

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

**Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«___» _____ 2026 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»
на тему: «Сенсорна система моніторингу та керування сміттєвим
контейнером на базі ESP32»**

Виконала:

студентка 4 курсу, групи KB-21

Оваденко Владислава Максимівна _____

Керівник:

ст. викл. каф. СПіСКС, д-р філос.

Радченко Костянтин Олександрович _____

Консультант з нормоконтролю:

доц. каф. СПіСКС, к.т.н., доц.

Клятченко Ярослав Михайлович _____

Рецензент:

ст. викл. каф. ОТ ФІОТ, д-р філос.,

Коренко Дмитро Володимирович _____

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

Студентка _____

Київ – 2026 року

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

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Освітньо-професійною програма

**«Системне програмування та спеціалізовані
комп'ютерні системи»**

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис)
“ ____ ” _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студента

Оваденко Владислави Максимівни

1. Тема проєкту «Сенсорна система моніторингу та керування сміттєвим контейнером на базі ESP32», керівник проєкту ст. викладач каф. СПіСКС, д-р філос. Радченко Костянтин Олександрович, затверджена наказом по університету від «28» травня 2026 р. № 1984-С
2. Термін подання студентом проєкту _____
3. Вихідні дані до проєкту: див. Технічне завдання
4. Зміст пояснювальної записки:
 - 1) аналіз предметної області та огляд існуючих рішень;
 - 2) апаратне забезпечення та проєктування розумного смітника;
 - 3) програмна реалізація та розробка алгоритмів керування;
 - 4) розробка та інтеграція моделі машинного навчання;
 - 5) тестування системи та оцінка результатів.
5. Перелік графічного матеріалу:
 - 1) схема електрична структурна;
 - 2) схема електрична принципова;
 - 3) схема структурна програмної частини;
 - 4) схема алгоритму.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я. М., доцент кафедри СПіСКС, к.т.н., доцент		

7. Дата видачі завдання «__» _____ 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення літератури за тематикою роботи	15.11.2025	
2	Розроблення та узгодження технічного завдання	27.11.2025	
3	Проектування апаратного забезпечення, розробка системи живлення та механізму сортування.	28.12.2025	
4	Розробка програмної частини, алгоритмів керування сервоприводами та локального вебсервера.	05.02.2026	
5	Формування набору даних та навчання моделі машинного навчання.	15.03.2026	
6	Інтеграція ШІ-моделі у мікроконтролер та фізична збірка діючого прототипу.	20.04.2026	
7	Комплексне тестування роботи системи, аналіз швидкодії та оцінка результатів.	21.05.2026	
8	Підготовка матеріалів першого розділу дипломного проєкту	25.05.2026	
9	Підготовка матеріалів другого розділу дипломного проєкту	29.05.2026	
10	Підготовка матеріалів третього розділу дипломного проєкту	01.06.2026	
11	Підготовка матеріалів четвертого та п'ятого розділів дипломного проєкту	02.06.2026	
12	Оформлення технічної документації та графічної частини	03.06.2026	
13	Попередній огляд матеріалів дипломної роботи на кафедрі	04.06.2026	

Студентка

Владислава ОВАДЕНКО

Керівник проєкту

Костянтин РАДЧЕНКО

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (54 с., 11 рис., 2 табл., список використаної літератури з 15 найменувань, 3 додатки).

Мета роботи: розробка та практична реалізація повноцінного прототипу розумного смітника на базі мікроконтролера ESP32, який здатний автоматично розпізнавати тип викинутого предмета за допомогою машинного зору та механічно сортувати його у відповідний відсік без допомоги людини..

Методи та технології: в основі обчислювальної частини лежить двоядерний мікроконтролер ESP32-CAM. Програмна частина реалізована мовою C++. Для розпізнавання об'єктів застосовано методи граничних обчислень (TinyML) та згорткову нейромережу архітектури FOMO з 8-бітною пост-тренувальною квантизацією. Для зв'язку мікроконтролера з контролером сервоприводів використано апаратний цифрову шину I2C.

Реалізація та результати: розроблено та зібрано діючий прототип пристрою. Він включає мікроконтролер ESP32-CAM з оптичним сенсором OV2640, ШІМ-контролер PCA9685 та потужні сервоприводи MG996R, що утворюють механізм поворотного скидача (pan-tilt). Використано нейронну мережу, яка класифікує відходи із середньою точністю 75%. У пристрої реалізовано автоматичну логіку сортування сміття на три категорії (папір, пластик, метал) та локальний вебсервер для віддаленого моніторингу процесу.

Практичне значення: отримано робочу модель програмно-апаратного комплексу для розумної утилізації відходів у житлових приміщеннях або офісах. Пристрій переносить відповідальність за прийняття рішень на машину, що повністю нівелює людський фактор і дозволяє кардинально підвищити відсоток чистої вторинної сировини.

Ключові слова: програмно-апаратний комплекс, розумний смітник, мікроконтролер ESP32-CAM, сортування відходів, комп'ютерний зір, машинне навчання, TinyML, Edge Impulse, ШІМ-контролер, C++.

ANNOTATION

The bachelor's thesis includes an explanatory note (54 pages, 11 figures, 2 tables, a list of references with 15 items, 3 appendices).

Objective of the work: development of a smart trash bin prototype based on the ESP32 microcontroller, capable of automatically recognizing and mechanically sorting discarded objects into appropriate compartments using machine vision.

Methods and technologies: the system is based on the dual-core ESP32-CAM microcontroller and programmed in C++. Object recognition utilizes edge computing (TinyML) and a FOMO architecture convolutional neural network with 8-bit quantization. An I2C bus provides communication with the servo controller.

Implementation and results: a working prototype was assembled, featuring an ESP32-CAM with an OV2640 sensor, a PCA9685 PWM controller, and MG996R servos for the pan-tilt mechanism. The neural network classifies waste into three categories (paper, plastic, metal) with an average accuracy of 75%. Additionally, a local web server enables remote process monitoring.

Practical significance: the developed hardware-software complex offers a practical solution for automated waste disposal in residential or office environments. By shifting decision-making to a machine, it eliminates the human factor and significantly increases the yield of clean recyclable materials.

Keywords: hardware-software complex, smart trash bin, ESP32-CAM microcontroller, waste sorting, computer vision, machine learning, TinyML, Edge Impulse, PWM controller, C++.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки	
	A4	ІАЛЦ.466120.002 ТЗ	Сенсорна система моніторингу та керування сміттєвим контейнером				
			Технічне завдання				
	A4	ІАЛЦ.466120.003 ТП	Сенсорна система моніторингу та керування сміттєвим контейнером				
			Відомість технічного проєкту				
	A4	ІАЛЦ.466120.004 ПЗ	Сенсорна система моніторингу та керування сміттєвим контейнером				
			Пояснювальна записка				
	A4	ІАЛЦ.466120.005 Е1	Сенсорна система моніторингу та керування сміттєвим контейнером				
			Схема електрична структурна				
			ІАЛЦ.466120.001 ОА Сенсорна система моніторингу та керування сміттєвим контейнером на базі ESP32 Опис альбому				
Змін	Арк.	№ докум.	Підпис	Дата	Літ.	Аркуш	Аркушів
Розробив		Оваденко В. М.					
Перевірив		Радченко К. О.					
Консуьлт.							
Н. контроль		Клятченко Я.М.					
Зав. каф.		Романкевич В.О.					
					КПІ ім. Ігоря Сікорського, ФПСПМ KB-21		

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
A4	IАЛЦ.466120.006 ЕЗ	Сенсорна система моніторингу та керування сміттєвим контейнером	Схема електрична принципова			
A4	IАЛЦ.466120.007 Д1	Сенсорна система моніторингу та керування сміттєвим контейнером	Схема структурна програмної частини	1		
A4	IАЛЦ.466120.008 Д2	Сенсорна система моніторингу та керування сміттєвим контейнером	Схема структурна програмної частини			
	Dиск CD-ROM	Текст пояснювальної записки. Графічний матеріал				

Змін	Арк.	№ докум.	Підпис	Дата	IАЛЦ.466120.001 ОА	Арк.

