МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

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

_____Ірина ЖУРАВСЬКА

« __ » _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

Пристрій керування на основі датчика APDS-9960

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач _____ Артем МЕЛЬНИКОВ
підпис

«__» _____ 20__ р.

Керівник старший викладач,
каф. комп'ютерної інженерії _____ Борис САЛТОВСЬКИЙ
підпис

«__» _____ 20__ р.

Миколаїв – 2026

Факультет	Комп’ютерних наук
Кафедра	Комп’ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп’ютерна інженерія
Освітня програма	Комп’ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп’ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«_____» _____ 202__ р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача
Мельникова Артема Геннадійовича

(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Пристрій керування на основі датчика APDS-9960

Затверджена наказом по ЧНУ ім. Петра Могили від 25.11.2025 № 294.

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

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

Очікуваним результатом роботи є прототипу пристрою безконтактного керування на основі датчика APDS-9960, який розпізнає прості жести користувача та перетворює їх на керуючі команди. Початковими даними є датчик APDS-9960, мікроконтролер, середовище Arduino IDE, бібліотеки для роботи з I2C та тестовий сценарій керування.

4. Перелік питань, що підлягають розробці:

1) проаналізувати особливості безконтактного керування електронними пристроями та основні засоби розпізнавання жестів;

2) дослідити технічні можливості, принцип роботи та обмеження датчика APDS-9960;

3) виконати порівняльний аналіз APDS-9960, PAJ7620U2 та камерних рішень для жестового керування;

4) обрати мікроконтролер, спосіб підключення датчика APDS-9960 та програмні засоби реалізації пристрою;

5) розробити структурну й електричну схеми пристрою, алгоритм зчитування та обробки жестів;

6) сформувати карту команд керування, реалізувати програмну частину та провести експериментальну перевірку роботи прототипу.

5. Перелік графічних матеріалів

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи _____ Борис САЛТОВСЬКИЙ
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ Артем МЕЛЬНИКОВ
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Дата видачі завдання «_____» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної бакалаврської роботи

Тема: Пристрій керування на основі датчика APDS-9960

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	28.11.2025	03.12.2025	Виконано
2	Складання календарного плану КБР	11.12.2025	14.12.2025	Виконано
3	Огляд літератури та рішень у сфері безконтактного жестового керування	15.12.2025	25.12.2025	Виконано
4	Аналіз технічних можливостей APDS-9960, PAJ7620U2 та камерних рішень	15.01.2026	02.02.2026	Виконано
5	Проєктування архітектури та вибір апаратно-програмних компонентів	03.02.2026	27.02.2026	Виконано
6	Розробка алгоритму зчитування, фільтрації та інтерпретації жестів	28.02.2026	13.03.2026	Виконано
7	Реалізація програмної частини пристрою та формування карти команд	14.03.2026	08.05.2026	Виконано
8	Оформлення КБР та презентації	27.04.2026	09.05.2026	Виконано
9	Перший передній захист КБР	22.05.2026	22.05.2026	Виконано
10	Другий передній захист КБР	05.06.2026	05.06.2026	Виконано
11	Завершення оформлення КБР та презентації	06.06.2026	09.06.2026	Виконано
12	Перевірка на академічний плагіат, фальсифікацію та списування	09.06.2026	10.06.2026	Виконано
13	Відгук керівника КБР	09.06.2026	10.06.2026	Виконано
14	Рецензування КБР	10.06.2026	13.06.2026	Виконано
15	Подання КБР, її електронної копії та інших документів (відгуку, рецензії)	15.06.2026	17.06.2026	Виконано
16	Захист кваліфікаційної бакалаврської роботи	24.06.2026	24.06.2026	Виконано

Керівник роботи _____ Борис САЛТОВСЬКИЙ

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ Артем МЕЛЬНИКОВ

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

«____» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Пристрій керування на основі датчика APDS-9960»
Здобувач: Мельников Артем Геннадійович
Керівник: старший викладач Салтовський Борис Григорович

Кваліфікаційна бакалаврська робота присвячена розробці пристрою безконтактного керування на основі оптичного датчика APDS-9960. У роботі розглянуто практичний підхід до побудови компактного апаратно-програмного прототипу, який зчитує прості жести користувача, виконує їх програмну обробку та перетворює на керуючі HID-команди для персонального комп'ютера. Такий тип керування є доцільним у тих випадках, коли користувачу потрібно виконувати короткі повторювані дії без прямого контакту з клавіатурою, мишею або сенсорною панеллю.

Актуальність теми пов'язана з поширенням безконтактних інтерфейсів у навчальних, презентаційних, мультимедійних і побутових системах. На відміну від камерних засобів розпізнавання рухів, APDS-9960 не потребує обробки відеопотоку та значних обчислювальних ресурсів. Сенсор працює в ближній зоні, аналізує відбите інфрачервоне випромінювання та дає змогу розпізнавати базові напрямлені жести: вліво, вправо, вгору та вниз. У роботі враховано обмеження такого підходу, зокрема невелику робочу відстань, залежність від положення руки та потребу у фільтрації повторних спрацьовувань.

У першому розділі проаналізовано предметну область безконтактного керування, класифіковано засоби розпізнавання жестів, розглянуто технічні можливості APDS-9960 і виконано порівняння з іншими сенсорними рішеннями ближньої дії. У другому розділі розроблено структуру пристрою, обґрунтовано вибір мікроконтролера ESP32-WROOM-32U, описано схему підключення, алгоритм зчитування та карту команд. У третьому розділі наведено практичну реалізацію макетного прототипу, налаштування Arduino IDE, використання бібліотек Wire, SparkFun APDS9960 і BleKeyboard, а також результати технічної перевірки HID-передавання команд.

Практичне значення роботи полягає у створенні доступного прототипу пристрою, який може використовуватися як основа для безконтактного керування презентаціями, мультимедійними функціями або простими режимами роботи комп'ютерних систем. Деталізовану блок-схему алгоритму роботи наведено в Додатку А, а повний код мікропрограмного забезпечення винесено в Додаток Б.

У цілому, кваліфікаційна бакалаврська робота, містить 56 с. (без додатків), 9 рис., 16 табл., 22 джерел посилання та 3 додатки.

Ключові слова: *APDS-9960, ESP32-WROOM-32U, безконтактне керування, жестовий інтерфейс, I2C, HID, Bluetooth HID, Arduino IDE, мікроконтролер, сенсор ближньої дії.*

ABSTRACT
of the Bachelor's Thesis
"Control device based on the APDS-9960 sensor"
Applicant: Artem Melnikov
Supervisor: Senior Lecturer Borys Saltovskyi

The bachelor's thesis is devoted to the development of a contactless control device based on the APDS-9960 optical gesture sensor. The proposed prototype reads simple hand movements, processes them on an ESP32-WROOM-32U microcontroller and converts the recognized gestures into HID control commands for a personal computer. This approach is useful for short repetitive actions, such as switching slides, controlling multimedia playback or confirming simple operations without using a keyboard, mouse or touch panel.

The relevance of the topic is related to the practical use of compact contactless interfaces in educational, presentation, multimedia and household systems. Unlike camera-based gesture recognition, the APDS-9960 does not require video processing or high computing resources. The sensor operates at a short distance, analyzes reflected infrared radiation and supports basic directional gestures: left, right, up and down. The thesis also considers the limitations of this solution, including the short working range, sensitivity to hand position and the need to filter repeated triggers.

The first chapter analyzes the subject area of contactless gesture control, classifies gesture recognition methods, describes the technical capabilities of the APDS-9960 sensor and compares it with other short-range sensor solutions. The second chapter presents the design of the device, the choice of the ESP32-WROOM-32U microcontroller, the connection scheme, the gesture processing algorithm and the command map. The third chapter describes the implementation of the prototype, Arduino IDE configuration, the use of Wire, SparkFun APDS9960 and BleKeyboard libraries, and the results of technical testing of HID command transmission.

The practical value of the work lies in creating an accessible prototype that can be used as a basis for contactless control of presentations, multimedia functions or simple computer system modes. The detailed block diagram of the algorithm is provided in Appendix A, and the full firmware code is given in Appendix B.

The paper consists of 56 pages (excluding appendices), 9 figures, 16 tables, 22 references, and 3 appendices.

Keywords: *APDS-9960, ESP32-WROOM-32U, contactless control, gesture interface, I2C, HID, Bluetooth HID, Arduino IDE, microcontroller, short-range sensor*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

КБР – кваліфікаційна бакалаврська робота

ALS – Ambient Light Sensor
FIFO – First In, First Out data buffer
HID – Human Interface Device
I2C – Inter-Integrated Circuit
IDE – Integrated Development Environment
INT – Interrupt pin
IR – Infrared radiation
LED – Light-Emitting Diode
RGBC – Red, Green, Blue and Clear channels
ToF – Time of Flight
USB – Universal Serial Bus

ВСТУП

Сучасні електронні пристрої дедалі частіше орієнтуються не тільки на функціональність, а й на зручність взаємодії з користувачем. Якщо раніше основну роль відігравали кнопки, перемикачі, клавіатура або сенсорний екран, то зараз усе більше уваги приділяється інтерфейсам, які дозволяють керувати пристроєм простішим і природнішим способом. До таких рішень належать безконтактні інтерфейси, де команда передається не через натискання, а через жест, наближення руки або інший рух у зоні роботи сенсора.

Жестове керування є зручним для тих випадків, коли користувачу потрібно виконати коротку команду без складного введення даних. Це може бути перемикання слайдів, керування мультимедіа, зміна режиму роботи пристрою, підтвердження дії або запуск окремої функції. У таких сценаріях використання клавіатури, миші або сенсорної панелі не завжди є найкращим варіантом, особливо якщо користувач знаходиться на відстані від пристрою, має зайняті руки або працює в умовах, де небажаний частий фізичний контакт з обладнанням.

Одним із практичних напрямів реалізації безконтактного керування є використання оптичних сенсорів ближньої дії. На відміну від камерних систем, вони не потребують обробки відеозображення та значних обчислювальних ресурсів. При цьому такі сенсори можуть визначати базові рухи руки та передавати дані мікроконтролеру для подальшої обробки. У цій роботі як основний сенсорний елемент розглядається APDS-9960, який підтримує розпізнавання жестів, визначення наближення, вимірювання освітленості та кольору.

Актуальність теми полягає у розробці компактного апаратно-програмного пристрою, здатного перетворювати прості жести користувача на керуючі команди. Такий пристрій може бути корисним у мультимедійних, навчальних, презентаційних, лабораторних або побутових сценаріях. Особливий інтерес становить не лише сам факт використання датчика APDS-9960, а й побудова повної логіки роботи системи – від зчитування сигналу до фільтрації випадкових спрацьовувань і формування команди для керованого пристрою.

Проблема розробки полягає в тому, що просте підключення сенсора жестів ще не забезпечує стабільної роботи системи. Потрібно врахувати робочу відстань, рівні напруги, особливості обміну через I2C, затримки між жестами, хибні спрацьовування, а також обмеження самого сенсора. APDS-9960 не є універсальним засобом для розпізнавання складних рухів руки, тому функціональність пристрою має бути правильно обмежена. Саме це дозволяє зробити систему більш передбачуваною та придатною для практичного використання.

Об’єктом дослідження є процес безконтактної взаємодії людини з електронним пристроєм за допомогою жестових команд.

Предметом дослідження є апаратно-програмне забезпечення для зчитування, обробки та інтерпретації жестів на основі оптичного датчика APDS-9960.

Метою роботи є розробка апаратно-програмного пристрою керування на основі датчика APDS-9960, який забезпечує розпізнавання простих жестів користувача та передавання відповідних HID-команд до керованого пристрою.

Для досягнення поставленої мети необхідно вирішити такі **задачі**:

– проаналізувати безконтактне керування як спосіб взаємодії з електронними пристроями та розглянути основні засоби розпізнавання жестів;

– дослідити технічні можливості APDS-9960, принцип його роботи, програмні засоби реалізації та сформувати вимоги до пристрою керування;

– визначити сценарій використання пристрою, обґрунтувати вибір мікроконтролера та способу підключення APDS-9960;

– розробити структурну схему, схему підключення пристрою, алгоритм зчитування та інтерпретації жестових команд;

– реалізувати мікропрограмне забезпечення для роботи з APDS-9960 та передавання команд до керованого пристрою;

– виконати апаратно-програмну перевірку роботи пристрою, оцінити точність розпізнавання жестів і стабільність формування HID-команд.

Під час виконання роботи використовуються методи аналізу технічної документації, порівняння апаратних рішень, структурного проєктування, алгоритмізації, мікропрограмної реалізації та технічного тестування. Для перевірки працездатності пристрою передбачається оцінювання стабільності розпізнавання жестів, часу реакції системи, кількості помилкових спрацьовувань і залежності результату від умов виконання жесту.

Практичне значення роботи полягає у створенні прототипу пристрою керування, який може використовуватися для виконання простих команд без фізичного контакту з елементами керування. Такий підхід може бути застосований у презентаційних системах, мультимедійних пристроях, навчальному обладнанні, лабораторних стендах або вбудованих системах, де достатньо невеликого набору команд і важлива простота взаємодії.

Апробація результатів кваліфікаційної бакалаврської роботи відбулася під час XXVIII Всеукраїнської щорічної науково-практичної конференції «Могилянські читання – 2025» (м. Миколаїв, 10–14 листопада 2025 р.) [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИХІДНИХ ВИМОГ

1.1 Безконтактне керування як спосіб взаємодії з електронними пристроями

Взаємодія людини з електронними пристроями традиційно пов’язана з прямим фізичним контактом. Користувач натискає кнопки, працює з клавіатурою, мишею, сенсорним екраном або панеллю керування. Такі засоби залишаються зручними для точного введення даних, налаштування параметрів і роботи зі складними інтерфейсами. Проте для коротких команд вони не завжди є найраціональнішим варіантом.

У багатьох ситуаціях користувачу потрібно виконати лише одну просту дію: перейти до наступного слайда, поставити відтворення на паузу, змінити гучність, підтвердити команду або перемкнути режим. Для цього не обов’язково відкривати меню чи шукати потрібну клавішу. Достатньо швидкого сигналу, який пристрій може розпізнати й перетворити на відповідну команду.

Безконтактне керування ґрунтується саме на такій ідеї. Команда подається не через натискання, а через рух руки, наближення до сенсора або інший жест у робочій зоні пристрою. У технічному плані цей процес складається з трьох основних етапів: зчитування дії користувача сенсором, обробки отриманих даних мікроконтролером і передавання результату до керованого пристрою [2–4].

Жестове керування найкраще працює тоді, коли набір команд невеликий і логічно зрозумілий. Наприклад, рух вправо може відповідати переходу до наступного елемента, рух вліво – поверненню назад, рух вгору або вниз – зміні певного параметра. За такого підходу користувачу не потрібно запам’ятовувати складні комбінації, а сама взаємодія з пристроєм стає швидшою.

Найбільш очевидними прикладами є мультимедійні та презентаційні системи. Під час виступу користувач не завжди перебуває біля ноутбука, а постійне звернення до клавіатури або миші може відволікати від доповіді. У побутових умовах схожа ситуація виникає під час перегляду відео або прослуховування музики, коли потрібно швидко виконати одну дію без пошуку кнопки чи пульта [5].

Безконтактне керування також має сенс у середовищах, де небажано часто торкатися поверхонь обладнання. Це можуть бути лабораторії, медичні кабінети, навчальні аудиторії або спільні робочі місця. У виробничих умовах користувач може працювати в рукавичках, мати зайняті руки або перебувати біля обладнання, де використання дрібних кнопок є незручним. Основні напрями застосування такого підходу подано в табл. 1.1.

