

## КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб жестового керування пристроями розумного будинку  
Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ. 220117.04.22.19 ПЗ

Виконав здобувач IV курсу, група KI2-22-4

  
Підпис

Ксенія ПОЛШЦУК  
Ініціали, прізвище

Керівник канд. техн. наук, доц.  
Науковий ступінь, учене звання

  
Підпис

Володимир ГРИГА  
Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.  
Науковий ступінь, учене звання

  
Підпис

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
завідувач кафедри КІС  
«01» червня 2026 р.

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

дата

Хмельницький 2026

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Поліщук Ксенії Костянтинівні

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб жестового керування пристроями розумного будинку

Керівник проекту (роботи) Грига В.М., к.т.н., доцент

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

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Строк подання студентом проекту (роботи) на кафедрі 01.06.2026 р.

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

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

Огляд технічних засобів для керування пристроями в розумному будинку

Проектування програмно-технічного засобу жестового керування пристроями розумного будинку

Реалізація та тестування програмно-технічний засіб жестового керування пристроями розумного будинку

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

Структура програмно-технічного засобу жестового керування пристроями розумного будинку

Результати роботи

Схема цифрового двійника виконавчого пристрою у середовищі Wokwi

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – Огляд традиційних та сучасних методів жестового керування пристроями у Розумному будинку	01.03.2026	виконано
4	Робота над розділом 2 – Проектування програмно-технічного засобу жестового керування пристроями Розумного будинку	01.04.2026	виконано
5	Робота над розділом 3 – Реалізація програмно-технічного засобу жестового керування пристроями Розумного будинку	28.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач

  
Підпис

Ксенія ПОЛЩУК

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

  
Підпис

Володимир ГРИГА

Імя, ПРІЗВИЩЕ



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб жестового керування пристроями розумного будинку».

Автор роботи *Поліщук Ксенія Костянтинівна*.

Керівник роботи: *Грига Володимир Михайлович*.

Пояснювальна записка: 60 с., 30 рис., 4 табл., 3 дод., 52 джерела.

Графічна частина: 3 креслення.

**МІКРОКОНТРОЛЕР ESP32, РОЗПІЗНАВАННЯ ЖЕСТІВ, РОЗУМНИЙ БУДИНОК.**

Мета кваліфікаційної роботи: підвищення ергономічності та доступності систем автоматизації житла шляхом проектування програмно-технічного засобу жестового керування пристроями розумного будинку.

Практичний інтерес даної теми полягає у розробці архітектури розподіленої системи, яка поєднує високорівневу обробку відеоданих за допомогою нейронних мереж та низькорівневе керування виконавчими пристроями через протоколи бездротового зв'язку. Використання сучасних алгоритмів детекції ключових точок кисті дозволяє реалізувати надійну ідентифікацію команд у реальному часі навіть на обчислювальних потужностях побутового сегмента. Водночас застосування мережевої архітектури на базі протоколу MQTT та хмарних технологій забезпечує високу масштабованість системи, дозволяючи інтегрувати в єдину екосистему різноманітні підсистеми, такі як освітлення, кліматичний контроль, безпека та мультимедіа.



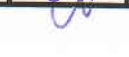
Підпис студента



Дата

## ЗМІСТ

Скорочення та умовні позначки .....	4
Вступ.....	5
1 Огляд технічних засобів для керування пристроями в розумному будинку .....	7
1.1 Розумний будинок та методи керування пристроями.....	7
1.2 Методи розпізнавання жестових комбінацій людини.....	9
1.3 Дослідження та реалізація скелетонних методів розпізнавання жестів.....	13
1.4 Огляд мікроконтролерних платформ для реалізації виконавчих вузлів системи розумного будинку.....	19
1.5 Висновки. Постановка задачі .....	23
2 Проектування програмно-технічного засобу жестового керування пристроями розумного будинку .....	24
2.1 Визначення вимог до програмно-технічного засобу жестового керування пристроями розумного будинку .....	24
2.2 Структура і функціонування програмно-технічного засобу жестового керування пристроями розумного будинку.....	28
2.3 Аналіз та вибір апаратних компонентів для програмно-технічного засобу жестового керування пристроями розумного будинку .....	31
2.4 Огляд та вибір програмних компонентів для проєктованого програмно-технічного засобу .....	37
2.5 Висновки .....	40
3 Реалізація та тестування програмно-технічний засіб жестового керування пристроями розумного будинку .....	41

КВРКІ. 220117.04.22.19 ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата
Виконав		Поліщук К.К.		
Перевір.		Грига В.М.		
Н.контр.		Кисіль Т.М.		
Затвер.		Павлова О.О.		09.06
Програмно-технічний засіб жестового керування пристроями розумного будинку			Літера	Арквш
				2
			Арквшів	60
ХНУ, КІ2-22-4				

3.1 Релазція програмно-технічний засіб жестового керування пристроями розумного будинку.....	41
3.2 Логіка жестової взаємодії та семантика команд керування .....	43
3.3 Створення кластера та налаштування хмарного брокера .....	45
3.4 Обробка відеоданих та реалізація ідентифікації ключових точок кисті на базі MediaPipe.....	48
3.5 Тестування програмно-технічного засобу жестового керування пристроями розумного будинку.....	52
3.6 Висновки .....	57
Висновки .....	59
Перелік джерел посилань .....	61
Додаток А Копія креслення «Структура програмно-технічного засобу жестового керування пристроями розумного будинку ..».....	63
Додаток Б Копія креслення «Результати роботи» .....	64
Додаток В Копія креслення «Схема цифрового двійника виконавчого пристрою у середовищі Wokwi».....	65

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АТЗ – Автономний транспортний засіб

КФС – Кіберфізична система

КС – Комп'ютерна система

ПЗ – Програмне забезпечення

ПЗП – Постійний запам'ятовуючий пристрій

IoT – Internet of Things

NMEA – National Marine Electronics Association

					КВРКІ. 220117.04.22.19 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

## ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімкою інтеграцією інтелектуальних систем у повсякденне життя, що зумовлює трансформацію традиційних методів керування побутовим середовищем. Актуальність розробки програмно-технічних засобів жестового керування диктується необхідністю створення більш природних, інтуїтивно зрозумілих та гігієнічних інтерфейсів взаємодії людини з автоматизованими системами розумного будинку. Традиційні методи контролю, такі як фізичні вимикачі або мобільні додатки, не завжди є зручними в умовах обмеженої мобільності користувача або за необхідності миттєвої безконтактної реакції системи. Впровадження технологій комп'ютерного зору та штучного інтелекту дозволяє перенести керування в площину візуальних образів, де прості рухи рук замінюють складні маніпуляції з пристроями введення, що особливо важливо для підвищення доступності технологій для людей з обмеженими фізичними можливостями та створення нового рівня комфорту в сучасному помешканні.

Практичний інтерес даної теми полягає у розробці архітектури розподіленої системи, яка поєднує високорівневу обробку відеоданих за допомогою нейронних мереж та низькорівневе керування виконавчими пристроями через протоколи бездротового зв'язку. Використання сучасних алгоритмів детекції ключових точок кисті дозволяє реалізувати надійну ідентифікацію команд у реальному часі навіть на обчислювальних потужностях побутового сегмента. Водночас застосування мережевої архітектури на базі протоколу MQTT та хмарних технологій забезпечує високу масштабованість системи, дозволяючи інтегрувати в єдину екосистему різноманітні підсистеми, такі як освітлення, кліматичний контроль, безпека та мультимедіа.

Практичне застосування подібних програмно-технічних засобів відкриває широкі перспективи для модернізації не лише приватних домоволодінь, а й спеціалізованих об'єктів соціальної та промислової інфраструктури. Насамперед, впровадження безконтактних інтерфейсів є критично важливим для створення

					КвРКІ. 220117.04.22.19 ПЗ	Арк. 5
Зм.	Арк.	№докум.	Підпис	Дата		

інклюзивного середовища, де люди з обмеженою мобільністю або порушеннями опорно-рухового апарату отримують можливість самостійно керувати кліматичними та освітлювальними параметрами приміщення без сторонньої допомоги. Окрім соціального аспекту, технологія жестового контролю знаходить своє застосування у приміщеннях з високими вимогами до стерильності, таких як медичні операційні блоки або науково-дослідні лабораторії, де традиційні контактні вимикачі є потенційними джерелами бактеріологічного забруднення.

У побутовому секторі система значно підвищує ефективність взаємодії з приладами в ситуаціях, коли прямий контакт із вимикачами є незручним або неможливим, наприклад, під час приготування їжі чи виконання господарських робіт, що робить процес експлуатації розумного будинку більш природним та ергономічним. Завдяки модульній структурі та використанню стандартних протоколів передачі даних, розроблений засіб може бути інтегрований у складні системи диспетчеризації, де оператор отримує можливість віддаленого контролю за станом обладнання за допомогою інтуїтивних жестів, що мінімізує час реакції на події та знижує когнітивне навантаження. Таким чином, перехід від механічних інтерфейсів до інтелектуальних систем розпізнавання образів стає фундаментом для побудови безпечного, адаптивного та високотехнологічного простору життєдіяльності людини.

Метою роботи є підвищення ергономічності та доступності систем автоматизації житла шляхом проектування програмно-технічного засобу жестового керування пристроями розумного будинку.

Об'єктом дослідження є процеси автоматизації керування виконавчими пристроями розумного будинку на основі безконтактних інтерфейсів взаємодії з використанням технологій комп'ютерного зору.

Предметом дослідження є методи розпізнавання жестів людини, алгоритми обробки відеоданих у реальному часі та програмно-технічні засоби передачі команд керування через розподілену мережеву інфраструктуру на виконавчі мікроконтролери.

					КвРКІ. 220117.04.22.19 ПЗ	Арк. 6
Зм.	Арк.	№докум.	Підпис	Дата		

# 1 ОГЛЯД ТЕХНІЧНИХ ЗАСОБІВ ДЛЯ КЕРУВАННЯ ПРИСТРОЯМИ В РОЗУМНОМУ БУДИНКУ

## 1.1 Розумний будинок та методи керування пристроями

Сучасна парадигма «розумного будинку» розглядається як складна кіберфізична система, що інтегрує різноманітні цифрові пристрої, сенсори та виконавчі механізми в єдине інформаційне середовище за допомогою концепції Інтернету речей (IoT). Фундаментальною основою функціонування такої системи є створення інтелектуального простору, який забезпечує автоматизацію побутових операцій, підвищення рівня безпеки та раціоналізацію використання енергетичних ресурсів. Ключову роль у забезпеченні життєдіяльності цього середовища відіграють методи керування, які еволюціонували від простих механічних перемикачів до складних багаторівневих архітектур, заснованих на алгоритмах штучного інтелекту.

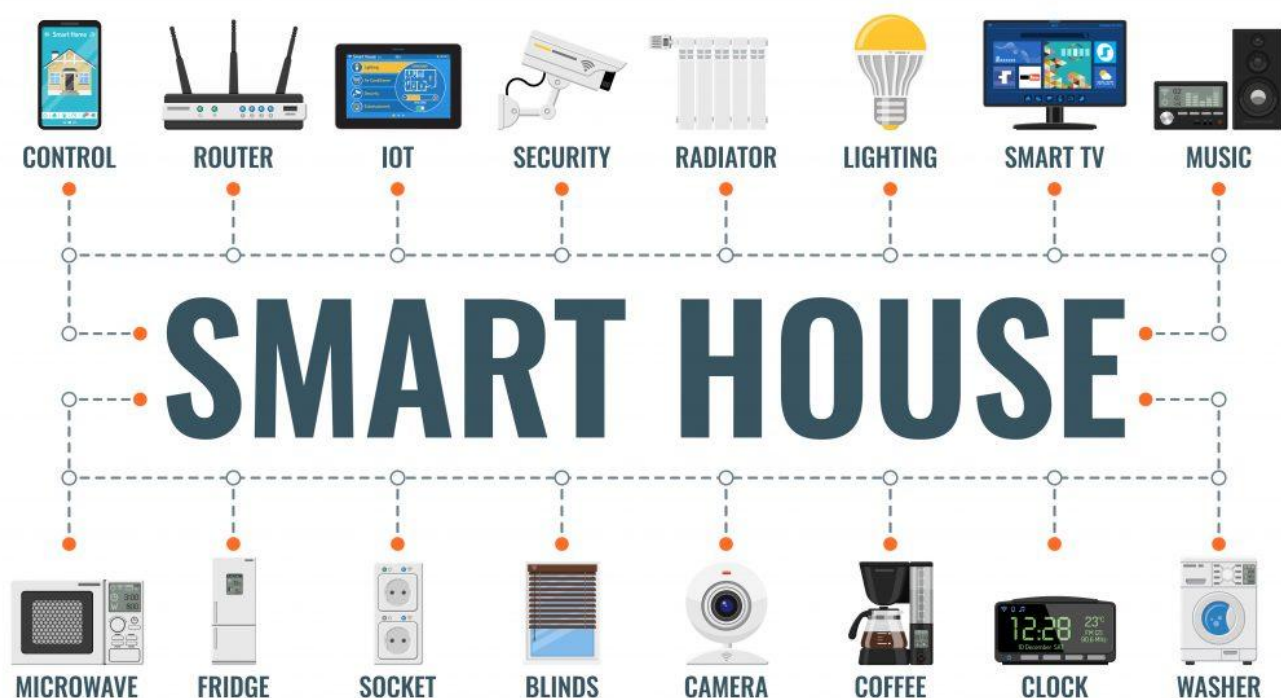


Рисунок 1.1 – Парадигма «Розумний будинок»

Зм.	Арк.	№докум.	Підпис	Дата

На початковому рівні взаємодії переважають методи прямого дистанційного керування, що реалізуються через спеціалізовані мобільні додатки, веб-інтерфейси або локальні пульти (рис. 1.2). Ця модель передбачає безпосередню участь людини в управлінні кожним окремим вузлом системи, де передача команд здійснюється через хмарні платформи або локальні шлюзи. Однак для забезпечення цілісності такої мережі критично важливим є вибір протоколів зв'язку, що визначають надійність та швидкість обміну даними. Серед основних стандартів виділяють Wi-Fi для високошвидкісного з'єднання, а також енергоефективні протоколи Zigbee та Z-Wave, які дозволяють створювати самоорганізовані комерційні мережі з низьким рівнем енергоспоживання. Особливої уваги заслуговує сучасний стандарт Matter, спрямований на вирішення проблеми інтеперабельності, що дозволяє пристроям від різних виробників безперешкодно взаємодіяти в межах однієї екосистеми.



Рисунок 1.2 – Методи керування пристроями у системі «Розумний будинок»

Більш досконалим методом керування є детермінована автоматизація на основі заздалегідь визначених сценаріїв та логічних конструкцій типу «якщо – то». У такому випадку керування пристроями відбувається автономно у відповідь на сигнали від сенсорної мережі: датчиків руху, температури,

вологості, освітленості або витоку газу. Це мінімізує потребу в ручному втручанні, оскільки система самостійно регулює мікроклімат, освітлення або системи безпеки відповідно до встановлених користувачем параметрів. Окрім сенсорного керування, важливим напрямом є мультимодальні інтерфейси взаємодії, зокрема технології розпізнавання природної мови. Голосові асистенти трансформують вербальні команди в цифрові сигнали, що робить процес управління більш ергономічним та доступним, включаючи методи розпізнавання жестів та біометричну ідентифікацію для розмежування прав доступу.

Вищим щаблем розвитку методів керування в розумних будинках є проактивні системи, що базуються на машинному навчанні та предиктивній аналітиці. На відміну від жорстких сценаріїв, такі системи здатні аналізувати величезні масиви даних про поведінкові патерни мешканців, вивчати їхні щоденні звички та самостійно адаптувати функціонування обладнання без явних команд. Наприклад, інтелектуальний контролер може прогнозувати необхідність увімкнення опалення на основі аналізу прогнозу погоди та часу повернення власника додому. Таким чином, сучасна методологія керування пристроями в розумному будинку демонструє стійку тенденцію до децентралізації та повної автономізації, де роль людини поступово зміщується від безпосереднього оператора до стратегічного контролера інтелектуальної екосистеми. Надійність таких методів підкріплюється впровадженням технологій туманних обчислень (Fog Computing), які дозволяють обробляти частину даних локально на крайових пристроях, зменшуючи затримки та підвищуючи рівень кібербезпеки системи.

## 1.2 Методи розпізнавання жестових комбінацій людини

У межах розробки програмно-технічного засобу для керування пристроями розумного будинку ключовим завданням є вибір та обґрунтування методів розпізнавання жестів (Hand Gesture Recognition, HGR). Відповідно до сучасних досліджень, процес розпізнавання функціонально поділяється на два вектори: класифікацію жестів та їх оцінку (рис. 1.3).

					КвРКІ. 220117.04.22.19 ПЗ	Арк. 9
Зм.	Арк.	№докум.	Підпис	Дата		

Класифікація жестів орієнтована на обчислювальний процес виявлення та інтерпретації вхідних даних для їх зіставлення з наперед визначеним набором міток-команд, що дозволяє системі розуміти семантичне значення руху без необхідності детальної цифрової реконструкції.

Натомість методи оцінки жестів спрямовані на визначення просторової структури та топології руки, що включає розрахунок координат суглобів, кутів їхнього згину або побудову детальної 3D-сітки поверхні. Для систем керування побутовою автоматикою найчастіше застосовуються гібридні підходи, де початкова оцінка пози руки слугує базою для подальшої семантичної класифікації команди.



Рисунок 1.3 – Процес розпізнавання: класифікація та оцінка жестів

Важливим аспектом є спосіб захоплення та представлення інформації про руку, де виділяють скелетні моделі та методи на основі обмежувальних рамок (box-based).

Скелетні методи апроксимують руку як граф, де вузли відповідають суглобам, а ребра – кістковим сегментам, що дозволяє ефективно моделювати кінематику рухів, хоча й потребує значних обчислювальних ресурсів для точного виділення ключових точок (рис. 1.4).

