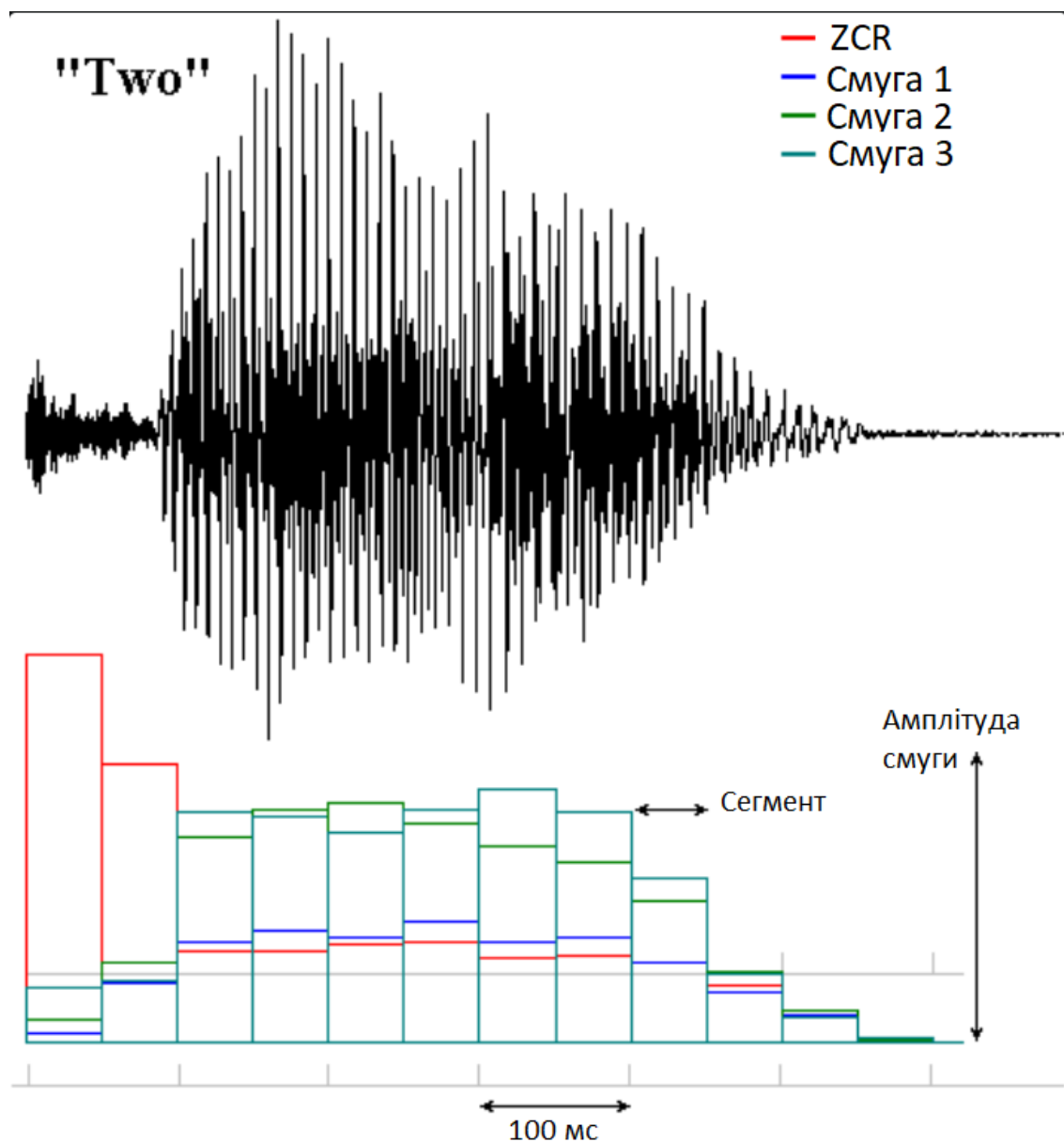


Рисунок 3.7 – Сегментоване висловлювання «two»

Під час експериментів було застосовано динамічне викривлення часу до вхідного висловлювання. Все висловлювання було переміщено та розтягнуто а також центральна його частина також була зрушена, розтягнута та переміщена.

Алгоритм полягав в тому, щоб знайти деформацію, яка найбільше відповідає шаблону вхідного висловлювання. Проблема полягає в тому, що викривлення застосовується щоб вислів краще відповідав правильному шаблону, але це також зробить його більш схожими на неправильні шаблони. Додаткові помилки, викликані поганими збігами, перевищують покращення, викликані хорошими збігами.

Одне слово є на стільки коротким, що динамічне викривлення часу не є корисним. Розтягування одного слова або його частини тільки погіршує ситуацію. Однак, зміщення висловлювання вправо, або вліво може дати більше позитивних збігів, не створюючи більше негативних.

Тому у фінальній версії все висловлювання може зміщуватись на два сегменти. Також йому дозволено зміщуватись на частину відрізка, тому зсув на 0.3 означає, що нове значення становить 70% від поточного значення і 30% суміжного значення.

3.7 Проведення навчання алгоритму розпізнавання шаблонів

Програму SpeechRecog1.exe, яка уже використовувалась для обчислення коефіцієнтів, також можна використовувати для обчислення шаблонів. Для цього слід підключити Arduino до ПК та вибрати правильний послідовний порт.

Далі потрібно завантажити скетч speakrecog1.ino на саму Arduino. Він використовує АЦП для вибірки мовлення з приблизною швидкістю 8000 секунд. Він фільтрує вибірки на 4 діапазони частот а також ZCR та зберігає 13 сегментів даних які мають тривалість 50 мс кожен. Мовлення починається, коли загальна енергія в смузі перевищує поріг. Після того, як 13 сегментів даних буде збережено, отримані 65 чисел відправляються на ПК.

На вкладці «Templates»/« Train Templates» можна переглянути деякі висловлювання, за допомогою яких можна обчислювати шаблони. Для цього за допомогою пункту меню «Файл/Відкрити» слід завантажити файл Train2raw.txt.