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3.	МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.....	2
4.	ДЖЕРЕЛА РОЗРОБКИ	2
5.	ТЕХНІЧНІ ВИМОГИ	3
6.	ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.466120.002 ТЗ			
Зм	Лист	№ докум.	Підп.	Дата				
Розроб.		Оваденко В. М.			Сенсорна система моніторингу та керування сміттєвим контейнером на базі Технічне завдання	Літ.	Аркуш	Аркушів
Перев.		Радченко К. О.						4
Н. контр.		Клятченко Я. М.				КПІ ім. Ігоря Сікорського, ФПСІМ, КВ-21		
Зав. Каф.		Романкевич В. О.						

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Сенсорна система моніторингу та керування сміттєвим контейнером на базі ESP32».

Галузь застосування: екологія та розумна утилізація відходів, системи розумного дому, розробка вбудованих систем (embedded systems) та граничних обчислень зі штучним інтелектом (Edge AI) для підвищення ефективності переробки вторинної сировини.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проєктування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є розробка та практична реалізація повноцінного прототипу розумного смітника.

Призначення системи полягає у вирішенні проблеми неефективного ручного сортування відходів на побутовому рівні, що спричинено людським фактором, браком часу та відсутністю зручної інфраструктури. Пристрій призначений для повної автоматизації процесу сортування. Він має автоматично, без жодної допомоги людини, розпізнавати тип викинутого предмета за допомогою комп'ютерного зору та механічно сортувати його у відповідний внутрішній відсік (для пластику, паперу або металу). Додатково система має забезпечувати гігієнічну безконтактну взаємодію з користувачем та підтримувати тривалий час автономної роботи від вбудованих акумуляторів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є наукові публікації та аналітичні звіти щодо глобальної проблеми генерації сміття, зокрема звіти Світового банку та статті в

					<i>ІАЛЦ.466120.002 ТЗ</i>	Арк.
Зм	Лист	№ докум.	Підп.	Дата		2

журналі Waste Management. Також використовувалася офіційна технічна документація на мікроконтролери сімейства ESP32 (зокрема ESP32-CAM), сенсор камери OV3660, ШІМ-контролер PCA9685. Для розробки програмного забезпечення та моделі машинного навчання джерелом слугувала офіційна документація мови C++, середовища Edge Impulse (для алгоритмів TinyML та архітектури FOMO), а також специфікації підключених бібліотек (esp_camera.h, Wire.h, Adafruit_PWM_Servo_Driver.h, WiFi.h).

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту:

- ПЗ на основі архітектури кінцевого автомата.
- Локальна ШІ-модель із порогом впевненості від 70%.
- Вбудований вебсервер для моніторингу та відлагодження.
- Режим енергозбереження (Deep Sleep).

5.2 Вимоги до апаратного забезпечення:

- Мікроконтролер: ESP32-S3 або ESP32-CAM (із PSRAM).
- Система зору: камера OV3660 з LED-підсвіткою.
- Механіка: ШІМ-контролер PCA9685 та сервоприводи MG996R.
- Живлення: акумулятори 18650 (з BMS та DC-DC перетворювачами).

5.3 Вимоги до забезпечення користувача:

- Автономне розпізнавання об'єктів без доступу до інтернету.
- Автоматичне переміщення об'єкту у відповідний відділ.
- Робота від акумулятора не менше кількох діб (при 30-50 циклах/день).

					ІАЛЦ.466120.002 ТЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		3

6.ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1	Вивчення літератури за тематикою роботи	15.11.2025
2	Розроблення та узгодження технічного завдання	27.11.2025
3	Проектування апаратного забезпечення, розробка системи живлення та механізму сортування.	28.12.2025
4	Розробка програмної частини, алгоритмів керування сервоприводами та локального вебсервера.	05.02.2026
5	Формування набору даних та навчання моделі машинного навчання.	15.03.2026
6	Інтеграція ШІ-моделі у мікроконтролер та фізична збірка діючого прототипу.	20.04.2026
7	Комплексне тестування роботи системи, аналіз швидкодії та оцінка результатів.	21.05.2026
8	Підготовка матеріалів першого розділу дипломного проекту	25.05.2026
9	Підготовка матеріалів другого розділу дипломного проекту	29.05.2026
10	Підготовка матеріалів третього розділу дипломного проекту	01.06.2026
11	Підготовка матеріалів четвертого розділу дипломного проекту	02.06.2026
12	Оформлення технічної документації та графічної частини	03.06.2026
13	Попередній огляд матеріалів дипломної роботи на кафедрі	04.06.2026

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість	№	Примітки
	A4	ІАЛЦ.466120.004 ПЗ	Сенсорна система моніторингу та керування сміттєвим контейнером на базі ESP32.			
			Пояснювальна записка			
	A4	ІАЛЦ.466120.005 Е1	Схема електрична структурна			
	A4	ІАЛЦ.466120.006 ЕЗ	Схема електрична принципова	1		
	A4	ІАЛЦ.466120.007 Д1	Схема структурна програмної частини			
	A4	ІАЛЦ.466120.008 Д2	Схема алгоритму			
		Диск CD-ROM	Текст пояснювальної записки.			
			Графічний матеріал			
					ІАЛЦ. 466120.003 ТП	
Змін	Арк.	№ докум.	Підпис	Дата		
Розробив		Оваденко В. М.			Літ.	Аркуш Аркушів
Перевірив		Радченко К. О.				
Консуьлт.						
Н. контроль		Клятченко Я.М.			КПП ім. Ігоря Сікорського, ФПСІМ KB-21	
Зав. каф.		Романкевич В.О.				

Пояснювальна записка
до дипломного проєкту
на тему: «Сенсорна система моніторингу та керування сміттєвим контейнером
на базі ESP32»

Київ – 2026

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	7
1.1. Проблематика сортування відходів у сучасному світі.....	7
1.2. Огляд наявних систем автоматичного сортування.....	8
1.3. Аналіз методів машинного навчання для мікроконтролерів.....	10
1.4. Аналіз апаратних платформ для систем розумного дому.....	11
1.5. Формування вимог до системи	13
1.6. Висновки до розділу 1	14
РОЗДІЛ 2. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ТА ПРОЄКТУВАННЯ РОЗУМНОГО СМІТНИКА.....	15
2.1. Апаратні характеристики базового мікроконтролера та камери	15
2.2. Проєктування системи керування сервоприводами	16
2.3. Розробка підсистеми живлення пристрою	19
2.4. Проєктування корпусу та механізму сортування відходів	21
2.5. Розробка принципової електричної схеми	24
2.6. Висновки до розділу 2	26
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА АЛГОРИТМІВ КЕРУВАННЯ.....	28
3.1. Вибір середовища розробки та програмних бібліотек.....	28
3.2. Розробка алгоритму ініціалізації периферійних пристроїв	29
3.3. Програмна реалізація вебсервера для віддаленого моніторингу	31
3.4. Розробка логіки керування сервоприводами через інтерфейс I2C	33

					<i>ІАЛЦ. 466120.004 ПЗ</i>			
Змін.	Арк.	№ докум.	Підпис	Дата	Сенсорна система моніторингу та керування сміттєвим контейнером на базі ESP32 Пояснювальня записка	Літ.	Аркуш	Аркушів
Розробив		Оваденко В.М.					1	54
Перевірив		Радченко К.О.						
Консульт.						КПІ		
Н. контроль		Клятченко Я. М.				Ім. Ігоря Сікорського, ФПСІМ КВ-21		
Зав. каф.		Романкевич В.О.						