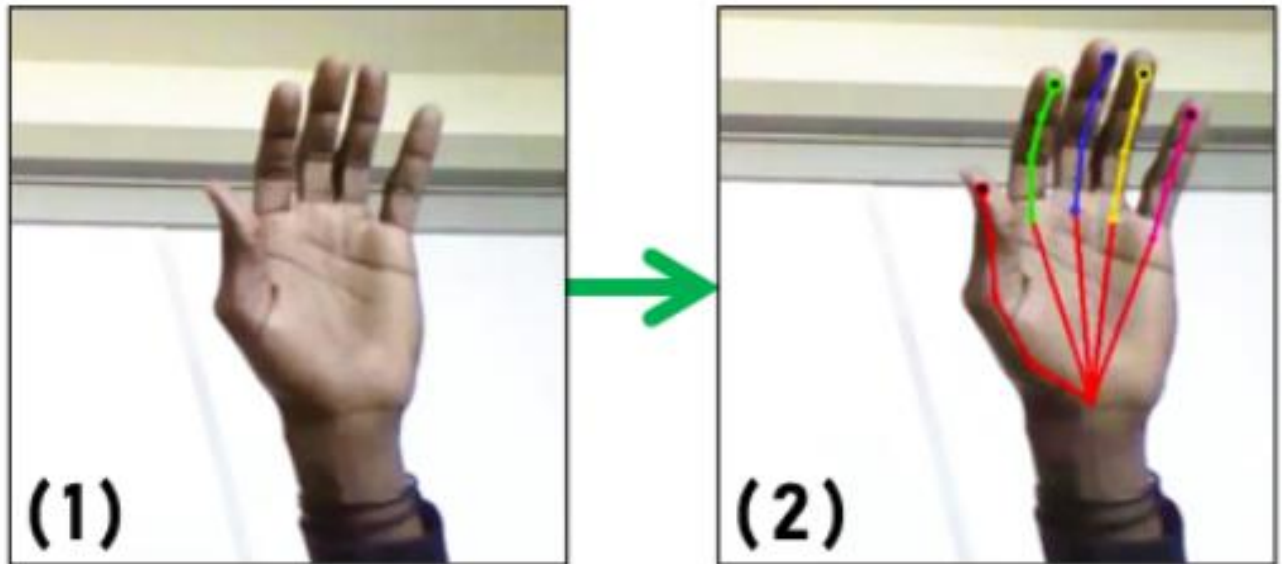


Рисунок 1.4 – Демонстрація методу на основі скелетону

Методи на основі обмежувальних рамок або фільтрів є менш вимогливими до ресурсів, оскільки вони фокусуються на локалізації області руки в кадрі та екстракції ознак форми, текстури або градієнтів орієнтації без деталізації внутрішньої структури (рис. 1.5). Такі підходи демонструють високу ефективність у завданнях класифікації статичних жестів, проте можуть виявляти чутливість до зміни освітлення або складного фону.

Алгоритмічний базис сучасних систем розпізнавання жестів зазнав трансформації від класичних методів машинного навчання до глибоких нейронних мереж. Традиційні підходи, такі як опорні вектори та приховані марковські моделі, тривалий час використовувалися для розпізнавання послідовностей станів жестів, спираючись на ручне проектування ознак. Проте домінуючу позицію посіли згорткові нейронні мережі, які здатні автоматично вивчати просторові ієрархії та складні візуальні ознаки безпосередньо з вхідних RGB-даних.

Для обробки динамічних жестів у відеопотоці CNN інтегруються з рекурентними архітектурами, зокрема мережами довгої короткострокової пам'яті, що дозволяє моделювати часові залежності та зберігати контекст руху протягом тривалих послідовностей кадрів.

Зм.	Арк.	№докум.	Підпис	Дата

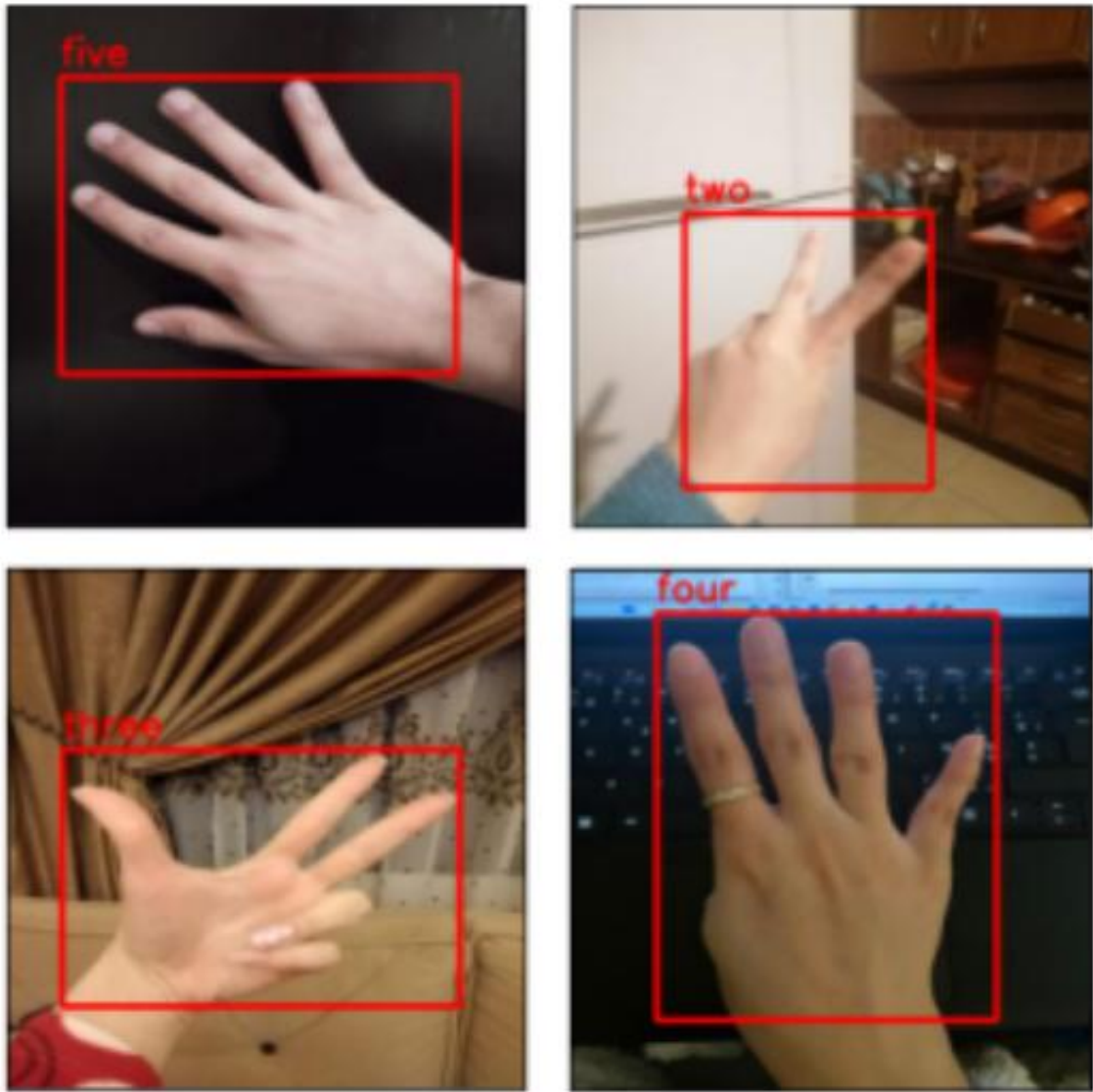


Рисунок 1.5 – Демонстрація методу на основі обмежувальних рамок

Перспективним напрямом розвитку методів розпізнавання є впровадження графових згорткових мереж та трансформерів. Графові згорткові мережі є оптимальними для обробки скелетних даних, оскільки вони дозволяють моделювати неевклідові структурні зв'язки між суглобами пальців, що забезпечує високу точність розпізнавання складної артикуляції. У свою чергу, архітектури на основі трансформерів, завдяки механізмам самоуваги, демонструють виняткову здатність до моделювання довготривалих зв'язків у динамічних жестах та паралельної обробки даних, що робить їх ефективнішими за традиційні рекурентні моделі у завданнях безперервного розпізнавання мови жестів або складних сценаріїв керування.

Зм.	Арк.	№докум.	Підпис	Дата

### 1.3 Дослідження та реалізація скелетонних методів розпізнавання жестів

Скелетонні методи розпізнавання жестів утворюють окремий клас підходів у комп'ютерному зорі, принципова відмінність яких від піксельно-орієнтованих та апаратних методів полягає в тому, що об'єктом аналізу слугує не сам растровий образ кисті, а її абстрактна кінематична модель – упорядкований граф ключових точок, що відповідають анатомічним суглобам. Така абстракція розривала пряму залежність між якістю класифікатора та характеристиками зображення: освітленістю, кольором шкіри, фоновим оточенням і роздільною здатністю камери. Ідея формалізації жесту через скелет кисті виникла ще в ранніх дослідженнях систем розпізнавання мови жестів 1990-х років, коли апаратні датчикові рукавички забезпечували прямий вимір кутів згину суглобів. Однак практична масова реалізація суто візуального скелетонування без спеціалізованого обладнання стала можливою лише з появою достатньо потужних нейронних архітектур та відповідного апаратного забезпечення для їх виконання в режимі реального часу.

Фундаментальна схема будь-якого скелетонного підходу включає чотири послідовних етапи: виявлення регіону інтересу, що відповідає кисті або долоні; локалізацію ключових точок всередині цього регіону; геометричну нормалізацію отриманого скелета; класифікацію нормалізованого вектора ознак. Кожен із цих етапів вирішується відносно незалежними підмодулями, що надає архітектурі модульності та дозволяє замінювати окремі компоненти без перенавчання всього конвеєра. Саме ця властивість відрізняє скелетонний підхід від наскрізних згорткових мереж, які безпосередньо відображають піксельний тензор на мітку класу жесту: у наскрізних архітектурах будь-яка зміна вхідних умов вимагає повного перенавчання.

Першим етапом є виявлення долоні. Він вирішується за допомогою легковагового детектора, архітектурно заснованого на однопрохідних мережах типу SSD з MobileNet як основою. Модель BlazePalm, розроблена командою Google Research та покладена в основу MediaPipe Hands, оперує зображенням у роздільній здатності 128×128 пікселів і повертає обмежувальний прямокутник кисті з

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 13
Зм.	Арк.	№докум.	Підпис	Дата		

субмілісекундною затримкою. Ключовою архітектурною особливістю BlazePalm є використання механізму focal loss при навчанні, що дозволяє ефективно справлятися з різким дисбалансом між кількістю негативних і позитивних прикладів у просторі котрих переглядається. Для мінімізації кількості дорогих повних запусків детектора реалізований механізм трекінгу: якщо у попередньому кадрі вже виявлена долоня, у поточному кадрі детектор запускається лише за умови, що регіон трекінгу втрачає мінімальний поріг впевненості. Такий підхід знижує середнє обчислювальне навантаження більш як утричі порівняно з повним запуском детектора для кожного кадру.

Другий і найбільш концептуально значущий етап – регресія ключових точок. Даний етап виконується мережею-регресором, що приймає на вхід кроп зображення долоні та повертає тривимірні координати фіксованої кількості точок. У топології MediaPipe HandLandmarker таких точок рівно 21, і вони охоплюють зап'ясток, чотири п'ястково-фалангові суглоби, чотири проксимальні міжфалангові суглоби, чотири дистальні міжфалангові суглоби та п'ять кінчиків пальців. Регресор є класичною надглибокою згортковою мережею з residual-з'єднаннями та squeeze-and-excitation блоками, навченою методом змішаного supervised-навчання: частина навчальних прикладів містить реальну 3D-анотацію отриману за допомогою мультикамерного захоплення, решта – синтетичні зображення рук із фотореалістичним рендерингом, для яких тривимірні координати відомі точно. Важливо, що координата z у вихідному представленні не є абсолютною глибиною в метричному просторі, а являє собою відносну глибину нормовану відносно z-координати зап'ястка: від'ємні значення відповідають точкам, що знаходяться ближче до камери ніж зап'ясток, додатні – далі. Це зберігає метричну інформацію в межах кисті без необхідності мати калібровану глибинну камеру.

Геометрична нормалізація скелета відіграє критичну роль у забезпеченні інваріантності класифікатора жестів до масштабу, положення та орієнтації кисті в кадрі. Найпоширеніший підхід полягає в перенесенні початку координат у точку зап'ястка (landmark 0) та нормалізації всіх координат відносно відстані між зап'ястком і середнім MCP-суглобом, яка слугує стабільним проксі розміру кисті.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 14
Зм.	Арк.	№докум.	Підпис	Дата		

Після такого перетворення кожен жест описується вектором відносних координат, що залишається майже незмінним при переміщенні і масштабуванні кисті. Для забезпечення інваріантності до повороту у площині зображення додатково застосовується вирівнювання за вектором між зап'ястком та середнім МСР: скелет повертається так, щоб цей вектор завжди був напрямлений вертикально вгору. Деякі реалізації також виконують симетризацію відносно лівої та правої рук через відображення x-координат, що дозволяє навчати єдиний класифікатор без подвоєння набору даних.

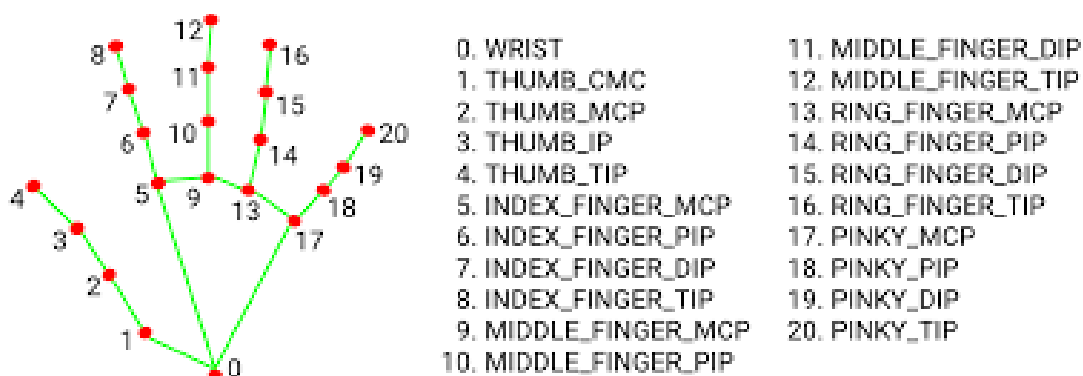


Рисунок 1.6 – Точки захоплення у MediaPipe HandLandmarker

Класифікація нормалізованого скелетного вектора реалізується принципово різними методами залежно від вимог до системи. Найпростішим і найбільш інтерпретованим підходом є правило-орієнтована класифікація на основі порівняння координат: для кожного пальця визначається бінарна ознака «піднятий/опущений» шляхом порівняння y-координати кінчика пальця (tip) із y-координатою відповідного PIP-суглоба. Для вказівного, середнього, безіменного та мізинця умова підняття формулюється як  $tip_y < pip_y$ , оскільки у системі координат зображення вісь y спрямована донизу. Великий палець обробляється окремо через його анатомічну особливість: його рух відбувається переважно в горизонтальній площині, тому стан визначається порівнянням x-координат  $tip[4]$  та  $pip[3]$ . Такий детерміністичний класифікатор дає п'ятибітний вектор стану  $\{f_1, f_2, f_3, f_4, f_5\} \in \{0,1\}^5$ , якому ставиться у відповідність один із 32 теоретично можливих

комбінацій, проте на практиці використовується лише підмножина фізично реалізованих і семантично розрізняваних жестів.

Альтернативою детерміністичному підходу є статистичні класифікатори, що оперують повним нормалізованим вектором координат. Метод k-найближчих сусідів (k-NN) у просторі скелетних ознак демонструє прийнятну точність без навчання параметрів моделі: жест класифікується за мажоритарним голосуванням серед k еталонних скелетів, що знаходяться найближче у евклідовому або зваженому метричному просторі. Проте k-NN чутливий до розмірності простору ознак та потребує зберігання всього навчального набору в пам'яті. Значно частіше у сучасних реалізаціях застосовуються невеликі повнозв'язні нейронні мережі з двома-трьома прихованими шарами, що навчаються на синтетично згенерованих або вручну зібраних скелетних векторах. На відміну від наскрізних CNN-класифікаторів, такий MLP оперує вектором довжини лише 63 (21 точка  $\times$  3 координати) або навіть 42 (без z), що забезпечує мілісекундний час інференції навіть на мікроконтролерах класу ESP32.

Суттєвою проблемою у реальних умовах експлуатації є нестабільність скелетних координат між кадрами, зумовлена як шумом детектора ключових точок, так і природним тремором кисті. Нефільтровані координати призводять до того, що статично утримуваний жест класифікується по-різному у сусідніх кадрах, що є неприйнятним для керуючих застосунків, де кожна класифікована зміна жесту інтерпретується як окрема команда. Для усунення цього явища застосовуються ковзне середнє або фільтри нижніх частот, зокрема найчастіше фільтр Калмана або простий експоненційно зважений ковзний середній із коефіцієнтом згладжування  $\alpha \in [0.3, 0.7]$ . ЕМА для координати  $x_t$  обчислюється за формулою  $\hat{x}_t = \alpha \cdot x_t + (1-\alpha) \cdot \hat{x}_{t-1}$ , де  $x_t$  – сире значення детектора,  $\hat{x}_{t-1}$  – відфільтроване значення попереднього кадру. Менші значення  $\alpha$  дають більш плавну траєкторію ціною збільшення затримки реакції, тому для жестових застосунків зазвичай обирають значення 0.5–0.6 як компроміс між стабільністю та чуйністю.

Поверх тимчасового згладжування ключових точок зазвичай будується другий рівень часової стабілізації – стабілізація самого класифікованого класу

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 16
Зм.	Арк.	№докум.	Підпис	Дата		

жесту. Класичним підходом є мажоритарне голосування у ковзному вікні (majority voting over sliding window) довжиною  $N$  кадрів: жест вважається розпізнаним, якщо одна і та сама мітка перемагає у щонайменше  $M$  із  $N$  останніх кадрів. Альтернативним і частіше вживаним у вбудованих системах є механізм утримання (hold detection): системний таймер запускається при першому виявленні потенційного жесту і скидається при зміні класифікованого класу; жест вважається підтвердженим лише після того, як таймер досяг порогу `HOLD_DURATION`, що зазвичай лежить в діапазоні 0.5–1.5 секунди. Таке проектне рішення природним чином усуває проміжні хибні активації під час переходу між жестами, оскільки транзитний стан не встигає накопичити достатній час утримання. Водночас після підтвердження команди вводиться охолодження (cooldown) – часовий проміжок `ACTION_COOLDOWN`, протягом якого нові команди ігноруються незалежно від стану детектора, що запобігає багаторазовому спрацюванню від одного фізичного жесту.

Важливим аспектом проектування скелетонних систем є вибір між двома архітектурними підходами до розпізнавання динамічних жестів – тих, що визначаються траєкторією руху кисті в часі, а не лише миттєвим положенням. Для статичних жестів (форма кисті у фіксований момент) достатньо поодинокого скелетного вектора, тоді як динамічні жести (помах, коловий рух) вимагають аналізу послідовності. Рекурентні нейронні мережі можуть оброблювати скелетні послідовності довжиною 30–90 кадрів, навчаючись виявляти часові патерни безпосередньо зі зміни координат. Альтернативним підходом є кодування траєкторії через похідні першого та другого порядків (швидкість та прискорення) і конкатенація цих ознак із поточним статичним вектором перед подачею до MLP-класифікатора. Такий підхід суттєво спрощує архітектуру, не вимагаючи рекурентних шарів, і при цьому зберігає частину динамічної інформації.

Для застосунків жестового керування у системах розумного будинку, де множина команд є наперед відомою і порівняно невеликою (5–15 жестів), доцільним є гібридний підхід: статичні позиційні жести для вибору пристрою та виконання команд «увімк/вимк», а один–два динамічних жести – для скасування

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 17
Зм.	Арк.	№докум.	Підпис	Дата		

вибору або переходу в інший режим. Такий підхід дозволяє уникнути складної класифікації динамічних послідовностей для більшості взаємодій, залишаючи рекурентні методи лише для тих жестів, що не можуть бути семантично розрізнені статично. Наприклад, жест відкритої долоні (всі п'ять пальців підняті) та жест «коза» (вказівний і мізинець підняті, решта опущені) мають принципово різний п'ятибітний код стану, тому статичний класифікатор безпомилково їх розрізняє, тоді як жест «помах» і жест «привіт» мають ідентичний статичний код і розрізняються виключно за наявністю або відсутністю відносного переміщення кисті у часі.