В вікні, яке відкрилось, можна натиснути на будь-яку клітинку і тоді відобразяться сегменти для цього прикладу. Якщо ж вибрати кілька клітинок, усі вони будуть відображені для їх порівняння. Приклад цього зображено на рисунках 3.8 та 3.9 відповідно.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						49
Зм.	Арк.	№докум.	Підпис	Дата		

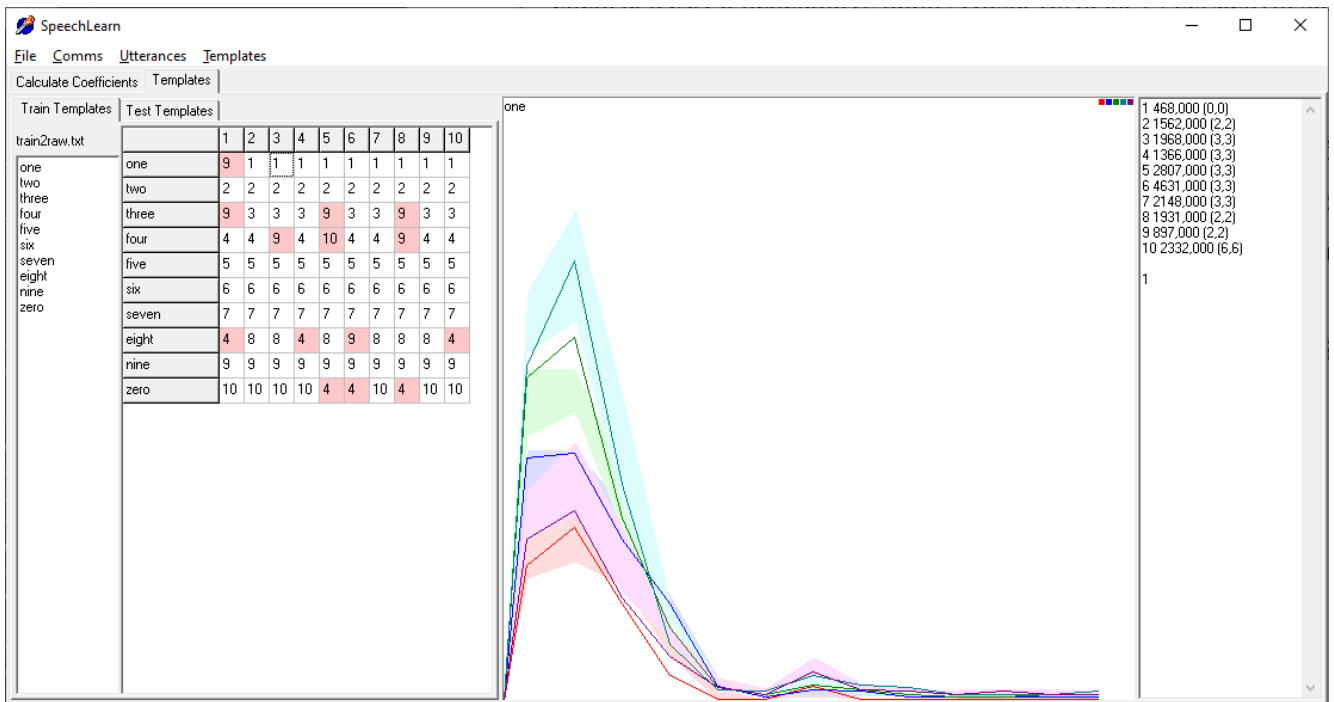


Рисунок 3.8 – Відображення сегментів у програмі SpeechRecog1.exe

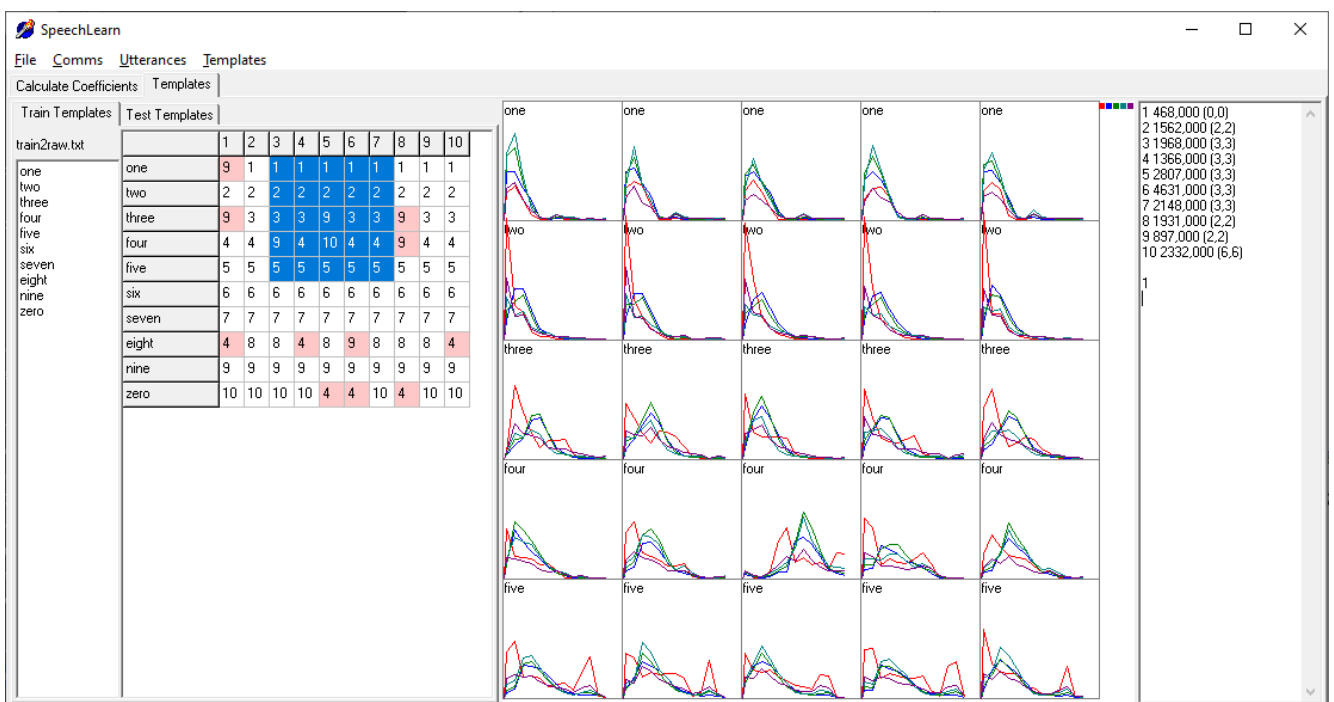


Рисунок 3.9 – Відображення сегментів декількох шаблонів в програмі SpeechRecog1.exe

Горизонтальна вісь – це час, а вертикальна вісь – амплітуда кожної смуги. Червона смуга – це ZCR. Якщо натиснути на лівий стовпець сітки, відобразиться середнє значення та SD усього рядка сітки.

Програма обчислює середнє значення та SD для кожного [сегмент, діапазон] для кожного з шаблонів (рядок сітки). Після підрахунку 10 шаблонів тепер містять середнє значення даних.

Тепер можна порівняти кожен з прикладів з певним шаблоном. Також є можливість зміщувати праворуч/ліворуч приклад, щоб отримати хорошу відповідність.

Коли відбувається натискання на клітинку, щоб відобразити висловлювання, воно автоматично порівнюється з шаблоном для усіх рядків (усіх висловлювань). Результати відображаються у правій записці. Число показує різницю (відстань) між висловлюваннями та даним шаблоном. Найнижча відстань є найкращою, тому вона відображається у сітці.

Результати, отримані на цьому етапі, не є великими. Отримано близько 15% поганих збігів. «Трійка» часто схожа на «дев'ятку», а «вісімка» та «нуль» іноді схожі на «четвірку».

