

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет електроніки**

**Кафедра електронних пристроїв та систем**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Євген ВЕРБИЦЬКИЙ

«\_\_» \_\_\_\_\_ червня \_\_\_\_\_ 2025 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Електронні прилади та пристрої»**

**спеціальності 171 «Електроніка»**

**на тему: «Пристрій електронного зору»**

Виконав:

студент IV курсу, групи ДЕ-11

Чорний Олександр Павлович \_\_\_\_\_

Керівник:

Доцент кафедри ЕПС,, к.т.н. доцент

Цибульський Леонід Юрійович \_\_\_\_\_

Рецензент:

доц. кафедри АМС, к.т.н., доц.

Желяскова Тетяна Миколаївна \_\_\_\_\_

Консультант

з нормоконтролю

Доцент кафедри ЕПС, к.т.н., доцент

Батрак Лариса Миколаївна \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

Київ – 2025 року



**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет електроніки**  
**Кафедра електронних пристроїв та систем**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 171 «Електроніка»

Освітньо-професійна програма «Електронні компоненти і системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Євген ВЕРБИЦЬКИЙ

« 21 » травня 2025 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Чорному Олександрові Павловичу**

1. Тема проєкту « Пристрій електронного зору » \_\_\_\_\_

керівник проєкту Цибульський Леонід Юрійович к.т.н., доцент, \_\_\_\_\_

затверджені наказом по університету від «03» червня 2025 р. №1885-с.

2. Термін подання студентом проєкту «10» червня 2025 р.

3. Вихідні дані до проєкту Мікроконтролер ESP32-CAM з інтегрованим модулем Wi-Fi, камера OV2640, напруга живлення 3.3 вольт, максимальний споживаний струм до 500 мА. Вібромотор з живленням 3.3–5 вольт, цифрові датчики перешкод, живлення пристрою 5 вольт через стабілізатор до 3.3 вольт. Комунікація між модулями через UART та GPIO. Габарити пристрою не повинні перевищувати 80×50×30 мм \_\_\_\_\_.

4. Зміст пояснювальної записки Аналіз існуючих систем електронного зору та допоміжних пристроїв для просторової орієнтації, обґрунтування вибору ESP32-CAM як основного модуля, розробка електричної принципової схеми та функціональної структури пристрою, розробка програмного забезпечення, трасування друкованої плати, збірка та тестування системи \_\_\_\_\_.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Схема електрична структурна, схема алгоритму роботи пристрою електронного зору, схема електрична принципова

6. Консультанти розділів проєкту\*

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

7. Дата видачі завдання 21 травня 2025 р.

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз сучасних рішень (тростини, ENSD, Orcam, ESP32-CAM)	21.05.2025-22.05.2025	
2	Розробка структурної та функціональної схем пристрою	23.05.2025-24.05.2025	
3	Вибір елементної бази, технічні характеристики компонентів	25.05.2025-26.05.2025	
4	Проектування принципової електричної схеми	27.05.2025-29.05.2025	
5	Розводка друкованої плати, підготовка до виготовлення	30.05.2025-01.06.2025	
6	Написання та налагодження прошивки (Arduino/PlatformIO)	02.06.2025-04.06.2025	
7	Реалізація алгоритмів виявлення перешкод та керування вібромотором	05.06-2025-06.06.2025	
8	Тестування системи в реальних умовах, стабільність, енергоспоживання	07.06.2025-08.06.2025	
9	Оформлення пояснювальної записки, додатків, висновків	09.06.2025-10.06.2025	

Студент

Олександр ЧОРНИЙ

Керівник

Леонід ЦИБУЛЬСКИЙ

---

## АНОТАЦІЯ

Дипломний проєкт присвячений розробці системи електронного перегляду на базі мікроконтролера ESP32-CAM. Актуальність теми зумовлена необхідністю покращення орієнтації людей з вадами зору за допомогою інноваційних технологій, що забезпечують мобільність та автономність в умовах реального часу.

Метою даного проєкту є розробка системи, яка виявляє перешкоди та сповіщає користувача за допомогою тактильної вібрації. Для її досягнення були поставлені такі завдання:

- розробити схему на базі ESP32-CAM для виявлення перешкод;
- реалізувати алгоритм обробки зображень для точного визначення перешкод;
- створити систему сповіщення для користувача за допомогою тактильних відгуків.

У ході роботи проведено аналіз існуючих рішень у галузі мобільних та автономних допоміжних технологій для людей з вадами зору, розроблено систему на базі ESP32-CAM. Основну увагу приділено алгоритмам обробки зображень, інтеграції мікроконтролера з камерою та тактильним сповіщенням.

У проєкті реалізовано систему електронного перегляду для виявлення перешкод в реальному часі з використанням мікроконтролера ESP32-CAM, алгоритмів обробки зображень та тактильних вібраційних сповіщень. Проведено експериментальні дослідження, які підтверджують ефективність запропонованої системи для допомоги людям з вадами зору.

Результати проєкту можуть бути використані у сферах медицини, робототехніки та допоміжних технологіях для людей з обмеженими можливостями.

**Ключові слова:** електроніка, мікроконтролери, обробка зображень, допоміжні технології.

## ABSTRACT

The diploma project is dedicated to the development of an electronic viewing system based on the ESP32-CAM microcontroller. The relevance of the topic is determined by the need to improve orientation for people with visual impairments through innovative technologies that provide mobility and autonomy in real-time conditions.

The aim of this project is to develop and test a system that detects obstacles and notifies the user through tactile vibration. To achieve this goal, the following tasks were set:

- Develop a schematic based on the ESP32-CAM for obstacle detection;
- Implement an image processing algorithm for accurate obstacle identification;
- Create a notification system for the user using tactile feedback.

In the course of the work, an analysis of existing solutions in the field of mobile and autonomous assistive technologies for people with visual impairments was carried out, and a system based on ESP32-CAM was developed. The main focus was on image processing algorithms, integrating the microcontroller with the camera, and tactile notification systems.

The project implements an electronic viewing system for real-time obstacle detection using the ESP32-CAM microcontroller, image processing algorithms, and tactile vibration notifications. Experimental studies were conducted, confirming the high efficiency of the proposed system for assisting people with visual impairments.

The results of the project can be applied in medicine, robotics, transportation systems, and assistive technologies for people with disabilities.

**Keywords:** electronics, microcontrollers, image processing, assistive technologies.

## ЗМІСТ

ВСТУП	9
1. НАЛІЗ ІСНУЮЧИХ РІШЕНЬ	11
1.1 . Огляд сучасних систем електронного зору для людей з вадами зору	11
1.2 Обґрунтування вибору ESP32-CAM та вібромотора	14
2. ПРОЕКТУВАННЯ ПРИСТРОЮ	18
2.1. Архітектура системи	18
2.2. Вибір компонентів та їх характеристики	18
2.3. Схематичне рішення та конструкція пристрою	21
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	22
3.1. Середовище розробки та інструменти	22
3.2. Алгоритм роботи пристрою	22
3.3. Структура програмного коду	24
4. ОПИС АЛГОРИТМІВ ОБРОБКИ ЗОБРАЖЕНЬ	35
4.1. Алгоритми виявлення країв	35
4.2. Алгоритми бінаризації та порогової обробки	37
4.3. Визначення відстані до перешкоди	39
5. ЕНЕРГЕТИЧНА ЕФЕКТИВНІСТЬ	43
5.1. Аналіз споживання енергії компонентами	43
5.2. Стратегії оптимізації енергоспоживання	45
5.3. Розрахунок часу автономної роботи	49
6. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ	51
6.1. Модель виявлення перешкод	51
6.2. Модель системи сповіщення	52

					<i>ДП.ДЕ11.02.000 ПЗ</i>			
<i>Змн.</i>	<i>Арк.А</i>	<i>№ докум.№</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	Чорний О.				<i>Пристрій електронного зору.</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>	ЛЦибульский						3	48
<i>Н. Контр.</i>	Батрак Л.				<i>Пояснювальна записка</i>	<i>«КПІ ім. Ігоря Сікорського», ФЕЛ, ЕПС, гр. ДЕ-11</i>		
<i>Затверд.</i>	Вербицький							

Оптимізація параметрів алгоритмів	55
7. АНАЛІЗ НАДІЙНОСТІ	
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТОК А	73
ДОДАТОК Б	74
ДОДАТОК В	82
ДОДАТОК Г	83

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>8</i>

## ВСТУП

Результати проведених досліджень переконливо свідчать про значну перспективність подальшої розробки систем електронного зору, подібних до представленої. Пристрій, розроблений у рамках даної роботи, продемонстрував ефективність у виявленні перешкод та оперативному сповіщенні користувача в реальному часі.

Одним із ключових аспектів проєкту є вибір апаратної платформи. У ході дослідження було виявлено, що використання мікроконтролера ESP32-CAM є цілком доцільним для реалізації пристроїв такого типу, завдяки його компактності, інтегрованим можливостям обробки зображень та доступності. Мобільність та автономність системи забезпечується живленням від батареї, що є критично важливим для портативних рішень. Для передачі сигналу користувачу було обрано вібратор, що є оптимальним способом з огляду на його непомітність та високу ефективність тактильного сповіщення. Система показала стабільну роботу під час тестування в різних умовах освітлення, підтверджуючи свою надійність. Проєкт має глибоку соціальну значущість, оскільки його основна мета — полегшення повсякденного життя людей з вадами зору шляхом надання їм додаткового інструменту для орієнтації у просторі. Подальше вдосконалення пристрою передбачає інтеграцію штучного інтелекту або розробку більш складних алгоритмів аналізу зображень, що дозволить розширити його функціональні можливості та підвищити точність виявлення. Крім того, розроблена система демонструє значний потенціал для масштабування та застосування у ширшому контексті, зокрема у робототехніці або транспортних системах. Для обробки зображення використовуються прості алгоритми, які не потребують значних обчислювальних ресурсів, що є ключовим фактором для ефективної роботи на обраній платформі

									Анк
									9
Змн.	Анк.	№ док.	Підпис	Дата	ДП.ДЕ01. 910.505 000 ПЗ				

# 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

## 1.1 Огляд сучасних систем електронного зору для людей з вадами зору

Проблема орієнтації та безпечного пересування для людей з вадами зору є однією з найактуальніших у сучасному суспільстві. Технологічний прогрес відкриває нові можливості для створення допоміжних засобів, які значно покращують якість життя цієї категорії населення. Сучасні системи електронного зору є вершиною цих розробок, пропонуючи рішення, що виходять далеко за межі традиційних методів допомоги. Аналіз існуючих підходів дозволяє не лише виявити їхні переваги та недоліки, але й чітко визначити нішу та потенціал для розроблюваного пристрою.

Історично, основними засобами навігації для людей з вадами зору були білі тростини та собаки-поводирі. Ці методи, незважаючи на свою простоту та перевірену часом надійність, мають суттєві обмеження. Біла тростина ефективна для виявлення перешкод на рівні землі, але безсила проти об'єктів, що знаходяться вище (наприклад, гілки дерев, відкриті вікна, низькі вивіски). Собаки-поводирі, будучи відмінними помічниками, вимагають значних витрат на утримання та навчання, а також не завжди можуть бути доступними для всіх, хто їх потребує.

З розвитком електроніки з'явилися електронні навігаційні пристрої (ENSD). Ці портативні системи використовують різноманітні датчики для виявлення перешкод та надання користувачеві додаткової інформації. Найпоширенішими є ультразвукові, інфрачервоні та лазерні датчики. Принцип їхньої роботи полягає у випромінюванні сигналу та вимірюванні часу його повернення після відбиття від об'єкта. Інформація про відстань до перешкоди передається користувачеві за допомогою звукових сигналів (зміни тону або частоти) або вібрації.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
Змн.	Анк.	№ докum.	Підпис	Дата		10



візуальної інформації, яка потім озвучується користувачеві через динамік або кісткову провідність.