Порівняльний аналіз скелетонних методів із суміжними підходами засвідчує їхні очевидні переваги та ряд обмежень. Перевагою є висока інваріантність до зовнішніх умов освітлення, кольору шкіри та фону – властивість, якої позбавлені методи сегментації шкіри та оптичного потоку. Скелетонне представлення є компактним і легко інтерпретованим, що спрощує налагодження та розширення набору жестів. Порівняно з апаратними датчикорними системами (рукавичками, інерційними датчиками) суто візуальне скелетонування не потребує носіння будь-якого обладнання і виключає проблему калібрування, зношення та зарядки акумулятора сенсора. Основним обмеженням є залежність якості скелетизатора від умов оклюзії: при суттєвому перекриванні пальців один одним або зовнішніми об'єктами детектор ключових точок повертає некоректні або нестабільні координати, що безпосередньо погіршує точність класифікації. Цей ефект посилюється при роботі з жестами, де пальці формують щільно зімкнений кулак, оскільки їхні кінчики стають невидимими для камери. Ще одним обмеженням є обчислювальна вартість повного конвеєра на малопотужних пристроях: дворівневий детектор (BlazePalm + регресор точок) вимагає апаратного прискорення, що доступне в сучасних мобільних SoC та одноплатних комп'ютерах із NPU, але може бути недостатнім для мікроконтролерних платформ без спеціалізованих обчислювальних блоків.

Таким чином, скелетонні методи розпізнавання жестів є на сьогодні найбільш збалансованим підходом для реалізації безконтактних жестових інтерфейсів у

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 18
Зм.	Арк.	№докум.	Підпис	Дата		

застосунках розумного будинку: вони поєднують прийнятну обчислювальну складність, достатню точність в умовах реального приміщення та природну масштабованість набору підтримуваних жестів без повного перенавчання системи.

#### 1.4 Огляд мікроконтролерних платформ для реалізації виконавчих вузлів системи розумного будинку

Виконавчий вузол системи розумного будинку є апаратним компонентом, що безпосередньо взаємодіє з керованим обладнанням – реле, диммерами, сервоприводами, вентиляційними модулями та системами сигналізації. На відміну від клієнтської частини, що відповідає за розпізнавання жестів та публікацію команд до брокера, виконавчий вузол вирішує зворотну задачу: підписується на відповідні MQTT-топіки та перетворює прийняте повідомлення на фізичну дію. Вибір апаратної платформи для такого вузла визначається декількома ключовими критеріями: наявністю вбудованого мережевого стеку для підтримки протоколу MQTT поперх Wi-Fi або іншого бездротового каналу, достатньою обчислювальною потужністю для обслуговування TLS-з'єднання із хмарним брокером, прийнятним енергоспоживанням у режимі очікування та низькою вартістю одиниці, що є критичним при масштабуванні системи на велику кількість приміщень.

Найбільш поширеною у прикладах побутової автоматизації класу розумний будинок є родина мікроконтролерів Arduino на базі AVR-архітектури, зокрема Arduino Uno із мікроконтролером ATmega328P. Ця платформа виконала важливу освітню роль і сформувала навколо себе обширну екосистему бібліотек та прикладів, проте має принципові архітектурні обмеження для IoT-застосунків: відсутність вбудованого мережевого модуля, у край обмежений обсяг оперативної пам'яті (2 КБ SRAM) та тактова частота 16 МГц, що не дозволяє апаратно прискорювати TLS-рукописання. Підключення Wi-Fi до Arduino Uno через зовнішній шилд ESP8266 формально розв'язує задачу мережевої з'єднаності, однак вносить додатковий рівень складності та залежність від UART-інтерфейсу між двома мікроконтролерами, що підвищує латентність і ускладнює відлагодження.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 19
Зм.	Арк.	№докум.	Підпис	Дата		

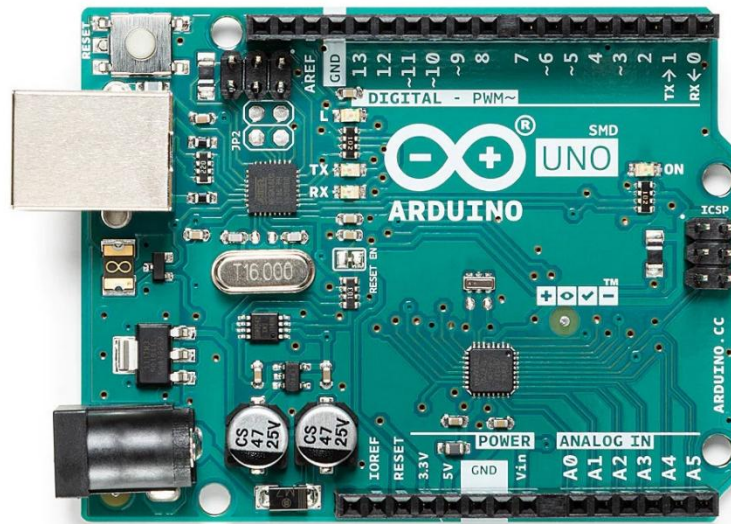


Рисунок 1.7 – Мікроконтролер Arduino на базі AVR-архітектури

Мікросхема ESP8266, що спочатку позиціювалася як недорогий Wi-Fi-комп'ютер для Arduino, згодом перетворилася на самостійну платформу завдяки наявності повноцінного 32-розрядного процесора Xtensa L106 із тактовою частотою 80 МГц та вбудованого TCP/IP-стеку. Модульне виконання NodeMCU на базі ESP8266 стало надзвичайно популярним для реалізації простих IoT-вузлів із підключенням до хмарних сервісів. Проте ESP8266 має суттєві обмеження: одне процесорне ядро, відсутність Bluetooth, обсяг SRAM лише 80 КБ (при цьому значна його частина резервується під Wi-Fi-стек, залишаючи застосунку близько 36–50 КБ), а також відомі труднощі зі стабільністю роботи при одночасному активному Wi-Fi та GPIO-операціях. Ці фактори роблять ESP8266 прийнятним рішенням для найпростіших сенсорних вузлів, але недостатнім для виконавців із вимогами до надійності та багатозадачності.

Принципово якіснішою альтернативою виступає мікроконтролер ESP32 виробництва Espressif Systems, який є прямим наступником ESP8266 із радикально розширеними характеристиками. ESP32 оснащений двоядерним процесором Xtensa LX6 із тактовою частотою до 240 МГц, 520 КБ SRAM, підтримкою Wi-Fi 802.11 b/g/n та Bluetooth 4.2 / BLE на одному кристалі. Наявність двох ядер має принципове значення для IoT-застосунків: одне ядро може бути повністю виділене

Зм..	Арк.	№докум.	Підпис	Дата

під обслуговування Wi-Fi-стеку та TLS-з'єднання із брокером, тоді як друге ядро виконує логіку застосунку – опитування MQTT-топиків, управління GPIO та обробку команд. Такий розподіл унеможлиблює ситуацію, коли мережева активність блокує своєчасну реакцію на прийняту команду, що є критичним для керування такими пристроями, як сигналізація або вентиляція. ESP32 підтримує апаратне прискорення AES та SHA, що суттєво знижує навантаження на процесор при встановленні TLS-сесії із хмарним брокером HiveMQ Cloud, який вимагає шифрування на рівні транспортного протоколу.

Варто окремо відзначити модифікацію ESP32-S3, що з'явилася у 2021 році та замінила ядра Xtensa LX6 на Xtensa LX7 із підтримкою векторних розширень. Ця версія орієнтована насамперед на задачі машинного навчання на пристрої (TinyML) і здатна виконувати прості моделі класифікації безпосередньо на мікроконтролері без звернення до хмари. У контексті систем жестового керування це відкриває перспективу перенесення частини логіки класифікації жестів на сам виконавчий вузол, хоча обчислювальних ресурсів ESP32-S3 недостатньо для запуску повного конвеєра MediaPipe HandLandmarker він залишається у компетенції клієнтської частини з повноцінним CPU або GPU.

Мікроконтролери сімейства STM32 виробництва STMicroelectronics на ядрі ARM Cortex-M широко застосовуються у промисловій автоматизації та вбудованих системах із жорсткими вимогами до детермінованості часу відгуку. Модель STM32F103 (відома у ринку як «Blue Pill») пропонує процесор Cortex-M3 на 72 МГц із 20 КБ SRAM і 128 КБ Flash за вартістю близько 2 USD, що робить її найдешевшим варіантом серед розглянутих. Однак, як і класичні Arduino, STM32-мікроконтролери не мають вбудованого мережевого модуля, що вимагає підключення зовнішнього Wi-Fi або Ethernet-чіпу і суттєво ускладнює схему для IoT-застосування. У системах розумного будинку, мікроконтролер STM32 доцільно використовувати як локальний виконавчий контролер нижнього рівня, що отримує команди від ESP32-шлюзу через UART або SPI, а не напряму від MQTT-брокера.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 21
Зм.	Арк.	№докум.	Підпис	Дата		



та фреймворку ESP-IDF. Низька вартість модуля (орієнтовно 500 грн) у поєднанні зі зрілістю програмного стеку і активною спільнотою розробників робить ESP32 де-факто стандартом для прототипування та дрібносерійного виробництва IoT-вузлів рівня розумного будинку. Саме тому у симуляційній частині даної роботи виконавчий вузол реалізований на базі ESP32 у середовищі Wokwi, яке підтримує повну емуляцію Wi-Fi-стеку та бібліотеки PubSubClient для протоколу MQTT.

## 1.5 Висновки. Постановка задачі

Розробка програмно-технічного засобу жестового керування пристроями розумного будинку на основі технологій комп'ютерного зору та протоколу MQTT є актуальним завданням, що відповідає сучасним тенденціям розвитку штучного інтелекту та концепції Інтернету речей. Це забезпечує створення інтуїтивно зрозумілих інтерфейсів безконтактної взаємодії, що суттєво підвищує рівень комфорту, гігієнічності та безпеки сучасного житла. Такі рішення стають фундаментальною частиною екосистеми інтелектуального простору, забезпечуючи інклюзивність для людей з обмеженими можливостями та дозволяючи ефективно інтегрувати різноманітні побутові підсистеми у єдину цифрову інфраструктуру з можливістю моніторингу та керування в реальному часі.

Для вирішення цього завдання слід виконати ряд етапів, що включають:

- 1) дослідження концепції «Розумний будинок» та методів керування пристроями та методів розпізнавання жестових комбінацій людини;
- 2) визначення вимог до програмно-технічного засобу жестового керування пристроями розумного будинку;
- 3) проєктування структури програмно-технічного засобу жестового керування пристроями розумного будинку;
- 4) аналіз та вибір апаратних та програмних компонентів для програмно-технічного засобу жестового керування пристроями розумного будинку;
- 5) релазція програмно-технічного засобу жестового керування пристроями розумного будинку, його симуляція, тестування та оцінка ефективності.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 23
Зм.	Арк.	№докум.	Підпис	Дата		

## 2 ПРОЕКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ЖЕСТОВОГО КЕРУВАННЯ ПРИСТРОЯМИ РОЗУМНОГО БУДИНКУ

### 2.1 Визначення вимог до програмно-технічного засобу жестового керування пристроями розумного будинку

Проектований програмно-технічний засіб повинен забезпечувати можливість селективного безконтактного керування чотирма ключовими підсистемами розумного будинку: освітленням, вентиляцією, безпекою (сигналізацією) та мультимедійним середовищем. Технічна реалізація цих підсистем передбачає взаємодію з п'ятьма функціональними вузлами, що мають різні алгоритми роботи. Зокрема, підсистема освітлення є комплексною і включає два окремі об'єкти керування – світильник для виконання базових операцій увімкнення та вимкнення, а також окремий вузол регулювання яскравості для плавного налаштування інтенсивності світлового потоку. Інші підсистеми представлені одиничними вузлами: вентилятором, модулем активації сигналізації та системою регулювання гучності музики. Підсистеми керування та об'єкти керування, що реалізуються в запропонованому програмно-технічному засобі жестового керування наведені у таблиці 2.1.

Таблиця 2.1 – Підсистеми керування та об'єкти керування

№	Підсистема Розумного будинку	Об'єкт керування
1	Освітлення	Світильник (виконання базових операцій увімкнення та вимкнення)
		Регулювання яскравості світильника
2	Вентиляція	Вентилятор
3	Безпека (сигналізація)	Модуль активації сигналізації
4	Мультимедійне середовище	Система регулювання гучності музики

Зм.	Арк.	№докум.	Підпис	Дата

Для забезпечення ефективної взаємодії система повинна розрізняти типи команд залежно від цільового вузла. Для світильника, вентилятора та сигналізації основною вимогою є підтримка стабільного перемикачів бінарних станів (ON/OFF), що дозволяє оперативно змінювати режим роботи пристроїв. Водночас для вузлів регулювання яскравості та гучності музики програмно-технічний засіб має реалізовувати інкрементальне керування, забезпечуючи можливість плавного збільшення або зменшення значень у заданому діапазоні. Це вимагає від алгоритму розпізнавання жестів здатності не лише ідентифікувати обраний об'єкт, а й коректно обробляти динаміку жестів для точного налаштування параметрів комфорту.

Для реалізації обчислювальної логіки програмна частина, розгорнута на базі мікрокомп'ютера Raspberry Pi, має здійснювати безперервний аналіз відеопотоку, детекцію ключових точок кисті руки та класифікацію жестів. Важливою вимогою є впровадження дворівневого алгоритму взаємодії, де спочатку за допомогою статичної конфігурації пальців здійснюється вибір цільового пристрою, а після підтвердження вибору виконання конкретної дії через динамічні жести великим пальцем. Це дозволить мінімізувати ймовірність випадкових спрацювань та забезпечить чітку послідовність команд у складному середовищі розумного будинку.

Однією з ключових вимог до архітектури системи є організація надійної передачі даних між обчислювальним вузлом та виконавчими механізмами через хмарний брокер HiveMQ за протоколом MQTT. Програмний засіб повинен формувати структуру топіків для кожного пристрою, що дозволить виконавчому мікроконтролеру ESP32 оперативно отримувати команди та трансформувати їх у відповідні електричні сигнали.

Крім того, система має підтримувати гібридну модель керування, яка передбачає можливість як жестового контролю, так і прямого ручного керування через зовнішній додаток. Це вимагає від засобу здатності синхронізувати поточний стан усіх пристроїв у реальному часі, незалежно від джерела надходження керуючого сигналу.

					КвРКІ. 220117.04.22.19 ПЗ	Арк. 25
Зм.	Арк.	№докум.	Підпис	Дата		

Додатково програмно-технічний засіб повинен передбачати механізми захисту та зворотного зв'язку для підвищення зручності користувача. Це включає реалізацію програмних фільтрів для ігнорування короточасних жестів, встановлення часових інтервалів для підтвердження наміру дії, а також візуалізацію процесу керування через графічний оверлей на відеопотоці. Такий підхід дозволить користувачеві бачити поточний вибраний пристрій та прогрес виконання команди, що є критично важливим для систем без фізичних кнопок. Основні функціональні та технічні вимоги до програмно-технічного засобу жестового керування наведені у таблиці 2.2.

Таблиця 2.2 – Основні функціональні та технічні вимоги до програмно-технічного засобу жестового керування.

Категорія вимог	Зміст вимоги	Технічні параметри
Функціональні	Кількість підсистем керування	4 підсистеми: освітлення, вентиляція, сигналізація, мультимедіа (звук)
	Об'єкти (вузли) керування	5 вузлів: світильник (вкл/викл), яскравість (діапазон), вентилятор (вкл/викл), сигналізація (вкл/викл), гучність (діапазон)
	Типи команд	Бінарні (ON/OFF) та інкрементальні (Step UP/DOWN) для регулювання рівнів
Алгоритмічні	Розпізнавання образів	Детекція 21 ключової точки кисті руки в реальному часі (за допомогою бібліотеки MediaPipe)

Продовження таблиці 2.2

	Логіка вибору та дії	Двоетапна: вибір пристрою статичним жестом (кількість пальців) та зміна стану динамічним (великий палець)
	Захист від помилок	Наявність програмних фільтрів та інтервалів відновлення між діями
Комунікаційні	Протокол передачі даних	MQTT для зв'язку між RPi та ESP32
	Мережева інфраструктура	Використання хмарного брокера (HiveMQ) з обов'язковим шифруванням TLS
	Спосіб керування пристроями Розумного будинку	Можливість паралельного керування через жести та мобільний додаток із синхронізацією станів
Програмно-технічні	Апаратна база	Обчислювальний вузол на Raspberry Pi та виконавчі контролери на базі ESP32
	Відеопотік	Обробка зображення з камери з частотою не менше 10-15 FPS
	Візуалізація стану розпізнавання жесту	Візуалізація стану системи через графічний інтерфейс на дисплеї
Експлуатаційні	Швидкодія	Затримка від моменту розпізнавання жесту до реакції пристрою не повинна перевищувати 200-500 мс.

## 2.2 Структура і функціонування програмно-технічного засобу жестового керування пристроями розумного будинку

Структура розроблюваного програмно-технічного засобу базується на принципах розподіленої архітектури Інтернету речей (IoT), що дозволяє розмежувати ресурсомісткі обчислювальні процеси та операції прямого керування апаратними компонентами.