Тому шаблони потрібно трохи відредагувати. Це можна зробити шляхом натискання на пункт меню Templates /OptimalShift. Кожен з прикладів зміститься вправо або вліво, поки він не буде найкраще відповідати потрібному шаблону для цього висловлювання. Середнє значення та SD перераховуються для кожного шаблону.

Результат редагування показано на рисунку 3.10. З хорошим набором для тренування зазвичай це дає 100% правильний результат.

Далі потрібно відкрити файл Test2.txt. Це завантажить деякі висловлювання, за допомогою яких можна перевірити шаблони. Далі знову потрібно розпізнати усі висловлення. Результати будуть не такими хорошими, але мають бути вищі ніж на 90% правильними.

Після цього можна додавати нові висловлювання, які надалі можуть бути командами для виконання приладами кіберфізичної системи «розумний будинок».

Для цього потрібно вирішити скільки буде висловлювань та ввести рядки для цих висловлень у нагадування зліва у вікні. Також слід перевірити чи немає порожніх рядків.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						51
Зм.	Арк.	№докум.	Підпис	Дата		

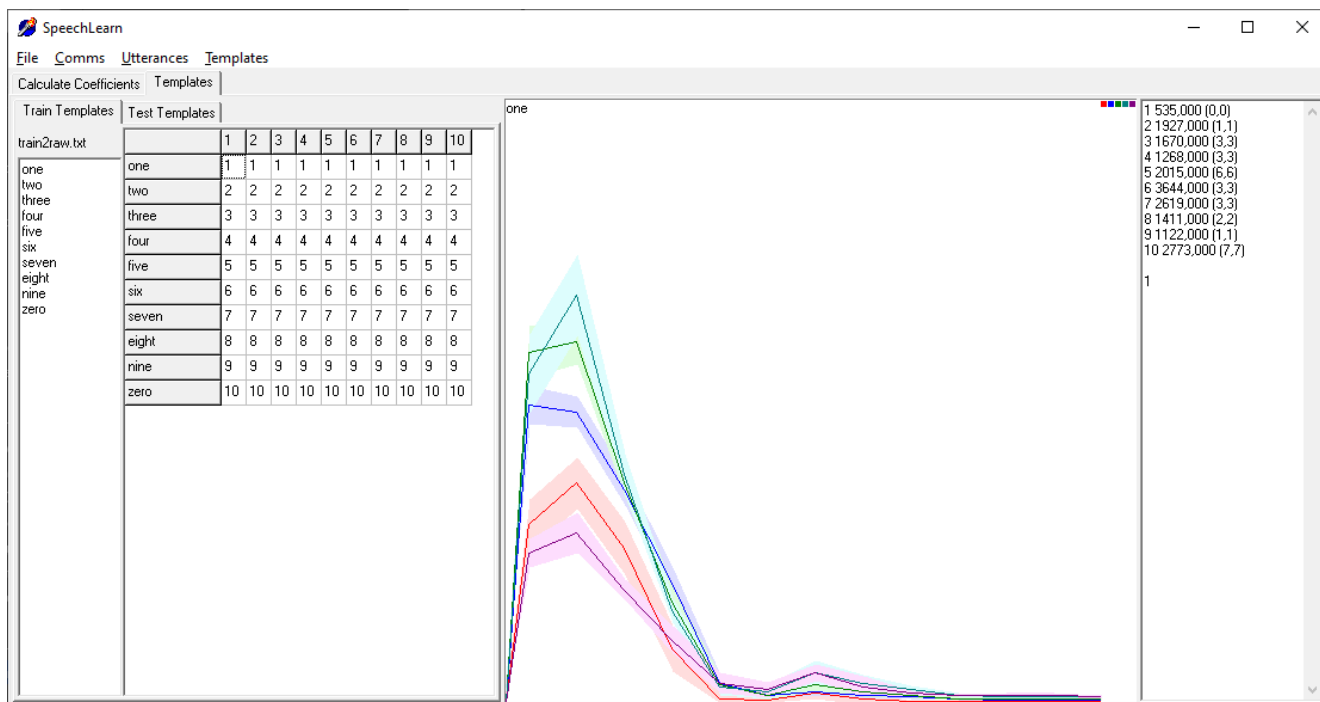


Рисунок 3.10 – Відображення сегментів після редагування шаблонів

Далі потрібно натиснути на пункт меню *Utterances/RecordTraining* для запису висловлювань. У діалоговому вікні, яке з'явилося, слід вибрати, скільки повторів кожного висловлювання буде використано як навчальний набір.

Після попередніх дій відкриється діалогове вікно, яке відображає кожне з висловлювань вибрану кількість разів. Висловлювання подані у довільному порядку. Тут Arduino надсилає сегментні дані до програми. Сітка у верхньому лівому куті вікна показує які з висловлювань же були записані.

Після запису всіх висловлювань сітка буде заповнена. Після цього необхідно розпізнати усі висловлення, щоб порівняти кожен з навчальних прикладів з кожним з шаблонів.

Також є можливість перезаписати висловлювання, яке являється неправильним. Після цього потрібно повторити попередню процедуру ще раз.

Далі можна зайти на вкладку «*Test Templates*» і записати навчальний набір. У список висловлювань не обов'язково вносити тільки правильні висловлювання. Для перевірки можна записати і декілька неправильних.

Після налаштування шаблонів їх можна експортувати на Arduino як файл *Templates.h*. Це можна зробити в вкладці меню *File/ExportTemplates*.

Далі потрібно скопіювати файл `Templates.h` у той самий каталог, де розташований ескіз `speakrecog2.ino`. Сам ескіз треба перекомпілювати, щоб він міг використовувати шаблони для розпізнавання висловлювань на Arduino. Також у цей каталог потрібно скопіювати файл `Coeffs.h`.

3.8 Алгоритм розпізнавання мовлення користувача

Скетч `speakrecog2.ino` використовується для розпізнавання мови на Arduino Uno, Nano, Mini та інших.

Після переміщення файлу `Templates.h` та файлу `Coeffs.h` у каталог, в якому знаходиться `speakrecog2.ino`, його потрібно повторно скомпілювати та завантажити на Arduino Nano.

Розпізнавач працює на Arduino Nano та використовує АЦП Arduino для оцифрування вхідного аудіо сигналу. У результаті є 16-ти бітове ціле число з центром в точці 0. Декілька смугових ПР фільтрів розділяють сигнал на діапазони частот.

Час поділений на сегменти по 50 мс. кожен. Далі вимірюється амплітуда кожної смуги у кожному з сегментів. Фіксована кількість сегментів (за замовчуванням 13) складають «висловлювання». Висловлювання починається коли перевищується поріг гучності.

Після цього вимірюється середня амплітуда усього висловлювання, щоб була можливість нормалізувати дані. При цьому розраховується швидкість перетину нуля (ZCR) сигналу.