Таблиця 1.1 – Сфери застосування безконтактного жестового керування

Сфера застосування	**Приклади використання**	**Практичне значення**
Мультимедійні системи	Керування відтворенням, паузою, гучністю, перемиканням треків	Дає змогу швидко виконувати типові команди без пошуку кнопок або пульта
Презентаційне обладнання	Перехід між слайдами, запуск демонстрації, зупинка показу	Дозволяє керувати виступом без постійного звернення до клавіатури або миші
Медичні та лабораторні робочі місця	Перегляд матеріалів, перемикання режимів, робота зі спільним обладнанням	Зменшує кількість фізичних дотиків до поверхонь пристроїв
Промислові панелі керування	Активація допоміжних функцій, перемикання режимів, підтвердження простих команд	Полегшує роботу в рукавичках або в умовах, коли прямий контакт із панеллю незручний
Елементи «розумного» середовища	Керування освітленням, побутовими пристроями, локальними сценаріями автоматизації	Робить взаємодію з пристроями більш швидкою та природною для користувача

Дані табл. 1.1 показують, що безконтактне жестове керування не обмежується однією сферою. Його доцільно використовувати там, де потрібні короткі повторювані команди, а не повноцінне введення даних. У таких умовах жест не замінює клавіатуру або мишу, а виконує роль допоміжного інтерфейсу для швидких дій.

Технічно безконтактне керування може реалізовуватися різними способами. Камерні системи дають змогу розпізнавати складні рухи руки та положення пальців, але потребують обробки зображення, достатнього освітлення і більшої обчислювальної потужності. Для компактного пристрою з кількома базовими командами такий підхід часто є надмірним.

Інший варіант – використання спеціалізованих сенсорів ближньої дії. Вони не формують зображення, а працюють із фізичними сигналами: відбитим

інфрачервоним світлом, зміною ємності, відстанню до об’єкта або іншими параметрами. Такі рішення мають меншу універсальність, зате краще підходять для невеликих апаратно-програмних систем.

Датчик APDS-9960 належить до оптичних сенсорів ближньої дії [1]. Він використовує інфрачервоне випромінювання та аналізує зміну відбитого сигналу від руки користувача. За характером зміни сигналів можна визначити базові напрямлені жести: вліво, вправо, вгору або вниз. Цього недостатньо для складного керування, але для пристрою з обмеженим набором функцій такий підхід є практично придатним.

Для пристрою на основі APDS-9960 найбільш виправданим є сценарій ближнього керування електронним пристроєм. Користувач виконує простий рух рукою, сенсор фіксує зміну сигналу, а мікроконтролер перетворює цей жест на команду для мультимедійної системи, презентаційного застосунку, локального контролера або іншого обладнання. Такий підхід дозволяє побудувати компактну апаратно-програмну систему без використання камери та складної обробки зображення.

1.2 Класифікація засобів розпізнавання жестів

Засоби розпізнавання жестів відрізняються за способом отримання інформації про рух користувача. Одні системи працюють із відеозображенням, інші аналізують відбитий сигнал, зміну електричного поля, відстань до об’єкта або дані з датчиків руху. Через це жестові інтерфейси не можна розглядати як одну технологію з однаковими можливостями. Кожен підхід має власні обмеження, вартість реалізації та вимоги до обчислювальних ресурсів.

Для КБР важливо не просто перелічити можливі технології, а відокремити ті рішення, які реально можуть бути використані у компактному пристрої ближнього керування. Якщо система повинна тільки перемикати слайд, зменшувати або збільшувати параметр і підтверджувати дію, використання складних засобів розпізнавання часто не дає суттєвої переваги. Навпаки, надмірна складність може ускладнити налагодження та зробити пристрій менш передбачуваним.

Найбільш загально такі засоби можна поділити на контактні та безконтактні. Контактні рішення передбачають використання додаткового пристрою на руці користувача: рукавички з датчиками, браслета, контролера руху або пульта з інерційними сенсорами. Вони можуть забезпечувати достатньо точне зчитування рухів, але потребують окремого обладнання, яке треба тримати, заряджати або закріплювати на користувачі. Для простого пристрою керування це не завжди зручно.

Безконтактні засоби розміщуються в самому пристрої або поруч із ним і зчитують рух користувача на певній відстані. До цієї групи належать камерні системи, оптичні ІЧ-сенсори, ємнісні сенсори, ультразвукові датчики, ToF-сенсори та радарні модулі [7; 10; 12]. Саме цей підхід відповідає темі розробки, оскільки пристрій на основі APDS-9960 має реагувати на рух руки без використання додаткових носимих елементів.

Камерні системи мають найширші можливості. Вони можуть розпізнавати складні жести, положення руки, форму пальців і траєкторію руху. Проте для цього потрібно обробляти зображення або відеопотік, враховувати освітлення, фон, положення камери та можливі перешкоди. Такий підхід доцільний для складних інтерактивних систем, але для невеликого пристрою з кількома командами він часто є надмірним.

Оптичні ІЧ-сенсори ближньої дії працюють простіше. Вони використовують інфрачервоне випромінювання, яке відбивається від руки користувача і потрапляє на фотоприймачі. За зміною сигналів можна визначити напрям руху руки. До цієї групи належить APDS-9960 [7]. Його можливості обмежені порівняно з камерою, однак для жестів типу «вліво», «вправо», «вгору» або «вниз» цього достатньо.

Ємнісні сенсори реагують на зміну електричного поля біля поверхні. Вони добре підходять для сенсорних кнопок або панелей керування, але мають невелику робочу зону. Ультразвукові датчики частіше застосовуються для визначення відстані або наявності об’єкта. Для розпізнавання напрямлених жестів одного такого датчика зазвичай недостатньо, тому схема може потребувати кількох сенсорів.

ToF-сенсори та радарні модулі дають більше можливостей для просторового аналізу руху й можуть працювати стабільніше за складних умов освітлення. Водночас вони дорожчі та складніші в інтеграції. Для навчального або прикладного прототипу з невеликим набором команд таке рішення не завжди виправдане.

Під час вибору технології слід враховувати також очікувану поведінку користувача. Жестове керування має бути зрозумілим без довгого навчання, тому найкраще працюють напрямлені рухи, які користувач інтуїтивно пов’язує з певною дією. Наприклад, рух вправо природно відповідає переходу вперед, а рух вліво – поверненню назад. Якщо сенсор підтримує велику кількість жестів, але частина з них нестабільно розпізнається в реальних умовах, практична користь такої функціональності зменшується.

Основні групи засобів розпізнавання жестів подано в табл. 1.2.

Таблиця 1.2 – Класифікація засобів розпізнавання жестів

Група засобів	Принцип роботи	Типові можливості	Основні обмеження
Камерні системи	Аналіз зображення або відеопотоку	Розпізнавання складних жестів, положення руки, форми пальців	Високі вимоги до обробки даних, залежність від освітлення та фону
Оптичні ІЧ-сенсори ближньої дії	Аналіз відбитого інфрачервоного сигналу	Визначення простих напрямлених жестів і наближення	Невелика робоча відстань, обмежений набір жестів
Ємнісні сенсори	Фіксація зміни електричного поля біля поверхні	Сенсорні кнопки, наближення руки, прості дії біля панелі	Мала зона дії, складність розпізнавання напрямлених жестів
Ультразвукові датчики	Вимірювання відстані до об’єкта за відбитим сигналом	Виявлення наближення або присутності руки	Для жестів потрібні кілька датчиків або додаткова обробка
ToF-сенсори та радарні модулі	Визначення відстані й руху за часовим або хвильовим принципом	Просторове розпізнавання руху, робота при слабкому освітленні	Вища вартість, складніша апаратна та програмна реалізація
Носимі контролери	Зчитування руху через акселерометри, гіроскопи або датчики згину	Висока точність для рухів руки, стабільне зчитування жестів	Потрібен окремий пристрій на руці користувача

З табл. 1.2 видно, що кожна група має власне призначення. Камерні системи доцільні для складних жестів, носимі контролери – для задач із високою точністю руху, а ємнісні й ультразвукові рішення краще підходять для простого виявлення

наближення. Для компактного пристрою ближнього керування найбільш збалансованими є оптичні ІЧ-сенсори, оскільки вони не потребують обробки відео й можуть працювати з мікроконтролером.

APDS-9960 у цій класифікації займає місце сенсора для коротких напрямлених жестів. Його не варто порівнювати з камерними системами за функціональністю, оскільки він не аналізує форму руки або положення пальців. Його перевага полягає в іншому: простому підключенні, компактності, невисоких вимогах до обчислень і достатніх можливостях для базових команд. Саме ці властивості роблять APDS-9960 придатним для подальшого використання в пристрої керування.

Для розроблюваного пристрою найбільш важливою є не максимальна кількість розпізнаваних жестів, а повторюваність результату. Якщо один і той самий рух руки кожного разу викликає однакову команду, пристрій сприймається користувачем як надійний. Тому у подальшому проєктуванні основна увага приділяється не розширенню набору жестів, а правильному підключенню датчика, фільтрації повторних спрацьовувань і формуванню зрозумілої карти команд.

1.3 Технічні можливості та принцип роботи APDS-9960

APDS-9960 є цифровим оптичним датчиком ближньої дії, який поєднує кілька функцій: розпізнавання жестів, визначення наближення, вимірювання рівня освітленості та зчитування кольору за каналами RGBC [7]. Для пристрою керування основне значення має жестовий режим, оскільки саме він дає змогу фіксувати рух руки без використання камери та складної обробки зображення.

Для практичної реалізації важливо, що APDS-9960 передає дані в цифровому вигляді через інтерфейс I2C. Це зменшує кількість необхідних з’єднань і спрощує роботу з мікроконтролером. На відміну від аналогових датчиків, де потрібно додатково обробляти рівень напруги на вході, у цьому випадку програма отримує вже підготовлені значення з внутрішніх регістрів сенсора.

Зовнішній вигляд модуля APDS-9960 наведено на рис. 1.1.

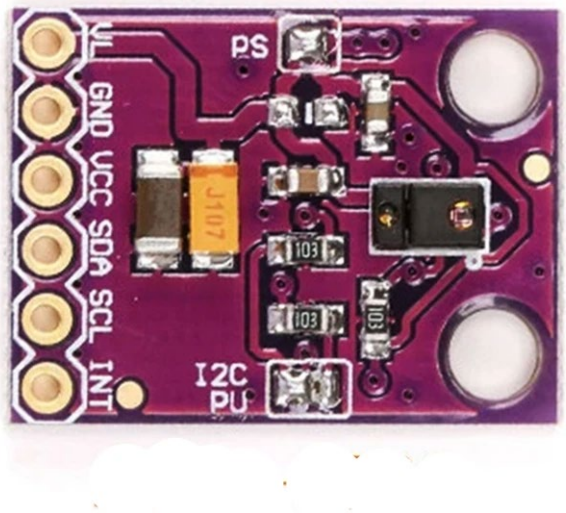

Рисунок 1.1 – Модуль датчика жестів APDS-9960 [7]

Типовий модуль APDS-9960 має контактні виводи для живлення та обміну даними через інтерфейс I2C. Для підключення до мікроконтролера використовуються лінії SDA та SCL, а також, за потреби, вивід INT для роботи з перериваннями. Такий формат підключення є зручним для компактних апаратно-програмних пристроїв, оскільки не потребує великої кількості зовнішніх елементів [7; 13].

Принцип роботи APDS-9960 ґрунтується на використанні інфрачервоного випромінювання. Датчик випромінює ІЧ-світло, яке відбивається від руки або іншого об’єкта перед сенсором і потрапляє на внутрішні фотоприймачі. Якщо рука рухається перед модулем, рівень відбитого сигналу на приймальних каналах змінюється не одночасно. За цією послідовністю зміни сигналів можна визначити напрям руху.

У жестовому режимі використовуються чотири напрямлені канали: U, D, L та R, що відповідають напрямам вгору, вниз, вліво та вправо. Наприклад, під час руху руки зліва направо спочатку змінюється сигнал з одного боку сенсора, а потім з іншого. Для руху вгору або вниз послідовність зміни каналів буде іншою. Саме порівняння таких змін дозволяє визначити базовий жест [7].

APDS-9960 не формує зображення руки, як це робить камера. Він не аналізує форму пальців або положення кисті в просторі, а працює з числовими значеннями

відбитого світла. Для пристрою керування це є перевагою, оскільки система залишається простою та може працювати на мікроконтролері без значних обчислювальних ресурсів.

Спрощений принцип визначення жесту за допомогою APDS-9960 подано на рис. 1.2.

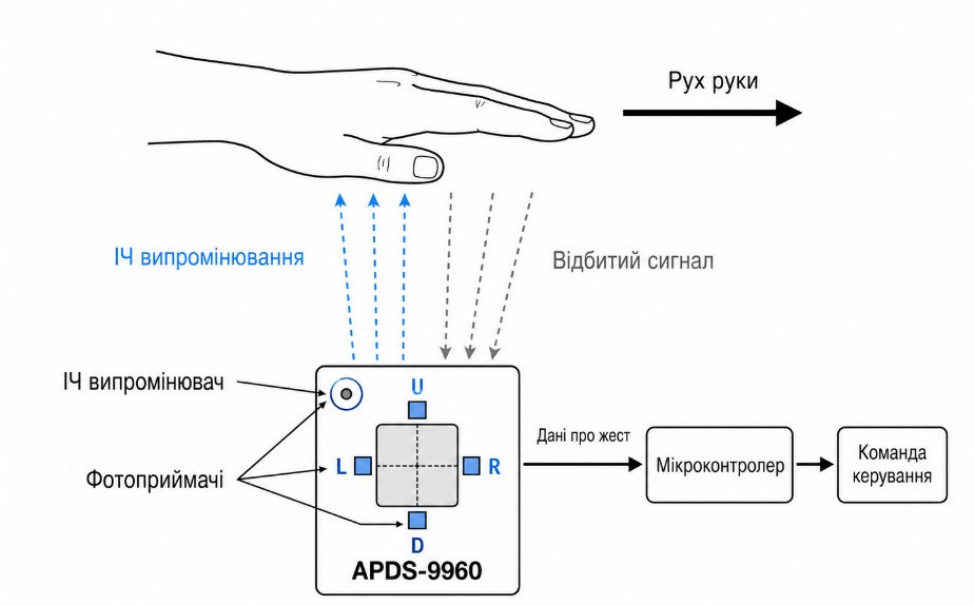

Рисунок 1.2 – Принцип визначення жесту датчиком APDS-9960

На рис. 1.2 показано загальну логіку роботи сенсора. Рука користувача проходить перед датчиком, інфрачервоне випромінювання відбивається від її поверхні, після чого фотоприймачі фіксують зміну сигналу за напрямами. Далі мікроконтролер отримує дані через цифровий інтерфейс і на їх основі формує керуючу команду.

Перед розпізнаванням жесту може використовуватися режим визначення наближення. Він дає змогу перевірити, чи знаходиться рука користувача в робочій зоні сенсора. Якщо об’єкт розташований занадто далеко або відсутній, жестова обробка може не активуватися. Це зменшує кількість випадкових спрацьовувань і робить роботу пристрою стабільнішою.

Жестові дані в APDS-9960 накопичуються у внутрішньому буфері FIFO. У ньому зберігаються послідовні значення для каналів U, D, L та R. Мікроконтролер може зчитувати ці дані й обробляти їх програмно. У простому варіанті

використовується готова бібліотека, яка повертає вже визначений напрям жесту. Для більш надійної роботи поверх цього можна додати перевірку повторних спрацьовувань, затримку між командами та фільтрацію випадкових рухів.