3.5. Оптимізація процесу захоплення зображення	35
3.6. Висновки до розділу 3	37
РОЗДІЛ 4. РОЗРОБКА ТА ІНТЕГРАЦІЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ	38
4.1. Вибір платформи для створення моделі машинного навчання.....	38
4.2. Збір та підготовка набору даних для класифікації сміття	39
4.3. Навчання нейронної мережі для розпізнавання об'єктів	40
4.4. Інтеграція навченої моделі в мікроконтролер.....	42
4.5. Обробка результатів розпізнавання та прийняття рішень	43
4.6. Висновки до розділу 4	44
РОЗДІЛ 5. ТЕСТУВАННЯ СИСТЕМИ ТА ОЦІНКА РЕЗУЛЬТАТІВ.....	46
5.1. Методика проведення тестування апаратної та програмної частин.....	46
5.2. Оцінка точності розпізнавання різних типів відходів.....	47
5.3. Перевірка надійності механізму фізичного сортування	49
5.4. Аналіз швидкодії системи та споживання ресурсів	49
5.5. Оцінка загальної ефективності та шляхи вдосконалення.....	51
5.6. Висновки до розділу 5	52
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТОК А.....	
ДОДАТОК Б	

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

АПІ (API) — Application Programming Interface (Інтерфейс програмування додатків).

БД — База даних.

ЗНМ (CNN) — Згорткова нейронна мережа (Convolutional Neural Network).

МК — Мікроконтролер.

ПЗ — Програмне забезпечення.

ТПВ — Тверді побутові відходи.

ІІІ — Штучний інтелект.

ШИМ (PWM) — Широтно-імпульсна модуляція (Pulse-Width Modulation).

Edge AI — Периферійний штучний інтелект (штучний інтелект на периферійних пристроях).

GPIO — General-Purpose Input/Output (інтерфейс введення/виведення загального призначення).

I2C — Inter-Integrated Circuit (послідовна асиметрична шина зв'язку).

IoT — Internet of Things (Інтернет речей).

PSRAM — Pseudostatic Random Access Memory (псевдостатична оперативна пам'ять).

SRAM — Static Random Access Memory (статична оперативна пам'ять).

UART — Universal Asynchronous Receiver-Transmitter (універсальний асинхронний приймач-передавач).

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

3

ВСТУП

Проблема накопичення твердих побутових відходів щороку набуває все більш загрозливих масштабів як на глобальному, так і на локальному рівнях. Традиційні підходи до утилізації, такі як захоронення на полігонах або спалювання, вже давно довели свою неефективність та небезпеку, оскільки вони призводять до незворотних змін у навколишньому середовищі, масштабного забруднення ґрунтових вод важкими металами та колосальних викидів парникових газів. Єдиним стійким та екологічно виправданим шляхом боротьби із цією кризою є глибока переробка вторинної сировини. Проте ефективна переробка фізично неможлива без якісного попереднього сортування сміття безпосередньо в місцях його утворення.

На жаль, ручне сортування відходів на побутовому рівні (у квартирах, офісах чи навчальних закладах) залишається вкрай неефективним. Це пояснюється людським фактором, постійним браком часу, небажанням заглиблюватися в складну систему маркування пластику та просто відсутністю зручної і зрозумілої інфраструктури. Люди часто помиляються, викидаючи жирний папір або непереробний пластик у контейнери для чистої сировини, що псує цілі партії потенційно корисних матеріалів. У контексті цих викликів повна автоматизація процесів сортування за допомогою сучасних технологій Інтернету речей (IoT) та граничних обчислень зі штучним інтелектом стає не просто елементом преміального комфорту, а гострою екологічною необхідністю.

Дослідження в рамках цього проєкту зосереджені на вивченні та практичному застосуванні методів комп'ютерного зору для точної класифікації типових побутових відходів, а також на принципах побудови надійних і автономних мікроконтролерних систем, здатних працювати в умовах жорстких обмежень обчислювальної потужності та енергоспоживання.

Головною метою роботи є розробка та практична реалізація повноцінного прототипу розумного смітника. Цей пристрій має бути здатним автоматично, без жодної допомоги людини, розпізнавати тип викинутого предмета, механічно

сортувати його у відповідний внутрішній відсік та забезпечувати максимально комфортну, гігієнічну безконтактну взаємодію з користувачем, зберігаючи при цьому тривалий час автономної роботи від вбудованих акумуляторів.

Для досягнення поставленої мети було визначено наступний комплекс завдань:

- детально проаналізувати поточний стан глобальної проблеми сортування відходів та здійснити критичний огляд існуючих на ринку комерційних, а також наукових рішень у сфері розумної утилізації;
- дослідити сучасні архітектури згорткових нейронних мереж та методи машинного навчання, які адаптовані для безпосереднього використання на малопотужних мікроконтролерах;
- розробити та спроектувати оптимальну апаратну частину пристрою, яка включатиме підсистему машинного зору, високоточні датчики вимірювання часу польоту світла, сервоприводи для механічного переміщення та надійну підсистему автономного живлення на базі літій-іонних акумуляторів;
- створити програмне забезпечення на основі архітектури кінцевого автомата, яке безшовно інтегруватиме навчену ШІ-модель, забезпечить опитування всіх датчиків у реальному часі та точне позиціонування механізмів;
- впровадити та налаштувати алгоритми глибокого енергозбереження мікроконтролера з метою максимізації часу безперебійної роботи пристрою від одного заряду батареї;
- провести комплексне експериментальне тестування зібраного прототипу в умовах, максимально наближених до реальних, для оцінки точності розпізнавання різних матеріалів, загальної стабільності механіки та показників енергоефективності.

Практичне значення отриманих результатів полягає у створенні готового до використання, компактного та автономного програмно-апаратного комплексу. Такий розумний смітник може бути з легкістю встановлений у будь-

якому житловому приміщенні чи сучасному офісі. Широке впровадження подібних пристроїв дозволить кардинально підвищити відсоток правильно відсортованої вторинної сировини, суттєво зменшити навантаження на комунальні служби і сміттесортувальні станції, а також зробити щоденний рутинний процес утилізації відходів гігієнічним і непомітним. Крім того, напрацьовані під час виконання проекту алгоритми енергозбереження та техніки оптимізації нейромереж для мікроконтролерів сімейства ESP32 мають універсальний характер і можуть бути успішно використані в широкому спектрі інших систем розумного дому.

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Проблематика сортування відходів у сучасному світі

Стрімке зростання населення планети та рівня споживання товарів призводять до безпрецедентного збільшення обсягів генерації сміття. За даними дослідження Kaza та співавторів, опублікованого Світовим банком у 2018 році під назвою "What a Waste 2.0" [1], людство щорічно продукує понад 2 мільярди тонн твердих побутових відходів. За відсутності радикальних змін ця цифра до 2050 року може зрости ще на 70 відсотків. При цьому значна частина відходів потрапляє на несанкціоновані звалища або спалюється, що завдає шкоди екосистемам, як показано на рисунку 1.1.