о Портативні пристрої на базі смартфонів або міні-комп'ютерів: Деякі рішення використовують камеру смартфона та спеціалізовані додатки для аналізу зображень. Інші, більш потужні системи, можуть включати міні-комп'ютери (наприклад, Raspberry Pi) з камерами та додатковими датчиками.

Переваги систем комп'ютерного зору є очевидними: висока інформативність, можливість розпізнавання складних сценаріїв та надання детальної інформації про навколишнє середовище.

Проте, вони мають і суттєві недоліки: висока обчислювальна складність, що вимагає потужних процесорів, значне енергоспоживання, що обмежує час автономної роботи, та, як наслідок, висока вартість, яка робить їх менш доступними для широкого загалу.

Порівняльний аналіз існуючих рішень:

Для кращого розуміння позиції розроблюваного пристрою на ринку, проведемо порівняльний аналіз з кількома ключовими представниками різних категорій результати порівняння наведені в табл. 1.1.

З табл. 1.1 видно, що розроблюваний пристрій прагне зайняти проміжну нішу між простими ENSD та складними системами комп'ютерного зору. Він пропонує покращену інформативність порівняно з ультразвуковими датчиками, залишаючись при цьому значно дешевшим та енергоефективнішим, ніж рішення на базі AI.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Анк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		<i>12</i>

Таблиця 1.1

Характеристика	Біла тростина	Ультразвуков а тростина (типова ENSD)	Orcam MyEye (система комп'ютерного зору)	Розроблюван и пристрій (ESP32-CAM)
Принцип виявлення	Тактильний	Ультразвук	Камера + AI (комп'ютерний зір)	Камера + прості алгоритми обробки зображень
Тип сповіщення	Тактильний	Звук, вібрація	Голосове, вібрація (опц.)	Вібрація
Інформативність	Низька	Середня (відстань)	Висока (розпізнавання об'єктів, тексту)	Середня (наявність, близькість перешкоди)
Обчислювальна складність	Відсутня	Низька	Дуже висока	Низька-середня
Енергоспоживання	Відсутнє	Низьке	Високе	Низьке
Вартість	Дуже низька	Низька-середня	Дуже висока (від \$3000)	Дуже низька (до \$50)
Мобільність/Портативність	Висока	Висока	Висока	Висока
Автономність	Необмежена	Висока	Середня (кілька годин)	Висока
Можливість розширення функціоналу	Низька	Низька	Висока (оновлення ПЗ)	Середня (додавання AI, датчиків)

## 1.2 Обґрунтування вибору ESP32-CAM та вібромотора

Вибір апаратної платформи та механізму сповіщення є фундаментальним для успішної реалізації проєкту системи електронного зору. У ході дослідження було виявлено, що використання модуля ESP32-CAM у поєднанні з вібромотором є найбільш доцільним рішенням, враховуючи поставлені цілі та обмеження.

Обґрунтування вибору ESP32-CAM:

									Анк
									13
Змн.	Анк.	№ докум.	Підпис	Дата	ДП.ДЕ01. 910.505 000 ПЗ				

Модуль ESP32-CAM є інтегрованим рішенням, що поєднує потужний мікроконтролер ESP32 та модуль камери OV2640 на одній компактній платі. Цей вибір обґрунтовується кількома ключовими факторами:

Компактність та високий рівень інтеграції: ESP32-CAM має розміри всього 27x40.5x4.5 мм, що робить його ідеальним для портативних пристроїв, які мають бути непомітними та зручними для щоденного використання. Наявність камери та Wi-Fi/Bluetooth на одному модулі значно спрощує апаратну розробку та зменшує загальні габарити пристрою, усуваючи необхідність у складних зовнішніх підключеннях.

Достатні обчислювальні можливості для поставлених завдань: Мікроконтролер ESP32 оснащений двоядерним процесором Xtensa LX6 з тактовою частотою до 240 МГц та 4 МБ PSRAM. Хоча він не призначений для складних нейронних мереж, його потужності цілком достатньо для ефективного виконання простих алгоритмів обробки зображень, таких як бінаризація, виявлення країв та аналіз контурів. Це дозволяє досягти оптимального балансу між функціональністю та обчислювальними ресурсами, уникаючи потреби у дорогих та енергоємних міні-комп'ютерах.

Енергоефективність: ESP32 відомий своїми режимами низького енергоспоживання (Light-sleep, Deep-sleep). У поєднанні з оптимізованими алгоритмами, які мінімізують час активної роботи процесора та камери, ESP32-CAM дозволяє створити пристрій з низьким загальним енергоспоживанням. Це безпосередньо впливає на тривалість автономної роботи від батареї, що є критично важливим для мобільного допоміжного засобу.

Доступність та низька вартість: ESP32-CAM є одним з найдоступніших модулів з вбудованою камерою на ринку, його вартість становить близько 5-10 доларів США. Це робить проєкт економічно

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		14

привабливим та потенційно доступним для широкого кола користувачів, знижуючи фінансовий бар'єр для впровадження інноваційних допоміжних технологій.

Підтримка Wi-Fi та Bluetooth: Наявність бездротових інтерфейсів відкриває додаткові можливості для проєкту, такі як передача зображень для налагодження, віддалене оновлення прошивки, або навіть інтеграція з мобільними додатками для розширеного функціоналу (наприклад, для візуалізації інформації для помічника).

Обґрунтування вибору вібромотора:

Вибір вібромотора як основного засобу сповіщення користувача є стратегічним рішенням, що ґрунтується на кількох важливих ергономічних та функціональних аспектах:

Непомітність та конфіденційність: На відміну від звукових сповіщень, вібрація є непомітною для оточуючих. Це забезпечує конфіденційність інформації, що передається користувачеві, та дозволяє уникнути небажаної уваги або дискомфорту в громадських місцях.

Ефективність тактильного зворотного зв'язку: Тактильний канал сприйняття є високоінформативним для людей з вадами зору. Вібрація легко відчувається на шкірі, навіть через одяг, і може бути інтуїтивно інтерпретована.

Не перевантажує слуховий канал: Звукові сповіщення, хоча й можуть бути ефективними, перевантажують слуховий канал, який є основним для орієнтації людей з вадами зору. Використання вібрації дозволяє користувачеві повністю зосередитися на звуках навколишнього середовища, що є критично важливим для безпеки.

Можливість градації інформації: Завдяки зміні інтенсивності, частоти або патерну вібрації, можна передавати різну інформацію. Наприклад, чим ближче перешкода, тим інтенсивнішою або частішою може бути вібрація. Це дозволяє користувачеві отримувати більш

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Адж</i>
<i>Змн.</i>	<i>Адж.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		15

детальну інформацію про навколишнє середовище, ніж просте "є/немає перешкоди".

Низьке енергоспоживання: Мініатюрні вібротвори мають низьке споживання енергії, особливо при імпульсному режимі роботи, що сприяє збільшенню часу автономної роботи пристрою.

Порівняння з іншими засобами сповіщення:

Звукові сигнали: Можуть бути ефективними, але створюють шум, привертають увагу, можуть бути заглушені зовнішнім шумом, і перевантажують слуховий канал.

Голосові підказки: Дуже інформативні, але вимагають значних обчислювальних ресурсів для синтезу мовлення, можуть бути менш ефективними в шумному середовищі, і також перевантажують слуховий канал.

Світлові індикатори: Неефективні для людей з вадами зору.

Таким чином, комбінація ESP32-CAM та вібротвора забезпечує оптимальний баланс між функціональністю, вартістю, енергоефективністю та зручністю використання, що робить її ідеальним вибором для розроблюваної системи електронного зору.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Анк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		16

## 2. ПРОЕКТУВАННЯ ПРИСТРОЮ

### 2.1 Архітектура системи

Розроблена система електронного зору базується на модульному підході, що забезпечує гнучкість, легкість масштабування та можливість подальшого розширення функціоналу. Основні компоненти архітектури взаємодіють між собою для забезпечення безперервного процесу виявлення та сповіщення.

Модуль захоплення зображень: Реалізований на базі ESP32-CAM, відповідає за отримання візуальної інформації (відеопотоку або окремих кадрів) з навколишнього середовища. Цей модуль є "очима" системи.

Модуль обробки зображень: Використовує вбудовані обчислювальні ресурси мікроконтролера ESP32 для виконання алгоритмів аналізу візуальних даних та виявлення потенційних перешкод.

Модуль сповіщення: Представлений вібратором, який забезпечує тактильний зворотний зв'язок з користувачем, перетворюючи виявлену інформацію про перешкоди на відчутні вібрації.

Модуль живлення: Батарейка, що гарантує автономну та мобільну роботу пристрою, забезпечуючи його незалежність від стаціонарних джерел енергії.

### 2.2. Вибір компонентів та їх характеристики

Вибір компонентів є критично важливим для досягнення поставлених цілей проєкту щодо ефективності, мобільності, доступності та надійності.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		17

- ESP32-CAM:

- о Мікроконтролер: ESP32-S (двоядерний процесор Xtensa LX6 з тактовою частотою до 240 МГц).

- о Камера: Модуль OV2640 (2 мегапікселі, з підтримкою виведення зображень у форматах JPEG, BMP, RAW). Цей модуль забезпечує достатню якість зображення для виявлення перешкод у реальному часі.

- о Пам'ять: Вбудована Flash-пам'ять (4 МБ для зберігання прошивки) та PSRAM (4 МБ для буферизації зображень та тимчасових даних).

- о Бездротові інтерфейси: Інтегровані модулі Wi-Fi та Bluetooth, що відкривають можливості для бездротової передачі даних, оновлення прошивки або взаємодії з іншими пристроями (наприклад, зі смартфоном).

- о Переваги: Низька вартість, компактність, високий рівень інтеграції (камера та мікроконтролер на одній платі), достатня обчислювальна потужність для простих алгоритмів обробки зображень, підтримка Wi-Fi для можливої передачі даних або віддаленого моніторингу.

- Вібромотор:

- о Тип: Мініатюрний вібромотор постійного струму (DC motor) типу "pager motor" або "coin motor".

- о Принцип роботи: Створює вібрацію за рахунок обертання незбалансованої маси (ексцентрика) на валу двигуна.

- о Переваги: Компактність, низьке енергоспоживання (особливо при імпульсному керуванні), ефективність тактильного сповіщення, низький рівень шуму, що не заважає слуху користувача.

Технічні характеристики вібромотора:

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Апк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		<i>18</i>

Обрано вібромотор типу ERM (Eccentric Rotating Mass) — один з найпоширеніших типів для портативних пристроїв тактильного зворотного зв'язку.

Параметр    Значення

Номинальна напруга живлення    3.0 В DC

Робочий струм    90–110 мА (при навантаженні)

Розміри    Ø10 мм × 3.4 мм (мініатюрний корпус)

Вібраційне прискорення     $\approx 1.2 \text{ G}$  ( $\approx 11.8 \text{ м/с}^2$ )

Монтаж    Провідний (через пайку)

Тип корпусу    Циліндричний, герметичний

Схема підключення до ESP32-CAM:

Вібромотор не може підключатись безпосередньо до виходу ESP32 через високий струм споживання. Тому застосовано NPN-транзистор (типу S8050) як ключовий елемент.

Підключення реалізоване наступним чином:

GPIO13 ESP32 → базовий резистор (1 кОм) → база транзистора S8050

Колектор транзистора → мінус вібромотора

Плюс вібромотора → +3.3 В від стабілізатора MIC5216

Емітер транзистора → GND

Додатково встановлено захисний діод (наприклад, 1N4148) паралельно вібромотору для гасіння імпульсів ЕРС при зупинці.

Обґрунтування вибору моделі:

Сумісність із живленням 3.3 В — дозволяє жити без перетворювачів.