Окремої уваги потребує робоча зона датчика. У технічній документації та описах готових модулів APDS-9960 зазвичай вказується, що сенсор розрахований на ближню взаємодію. На практиці для жестового керування найзручнішою є відстань приблизно 5–20 см. Якщо рука проходить занадто далеко від сенсора, відбитий сигнал стає слабшим. Якщо рух виконується надто близько, датчик може сприймати його як наближення або формувати неповну послідовність даних.

Основні технічні можливості APDS-9960 наведено в табл. 1.3.

Таблиця 1.3 – Основні технічні можливості APDS-9960

Характеристика	Опис	Значення для пристрою керування
Розпізнавання жестів	Визначення рухів руки за напрямами вліво, вправо, вгору, вниз	Дозволяє формувати базові команди керування
Датчик наближення	Фіксація появи об’єкта перед сенсором	Може використовуватися для активації обробки жестів
Вимірювання освітленості	Оцінка рівня зовнішнього світла	Дає змогу враховувати умови середовища
Зчитування кольору RGBC	Вимірювання червоного, зеленого, синього та загального каналів	Для основної функції керування не є обов’язковим, але може використовуватися в розширених режимах
Інтерфейс I2C	Цифровий обмін даними з мікроконтролером	Спрощує підключення та програмне керування датчиком
Вивід INT	Сигнал переривання при появі події	Дає змогу не опитувати сенсор постійно
Робоча зона	Ближня відстань перед датчиком	Визначає спосіб виконання жестів користувачем

З табл. 1.3 видно, що APDS-9960 є багатофункціональним оптичним модулем. У межах розроблюваного пристрою основною є функція розпізнавання жестів, а датчик наближення може використовуватися як допоміжний елемент для визначення моменту початку взаємодії користувача з пристроєм.

Під час підключення APDS-9960 потрібно враховувати його електричні характеристики. Датчик належить до низьковольтних компонентів, тому його не можна без перевірки підключати до 5-вольтової логіки. Якщо використовується мікроконтролер із рівнями 5 В, потрібне узгодження сигналів або модуль із

перетворювачем рівнів. Для контролерів із логікою 3,3 В підключення є простішим, але напруга живлення все одно має відповідати характеристикам конкретного модуля.

Стабільність розпізнавання залежить не лише від самого датчика. На роботу впливають відстань до руки, швидкість руху, зовнішнє освітлення, положення модуля та конструкція корпусу. Якщо сенсор перекритий непрозорим матеріалом або розташований занадто глибоко, якість зчитування може погіршитися. Тому під час проєктування потрібно залишити перед датчиком відкриту зону або прозоре вікно для проходження ІЧ-випромінювання.

У практичному пристрої датчик не повинен розташовуватися поруч із елементами, які можуть випадково відбивати ІЧ-сигнал. Це стосується не тільки корпусу, а й дротів, кріплень або декоративних елементів. Для макетного прототипу достатньо розмістити APDS-9960 так, щоб перед ним залишалася відкрита область для проходження руки. Для завершеного варіанта пристрою доцільно передбачити невелике вікно або виріз у корпусі.

APDS-9960 доцільно розглядати як сенсор для простих жестів у ближній зоні. Він не призначений для складного аналізу рухів руки, але добре підходить для пристрою, який має перетворювати кілька базових жестів на керуючі команди. Саме ці особливості визначають подальшу логіку проєктування апаратної та програмної частини системи.

1.4 Порівняльний аналіз APDS-9960, PAJ7620U2 та ToF-сенсорів ближньої дії

Для вибору сенсорного модуля важливо порівнювати не лише кількість підтримуваних жестів. У компактному пристрої керування суттєвими є робоча зона, спосіб підключення, вимоги до мікроконтролера, стабільність у ближній зоні та складність програмної інтеграції. Тому доцільно розглядати рішення одного класу – невеликі оптичні та дистанційні модулі, які можуть працювати без відеокамери та окремого комп’ютерного зору.

APDS-9960 належить до оптичних ІЧ-сенсорів ближньої дії. Його робота орієнтована на короткі рухи руки перед модулем. Практична зона для жестового керування становить приблизно 5–20 см, залежно від освітлення, положення руки та конкретного модуля [7; 13]. Для пристрою з невеликою кількістю команд цього достатньо, оскільки користувач взаємодіє з датчиком на близькій відстані.

PAJ7620U2 також використовується для розпізнавання жестів, але має ширший набір попередньо визначених рухів. Типова зона роботи таких модулів становить близько 5–15 см [8]. Це робить PAJ7620U2 зручним для готових жестових інтерфейсів, однак у межах цієї роботи важливішою є не максимальна кількість жестів, а простота логіки та можливість пояснити повний шлях від зчитування сигналу до формування команди.

ToF-сенсори, наприклад модулі на основі VL53L0X, працюють за принципом вимірювання відстані до об’єкта. Вони можуть визначати наближення або віддалення руки в діапазоні до кількох метрів, але самі по собі не розпізнають напрямлені жести типу LEFT або RIGHT без додаткової схеми з кількох сенсорів чи складнішої програмної обробки. Тому для цієї роботи такі модулі доцільно розглядати як альтернативу для задач вимірювання відстані, а не як прямий аналог APDS-9960.

Під час порівняння цих рішень варто враховувати й те, як саме користувач буде взаємодіяти з пристроєм. Для презентаційного або мультимедійного контролера не потрібна велика дальність роботи. Навпаки, занадто велика зона спрацювання може призводити до випадкових реакцій на рухи, які не були призначені як команда. У цьому сенсі ближня зона APDS-9960 є не лише обмеженням, а й способом зробити керування більш контрольованим.

Порівняння APDS-9960, PAJ7620U2 та ToF-сенсорів ближньої дії наведено в табл. 1.4.

Таблиця 1.4 – Порівняння APDS-9960, PAJ7620U2 та ToF-сенсорів ближньої дії

Критерій	**APDS-9960**	**PAJ7620U2**	**ToF-сенсор ближньої дії**
Принцип роботи	Аналіз відбитого інфрачервоного сигналу за напрямленими каналами	Оптичне розпізнавання жестів із вбудованою логікою	Вимірювання часу проходження світлового імпульсу до об’єкта і назад
Типова робоча зона	приблизно 5–20 см для жестів руки	приблизно 5–15 см для готових жестів	до 200 см для вимірювання відстані, але без готового розпізнавання жестів
Тип команд	LEFT, RIGHT, UP, DOWN, NEAR/FAR	ширший набір попередньо визначених жестів	наближення, віддалення, зміна дистанції
Інтерфейс	I2C	I2C	I2C
Вимоги до обчислень	низькі	низькі або середні	низькі для відстані, вищі для складної інтерпретації руху
Доцільність для пристрою	висока для 4–5 базових команд	висока, якщо потрібна більша кількість жестів	середня; потрібна додаткова логіка для жестів

Дані табл. 1.4 показують, що APDS-9960 не є найпотужнішим сенсором за кількістю жестів, але він добре відповідає задачі компактного пристрою ближнього керування. Його можливостей достатньо для команд вліво, вправо, вгору та вниз, а підключення через I2C дає змогу реалізувати пристрій на доступному мікроконтролері ESP32.

PAJ7620U2 може бути кращим варіантом тоді, коли потрібна більша кількість готових жестів. ToF-модулі мають інше призначення: вони краще підходять для вимірювання дистанції, контролю наближення або просторових сценаріїв із додатковою обробкою. Для розроблюваного пристрою обрано APDS-9960, оскільки він забезпечує достатню функціональність, не потребує камери та дозволяє будувати зрозумілу апаратно-програмну логіку.

Ще однією перевагою APDS-9960 є те, що його обмеження легко врахувати на етапі проєктування. Пристрій можна одразу орієнтувати на роботу в ближній зоні, передбачити невелику затримку після команди та використовувати лише ті

жести, які розпізнаються стабільно. Це краще, ніж намагатися реалізувати велику кількість команд, частина з яких буде працювати нестабільно в реальних умовах.

1.5 Аналіз бібліотек і засобів програмної реалізації

Програмна реалізація пристрою на основі APDS-9960 передбачає кілька послідовних дій: ініціалізацію датчика, налаштування обміну через I2C, зчитування жестів, перевірку отриманих даних і формування керуючої команди. Для цього доцільно використовувати готові бібліотеки, оскільки вони спрощують роботу з регістрами сенсора та дозволяють швидше перевірити працездатність апаратного підключення.

Основним інтерфейсом взаємодії з APDS-9960 є I2C [7; 14]. У середовищі Arduino для цього зазвичай використовується бібліотека Wire, яка забезпечує передавання даних між мікроконтролером і сенсорним модулем. Через цю шину виконуються команди ініціалізації, налаштування режимів роботи та зчитування інформації про жест. Такий підхід зручний для компактного пристрою, оскільки потребує лише двох основних сигнальних ліній – SDA та SCL.

Для самого APDS-9960 можуть використовуватися готові бібліотеки, які надають функції запуску датчика, перевірки доступності модуля, увімкнення жестового режиму та отримання напряму руху руки [15]. У найпростішому варіанті програма перевіряє, чи доступний новий жест, після чого отримує його результат у вигляді готового значення. Це дозволяє не працювати вручну з усіма внутрішніми регістрами сенсора на початковому етапі розробки.

Разом з тим готова бібліотека не повністю визначає поведінку пристрою. Вона може повернути напрям жесту, але не завжди враховує конкретні умови використання: відстань до руки, швидкість руху, можливе дублювання команди або випадкові спрацьовування. Тому поверх бібліотечних функцій необхідно реалізувати власну прикладну логіку. До неї належать затримка між командами, перевірка повторних жестів, карта відповідності жестів діям і діагностичний вивід під час налагодження.

Основні програмні засоби, які можуть застосовуватися під час розробки пристрою, наведено в табл. 1.5.

Таблиця 1.5 – Програмні засоби для реалізації пристрою на основі APDS-9960

Засіб або бібліотека	Призначення	Практичне використання в пристрої
Wire	Обмін даними через інтерфейс I2C	Забезпечує зв’язок між мікроконтролером і датчиком APDS-9960
Бібліотека для APDS-9960	Ініціалізація сенсора та зчитування жестів	Дозволяє швидко отримати напрям жесту без ручної роботи з усіма регістрами
Arduino IDE	Середовище написання, компіляції та завантаження прошивки	Підходить для швидкого прототипування та перевірки роботи датчика
PlatformIO	Розширене середовище розробки для вбудованих систем	Зручне для структурованого проєкту, роботи з бібліотеками та різними платами
Бібліотеки HID	Передавання команд як клавіатура, миша або мультимедійний пристрій	Використовуються, якщо пристрій має керувати комп’ютером через USB або Bluetooth
Serial Monitor	Виведення діагностичних повідомлень	Допомагає перевіряти, який жест розпізнано та яку команду сформовано

З табл. 1.5 видно, що програмна частина складається з кількох рівнів. На нижньому рівні відбувається обмін даними через I2C, далі працює бібліотека для APDS-9960, а прикладна логіка вже визначає, яку дію потрібно виконати після розпізнавання жесту. Такий поділ зручний для налагодження, оскільки кожен етап можна перевіряти окремо.

Для початкової реалізації достатньо середовища Arduino IDE, оскільки воно має простий механізм підключення бібліотек і дозволяє швидко завантажити тестову прошивку на плату [16]. PlatformIO доцільно використовувати тоді, коли проєкт стає більшим і потребує кращої структури файлів, контролю бібліотечних залежностей та окремих параметрів збірки [17].

Якщо пристрій має керувати комп’ютером, важливим стає використання HID-бібліотек. У такому режимі мікроконтролер може передавати команди як клавіатура, миша або мультимедійний контролер [18–19]. Наприклад, жест вправо може відповідати переходу до наступного елемента, а жест вліво – поверненню до

попереднього. Це зручно, оскільки операційна система сприймає пристрій як стандартний засіб введення. Для ESP32 такий підхід доцільно реалізувати через Bluetooth HID, оскільки звичайний USB-порт більшості плат ESP32 DevKit використовується переважно для прошивки та послідовного обміну [16].

Під час налагодження доцільно використовувати Serial Monitor. Через нього можна виводити повідомлення про стан сенсора, розпізнаний жест і команду, яка була сформована програмою. Це спрощує пошук помилок у підключенні, бібліотеці або власній логіці обробки.

Для розроблюваного пристрою програмна реалізація має будуватися поступово: спочатку перевіряється зв’язок з APDS-9960, потім зчитування жестів, після цього додається фільтрація повторних спрацьовувань і карта команд. Такий підхід дозволяє зменшити кількість помилок під час розробки та забезпечити більш стабільну роботу системи в реальних умовах.

1.6 Формування вимог до пристрою керування та критеріїв оцінювання

На основі аналізу безконтактних інтерфейсів, технічних можливостей APDS-9960 та програмних засобів реалізації можна сформувати вимоги до пристрою керування. Вони мають враховувати реальні можливості сенсора, а не лише бажаний набір функцій. APDS-9960 працює в ближній зоні та орієнтований на прості напрямлені жести, тому пристрій доцільно проєктувати для коротких команд, які користувач може виконувати без спеціального навчання.

Основна функція пристрою полягає в тому, щоб зафіксувати рух руки перед сенсором, визначити напрям жесту та перетворити його на керуючу команду. Такий підхід підходить для керування мультимедіа, презентаціями, локальними режимами роботи або іншими простими діями. Доцільно використовувати базові жести: рух вліво, вправо, вгору та вниз. За потреби окремо може враховуватися наближення руки, але воно має виконувати допоміжну роль, наприклад для активації пристрою або зменшення випадкових спрацьовувань.

Вимоги до пристрою можна умовно поділити на функціональні, апаратні, програмні та експлуатаційні. Функціональні вимоги визначають, які саме команди

має виконувати система. Апаратні вимоги стосуються правильного підключення APDS-9960, живлення, логічних рівнів та стабільності I2C-обміну. Програмні вимоги охоплюють ініціалізацію датчика, обробку жестів, фільтрацію помилкових подій і формування HID-команд. Експлуатаційні вимоги пов’язані з тим, щоб пристрій був зрозумілим для користувача й не потребував складного налаштування перед кожним використанням.

Основні вимоги до пристрою та критерії його оцінювання подано в табл. 1.6.

Таблиця 1.6 – Вимоги до пристрою керування та критерії оцінювання

Група	Вимога або критерій	Практичне значення
Функціональні вимоги	Розпізнавання базових жестів вліво, вправо, вгору та вниз	Забезпечує виконання основних команд керування
	Відповідність кожного жесту конкретній дії	Робить роботу пристрою зрозумілою для користувача
	Захист від повторних і випадкових спрацьовувань	Підвищує стабільність роботи системи
Апаратні вимоги	Підключення APDS-9960 до мікроконтролера через I2C	Забезпечує цифровий обмін даними з сенсором
	Узгодження рівнів напруги між сенсором і мікроконтролером	Запобігає некоректній роботі або пошкодженню модуля
Програмні вимоги	Ініціалізація датчика, зчитування жестів і формування команд	Визначає основну логіку роботи пристрою
	Наявність затримки після розпізнавання жесту	Зменшує дублювання однієї команди
Критерії оцінювання	Точність розпізнавання жестів	Перевіряється кількістю правильних спрацьовувань у серії повторів
	Час реакції системи	Оцінюється затримкою між жестом і виконанням команди
	Кількість хибних спрацьовувань	Показує, наскільки пристрій стійкий до випадкових рухів
	Стабільність на різній відстані	Дозволяє визначити оптимальну робочу зону сенсора

З табл. 1.6 видно, що вимоги стосуються не лише датчика APDS-9960, а й усієї апаратно-програмної системи. Навіть якщо сенсор правильно визначає жест, пристрій може працювати незручно без затримки між командами або без логічної карти відповідності жестів діям. Тому важливо оцінювати не окремий модуль, а повну реакцію системи на дії користувача.