Загальна конфігурація пропонованого програмно-технічного засобу охоплює залучення чотирьох основних рівнів:

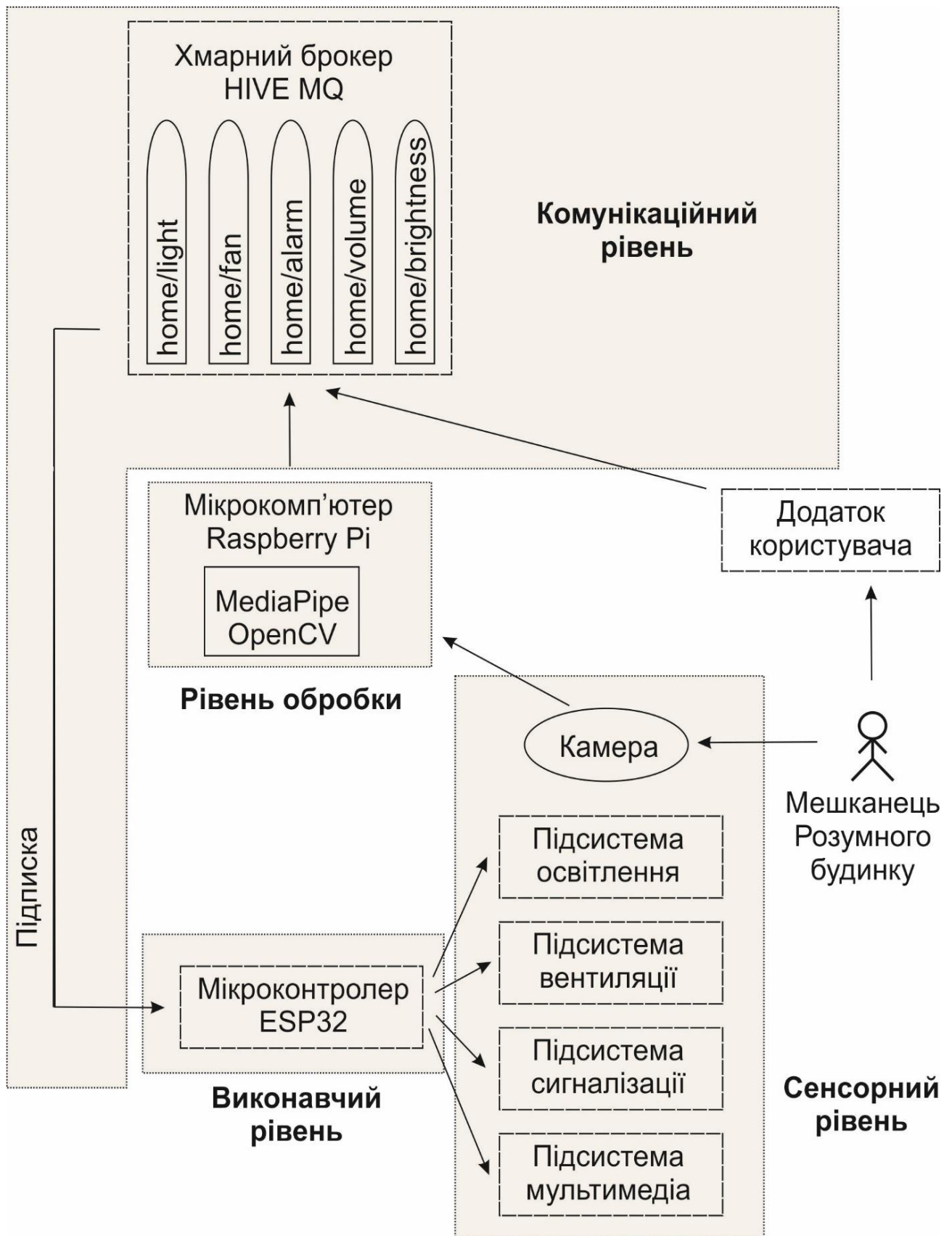
1. сенсорний інтерфейс;
2. центральний вузол обробки даних;
3. комунікаційне середовище;
4. виконавчий рівень.

На рисунку 2.3 наведено структуру програмно-технічного засобу жестового керування пристроями розумного будинку.

Кожен із цих рівнів виконує специфічний набір функцій, що забезпечують перетворення візуальної інформації про жести користувача у фізичні дії кінцевих пристроїв розумного будинку.

Функціонування системи починається на сенсорному рівні, де за допомогою відеокамери здійснюється захоплення зображення робочої зони користувача. Отриманий відеопотік у реальному часі передається до центрального обчислювального вузла, реалізованого на базі мікрокомп'ютера Raspberry Pi, оснащеного дисплеєм для забезпечення інтерактивної взаємодії та візуального моніторингу процесів розпізнавання.

Основною функцією центрального обчислювального вузла є інтелектуальна обробка кадрів із використанням алгоритмів комп'ютерного зору. Програмна частина на базі бібліотек MediaPipe та OpenCV проводить ідентифікацію кисті руки, визначає просторові координати 21-ї ключової точки та проводить аналіз їхньої взаємної конфігурації.



Рисунко 2.3 – Структура програмно-технічного засобу жестового керування пристроями розумного будинку

Зм.	Арк.	№докум.	Підпис	Дата

Паралельно з обчислювальним процесом система здійснює рендеринг графічного оверлею на підключеному дисплеї, де в режимі реального часу відображається скелетна модель руки, виявлені жести та службова інформація щодо поточного обраного пристрою. Це дозволяє користувачеві оперативно отримувати підтвердження правильності виконання рухів та коригувати свої дії у разі потреби. На основі розпізнаних патернів програма реалізує логіку вибору одного з п'яти функціональних вузлів та формує відповідну команду керування залежно від динаміки жестів, забезпечуючи прозорість та наочність процесу безконтактного контролю.

Комунікаційний рівень системи виконує роль сполучної ланки та базується на використанні хмарного брокера повідомлень HiveMQ. Функціонування цього рівня організовано за протоколом MQTT, де одноплатна комп'ютерна система Raspberry Pi виступає у ролі видавця (publisher), що відправляє сформовані команди у відповідні топіки, такі як home/light або home/volume. Це забезпечує високу швидкість передачі даних та дозволяє реалізувати гібридну модель керування, при якій брокер може приймати команди одночасно від модуля розпізнавання жестів та від мобільного додатка користувача, підтримуючи єдину чергу повідомлень для кожного об'єкта.

Виконавчий рівень системи представлений мікроконтролером ESP32, який забезпечує безпосередню взаємодію із пристроями розумного будинку та виконує роль керуючого вузла для реалізації фізичних дій. Основною функцією мікроконтролера є постійне прослуховування повідомлень, що надходять від хмарного брокера за протоколом MQTT. Для цього ESP32 виконує операцію підписки на визначені топіки брокера та перебуває у режимі очікування нових повідомлень. Після надходження керуючої команди від клієнтського застосунку мікроконтролер здійснює обробку отриманих даних та формує відповідний сигнал керування для підключених пристроїв.

Отримана команда перетворюється мікроконтролером у конкретну фізичну дію, що виконується через апаратні інтерфейси вводу-виводу. Зокрема, за допомогою цифрових логічних виходів здійснюється керування такими

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 30
Зм.	Арк.	№докум.	Підпис	Дата		

пристроями, як вентилятор, освітлювальні прилади або система сигналізації. Крім того, для більш гнучкого керування параметрами обладнання використовується широтно-імпульсна модуляція (PWM), яка дозволяє плавно змінювати рівень яскравості освітлення або регулювати гучність відтворення музики. Завдяки такому підходу забезпечується оперативне та точне виконання команд, сформованих системою розпізнавання жестів, що підвищує ефективність і зручність керування пристроями розумного будинку.

### 2.3 Аналіз та вибір апаратних компонентів для програмно-технічного засобу жестового керування пристроями розумного будинку

Ефективність функціонування програмно-технічного засобу жестового керування безпосередньо залежить від раціонального вибору апаратної бази, яка повинна забезпечувати оптимальний баланс між обчислювальною потужністю, енергоефективністю та надійністю мережевої взаємодії. Ключовим критерієм при аналізі компонентів виступає здатність системи в реальному часі обробляти складні алгоритми комп'ютерного зору, що висуває високі вимоги до центрального обчислювального вузла.

У ролі центрального сервера та головного обчислювального вузла системи обрано мікрокомп'ютер Raspberry Pi, що зумовлено необхідністю поєднання компактності з високою продуктивністю для виконання завдань штучного інтелекту. Вибір саме цієї платформи є стратегічно обґрунтованим у порівнянні з аналогами, такими як Orange Pi або Banana Pi. Хоча Orange Pi часто пропонує вищу тактову частоту за нижчу ціну, Raspberry Pi суттєво переважає конкурентів у стабільності програмного забезпечення та підтримці бібліотек комп'ютерного зору. Для реалізації жестового керування критично важливою є коректна робота фреймворків MediaPipe та OpenCV, які мають найкращу оптимізацію саме під архітектуру та операційну систему Raspberry Pi OS. Крім того, наявність величезної спільноти та детальної документації дозволяє оперативно вирішувати питання

сумісності драйверів для камер та периферійних модулів, що часто є проблемним місцем для китайських аналогів.

Особлива увага при виборі приділялася можливостям візуалізації та розширення. Raspberry Pi оснащений спеціалізованим роз'ємом DSI та двома портами Micro-HDMI, що дозволяє підключити сенсорний дисплей для відображення графічного оверлею в реальному часі. Це перетворює пристрій на повноцінний інтерактивний термінал розумного будинку, де користувач може не лише подавати команди жестами, а й отримувати миттєвий візуальний зворотний зв'язок. Наявність 40-пінового роз'єму GPIO також забезпечує потенціал для подальшого апаратного розширення системи безпосередньо на серверному рівні.

Для забезпечення безперебійної обробки відеопотоку з частотою не менше 15–20 FPS було обрано модель Raspberry Pi 4 Model B з 8 ГБ оперативної пам'яті, технічні характеристики якої наведені нижче:

Процесор: Broadcom BCM2711, 4-ядерний ARM Cortex-A72 (64-bit), тактова частота до 1.5 ГГц

- оперативна пам'ять: 8 ГБ LPDDR4;
- графіка: VideoCore VI, підтримка OpenGL ES 3.0;
- відеовиходи: 2 × micro-HDMI з підтримкою роздільної здатності до 4K;
- аудіо: через HDMI та 3.5 мм аудіороз'єм (комбінований);
- USB-інтерфейси: 2 × USB 3.0, 2 × USB 2.0;
- мережева підтримка: Gigabit Ethernet;
- бездротові інтерфейси: Wi-Fi 802.11 b/g/n/ac (2.4 та 5 ГГц), Bluetooth 5.0, BLE;
- сховище даних: слот для карти microSD;
- інтерфейси розширення: 40-контактний GPIO, CSI (камера), DSI (дисплей);
- живлення: 5 В через USB-C (рекомендовано 3 А);
- підтримка PoE: через окремий PoE HAT.



Рисунок 2.4 – Raspberry Pi 4 Model B

Для забезпечення візуалізації процесу керування та надання користувачеві оперативного зворотного зв'язку в межах проєкту обрано офіційний 7-дюймовий сенсорний дисплей Raspberry Pi Touch Screen Display. Цей компонент відіграє головну роль у реалізації графічного оверлею, який накладається на відеопотік у реальному часі, дозволяючи користувачеві чітко бачити ідентифіковані ключові точки кисті, поточний вибраний об'єкт керування та динаміку зміни параметрів, таких як яскравість чи гучність. Завдяки роздільній здатності 800x480 пікселів та підтримці десятипальцевого сенсорного введення, дисплей не лише слугує засобом моніторингу, а й повноцінно забезпечує виконання вимоги щодо гібридного керування, дозволяючи дублювати жестові команди прямим ручним маніпулюванням інтерфейсом у випадках, коли безконтактна взаємодія є незручною.

Технічна інтеграція дисплея здійснюється через спеціалізований порт DSI на платі Raspberry Pi, що гарантує високу пропускну здатність для виводу графіки без навантаження на основні інтерфейси передачі даних та дозволяє залишити вільними контакти GPIO для можливого розширення функціоналу. Використання саме цього модуля дозволяє створити ергономічний та компактний центральний вузол керування, де програмні алгоритми обробки зображень та графічний інтерфейс користувача об'єднані в єдиний апаратно-програмний комплекс.

Зм.	Арк.	№докум.	Підпис	Дата



Рисунок 2.5 – Монітор Raspberry Pi 7" Touch Screen Display

Для забезпечення високоточного введення візуальних даних у системі жестового керування обрано модуль камери Raspberry Pi Camera Module 3 Wide. Основною перевагою даного апаратного компонента є використання ширококутної оптики з кутом огляду 120 градусів, що є критично важливим для побудови інтерфейсів розумного будинку. Така конфігурація дозволяє значно розширити робочу зону детекції, надаючи користувачеві можливість здійснювати керування підсистемами освітлення чи мультимедіа з різних точок приміщення без необхідності суворого позиціювання руки безпосередньо перед об'єктивом. Така особливість є значною перевагою даної камери у порівнянні із іншими. Також завдяки сенсору Sony IMX708 з підтримкою технології HDR, камера забезпечує стабільну якість зображення навіть у складних умовах освітлення, як-от при наявності яскравих вікон на фоні або в сутінках, що гарантує надійну сегментацію кисті руки алгоритмами комп'ютерного зору.

Зм..	Арк.	№докум.	Підпис	Дата

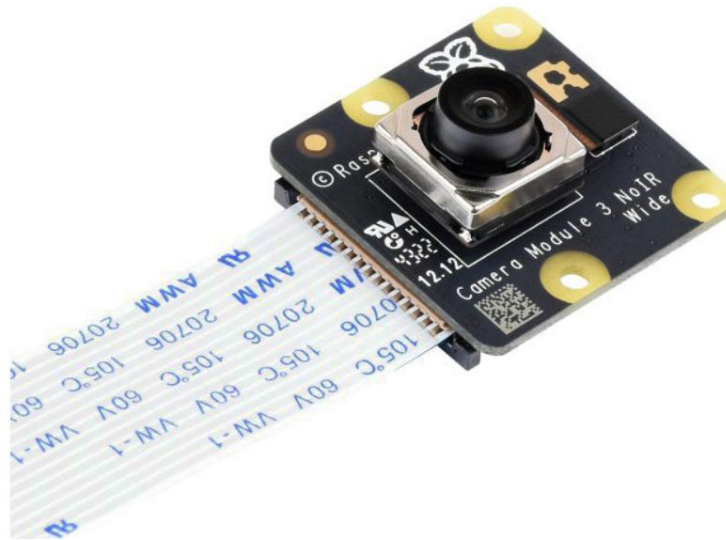


Рисунок 2.6 – Ширококутна камера Raspberry Pi Camera Module 3 Wide NoIR

Технічна реалізація дворівневого алгоритму взаємодії, що передбачає відстеження мікрорухів великого пальця для інкрементального регулювання яскравості чи гучності, висуває високі вимоги до чіткості зображення. Наявність вбудованої системи швидкого фазового автофокусу у Camera Module 3 дозволяє камері миттєво адаптуватися до зміни відстані між користувачем і пристроєм у діапазоні від декількох сантиметрів до трьох метрів. Це забезпечує постійну різкість контурів долоні та стабільну ідентифікацію ключових точок скелета руки, мінімізуючи тремтіння координат у відеопотоці. Підключення через спеціалізований інтерфейс CSI дозволяє передавати дані безпосередньо до графічного процесора мікрокомп'ютера Raspberry Pi з мінімальними затримками, що є вирішальним фактором для досягнення високої частоти кадрів та миттєвого відгуку виконавчих пристроїв на команди користувача.

Для безпосередньої взаємодії з виконавчими механізмами підсистем освітлення, вентиляції та безпеки в проекті обрано мікроконтролер ESP32 (рис. 2.5). Головною перевагою цього рішення є наявність вбудованого модуля Wi-Fi, що дозволяє реалізувати повністю бездротову архітектуру керування через протокол MQTT. На відміну від менш потужних аналогів, ESP32 має двоядерну архітектуру, що дає змогу розділити завдання підтримки стабільного мережевого з'єднання з брокером HiveMQ та безпосередню генерацію керуючих сигналів для периферії.

Це забезпечує мінімальну затримку між моментом розпізнавання жесту на Raspberry Pi та фізичним спрацюванням пристрою, що є критичним для комфортного сприйняття системи користувачем.



Рисунок 2.7 – Мікроконтролер ESP32

Основні характеристики мікроконтролера ESP32 включають в себе:

- процесор: двоядерний 32-бітний Xtensa® LX6 із частотою до 240 МГц;
- пам'ять: 520 КБ внутрішньої оперативної пам'яті (SRAM) та підтримка зовнішньої Flash-пам'яті (зазвичай 4 МБ);
- бездротовий зв'язок: вбудовані модулі Wi-Fi (802.11 b/g/n) та Bluetooth (Dual-mode: Classic + BLE);
- периферія (GPIO): 34 програмовані порти введення-виведення з підтримкою переривань;
- ШІМ (PWM): до 16 незалежних каналів для плавного керування яскравістю світла та швидкістю вентиляторів;
- аналого-цифрове перетворення: 12-бітний АЦП (до 18 каналів) для зчитування показників із датчиків;
- інтерфейси: підтримка протоколів передачі даних UART, SPI та I2C;

Зм.	Арк.	№докум.	Підпис	Дата

– безпека: апаратне прискорення шифрування (AES, SHA-2, RSA, ECC) для захищеної передачі даних;

– живлення: робоча напруга 3.3 В з підтримкою режимів глибокого енергозбереження.

Таким чином, обрані апаратні компоненти складають основу проєктованого програмно-технічного засобу жестового керування пристроями розумного будинку, забезпечуючи необхідну обчислювальну потужність у реальному часі та надійну інтеграцію з периферійними вузлами через хмарну інфраструктуру.

#### 2.4 Огляд та вибір програмних компонентів для проєктованого програмно-технічного засобу

Програмна складова розроблюваного проєкту головним чином передбачала вибір та аналіз бібліотек для розпізнавання рухів та вибір брокера MQTT, через який здійснювалась передача повідомлень у проєктованого програмно-технічному засобі.

Для реалізації алгоритмів комп'ютерного зору та розпізнавання жестів ключовим програмним компонентом було обрано фреймворк MediaPipe від Google.

Ця бібліотека базується на високоефективних нейромережевих моделях, які дозволяють у реальному часі здійснювати прецизійне відстеження 21 ключової точки кисті руки. Вибір MediaPipe зумовлений її високою оптимізацією під архітектуру ARM, на якій побудовано процесор Raspberry Pi, що забезпечує стабільну частоту кадрів без надмірного навантаження на апаратні ресурси. На відміну від громіздких нейромереж загального призначення, цей інструмент дозволяє системі миттєво ідентифікувати просторову конфігурацію пальців, що є критично важливим для реалізації дворівневої логіки вибору пристрою та подальшого виконання команд.

У парі з фреймворком розпізнавання використовується бібліотека OpenCV, яка виконує роль інструменту для попередньої та фінальної обробки відеопотоку. Вона забезпечує захоплення зображення з камери, корекцію його розміру та

фільтрацію цифрових шумів перед передачею даних у MediaPipe. Крім того, засобами OpenCV реалізується графічний оверлей на 7-дюймовому дисплеї, що дозволяє візуалізувати процес керування через накладання інтерактивних елементів безпосередньо на відеоряд. Такий підхід створює інтуїтивно зрозумілий інтерфейс зворотного зв'язку, де користувач може бачити графічні підказки щодо поточного вибраного об'єкта та прогресу регулювання параметрів.

Для організації зв'язку між обчислювальним вузлом на базі Raspberry Pi та виконавчими мікроконтролерами ESP32 обрано протокол передачі даних MQTT. Цей стандарт де-факто у сфері інтернету речей забезпечує надзвичайно малу затримку та мінімальні вимоги до пропускну здатності каналу зв'язку завдяки архітектурі «видавець-підписник». Програмна реалізація клієнтської частини на Raspberry Pi базується на бібліотеці Paho-MQTT для Python, яка дозволяє гнучко структурувати потоки даних у вигляді тематичних топіків для кожної підсистеми розумного будинку. Це дає змогу розділяти команди для освітлення, безпеки чи мультимедіа, уникаючи плутанини в керуючих сигналах.