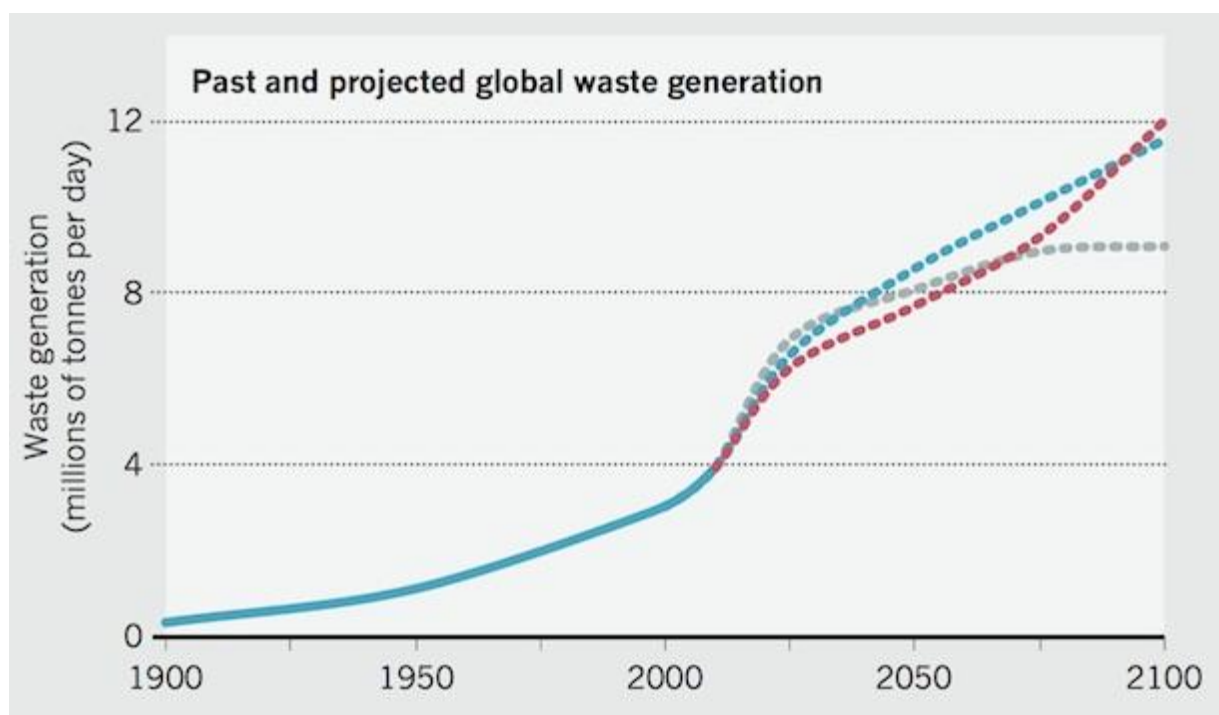


Рисунок 1.1 – Динаміка зростання обсягів твердих побутових відходів у світі

Наукові публікації, зокрема аналітичні статті у виданні Waste Management (наприклад, стаття "Solid waste management in the world's cities"), наголошують на тому, що найважливішим етапом у ланцюжку переробки є первинне сортування [2]. Коли папір, пластик та органічні залишки змішуються в одному

контейнері ще на етапі викидання, відбувається взаємне забруднення. Наприклад, картон, просочений вологою з харчових відходів, втрачає свою цінність як вторинна сировина. Це означає, що ефективно розділяти сміття необхідно безпосередньо в момент його утворення.

На практиці впровадження систем роздільного збору стикається з низкою перешкод. Багато людей ігнорують правила сортування через брак часу або небажання розбиратися в маркуваннях пластику. Звичайні побутові контейнери не можуть виправити помилку користувача.

Тому виникає потреба в автоматизації цього процесу. Перенесення відповідальності за прийняття рішення про категорію відходів на пристрій дозволяє нівелювати вплив людського фактору. Розумні сенсорні системи можуть аналізувати зовнішній вигляд предмета і направляти його у відповідний відсік.

1.2. Огляд наявних систем автоматичного сортування

Концепція створення розумного смітника вже частково реалізована в деяких продуктах та наукових дослідженнях. Їх можна розділити на комерційні пристрої для широкого споживача та прототипи, розроблені в лабораторіях.



Рисунок 1.2 – Приклади існуючих комерційних систем сортування сміття

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

На споживчому ринку популярними є пристрої, такі як смітники серії Townnew [3], які наведено на рисунку 1.2. Їх функціонал обмежується інфрачервоним датчиком для безконтактного відкриття кришки та механізмом, який автоматично запаює пластиковий пакет. Це забезпечує високий рівень гігієни, проте такі пристрої не вирішують проблему екології, оскільки все сміття накопичується в одному об'ємі.

Більш складним комерційним рішенням є система Trashbot від компанії CleanRobotics [4]. Цей пристрій використовує комбінацію камер, датчиків ваги та металошукачів для класифікації відходів, як показано на рисунку 1.3.



Рисунок 1.3 – Зовнішній вигляд системи Trashbot

Завдяки хмарним обчисленням заявлена точність розпізнавання становить понад 90 відсотків. Однак габарити Trashbot роблять його придатним переважно для великих торгових центрів чи офісів. Вартість такого комплексу становить тисячі доларів, а архітектура вимагає постійного підключення до електромережі.

У науковій площині також існує багато досліджень. У статті "Artificial Intelligence based Smart Dustbin", опублікованій Sharma та співавторами у 2020 році в International Journal of Innovative Technology and Exploring Engineering [5], описується прототип на базі мікрокомп'ютера Raspberry Pi та веб-камери. Цей підхід демонструє високу точність, проте Raspberry Pi вимагає використання операційної системи Linux, через що плата споживає занадто багато енергії. Інші роботи часто використовують хмарні сервери для розпізнавання фотографій, що створює залежність від інтернету.

Аналіз цих рішень свідчить про наявність вільної ніші для створення компактного, автономного розумного смітника. Він має локально розпізнавати відходи, мати низьке енергоспоживання та бути доступним для масового впровадження.

1.3. Аналіз методів машинного навчання для мікроконтролерів

Для того, щоб система могла самостійно відрізнити пластик від паперу, необхідно застосувати методи комп'ютерного зору. Класичні алгоритми виділення ознак показують низьку ефективність у побутових умовах, коли предмети зім'яті або освітлення є недостатнім. Найкращі результати у задачах класифікації зображень демонструють згорткові нейронні мережі.

Проте класичні архітектури, такі як VGG16 або ResNet, вимагають значних обчислювальних ресурсів. Запустити таку модель на мікроконтролері неможливо.

Для задач граничних обчислень розробляються спеціально оптимізовані легковикові архітектури, зокрема сімейство MobileNet [6]. Головною особливістю цієї архітектури є використання глибинних поглинаючих згорток. Цей метод дозволяє зменшити загальну кількість математичних параметрів та

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		10

обчислень майже у вісім разів порівняно зі звичайними згортками, зі збереженням прийнятної точності розпізнавання.

Окрім вибору легкої архітектури, необхідним кроком є квантування моделі. Ця процедура передбачає перетворення ваг нейромережі з формату з рухомою комою у цілочисельний формат. Таке перетворення в чотири рази зменшує розмір моделі в пам'яті та прискорює обчислення. Мікроконтролерам значно легше виконувати операції з цілими числами.

Для розгортання моделей на мікроконтролерах використовується бібліотека TensorFlow Lite for Microcontrollers [7]. Цей інструмент дозволяє конвертувати навчену нейромережу у масив байтів мовою програмування C. Застосування цих методів робить можливим локальне розпізнавання предметів на малопотужних пристроях.

1.4. Аналіз апаратних платформ для систем розумного дому

Обрання правильної мікроконтролерної платформи є критично важливим етапом. Для розробки розумного смітника було розглянуто кілька популярних мікроконтролерів.

Платформи на базі 8-бітних мікроконтролерів архітектури AVR, такі як плати Arduino Uno, є доступними і підходять для простих задач керування моторами. Однак їх низька тактова частота та малий обсяг оперативної пам'яті унеможливають запуск нейронних мереж або обробку зображень з камери.