Найбільш важливими критеріями для подальшого тестування є точність розпізнавання, час реакції та кількість хибних спрацьовувань. Для перевірки

точності кожен жест потрібно виконати кілька разів у однакових умовах і порівняти кількість правильних результатів із загальною кількістю спроб. Час реакції показує, наскільки швидко система переходить від зчитування жесту до виконання команди. Якщо затримка буде занадто великою, користувач сприйматиме пристрій як повільний.

Окремо потрібно враховувати робочу відстань. APDS-9960 є сенсором ближньої дії, тому жести мають виконуватися в обмеженій зоні перед модулем. Якщо рука знаходиться занадто далеко, відбитий сигнал може бути слабким. Якщо рух виконується надто близько або занадто швидко, сенсор може сформувати неповні дані для розпізнавання. Через це під час технічної перевірки потрібно визначити діапазон, у якому пристрій працює найбільш стабільно.

Карта команд має бути простою та інтуїтивною. Наприклад, жест вправо доцільно пов’язати з переходом до наступного елемента, жест вліво – з поверненням до попереднього, рух вгору або вниз – зі зміною параметра. Така відповідність полегшує використання пристрою і зменшує кількість помилок з боку користувача.

Для оцінювання пристрою також доцільно враховувати повторюваність результату. Якщо один жест іноді спрацьовує як інший, система не може вважатися зручною, навіть якщо загальна кількість правильних спрацьовувань виглядає прийнятною. Тому під час тестування потрібно фіксувати не лише факт розпізнавання, а й відповідність сформованої HID-команди очікуваній дії.

Сформовані вимоги та критерії оцінювання є основою для подальшого проєктування. На їх основі у наступному розділі можна перейти до вибору сценарію використання, мікроконтролера, схеми підключення APDS-9960, алгоритму обробки жестів і карти команд керування.

Висновки до розділу 1

У першому розділі було розглянуто предметну область безконтактного керування електронними пристроями. Визначено, що жестовий інтерфейс доцільно використовувати для простих і повторюваних команд, зокрема перемикання, підтвердження дії, керування мультимедіа, презентаціями або окремими режимами роботи пристрою.

Було проаналізовано основні засоби розпізнавання жестів: камерні, ємнісні, ультразвукові, ToF-, радарні та оптичні рішення. Для компактного пристрою ближнього керування найбільш придатними є оптичні ІЧ-сенсори, оскільки вони не потребують складної обробки зображень і великих обчислювальних ресурсів.

Окремо розглянуто датчик APDS-9960, який працює на основі аналізу інфрачервоного випромінювання, відбитого від руки користувача. Встановлено, що він може розпізнавати базові напрямлені жести: вліво, вправо, вгору та вниз. Такого набору достатньо для реалізації простого пристрою керування.

Порівняння APDS-9960, PAJ7620U2 та ToF-сенсорів ближньої дії показало, що APDS-9960 має обмеженіший набір жестів, але краще відповідає задачі простого керування, оскільки працює через I2C, має невелику робочу зону та не потребує складної просторової обробки.

2 ПРОЄКТУВАННЯ ПРИСТРОЮ КЕРУВАННЯ

2.1 Вибір сценарію використання та функціонального призначення

Після аналізу предметної області можна перейти до визначення практичного сценарію використання пристрою. Для датчика APDS-9960 найбільш доцільним є не складний універсальний інтерфейс, а компактний пристрій ближнього керування, який реагує на кілька простих жестів руки. Такий підхід відповідає технічним можливостям сенсора і не створює зайвих вимог до програмної частини [7; 13].

Основним сценарієм використання обрано керування електронним пристроєм за допомогою базових напрямлених жестів. Користувач виконує рух рукою перед сенсором APDS-9960, мікроконтролер зчитує результат розпізнавання та перетворює його на відповідну команду. У такому режимі пристрій може застосовуватися для керування презентаціями, мультимедійними функціями або простими локальними режимами роботи обладнання. Для розроблюваного прототипу важливо, щоб взаємодія залишалася зрозумілою без додаткового навчання. Тому команди мають бути пов’язані з природним напрямом руху руки. Наприклад, жест вправо може використовуватися для переходу до наступного елемента, жест вліво – для повернення до попереднього, рух вгору – для збільшення параметра, а рух вниз – для його зменшення. Така логіка є простою і добре підходить для обмеженого набору команд.

Функціональне призначення пристрою полягає у виконанні ролі проміжного вузла між користувачем і керованою системою. Сенсор APDS-9960 відповідає за фіксацію жесту, мікроконтролер – за обробку даних і формування команди, а керований пристрій виконує відповідну дію. У загальному вигляді така система працює як безконтактний пульт керування ближньої дії. Основні сценарії використання пристрою наведено в табл. 2.1.

Таблиця 2.1 – Можливі сценарії використання пристрою керування

Сценарій використання	**Приклад команди**	**Доцільність для APDS-9960**
Керування презентацією	Перехід до наступного або попереднього слайда	Висока, оскільки потрібні прості напрямлені жести
Керування мультимедіа	Пауза, перемикання треку, зміна гучності	Висока, команди короткі та повторювані
Локальне керування пристроєм	Зміна режиму роботи або підтвердження дії	Доцільно за умови чіткої карти команд
Навчальний або лабораторний стенд	Демонстрація роботи сенсора та мікроконтролера	Висока, оскільки добре показує апаратно-програмну взаємодію

Дані табл. 2.1 показують, що APDS-9960 найкраще підходить для сценаріїв, де не потрібно вводити складні дані або точно керувати курсором. Його сильна сторона – швидке розпізнавання простих рухів у ближній зоні. Тому функціональність пристрою не повинна бути надмірною. Надійність у цьому випадку важливіша за велику кількість підтримуваних команд.

Для подальшого проєктування доцільно прийняти базовий сценарій, у якому пристрій використовується як безконтактний контролер простих команд. Такий варіант дозволяє зосередитися на правильному підключенні сенсора, виборі мікроконтролера, розробці алгоритму обробки жестів і перевірці стабільності роботи. Якщо прототип стабільно виконує базові функції, його можна буде розширити додатковими режимами або змінити карту команд під конкретну сферу застосування.

2.2 Вибір мікроконтролера і способу узгодження з APDS-9960

Вибір мікроконтролера для пристрою керування на основі APDS-9960 залежить від кількох практичних вимог. Плата має підтримувати інтерфейс I2C, працювати з логічними рівнями 3,3 В або мати можливість безпечного узгодження з датчиком, забезпечувати достатню швидкодію для обробки жестів і мати засоби передавання команд до керованого пристрою. Для реалізації HID-керування також важливо, щоб мікроконтролер міг працювати як пристрій введення або підтримував відповідні бібліотеки [16; 18; 20].

APDS-9960 є низьковольтним датчиком, тому питання узгодження рівнів сигналів є одним із головних. Якщо підключити модуль до 5-вольтової логіки без

перетворювача рівнів, є ризик некоректної роботи або пошкодження сенсора. З цієї причини плати з логікою 3,3 В мають перевагу, оскільки дозволяють підключати APDS-9960 простіше та без додаткових елементів узгодження.

Для реалізації пристрою було розглянуто кілька можливих апаратних платформ. Їх порівняння подано в табл. 2.2.

Таблиця 2.2 – Порівняння мікроконтролерів для пристрою керування

Мікроконтролерна плата	**Переваги**	**Обмеження**	**Доцільність використання**
Arduino Uno	Простота використання, велика кількість прикладів і бібліотек	Логіка 5 В, відсутність нативного USB HID, потрібне узгодження рівнів	Можлива для тестів, але не найкраща для фінального пристрою
Arduino Pro Micro	Підтримка USB HID, компактність, зручна робота як пристрій введення	Частина плат працює з логікою 5 В, потрібно уважно перевіряти версію	Доцільна для USB-контролера
ESP32	Логіка 3,3 В, I2C, достатня продуктивність, підтримка Bluetooth	Потребує налаштування бібліотек для HID або BLE HID	Найбільш зручна для бездротового прототипу
STM32	Гнучке налаштування периферії, висока продуктивність, підтримка USB	Складніше середовище розробки та налаштування	Доцільна для більш складної версії пристрою

З табл. 2.2 видно, що для розроблюваного пристрою найбільш придатною є плата на основі ESP32. Вона підтримує I2C, працює з логічними рівнями 3,3 В і має достатні ресурси для обробки жестів. Наявність Bluetooth дозволяє реалізувати HID-передавання команд без використання окремого приймача або складної апаратної схеми.

Arduino Uno є зручною платою для початкового навчання, але в цьому випадку має кілька недоліків. Основний із них – логічний рівень 5 В, через що для APDS-9960 потрібно передбачати узгодження сигналів. Крім того, Arduino Uno не є зручним варіантом для HID-керування комп’ютером без додаткових рішень.

Arduino Pro Micro є хорошим варіантом для пристрою, який має працювати через USB як клавіатура або миша. Однак у цій роботі використовується ESP32-WROOM-32U, оскільки він дає можливість поєднати роботу з APDS-9960 через I2C

і передавання команд через Bluetooth HID. Це відповідає обраному сценарію безконтактного керування комп’ютером.

У цій роботі доцільно використати ESP32 як основний мікроконтролер. Він забезпечує сумісність за рівнями напруги з APDS-9960, має достатню кількість виводів і підтримує програмні бібліотеки для Arduino IDE. Крім того, ESP32 є поширеною платою, що полегшує повторення прототипу та пошук прикладів для налагодження.

Підключення APDS-9960 до ESP32 виконується через інтерфейс I2C. Лінія SDA датчика з’єднується з відповідним виводом GPIO21, а лінія SCL – з GPIO22. Такий варіант є типовим для ESP32 і підтримується бібліотекою Wire. Живлення сенсора подається від лінії 3,3 В, а земля датчика і мікроконтролера об’єднується.

Для коректної роботи важливо, щоб підтягувальні резистори ліній SDA та SCL були підключені до 3,3 В, а не до 5 В. У багатьох готових модулях APDS-9960 такі резистори вже встановлені на платі. Якщо модуль їх не має, підтягування потрібно додати окремо. У будь-якому випадку рівень сигналів на I2C має відповідати логіці 3,3 В [7; 14].

Окремим фактором є спосіб передавання команд до комп’ютера. Для звичайних плат ESP32 DevKit USB-роз’єм переважно використовується для живлення, прошивки та послідовного обміну. Тому для HID-керування доцільно використовувати Bluetooth HID. У такому випадку комп’ютер сприймає ESP32 як бездротову клавіатуру або мультимедійний контролер, а програма після розпізнавання жесту надсилає відповідну команду.

Узгодження APDS-9960 з ESP32 є простішим, ніж з 5-вольтовими платами, оскільки обидва компоненти працюють у низьковольтній логіці. Це зменшує кількість зовнішніх елементів і робить схему придатною для швидкого складання на макетній платі. Обраний варіант також залишає можливість подальшого вдосконалення: використання переривання INT, додавання індикатора стану або розроблення компактного корпусу.

2.3 Розробка структурної схеми пристрою

Структурна схема пристрою керування має відображати не окремі електричні з’єднання, а загальну логіку взаємодії між основними вузлами системи. Для пристрою на основі APDS-9960 така схема будується навколо трьох основних елементів: сенсорного модуля, мікроконтролера та керованого пристрою. Сенсор відповідає за фіксацію жесту, мікроконтролер – за обробку отриманих даних, а керований пристрій виконує команду, сформовану після розпізнавання руху руки.

У розроблюваній системі APDS-9960 використовується як вхідний сенсорний вузол. Коли користувач виконує жест перед датчиком, модуль фіксує зміну відбитого інфрачервоного сигналу та передає дані мікроконтролеру через інтерфейс I2C. Мікроконтролер ESP32 зчитує інформацію про жест, перевіряє її, виконує програмну фільтрацію та визначає, яку команду потрібно сформувати.

Структурну схему пристрою керування наведено на рис. 2.1.

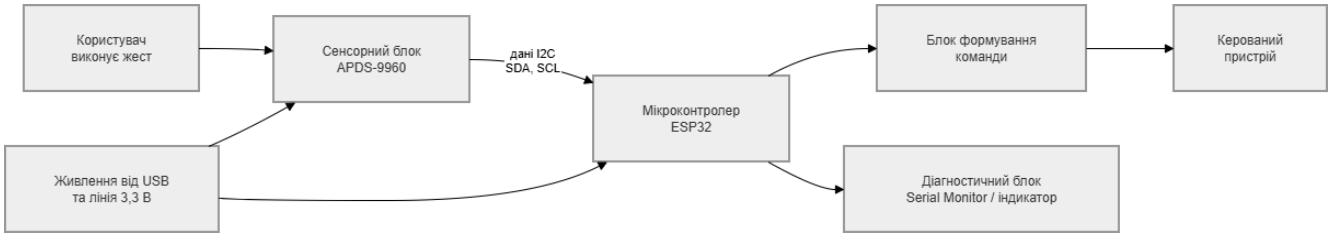

Рисунок 2.1 – Структурна схема пристрою керування на основі APDS-9960

На рис. 2.1 показано послідовність роботи пристрою від виконання жесту до формування команди. Користувач виконує рух рукою перед сенсорним блоком APDS-9960, після чого дані через лінії SDA та SCL передаються до мікроконтролера ESP32. Далі мікроконтролер обробляє отриманий результат і передає його до логічного блоку формування команди. У схемі цей блок не є окремим фізичним модулем, а позначає частину програмної логіки, яка визначає, яку дію має виконати керований пристрій.

Керований пристрій у цій структурі може бути різним залежно від обраного сценарію використання. Якщо система застосовується для презентацій, команда може відповідати переходу до наступного або попереднього слайда. Для мультимедійного сценарію це може бути пауза, відтворення, зміна гучності або

перемикання треку. У локальному вбудованому пристрої команда може використовуватися для перемикання режиму або активації окремої функції. Окремо на структурній схемі показано живлення від USB та лінію 3,3 В. Це означає, що пристрій може отримувати живлення від USB-порту, а необхідна для роботи мікроконтролера і датчика напруга формується на платі ESP32. Для APDS-9960 важливо використовувати саме допустимий рівень живлення та сигналів, оскільки датчик належить до низьковольтних компонентів. Спільна земля мікроконтролера і сенсора є обов’язковою умовою коректного обміну даними через I2C.

Діагностичний блок у структурі використовується переважно на етапі розробки та перевірки. Через Serial Monitor або простий індикатор можна контролювати, чи ініціалізувався датчик, який жест було розпізнано та яку команду сформувала програма. Це спрощує пошук помилок, оскільки дозволяє окремо перевірити роботу сенсора, обмін через I2C і програмну обробку жестів. Запропонована структура є модульною. Сенсорний блок відповідає за зчитування фізичної дії користувача, мікроконтролер – за обробку даних, а блок формування команди – за перетворення розпізнаного жесту на дію для зовнішньої системи. Такий поділ полегшує подальше проєктування електричної схеми, оскільки вже зрозуміло, які вузли потрібно з’єднати між собою та які сигнали між ними передаються.