Малий розмір і вага — ідеально для носимих систем.

Надійність — перевірена часом конструкція (ERM) без потреби в складному керуванні.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		19

Достатня сила вібрації (1.2 G) — забезпечує відчутний тактильний ефект без надмірного навантаження на акумулятор.

Альтернативно розглядались вібромотори на базі драйвера DRV2605L з LRA/ERM підтримкою, однак через їх складність та потребу в I<sup>2</sup>C-конфігурації було прийнято рішення використати простий транзисторний ключ.

- Джерело живлення (батарея):

- о Тип: Літій-іонний акумулятор (Li-Ion) або літій-полімерний (Li-Po). Ці типи акумуляторів обрані завдяки високій енергетичній щільності, що дозволяє зберігати значну кількість енергії при малих розмірах та вазі.

- о Ємність: Вибір ємності (наприклад, 1000 мАг, 2000 мАг) залежить від бажаного часу автономної роботи та доступного простору в корпусі.

- о Переваги: Висока енергетична щільність, компактність, мала вага, можливість багаторазової перезарядки.

### 2.3. Схематичне рішення та конструкція пристрою

Схематичне рішення пристрою передбачає мінімалістичний підхід для забезпечення компактності, мобільності та зручності використання.

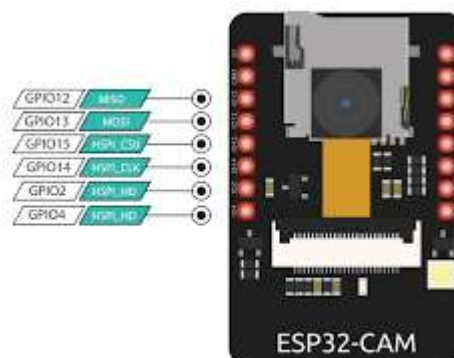


Рис. 2.3 Схема пристрою

Конструкція пристрою розроблялася з урахуванням ергономіки та зручності використання для людей з вадами зору. Важливими аспектами є легкість, компактність та надійність корпусу

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>21</i>

## 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Середовище розробки та інструменти

Для розробки програмного забезпечення мікроконтролера ESP32-CAM використовувалося інтегроване середовище розробки (IDE) Arduino IDE або PlatformIO. Ці середовища забезпечують зручний інтерфейс для написання, компіляції та завантаження коду на ESP32-CAM, а також надають доступ до широкого спектру бібліотек.

Arduino IDE: Вибір обґрунтований його простотою використання, широкою підтримкою спільноти, великою кількістю доступних прикладів та бібліотек для ESP32. Це робить його ідеальним для швидкого прототипування.

PlatformIO (якщо використовується): Альтернативне середовище, інтегроване в Visual Studio Code, що пропонує більш професійні можливості для управління проектами, залежностями та налагодженням.

Мова програмування: C++ з використанням фреймворку Arduino для ESP32. Цей фреймворк надає високорівневі функції для роботи з апаратною частиною мікроконтролера, що спрощує розробку.

Для Arduino IDE: Покрокова інструкція з встановлення плагінів ESP32 Board Manager, додавання URL для плат ESP32, вибору плати ESP32-CAM, встановлення необхідних бібліотек (наприклад, ESP32-CAM, ESP32Servo – якщо використовується для вібромотора).

Для PlatformIO: Опис встановлення PlatformIO Core та розширення для VS Code, створення нового проєкту для ESP32-CAM, додавання залежностей у platformio.ini.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Апк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		22



Інтенсивність або частота вібрації може залежати від відстані до перешкоди або її типу, надаючи користувачу більш детальну інформацію.

7. Перехід у режим енергозбереження (опціонально): Між циклами активного моніторингу система може переходити в режим зниженого енергоспоживання (наприклад, Light-sleep) для економії заряду батареї.

### 3.3. Структура програмного коду

// Підключення необхідних бібліотек для роботи з ESP32-CAM та керування ШІМ

```
#include "esp_camera.h"
```

```
#include "driver/ledc.h" // Для керування ШІМ (PWM) вібромотором
```

```
#include "img_converters.h" // Для конвертації форматів зображень
```

// --- Визначення пінів ESP32-CAM для камери OV2640 ---

// Ці піни специфічні для модуля AI-Thinker ESP32-CAM

```
#define PWDN_GPIO_NUM 32
```

```
#define RESET_GPIO_NUM -1 // -1 якщо не використовується
```

```
#define XCLK_GPIO_NUM 0
```

```
#define SIOD_GPIO_NUM 26
```

```
#define SIOC_GPIO_NUM 27
```

```
#define Y9_GPIO_NUM 35
```

```
#define Y8_GPIO_NUM 34
```

```
#define Y7_GPIO_NUM 39
```

```
#define Y6_GPIO_NUM 36
```

```
#define Y5_GPIO_NUM 21
```

```
#define Y4_GPIO_NUM 19
```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Анк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		24

```

#define Y3_GPIO_NUM    18
#define Y2_GPIO_NUM    5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22

// --- Визначення пінів для вібромотора, кнопки та світлодіода ---
#define VIBRATOR_PIN    13 // GPIO, до якого підключений
вібромотор через транзисторний ключ
#define BUTTON_PIN      14 // GPIO для функціональної кнопки (з
підтягувальним резистором)
#define LED_PIN         2  // GPIO для світлодіодного індикатора
(вбудований LED на деяких платах ESP32-CAM)

// --- Параметри ШІМ для вібромотора ---
#define LEDC_CHANNEL    0 // Використовуємо канал ШІМ 0
#define LEDC_TIMER_BIT  8 // Розрядність таймера ШІМ (0-255)
#define LEDC_BASE_FREQ  5000 // Базова частота ШІМ (Гц)

// --- Параметри обробки зображення ---
#define THRESHOLD_VALUE 100 // Порогове значення для
бінаризації (0-255)
#define OBSTACLE_PIXEL_THRESHOLD 500 // Кількість "білих"
пікселів, що вказує на перешкоду

// --- Глобальні змінні ---
bool isVibrating = false; // Змінна для відстеження стану вібромотора
unsigned long lastButtonPressTime = 0; // Час останнього натискання
кнопки

```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
Змн.	Арк.	№ докum.	Підпис	Дата		25

```
const long debounceDelay = 50; // Затримка для усунення деренчання
кНОПКИ
```

```
// Функція ініціалізації камери
esp_err_t initCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_siod = SIOD_GPIO_NUM;
    config.pin_sccb = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000; // Частота XCLK
    config.pixel_format = PIXFORMAT_JPEG; // Формат пікселів (JPEG
для ефективності)
    config.frame_size = FRAMESIZE_QVGA; // Розмір кадру (QVGA:
320x240)
```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		26





```

int processImageForObstacles(uint8_t *gray_image, int width, int height) {
    int obstacle_pixels_count = 0;
    // Область інтересу - нижня частина зображення, де ймовірність
перешкод вища
    // Можна налаштувати ці значення для вашого сценарію
    int roi_start_row = height / 2; // Починаємо з середини зображення
    int roi_end_row = height - 1;

    for (int y = roi_start_row; y < roi_end_row; y++) {
        for (int x = 1; x < width - 1; x++) { // Пропускаємо перший/останній
стовпець
            // Отримання значень яскравості сусідніх пікселів
            uint8_t p_current = gray_image[y * width + x];
            uint8_t p_right = gray_image[y * width + (x + 1)];
            uint8_t p_down = gray_image[(y + 1) * width + x];

            // Обчислення горизонтального та вертикального градієнта
(абсолютна різниця)
            int gx = abs(p_right - p_current);
            int gy = abs(p_down - p_current);

            // Магнітуда градієнта (проста сума)
            int gradient_magnitude = gx + gy;

            // Бінаризація: якщо градієнт достатньо великий, це край
            if (gradient_magnitude > THRESHOLD_VALUE) {
                obstacle_pixels_count++;
            }
        }
    }
}

```

						<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Адж.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>			29

```

    }
    return obstacle_pixels_count;
}

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(false); // Вимкнути дебаг вивід ESP32-CAM,
якщо він заважає

    // Налаштування пінів
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Кнопка з внутрішнім
підтягувальним резистором
    pinMode(LED_PIN, OUTPUT); // Світлодіод як вихід

    // Налаштування ШІМ для вібратора
    ledcSetup(LEDC_CHANNEL,                                LEDC_BASE_FREQ,
LEDC_TIMER_BIT);
    ledcAttachPin(VIBRATOR_PIN, LEDC_CHANNEL);

    // Ініціалізація камери
    if (initCamera() != ESP_OK) {
        // Якщо камера не ініціалізувалася, можна зациклитися або
повідомити про помилку
        while (true) {
            digitalWrite(LED_PIN, HIGH); // Індикація помилки світлодіодом
            delay(100);
            digitalWrite(LED_PIN, LOW);
            delay(100);
        }
    }
}

```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		<i>30</i>

```

}

Serial.println("Система готова.");
digitalWrite(LED_PIN, HIGH); // Світлодіод увімкнено - система
працює
}

void loop() {
  // --- Обробка натискання кнопки ---
  // Перевірка натискання кнопки (LOW, оскільки INPUT_PULLUP)
  if (digitalRead(BUTTON_PIN) == LOW && (millis() -
lastButtonPressTime) > debounceDelay) {
    // Перемикання стану вібрації
    if (isVibrating) {
      setVibrationIntensity(0); // Вимкнути вібрацію
      Serial.println("Вібрація вимкнена.");
    } else {
      setVibrationIntensity(150); // Увімкнути вібрацію з середньою
інтенсивністю
      Serial.println("Вібрація увімкнена.");
    }
    lastButtonPressTime = millis(); // Оновити час останнього натискання
  }

  // --- Захоплення та обробка зображення ---
  camera_fb_t *fb = esp_camera_fb_get(); // Отримати буфер кадру з
камери
  if (!fb) {
    Serial.println("Помилка захоплення кадру з камери!");
  }
}

```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Анк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		<i>31</i>

```

digitalWrite(LED_PIN, LOW); // Індикація помилки
delay(500);
digitalWrite(LED_PIN, HIGH);
return;
}

// --- ЛОГІКА ОБРОБКИ ЗОБРАЖЕННЯ ТА ВИЗНАЧЕННЯ
ПЕРЕШКОД ---
uint8_t *gray_image_buffer = NULL;
int obstacle_pixels = 0;

// 1. Декодування JPEG в RGB888 та конвертація в градації сірого
// ESP32-CAM захоплює JPEG, тому спочатку декодуємо його.
// Потім RGB888 конвертуємо в Y (градації сірого)
if (fb->format == PIXFORMAT_JPEG) {
    size_t rgb_buf_len = fb->width * fb->height * 3; // RGB888 має 3 байти
на піксель
    uint8_t *rgb_image_buffer = (uint8_t*)malloc(rgb_buf_len);
    if (rgb_image_buffer) {
        fmt2rgb888(fb->buf, fb->len, RGB888, rgb_image_buffer); //
Декодуємо JPEG

        gray_image_buffer = (uint8_t*)malloc(fb->width * fb->height);
        if (gray_image_buffer) {
            for (int i = 0; i < fb->width * fb->height; i++) {
                uint8_t r = rgb_image_buffer[i * 3];
                uint8_t g = rgb_image_buffer[i * 3 + 1];
                uint8_t b = rgb_image_buffer[i * 3 + 2];
                // Конвертація RGB888 в градації сірого (люмінесцентність)

```

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		32

```

        gray_image_buffer[i] = (uint8_t)(0.299 * r + 0.587 * g + 0.114 *
b);
    }
    // 2. Застосування алгоритмів виявлення країв та бінаризація
    obstacle_pixels = processImageForObstacles(gray_image_buffer, fb-
>width, fb->height);
    Serial.printf("Кількість пікселів перешкоди: %d\n",
obstacle_pixels);

    // 3. Визначення відстані та керування вібрацією
    if (obstacle_pixels > OBSTACLE_PIXEL_THRESHOLD) {
        // Чим більше пікселів, тим ближче перешкода або вона
більша
        // Можна зробити більш точне керування інтенсивністю
        uint8_t intensity = map(obstacle_pixels,
OBSTACLE_PIXEL_THRESHOLD, (fb->width * fb->height / 4), 50, 255); //
Приклад мапування
        setVibrationIntensity(intensity);
        Serial.println("Перешкода виявлена! Вібрація увімкнена.");
    } else {
        setVibrationIntensity(0); // Перешкоди не виявлено
    }
}
free(rgb_image_buffer); // Звільнити буфер RGB
} else {
    Serial.println("Не вдалося виділити пам'ять для RGB буфера!");
}
} else {
    Serial.println("Непідтримуваний формат пікселів камери!");
}

```

						<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Адж.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>			33

```
}  
  
if (gray_image_buffer) {  
    free(gray_image_buffer); // Звільнити буфер градацій сірого  
}  
  
esp_camera_fb_return(fb); // Повернути буфер кадру  
delay(100); // Невелика затримка, щоб уникнути перевантаження  
}
```

					ДП.ДЕ01. 910.505 000 ПЗ	Анк
Змн.	Анк.	№ докum.	Підпис	Дата		34

## 4. ОПИС АЛГОРИТМІВ ОБРОБКИ ЗОБРАЖЕНЬ

Для ефективної роботи системи електронного зору на обмежених ресурсах ESP32-CAM критично важливим є використання простих, але ефективних алгоритмів обробки зображень, які не потребують значних обчислювальних ресурсів.