Мікрокомп'ютери сімейства Raspberry Pi мають великі обчислювальні ресурси та багатоядерні процесори. Запуск нейромережі на них відбувається швидко. Але вони вимагають повноцінної операційної системи Linux, що призводить до постійного високого споживання енергії. Автономна робота від акумуляторів у такому режимі є неефективною для побутового пристрою.

Мікроконтролери серії STM32 з ядром ARM Cortex-M надають хороші можливості, проте старші моделі, здатні обробляти зображення, є дорогими. Крім того, розробка під них є більш складною, а необхідність зовнішньої пам'яті ускладнює схемотехніку.

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		11

Оптимальним компромісом між обчислювальною потужністю та енергоспоживанням є сімейство мікроконтролерів ESP32. Ці чипи мають потужну двоядерну архітектуру Xtensa, підтримку зовнішньої пам'яті та модулі бездротового зв'язку. Особливу увагу привертає модуль ESP32-CAM [8], який наведено на рисунку 1.4.



Рисунок 1.4 – Мікроконтролерний модуль ESP32-CAM з камерою OV2640

Модуль ESP32-CAM об'єднує в собі продуктивний процесор ESP32, достатній обсяг зовнішньої оперативної пам'яті та інтегрований оптичний сенсор OV2640. Це дозволяє використовувати лише одну плату для вирішення всіх завдань проєкту, що стосуються отримання зображення, його обробки, запуску нейронної мережі та управління периферією. Використання єдиного мікроконтролера ESP32-CAM значно спрощує архітектуру пристрою, зменшує його габарити та кінцеву вартість, а також забезпечує можливість переведення системи у режим сну для економії енергії акумуляторів.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
12

1.5. Формування вимог до системи

На основі аналізу проблематики та апаратних платформ, було сформовано перелік технічних та функціональних вимог до пристрою.

Функціональні вимоги до взаємодії з користувачем:

- пристрій повинен локально аналізувати зображення предмета за допомогою вбудованої камери і відносити його до однієї з відповідних категорій (наприклад: пластик, папір, метал);
- направлення предмета у відповідний відсік накопичувача повинно відбуватися повністю автоматично за допомогою електромеханічного механізму розподілу;
- система має надавати можливість моніторингу процесу сортування та перегляду зображення з камери через локальний вебінтерфейс.

Вимоги до апаратної та програмної архітектури:

- головним і єдиним обчислювальним пристроєм має виступати мікроконтролерний модуль ESP32-CAM;
- модуль повинен здійснювати безперервне циклічне захоплення зображень, обробку даних класифікаційною нейромережею (Edge Impulse) та координацію периферії;
- для апаратного керування сервоприводами розподільвача необхідно використовувати зовнішній ШІМ-контролер, підключений по цифровій шині I2C (для уникнення конфліктів апаратних таймерів);
- механізм просторового розподілення (поворот та скидання) має приводитись у рух потужними сервоприводами з металевими редукторами;
- живлення системи повинно забезпечуватися через розподілену мережу з ізольованими понижуючими перетворювачами напруги DC-DC для логічної та силової частин.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

1.6. Висновки до розділу 1

У ході аналізу предметної області підтверджено актуальність проблеми сортування відходів та необхідність автоматизації цього процесу на побутовому рівні. Огляд існуючих систем виявив відсутність компактних рішень, здатних локально розпізнавати сміття без підключення до хмарних сервісів.

Досліджено алгоритми машинного навчання та встановлено, що використання оптимізованих згорткових нейромереж з квантуванням ваг дозволяє вирішувати задачі класифікації на вбудованих пристроях. Найбільш придатним варіантом апаратної платформи обрано модуль ESP32-CAM, який поєднує необхідну обчислювальну потужність для роботи з нейромережами та можливість інтеграції камери в рамках одного чипу, що дозволяє відмовитись від додаткових контролерів.

Сформовано перелік технічних та функціональних вимог до розумного смітника. Ці вимоги визначають архітектуру системи на базі єдиного мікроконтролера ESP32-CAM, перелік необхідних датчиків, механізмів та логіку роботи, які будуть розроблені у наступних розділах.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
14

РОЗДІЛ 2. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ТА ПРОЄКТУВАННЯ РОЗУМНОГО СМІТНИКА

Переходячи від аналізу існуючих систем та формування загальних вимог до безпосереднього проєктування пристрою, першочерговим завданням є вибір оптимальної апаратної платформи. Апаратна частина є фундаментом усього проєкту, оскільки вона визначає обчислювальні можливості для алгоритмів машинного навчання, надійність механічних вузлів під час сортування відходів та загальну стабільність роботи системи. У цьому розділі буде детально розглянуто процес підбору кожного компонента, проєктування системи живлення, розробку механічної конструкції сортувального лотка, а також побудову структурної та електричної принципової схем.

2.1. Апаратні характеристики базового мікроконтролера та камери

Як було обґрунтовано у попередньому розділі, базовим мікроконтролером системи обрано модуль ESP32-CAM. Він базується на потужному двоядерному процесорі Xtensa 32-bit LX6, який працює на тактовій частоті до 240 мегагерц. Наявність двох обчислювальних ядер є надзвичайно важливою перевагою для реалізованого програмного забезпечення. Вона дозволяє реалізувати паралельну обробку завдань: одне ядро ефективно обслуговує бездротове з'єднання Wi-Fi та вбудований веб-сервер для відеотрансляції в режимі реального часу, тоді як друге ядро займається виключно захопленням кадрів з камери та виконанням складних математичних операцій класифікації об'єктів за допомогою бібліотеки Edge Impulse. Завдяки цьому забезпечується безперебійний логічний висновок без затримок у роботі веб-інтерфейсу.

Критично важливою характеристикою модуля ESP32-CAM є наявність додаткової мікросхеми оперативної пам'яті (PSRAM) об'ємом 4 мегабайти. Стандартної вбудованої пам'яті SRAM мікроконтролера абсолютно недостатньо для одночасного зберігання буфера кольорового зображення, змінних нейронної мережі та підтримки роботи операційної системи. Використання зовнішньої

PSRAM дозволяє безперешкодно захоплювати кадри з камери, зберігати їх та перетворювати у формат RGB888 (адаптовано під вимоги моделі машинного навчання). Крім того, вбудований модуль бездротового зв'язку дозволив реалізувати вебінтерфейс, який запущено на 80-му порту для зручного відлагодження, налаштування кута огляду та моніторингу процесу розпізнавання.

Як сенсор зображення для модуля використовується камера OV2640. Згідно з технічною документацією від компанії OmniVision [8], сенсор OV2640 базується на технології CMOS і забезпечує роздільну здатність 2 мегапікселі. Він відзначається достатнім динамічним діапазоном, що гарантує стабільну світлочутливість. Для задач комп'ютерного зору, особливо всередині замкненого простору, де рівень освітлення може бути сталим, якість захопленого кадру відіграє вирішальну роль у точності розпізнавання відходів. Модуль камери підключається до мікроконтролера через стандартний 24-контактний гнучкий шлейф і підтримує апаратне стиснення зображень у формат JPEG безпосередньо на кристалі сенсора. Згідно з дослідженнями, опублікованими у статті "Performance Evaluation of ESP32-CAM for IoT Edge Computing" [9], використання апаратного стиснення JPEG дозволяє збільшити пропускну здатність відеопотоку у 3 рази при збереженні прийнятної якості для нейромереж, суттєво зменшуючи навантаження на системну шину мікроконтролера.