На основі структурної схеми можна перейти до розробки схеми підключення пристрою. На наступному етапі потрібно уточнити конкретні виводи ESP32 для підключення APDS-9960, описати живлення модуля, лінії SDA та SCL, а також визначити, чи буде використовуватися вивід INT для обробки подій від датчика.

2.4 Розробка схеми підключення та опис елементної бази

Схема підключення пристрою керування розробляється на основі структурної схеми, наведеної в попередньому підрозділі. На цьому етапі потрібно перейти від загальної логіки роботи до конкретних з’єднань між мікроконтролером ESP32, датчиком APDS-9960 та лініями живлення. Основна увага приділяється

правильному підключенню інтерфейсу I2C, вибору напруги живлення та забезпеченню спільної землі між усіма компонентами пристрою.

У розроблюваному пристрої датчик APDS-9960 підключається до ESP32 через інтерфейс I2C. Для цього використовуються дві сигнальні лінії: SDA, яка відповідає за передавання даних, і SCL, яка задає тактовий сигнал. Для плати ESP32 типовим варіантом є використання GPIO21 як SDA та GPIO22 як SCL. Таке підключення є поширеним і добре підтримується бібліотеками для роботи з I2C [2].

Живлення APDS-9960 подається від лінії 3,3 В. Це важливо, оскільки датчик належить до низьковольтних компонентів і не повинен напряму працювати з 5-вольтовою логікою без узгодження рівнів. Плата ESP32 також працює з логікою 3,3 В, тому в такій конфігурації підключення є простішим і не потребує окремого перетворювача рівнів для ліній SDA та SCL [14].

У конкретному модулі APDS-9960 можуть бути окремі виводи VCC та VL. VCC використовується для живлення сенсорної частини модуля, а VL – для логічного рівня інтерфейсу. У прототипі обидва ці виводи доцільно підключати до 3,3 В, щоб сенсор і мікроконтролер працювали в одному електричному рівні. Вивід INT у базовій версії можна не підключати, оскільки програма виконує періодичне опитування датчика.

Схему підключення датчика APDS-9960 до ESP32 наведено на рис. 2.2.

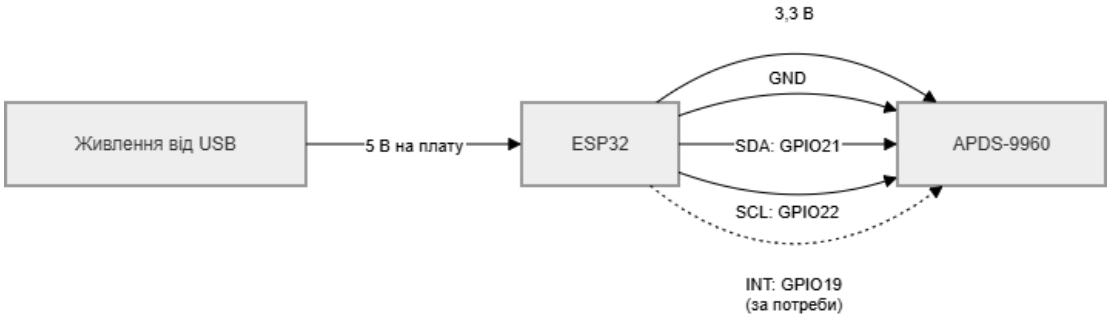

Рисунок 2.2 – Схема підключення APDS-9960 до мікроконтролера ESP32

На рис. 2.2 показано базове підключення сенсорного модуля до мікроконтролера. Вивід VCC датчика з’єднується з лінією 3,3 В плати ESP32, GND – із загальною землею, SDA – з GPIO21, а SCL – з GPIO22. Якщо використовується вивід INT, його можна підключити до одного з вільних цифрових входів ESP32,

наприклад GPIO19. У початковій версії пристрою роботу можна організувати без INT, через періодичне опитування датчика.

Під час розробки схеми потрібно враховувати наявність підтягувальних резисторів на лініях SDA та SCL. У багатьох готових модулях APDS-9960 вони вже встановлені на платі. Якщо таких резисторів немає, їх потрібно додати окремо, підключивши лінії SDA та SCL до 3,3 В через резистори номіналом приблизно 4,7 кОм. Підтягування саме до 3,3 В є принциповим, оскільки підключення до 5 В може бути небезпечним для сенсора.

Елементну базу пристрою подано в табл. 2.3.

Таблиця 2.3 – Основні елементи пристрою керування

Елемент	Призначення в пристрої	Особливості використання
ESP32	Основний мікроконтролер пристрою	Зчитує дані з APDS-9960, обробляє жести та формує команди
APDS-9960	Сенсор жестів і наближення	Працює через I2C, потребує живлення та логіки 3,3 В
USB-кабель	Живлення та програмування ESP32	Подає живлення на плату та використовується під час завантаження прошивки
Лінія 3,3 В	Живлення сенсорного модуля	Має відповідати допустимому діапазону APDS-9960
Лінії SDA і SCL	Передавання даних через I2C	Забезпечують обмін між ESP32 та APDS-9960
Лінія GND	Спільна земля схеми	Необхідна для коректної роботи інтерфейсу I2C
INT	Сигнал переривання від датчика	Може використовуватися для реакції на події без постійного опитування

Дані табл. 2.3 показують, що апаратна частина пристрою є відносно простою. Основними активними компонентами виступають мікроконтролер ESP32 та сенсор APDS-9960. Інші елементи забезпечують живлення, обмін даними та можливість налагодження. Така структура зручна для прототипування, оскільки кількість з’єднань невелика, а кожну лінію можна легко перевірити окремо.

Окремо потрібно врахувати особливості живлення через USB. Якщо ESP32 підключається до комп’ютера, на плату надходить напруга 5 В від USB-порту. Далі вбудований стабілізатор плати формує напругу 3,3 В, яка використовується мікроконтролером і може подаватися на сенсор APDS-9960. У такому випадку окремий зовнішній блок живлення не потрібен. Водночас не слід підключати VCC

або VL датчика до 5 В, якщо конкретний модуль не має власного стабілізатора та узгодження рівнів.

Під час складання макетного прототипу важливо не тільки правильно вибрати піни, а й забезпечити якісний фізичний контакт. На макетній платі один ряд контактів може не збігатися з іншим, а довгі або нещільно вставлені дроти призводять до нестабільного обміну через I2C. Тому перевірку підключення доцільно починати не з основної програми жестового керування, а з простого I2C-сканера, який показує, чи бачить ESP32 датчик на шині.

Запропонована схема підключення відповідає вимогам, сформованим у першому розділі. Вона забезпечує живлення датчика в допустимому діапазоні, використовує стандартний інтерфейс I2C для обміну даними та залишає можливість підключення лінії INT для подальшого вдосконалення логіки роботи. На основі цієї схеми можна переходити до розробки алгоритму зчитування, фільтрації та інтерпретації жестів.

2.5 Алгоритм зчитування, фільтрації та інтерпретації жестів

Алгоритм роботи пристрою має забезпечити не тільки отримання даних від APDS-9960, а й їхню перевірку перед формуванням команди. У простому варіанті мікроконтролер міг би одразу виконувати дію після кожного розпізнаного жесту. Проте на практиці такий підхід є ненадійним, оскільки один рух руки може бути сприйнятий кілька разів або випадковий рух біля сенсора може викликати небажану команду.

Робота алгоритму починається з ініціалізації мікроконтролера ESP32 та налаштування інтерфейсу I2C. Після запуску програма перевіряє наявність датчика APDS-9960 на шині, виконує його початкове налаштування та вмикає режим розпізнавання жестів. Якщо датчик не відповідає, пристрій не переходить до основного циклу обробки, а виводить діагностичне повідомлення через послідовний порт. Це спрощує пошук помилок під час складання прототипу.

Після успішної ініціалізації мікроконтролер переходить до основного циклу роботи. У цьому циклі він очікує появи жесту, зчитує дані з APDS-9960 та визначає

напрям руху руки. Для початкової реалізації доцільно використовувати періодичне опитування датчика. Такий підхід простіший у налагодженні, оскільки дозволяє контролювати кожен етап зчитування через Serial Monitor. У подальшому алгоритм можна вдосконалити за допомогою виводу INT, щоб мікроконтролер реагував на подію від сенсора без постійного опитування.

Узагальнену логіку зчитування та інтерпретації жесту подано на рис. 2.3.

Рисунок 2.3 – Узагальнена логіка зчитування та інтерпретації жесту

На рис. 2.3 показано не повну блок-схему програми, а скорочену послідовність проходження даних. Жест користувача спочатку фіксується сенсорним модулем, потім передається до ESP32, проходить етап фільтрації та перетворюється на HID-команду для комп’ютера. Такий рисунок зручний для основного тексту, оскільки не перевантажує розділ умовними переходами та повторними циклами. Деталізовану блок-схему роботи алгоритму, де показано перевірку доступності датчика, обробку помилок, повторні спроби та повернення до очікування, наведено в *Додатку А*.

Фільтрація є важливою частиною алгоритму. Якщо користувач проводить рукою перед датчиком повільно, APDS-9960 може кілька разів зафіксувати схожий рух. Без затримки між командами це призведе до дублювання дії. Наприклад, один жест вправо може викликати не один, а два переходи до наступного елемента. Щоб цього уникнути, після розпізнавання жесту вводиться короткий інтервал блокування повторної команди.

Тривалість такого інтервалу не повинна бути надто великою. Якщо затримка буде короткою, дублювання команд залишиться. Якщо занадто довгою – пристрій здаватиметься повільним і не зможе швидко реагувати на послідовні жести. Для прототипу доцільно передбачити експериментальний підбір цього параметра під час технічної перевірки.

Ще одним етапом фільтрації є перевірка коректності самого жесту. Програма має враховувати тільки ті значення, які відповідають передбаченим напрямам руху: вліво, вправо, вгору або вниз. Якщо бібліотека повертає невизначений результат або сигнал не відповідає жодній команді, така подія ігнорується. Це дозволяє зменшити кількість випадкових реакцій на слабкий або неповний рух руки.

Після проходження фільтрації жест передається до етапу інтерпретації. На цьому етапі програма не просто фіксує напрям руху, а визначає, яку дію потрібно виконати. Наприклад, жест вправо може означати перехід до наступного елемента, жест вліво – повернення назад, жест вгору – збільшення параметра, а жест вниз – його зменшення. Конкретна відповідність між жестами та діями задається в карті команд.

Для зручності налагодження кожен важливий етап алгоритму має супроводжуватися діагностичним повідомленням. У Serial Monitor можна виводити інформацію про успішну ініціалізацію датчика, розпізнаний жест, відхилену повторну команду або сформовану HID-дію. Це дає змогу швидко перевірити, чи правильно працює сенсор, чи не виникають помилки на шині I2C і чи коректно виконується фільтрація.

У програмній реалізації ця логіка розгортається у повніший алгоритм: запуск системи, перевірка APDS-9960, очікування жесту, зчитування результату, порівняння з допустимими значеннями, перевірка паузи після попередньої команди та передавання результату до функції HID-керування. Така послідовність дозволяє відокремити роботу з датчиком від прикладної логіки команд.

Запропонований алгоритм відповідає обмеженням APDS-9960. Він не намагається розпізнавати складні рухи руки, а працює з невеликим набором стабільних жестів. Основна увага приділяється не кількості команд, а передбачуваній реакції пристрою. Саме така логіка є найбільш придатною для компактного пристрою безконтактного керування на основі сенсора ближньої дії.

2.6 Формування карти команд керування

Карта команд керування визначає, яка дія має виконуватися після розпізнавання конкретного жесту. Для пристрою на основі APDS-9960 це один із ключових елементів програмної логіки, оскільки сам датчик лише фіксує напрям руху руки, а остаточне призначення цього руху задається вже в мікропрограмі. Завдяки цьому один і той самий апаратний пристрій можна адаптувати під різні сценарії використання без зміни схеми підключення.

Для базової версії пристрою доцільно використати чотири основні жести: рух вправо, вліво, вгору та вниз. Такий набір відповідає можливостям APDS-9960 і не перевантажує користувача зайвими командами. Кожен жест має бути пов’язаний з дією, яка логічно відповідає напряму руху. Наприклад, рух вправо природно сприймається як перехід до наступного елемента, а рух вліво – як повернення до попереднього.

Карту команд для розроблюваного пристрою наведено в табл. 2.4.

Таблиця 2.4 – Карта команд керування для пристрою на основі APDS-9960

Жест користувача	**Інтерпретація жесту**	**Команда керування**
Рух вправо	Перехід уперед	Наступний слайд, трек або режим
Рух вліво	Повернення назад	Попередній слайд, трек або режим
Рух вгору	Збільшення параметра	Збільшення гучності або значення налаштування
Рух вниз	Зменшення параметра	Зменшення гучності або значення налаштування
Наближення руки	Підтвердження або активація	Виконання вибраної дії або увімкнення режиму очікування жесту

Дані табл. 2.4 показують, що карта команд побудована за простою логікою відповідності між напрямом руху та дією. Це важливо для зручності користування, оскільки користувач швидше запам’ятовує команди, якщо вони не виглядають випадковими. У такому випадку пристрій не потребує складного навчання: достатньо один раз пояснити основні жести.

У програмній реалізації карта команд може бути задана у вигляді окремого блоку умов або структури відповідностей. Після розпізнавання жесту програма перевіряє його тип і викликає потрібну дію. Наприклад, якщо датчик повертає

значення, що відповідає руху вправо, мікроконтролер формує команду переходу до наступного елемента. Якщо жест не входить до передбаченого набору або був розпізнаний невпевнено, команда не виконується.

Окремо потрібно врахувати захист від повторного виконання однієї дії. Якщо користувач виконав жест вправо, система повинна сформувати лише одну команду переходу, а не декілька поспіль. Для цього після успішного розпізнавання жесту вводиться коротка затримка. Протягом цього часу нові однакові сигнали ігноруються. Такий механізм робить роботу пристрою більш передбачуваною.

Карта команд також має залишатися гнучкою. У різних сценаріях один і той самий жест може мати різне призначення. Наприклад, у режимі презентації рух вправо означає перехід до наступного слайда, а в мультимедійному режимі – перемикання на наступний трек. Через це доцільно не закладати команди безпосередньо в код обробки жестів, а виділити їх в окрему логічну частину програми. Це спростить подальше редагування функціональності.

У випадку HID-керування карта команд фактично виконує роль проміжного шару між сенсором і комп’ютером. APDS-9960 повертає напрям жесту, програма перетворює його на логічну дію, а HID-бібліотека передає відповідну клавішу або мультимедійну команду. Завдяки цьому комп’ютер не повинен знати, яким саме способом було отримано команду: для операційної системи вона виглядає як звичайне натискання клавіші або дія стандартного пристрою введення.

Для початкового прототипу достатньо реалізувати одну базову карту команд. Вона дозволить перевірити, чи правильно працює весь ланцюг: жест користувача, зчитування APDS-9960, обробка мікроконтролером, фільтрація та виконання HID-дії. Після перевірки стабільності роботи можна буде розширити карту команд або додати кілька режимів використання пристрою.

Сформована карта команд завершує проєктування логіки роботи пристрою на рівні другого розділу. Вона пов’язує апаратну частину з практичним сценарієм використання, оскільки визначає, яку саме дію виконує система після розпізнавання жесту. Надалі на основі цієї карти можна переходити до реалізації мікропрограмного забезпечення та технічної перевірки роботи пристрою.

Висновки до розділу 2

У другому розділі було виконано проєктування пристрою керування на основі датчика APDS-9960. Спочатку визначено сценарій використання пристрою як компактного засобу ближнього безконтактного керування. Такий сценарій передбачає розпізнавання простих напрямлених жестів руки та їх подальше перетворення на керуючі команди.