Далі значення амплітуди смуги порівнюються зі значеннями шаблону. Сегменти можна зміщувати вправи або вліво, аби покращити співпадіння. При знаходженні найкращого співпадіння ескіз `speakrecog2.ino` надсилає текст розпізнаного слова на ПК через послідовний рядок, який використовується програмним забезпеченням для керування системою. Даний алгоритм зображено на рисунку 3.11.

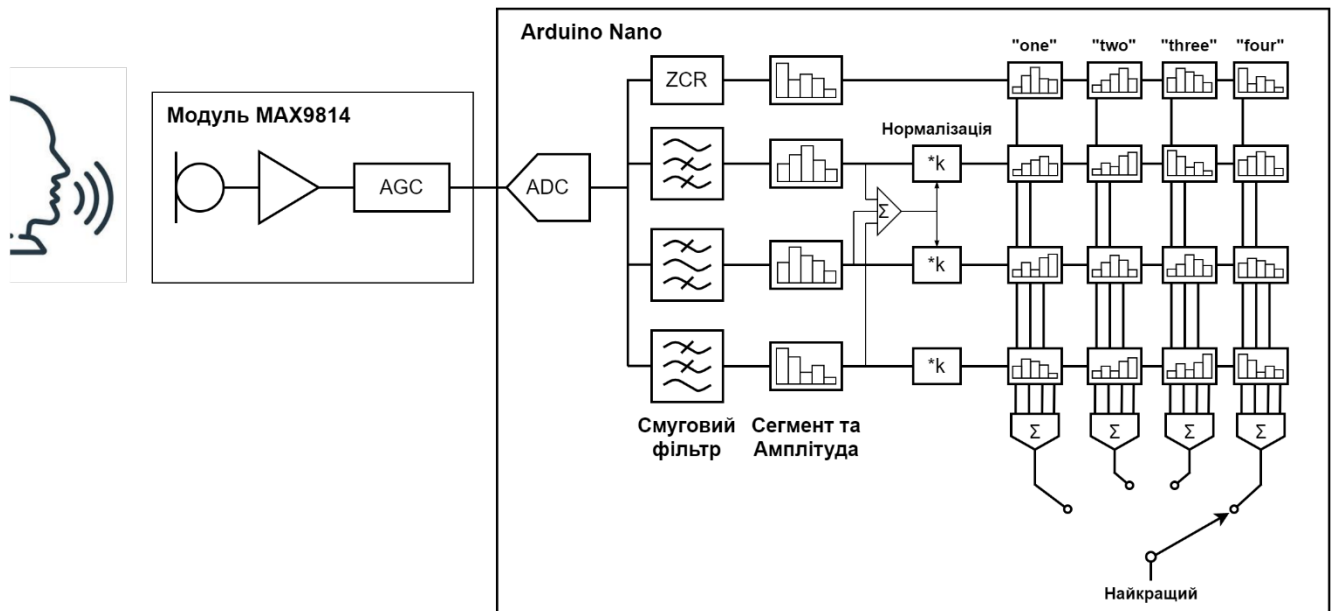


Рисунок 3.11 – Алгоритм розпізнавання мовлення користувача

3.9 Програмно-апаратна реалізація голосового асистента для кіберфізичної системи «Розумний будинок»

Загальна програмна система являється трохи складною. Її навчання проводиться на ПК, але уже навчена система повністю працює на Arduino Nano.

Arduino надсилає зразки висловлювань на ПК, який, у свою чергу, обчислює шаблони висловлювань. ПК експортує шаблони у вигляді файлу розширення .h, який компілюється в ескіз. Після цього ескіз може розпізнавати висловлювання без підключення до ПК.

Програма SpeechRecog1.exe, яка написана під Windows, обчислює коефіцієнти цифрового фільтру. Коефіцієнти цифрового фільтру експортуються як файл з назвою Coeffs.h.

Скетч speakrecog1.ino компілюється при використанні даних коефіцієнтів. Скетч speakrecog1.ino отримує зразки висловлювань та надсилає їх на ПК. На ПК SpeechRecog1.exe обчислює шаблони, які розпізнають дані висловлювання.

При необхідності, SpeechRecog1.exe збирає більше висловлювань для проведення тестування. Також він перевіряє шаблони за допомогою цих висловлювань.

Шаблони імпортуються як файл з назвою Templates.h.

Ескіз speakrecog2.ino компілюється уже за рахунок файлів Templates.h та Coeffs.h.

Ескіз speakrecog2.ino використовує шаблони для проведення розпізнавання висловлювань.

Загальну схему роботи програми зображено на рисунку 3.12.

На кінцевому кроці, замість відображення даних при тестуванні, дані надсилаються до програми, що знаходиться на ПК для розпізнавання команд, які знаходяться в висловлюваннях.

Далі на основі цих команд програма надає вказівки приладам, які під'єднані до кіберфізичної системи «Розумний будинок».

3.10 Інтерпретація команд, отриманих голосовим асистентом

Наступним кроком є написання програмного забезпечення, яке знаходить команди користувача в уже розпізнаних висловлюваннях та перетворює їх в вказівки для приладів, підключених до системи. В основні «вміння» голосового асистенту входить розпізнавання та синтез мовлення та керування приладами на основі отриманих інструкцій. Але на цьому перелік можливостей голосового асистенту не закінчується. Під час роботи було вирішено додати до можливостей асистента наступні пункти, які наведено в таблиці 3.1.

Для початку потрібно налаштувати голосового асистента. Оскільки голосовий асистент може мати стать, мову та ім'я, потрібно виділити під ці дані окремий клас, з яким буде йти робота в майбутньому. Зробити це можна наступним чином:

```
class VAssistant:  
    Name = ""  
    Sex = ""  
    SpeechLanguage = ""
```