2.2. Проектування системи керування сервоприводами

Процес автоматичного сортування відходів вимагає фізичного переміщення сміття у відповідний відсік (для паперу, пластику або металу). Для реалізації цього механізму необхідне використання сервоприводів — електродвигунів із зворотним зв'язком, які здатні точно повертати вал на заданий кут. Проте підключення сервоприводів безпосередньо до виводів ESP32-CAM пов'язане з кількома суттєвими проблемами.

По-перше, модуль ESP32-CAM страждає від значної нестачі вільних контактів вводу/виводу (GPIO). Більшість пінів мікроконтролера вже задіяні для паралельної шини камери та мікросхеми PSRAM. Використання тих небагатьох вільних контактів, що залишилися, для програмної генерації широтно-імпульсної модуляції (ШИМ) може призвести до конфліктів апаратних переривань. У багатозадачній системі з активним Wi-Fi та високим навантаженням на процесор програмний ШИМ стає нестабільним. Зміна тривалості керуючого імпульсу навіть на кілька мікросекунд викликає тремтіння сервоприводів ("джитер") та неточне позиціонування механізму сортування, що є неприпустимим для надійної роботи.

Для вирішення цієї проблеми було прийнято рішення використати зовнішній контролер ШИМ-сигналів на базі мікросхеми PCA9685 [10]. Ця мікросхема є спеціалізованим 16-канальним контролером, розробленим для керування світлодіодами та сервомоторами. Вона забезпечує наступні ключові переваги:

- керування відбувається по цифровій шині I2C (Inter-Integrated Circuit), яка потребує лише двох ліній зв'язку (лінія даних SDA та лінія тактування SCL). Це дозволяє зберегти дефіцитні пini мікроконтролера;
- генерація ШИМ-сигналів здійснюється повністю апаратно за допомогою вбудованого тактового генератора частотою 25 мегагерц, що повністю знімає навантаження з процесора ESP32;
- підтримується 16 незалежних каналів з 12-бітною роздільною здатністю (4096 дискретних кроків), що гарантує високу плавність і бездоганну точність рухів сервоприводів;
- мікросхема має окремі клемні колодки для підключення зовнішнього потужного джерела живлення, фізично ізолюючи мікроконтролер від руйнівних зворотних струмів, що виникають в обмотках двигунів.

Зовнішній вигляд модуля PCA9685 наведено на рисунку 2.1. У розробленому проєкті шину I2C програмно перепризначено на пini GPIO 14 (для

лінії SDA) та GPIO 15 (для лінії SCL), оскільки стандартні піни можуть конфліктувати з внутрішніми шинами камери.

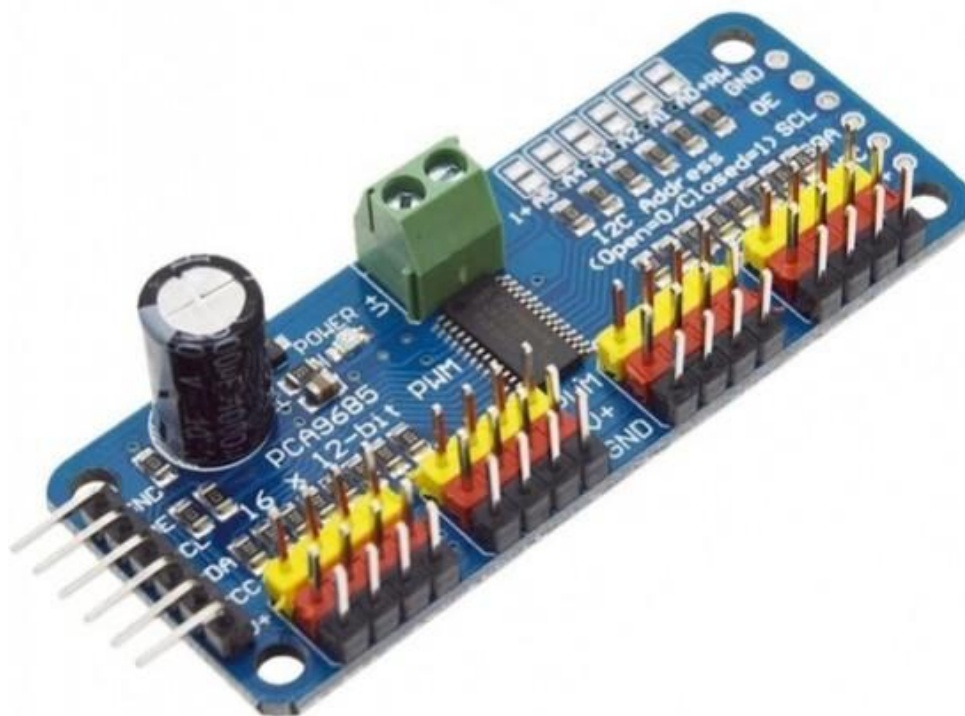


Рисунок 2.1 – Зовнішній вигляд 16-канального ШІМ-контролера PCA9685

У якості виконавчих механізмів обрано стандартні цифрові сервоприводи MG996R [11]. На відміну від мініатюрних бюджетних сервоприводів (наприклад, SG90), які мають пластикові шестерні та мінімальний крутний момент (близько 1.5 кілограма на сантиметр), моделі MG996R оснащені металевим редуктором і забезпечують крутний момент до 11 кілограмів на сантиметр. Такий вибір обґрунтовано специфікою навантаження: лоток для сміття разом із відходами може мати значну масу (наприклад, металеві бляшанки або наповнені пластикові ємності). Спроба використати слабкий двигун призведе до його швидкого згоряння через перевантаження.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

18

Система керування просторовим переміщенням побудована за принципом кінематики з двома ступенями свободи. Вона використовує два сервоприводи, які працюють у синхронній зв'язці:

- 1) сервопривід напрямку (Pan servo), який відповідає за обертання всієї сортувальної платформи до обраного відсіку. Він налаштовується на роботу у діапазоні кутів: 45 градусів (відсік "Пластик"), 135 градусів (відсік "Папір") та 180 градусів (відсік "Метал");
- 2) сервопривід скидання (Tilt servo), який здійснює безпосередній нахил лотка для фізичного скидання відходу в бак. Він змінює свій робочий кут від 90 градусів (горизонтальне положення очікування) до 150 градусів (кут скидання).

2.3. Розробка підсистеми живлення пристрою

Підсистема живлення є однією з найважливіших складових будь-якого вбудованого інженерного проєкту, що містить як чутливу цифрову електроніку, так і потужні електромеханічні компоненти. Неправильне або недбале проєктування ланцюгів живлення найчастіше стає причиною нестабільної роботи мікроконтролера, раптових перезавантажень через падіння напруги (brownout resets), зависань шини I2C та викривлення або повного спотворення зображень з камери через електромагнітні перешкоди.

Щоб правильно спроектувати живлення, необхідно провести розрахунок балансу енергоспоживання. Основними споживачами енергії у розробленому пристрої розумного смітника є:

- мікроконтролер ESP32-CAM: під час передачі даних по Wi-Fi та виконання ресурсоємних математичних операцій для розпізнавання об'єктів його споживання може досягати пікових значень до 500 міліампер. Також варто враховувати живлення зовнішньої мікросхеми PSRAM;
- камера OV2640: її власне споживання становить близько 70 міліампер в активному режимі захоплення кадрів;

— два потужних сервоприводи MG996R: ці двигуни постійного струму мають значні пускові струми.

Згідно з технічними характеристиками, один сервопривід MG996R під час швидкого старту, зміни напрямку обертання або при заклинюванні (струм блокування, stall current) може споживати від 1.5 до 2.5 ампер при напрузі 5 вольт.