Для апаратної реалізації обґрунтовано використання мікроконтролера ESP32. Його вибір пов’язаний із підтримкою інтерфейсу I2C, роботою з логікою 3,3 В та достатніми ресурсами для обробки даних від сенсора. Також було враховано питання узгодження рівнів сигналів між ESP32 і APDS-9960.

Розроблено структурну схему пристрою, у якій показано взаємодію користувача, сенсорного модуля APDS-9960, мікроконтролера ESP32, блоку формування команди та керованого пристрою. Описано електричну схему підключення APDS-9960 до ESP32.

Окремо розроблено алгоритм зчитування, фільтрації та інтерпретації жестів. У ньому передбачено ініціалізацію датчика, очікування жесту, перевірку отриманого результату і передавання жесту до карти команд. Такий підхід дає змогу зробити роботу пристрою більш стабільною та зменшити кількість випадкових реакцій. Сформована карта команд пов’язує базові жести з конкретними діями керування, і відповідає можливостям APDS-9960. Результати проєктування створюють основу для переходу до реалізації пристрою. У наступному розділі доцільно описати налаштування APDS-9960, написання мікропрограмного забезпечення, організацію обміну з керованим пристроєм та експериментальну перевірку точності розпізнавання жестів.

3 РЕАЛІЗАЦІЯ ТА ТЕХНІЧНА ПЕРЕВІРКА ПРИСТРОЮ

3.1 Складання макетного прототипу пристрою

Макетний прототип було зібрано на основі плати ESP32-WROOM-32U та модуля APDS-9960. З’єднання виконувалися на макетній платі за допомогою дротів Dupont. Такий спосіб складання обрано не випадково: на етапі налагодження потрібно мати доступ до кожної лінії, швидко перевіряти живлення, міняти місцями SDA та SCL у разі помилки, а також контролювати, чи не перекривається оптична зона сенсора.

APDS-9960 має окремі виводи VCC та VL, тому обидва контакти були підключені до лінії 3,3 В. Це важливо для використаного модуля, оскільки датчик і логічні рівні ESP32 працюють у низьковольтному режимі. Вивід INT у базовій реалізації залишено непідключеним. Для першої версії пристрою достатньо періодичного опитування сенсора у головному циклі програми, а використання переривання можна розглядати як подальше вдосконалення.

Зовнішній вигляд макетного прототипу, зібраного на макетній платі, наведено на рис. 3.1.

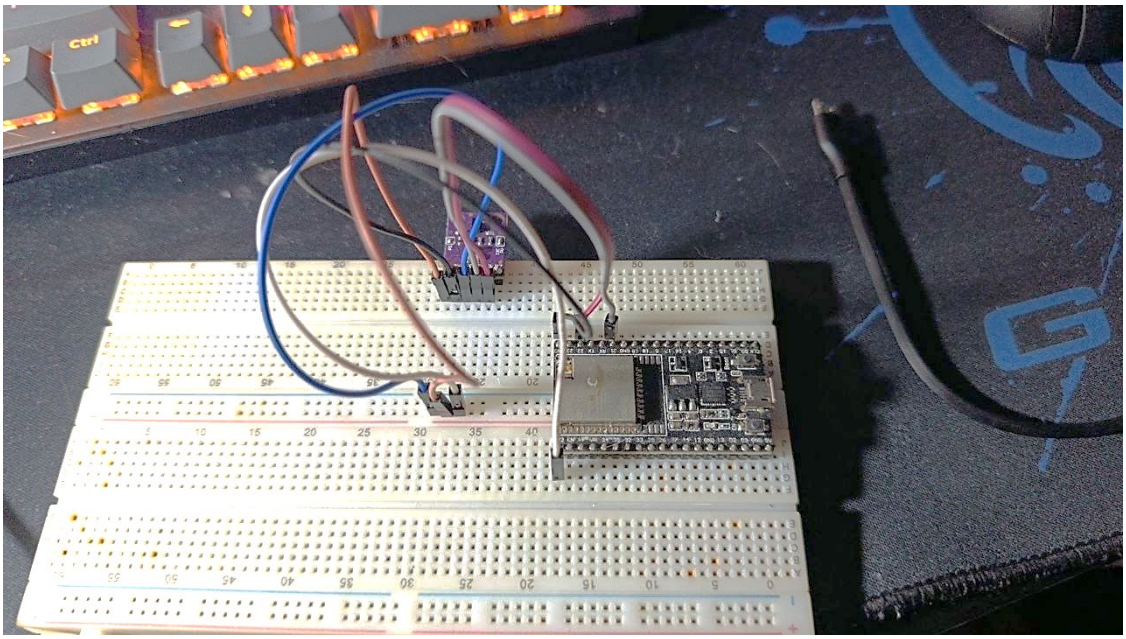

Рисунок 3.1 – Макетний прототип пристрою керування на основі ESP32 та APDS-9960

На рис. 3.1 показано робочий макет пристрою. Сенсорний модуль розміщено окремо від плати ESP32, щоб перед оптичним вікном залишалася вільна зона для руху руки. Для датчика ближньої дії це має практичне значення: надто близьке розташування дротів або інших елементів біля сенсора може погіршувати стабільність зчитування жестів.

Фактичне підключення виводів датчика APDS-9960 до плати ESP32 наведено в табл. 3.1.

Таблиця 3.1 – Підключення датчика APDS-9960 до ESP32

Вивід APDS-9960	Вивід ESP32	Призначення
VL	3V3	Живлення логічної частини модуля
VCC	3V3	Живлення сенсорного модуля
GND	GND	Спільна земля пристрою
SDA	GPIO21	Лінія передавання даних I2C
SCL	GPIO22	Лінія тактування I2C
INT	не використовується	Сигнал переривання, не задіяний у базовій версії

Після складання макета було виконано перевірку I2C-шини. Для цього використано сканер адрес, який дозволяє переконатися, що ESP32 бачить підключений сенсор. Для APDS-9960 типовою адресою є 0x39. Така перевірка корисна перед запуском основної програми, оскільки вона відокремлює помилки апаратного підключення від помилок у програмній логіці.

3.2 Налаштування Arduino IDE та підготовка програмних засобів

Програмна частина пристрою була підготовлена у середовищі Arduino IDE. Для роботи з ESP32 у менеджері плат було встановлено пакет ESP32 by Espressif Systems, після чого в налаштуваннях обрано плату ESP32 Dev Module. Така конфігурація відповідає використаній платі на базі ESP32-WROOM-32U і дозволяє завантажувати прошивку через USB-порт.

Для обміну з APDS-9960 застосовано бібліотеку Wire, яка відповідає за роботу I2C. У програмі явно задано стандартні для ESP32 виводи: GPIO21 як SDA та GPIO22 як SCL. Для роботи з жестами використано бібліотеку SparkFun APDS9960, а для передавання команд комп’ютеру – бібліотеку BleKeyboard, яка

дозволяє ESP32 працювати як Bluetooth HID-пристрій. Serial Monitor у цій схемі залишено для контролю запуску, діагностики та перевірки сформованих команд. Описані програмні засоби відповідають обраному середовищу реалізації та бібліотекам, наведеним у документації Arduino та SparkFun [2; 3; 7; 17].

Перелік основних програмних засобів, використаних під час реалізації пристрою, подано в табл. 3.2.

Таблиця 3.2 – Програмні засоби, використані під час реалізації

Засіб	**Роль у реалізації пристрою**
Arduino IDE	Написання, компіляція та завантаження мікропрограми на ESP32
ESP32 by Espressif Systems	Підтримка плат ESP32 у середовищі Arduino IDE
Wire	Обмін даними між ESP32 та APDS-9960 через I2C
SparkFun APDS9960	Ініціалізація датчика та отримання розпізнаного жесту
BleKeyboard	Формування Bluetooth HID-команд для комп’ютера
Serial Monitor	Діагностика запуску, жестів і сформованих команд

Повний лістинг мікропрограмного забезпечення наведено в *Додатку Б*. В основному тексті подано лише пояснення ключових частин програми та скріншоти фрагментів Arduino IDE, оскільки надмірне розміщення коду в розділі ускладнює читання практичної частини.

Фрагмент ініціалізації датчика та запуску Bluetooth HID у середовищі Arduino IDE наведено на рис. 3.2.

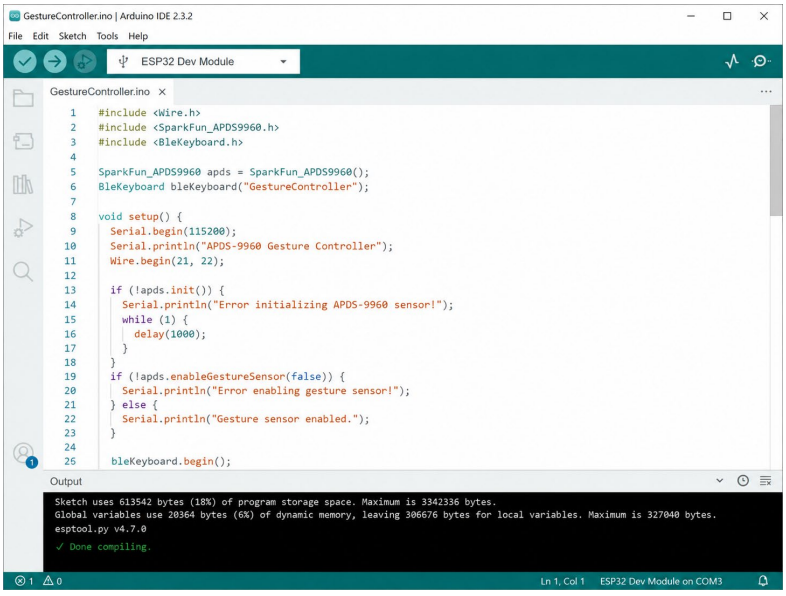

Рисунок 3.2 – Фрагмент ініціалізації APDS-9960 та Bluetooth HID в Arduino IDE

На рис. 3.2 наведено початкову частину програми, де підключаються бібліотеки, створюються об’єкти для роботи з APDS-9960 та HID-передаванням, запускається Serial Monitor і налаштовується I2C. На цьому етапі програма також перевіряє, чи вдалося ініціалізувати датчик. Якщо сенсор не відповідає, виконання основної логіки не продовжується, що спрощує пошук помилок у підключенні.

3.3 Реалізація зчитування та первинної обробки жестів

Після успішної ініціалізації пристрій переходить у режим очікування жесту. У головному циклі програма перевіряє, чи є нова жестова подія, після чого зчитує результат із APDS-9960. Бібліотека повертає напрям руху, а програма порівнює його з допустимим набором значень: RIGHT, LEFT, UP, DOWN та NEAR.

Не кожне спрацьовування сенсора одразу має перетворюватися на команду. У реальних умовах один рух руки іноді може зчитуватися кілька разів, особливо якщо жест виконується повільно або рука затримується перед датчиком. Тому в програмі використано коротку затримку після формування команди. Це не робить систему складною, але зменшує кількість повторних спрацьовувань.

Фрагмент обробки жестових команд у головному циклі програми наведено на рис. 3.3.

GestureController.ino ×

```
#include <Wire.h>
#include "Adafruit_APDS9960.h"

Adafruit_APDS9960 apds = Adafruit_APDS9960();

void handleGesture(const char* direction);

void setup() {
  Serial.begin(115200);
  while (!Serial) { delay(10); }

  Wire.begin();

  if (!apds.begin()) {
    Serial.println("APDS-9960 not found. Check wiring!");
    while (1) { delay(100); }
  }

  apds.enableGesture(true);
  apds.setGestureGain(GGAIN_4X);
  apds.setGestureLEDDrive(GLED_100MA);
  Serial.println("APDS-9960 Gesture Sensor Ready.");
}

void loop() {
  // Check if a new gesture is available
  if (apds.isGestureAvailable()) {
    int gesture = apds.readGesture();

    switch (gesture) {
      case DIR_RIGHT:
        handleGesture("RIGHT");
        break;
      case DIR_LEFT:
        handleGesture("LEFT");
        break;
      case DIR_UP:
        handleGesture("UP");
        break;
      case DIR_DOWN:
        handleGesture("DOWN");
        break;
      case DIR_NEAR:
        handleGesture("NEAR");
        break;
      default:
        // Ignore other values
        break;
    }
```

Рисунок 3.3 – Фрагмент обробки жестових команд у програмі

На рис. 3.3 показано фрагмент основного циклу, де програма перевіряє доступність жесту та визначає його напрям. У цій частині ще не виконується сама дія керування. Вона лише передає розпізнаний напрям до окремої функції, яка відповідає за вибір HID-команди. Такий поділ зручний, оскільки логіка зчитування сенсора не змішується з логікою керування комп’ютером.

3.4 Реалізація HID-передавання команд

Головною відмінністю реалізованого пристрою від простого демонстраційного прикладу є використання HID-передавання. Після розпізнавання жесту ESP32 формує команду, яку комп’ютер сприймає як дію стандартного пристрою введення. Для користувача це означає, що не потрібно створювати окрему програму-приймач на ПК: достатньо підключити Bluetooth HID-пристрій, після чого жести можуть виконувати функції клавіш або мультимедійних команд. [5; 7; 9].

У межах прототипу використано логіку Bluetooth HID-клавіатури. Жест RIGHT відповідає переходу вперед, LEFT – поверненню назад, UP та DOWN можуть застосовуватися для зміни гучності або параметра, а NEAR – для підтвердження дії. Такий набір не перевантажує користувача і відповідає можливостям APDS-9960.

Фрагмент формування HID-команд після розпізнавання жестів наведено на рис. 3.4.

```
3
4   void setup() {
5     Serial.begin(115200);
6     while (!Serial) {
7       delay(10);
8     }
9     Serial.println("Gesture Controller starting...");
10    bleKeyboard.begin();
11    Serial.println("BLE HID Keyboard started");
12  }
13
14  void loop() {
15    // Main loop can handle gesture detection and call handleGesture(dir)
16  }
17
18  // Handle recognized gestures and send HID keyboard commands
19  void handleGesture(const char* dir) {
20    Serial.print("Gesture: ");
21    Serial.println(dir);
22
23    if (strcmp(dir, "RIGHT") == 0) {
24      bleKeyboard.write(KEY_RIGHT_ARROW);
25      Serial.println("HID command: NEXT");
26    } else if (strcmp(dir, "LEFT") == 0) {
27      bleKeyboard.write(KEY_LEFT_ARROW);
28      Serial.println("HID command: PREVIOUS");
29    } else if (strcmp(dir, "UP") == 0) {
30      bleKeyboard.write(KEY_MEDIA_VOLUME_UP);
31      Serial.println("HID command: VOLUME_UP");
32    } else if (strcmp(dir, "DOWN") == 0) {
33      bleKeyboard.write(KEY_MEDIA_VOLUME_DOWN);
34      Serial.println("HID command: VOLUME_DOWN");
35    } else if (strcmp(dir, "NEAR") == 0) {
36      bleKeyboard.write(KEY_RETURN);
37      Serial.println("HID command: SELECT");
38    } else {
39      Serial.println("Unknown gesture");
40    }
41
42    delay(700); // Prevent repeated triggers
43  }
```

Рисунок 3.4 – Формування HID-команд після розпізнавання жестів

На рис. 3.4 показано фрагмент програми, де розпізнаний жест перетворюється на HID-команду. Наприклад, для жесту RIGHT формується команда переходу до наступного елемента, а для LEFT – до попереднього. Діагностичний рядок у Serial Monitor дублює цю дію, але не є основним способом передавання. Основна команда надходить до комп’ютера саме через HID-логіку.