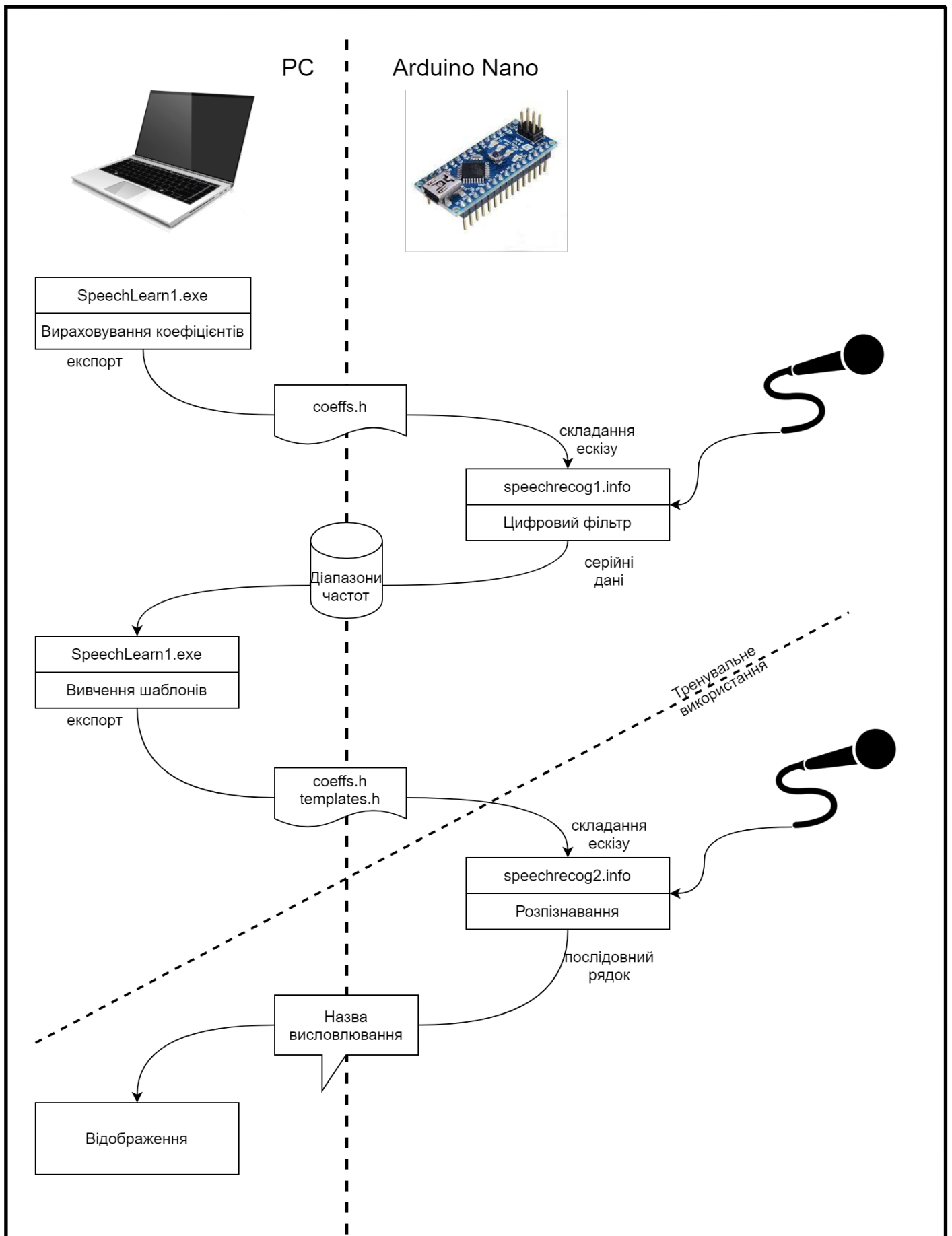


Рисунок 3.12 – Загальна структура програмного забезпечення та його зв'язок з апаратною частиною

Зм.	Арк.	№докум.	Підпис	Дата

Таблиця 3.1 – Опис можливостей голосового асистента

Опис вміння	Підтримка в offline режимі	Потрібні залежності
Синтез мовлення	Підтримується	pip install pyttsx3 – Синтез мовлення
Керування приладами системи «Розумний будинок»	Підтримується	-
Привітання та прощання	Підтримується	-
Зміна налаштувань мови розпізнавання та синтезу мовлення на ходу	Підтримується	-
Оповіщення про прогноз погоди	Не підтримується	pip install pyowm (OpenWeatherMap)
Проводити пошукові запити в пошуковій системі Google	Не підтримується	pip install google
Додавання та налаштування нагадувань	Підтримується	-

Для того, щоб задати голос асистенту було вирішено використати бібліотеку для offline синтезу мовлення pyttsx3. Вона автоматично знайде голоси, які доступні для синтезу, на комп'ютері в залежності від налаштувань оперативної системи. При необхідності потрібно встановити деякі додаткові мовні пакети.

Також в main функцію потрібно додати ініціалізацію синтезу мовлення та окрему функцію для її програвання.

В функцію `setup_assistant_voice()` слід вписати наступне:

В змінну `voices` записуються усі встановлені варіанти голосу з операційної системи за допомогою рядка `voices = ttsEngine.getProperty("voices")`.

Далі в залежності від вибраної мови для розпізнавання та синтезу мовлення (українська або англійська) параметром `assistant.recognition_language` встановлюється або "en-US" або "ua-UA" відповідно.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		57

Після цього встановлюється налаштування голосу асистента в залежності від вибраної статі.

Якщо вибрана англійська мова та жіноча стаття, встановлення налаштувань буде виконуватись за допомогою рядку `ttsEngine.setProperty("voice", voices[1].id)`. Після цього голосом голосового асистента буде встановлено Microsoft Zira Desktop.

Якщо ж вибрана англійська мова та чоловіча стаття, встановлення налаштувань буде виконуватись за допомогою рядку `ttsEngine.setProperty("voice", voices[2].id)`. Тоді голосом буде встановлено Microsoft David Desktop.

При виборі української мови на даний момент доступний тільки голос жіночої статі, який встановлюється за допомогою рядку `ttsEngine.setProperty("voice", voices[0].id)`.

Щоб синтезувати фразу з заданими раніше налаштуваннями була створена функція `play_voice_assistant_speech()`, яка приймає в якості аргументу в змінній `text_to_speech` фразу, яку потрібно озвучити. Для самого синтезу в функції потрібно вписати два наступні рядки:

```
ttsEngine.say(str(text_to_speech))
ttsEngine.runAndWait()
```

Далі потрібно написати логіку для обробки команд. Це можна зробити у два способи.

Перший спосіб це використання JSON-подібного об'єкту, в якому зберігаються наміри, сценарії розвитку та відповіді на команди. Приклад коду для такого способу наведено нижче:

```
config = {"intents": {"greeting": {
    "examples": ["привіт", "добрий день", "hello", "good morning"],
    "responses": play_greetings ... },
    "failure_phrases": play_failure_phrase }
```

В цьому фрагменті зображено приклад привітання. Коли в мові користувача звучить фраза у якій є одне з ключових слів програється привітання від імені голосового асистента. У випадку, якщо жоден з варіантів не підходить програється фраза помилки.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		58

Такий варіант буде вдалим при бажанні навчити голосового асистента відповідати на складні фрази. Більш того, при цьому підході можна застосувати NLU підхід та створити можливість передбачати наміри користувача, звіряючи їх з тими, що є у конфігурації.

Другий спосіб полягає у використанні спрощеного словника, в якому в якості ключів буде хешований тип кортежу, а в вигляді значень будуть назви функцій, які будуть виконуватись. Приклад коду для такого методу зображено нижче:

```
commands = {  
    ("hello", "hi", "morning", "привіт"): play_greetings,  
    ("goodbye", "quit", "exit", "stop", "бувай"): play_farewell_and_quit, ... }
```

Наступним кроком буде додавання до програми підтримки різних мов. Для того, щоб навчити голосового асистента працювати з декількома мовними моделями, необхідно організувати невеликий JSON файл з подібною структурою:

```
{  
    "Can you check if your microphone is on, please?": {  
        "ua": "Чи не могли б ви перевірити чи ваш мікрофон увімкнено?",  
        "en": "Can you check if your microphone is on, please?" }, ... }
```

Структура голосового асистента дає можливість переключити мову між англійською та українською. Тому мова буде вибрана в залежності від вибраної мови синтезу самого голосового асистента.