Враховуючи одночасну роботу двох таких моторів під час циклу сортування (один утримує нахил, інший обертає масивну платформу), короткочасне пікове споживання струму всією системою може сягати 4-5 ампер. Застосування звичайного живлення від USB-порту комп'ютера або стандартного зарядного пристрою для мобільних телефонів (що видають 1-2 амperi) є абсолютно неприпустимим і призведе до миттєвого відключення пристрою. Більше того, спроба жити мікроконтролер та індуктивні навантаження (двигуни) від однієї провідної лінії призведе до значних просадок напруги, що гарантовано спричинить спрацювання внутрішнього детектора падіння напруги (BOD) на платі ESP32.

Для вирішення цієї інженерної проблеми було спроектовано розподілену систему живлення з топологією зірки. Вона базується на використанні імпульсних понижуючих перетворювачів напруги (Buck Converter) типу LM2596 [12]. На відміну від лінійних стабілізаторів (типу LM7805), які перетворюють надлишок напруги в тепло і мають низький коефіцієнт корисної дії (ККД), імпульсні перетворювачі забезпечують ККД понад 85 відсотків і здатні видавати великі струми без значного нагрівання.

Джерелом первинної енергії слугує зовнішній мережевий адаптер живлення з вихідною напругою 12 вольт та струмом навантаження не менше 3 ампер (потужністю 36 ват). Ця первинна напруга подається паралельно на вхід двох незалежних DC-DC перетворювачів:

— перший перетворювач налаштований на прецизійну вихідну напругу 5 вольт і забезпечує живлення виключно "чистої" зони — мікроконтролера ESP32-CAM та сенсорів. Вхідний LDO-стабілізатор

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
20

самого модуля ESP32 додатково фільтрує та знижує напругу до 3.3 вольт, необхідних для живлення чипа;

- другий перетворювач також налаштований на вихідну напругу 5 вольт і підключений безпосередньо до клем живлення модуля PCA9685 для енергозабезпечення "брудної" зони — сервоприводів.

Крім того, для компенсації короточасних просадок напруги в моменти різкого старту двигунів, згідно з правилами проектування електроніки, на шині живлення сервоприводів (безпосередньо біля клем PCA9685) присутній електролітичний конденсатор розв'язки великої ємності (1000 мікрофарад). Такий конденсатор виконує роль локального накопичувача енергії та гасить небезпечні стрибки напруги (spikes). Цей комплексний підхід фізично ізолює кола живлення цифрової та силової частин пристрою, забезпечуючи безперебійну роботу логіки розпізнавання навіть при різких рухах сортувального механізму. Структуру цієї підсистеми більш наочно представлено далі у підрозділі, присвяченому електричним схемам.

2.4. Проектування корпусу та механізму сортування відходів

Проектування механічної та конструктивної частини розумного смітника вимагає вирішення задачі ефективного, надійного та безперебійного переміщення побутових предметів різної геометричної форми, матеріалу та ваги. Конструкція корпусу повинна строго відповідати кільком основним експлуатаційним вимогам: мати повністю ізольований простір для камери зі стабільним внутрішнім освітленням, містити зручний та безпечний приймальний лоток для сміття, а також надійний розподільчий вузол, який безпомилково спрямовує відходи до одного з накопичувальних баків.

Концептуально архітектуру корпусу розділено на дві основні функціональні зони:

1) верхній аналітичний блок. Це інтелектуальна частина пристрою, де надійно закріплені камера, мікроконтролер, модулі живлення, система штучного освітлення та рухомий приймальний лоток;

2) нижній накопичувальний блок. Ця частина є суто утилітарною і розділена перегородками на чотири рівних і містких відсіки, призначені для збору пластику, паперу, металу та іншого (невизначеного) типу сміття відповідно.

Приймальний лоток розташовується в центрі верхнього блоку безпосередньо в полі зору оптичної осі камери. Система штучного освітлення забезпечує рівномірне та безтіньове освітлення лотка. Це дозволяє камері фіксувати кольори та текстури об'єктів з максимальною контрастністю та точністю, що є критично важливим для отримання високого відсотка правильних класифікацій моделлю машинного навчання.

Механізм сортування спроектовано за поширеним у робототехніці принципом поворотного скидача (pan-tilt dumper mechanism). Його плавну та швидку роботу забезпечують два сервоприводи MG996R. Сам лоток для сміття закріплено безпосередньо на шліцьовому валу сервопривода скидання (Tilt), який у стані спокою утримує його в строго горизонтальному положенні (паралельно землі, під кутом 90 градусів за шкалою привода). Вся ця рухома збірка, що складається з лотка та Tilt-мотора, встановлена на спеціальну міцну обертову платформу. Платформа, у свою чергу, жорстко з'єднана з валом сервопривода напрямку (Pan). Обертова платформа має можливість вільно повертатися у горизонтальній площині на 180 градусів, проходячи над отворами всіх чотирьох накопичувальних відсіків.

Фізичний алгоритм процесу сортування відбувається за такою програмною циклограмою:

- мікроконтролер циклічно (кожні 3 секунди) захоплює кадр з камери, одночасно транслюючи його на локальний веб-сервер для відлагодження;
- захоплене зображення перетворюється у формат RGB888 та

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
22

передається в модель машинного навчання (Edge Impulse) для класифікації;

- якщо нейромережа виявляє об'єкт із впевненістю (confidence) понад 70 відсотків, визначається його категорія;
- отримавши результат, сервопривід напрямку (Pan) обертає платформу до відповідного відсіку: кут 45 градусів для відсіку "Пластик", 90 градусів інше, 135 градусів для "Папір", та 180 градусів для "Метал";
- після зупинки обертання (програмна затримка 500 мс) подається команда на сервопривід скидання (Tilt). Він змінює кут лотка з 90 до 150 градусів, і предмет зісковзує у бак-накопичувач;
- після затримки в 1 секунду для гарантованого падіння, Tilt-сервопривід повертає лоток у горизонтальне положення (90 градусів);
- рап-сервопривід повертає платформу в центральну нейтральну позицію (90 градусів), і система продовжує цикл розпізнавання.

Всі несучі деталі та елементи механізму, що піддаються найбільшому механічному навантаженню під час роботи (кріплення моторів, поворотна платформа, осі обертання), ретельно спроектовані за допомогою систем автоматизованого проєктування тривимірних об'єктів (САПР). Їх передбачено виготовляти методом технології 3D-друку (FDM) зі спеціального інженерного пластику PETG. На відміну від популярного екологічного пластику PLA, який є крихким і має низьку температуру розм'якшення, матеріал PETG відрізняється надзвичайно високою механічною міцністю, гнучкістю та стійкістю до ударних та вібраційних навантажень, що є ідеальним для рухомих механізмів. Запропоновану 3D-модель механізму сортування відходів наведено на рисунку 2.2.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
23