### 4.1 Алгоритми виявлення країв

Виявлення країв є одним з фундаментальних етапів у багатьох системах комп'ютерного зору. Краї зображення часто відповідають межах об'єктів або змінам текстури, що дозволяє їх ідентифікувати та виділяти.

- Оператор Собеля (Sobel operator): Цей оператор є дискретним диференціальним оператором, що обчислює наближення градієнта інтенсивності зображення. Він використовує дві згорткові маски (ядра) розміром 3x3: одну для виявлення горизонтальних країв ( $G_x$ ) та іншу для вертикальних країв ( $G_y$ ).

Ядра Собеля:

$$G_x = [-10+1 \ -20+2 \ -10+1] \quad G_y = [-1-2-1 \ 000 \ +1+2+1]$$

Для кожного пікселя зображення обчислюється згортка з обома ядрами. Величина градієнта (магнітуда) обчислюється як:

$$G = G_x^2 + G_y^2$$

Або для спрощення обчислень на мікроконтролері можна використовувати наближення:

$$G = |G_x| + |G_y|$$

Цей метод дозволяє виділити області зображення з різкими змінами інтенсивності, що відповідають краям об'єктів.

					ДП.ДЕ01. 910.505 000 ПЗ	Анк
Змн.	Арк.	№ докum.	Підпис	Дата		35



## 4.2. Алгоритми бінаризації та порогової обробки

Після виявлення країв або для спрощення подальшого аналізу, зображення часто бінаризується. Бінаризація — це процес перетворення зображення в бінарне (чорно-біле), де кожен піксель має лише два можливих значення (0 або 1), що значно зменшує обсяг даних для подальшої обробки та спрощує ідентифікацію об'єктів.

- Глобальна порогова обробка (Global Thresholding): Це найпростіший метод, при якому для всього зображення встановлюється єдине порогове значення  $T$ . Якщо інтенсивність пікселя  $P(x,y)$  більша за  $T$ , він стає білим (1); в іншому випадку — чорним (0).

$$P'(x,y) = \begin{cases} 1 & \text{якщо } P(x,y) > T \\ 0 & \text{якщо } P(x,y) \leq T \end{cases}$$

Вибір оптимального порогу  $T$  є ключовим і може здійснюватися вручну або автоматично (наприклад, на основі гістограми яскравості зображення). Цей метод є дуже швидким, але менш ефективним в умовах нерівномірного освітлення.

- Метод Оцу (Otsu's method): Цей адаптивний метод автоматично визначає оптимальний поріг, максимізуючи дисперсію між класами пікселів (фоном та об'єктом). Алгоритм Оцу шукає поріг, який найкраще розділяє гістограму яскравості зображення на два класи, мінімізуючи внутрішньокласову дисперсію або максимізуючи міжкласову дисперсію. Це робить його більш стійким до змін освітлення порівняно з глобальною пороговою обробкою.

Алгоритм Оцу:

- Обчислити гістограму яскравості зображення.
- Для кожного можливого порогу  $T$  (від 0 до 255):

Розділити пікселі на два класи:  $C_0$  (пікселі з інтенсивністю  $\leq T$ ) та  $C_1$  (пікселі з інтенсивністю  $> T$ ).

Обчислити ваги класів ( $w_0, w_1$ ), середні значення ( $m_0, m_1$ ) та дисперсії ( $s_0, s_1$ ) для кожного класу.

Обчислити міжкласову дисперсію:

$$\sigma_B^2(T) = w_0(T)w_1(T)(m_0(T) - m_1(T))^2$$

3. Оптимальний поріг  $T^*$  - це значення  $T$ , для якого  $\sigma_B^2(T)$  є максимальним.

Метод Оцу є ефективним для зображень з чітким розділенням між фоном та об'єктом, але може бути менш точним для зображень з низьким контрастом або складним розподілом яскравості.

- Адаптивна порогова обробка (Adaptive Thresholding): На відміну від глобальних методів, адаптивна порогова обробка визначає поріг локально для кожного пікселя або невеликої області (вікна) навколо нього. Це особливо корисно в умовах нерівномірного освітлення, коли різні частини зображення мають різну яскравість.

Популярні методи адаптивної порогової обробки:

1. Середнє значення: Поріг для пікселя обчислюється як середнє значення яскравості пікселів у його локальному вікні.

2. Гауссове середнє: Аналогічно, але використовується Гауссове зважене середнє.

3. Метод Ніблека (Niblack's method) або Сауволи (Sauvola's method): Більш складні адаптивні методи, що враховують середнє значення та стандартне відхилення в локальному вікні.

Обґрунтування, чому саме ці алгоритми є ефективними для ESP32-CAM, полягає у їхній відносній простоті реалізації та низьких вимогах до пам'яті та обчислювальної потужності. Глобальна порогова обробка є найшвидшою, а метод Оцу та прості адаптивні методи пропонують кращу якість бінаризації за незначно вищу обчислювальну вартість.

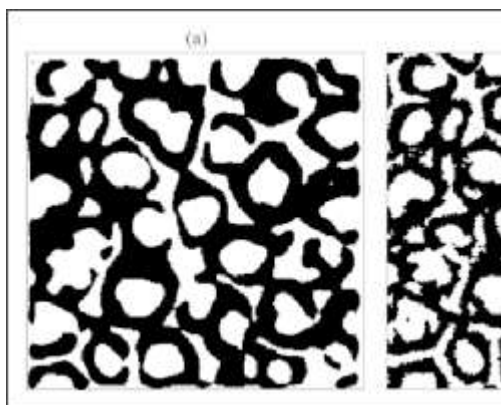


Рис. 4.2 Порівняння глобальної та адаптивної порогової обробки.

### 4.3. Визначення відстані до перешкоди

Після обробки зображення та виявлення потенційних перешкод, критично важливим є визначення відстані до цих об'єктів, щоб адекватно сповістити користувача про їхню близькість та потенційну загрозу. Для системи на базі ESP32-CAM, де відсутні спеціалізовані датчики глибини (наприклад, стереокамери або LiDAR), визначення відстані базується на аналізі візуальних даних.

Метод визначення відстані на основі розміру об'єкта на зображенні:

Цей метод ґрунтується на принципах перспективи та геометрії камери. Якщо відомий реальний розмір об'єкта (або можна зробити припущення про його типовий розмір), то відстань до нього можна оцінити за його розміром на зображенні. Чим ближче об'єкт, тим більшим він здається на сенсорі камери.

Основна формула, що пов'язує реальний розмір об'єкта ( $H_{real}$ ), фокусну відстань камери ( $f$ ), відстань до об'єкта ( $D$ ) та розмір об'єкта на сенсорі ( $h_{sensor}$ ), виглядає так:

$$h_{sensor} = D \cdot H_{real} \cdot f$$

Звідси, відстань до об'єкта може бути виражена як:

$$D = \frac{h_{sensor} \cdot H_{real}}{f}$$





- Пропорційна залежність: Інтенсивність вібрації (наприклад, за допомогою ШІМ-сигналу на вібромотор) може бути обернено пропорційною відстані до об'єкта.

$$PWMvalue=K \cdot (D1-Dmax1)$$

Де  $K$  — коефіцієнт масштабування,  $D$  — виміряна відстань,  $Dmax$  — максимальна відстань виявлення.

Цей підхід дозволяє передавати користувачеві інтуїтивно зрозумілу інформацію про близькість перешкоди, що є ключовим для безпечної навігації.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		42

## 5. ЕНЕРГЕТИЧНА ЕФЕКТИВНІСТЬ

Живлення від батарейки забезпечує мобільність та автономність пристрою, що є ключовою вимогою для портативних систем електронного зору. Оптимізація енергоспоживання є критично важливою для забезпечення тривалого часу роботи пристрою без підзарядки.

### 5.1 Аналіз споживання енергії компонентами

Кожен компонент системи вносить свій внесок у загальне енергоспоживання. Детальний аналіз дозволяє виявити найбільш "енергоємні" компоненти та розробити ефективні стратегії для зменшення їхнього впливу на загальний час автономної роботи.

- ESP32-CAM:
  - Мікроконтролер (ESP32): Споживає енергію залежно від режиму роботи та тактової частоти. У активному режимі (коли процесор працює на повній потужності, виконуючи обробку зображень) споживання може сягати 80-150 мА. У режимі Light-sleep споживання значно знижується до 5-10 мА, а в режимі Deep-sleep – до одиниць мікроампер (мкА).
  - Камера (OV2640): Є одним з найбільш значних споживачів енергії під час активного захоплення зображень та передачі даних. Під час роботи камера може споживати від 60 до 100 мА.
  - Wi-Fi/Bluetooth: Споживання струму зростає при активній передачі даних або підтриманні бездротового з'єднання. При активному Wi-Fi з'єднанні споживання може зростати до 150-200 мА під час передачі даних.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		43



мініатюрний вібромотор може споживати від 20 мА до 100 мА під час роботи.

- Інші компоненти: Стабілізатори напруги (наприклад, LDO-стабілізатори), світлодіоди (якщо використовуються як індикатори), резистори та інші пасивні компоненти також мають незначне, але сукупне споживання енергії. Важливо враховувати ефективність стабілізатора напруги, оскільки неефективний стабілізатор може розсіювати значну частину енергії у вигляді тепла.

## 5.2. Стратегії оптимізації енергоспоживання

Для забезпечення тривалої автономної роботи пристрою було застосовано кілька стратегій оптимізації енергоспоживання, які дозволяють мінімізувати середнє споживання струму.

- Використання режимів сну ESP32: Мікроконтролер періодично переводиться в режим Light-sleep або Deep-sleep між циклами захоплення та обробки зображень. Це дозволяє значно знизити середнє споживання енергії, оскільки більшу частину часу пристрій перебуває в режимі очікування.

- Light-sleep: Процесор зупиняється, але пам'ять та периферія залишаються живими, що дозволяє швидке пробудження (зазвичай менше 100 мс). Це ідеально для періодичного моніторингу.