Для того, щоб отримувати переклад потрібно створити клас таким чином:

```
class Translations:  
    with open("Translation.json", "r", encoding="UTF-8") as file:  
        Translations = json.load(file)
```

Також потрібно передбачити функцію яка буде повертати строку з перекладом:

```
def GetStr(self, Text: str):  
    if Text in self.Translation:  
        return self.Translation[Text][Assistant.SpeechLanguage]  
    else:
```

```
print(colored("Not translated: {}".format(Text), "red"))
```

```
return Text
```

У випадку відсутності перекладу відбувається вивід повідомлення про це в логи та повернення початкового тексту.

У таблиці 3.2 наведено основні команди для взаємодії з голосовим асистентом.

Таблиця 3.2 – Основні команди для взаємодії з голосовим асистентом

Назва функції	Функціонал	Приклади команди
TurnOnTheLight(room, level)	Ввімкнення світла в кімнаті room з рівнем освітлення level	Ввімкни світло, turn on the light
TurnOffTheLight(room)	Вимкнення світла в кімнаті room	Вимкни світло, turn off the ligh
ChangeLightingLevel(level)	Встановлення рівня освітлення на рівень level	Зміни рівень освітлення, change the lighting level
TurnOnMusic(music)	Ввімкнення музики music	Ввімкни музику, play music
TurnOnTV(mode, video)	Ввімкнути телевізор в режимі mode. Вмикає телебачення або відео з відеохостингу	Ввімкни телевізор, ввімкни відео, turn on the TV, play video
TurnOnTeapot(deferment)	Ввімкнення електрочайнику з відстрочкою в часі deferment	Ввімкни чайник, turn on the teapot
FindSmthInternet(find)	Пошук інформації в мережі Інтернет	Знайди, find in the internet
ChangeTheSettings(setting, parameter)	Змінює налаштування setting голосового асистента на parameter	Встанови параметр, налаштуй, configurate

3.11 Висновки

У третьому розділі було встановлено апаратні з'єднання між компонентами голосового помічника. Після цього проводилось налаштування збору даних з АЦП Arduino Nano для подальшої обробки.

Далі було створено алгоритм роботи цифрового фільтру діапазону частот та розраховано коефіцієнти для нього. Після цього було реалізовано алгоритм обробки шаблонів висловлювань та проведено навчання алгоритму розпізнавання шаблонів з використанням прийому динамічного викривлення часу.

Під кінець було завершено розробку алгоритму розпізнавання мовлення користувача та програмно-технічну реалізацію голосового асистента. Також було створено програму-інтерпритатор команд та налаштовано самого асистента.

					КВРКІ 180204.18.02.25 ПЗ	Арк
						61
Зм.	Арк.	№докум.	Підпис	Дата		

ВИСНОВКИ

Метою кваліфікаційної роботи є проектування та програмно-технічна реалізація голосового асистента для підвищення комфорту керування кіберфізичною системою «Розумний будинок» за допомогою голосу.

Об'єктом дослідження є процес розпізнавання голосу та процес керування кіберфізичною системою «Розумний будинок» за допомогою голосу.

Предметом дослідження є програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок".

Для досягнення поставленої мети використовуються алгоритми розпізнавання голосу.

Практична цінність полягає в розробленні програмно-технічної реалізації голосового асистента для кіберфізичної системи "Розумний будинок", що надає можливість підвищити комфорт керування кіберфізичною системою «Розумний будинок».

В ході першого розділу було розглянуто такі поняття, як інтернет речей та технології кіберфізичною системою «Розумний будинок». Також при цьому було досліджено уже відомі реалізації голосового асистента від всесвітньовідомих брендів, такі як : Apple HomeKit, Google Home, Amazon Echo.

На основі даних досліджень було виявлено їхні переваги та недоліки та задано перспективи для майбутньої розробки. Також було розглянуто алгоритми для розпізнавання голосу, які будуть використані при реалізації розумного асистента. На цьому етапі паралельно було проведено огляд характеристик плати Arduino Nano та цифрових та аналогових IIR фільтрів.

У другому розділі було проведено обґрунтування вибору апаратних ресурсів та програмного забезпечення відповідно до поставленої задачі. Також на цьому етапі був складений список дій, що потрібно реалізувати для розробки голосового асистента для кіберфізичної системи «Розумний будинок».

					КВРКІ 180204.18.02.25 ПЗ	Арк
						62
Зм.	Арк.	№докум.	Підпис	Дата		

Далі було проведено підключення мікрофону до плати та подальше його тестування. Також у другому розділі була створена приблизна модель взаємодії голосового асистента з елементами кіберфізичної системи «Розумний будинок».

У третьому розділі було встановлено апаратні з'єднання між компонентами голосового помічника. Після цього було проведено налаштування збору даних з АЦП Arduino Nano для подальшої обробки.

Наступним кроком було створення алгоритму роботи цифрового фільтру діапазону частот та розрахування коефіцієнти для нього. Після цього було реалізовано алгоритм обробки шаблонів висловлювань та проведено навчання алгоритму розпізнавання шаблонів з використанням прийому динамічного викривлення часу.

Під кінець було завершено розробку алгоритму розпізнавання мовлення користувача та програмно-технічну реалізацію голосового асистента. Також було створено програму-інтерпритатор команд та налаштовано самого асистента.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		63

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Patil H., Mangal M., Salunke S., Bhoir A., Patil K. Voice assistant home automation using Arduino. *Int. Res. J. Mod. Eng. Technol. Sci.* 2021. Pp. 160-164.
2. Saravanan M., Selvababu B., Jayan A., Anand A., Raj A. Arduino Based Voice Controlled Robot Vehicle. *In IOP Conference Series: Materials Science and Engineering.* 2020. Pp. 2-7.
3. Alam T., Benaida M. “The Role of Cloud-MANET Framework in the Internet of Things (IoT)”. *International Journal of Online Engineering (iJOE).* 2019. Vol. 14. Pp. 97-111.
4. Irawan Y., Yulisman Y., Belarbi N., Josephine M. M. Voice-Based Home Security and SMS Gateway Using Arduino Uno Microcontroller And Passive Infra Red Sensor. *Journal of Applied Engineering and Technological Science (JAETS).* 2021. Pp. 19-25.
5. Dhiren T., Mohammed A., Al-Kuwari A. H., Potdar V. Energy conservation in a smart home. *In 2017 5th IEEE International Conference on Digital Ecosystems and Technologies,* 2017. Pp. 241-246.
6. Alam T. Middleware Implementation in Cloud-MANET Mobility Model for Internet of Smart Devices, *International Journal of Computer Science and Network Security,* 2017. Pp. 86-94.
7. Ehsani F., Silke M., Rolandi W. Smart home automation systems and methods. U.S. Patent. 2019. Pp. 9-11.
8. Alam T., Benaida M. CICS: Cloud–Internet Communication Security Framework for the Internet of Smart Devices. *International Journal of Interactive Mobile Technologies (IJIM).* 2018. No. 12. Pp. 74-84.
9. Tanweer A., Rababah B., Convergence of MANET in Communication among Smart Devices in IoT. *International Journal of Wireless and Microwave Technologies(IJWMT).* 2019. Vol. 9. No. 2. Pp. 1-10.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		64