Відповідність між жестами користувача та HID-командами наведено в табл. 3.3.

Таблиця 3.3 – Відповідність жестів HID-командам

Жест	HID-команда	Практичне призначення
RIGHT	KEY_RIGHT_ARROW / NEXT	Перехід до наступного слайда або елемента
LEFT	KEY_LEFT_ARROW / PREVIOUS	Повернення до попереднього слайда або елемента
UP	KEY_MEDIA_VOLUME_UP	Збільшення гучності або параметра
DOWN	KEY_MEDIA_VOLUME_DOWN	Зменшення гучності або параметра
NEAR	KEY_RETURN / SELECT	Підтвердження дії або вибір

Карта команд залишена простою, оскільки для сенсора ближньої дії важливіша стабільність базових рухів, ніж велика кількість функцій. За потреби призначення жестів можна змінити без перероблення апаратної частини: достатньо змінити відповідність у функції формування HID-команд.

Під час реалізації HID-передавання важливо було відокремити розпізнавання жесту від виконання команди. У програмі спочатку визначається напрям руху руки, після чого окрема функція вже приймає рішення, яку клавішу або мультимедійну дію потрібно передати комп’ютеру. Такий поділ робить код зрозумілішим і спрощує подальші зміни. Наприклад, якщо пристрій потрібно буде використовувати не для презентацій, а для керування медіаплеєром, достатньо змінити карту команд, не переписуючи логіку роботи з APDS-9960.

Окрему роль має перевірка підключення Bluetooth HID. Якщо комп’ютер ще не встановив з’єднання з ESP32, команда не повинна вважатися виконаною. У такому випадку в діагностичний порт виводиться службове повідомлення. Це дало можливість під час налагодження відрізняти дві різні ситуації: коли жест не

розпізнано сенсором і коли жест розпізнано, але HID-команда не була передана через відсутність з’єднання.

3.5 Налагодження та діагностичне виведення

Під час налагодження пристрою Serial Monitor використовувався як допоміжний інструмент. Через нього виводилися повідомлення про запуск контролера, ініціалізацію APDS-9960, запуск Bluetooth HID-пристрою, розпізнаний жест і сформовану команду. Це дозволяє швидко зрозуміти, на якому етапі виникає проблема: у підключенні сенсора, у зчитуванні жесту або у формуванні HID-команди.

Приклад діагностичного виведення розпізнаних жестів і сформованих HID-команд наведено на рис. 3.5.

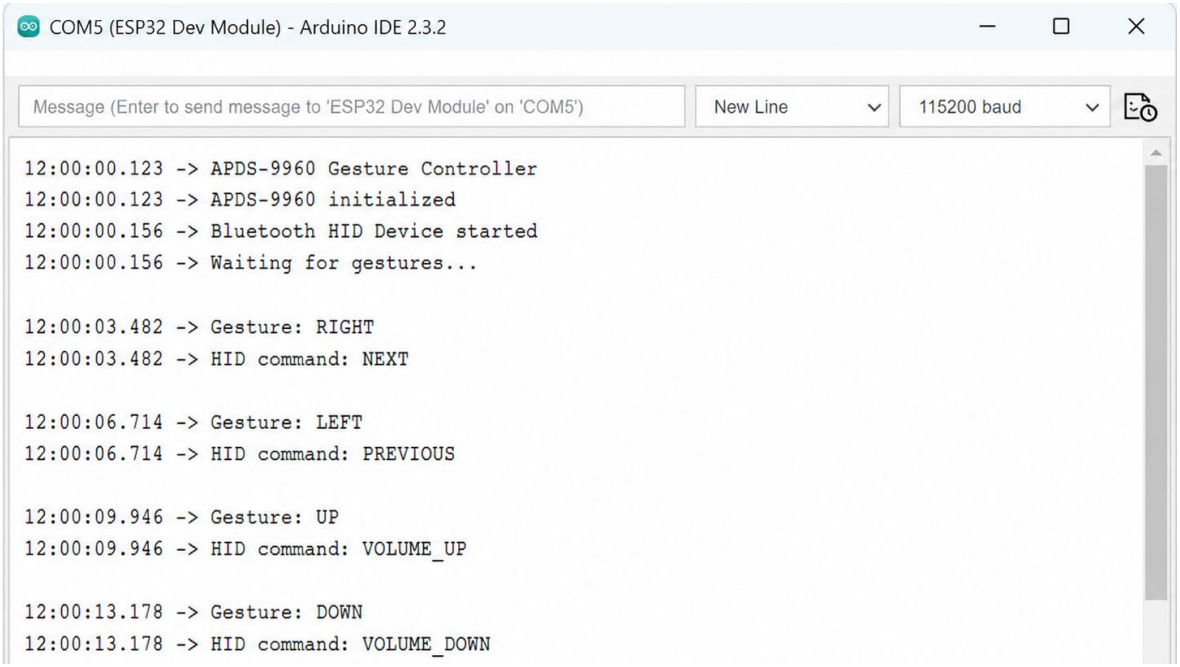

Рисунок 3.5 – Діагностичне виведення розпізнаних жестів та HID-команд

На рис. 3.5 видно приклад діагностичного виведення. Після запуску програма повідомляє про готовність APDS-9960 та Bluetooth HID. Далі для кожного жесту виводиться його назва і відповідна команда. Такий формат зручний для тестування, оскільки дозволяє порівняти фактичний рух руки з тим, яку команду сформувала програма.

3.6 Методика технічної перевірки та результати

Перевірка пристрою проводилася для оцінювання стабільності розпізнавання базових жестів і правильності формування HID-команд. Макетний прототип розміщувався на робочому столі, а рука користувача виконувала рухи перед датчиком на відстані приблизно 5–10 см. Такий діапазон обрано як практично зручний для APDS-9960: рука знаходиться достатньо близько до сенсора, але не перекриває його механічно.

Кожен із чотирьох основних жестів було виконано серією з 20 повторів. Правильним вважалося спрацьовування, за якого програма визначала потрібний напрям і формувала відповідну HID-команду. Якщо жест не розпізнавався або команда не відповідала руху руки, результат зараховувався як помилка.

Умови проведення технічної перевірки пристрою наведено в табл. 3.4.

Таблиця 3.4 – Умови технічної перевірки пристрою

Параметр	Значення
Мікроконтролер	ESP32-WROOM-32U
Датчик	APDS-9960
Інтерфейс датчика	I2C
Канал передавання команд	Bluetooth HID
Середовище розробки	Arduino IDE
Перевірені жести	LEFT, RIGHT, UP, DOWN
Кількість повторів кожного жесту	20
Орієнтовна робоча відстань	5–10 см

Кількісні результати перевірки розпізнавання жестів подано в табл. 3.5.

Таблиця 3.5 – Результати перевірки розпізнавання жестів

Жест	Кількість спроб	Правильні спрацьовування	Помилки	Точність, %
LEFT	20	18	2	90
RIGHT	20	19	1	95
UP	20	17	3	85
DOWN	20	18	2	90

За отриманими результатами найстабільніше розпізнавався жест RIGHT. Для нього було зафіксовано 19 правильних спрацьовувань із 20. Жести LEFT та DOWN

також показали достатньо близькі результати. Найменша точність отримана для жесту UP, що може бути пов’язано з траєкторією руху руки та положенням сенсора на макетній платі.

Середня точність для чотирьох основних жестів становила близько 90 %. Для макетного прототипу цей результат можна вважати прийнятним, оскільки пристрій орієнтований на прості команди ближнього керування, а не на складне розпізнавання положення руки. Водночас результати показують, що на стабільність впливають швидкість жесту, відстань до датчика та повторне проходження руки через робочу зону.

Для додаткової перевірки також оцінювалася стабільність роботи пристрою на різній відстані від сенсора. Оскільки APDS-9960 є сенсором ближньої дії, занадто велика відстань між рукою та модулем призводить до послаблення відбитого інфрачервоного сигналу. Найкраще пристрій реагував тоді, коли рух виконувався плавно, без різкого прискорення руки та без перекриття датчика сторонніми предметами.

Під час тестування було помітно, що положення модуля на макетній платі також впливає на результат. Якщо датчик розташований під кутом або частково закритий дротами, частина жестів визначається гірше. Через це в остаточній конструкції доцільно передбачити фіксоване розміщення APDS-9960 та відкриту зону перед його оптичним вікном. Для макетного прототипу це обмеження є очікуваним, але його потрібно враховувати під час інтерпретації результатів.

3.7 Обмеження реалізації та напрями вдосконалення

Реалізований прототип підтверджує працездатність запропонованої логіки, але має обмеження, характерні для макетної збірки. З’єднання на макетній платі менш надійні, ніж пайка або друкована плата, тому під час тривалої роботи можливі нестабільні контакти. Також датчик APDS-9960 чутливий до положення руки та умов освітлення, що впливає на повторюваність результатів.

Для практичного використання пристрій доцільно перенести на компактну плату або хоча б зафіксувати модуль APDS-9960 у стабільному положенні. Окрему

увагу потрібно приділити корпусу: перед сенсором має залишатися відкрите вікно, через яке проходить інфрачервоне випромінювання. Якщо датчик буде розміщено надто глибоко або перекрито непрозорим матеріалом, якість розпізнавання знизиться.

Програмну частину також можна розширити. Наприклад, замість сталої затримки після жесту можна використовувати більш гнучку фільтрацію, яка враховує тип попередньої команди та час між спрацьовуваннями. Крім того, можна додати перемикання режимів роботи: презентаційний режим, мультимедійний режим або режим керування окремим застосунком.

Ще одним напрямом удосконалення є створення кількох режимів роботи. Наприклад, один режим може використовуватися для презентацій, де основними діями є перехід уперед і назад, а інший – для мультимедіа, де важливішими стають зміна гучності та керування відтворенням. У такому разі APDS-9960 залишається тим самим сенсорним елементом, але змінюється програмна інтерпретація жестів.

З погляду апаратної частини корисним було б перейти від макетної плати до компактнішого монтажу. Дроти Dupont зручні для налагодження, але вони не забезпечують достатньої механічної стабільності. У практичному пристрої варто використовувати коротші з’єднання, фіксацію датчика на передній панелі та окреме вікно для проходження інфрачервоного випромінювання.

3.8 Оцінка відповідності реалізації сформованим вимогам

Оцінку відповідності реалізованого прототипу основним вимогам наведено в табл. 3.6.

Таблиця 3.6 – Відповідність реалізованого прототипу основним вимогам

Вимога	Реалізація в прототипі	Результат
Робота з APDS-9960 через I2C	Сенсор підключено до ESP32 за лініями SDA GPIO21 та SCL GPIO22	Виконано
Живлення сенсора 3,3 В	Виводи VCC та VL підключено до лінії 3,3 В, GND об’єднано з ESP32	Виконано

Вимога	Реалізація в прототипі	Результат
Розпізнавання базових жестів	Обробляються жести LEFT, RIGHT, UP, DOWN та NEAR	Виконано частково з урахуванням стабільності
Фільтрація повторних спрацьовувань	Після виконання команди використовується затримка між жестами	Виконано
HID-передавання команд	ESP32 формує Bluetooth HID-команди для комп’ютера	Виконано
Діагностика роботи	Serial Monitor використовується для контролю запуску, жестів і команд	Виконано

Після складання та перевірки прототипу доцільно зіставити отриману реалізацію з вимогами, сформованими в першому розділі. Така перевірка не дублює результати тестування, а показує, наскільки макетний пристрій відповідає початковій логіці розробки. Для пристрою керування важливими є не лише правильні електричні з’єднання, а й повний цикл роботи: зчитування жесту, його інтерпретація, фільтрація та передавання команди до комп’ютера через HID-рівень.

Дані табл. 3.6 показують, що основні вимоги до макетного прототипу виконано. Найбільш вразливою частиною залишається стабільність розпізнавання жестів, оскільки вона залежить від відстані до руки, швидкості руху та положення сенсора. Разом із тим логіка пристрою працює як єдиний ланцюг: фізичний жест перетворюється на цифрові дані, проходить програмну обробку і завершується HID-командою для комп’ютера.

Для КБР такий результат є достатнім, оскільки було перевірено не окремий датчик, а повну апаратно-програмну взаємодію. Подальше вдосконалення має стосуватися не зміни самої ідеї пристрою, а покращення конструкції, стабільності живлення, фіксації модуля APDS-9960 та точнішого налаштування фільтрації.

Висновки до розділу 3

У третьому розділі було описано реалізацію макетного прототипу пристрою керування на основі ESP32-WROOM-32U та датчика APDS-9960. Підключення сенсора виконано через інтерфейс I2C з використанням ліній SDA та SCL, а живлення модуля організовано від лінії 3,3 В.

Програмну частину реалізовано в Arduino IDE з використанням бібліотек Wire, SparkFun APDS9960 та BleKeyboard. Повний лістинг програми винесено в додаток Б, а в основному тексті розглянуто логіку запуску пристрою, зчитування жестів, фільтрації повторних спрацьовувань і формування HID-команд.

Під час технічної перевірки встановлено, що пристрій здатний розпізнавати базові жести LEFT, RIGHT, UP та DOWN і перетворювати їх на керуючі дії для комп’ютера через Bluetooth HID. Serial Monitor використовувався для діагностики, але не як основний канал керування.

Середня точність розпізнавання жестів у проведеній серії перевірок становила близько 90 %. Це підтверджує придатність APDS-9960 для побудови простого безконтактного контролера ближньої дії. Подальше вдосконалення пристрою може бути пов’язане з покращенням конструкції, стабілізацією положення датчика та розширенням програмної фільтрації жестів.

ВИСНОВКИ

У КБР розроблено та перевірено прототип пристрою безконтактного керування на основі датчика APDS-9960. Отриманий результат відповідає поставленій у вступі меті: пристрій зчитує прості жести користувача, виконує їхню програмну інтерпретацію та формує HID-команди, які можуть сприйматися комп’ютером як дії стандартного пристрою введення.

Першу задачу було вирішено через аналіз безконтактного керування та засобів розпізнавання жестів. Порівняння показало, що APDS-9960 не є універсальним сенсором для складного просторового аналізу руки, однак має достатні можливості для ближнього керування короткими командами. На відміну від ToF-сенсорів, він одразу орієнтований на визначення напрямлених жестів, а порівняно з більш функціональним PAJ7620U2 має простішу логіку підключення і достатній набір команд для обраного сценарію.

У процесі вирішення другої та третьої задач було досліджено технічні можливості APDS-9960, його роботу через I2C, особливості живлення від 3,3 В та обмеження робочої зони. На основі цього сформовано вимоги до пристрою: розпізнавання жестів LEFT, RIGHT, UP і DOWN, зменшення повторних спрацьовувань, передавання команд через HID та можливість діагностики під час налагодження.

Проєктна частина роботи охопила вибір мікроконтролера ESP32-WROOM-32U, побудову структурної схеми, схеми підключення, алгоритму зчитування жестів і карти команд. Деталізовану блок-схему алгоритму винесено в додаток А, що дозволило не перевантажувати основний текст великою схемою, але зберегти повний опис логіки роботи пристрою.

Практична реалізація підтвердила працездатність запропонованого рішення. Програмну частину створено в Arduino IDE з використанням бібліотек Wire, SparkFun APDS9960 та BleKeyboard. У межах прототипу Serial Monitor застосовувався як допоміжний засіб діагностики, а основним результатом роботи програми було формування Bluetooth HID-команд.