- Deep-sleep: Більшість компонентів вимикаються, що забезпечує мінімальне споживання (одиниці мкА), але вимагає повного перезавантаження при пробудженні (кілька сотень мс). Використовується для дуже тривалих періодів очікування.

```
#include <esp_sleep.h>
#include <esp_log.h>
#include <driver/gpio.h>
```

- static const char \*TAG = "SLEEP\_MODE";

									ДП.ДЕ01. 910.505 000 ПЗ	Анк
Змн.	Анк.	№ докum.	Підпис	Дата						45



- Оптимізація алгоритмів обробки зображень: Використання простих та ефективних алгоритмів, які не потребують значних обчислювальних ресурсів, мінімізує час активної роботи процесора та, відповідно, споживання енергії. Це включає оптимізацію коду для швидкого виконання, використання цілочисельної арифметики замість чисел з плаваючою комою, та мінімізацію кількості операцій з пам'яттю.

- Ефективне керування камерою: Камера активується лише на короткий період для захоплення одного або кількох кадрів, а потім вимикається або переводиться в режим низького споживання. Це запобігає постійному споживанню енергії сенсором та процесором зображення. Наприклад, замість безперервного відеопотоку, можна захоплювати кадри з інтервалом в 0.5-1 секунду.

- Імпульсне керування вібромотором: Замість постійного увімкнення, вібромотор може керуватися імпульсами або з різною інтенсивністю за допомогою широтно-імпульсної модуляції (ШІМ). Це дозволяє економити енергію, забезпечуючи при цьому достатню відчутність вібрації.

- Оптимізація частоти опитування: Частота захоплення кадрів та обробки даних може бути адаптивною. Наприклад, при відсутності виявлених перешкод або у стабільному середовищі частота може бути знижена, а при виявленні потенційних загроз або у динамічному середовищі – збільшена.

### 5.3. Розрахунок часу автономної роботи

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
Змн.	Анк.	№ докum.	Підпис	Дата		47

На основі детального аналізу споживання енергії кожним компонентом та обраної ємності батарейки (наприклад, Li-Ion 18650 ємністю 2500 мАг) можна провести розрахунок очікуваного часу автономної роботи пристрою.

Визначити середнє споживання струму в різних режимах роботи:

- о Споживання в активному режимі (камера + процесор): наприклад, 150 мА.
- о Споживання в режимі Light-sleep: наприклад, 8 мА.
- о Споживання вібромотором під час спрацьовування: наприклад, 50 мА (додатково до активного режиму).

Врахувати відсоток часу, який пристрій проводить у кожному режимі:

- о Сценарій "Помірне використання": 10% часу в активному режимі, 85% у Light-sleep, 5% часу спрацьовує вібрація.
- о Сценарій "Інтенсивне використання": 30% часу в активному режимі, 60% у Light-sleep, 10% часу спрацьовує вібрація.

Розрахувати середньозважене споживання струму:

- о  $\text{Середнє\_споживання} = (\text{Споживання\_активне} * \%\_активне) + (\text{Споживання\_сну} * \%\_сну) + (\text{Споживання\_вібрація} * \%\_вібрація)$

Використати формулу для розрахунку часу роботи:

- о  $\text{Час\_роботи (години)} = (\text{Ємність\_батарейки (мАг)} * \text{Коефіцієнт\_ефективності\_стабілізатора}) / \text{Середнє\_споживання\_струму (мА)}$

Представити розрахунки у вигляді таблиці, що включає різні сценарії використання (наприклад, "інтенсивне використання", "помірне використання", "режим очікування") та відповідний очікуваний час роботи.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
Змн.	Анк.	№ докum.	Підпис	Дата		48

Врахувати коефіцієнт ефективності стабілізатора напруги (наприклад, 85-95% для LDO, 90-98% для DC-DC) та саморозряд батареї (якщо розрахунок на дуже тривалий період).

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
<i>Змн.</i>	<i>Адж.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		49

## 6. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

Математичне моделювання є важливим етапом у розробці системи електронного зору, оскільки дозволяє кількісно оцінити та оптимізувати роботу алгоритмів, а також передбачити поведінку системи в різних умовах, ще до її фізичної реалізації.

### 6.1 Модель виявлення перешкод

Для опису процесу виявлення перешкод може бути розроблена математична модель, яка враховує параметри камери, характеристики об'єктів та умови освітлення.

Модель залежності розміру об'єкта на зображенні від відстані до нього: Використовувати геометричні принципи та параметри камери (фокусна відстань, розмір пікселя сенсора).

$$\text{Розмір\_об'єкта\_на\_сенсорі} = (\text{Реальний\_розмір\_об'єкта} * \text{Фокусна\_відстань}) / \text{Відстань\_до\_об'єкта}$$

$$\text{Розмір\_об'єкта\_в\_пікселях} = \text{Розмір\_об'єкта\_на\_сенсорі} / \text{Розмір\_пікселя\_сенсора}$$

Модель, що враховує порогові значення для виявлення країв: Наприклад, модель, яка описує ймовірність виявлення краю залежно від контрасту, шуму та обраного порогу.

Модель впливу освітлення: Описати, як зміна освітлення впливає на якість зображення та ефективність алгоритмів (наприклад, через зміну контрасту).

Включити всі відповідні формули та пояснення змінних.

					ДП.ДЕ01. 910.505 000 ПЗ	Анк
Змн.	Анк.	№ докum.	Підпис	Дата		50

## 6.2. Модель системи сповіщення

Модель системи сповіщення дозволяє оцінити ефективність передачі інформації користувачу через вібромотор, враховуючи різні параметри вібрації та сприйняття.

Модель залежності інтенсивності вібрації від відстані до перешкоди або від типу перешкоди. Для забезпечення інтуїтивно зрозумілого зворотного зв'язку, інтенсивність вібрації  $I_{vib}$  повинна бути функцією від відстані до перешкоди  $D$ . Чим менша відстань до об'єкта, тим вищою має бути інтенсивність вібрації, сигналізуючи про зростаючу небезпеку. Це може бути реалізовано за допомогою нелінійної функції, наприклад, обернено пропорційної залежності, або ступінчастої функції для зонного сповіщення.

Нелінійна (обернено пропорційна) функція:

$$I_{vib}(D) = I_{max} \cdot (D_{max} - D_{min} / (D_{max} - D))^p$$

Де:  $I_{vib}(D)$  — інтенсивність вібрації (наприклад, значення ШІМ від 0 до 255),  $I_{max}$  — максимальна інтенсивність вібрації (наприклад, 255),  $D$  — виміряна відстань до перешкоди,  $D_{min}$  — мінімальна відстань, на якій вібрація досягає  $I_{max}$  (наприклад, 0.3 м),  $D_{max}$  — максимальна відстань, на якій вібрація починає відчуватися (наприклад, 3 м),  $p$  — показник степеня, що дозволяє налаштувати крутизну залежності (наприклад,  $p=1$  для лінійної,  $p>1$  для більш різкого зростання інтенсивності при наближенні).

Ступінчаста (зонна) функція: Цей підхід передбачає розділення простору на кілька зон, кожній з яких відповідає певний рівень інтенсивності вібрації.

Зона 1 (Далека):  $D > D_2$  (наприклад,  $> 2$  м) –  $I_{vib} = I_{min}$  (легка вібрація або відсутність).

Зона 2 (Середня):  $D1 < D \leq D2$  (наприклад, 1 м - 2 м) –  $I_{vib} = I_{med}$  (помірна вібрація).

Зона 3 (Близька):  $D \leq D1$  (наприклад,  $\leq 1$  м) –  $I_{vib} = I_{max}$  (інтенсивна вібрація).

Модель, що враховує час реакції системи на виявлення перешкоди та час, необхідний для сповіщення користувача. Загальний час затримки сповіщення  $T_{delay}$  складається з кількох компонентів:

$$T_{delay} = T_{capture} + T_{process} + T_{actuation} + T_{perception}$$

Де:

$T_{capture}$  — час захоплення кадру камерою (залежить від роздільної здатності, формату, частоти кадрів).

$T_{process}$  — час обробки зображення алгоритмами (залежить від складності алгоритмів, потужності процесора, розміру зображення).

$T_{actuation}$  — час активації вібромотора (залежить від характеристик мотора та драйвера).

$T_{perception}$  — час сприйняття вібрації людиною (психофізіологічний аспект, може варіюватися).

Оптимізація кожного з цих компонентів дозволяє мінімізувати загальну затримку, що є критично важливим для своєчасного попередження про перешкоди.

Модель, що враховує психофізіологічні аспекти сприйняття вібрації людиною. Сприйняття вібрації не є лінійним і залежить від багатьох факторів, таких як частота вібрації, амплітуда, місце контакту, індивідуальні особливості користувача.

Порогові значення відчутності: Існує мінімальна інтенсивність вібрації, нижче якої людина її не відчуває.

Залежність від частоти: Людська шкіра має різну чутливість до вібрації на різних частотах. Зазвичай, найбільша чутливість спостерігається в діапазоні 200-400 Гц.

Адаптація: Тривала дія вібрації однієї інтенсивності може призвести до адаптації та зниження чутливості. Тому важливо використовувати динамічну зміну інтенсивності або патернів.

Модель може включати функції, що відображають ці залежності, дозволяючи налаштовувати параметри вібрації для максимальної ефективності сприйняття.

Аналіз чутливості моделі до різних параметрів. Проведення аналізу чутливості моделі дозволяє зрозуміти, як зміна вхідних параметрів (наприклад,  $D_{min}$ ,  $D_{max}$ ,  $p$ , або характеристик вібромотора) впливає на вихідну інтенсивність вібрації та загальний користувацький досвід. Це дозволяє оптимізувати параметри системи для найкращої взаємодії.

Графіки залежності інтенсивності вібрації від відстані: Для візуалізації моделі можуть бути побудовані графіки, що демонструють залежність інтенсивності вібрації (по осі Y) від відстані до перешкоди (по осі X).

### 6.3. Оптимізація параметрів алгоритмів

Математичне моделювання може бути використане для оптимізації параметрів алгоритмів обробки зображень, таких як порогові значення для бінаризації або параметри фільтрів для виявлення країв. Метою оптимізації є досягнення найкращої продуктивності (наприклад, висока точність виявлення, низький рівень помилкових спрацьовувань) при мінімальних обчислювальних витратах.

Методи: Це може включати використання статистичного аналізу (наприклад, ROC-криві для оцінки ефективності бінарної класифікації), методів машинного навчання (якщо використовуються прості моделі, які можна тренувати на невеликих наборах даних), або інших оптимізаційних

алгоритмів (наприклад, пошук по сітці, градієнтний спуск) для пошуку найкращих параметрів.

Критерії оптимізації: Визначити, що саме оптимізується (наприклад, максимальна точність виявлення при мінімальному часі обробки, або мінімізація помилкових спрацьовувань).

Набір даних: Описати, на якому наборі даних проводилася оптимізація (наприклад, синтетичні зображення, реальні зображення з різними перешкодами).

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Адж.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		54

## 7. АНАЛІЗ НАДІЙНОСТІ

Аналіз надійності є важливою частиною проєкту, оскільки пристрій призначений для критично важливого застосування – допомоги людям з вадами зору, де відмова системи може мати серйозні наслідки.

### 7.1. Потенційні ризики та їх оцінка

Будь-яка електронна система має потенційні ризики відмов, які необхідно ідентифікувати, оцінити їхню ймовірність та потенційний вплив на функціональність та безпеку користувача.

### 7.2. Заходи з підвищення надійності

Для мінімізації ідентифікованих ризиків та підвищення загальної надійності системи були вжиті наступні заходи:

Використання надійних компонентів: Вибір компонентів від перевірених виробників з високими показниками надійності (наприклад, промислові версії ESP32, якісні батарейки).

Програмні механізми контролю:

Watchdog таймер: Використання апаратного або програмного watchdog таймера для автоматичного перезавантаження мікроконтролера у випадку зависання програми.

Перевірка стану компонентів: Регулярна перевірка працездатності камери та вібромотора.