10. Masykur F., Prasetiyowati F., Studi P., Informatika T., Ponorogo U. M., Pi R. Aplikasi Rumah Pintar (Smart Home) Pengendali Peralatan. *J. Teknol. Inf. dan Ilmu Komput.* 2016. Vol. 3. No. 1. Pp. 51–58.
11. Putro M. D., Kambey F. D. Sistem Pengaturan Pencahayaan Ruangan Berbasis Android pada Rumah Pintar. *J. Nas. Tek. Elektro.* 2016. Vol. 5. No. 3. P. 297.
12. Sadewo A. D. B., Widasari E. R., Muttaqin A. Perancangan Pengendali Rumah menggunakan Smartphone Android dengan Konektivitas Bluetooth. *J. Pengemb. Teknol. Inf. dan Ilmu Komput.* 2017. Vol. 1. No. 5. Pp. 415–425.
13. Muslihudin M., Renvilia W., Andoyo A., Susanto F. Implementasi Aplikasi Rumah Pintar Berbasis Android Dengan Arduino Microcontroller. *J. Keteknikan dan Sains*, 2018. Vol. 1. No. 1. Pp. 23–31.
14. Mowad M., Fathy A., Hafez A. Smart home automated control system using android application and microcontroller. *Int. J. Sci. Eng. Res.* 2018. Vol. 5. No. 5. Pp. 935–939.
15. Mahdi O. A., Alankar B. Wireless Controlling Of Remote Electrical Device Using Android Smartphone. *IOSR J. Comput. Eng.* 2018. Vol. 16. No. 3. Pp. 23–27.
16. Supriyono P., Marjuki P. Ultrasonic Sensor Parking Assistant With Arduino Uno. *International Journal of Advanced Research in Engineering and Technology (IJARET)*. 2020. P. 11.
17. Shakthidhar S., Srikrishnan P., Santhosh S., Sandhya M. K. Arduino and NodeMcu based ingenious household objects monitoring and control environment. *In 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*. 2019. Vol. 1. Pp. 119-124.
18. Al Rakib M. A., Rahman M. M., Anik M. S. A., Masud F. A. J., Rahman M. A., Islam S., Abbas F. I. Arduino Uno Based Voice Conversion System for Dumb People. *European Journal of Engineering and Technology Research*. 2022. Pp. 118-123.
19. Noda K. Google Home: smart speaker as environmental control unit. *Disability and rehabilitation: assistive technology*. 2018. Pp. 674-675.

					КВРКІ 180204.18.02.25 ПЗ	АДК
Зм.	АДК.	№докум.	Підпис	Дата		65

20. Pyae A., Joelsson T. N. Investigating the usability and user experiences of voice user interface: a case of Google home smart speaker. *In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 2018. Pp. 127-131.

21. Pêgo P. R., Nunes L. Automatic discovery and classifications of IoT devices. *In 2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. 2017. Pp. 1-10.

22. Poushneh A. Humanizing voice assistant: The impact of voice assistant personality on consumers' attitudes and behaviors. *Journal of Retailing and Consumer Services*. 2021. Pp. 58-59.

23. Beirl D., Rogers Y., Yuill N. Using voice assistant skills in family life. *In Computer-Supported Collaborative Learning Conference*. 2019. Vol. 1. Pp. 96-103.

24. Liao S., Wilson C., Cheng L., Hu H., Deng H. Measuring the effectiveness of privacy policies for voice assistant applications. *In Annual Computer Security Applications Conference*. 2020. Pp. 856-869.

25. Subhash S., Srivatsa P. N., Siddesh S., Ullas A., Santhosh B. Artificial intelligence-based voice assistant. *In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. 2020. Pp. 593-596.

26. Palanica A., Thommandram A., Lee A., Li M., Fossat, Y. Do you understand the words that are comin outta my mouth? Voice assistant comprehension of medication names. *NPJ digital medicine*. 2019. Pp. 1-6.

27. Loureiro S. M. C., Japutra A., Molinillo S., Bilro R. G. Stand by me: analyzing the tourist–intelligent voice assistant relationship quality. *International Journal of Contemporary Hospitality Management*. 2021. Pp. 15-17.

28. Raveh E., Siegert I., Steiner I., Gessinger I., Möbius B. Three's a Crowd? Effects of a Second Human on Vocal Accommodation with a Voice Assistant. *In INTERSPEECH*. 2019. Pp. 405-409).

29. Yan C., Ji X., Wang K., Jiang Q., Jin Z., Xu W. A Survey on Voice Assistant Security: Attacks and Countermeasures. *ACM Computing Surveys (CSUR)*. 2022. Pp. 57-60.

					КВРКІ 180204.18.02.25 ПЗ	АДК
Зм.	АДК.	№докум.	Підпис	Дата		66

30. Tassiello V., Tillotson J. S., Rome A. S. “Alexa, order me a pizza!”: The mediating role of psychological power in the consumer–voice assistant interaction. *Psychology & Marketing*. 2021. Pp. 1069-1080.

31. Cheng P., Roedig U. Personal Voice Assistant Security and Privacy—A Survey. *Proceedings of the IEEE*. 2022. Pp. 476-507.

32. Kumar R., Sarupria G., Panwala V., Shah S., Shah N. Power efficient smart home with voice assistant. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2022. Pp. 1-5.

33. Baeza R. R., Kumar A. R. Perceived Usefulness of Multimodal Voice Assistant Technology. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2019. Vol. 63. No. 1. Pp. 150-154.