За для організації обміну даними між обчислювальним центром на базі Raspberry Pi та виконавчими модулями ESP32 у проєкті обрано хмарний MQTT-брокер HiveMQ. Його використання як центрального вузла комунікації надає проєктованому програмно-апаратному засобі низку стратегічних переваг у порівнянні з класичними локальними рішеннями, такими як Eclipse Mosquitto. Хоча Mosquitto є популярним вибором для локальних мереж через свою легковажність, він вимагає складного налаштування мережевої інфраструктури, зокрема статичних IP-адрес та прокидання портів на роутері, для забезпечення доступу до системи ззовні. Натомість HiveMQ Cloud забезпечує миттєву доступність системи з будь-якої точки світу через захищене TLS-з'єднання без додаткових маніпуляцій із мережевим обладнанням. Це критично важливо для реалізації вимоги щодо гібридного керування, оскільки дозволяє зовнішньому мобільному додатку синхронізувати стан пристроїв паралельно із жестовими командами, незалежно від фізичного розташування користувача.

Однією з ключових переваг HiveMQ є наявність потужного веб-інструменту моніторингу – Control Center. На відміну від Mosquitto, де для відстеження проходження повідомлень зазвичай потрібно використовувати сторонні термінальні утиліти, HiveMQ надає наочну графічну панель для спостереження за всіма активними клієнтами та структурою топиків у реальному часі. Це значно спрощує процес розробки та налагодження алгоритму, дозволяючи візуально контролювати, чи коректно Raspberry Pi публікує інкрементальні значення яскравості або гучності, та чи отримують ці дані відповідні мікроконтролери.

Крім того, архітектура HiveMQ відзначається високою масштабованістю та відмовостійкістю. У той час як локальний брокер Mosquitto є критичною точкою відмови (якщо Raspberry Pi вимкнеться або перезавантажиться, вся мережа IoT припинить функціонування), хмарна інфраструктура HiveMQ підтримує постійну роботу черги повідомлень. Це дозволяє реалізувати механізм «останньої волі» (Last Will and Testament), за допомогою якого система автоматично ідентифікує втрату зв'язку з будь-яким виконавчим вузлом та сповіщає про це користувача через інтерфейс.

Для етапу розробки та тестування виконавчого рівня системи у проєкті було використано середовище симуляції Wokwi, яке дозволяє моделювати роботу мікроконтролерних систем без необхідності використання фізичного обладнання. Даний інструмент надає можливість створювати віртуальні електронні схеми на базі популярних платформ, зокрема ESP32, Arduino та інших мікроконтролерів, і виконувати програмний код безпосередньо у браузері.

Використання Wokwi дозволило перевірити коректність логіки підключення до MQTT-брокера, механізму підписки на відповідні топіки та реакції виконавчих модулів на отримані команди ще до реалізації апаратної частини системи. Такий підхід значно скорочує час розробки та знижує ризик виникнення помилок на етапі інтеграції програмного та апаратного забезпечення.

Однією з важливих переваг середовища Wokwi є підтримка мережевих протоколів та інтеграції із зовнішніми сервісами, зокрема MQTT-брокерами. Завдяки цьому у симуляторі можливо реалізувати повноцінну модель взаємодії між

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 39
Зм.	Арк.	№докум.	Підпис	Дата		

програмним клієнтом на базі Raspberry Pi та виконавчим пристроєм ESP32. У рамках проєкту віртуальний мікроконтролер у середовищі Wokwi підключався до брокера HiveMQ через Wi-Fi інтерфейс, після чого виконував підписку на визначені топіки та реагував на повідомлення, що надходили від програмного модуля розпізнавання жестів. Це дозволило наочно перевірити правильність формування MQTT-повідомлень, логіку їх маршрутизації через брокер та відповідну реакцію виконавчого пристрою.

## 2.5 Висновки

Підсумовуючи результати проведеного проєктування, можна відзначити, що визначена архітектура та алгоритм функціонування системи забезпечують гнучке дворівневе керування ключовими підсистемами розумного будинку. Сформовані технічні вимоги дозволили підійти до вибору апаратної бази. Обраний програмний стек, що базується на інтелектуальних моделях MediaPipe та хмарній інфраструктурі HiveMQ, відповідає завданням прецизійного розпізнавання жестів та оперативності передачі команд.

					КВРКІ. 220117.04.22.19 ПЗ	Арк.
						40
Зм.	Арк.	№докум.	Підпис	Дата		

### 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНИЙ ЗАСІБ ЖЕСТОВОГО КЕРУВАННЯ ПРИСТРОЯМИ РОЗУМНОГО БУДИНКУ

#### 3.1 Реалізація програмно-технічний засіб жестового керування пристроями розумного будинку

Реалізація програмно-технічного засобу жестового керування на етапі прототипування та апробації алгоритмів зазнала певних модифікацій у порівнянні з базовою структурою. Основним відхиленням стало використання персонального комп'ютера (ноутбука) як центрального обчислювального вузла замість мікрокомп'ютера Raspberry Pi. Узагальнену схему реалізації та розгортання програмно-технічного засобу жестового керування пристроями розумного будинку наведено на рис. 3.1.

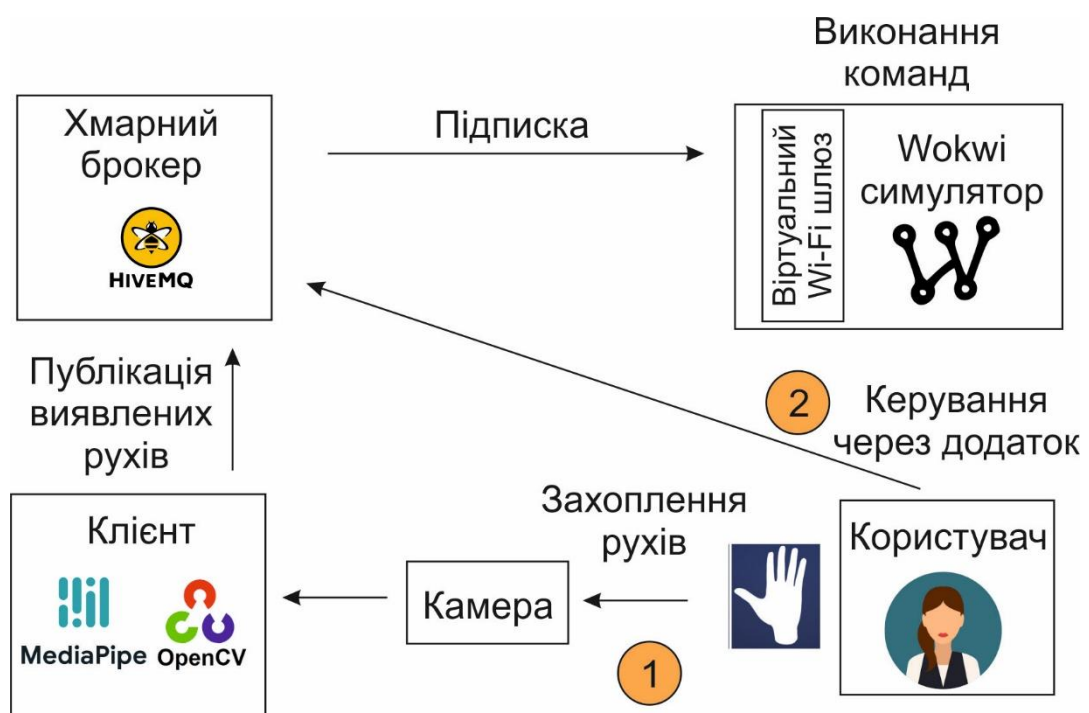


Рисунок 3.1 – Узагальнена схема реалізації та розгортання програмно-технічного засобу жестового керування пристроями розумного будинку

Програмна логіка розпізнавання жестів була повністю реалізована мовою Python у середовищі ноутбука. Вбудована камера виконувала роль сенсорного інтерфейсу, транслюючи зображення до скрипту, який у реальному часі проводив

сегментацію ключових точок руки та класифікацію жестів за розробленою семантикою. Після ідентифікації команди програма за допомогою бібліотеки `raho-mqtt` ініціювала публікацію повідомлень у відповідні топіки на хмарному брокері `HiveMQ`. Використання ноутбука на етапі реалізації спростило процес налагодження графічного інтерфейсу користувача (оверлею), що відображав стан системи та результати розпізнавання безпосередньо на моніторі.

Для відпрацювання взаємодії з виконавчим рівнем без необхідності фізичного монтажу всіх компонентів розумного будинку було використано середовище віртуального моделювання `Wokwi` (рис. 3.2). Це дозволило створити цифровий двійник виконавчого пристрою на базі мікроконтролера `ESP32`. У середовищі `Wokwi` була реалізована програмна модель контролера, яка через віртуальний `Wi-Fi` шлюз підключалася до того ж самого брокера `HiveMQ`, що і `Python`-скрипт на ноутбуці. Основною функцією мікроконтролера полягала в отриманні та інтерпретації команд, що надходять через брокер `HiveMQ` від модуля розпізнавання жестів. Для візуалізації стану підсистем, що мають бінарний режим роботи, а саме вентиляції, сигналізації та базового освітлення, до цифрових портів мікроконтролера через струмообмежувальні резистори підключені світлодіодні індикатори. Їх активація відповідає отриманню логічних сигналів увімкнення або вимкнення у відповідних топіках `MQTT`, що дозволяє миттєво відстежити реакцію системи на жести.

Водночас для підсистем з інкрементальним керуванням, таких як регулювання гучності музики та яскравості світильника, у схемі використані адресні світлодіодні кільця `NeoPixel`. Програмна логіка `ESP32` перетворює числові значення, отримані з хмари, у відповідну кількість активних сегментів на кільцях, що забезпечує наочну демонстрацію рівнів гучності та яскравості від мінімального до максимального. Усі компоненти об'єднані за схемою зі спільною шиною заземлення та лінією живлення, а інтегрований у симуляцію модуль `Wi-Fi` забезпечує стабільне підключення до брокера. Така модель дозволяє повністю перевірити правильність проходження команд від камери ноутбука через `Python`-скрипт до фізичного спрацювання конкретного виконавчого пристрою,

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 42
Зм.	Арк.	№докум.	Підпис	Дата		

підтверджуючи працездатність та надійність розробленої архітектури безконтактного керування.

Як наслідок така архітектура дозволила перевірити повний цикл проходження команди: від фізичного жесту перед камерою ноутбука до спрацювання віртуальних світлодіодів, сервоприводів та індикаторів у симуляторі.

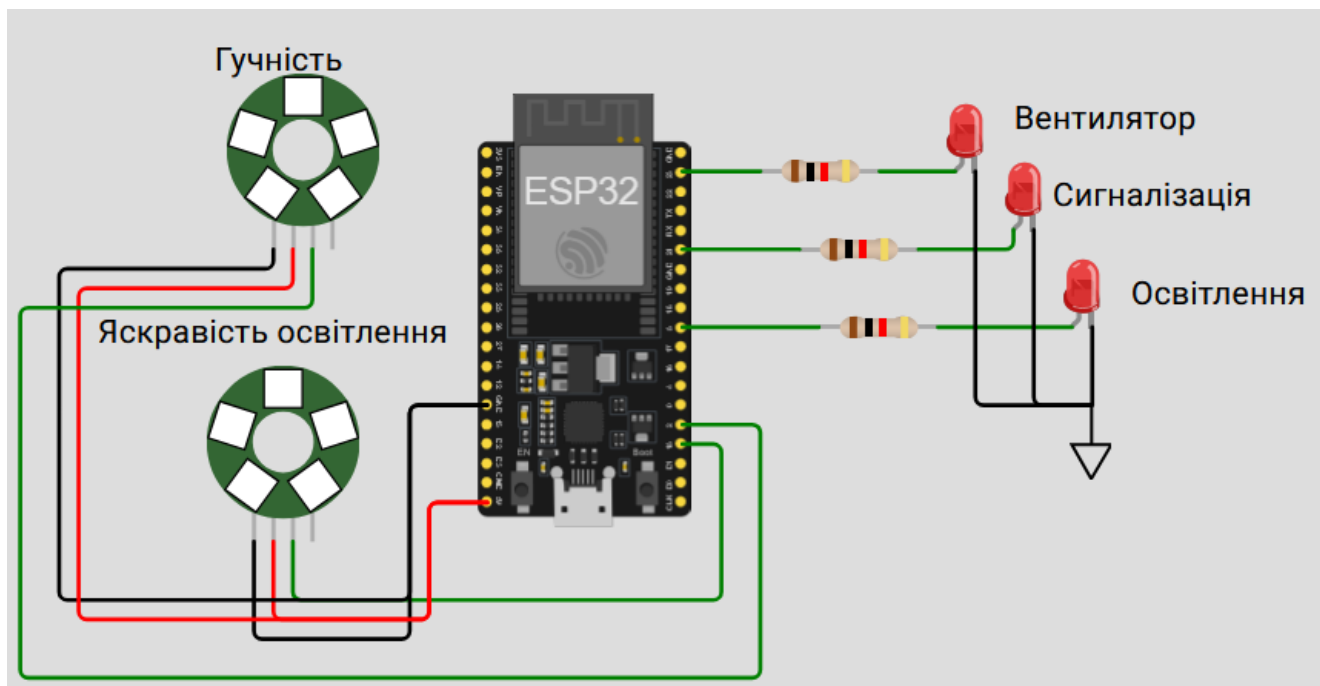


Рисунок 3.2 – Схема цифрового двійника виконавчого пристрою на базі мікроконтролера ESP32 у середовищі Wokwi

### 3.2 Логіка жестової взаємодії та семантика команд керування

Програмна реалізація інтерфейсу безконтактного керування базується на концепції ієрархічної обробки візуальних образів, що дозволяє організувати дворівневий алгоритм взаємодії для послідовного вибору цільового об'єкта та виконання над ним відповідних керуючих впливів. Така архітектура розроблена з метою вирішення фундаментальної проблеми «випадкових жестів» або помилкових спрацювань, що є критично важливим для стабільної роботи систем розумного будинку в реальному часі.

Зм.	Арк.	№докум.	Підпис	Дата

Перший рівень логіки – етап вибору (IDLE state) – використовує кількість піднятих пальців для активації одного з п'яти функціональних вузлів. Кожному пристрою відповідає унікальна цифрова комбінація, що базується на анатомічних особливостях кисті. Другий рівень – етап дії (SELECTED state) – активується після утримання жесту вибору протягом заданого часу (HOLD\_DURATION). На цьому етапі система переходить у режим очікування команд регулювання, що виконуються за допомогою жестів великим пальцем («Вгору»/«Вниз»), або жесту повернення до головного меню.

Нижче у таблиці 3.1 наведено класифікацію жестів та їх функціональне призначення в межах розробленого програмно-технічного засобу.

Таблиця 3.1 – Специфікація жестів керування та відповідних функцій системи

Візуальний патерн жесту	Кількість пальців / Опис	Цільовий вузол / Дія	Тип операції
Вказівний палець	1 (Вказівний)	Світильник	Вибір підсистеми освітлення
Вказівний та середній	2 (Вказівний + Середній)	Вентилятор	Вибір підсистеми вентиляції
Три пальці	3 (Вказівний + Середній + Безіменний)	Сигналізація	Вибір системи безпеки
Чотири пальці	4 (Без великого пальця)	Гучність музики	Вибір мультимедійного вузла
Відкрита долоня	5 (Усі пальці)	Яскравість світла	Вибір вузла регулювання освітлення

### Продовження таблиці 3.1

Великий палець вгору	Thumb Up	Увімкнення / Збільшення (+)	Бінарна або інкрементальна дія
Великий палець вниз	Thumb Down	Вимкнення / Зменшення (-)	Бінарна або інкрементальна дія
«Рок» (Вказівний + Мізинець)	Rock Gesture	Повернення (Back)	Вихід до стану очікування (IDLE)

### 3.3 Створення кластера та налаштування хмарного брокера

Важливим етапом побудови мережевої інфраструктури системи є створення та налаштування хмарного MQTT-брокера, який виступає центральною ланкою обміну даними між обчислювальним вузлом та виконавчими пристроями.

У межах даного проєкту використання хмарного рішення на базі платформи HiveMQ Cloud дозволяє реалізувати надійну архітектуру «видавець-підписник», забезпечуючи високу швидкість передачі команд керування без необхідності розгортання та адміністрування власного фізичного сервера (рис. 3.2).

Основна роль брокера полягає в децентралізації потоків інформації, де мікрокомп'ютер Raspberry Pi публікує розпізнані команди у відповідні тематичні топіки хмарного брокера, а мікроконтролери ESP32, перебуваючи в стані очікування, отримують ці дані для виконання фізичних дій.

Таким чином, окрім кількісного масштабування, такий підхід забезпечує високий рівень інтероперабельності, створюючи умови для безперешкодної інтеграції альтернативних методів взаємодії у єдине інформаційне середовище. Зокрема, використання стандартних інтерфейсів обміну даними дозволяє доповнити систему мобільними застосунками та веб-панелями керування, які функціонують паралельно з модулем розпізнавання жестів. Це дає змогу реалізувати концепцію гібридного керування, де користувач може ініціювати дію за допомогою жесту, а контролювати її виконання або змінювати параметри дистанційно через смартфон. Оскільки хмарний брокер виступає універсальним

вузлом агрегації даних, стан кожного пристрою синхронізується між усіма підключеними клієнтами в реальному часі.

Free #1				▲ Running
Plan	Started	Port (TLS)	URL	
Serverless	2026-02-03T15:39:35Z	8883	94c5d4070f094eefb7d960e1d9f6d3b9.s1.eu.hivemq.cloud	

Manage Cluster

Рисунок 3.2 – Створений кластер

Процес створення хмарного кластера починався із реєстрації в сервісі HiveMQ Cloud, де ініціалізується новий екземпляр брокера в оптимальному географічному регіоні для мінімізації мережевих затримок.

Після створення кластера система згенерувала унікальну адресу хоста (endpoint), яка використовується клієнтськими пристроями для підключення. Налаштування безпеки є критичним аспектом функціонування, тому на рівні брокера створюються облікові записи користувачів із визначенням логінів та паролів, що обмежує несанкціонований доступ до керування пристроями розумного будинку.

Connection Details	
Comprehensive details and statistics for your cluster	
URL	<input type="text" value="d8f617f52e2d4f52b09f153bcd11ac06.s1.eu.hivemq.cloud"/>
Port	<input type="text" value="8883"/>
WebSocket Port	<input type="text" value="8884"/>
TLS MQTT URL	<input type="text" value="d8f617f52e2d4f52b09f153bcd11ac06.s1.eu.hivemq.cloud:8883"/>
TLS WebSocket URL	<input type="text" value="d8f617f52e2d4f52b09f153bcd11ac06.s1.eu.hivemq.cloud:8884/mqtt"/>

Рисунок 3.3 – Параметри підключення до створеного брокера

Окрім автентифікації, хмарний брокер автоматично забезпечує шифрування трафіку за допомогою протоколу TLS через стандартний порт 8883, що відповідає вимогам безпеки сучасних IoT-рішень та запобігає перехопленню команд у відкритих мережах.

Додатковою перевагою обраної платформи HiveMQ є наявність інтегрованого вебклієнта, який надає можливість моніторингу стану мережі безпосередньо через браузер без необхідності встановлення стороннього програмного забезпечення (рис. 3.4). Цей інструмент виступає ефективним засобом верифікації та налагодження системи, дозволяючи в реальному часі відстежувати всі повідомлення, що проходять через брокер. Через інтерфейс вебклієнта можна підписуватися на будь-які топіки розумного будинку, що дає змогу не лише контролювати правильність публікації команд від Raspberry Pi, а й вручну відправляти тестові повідомлення для перевірки реакції виконавчих пристроїв ESP32.