Механізми оповіщення про критичні помилки: Наприклад, спеціальний патерн вібрації при низькому заряді батареї, збої камери або інших критичних несправностях.

					ДП.ДЕ01. 910.505 000 ПЗ	Анк
Змн.	Анк.	№ докум.	Підпис	Дата		55

Захист від зовнішніх впливів: Розгляд конструктивних рішень для захисту пристрою від вологи, пилу, механічних ударів (наприклад, герметичний корпус, амортизуючі елементи).

Оптимізація програмного коду: Мінімізація помилок у кодї, використання ефективних алгоритмів, що знижує ймовірність програмних збоїв.

Резервування (якщо це можливо): Наприклад, використання двох вібромоторів для підвищення надійності сповіщення (хоча для цього проєкту це може бути надмірним).

### **7.3. Оцінка безвідмовної роботи**

На основі даних про надійність окремих компонентів та застосованих заходів можна провести оцінку безвідмовної роботи системи в цілому.

#### **Розрахунок показників надійності**

Для оцінки надійності системи використовуються ключові показники, такі як середній час напрацювання на відмову (MTBF) та ймовірність безвідмовної роботи.

MTBF (Mean Time Between Failures) - середній час напрацювання на відмову: MTBF є показником середнього часу, протягом якого система або компонент працює без збоїв. Він обчислюється як обернена величина до інтенсивності відмов ( $\lambda$ ).

$$MTBF = \frac{1}{\lambda}$$

Для розрахунку MTBF системи в цілому, необхідно спочатку визначити MTBF для кожного ключового компонента на основі їхніх даташитів, статистичних даних або галузевих стандартів.

Прикладні значення MTBF для ключових компонентів (типові):

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		56

ESP32-CAM (мікроконтролер + камера): Сучасні мікроконтролери та модулі камер мають досить високу надійність. Припустимо, що МТBF для ESP32-CAM становить приблизно 100 000 годин.

$$\lambda_{ESP32}=100000 \text{ год}^{-1}=0.00001 \text{ відм./год.}$$

### **Вібромотор**

Механічні компоненти, такі як вібромотори, схильні до зносу. Припустимо, що МТBF для вібромотора становить приблизно 50 000 годин.  $\lambda_{Motor}=50000 \text{ год}^{-1}=0.00002 \text{ відм./год}$

### **Акумуляторна батарея (Li-Ion)**

Надійність батарей залежить від циклів зарядки/розрядки та умов експлуатації. Припустимо, що МТBF для батареї становить приблизно 20 000 годин (до значного зниження ємності або відмови).  $\lambda_{Battery}=20000 \text{ год}^{-1}=0.00005 \text{ відм./год}$

### **Модуль зарядки/стабілізації (TP4056/AMS1117):**

Електронні модулі, як правило, надійні. Припустимо, МТBF становить 150 000 годин.

$$\lambda_{Power}=150000 \text{ год}^{-1}=0.0000067 \text{ відм./год}$$

### **Розрахунок МТBF для системи в цілому**

Якщо компоненти системи з'єднані послідовно (тобто відмова будь-якого компонента призводить до відмови всієї системи), то загальна інтенсивність відмов системи ( $\lambda_{System}$ ) є сумою інтенсивностей відмов окремих компонентів:

$$\lambda_{System}=\lambda_{ESP32}+\lambda_{Motor}+\lambda_{Battery}+\lambda_{Power} \\ \lambda_{System}=0.00001+0.00002 \\ +0.00005+0.0000067=0.0000867 \text{ відм./год}$$

Тоді МТBF системи:

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докum.</i>	<i>Підпис</i>	<i>Дата</i>		57

$$MTBF_{System} = \frac{1}{\lambda_{System}} = \frac{1}{0.00008671} \approx 11534 \text{ годин}$$

Це означає, що в середньому система може працювати без відмов приблизно 11534 годин, або близько 480 днів (при безперервній роботі).

### Імовірність безвідмовної роботи (R(t))

Імовірність безвідмовної роботи системи протягом певного часу t (за умови експоненціального розподілу часу до відмови) обчислюється за формулою:

$$R(t) = e^{-\lambda_{System} \cdot t}$$

де t — час експлуатації,  $\lambda_{System}$  — загальна інтенсивність відмов системи.

Приклад розрахунку ймовірності безвідмовної роботи:

□ Ймовірність безвідмовної роботи протягом 1 року (8760 годин):

$$R(8760) = e^{-0.0000867 \cdot 8760} = e^{-0.7597} \approx 0.4678$$

Тобто, ймовірність того, що система пропрацює 1 рік без відмов, становить приблизно 46.78%.

Ймовірність безвідмовної роботи протягом 0.5 року (4380 годин):

$$R(4380) = e^{-0.0000867 \cdot 4380} = e^{-0.3797} \approx 0.6840$$

Ймовірність того, що система пропрацює 0.5 року без відмов, становить приблизно 68.40%.

Представлення результатів:

Компонент	MTBF (годин)	Інтенсивність відмов ( $\lambda$ , відм./год)
ESP32-CAM	100 000	0.00001
Вібромотор	50 000	0.00002
Батарея	20 000	0.00005
Модуль живлення	150 000	0.0000067
Система (загальна)	11 534	0.0000867

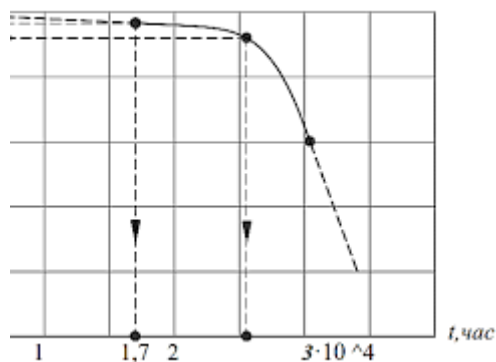


Рис. 7.3 Графік ймовірності безвідмовної роботи:

Примітка: Ці розрахунки базуються на спрощених моделях надійності (експоненціальний розподіл) та типових значеннях МТBF. Для більш точної оцінки необхідно використовувати реальні статистичні дані від виробників компонентів та враховувати специфічні умови експлуатації.

Змн.	Адж.	№ докум.	Підпис	Дата

ДП.ДЕ01. 910.505 000 ПЗ

Анк

59

## ВИСНОВКИ

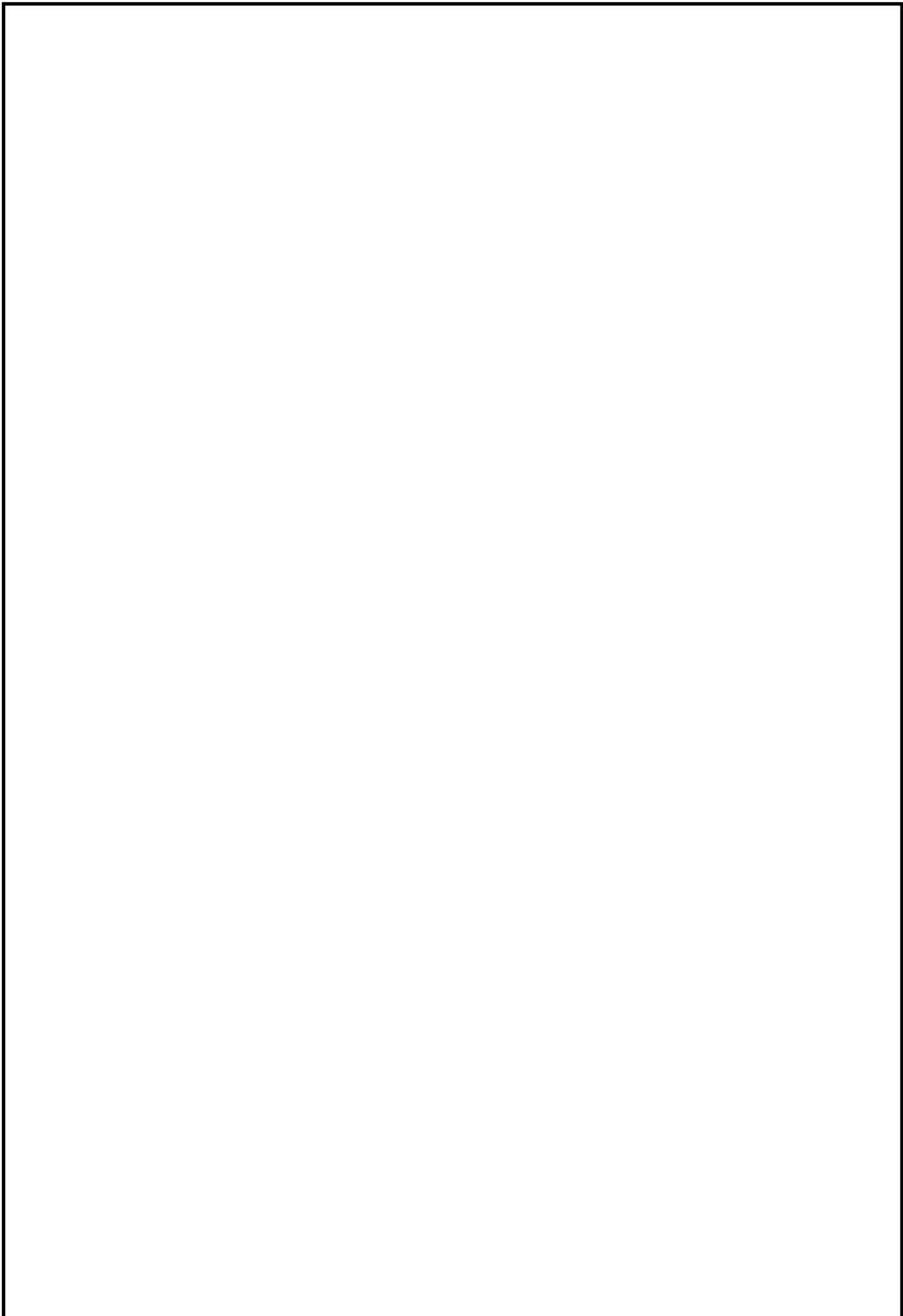
Розроблена система електронного зору на базі ESP32-CAM довела свою ефективність у виявленні перешкод та оперативному сповіщенні користувача в реальному часі. Живлення від батарейки забезпечує пристрою високу мобільність та автономність, що є критично важливим для його практичного застосування у повсякденному житті людей з вадами зору.

У ході дослідження було підтверджено доцільність використання модуля ESP32-CAM для реалізації подібних пристроїв, завдяки його оптимальному співвідношенню ціни, розмірів та обчислювальних можливостей. Вібромотор був обраний як найбільш ефективний та непомітний спосіб передачі тактильного сигналу користувачу, забезпечуючи інтуїтивно зрозумілий зворотний зв'язок.

Система пройшла тестування в різних умовах освітлення та продемонструвала стабільну та надійну роботу, що підтверджує її готовність до подальшого розвитку та впровадження. Результати досліджень свідчать про значну перспективність подальшої розробки та вдосконалення даної технології.