Під час технічної перевірки отримано середню точність розпізнавання базових жестів близько 90 %. Найстабільніше розпізнавався жест RIGHT, а найбільше похибок спостерігалося для жесту UP. Такий результат пояснюється особливостями траєкторії руху руки, відстані до датчика та положенням модуля на макетній платі. Порівняно з аналогами розроблений прототип поступається за кількістю можливих жестів, але має перевагу в простоті реалізації, компактності та відсутності потреби в камері або складній обробці зображень.

Ступінь новизни отриманого результату полягає не у створенні нового сенсора або нового алгоритму розпізнавання, а в практичному поєднанні доступного оптичного датчика APDS-9960, мікроконтролера ESP32 та HID-передавання команд у вигляді компактного навчально-прикладного прототипу. Практичне значення роботи полягає в можливості використання такого пристрою для керування презентаціями, мультимедійними функціями, навчальними стендами або простими локальними режимами обладнання без фізичного контакту з кнопками чи клавіатурою.

Для впровадження результатів у реальні умови експлуатації рекомендується перейти від макетної плати до більш надійного монтажу, зафіксувати датчик у стабільному положенні, передбачити отвір або прозоре вікно перед сенсором і додатково налаштувати фільтрацію повторних спрацьовувань. Подальший розвиток об’єкта розробки може бути пов’язаний із додаванням кількох режимів роботи, розширенням карти HID-команд, використанням виводу INT та створенням компактного корпусу для повсякденного використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Салтовський Б. Г., Мельников А. Г. Пристрій керування на основі датчика APDS-9960. *Могилянські читання – 2025* : тези доп. XXVIII Всеукр. наук.-практ. конф., Миколаїв, 10–14 листоп. 2025 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2025. С. 75–78.

2. Ionescu M., Borcosi I. The Use of the Gestures Sensor within a Biomimetic Structure. *Annals of the “Constantin Brancusi” University of Targu Jiu, Engineering Series*. 2018. № 4. P. 107–111. URL: https://www.utgjiu.ro/rev_ing/pdf/2018-4/17_M.%20Ionescu,%20I.%20Borcosi%20-THE%20USE%20OF%20THE%20GESTURES%20SENSOR%20WITHIN%20A%20BIOMIMETIC%20STRUCTURE.pdf (Last accessed: 06.05.2026).

3. Liu Y., Wang C., Hou M., Wang X. Design of Non-contact Gesture Recognition Control System Based on Embedded System. *Proceedings of SPIE*. 2024. Vol. 12981. DOI: 10.1117/12.3014928.

4. Saffer D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. Sebastopol : O’Reilly Media, 2008. 264 p.

5. Koptyra K., Ogiela M. R. Steganography in IoT: Information Hiding with APDS-9960 Proximity and Gestures Sensor. *Sensors*. 2022. Vol. 22. № 7:2612. P. 1–18. DOI: 10.3390/s22072612.

6. Wang R.-J., Lai S.-C., Jhuang J.-Y., Ho M.-C., Shiau Y.-C. Development of Smart Home Gesture-based Control System. *Sensors and Materials*. 2021. Vol. 33. № 10. P. 3459–3471. DOI: 10.18494/SAM.2021.3522.

7. APDS-9960 Digital Proximity, Ambient Light, RGB and Gesture Sensor : datasheet. Broadcom / Avago Technologies. URL: https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf (Last accessed: 06.05.2026).

8. PAJ7620U2 Integrated Gesture Recognition Sensor : datasheet. PixArt Imaging Inc. URL: https://files.seeedstudio.com/wiki/Grove_Gesture_V_1.0/res/PAJ7620U2_Datasheet_V0.8_20140611.pdf (Last accessed: 06.05.2026).

9. Rautaray S. S., Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*. 2015. Vol. 43. P. 1–54. DOI: 10.1007/s10462-012-9356-9.

10. ST VL53L0X Time-of-Flight Ranging Sensor : datasheet. STMicroelectronics. URL: https://www.st.com/resource/en/datasheet/vl53l0x.pdf (Last accessed: 06.05.2026).

11. Wobbrock J. O., Wilson A. D., Li Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. 2007. P. 159–168. DOI: 10.1145/1294211.1294238.

12. Wobbrock J. O., Morris M. R., Wilson A. D. User-defined gestures for surface computing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009. P. 1083–1092. DOI: 10.1145/1518701.1518866.

13. SparkFun APDS-9960 RGB and Gesture Sensor Hookup Guide. SparkFun Electronics. URL: https://learn.sparkfun.com/tutorials/apds-9960-rgb-and-gesture-sensor-hookup-guide/all (Last accessed: 06.05.2026).

14. Arduino Wire Library. Arduino Documentation. URL: https://www.arduino.cc/reference/en/language/functions/communication/wire/ (Last accessed: 06.05.2026).

15. Arduino_APDS9960 Library. Arduino Documentation. URL: https://docs.arduino.cc/libraries/arduino_apds9960/ (Last accessed: 06.05.2026).

16. Arduino IDE Documentation. Arduino Documentation. URL: https://docs.arduino.cc/software/ide/ (Last accessed: 06.05.2026).

17. PlatformIO Documentation. PlatformIO. URL: https://docs.platformio.org/en/latest/ (Last accessed: 06.05.2026).

18. Hanulich S., Saltovskyi B. Development of a Multimedia Keyboard Based on the PAJ7620 Sensor. *Materials of Scientific Conferences of the Petro Mohyla Black Sea National University*. 2025. № 1. DOI: 10.34132/mspc2025.01.06.12.

19. HID-Project Library. Arduino Documentation. URL: https://docs.arduino.cc/libraries/hid-project/ (Last accessed: 06.05.2026).

20. Espressif Systems. ESP32-WROOM-32D & ESP32-WROOM-32U Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf (Last accessed: 06.05.2026).

21. ESP32-BLE-Keyboard Library. GitHub. URL: https://github.com/T-vK/ESP32-BLE-Keyboard (Last accessed: 06.05.2026).

22. Bluetooth SIG. Human Interface Device Profile 1.1.1. Bluetooth Specifications. URL: https://www.bluetooth.com/specifications/specs/human-interface-device-profile-1-1-1/ (Last accessed: 06.05.2026).

ДОДАТОК А

Деталізована блок-схема алгоритму роботи пристрою

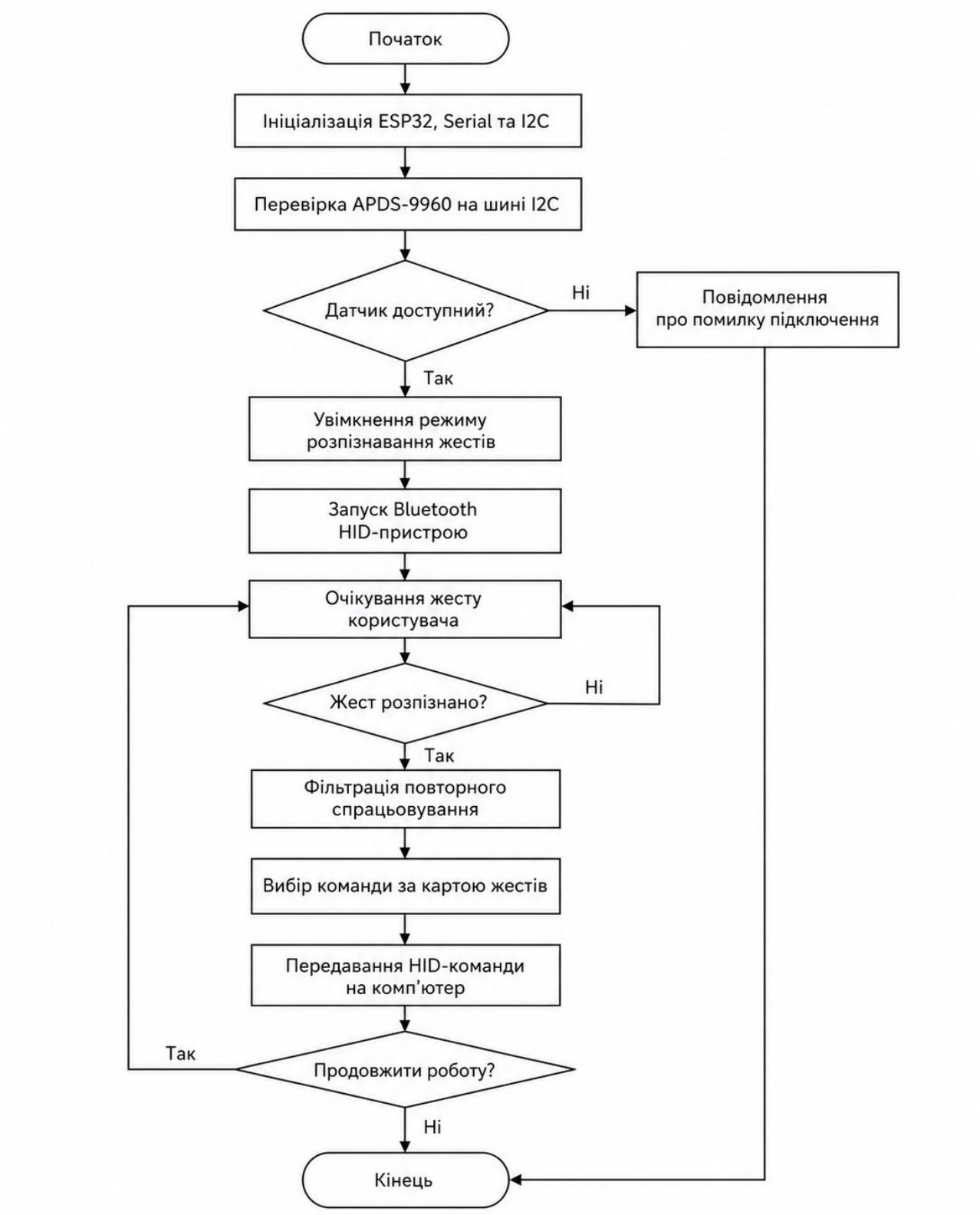

Рисунок А.1 – Деталізована блок-схема алгоритму роботи пристрою

ДОДАТОК Б

Код мікропрограмного забезпечення пристрою

```
#include <Wire.h>
#include <SparkFun_APDS9960.h>
#include <BleKeyboard.h>


SparkFun_APDS9960 apds = SparkFun_APDS9960();
BleKeyboard bleKeyboard("GestureController", "ESP32", 100);


const int I2C_SDA = 21;
const int I2C_SCL = 22;


unsigned long lastGestureTime = 0;
const unsigned long gestureDelay = 700;


void setup() {
Serial.begin(115200);
delay(1000);


Serial.println("APDS-9960 Gesture Controller");
Serial.println("Starting I2C bus...");


Wire.begin(I2C_SDA, I2C_SCL);
delay(200);


if (!apds.init()) {
Serial.println("APDS-9960 not found. Check wiring!");
while (1) {
delay(1000);
}
}


Serial.println("APDS-9960 initialized");


if (!apds.enableGestureSensor(false)) {
Serial.println("Gesture sensor error");
while (1) {
delay(1000);
}
}


bleKeyboard.begin();
Serial.println("Bluetooth HID Device started");
Serial.println("Waiting for gestures...");
}


void loop() {
if (apds.isGestureAvailable()) {
int gesture = apds.readGesture();


if (millis() - lastGestureTime < gestureDelay) {
Serial.println("Repeated trigger blocked");
return;
}


switch (gesture) {
case DIR_RIGHT:
handleGesture("RIGHT");
break;
```

```
case DIR_LEFT:
handleGesture("LEFT");
break;
case DIR_UP:
handleGesture("UP");
break;
case DIR_DOWN:
handleGesture("DOWN");
break;
case DIR_NEAR:
handleGesture("NEAR");
break;
case DIR_FAR:
Serial.println("Gesture: FAR");
break;
default:
Serial.println("Gesture: UNKNOWN");
break;
}
}
}

void handleGesture(const char* dir) {
Serial.print("Gesture: ");
Serial.println(dir);

if (!bleKeyboard.isConnected()) {
Serial.println("Bluetooth HID is not connected");
lastGestureTime = millis();
return;
}

if (strcmp(dir, "RIGHT") == 0) {
bleKeyboard.write(KEY_RIGHT_ARROW);
Serial.println("HID command: NEXT");
}
else if (strcmp(dir, "LEFT") == 0) {
bleKeyboard.write(KEY_LEFT_ARROW);
Serial.println("HID command: PREVIOUS");
}
else if (strcmp(dir, "UP") == 0) {
bleKeyboard.write(KEY_MEDIA_VOLUME_UP);
Serial.println("HID command: VOLUME_UP");
}
else if (strcmp(dir, "DOWN") == 0) {
bleKeyboard.write(KEY_MEDIA_VOLUME_DOWN);
Serial.println("HID command: VOLUME_DOWN");
}
else if (strcmp(dir, "NEAR") == 0) {
bleKeyboard.write(KEY_RETURN);
Serial.println("HID command: SELECT");
}
else {
Serial.println("Unknown gesture direction");
}

lastGestureTime = millis();
}

void printConnectionInfo() {
Serial.println("Device name: GestureController");
Serial.println("Sensor: APDS-9960");
Serial.println("I2C address: 0x39");
Serial.println("SDA: GPIO21");
Serial.println("SCL: GPIO22");
}
```

ДОДАТОК В

Матеріали апробації роботи

В.1 XXVIII Всеукраїнська науково-практична конференція «Могилянські читання – 2025»

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
ДНУ «Інститут модернізації змісту освіти»
Південний науковий центр НАН та МОН
Інститут української археографії та джерелознавства
імені М. С. Грушевського НАН України
Первинна профспілкова організація ЧНУ ім. Петра Могили

**«МОГИЛЯНСЬКІ ЧИТАННЯ – 2025:
досвід та тенденції розвитку суспільства в Україні: глобальний,
національний та регіональний аспекти»**

XXVIII Всеукраїнська науково-практична конференція

ТЕЗИ ДОПОВІДЕЙ

ТЕХНІЧНІ НАУКИ

Миколаїв, 10–14 листопада 2025 року

Миколаїв – 2025

Підсекції:

- **КОМП’ЮТЕРНА ІНЖЕНЕРІЯ**

УДК 004.353.4

Мельников А. Г.,
бакалаврант кафедри комп'ютерної інженерії,
Салтовський Б. Г.,
старший викладач кафедри комп'ютерної інженерії,
Чорноморський нац. ун-т ім. Петра Могили, м. Миколаїв, Україна

ПРИСТРІЙ КЕРУВАННЯ НА ОСНОВІ ДАТЧИКА APDS-9960

Актуальність теми. У сучасному світі спостерігається тенденція до створення безконтактних систем керування, які забезпечують зручність, гігієнічність та інтуїтивність взаємодії людини з технікою. Це особливо важливо у медичних і промислових середовищах, де дотик до екрана або клавіатури може бути неможливим або небажаним. Одним із перспективних рішень у цій сфері є застосування жестового керування.

Мета роботи. Розробити пристрій для безконтактного керування комп’ютером або смартфоном за допомогою жестів, з використанням

датчика APDS-9960 та мікроконтролера ESP32, який працює в режимі Bluetooth HID-клавіатури.

Основні компоненти системи. Датчик APDS-9960 здатен визначати напрямок руху руки (вліво, вправо, вгору, вниз), а також наближення чи віддалення об’єкта. Мікроконтролер ESP32 [1] (рис. 1) виконує обробку даних із сенсора та передає відповідні команди через Bluetooth на під’єднаний пристрій. Для реалізації зв’язку використано стандарт Bluetooth HID, який забезпечує сумісність із більшістю операційних систем.