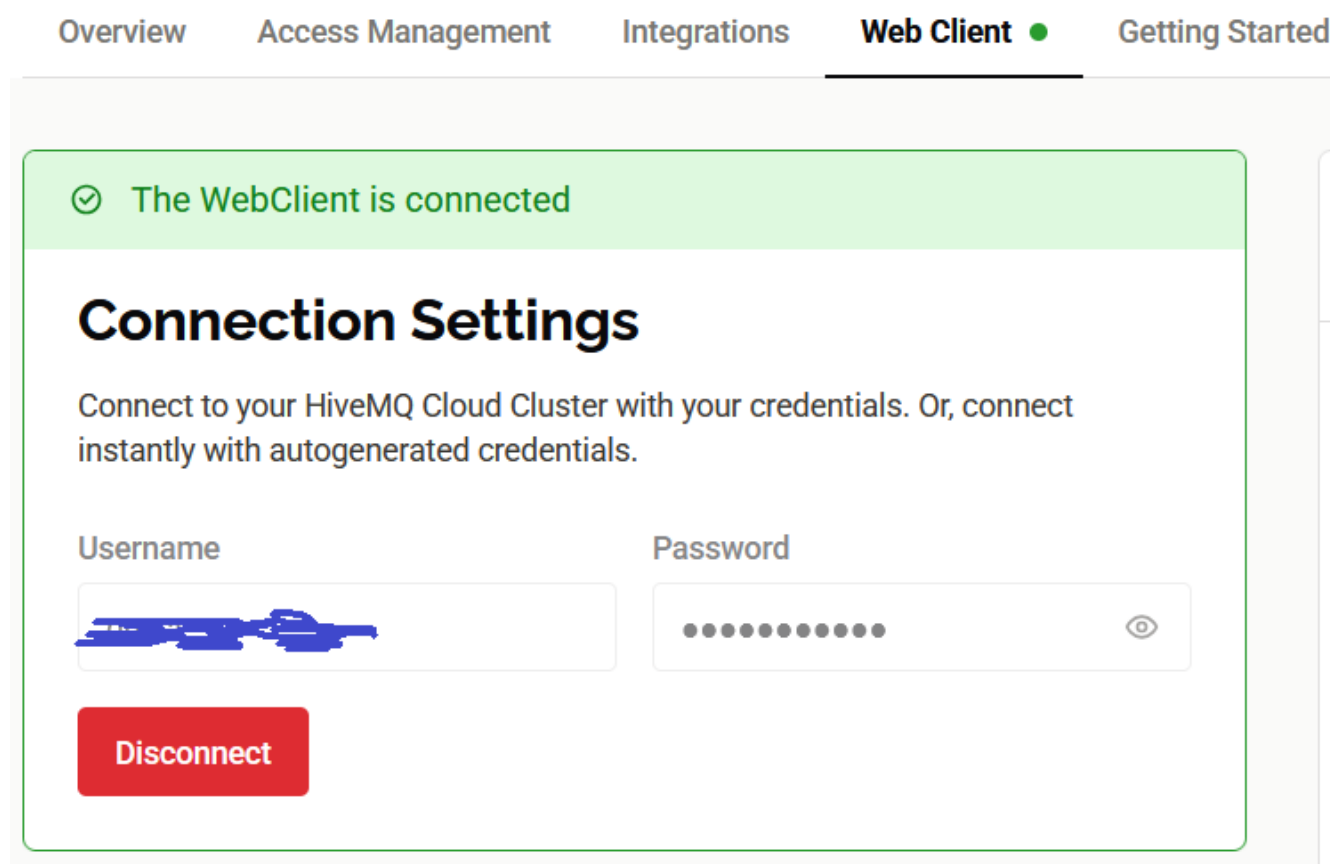


Рисунок 3.4 – Веб клієнт створеного кластеру хмарного брокера HIVE MQ

Зм.	Арк.	№докум.	Підпис	Дата

Використання налаштованого брокера в програмній частині реалізується через підключення бібліотек для роботи з протоколом MQTT, де кожен пристрій ідентифікується за унікальним клієнтським ID.

Програмний засіб на Raspberry Pi налаштовується на постійне з'єднання з кластером, при якому кожна успішно розпізнана дія трансформується у повідомлення для топіків, таких як освітлення, вентиляція чи регулювання параметрів комфорту. Одночасно з цим мікроконтролери ESP32 проходять процедуру підписки на ці ж топіки, що дозволяє їм функціонувати в режимі реального часу. Такий механізм функціонування через хмарний кластер забезпечує стабільність зв'язку незалежно від локальних змін у мережевій структурі користувача та створює фундамент для побудови відмовоїстійкого програмно-технічного комплексу.

### 3.4 Обробка відеоданих та реалізація ідентифікації ключових точок кисті на базі MediaPipe

Ключовим елементом програмно-технічного засобу жестового керування є підсистема комп'ютерного зору, яка забезпечує безперервний аналіз відеопотоку з камери та виявлення геометричних характеристик кисті руки користувача в реальному часі. Реалізація цього компонента базується на використанні двох бібліотек: OpenCV для захоплення та попередньої обробки відеоданих і MediaPipe HandLandmarker для детектування та локалізації анатомічних точок кисті.

Відеозахоплення здійснюється засобами модуля cv2.VideoCapture з примусовим встановленням роздільної здатності 1280×720 пікселів. Підвищена роздільна здатність забезпечує збереження дрібних морфологічних деталей кисті – міжфалангових проміжків та взаємного розміщення кінчиків пальців – що безпосередньо позначається на точності класифікації жестів. Зчитаний кадр піддається горизонтальному дзеркальному відображенню (cv2.flip з прапорцем 1), яке усуває дзеркальну інверсію, характерну для фронтальних веб-камер, та

забезпечує природну відповідність між рухами руки користувача і відображуваною на екрані моделлю.

Перед передачею кадру до інференційного модуля виконується перетворення колірного простору з BGR (стандарт OpenCV) до RGB (стандарт MediaPipe) за допомогою функції `cv2.cvtColor`. Конвертований масив обгортається в об'єкт `mp.Image` із зазначенням формату SRGB, що задовольняє вимоги інтерфейсу Tasks API фреймворку MediaPipe версії 0.10+. Синхронний режим інференції (на відміну від асинхронного потокового) обраний свідомо: він гарантує відповідність результатів детектування кадру, обробленому в поточній ітерації основного циклу, що спрощує синхронізацію з логікою розпізнавання жестів.

Модель HandLandmarker, навчена на основі архітектури BlazePalm та MediaPipe Hands, повертає тривимірні нормалізовані координати 21 анатомічної точки кисті, що відповідають HAND\_CONNECTIONS топології: зап'ясток (0), пп'ястково-фалангові (1,5,9,13,17), проксимальні міжфалангові (2,6,10,14,18), дистальні міжфалангові (3,7,11,15,19) та кінцеві (4,8,12,16,20) суглоби. Нормалізовані значення координат  $x$  та  $y$  знаходяться в діапазоні  $[0,1]$  відносно розмірів кадру, тому для отримання піксельних координат виконується масштабування:  $px = \text{int}(lm.x \cdot W)$ ,  $py = \text{int}(lm.y \cdot H)$ , де  $W$  та  $H$  – ширина і висота кадру в пікселях.

Отримані координати зберігаються у структурі `lm_list` у вигляді впорядкованого списку триплетів `[id, px, py]`, де `id` відповідає індексу точки у топології HandLandmarker. Ця структура виступає єдиним інтерфейсом між модулем детектування та модулем класифікації жестів: будь-яка подальша логіка визначення стану пальців та ідентифікації жестів оперує виключно значеннями пікселів з `lm_list`, не маючи прямого доступу до моделі детектування.

Для візуальної верифікації коректності детектування реалізовано процедуру `draw_hand`, яка відтворює кістяк кисті поверх відеокадру: кожне кісткове з'єднання з набору HAND\_CONNECTIONS відображається у вигляді лінії, а самі ключові точки – у вигляді заповнених кіл. Рендеринг виконується безпосередньо у буфер вхідного кадру засобами OpenCV до його відображення у вікні перегляду.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 49
Зм.	Арк.	№докум.	Підпис	Дата		

Загальна архітектура pipeline обробки відеоданих наведена на блок-схемі 2.1. Логіка класифікації жестів, яка використовує `lm_list` як вхідні дані, відображена на блок-схемі 3.5.

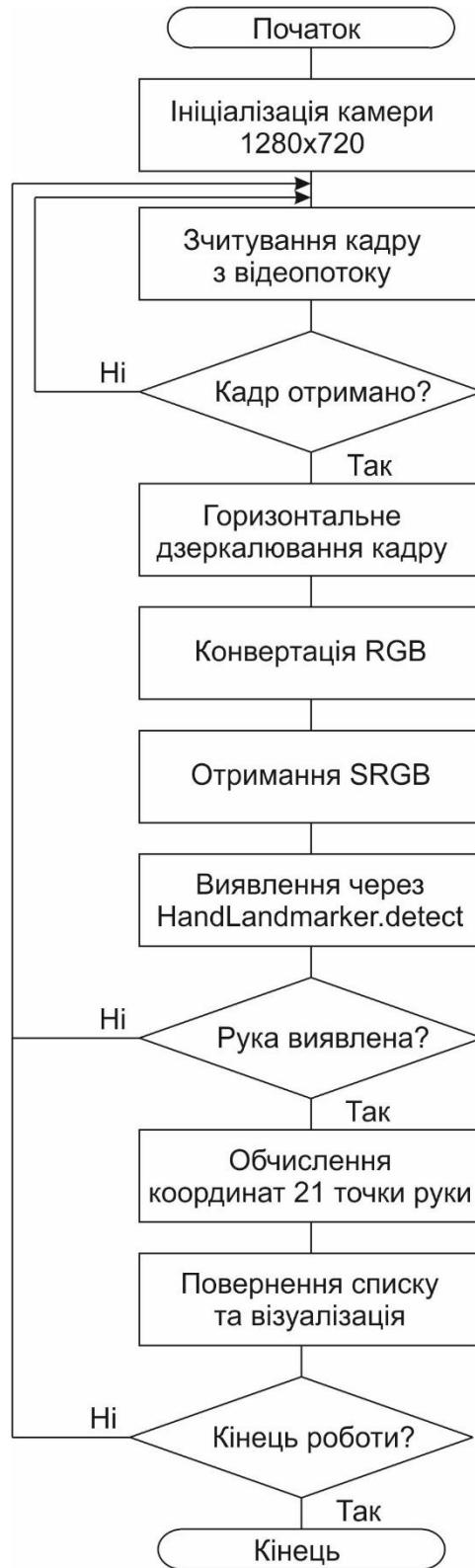


Рисунок 3.5 – Блок-схема алгоритму реалізації конвеєру обробки відеопотоку та виявлення ключових точок кисті

Зм.	Арк.	№докум.	Підпис	Дата

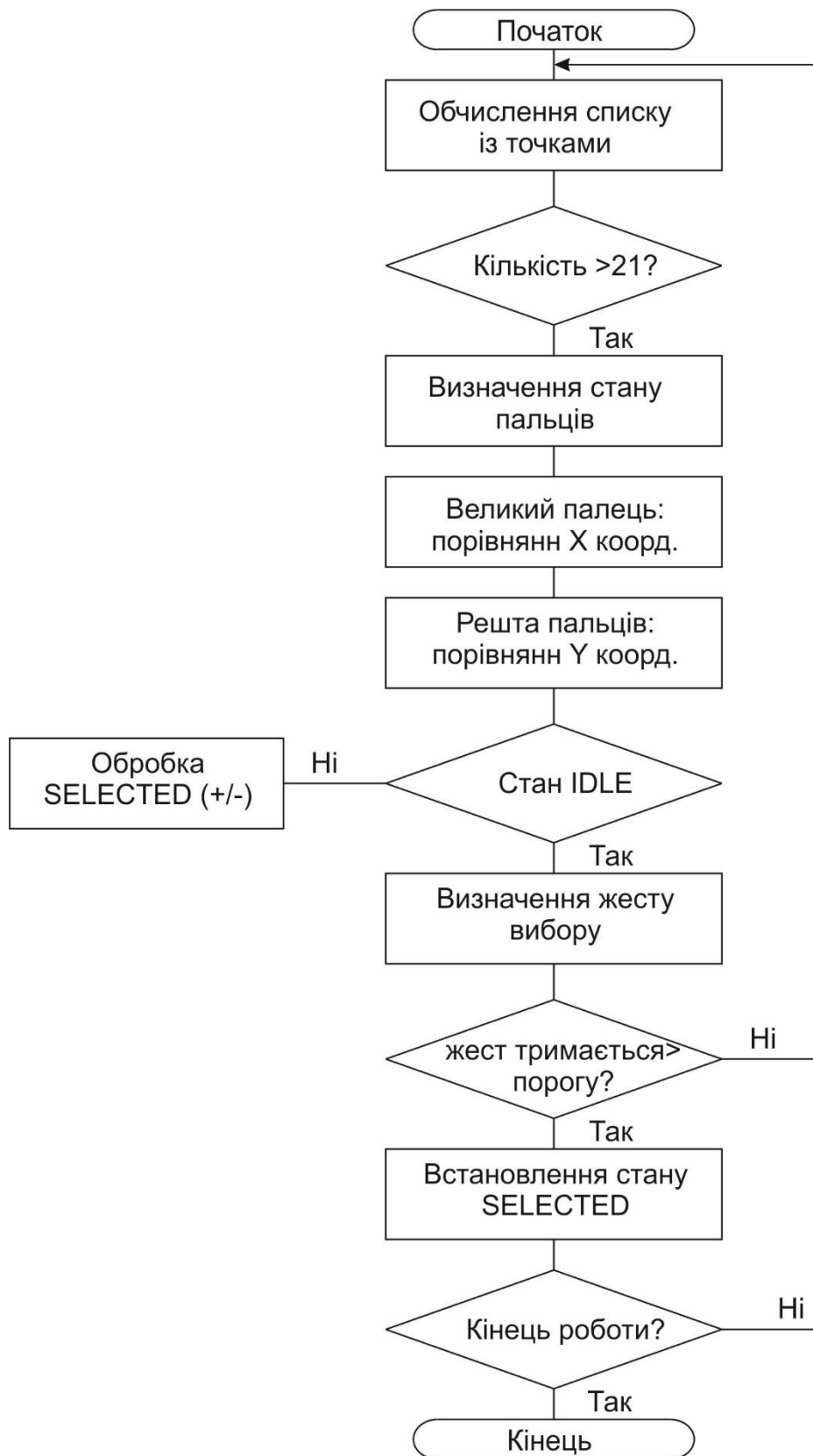


Рисунок 3.6 – Блок-схема алгоритму класифікація жестів та формування керуючих команд

Надійність виявлення кисті забезпечується трьома порогоми впевненості, що налаштовуються в параметрах `HandLandmarkerOptions`: мінімальна впевненість детектування (`min_hand_detection_confidence = 0.7`), мінімальна впевненість присутності кисті (`min_hand_presence_confidence = 0.7`) та мінімальна впевненість трекінгу між кадрами (`min_tracking_confidence = 0.5`). Вибір підвищених порогів детектування та присутності мінімізує кількість хибнопозитивних спрацювань у складних умовах освітлення, тоді як нижчий поріг трекінгу забезпечує стабільне супроводження вже виявленої руки при короткочасних часткових перекриттях.

### 3.5 Тестування програмно-технічного засобу жестового керування пристроями розумного будинку

Тестування програмно-технічного засобу жестового керування пристроями розумного будинку проводилося з метою перевірки коректності роботи алгоритмів розпізнавання жестів, стабільності передачі команд та загальної працездатності системи. У процесі тестування користувач формував визначені жести перед камерою, після чого програмний модуль на основі бібліотек `MediaPipe` та `OpenCV` виконував обробку відеопотоку, визначав положення руки та конфігурацію пальців, на основі чого ідентифікував відповідну команду. Після розпізнавання жесту сформоване повідомлення передавалося через MQTT-протокол до хмарного брокера `HiveMQ`, що забезпечувало подальшу доставку команди виконавчому пристрою.

Для перевірки функціонування системи керування використовувався симулятор `Wokwi`, у якому реалізовано модель пристрою розумного будинку з віртуальним Wi-Fi шлюзом. Після отримання повідомлення від брокера пристрій підписувався на відповідний MQTT-топік та виконував задану команду, що дозволило оцінити правильність обміну повідомленнями та затримку реагування системи.

При підготовці до тестування спроектованого програмно-технічного засобу спочатку було виконано підписку на всі MQTT теми у хмарному брокері HIVE MQ (рис. 3.7).