34. Feng H., Fawaz K., Shin K. G. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 2017. Pp. 343-355.

					КВРКІ 180204.18.02.25 ПЗ	Арк
Зм.	Арк.	№докум.	Підпис	Дата		67

Додаток Г

(обов'язково)

Лістинг коду програми по обробці команд для кіберфізичної системи «Розумний будинок»

```
from vosk import Model, KaldiRecognizer
import speech_recognition
import pyttsx3
import wave
import json
import os

class VoiceAssistant:
    """
    Налаштування голосового асистента, включаючи ім'я, стать та мову
    """
    name = ""
    sex = ""
    speech_language = ""
    recognition_language = ""

def setup_assistant_voice():
    """
    Встановлення голосу за замовчуванням (індекс може мінятись
    в залежності від налаштувань операційної системи)
    """
    voices = ttsEngine.getProperty("voices")

    if assistant.speech_language == "en":
        assistant.recognition_language = "en-US"
        if assistant.sex == "female":
            # Microsoft Zira Desktop - English (United States)
            ttsEngine.setProperty("voice", voices[1].id)
        else:
            # Microsoft David Desktop - English (United States)
            ttsEngine.setProperty("voice", voices[2].id)
    else:
        assistant.recognition_language = "ua-UA"
        ttsEngine.setProperty("voice", voices[0].id)
```

```

def play_voice_assistant_speech(text_to_speech):
    """
    Програвання відповідей голосового асистента (без збереження аудіо)
    :param text_to_speech: текст, який потрібно перетворити в звук
    """
    ttsEngine.say(str(text_to_speech))
    ttsEngine.runAndWait()

def record_and_recognize_audio(*args: tuple):
    """
    Запис та розпізнавання аудіо
    """
    with microphone:
        recognized_data = ""

        recognizer.adjust_for_ambient_noise(microphone, duration=2)

        try:
            print("Listening...")
            audio = recognizer.listen(microphone, 5, 5)

            with open("microphone-results.wav", "wb") as file:
                file.write(audio.get_wav_data())

        except speech_recognition.WaitTimeoutError:
            print("Can you check if your microphone is on, please?")
            return

        try:
            print("Started recognition...")
            recognized_data = recognizer.recognize_google(audio, language="ua").lower()

        except speech_recognition.UnknownValueError:
            pass

        except speech_recognition.RequestError:
            print("Trying to use offline recognition...")
            recognized_data = use_offline_recognition()

    return recognized_data

def use_offline_recognition():
    recognized_data = ""
    try:

```

```
if not os.path.exists("models/vosk-model-small-ru-0.4"):
    print("Please download the model from:\n"
          "https://alphacephei.com/vosk/models and unpack as 'model' in the current
folder.")
    exit(1)
```

```
wave_audio_file = wave.open("microphone-results.wav", "rb")
model = Model("models/vosk-model-small-ru-0.4")
offline_recognizer = KaldiRecognizer(model, wave_audio_file.getframerate())
```

```
data = wave_audio_file.readframes(wave_audio_file.getnframes())
```

```
if len(data) > 0:
```

```
    if offline_recognizer.AcceptWaveform(data):
```

```
        recognized_data = offline_recognizer.Result()
```

```
        recognized_data = json.loads(recognized_data)
```

```
        recognized_data = recognized_data["text"]
```

```
except:
```

```
    print("Sorry, speech service is unavailable. Try again later")
```

```
return recognized_data
```

```
if __name__ == "__main__":
```

```
    recognizer = speech_recognition.Recognizer()
```

```
    microphone = speech_recognition.Microphone()
```

```
    ttsEngine = pyttsx3.init()
```

```
    assistant = VoiceAssistant()
```

```
    assistant.name = "Alice"
```

```
    assistant.sex = "female"
```

```
    assistant.speech_language = "ua"
```

```
    setup_assistant_voice()
```

```
    while True:
```

```
        voice_input = record_and_recognize_audio()
```

```
        os.remove("microphone-results.wav")
```

```
        print(voice_input)
```

```
        voice_input = voice_input.split(" ")
```

```
        command = voice_input[0]
```

```
        if command == "Привіт":
```

```
            play_voice_assistant_speech("Доброго дня")
```

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011400099

Дата перевірки:
31.05.2022 16:07:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
31.05.2022 16:07:58 EEST

ID користувача:
100005591

Назва документа: Головатюк А.О_Програмно-технічна реалізація голосового асистента для кіберфізичної си...

Кількість сторінок: 60 Кількість слів: 9828 Кількість символів: 71850 Розмір файлу: 6.86 MB ID файлу: 1011282824

2% Схожість

Найбільша схожість: 0.64% з джерелом з Бібліотеки (ID файлу: 1011215216)

1.21% Джерела з Інтернету

31

Сторінка 62

0.96% Джерела з Бібліотеки

58

Сторінка 62

3.14% Цитат

Цитати

4

Сторінка 63

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

19

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 11%

ID: 104277 Название: Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок" Добавлено в БД: 2022-05-31 Авторы: А.О. Головатюк Руководители: К.Ю. Бобровнікова Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	65456	682	500 (1%)	8 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Головатюк Андрій Олександрович

Тема: Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок"

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 58

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проектування та програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок" для підвищення комфорту керування кіберфізичною системою «Розумний будинок».

2. Висновок про відповідність роботи дипломному завданню: Кваліфікаційна робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: Розділ 1 – проаналізовано предметну область, досліджено відомі реалізації голосового асистента, проаналізовано алгоритми розпізнавання голосу та здійснено постановку задачі. Розділ 2 – здійснено проектування голосового асистента для кіберфізичної системи "Розумний будинок", здійснено програмну реалізацію підключення мікрофону. Розділ 3 – створена та протестована програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок". Усі розділи відповідають завданню.

4. Позитивні сторони роботи: розроблення програмно-технічної реалізації голосового асистента для кіберфізичної системи "Розумний будинок" надає можливість підвищити комфорт керування кіберфізичною системою «Розумний будинок».

5. Негативні сторони роботи: недоліком розроблюваного пристрою є те, що інтерпритація команд здійснюється за допомогою ключових слів, які наперед записані в кодї програми. Для розуміння більш складних команд та конструкцій мовлення потрібно задіяти штучний інтелект.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Оформлення пояснювальної записки відповідає діючим стандартам оформлення документації.

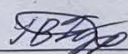
7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: відмінно.

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Гурман Іван Васильович, к.т.н., доцент кафедри інженерії програмного забезпечення

“1” __червня__ 2022 р.

 (підпис)

Завідувачу кафедри КПС
д-ру техн.наук, проф. Говорущенко Т. О.

Головатюк А.О.

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ-18-2

ЗАЯВА

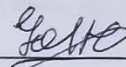
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

30.05.2022

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічна реалізація голосового асистента для кіберфізичної системи "Розумний будинок"

Автор: Головатюк Андрій Олександрович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Бобровнікова Кіра Юліївна, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

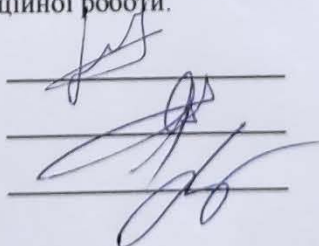
- 1) запозичення розміщені у розділах дослідження відомих реалізацій голосового асистента, які не є авторською роботою та не стосуються результатів роботи;
- 2) усі запозичення, використані в першому та частині другого розділах мають належним чином оформлені джерела та посилання, звідки їх було взято;
- 3) деякі виявлені збіги є загальноживаними фразами або висловлюваннями;
- 4) усі зафіксовані системою ознаки модифікації тексту є назвами, написаними латинськими символами, або шматки коду.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 2% і адресується до 89 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



К.Ю. Бобровнікова

С. М. Лисенко

Т. О. Говорущенко