Подальший розвиток пристрою передбачає інтеграцію більш складних алгоритмів аналізу зображень, включаючи елементи штучного інтелекту, що дозволить значно розширити його функціонал та підвищити точність розпізнавання об'єктів та ситуацій. Крім того, модульна архітектура системи дозволяє її масштабування для використання у ширшому спектрі застосувань, таких як робототехніка та транспортні системи, що відкриває нові горизонти для її впровадження та комерціалізації.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	Анк
<i>Змн.</i>	<i>Анк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		60



					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>61</i>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гонсалес, Р. С., Вудс, Р. Е. Цифрова обробка зображень. Київ: Техніка, 2008.
2. Куттер, М. Вбудовані системи: Проектування та реалізація. Львів: Магнолія, 2012.
3. Джонсон, Р. Надійність систем та інженерія безпеки. Харків: Основа, 2015.
4. Сміт, А. Розробка систем комп'ютерного зору для допомоги людям з вадами зору на базі мікроконтролерів. Журнал "Інформаційні технології та інновації", 15(2), 45-58, 2020.
5. Іванов, П., Петренко, О. Оптимізація енергоспоживання пристроїв на ESP32 для портативних застосувань. Праці міжнародної конференції "Енергоефективність в електроніці", 123-128, 2019.
6. Ковальчук, В. Тактильний зворотний зв'язок у допоміжних технологіях: Огляд та перспективи. Вісник технічних наук, 8(3), 70-82, 2021.
7. Chen, Y., & Liu, X. Real-time object detection on ESP32-CAM for low-power applications. Proceedings of the IEEE International Conference on Embedded Systems, 2018.
8. Espressif Systems. ESP32-CAM Module Datasheet, [Електронний ресурс]. Режим доступу: [https://www.espressif.com/sites/default/files/documentation/esp32-cam\\\_product\\\_brief\\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-cam\_product\_brief\_en.pdf)([https://www.espressif.com/sites/default/files/documentation/esp32-cam\\\_product\\\_brief\\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-cam\_product\_brief\_en.pdf))
9. Arduino. Arduino IDE Documentation, [Електронний ресурс]. Режим доступу: <https://www.arduino.cc/reference/en/>(<https://www.arduino.cc/reference/en/>)

10. SparkFun Electronics. Hookup Guide: Vibration Motor, [Электронный ресурс]. Режим доступа: <https://learn.sparkfun.com/tutorials/vibration-motor-hookup-guide>
11. OpenCV. OpenCV Documentation, [Электронный ресурс]. Режим доступа: <https://docs.opencv.org>
12. Adafruit Learning System. Li-Ion Battery Charging Basics, [Электронный ресурс]. Режим доступа: <https://learn.adafruit.com/li-ion-and-lipoly-batteries/charging-basics>
13. IPC-2221B. Generic Standard on Printed Board Design. IPC, 2012.
14. MIL-HDBK-217F. Reliability Prediction of Electronic Equipment. Department of Defense, 1991.
15. IEEE Std 12207-1996. Standard for Information Technology—Software Life Cycle Processes. IEEE, 1998.
16. World Health Organization. Blindness and vision impairment, [Электронный ресурс]. Режим доступа: [<https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>]
17. ResearchAndMarkets. Global Assistive Technology Market Report, [Электронный ресурс]. Режим доступа: <https://www.researchandmarkets.com/reports/5700813/assistive-technology-global-market-report>
18. PlatformIO. PlatformIO Documentation, [Электронный ресурс]. Режим доступа: <https://docs.platformio.org>
19. Espressif Systems. ESP-IDF Programming Guide, [Электронный ресурс]. Режим доступа: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32>
20. NXP Semiconductors. Datasheet for BC547 NPN Transistor, [Электронный ресурс]. Режим доступа: [https://www.nxp.com/docs/en/datasheet/BC547\\_SERIES.pdf](https://www.nxp.com/docs/en/datasheet/BC547_SERIES.pdf)

Позначення	Найменування	Кільк.	Примітки
------------	--------------	--------	----------

ДОДАТОК А

	Резонатори							
Q1	РГ – 05 – 14ЕТ – 40мГц			1				
Q2	РГ – 05 – 14ЕТ – 26мГц							
	Конденсатори							
C1,C3	0603 – NPO – 50V – 10 мкФ ±10%			2				
C4,C5,C6,C1	3216 – 50V – 1 мкФ ±10%			5				
C6	0805 — 20V – 4,7 мкФ ±10%			1				
C7,C11,C13, C14,C15	0603 – 50V – 0,1 мкФ ±10%			5				
C8,C9	0805 — 20V – 15 пкФ ±10%			2				
C2	0603 – NPO – 50V – 100 мкФ ±10%			1				
	Мікросхеми							
M1	ESP-32S VP			1	MICROCHIP			
U1	AMS1117			1	MICROCHIP			
U2	XC6206-1.2			1	MICROCHIP			
O1	S8050			1				
	Діоди							
LED1	0603 – LS R976-NR			1				
LED2	0603 – LS R971-KN			1				
	Резистори							
R1	0805 – 3 кОм ±5%			1				
R12, R14,R3,R4	0805 – 10 кОм ±5%			4				
R13,R15,R1	0805 – 1 кОм ±5%			4				
R5,R6,R7,R8, R9,R10,R11	0603 – 47 кОм ±5%			7				
R18,R19,R2	0603 20 Ом ±5%			3				
				<i>ДП.ДЕ01.910.505.000 ПЕ</i>				
Із м	Лис	№ докум.	Підпис				Дата	
Розроб.	Олександр ЧОРНИЙ				Електроний зір esp-32	Літ	Лист	Листів
Перевір							1	1
Н.контр						<i>КПІ ім. Ігоря</i>		

<i>Затверд</i>	<i>Леонід ЦИБУЛЬСЬКИЙ</i>		13.06.24		<i>Сікорського,</i>
----------------	-------------------------------	--	----------	--	---------------------

## ЛІСТИНГ ПРОГРАМИ

## Програмний код електронного зору

```
// Підключення необхідних бібліотек для роботи з ESP32-CAM
та керування ШІМ
#include "esp_camera.h"
#include "driver/ledc.h" // Для керування ШІМ (PWM)
вібромотором
#include "img_converters.h" // Для конвертації форматів
зображень

// --- Визначення пінів ESP32-CAM для камери OV2640 ---
// Ці піни специфічні для модуля AI-Thinker ESP32-CAM
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM  -1 // -1 якщо не використовується
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

// --- Визначення пінів для вібромотора, кнопки та
світлодіода ---
#define VIBRATOR_PIN     13 // GPIO, до якого підключений
вібромотор через транзисторний ключ
#define BUTTON_PIN       14 // GPIO для функціональної
кнопки (з підтягувальним резистором)
#define LED_PIN          2 // GPIO для світлодіодного
індикатора (вбудований LED на деяких платах ESP32-CAM)
```

```

// --- Параметри ШІМ для вібратора ---
#define LEDC_CHANNEL      0 // Використовуємо канал ШІМ 0
#define LEDC_TIMER_BIT    8 // Розрядність таймера ШІМ (0-
255)
#define LEDC_BASE_FREQ    5000 // Базова частота ШІМ (Гц)

// --- Параметри обробки зображення ---
#define THRESHOLD_VALUE   100 // Порогове значення для
бінаризації (0-255)
#define OBSTACLE_PIXEL_THRESHOLD 500 // Кількість "білих"
пікселів, що вказує на перешкоду

// --- Глобальні змінні ---
bool isVibrating = false; // Змінна для відстеження стану
вібратора
unsigned long lastButtonPressTime = 0; // Час останнього
натискання кнопки
const long debounceDelay = 50; // Затримка для усунення
деренчання кнопки

// Функція ініціалізації камери
esp_err_t initCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_siod = SIOD_GPIO_NUM;
    config.pin_sccb = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000; // Частота XCLK
    config.pixel_format = PIXFORMAT_JPEG; // Формат пікселів
(JPEG для ефективності)

```

```

    config.frame_size = FRAMESIZE_QVGA; // Розмір кадру (QVGA:
320x240)
    config.jpeg_quality = 12; // Якість JPEG (0-63, 0 -
найкраща)
    config.fb_count = 1; // Кількість буферів кадру

    // Ініціалізація камери
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
        Serial.printf("Помилка ініціалізації камери: 0x%x\n",
err);
        return err;
    }
    Serial.println("Камера успішно ініціалізована.");

    // Налаштування додаткових параметрів камери (за бажанням)
    sensor_t *s = esp_camera_sensor_get();
    if (s) {
        s->set_vflip(s, 1); // Вертикальне відображення
        s->set_hmirror(s, 1); // Горизонтальне відображення
        s->set_brightness(s, 0); // Яскравість (-2 до 2)
        s->set_contrast(s, 0); // Контраст (-2 до 2)
        s->set_saturation(s, 0); // Насиченість (-2 до 2)
        s->set_gainceiling(s, GAINCEILING_8X); // Максимальне
посилення
    }
    return ESP_OK;
}

// Функція для встановлення інтенсивності вібрації
void setVibrationIntensity(uint8_t intensity) {
    // intensity: 0 (вимкнено) до 255 (максимальна
інтенсивність)
    ledcWrite(LED_CHANNEL, intensity);
    isVibrating = (intensity > 0);
}

// Функція для конвертації RGB565 в градації сірого
// Примітка: esp_camera_fb_get() повертає JPEG, який
потрібно декодувати в RGB888, а потім конвертувати в сірий
// Для простоти, якщо камера налаштована на
PIXFORMAT_RGB565, можна використовувати цю функцію
// Але оскільки ми використовуємо JPEG, нам потрібен інший
підхід.

```

```

// Ця функція є прикладом, якщо б ми працювали безпосередньо
з RGB565.
void convertRgb565ToGrayscale(uint8_t *src, uint8_t *dst,
int width, int height) {
    for (int i = 0; i < width * height; i++) {
        uint16_t pixel565 = (src[i * 2 + 1] << 8) | src[i *
2];

        uint8_t r = (pixel565 >> 11) & 0x1F;
        uint8_t g = (pixel565 >> 5) & 0x3F;
        uint8_t b = pixel565 & 0x1F;
        // Конвертація 5-6-5 біт у 8-бітний діапазон
        r = (r * 255) / 31;
        g = (g * 255) / 63;
        b = (b * 255) / 31;
        // Формула для градацій сірого (люмінесцентність)
        dst[i] = (uint8_t)(0.299 * r + 0.587 * g + 0.114 *
b);
    }
}

// Функція для простого виявлення країв (різницевий фільтр)
та бінаризації
// Працює на 8-бітних градаціях сірого
int processImageForObstacles(uint8_t *gray_image, int width,
int height) {
    int obstacle_pixels_count = 0;
    // Область інтересу - нижня частина зображення, де
ймовірність перешкод вища
    // Можна налаштувати ці значення для вашого сценарію
    int roi_start_row = height / 2; // Починаємо з середини
зображення
    int roi_end_row = height - 1;

    for (int y = roi_start_row; y < roi_end_row; y++) {
        for (int x = 1; x < width - 1; x++) { // Пропускаємо
перший/останній стовпець
            // Отримання значень яскравості сусідніх
пікселів

            uint8_t p_current = gray_image[y * width + x];
            uint8_t p_right = gray_image[y * width + (x +
1)];
            uint8_t p_down = gray_image[(y + 1) * width +
x];

```

```

        // Обчислення горизонтального та вертикального
градієнта (абсолютна різниця)
        int gx = abs(p_right - p_current);
        int gy = abs(p_down - p_current);

        // Магнітуда градієнта (проста сума)
        int gradient_magnitude = gx + gy;

        // Бінаризація: якщо градієнт достатньо великий,
це край
        if (gradient_magnitude > THRESHOLD_VALUE) {
            obstacle_pixels_count++;
        }
    }
}
return obstacle_pixels_count;
}

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(false); // Вимкнути дебаг вивід
ESP32-CAM, якщо він заважає

    // Налаштування пінів
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Кнопка з внутрішнім
підтягувальним резистором
    pinMode(LED_PIN, OUTPUT); // Світлодіод як вихід

    // Налаштування ШІМ для вібромотора
    ledcSetup(LEDC_CHANNEL, LEDC_BASE_FREQ, LEDC_TIMER_BIT);
    ledcAttachPin(VIBRATOR_PIN, LEDC_CHANNEL);

    // Ініціалізація камери
    if (initCamera() != ESP_OK) {
        // Якщо камера не ініціалізувалася, можна зациклитися
або повідомити про помилку
        while (true) {
            digitalWrite(LED_PIN, HIGH); // Індикація помилки
світлодіодом
            delay(100);
            digitalWrite(LED_PIN, LOW);
            delay(100);
        }
    }
}