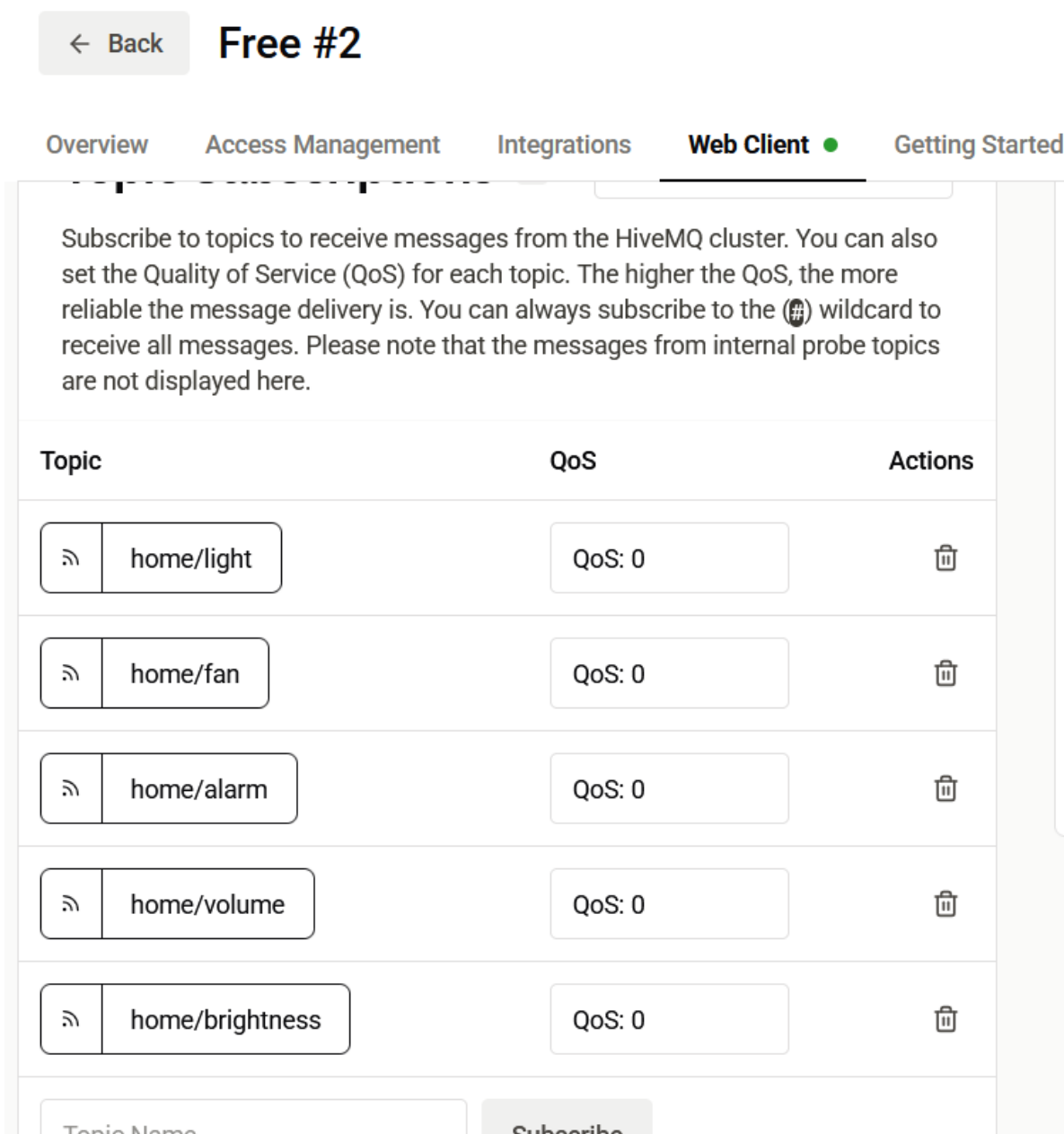


Рисунок 3.7 – Підписка на теми MQTT у брокері HIVE MQ

Далі було реалізовано клієнта на базі Python, який здійснював розпізнавання рухів, а також клієнта у Wowki, що моделював роботу реальної системи Розумного будинку. Головне вікно розробленого клієнта наведено на рис. 3.8. Можна відзначити підказку вгорі вікна, яка показує кількість пальців

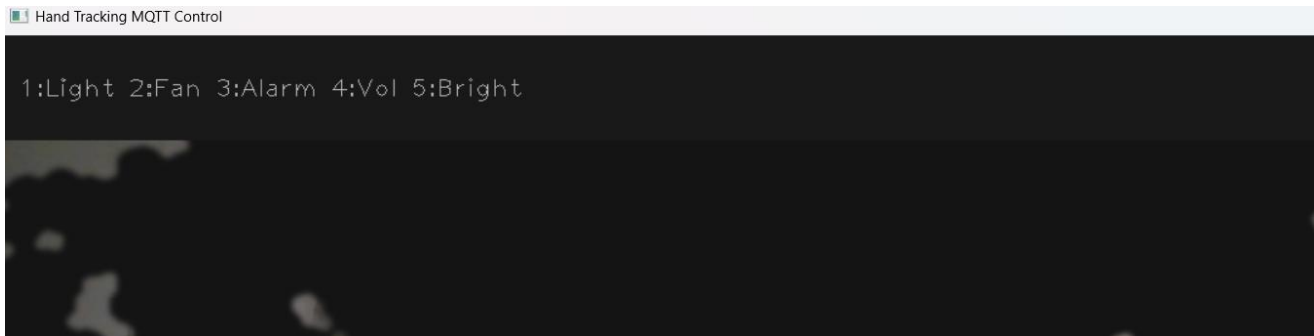


Рисунок 3.8 – Головне вікно розробленого клієнта

В процесі роботи було протестовано всі комбінації жестів кисті руки (що наведені в таблиці 3.1).

Наприклад на рис. 3.9 наведено збільшення рівня гучності, шляхом розпізнавання жесту «Великий палець вгору» для керування звуком, вимкнення вентилятору, шляхом розпізнавання жесту «Великий палець вниз», керування вентилятором, шляхом розпізнавання жесту двох пальців, а також виконання повернення у головне меню, шляхом розпізнавання жесту «Рок» (Вказівний + Мізинець) – рис. 3.10.

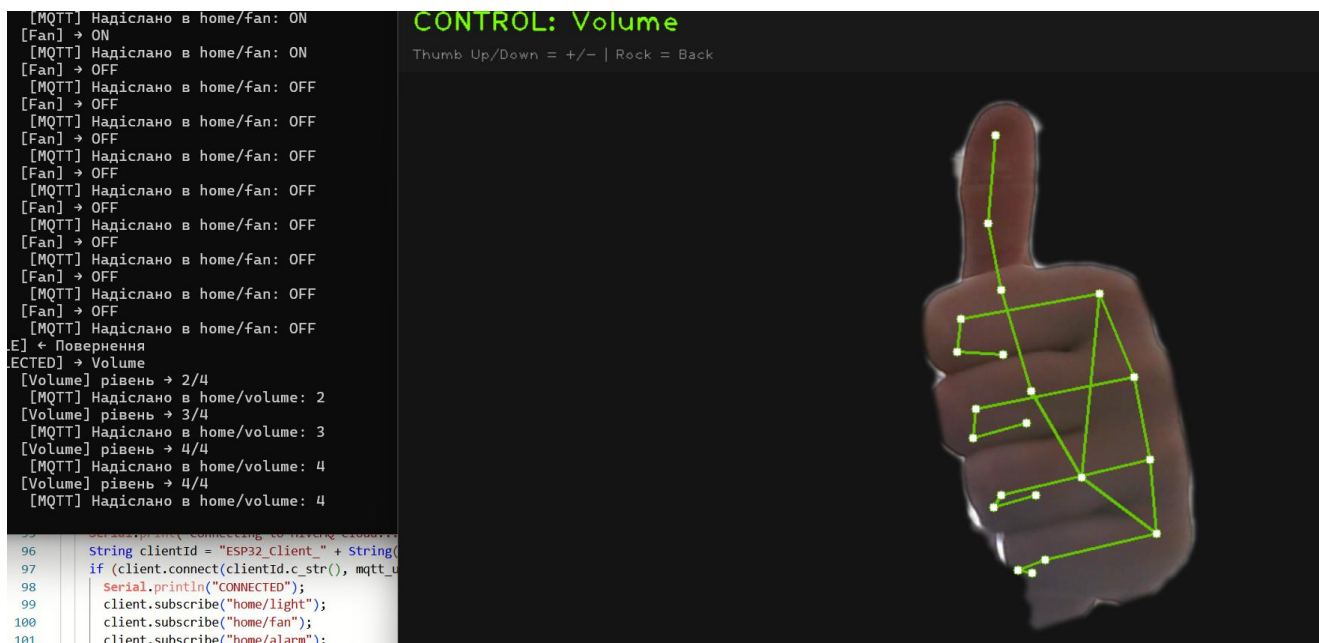


Рисунок 3.10 – Збільшення рівня гучності, шляхом розпізнавання жесту «Великий палець вгору»

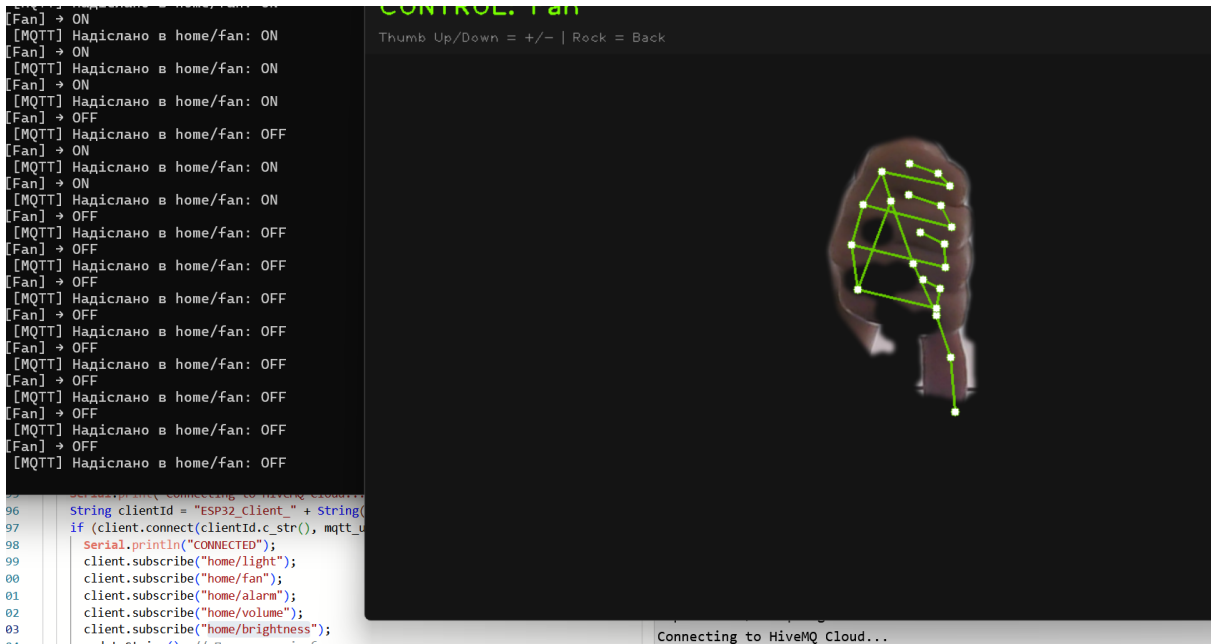


Рисунок 3.11 – Вимкнення вентилятору, шляхом розпізнавання жесту «Великий палець вниз»

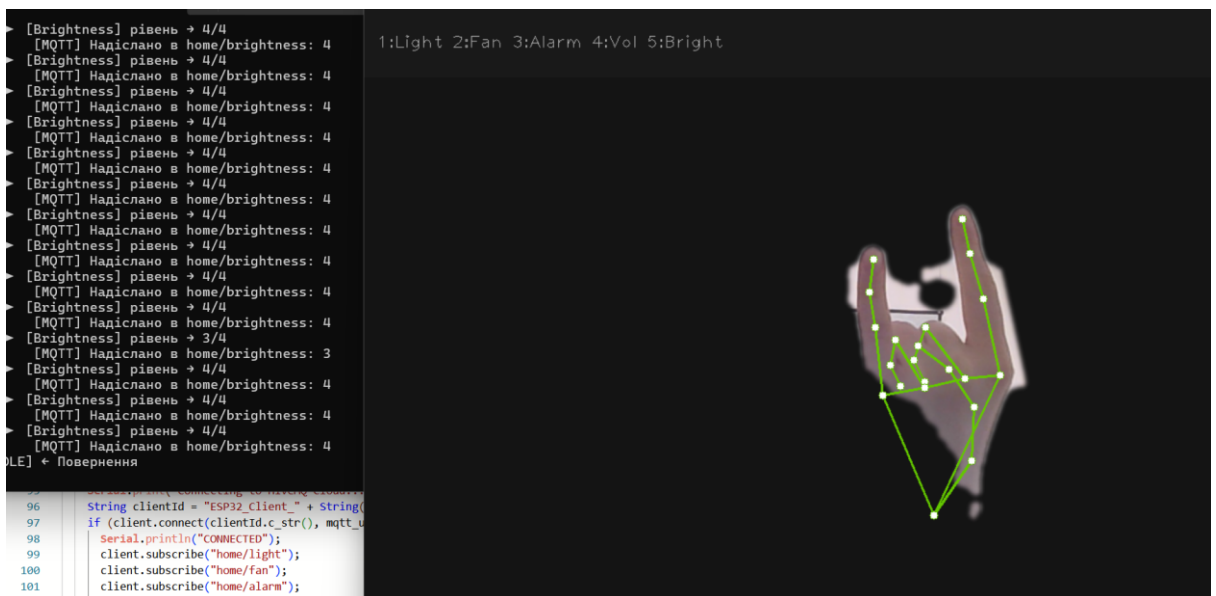


Рисунок 3.12 – Виконання повернення у головне меню, шляхом розпізнавання жесту «Рок» (Вказівний + Мізинець)

Результати розпізнавання жестів відображаються у консолі програмного застосунку, що дозволяє спостерігати процес визначення жестів у реальному часі. Це дає можливість контролювати коректність роботи алгоритму та перевіряти, які саме команди формуються системою на основі розпізнаних жестів.

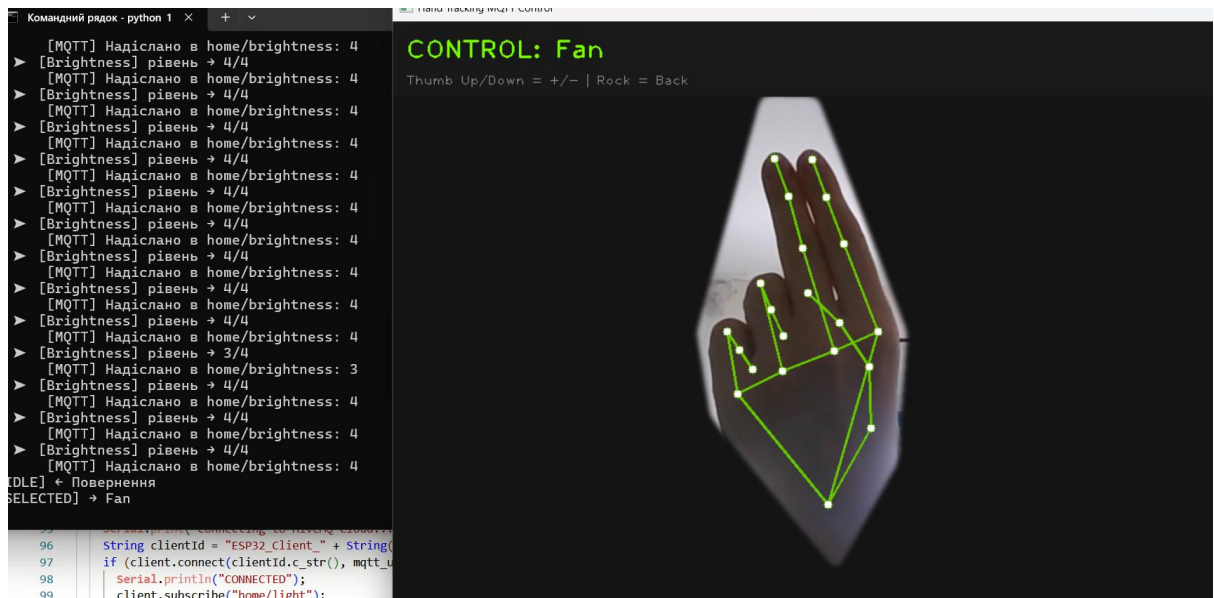


Рисунок 3.13 – Керування вентилятором, шляхом розпізнавання жесту двох пальців

Також під час тестування було перевірено коректність передавання повідомлень між усіма компонентами системи через брокер повідомлень HiveMQ. Зокрема, контролювалося формування та публікація повідомлень клієнтським застосунком у відповідні MQTT-топіки після розпізнавання жестів користувача (рис. 3.14). Для цього здійснювався моніторинг активності брокера, що дозволило переконатися у своєчасному надходженні повідомлень, правильності структури переданих даних та відсутності втрат під час передавання.

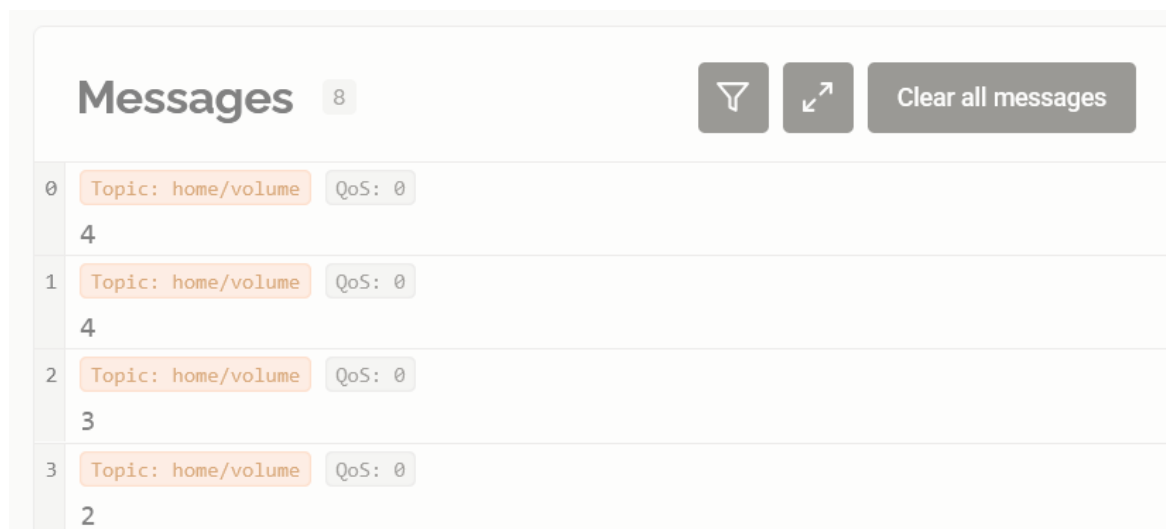


Рисунок 3.14 – Надходження повідомлень у теми HIVE MQ

Окрім цього, було перевірено отримання опублікованих повідомлень клієнтським пристроєм у середовищі симуляції Wokwi (рис. 3.15).

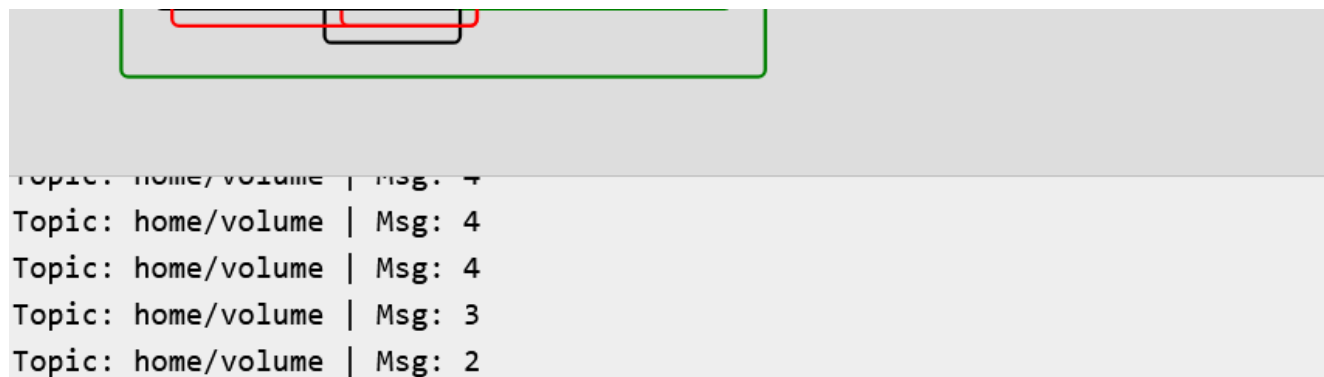


Рисунок 3.15 – Надходження повідомлень у Wokwi

Віртуальний пристрій, підключений до мережі через Wi-Fi та підписаний на відповідні MQTT-топіки, приймав повідомлення від брокера та виконував відповідні команди керування. Це дозволило оцінити правильність механізму підписки, стабільність комунікації між клієнтом і виконавчим модулем, а також загальну узгодженість роботи системи жестового керування пристроями розумного будинку. Проведені перевірки підтвердили, що повідомлення успішно доставляються від програмного клієнта до виконавчого пристрою через брокер HiveMQ, що забезпечує надійне функціонування розробленої системи.

### 3.6 Висновки

У даному розділі було розглянуто повний цикл створення та апробації програмно-технічного засобу жестового керування пристроями розумного будинку – від формування хмарної інфраструктури до перевірки функціонування виконавчого рівня в середовищі віртуального моделювання.

На етапі інфраструктурної підготовки здійснено створення та налаштування хмарного MQTT-брокера на платформі HiveMQ. Використання захищеного з'єднання (TLS) та механізмів автентифікації підтвердило можливість побудови безпечного каналу обміну даними в IoT-архітектурі.

Особливістю реалізації прототипу стало застосування персонального комп'ютера як центрального обчислювального вузла замість мікрокомп'ютера Raspberry Pi. У результаті було реалізовано стійкий механізм сегментації та інтерпретації жестів у режимі реального часу з подальшою передачею команд через протокол MQTT.

Для перевірки взаємодії з виконавчим рівнем без фізичного монтажу апаратної частини використано середовище моделювання Wokwi. У ньому створено цифровий двійник пристрою на базі мікроконтролера ESP32, який підключався до того ж брокера та виконував інтерпретацію отриманих команд. Бінарні підсистеми (освітлення, вентиляція, сигналізація) були реалізовані за допомогою світлодіодних індикаторів, тоді як інкрементальні параметри (рівень гучності та яскравості) відображалися через адресні світлодіоди NeoPixel, що дозволило наочно продемонструвати дискретні рівні керування.

Проведене тестування підтвердило коректність проходження команд по всьому ланцюгу: від фізичного жесту перед камерою ноутбука до спрацювання віртуальних виконавчих елементів у симуляторі. Отримані результати засвідчили працездатність запропонованої архітектури, її масштабованість та придатність до подальшого перенесення з віртуального середовища у фізичну реалізацію.

					КВРКІ. 220117.04.22.19 ПЗ	Арк.
						58
Зм.	Арк.	№докум.	Підпис	Дата		

## ВИСНОВКИ

За результатами виконання кваліфікаційної роботи було спроектовано та розроблено програмно-технічний засіб жестового керування пристроями розумного будинку. Даний засіб забезпечує безконтактне керування підсистемами розумного будинку (освітленням, вентиляцією, безпекою та мультимедіа) за допомогою розпізнавання жестів у реальному часі. Система базується на дворівневому алгоритмі: вибір пристрою здійснюється статичним жестом, а виконання бінарних або інкрементальних команд – динамічним рухом великого пальця.

Обробка відеопотоку реалізована на Python із використанням фреймворку MediaPipe для детекції ключових точок кисті та OpenCV для візуалізації інтерфейсу. Надійний зв'язок між модулем розпізнавання та виконавчим рівнем на базі ESP32 забезпечується через хмарний брокер HiveMQ за протоколом MQTT, що дозволяє миттєво синхронізувати стан усіх вузлів системи.

Під час апробації прототипу в середовищі Wokwi було підтверджено повний цикл проходження команди: від ідентифікації жесту камерою до спрацювання віртуальних індикаторів та адресних світлодіодів NeoPixel. Тестування довело достатньо високу точність розпізнавання та відсутність критичних затримок при передачі даних через хмару.

У першому розділі виконано аналіз концепції розумного будинку та сучасних підходів до керування його виконавчими пристроями. Розглянуто основні методи керування, зокрема безконтактні інтерфейси взаємодії, та проведено огляд методів розпізнавання жестових комбінацій людини на основі технологій комп'ютерного зору. Також здійснено порівняльний аналіз мікроконтролерних платформ, придатних для реалізації виконавчих вузлів системи розумного будинку. За результатами огляду сформульовано висновки та постановку задачі дослідження.

У другому розділі розглянуто структуру та принципи функціонування програмно-технічного засобу жестового керування пристроями розумного будинку. Визначено вимоги до проєктованої системи з урахуванням

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 59
Зм.	Арк.	№докум.	Підпис	Дата		

ергономічності, надійності, масштабованості та реального часу роботи. На основі сформульованих вимог обґрунтовано вибір апаратних та програмних засобів, необхідних для реалізації системи, а також запропоновано загальну архітектуру взаємодії між модулем розпізнавання жестів, хмарною інфраструктурою та виконавчими пристроями.

У третьому розділі виконано реалізацію програмно-технічного засобу жестового керування пристроями розумного будинку. Розглянуто обробку відеоданих та реалізацію ідентифікації ключових точок кисті руки на базі бібліотеки MediaPipe, а також механізми формування керуючих команд. Описано процес створення кластера та налаштування хмарного брокера для передачі команд керування. Проведено тестування програмно-технічного засобу, включно з перевіркою коректності взаємодії між модулем розпізнавання жестів і виконавчими пристроями, що підтвердило працездатність та ефективність запропонованого рішення.

Метою подальшої розробки планується зосередитись на підвищенні точності та стійкості розпізнавання жестів за рахунок використання адаптивних моделей машинного навчання та персоналізації під конкретного користувача, а також впровадження механізмів контекстного аналізу та зворотного зв'язку для запобігання помилковим спрацюванням. Таким чином користувач матиме змогу відмінити або виправити вибраний жест, і як наслідок, змінювати стан керованого ним пристрою Розумного будинку.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Darrell T., Pentland A. Space-time gestures. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 15–17 June 1993. P. 335–340.
2. Ren Z., Meng J., Yuan J. Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction. *Proceedings of the 2011 8th International Conference on Information, Communications & Signal Processing*, Singapore, 13–16 December 2011. P. 1–5.
3. Liu J., Liu H., Chen Y., Wang Y., Wang C. Wireless Sensing for Human Activity: A Survey. *IEEE Commun. Surv. Tutor.* 2020. № 22. P. 1629–1645.
4. Pan M., Tang Y., Li H. State-of-the-Art in Data Gloves: A Review of Hardware, Algorithms, and Applications. *IEEE Trans. Instrum. Meas.* 2023. № 72. P. 4002515.
5. Tang Y., Pan M., Li H., Cao X. A Convolutional-Transformer-Based Approach for Dynamic Gesture Recognition of Data Gloves. *IEEE Trans. Instrum. Meas.* 2024. № 73. P. 2518813.
6. Chakraborty B.K., Sarma D., Bhuyan M., MacDorman K.F. Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Comput. Vis.* 2018. № 12. P. 3–15.
7. Aggarwal J., Ryoo M. Human activity analysis: A review. *ACM Comput. Surv.* 2011. № 43. P. 16.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. arXiv, 2023, arXiv:1706.03762v7.
9. D’Eusanio, A.; Simoni, A.; Pini, S.; Borghi, G.; Vezzani, R.; Cucchiara, R. A Transformer-Based Network for Dynamic Hand Gesture Recognition. In *Proceedings of the 2020 International Conference on 3D Vision (3DV)*, Fukuoka, Japan, 25–28 November 2020, pp. 1–8.
10. Qi J., Ma L., Cui Z., Yu Y. Computer vision-based hand gesture recognition for human–robot interaction: A review. *Complex Intell. Syst.* 2024. № 10. P. 1581–1606.

					КВРКІ. 220117.04.22.19 ПЗ	Арк. 61
Зм.	Арк.	№докум.	Підпис	Дата		

11. Fertl E., Nguyen D.D.T., Krueger M., Stettinger G., Padiá-Allué R., Castillo E., Cuéllar M.P. End-to-End Ultrasonic Hand Gesture Recognition. *Sensors*. 2024. № 24. P. 2740.

12. Strobel M., Schoenfeldt S., Daugalas J. Gesture Recognition for FMCW Radar on the Edge. *Proceedings of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, Las Vegas, NV, USA, 22–25 January 2023. P. 45–48.

13. Besrou, S.; Surapaneni, Y.; Mubibya, G.S.; Ashkar, F.; Almhana, J. A Transformer-Based Approach for Better Hand Gesture Recognition. In *Proceedings of the 2024 20th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Limassol, Cyprus, 27–31 May 2024, pp. 1135–1140.

14. Tang G., Wu T., Li C. Dynamic Gesture Recognition Based on FMCW Millimeter Wave Radar: Review of Methodologies and Results. *Sensors*. 2023. № 23. P. 7478.

15. Zhang J., Tao D. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet Things J.* 2020. № 8. P. 7789–7817.

16. Ahmed S., Kallu K.D., Ahmed S., Cho S.H. Hand Gestures Recognition Using Radar Sensors for Human-Computer-Interaction: A Review. *Remote Sens.* 2021. № 13. P. 527.

17. Dahl T., Ealo J.L., Bang H.J., Holm S., Khuri-Yakub P. Applications of airborne ultrasound in human–computer interaction. *Ultrasonics*. 2014. № 54. P. 1912–1921.

18. Cai C., Zheng R., Luo J. Ubiquitous Acoustic Sensing on Commodity IoT Devices: A Survey. *IEEE Commun. Surv. Tutor.* 2022. № 24. P. 432–454.

19. Jiang S., Kang P., Song X., Lo B., Shull P. Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey. *IEEE Rev. Biomed. Eng.* 2021. № 15. P. 85–102.

20. Reuther A., Michaleas P., Jones M., Gadepally V., Samsi S., Kepner J. AI and ML Accelerator Survey and Trends. *Proceedings of the IEEE High Performance*

*Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 19–23 September 2022. P. 1–10.

21. Mendez J., Bierzynski K., Cuéllar M.P., Morales D.P. Edge Intelligence: Concepts, Architectures, Applications, and Future Directions. *ACM Trans. Embed. Comput. Syst.* 2022. № 21. P. 1–41.

22. Reuther A., Michaleas P., Jones M., Gadepally V., Samsi S., Kepner J. Survey and Benchmarking of Machine Learning Accelerators. *Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 24–26 September 2019. P. 1–9.

23. Zeng X., Wu C., Ye W.B. User-Definable Dynamic Hand Gesture Recognition Based on Doppler Radar and Few-Shot Learning. *IEEE Sens. J.* 2021. № 21. P. 23224–23233.

24. Fan Z., Zheng H., Feng X. A Meta-Learning-Based Approach for Hand Gesture Recognition Using FMCW Radar. *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, 21–23 October 2020. P. 522–527.

25. Mauro G., Chmurski M., Servadei L., Pegalajar-Cuellar M., Morales-Santos D.P. Few-Shot User-Definable Radar-Based Hand Gesture Recognition at the Edge. *IEEE Access.* 2022. № 10. P. 29741–29759.

26. Wang L., Zhang J., Li Y., Wang H. AudioWrite: A Handwriting Recognition System Using Acoustic Signals. *Proceedings of the IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, Nanjing, China, 10–12 January 2023. P. 81–88.

27. Ibrahim A., El-Refai A., Ahmed S., Aboul-Ela M., Eraqi H.M., Moustafa M. Pervasive Hand Gesture Recognition for Smartphones using Non-audible Sound and Deep Learning. *Proceedings of the 13th International Joint Conference on Computational Intelligence (IJCCI 2021)* – NCTA, online, 25–27 October 2021. P. 310–317.

28. Xu, S.; Kaul, C.; Ge, X.; Murray-Smith, R. Continuous Interaction with a Smart Speaker via Low-Dimensional Embeddings of Dynamic Hand Pose. *arXiv*, 2023, arXiv:2302.14566.

29. Jo, B.J.; Kim, S.K.; Kim, S. Enhancing Virtual and Augmented Reality Interactions with a MediaPipe-Based Hand Gesture Recognition User Interface. *Ingénierie des Systèmes d'Information*, 2023, 28, 633–638.

30. Wang Z., Hou Y., Jiang K., Dou W., Zhang C., Huang Z., Guo Y. Hand Gesture Recognition Based on Active Ultrasonic Sensing of Smartphone: A Survey. *IEEE Access*. 2019. № 7. P. 111897–111922.

31. Kumar, R.; Singh, S.K.; Bajpai, A.; Sinha, A. Mediapipe and CNNs for Real-Time ASL Gesture Recognition. *In Proceedings of the 2023 International Conference on Computer Vision and Pattern Recognition*, Greater Noida, India, 28–30 April 2023.

32. Althubiti, A.H.; Algethami, H. Dynamic Gesture Recognition Using a Transformer and Mediapipe. *International Journal of Advanced Computer Science and Applications*, 2024, 15(6).

33. Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017, pp. 7291–7299.

34. Mazzia, V.; Angarano, S.; Salvetti, F.; Angelini, F.; Chiaberge, M. Action Transformer: A Self-Attention Model for Short-Time Pose-Based Human Action Recognition. *arXiv*, 2022, arXiv:2107.00606v6.

35. Gupta, K. Hand Keypoint Detection Using RoboComp. RoboComp Blog, August 2020. URL: <https://robocomp.github.io/web/gsoc/2020/posts/kanav/HandKeypoint> (дата звернення: 19.03.2026).

36. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, 2014, 27, 3104–3112.

37. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *In Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 7–9 May 2015.

38. Tan, C.K.; Lim, K.M.; Chang, R.K.Y.; Lee, C.P.; Alqahtani, A. HGR-ViT: Hand Gesture Recognition with Vision Transformer. *Sensors*, 2023, 23, 5555.

39. Gesture Recognition Techniques. URL: <https://makeitanimated.dev/blog/gesture-recognition-techniques> (дата звернення: 17.03.2026).

40. Building a Real-Time Gesture Recognition System from Scratch (Medium). URL: <https://medium.com/@akhavanparisa90/building-a-real-time-gesture-recognition-system-from-scratch-part-1-fe022fccfffc> (дата звернення: 17.03.2026).

41. Gesture Control & Motion Recognition Technologies. URL: <https://www.hostragons.com/en/blog/gesture-control-motion-recognition-technologies/> (дата звернення: 17.03.2026).

42. Augmented Reality Gesture Recognition – The Invisible Interface of Tomorrow. URL: <https://inairspace.com/blogs/learn-with-inair/augmented-reality-gesture-recognition-the-invisible-interface-of-tomorrow> (дата звернення: 17.03.2026).

43. Vision AI Enables Touch-Free Gesture Recognition Technology. URL: <https://www.ultralytics.com/blog/vision-ai-enables-touch-free-gesture-recognition-technology> (дата звернення: 17.03.2026).

44. Enhance Touch and Gesture AI. URL: <https://www.renesas.com/en/blogs/enhance-touch-and-gesture-ai> (дата звернення: 17.03.2026).

45. Модуль розпізнавання напрямку руху і жестів GY-9960 (APDS-9960, RGB). URL: <https://beegreen.com.ua/modul-rozpiznavannya-napryamki-ruhu-i-zhestiv-gy-9960-33-apds-9960-rgb-17878> (дата звернення: 17.03.2026).

46. ESP8266 – Arduino.ua. URL: [https://arduino.ua/cat122-esp8266?categoryID=122&show\\_all=yes](https://arduino.ua/cat122-esp8266?categoryID=122&show_all=yes) (дата звернення: 17.03.2026).

47. Wrist & Hand Active Range of Motion. URL: <https://www.physiotutors.com/uk/wiki/wrist-hand-active-range-of-motion/> (дата звернення: 17.03.2026).

48. Active Range of Motion – Blog Agrocontrol. URL: <https://blog.agrocontrol.net/?p=4480> (дата звернення: 17.03.2026).

49. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *arXiv*, 2020, arXiv:2010.11929.

50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv*, 2023, arXiv:1706.03762v7.

51. Ahmed S., Kallu K.D., Ahmed S., Cho S.H. Hand Gestures Recognition Using Radar Sensors for Human-Computer-Interaction: A Review. *Remote Sens.* 2021. № 13. P. 527.

52. Dahl T., Ealo J.L., Bang H.J., Holm S., Khuri-Yakub P. Applications of airborne ultrasound in human–computer interaction. *Ultrasonics.* 2014. № 54. P. 1912–1921.

					КВРКІ. 220117.04.22.19 ПЗ	Арк.
						66
Зм.	Арк.	№докум.	Підпис	Дата		







Зав. кафедри КІС  
д-р. філософії Ользі ПАВЛОВІЙ

Ксенія ПОЛІЩУК

---

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-4

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року



## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Ксенія ПОЛІЩУК

Тема: Програмно-технічний засіб жестового керування пристроями розумного будинку

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   60  

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є підвищення ергономічності та доступності систем автоматизації житла шляхом проектування програмно-технічного засобу жестового керування пристроями розумного будинку
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено дослідження предметної області програмно-технічних засобів керування пристроями розумного будинку, а саме: розглянуто поняття розумного будинку та сучасні методи керування пристроями; проаналізовано методи розпізнавання жестових комбінацій людини; досліджено скелетонні методи розпізнавання жестів; виконано огляд мікроконтролерних платформ для реалізації виконавчих вузлів системи розумного будинку; сформовано висновки та постановку задачі дослідження. У другому розділі кваліфікаційної роботи виконано проектування програмно-технічного засобу жестового керування пристроями розумного будинку, а саме: визначено вимоги до програмно-технічного засобу; розроблено структуру та описано принцип функціонування системи; проведено аналіз та вибір апаратних компонентів для реалізації системи; виконано огляд і вибір програмних компонентів проєктованого засобу; сформовано висновки щодо результатів проєктування. У третьому розділі кваліфікаційної роботи виконано реалізацію та тестування програмно-технічного засобу жестового керування пристроями розумного будинку, а

саме: реалізовано програмно-технічний засіб жестового керування; розроблено логіку жестової взаємодії та семантику команд керування; створено кластер та налаштовано хмарний брокер для передачі даних; реалізовано обробку відеоданих та ідентифікацію ключових точок кисті на основі технології MediaPipe; проведено тестування працездатності та функціональних можливостей розробленої системи.4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: До недоліків роботи можна віднести обмежену кількість протестованих жестових команд та залежність точності розпізнавання жестів від умов освітлення і положення камери.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

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

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре С (75)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Дасішвілі Олександр Анатолійович  
доцент кафедри КН, ХНУ

“ ” \_\_\_\_\_ 2026 р.

 (підпис)

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Ксенія ПОЛІЩУК

**Співавтор:**

**Назва:** Програмно-технічний засіб жестового керування пристроями розумного будинку

**Експерт:** Володимир ГРИГА

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 6.79%

**Коефіцієнт подібності 2:** 2.44%

**Мікропробіли:** 3

**Заміна букв:** 3

**Інтервали:** 0

**Білі знаки:** 18

**Дата створення звіту:** 2026-05-26 00:56:40.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

**Обґрунтування:**

2026-05-26

*Дата*



Доцент Андрій Нічепорук

експерт

Tue May 26 08:59:36 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (<http://ap.km.ua>) v-15.701

**Максимальне співпадіння з одним документом 1.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилоч в документах: 14%**

ID: 272270 Назва: БКР Програмно-технічний засіб жестового керування пристроями розумного будинку Додано в БД: 2026-05-26 Автора: Ксенія ПОЛЩУК Керівники: Володимир ГРИГА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	85896	527	772 (1%)	12 (2%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

### КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб жестового керування пристроями розумного будинку

Автор Ксенія ПОЛІЩУК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент, Володимир ГРИГА

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

#### Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.


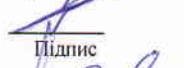

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6,79% і адресується до 35 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.05.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Підпис  
  
Підпис  
  
Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК  
Ім'я, ПРІЗВИЩЕ

Володимир ГРИГА  
Ім'я, ПРІЗВИЩЕ