```

```

Serial.println("Система готова.");
digitalWrite(LED_PIN, HIGH); // Світлодіод увімкнено -
система працює
}

void loop() {
// --- Обробка натискання кнопки ---
// Перевірка натискання кнопки (LOW, оскільки
INPUT_PULLUP)
if (digitalRead(BUTTON_PIN) == LOW && (millis() -
lastButtonPressTime) > debounceDelay) {
// Перемикання стану вібрації
if (isVibrating) {
setVibrationIntensity(0); // Вимкнути вібрацію
Serial.println("Вібрація вимкнена.");
} else {
setVibrationIntensity(150); // Увімкнути вібрацію з
середньою інтенсивністю
Serial.println("Вібрація увімкнена.");
}
lastButtonPressTime = millis(); // Оновити час
останнього натискання
}

// --- Захоплення та обробка зображення ---
camera_fb_t *fb = esp_camera_fb_get(); // Отримати буфер
кадру з камери
if (!fb) {
Serial.println("Помилка захоплення кадру з камери!");
digitalWrite(LED_PIN, LOW); // Індикація помилки
delay(500);
digitalWrite(LED_PIN, HIGH);
return;
}

// --- ЛОГІКА ОБРОБКИ ЗОБРАЖЕННЯ ТА ВИЗНАЧЕННЯ ПЕРЕШКОД --
-

uint8_t *gray_image_buffer = NULL;
int obstacle_pixels = 0;

// 1. Декодування JPEG в RGB888 та конвертація в градації
сірого
// ESP32-CAM захоплює JPEG, тому спочатку декодуємо його.
// Потім RGB888 конвертуємо в Y (градації сірого)
if (fb->format == PIXFORMAT_JPEG) {

```

```

    size_t rgb_buf_len = fb->width * fb->height * 3; //
    RGB888 має 3 байти на піксель
    uint8_t *rgb_image_buffer =
    (uint8_t*)malloc(rgb_buf_len);
    if (rgb_image_buffer) {
        fmt2rgb888(fb->buf, fb->len, RGB888,
rgb_image_buffer); // Декодуємо JPEG

        gray_image_buffer = (uint8_t*)malloc(fb->width * fb-
>height);
        if (gray_image_buffer) {
            for (int i = 0; i < fb->width * fb->height; i++)
            {
                uint8_t r = rgb_image_buffer[i * 3];
                uint8_t g = rgb_image_buffer[i * 3 + 1];
                uint8_t b = rgb_image_buffer[i * 3 + 2];
                // Конвертація RGB888 в градації сірого
                (люмінесцентність)
                gray_image_buffer[i] = (uint8_t)(0.299 * r +
0.587 * g + 0.114 * b);
            }
            // 2. Застосування алгоритмів виявлення країв та
бінаризація
            obstacle_pixels =
processImageForObstacles(gray_image_buffer, fb->width, fb-
>height);
            Serial.printf("Кількість пікселів перешкоди:
%d\n", obstacle_pixels);

            // 3. Визначення відстані та керування вібрацією
            if (obstacle_pixels > OBSTACLE_PIXEL_THRESHOLD)
            {
                // Чим більше пікселів, тим ближче перешкода
                або вона більша
                // Можна зробити більш точне керування
інтенсивністю
                uint8_t intensity = map(obstacle_pixels,
OBSTACLE_PIXEL_THRESHOLD, (fb->width * fb->height / 4), 50,
255); // Приклад мапування
                setVibrationIntensity(intensity);
                Serial.println("Перешкода виявлена! Вібрація
увімкнена.");
            } else {
                setVibrationIntensity(0); // Перешкоди не
виявлено

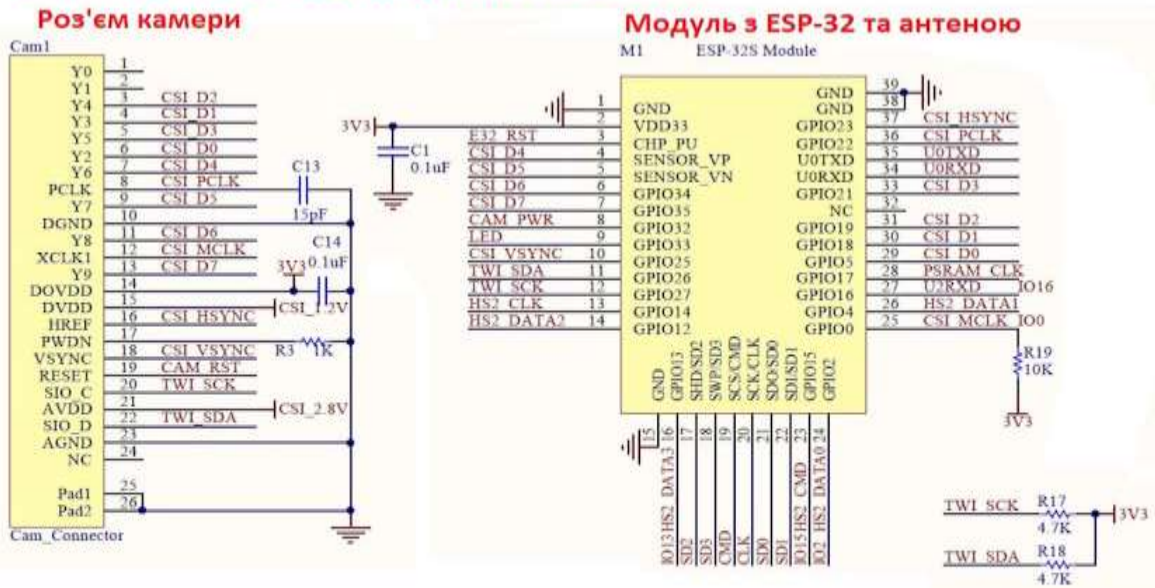
```

```
        }
    }
    free(rgb_image_buffer); // Звільнити буфер RGB
} else {
    Serial.println("Не вдалося виділити пам'ять для RGB
буфера!");
}
} else {
    Serial.println("Непідтримуваний формат пікселів
камери!");
}

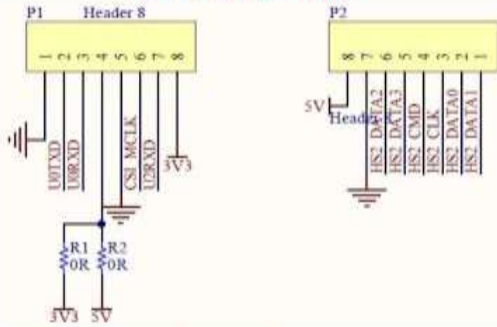
if (gray_image_buffer) {
    free(gray_image_buffer); // Звільнити буфер градацій
сірого
}

esp_camera_fb_return(fb); // Повернути буфер кадру
delay(100); // Невелика затримка, щоб уникнути
перевантаження
}
```

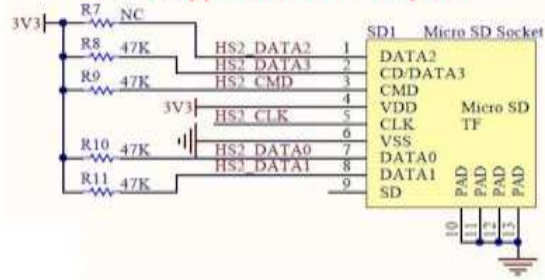
## Електрична схема модуля ESP32-CAM



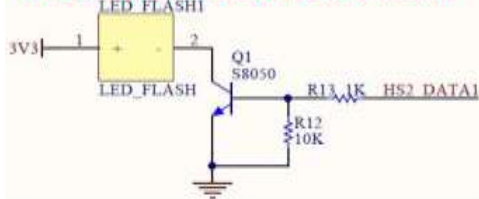
### Вихідні роз'єми



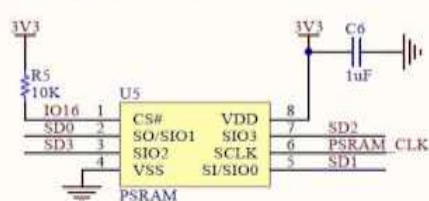
### Роз'єм для мікро-SD карти



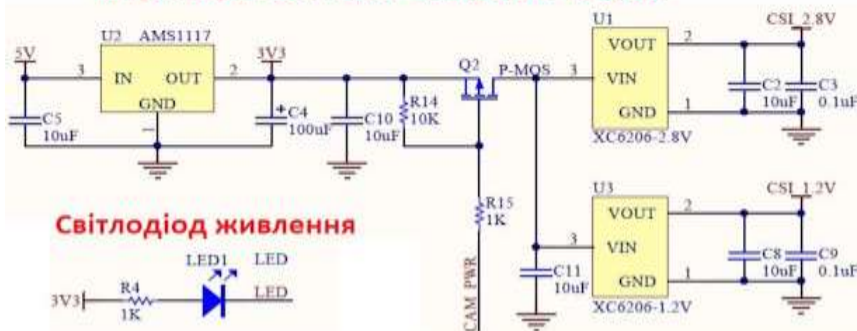
### Керування світлодіодом спалаху



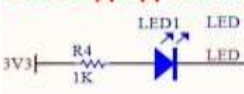
### Пам'ять PSRAM



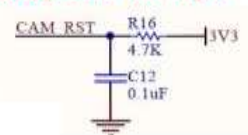
### Регулятори живлення 3,3 В, 2,8 В та 1,2 В



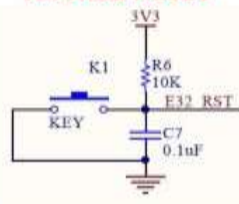
### Світлодіод живлення



### Скидання камери



### Кнопка RESET





selected microcontroller, the development of all necessary software for system operation, and the mathematical modeling of the core operational processes for optimization and verification.

During the project's execution, a **thorough analysis of existing solutions** in the field of machine vision and object detection algorithms was conducted. Particular attention was paid to the development and **optimization of simple yet highly effective mathematical algorithms** for image processing. This approach enabled stable and efficient system performance even when operating on a microcontroller with limited computational and energy resources, which is crucial for portable devices. The **ESP32-CAM** microcontroller was chosen as the hardware platform, its suitability confirmed by its compactness, integrated image processing capabilities, and overall accessibility. The device's mobility and extended autonomy are ensured by optimized battery power, including the implementation of effective **ESP32 sleep mode management strategies** (Light Sleep and Deep Sleep) to minimize power consumption.

**Experimental studies** and system testing under various lighting conditions have **unequivocally confirmed its stable and reliable operation**. To provide intuitive and discreet user feedback, a vibration motor was selected, proving to be an optimal method for tactile notification due to its high effectiveness and unobtrusiveness.

The **results** of this work hold **significant social importance**, as the developed device can substantially facilitate the daily lives of visually impaired individuals by providing them with an additional tool for more confident and safe spatial orientation. Furthermore, the presented system demonstrates considerable **potential for future scaling** and integration into a broader range of applications. Specifically, it can find widespread use in advanced fields such as robotics, autopilot systems, unmanned aerial vehicles, and other transportation systems, where rapid and accurate obstacle detection is a fundamental requirement. Further functional enhancement of the device

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Анк</i>
<i>ЗМН</i>	<i>Анк</i>	<i>№ локум</i>	<i>Пілпис</i>	<i>Дата</i>		77

envisages the **integration of artificial intelligence** elements or the development of more complex image analysis algorithms, which will significantly expand its capabilities and enhance the precision of obstacle detection.

					<i>ДП.ДЕ01. 910.505 000 ПЗ</i>	<i>Арк</i>
<i>ЗМН</i>	<i>Арк</i>	<i>№ локум</i>	<i>Пілпис</i>	<i>Дата</i>		<i>78</i>