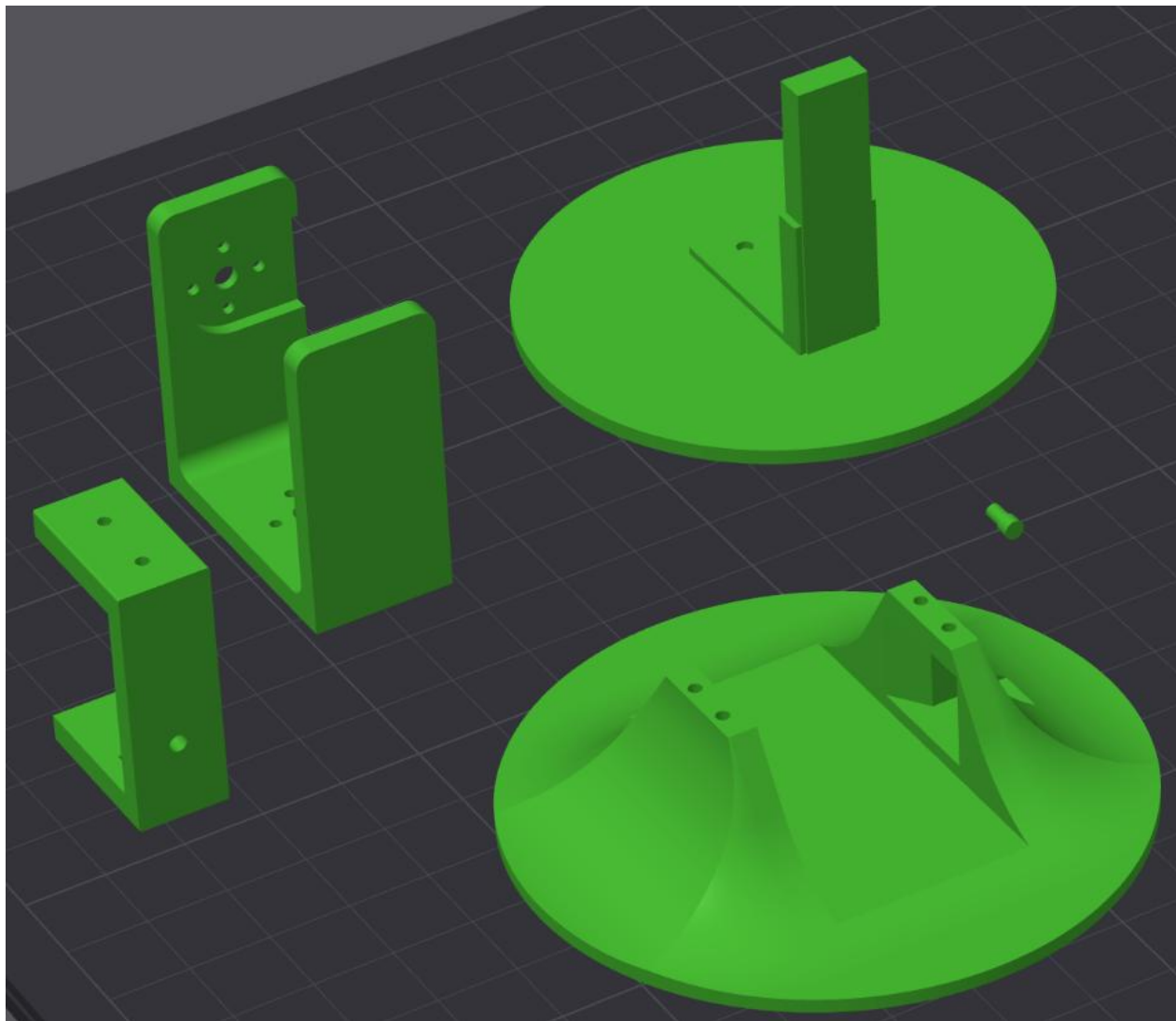


Рисунок 2.2 – Модель механізму сортування відходів

2.5. Розробка принципової електричної схеми

Для детальної формалізації архітектури розробленого апаратного забезпечення важливо розуміти загальні принципи взаємодії компонентів. Основне ядро системи становить центральний блок управління та аналізу на базі мікроконтролера ESP32-CAM. До цього блоку безперервно надходять вхідні дані у вигляді відеопотоку від периферійного блоку захоплення зображення (який містить камеру OV2640). Мікроконтролер аналізує ці мультимедійні дані, використовуючи математичний апарат внутрішньої логіки машинного навчання. Після розпізнавання об'єкта мікроконтролер генерує та форматує керуючі команди, які через цифрову послідовну шину I2C передаються до блоку ШІМ-

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

24

контролера PCA9685. ШІМ-контролер, отримавши цифрові команди, перетворює їх і генерує відповідні аналогові широтно-імпульсні сигнали для блоку виконавчих механізмів, до складу якого входять сервоприводи Pan та Tilt. Усі перелічені блоки стабільно отримують необхідну електричну енергію від блоку розподілу живлення, який містить необхідні перетворювачі та фільтри.

Для переходу від концептуального рівня до етапу практичної реалізації, макетування та зборки готового пристрою, була розроблена детальна принципова електрична схема. Вона точно описує підключення всіх контактів, типи з'єднувальних провідників та номінали електронних компонентів.

На принциповій електричній схемі наочно відображено наступні ключові з'єднання та топологічні рішення:

- зовнішній мережевий адаптер живлення постійного струму (DC 12 вольт) підключено паралельно до входних клем (IN+ та IN-) двох знижувальних DC-DC модулів;
- вихід першого DC-DC модуля (попередньо відкалібрований на рівень 5 вольт) з'єднано з контактом 5V мікроконтролерного модуля ESP32-CAM та відповідним загальним контактом GND. Цей ланцюг призначений виключно для живлення логічної частини;
- вихід другого DC-DC модуля (також жорстко налаштований на 5 вольт) з'єднано товстим мідним провідником з клемною колодкою живлення (V+ та GND) на платі PCA9685. Це топологічне рішення гарантує гальванічно ізольоване силове живлення виключно для обмоток сервоприводів;
- контакти живлення вбудованої цифрової логіки самої плати PCA9685 (контакт VCC) підключено безпосередньо до стабілізованого піна 3.3V на платі ESP32-CAM, оскільки шина передачі даних I2C в ESP32 функціонує виключно на логічних рівнях 3.3 вольт, і подача 5 вольт на шину може вивести мікроконтролер з ладу;
- інформаційна лінія передачі даних SDA модуля PCA9685 надійно підключена до піна GPIO 14 мікроконтролера, а лінія тактування SCL

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		25

підключена до піна GPIO 15. Дані піни були обрані програмно для уникнення будь-яких конфліктів з вбудованим обладнанням;

- керівний (сигнальний) дріт сервопривода напрямку (Pan) підключено до нульового вихідного каналу (Channel 0) модуля розширення PCA9685;
- керівний дріт сервопривода нахилу (Tilt) аналогічно підключено до першого вихідного каналу (Channel 1) того ж модуля розширення;
- всі лінії "землі" (GND) від усіх без винятку модулів та компонентів з'єднані в одну загальну еквипотенційну точку. Це є класичним і необхідним правилом схемотехніки для вирівнювання потенціалів та забезпечення коректної і безпомилкової передачі високочастотних цифрових сигналів між пристроями.

Завдяки ретельному та всебічному проектуванню принципової схеми на початковому етапі вдалося уникнути серйозних схемотехнічних помилок під час фізичного макетування пристрою, а також гарантувати його безперебійну та надійну роботу на всіх етапах польового тестування.

2.6. Висновки до розділу 2

У даному розділі було проведено глибоке та комплексне проектування всієї апаратної частини розумного смітника, від вибору керуючого мікроконтролера до детального опису механічної та електричної структури. У результаті порівняльного аналізу архітектур було обрано мікроконтролер ESP32-CAM у поєднанні з камерою OV2640. Це рішення є оптимальною платформою, що забезпечує ідеальний баланс між високою обчислювальною потужністю (яка критично необхідна для швидкої роботи нейромереж технології TinyML на крайових пристроях), наявністю бездротового зв'язку та економічною доступністю. Обґрунтовано абсолютну необхідність використання виділеного ШІМ-контролера PCA9685 для апаратного керування двома сервоприводами MG996R через шину I2C, що дозволило елегантно усунути проблему нестачі вільних контактів вводу-виводу та запобігти виникненню фатальних апаратних

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		26

конфліктів під час виконання паралельних задач операційною системою реального часу.

Окрім вибору цифрових компонентів, було надзвичайно детально спроектовано надійну, відмовостійку підсистему живлення з використанням двох окремих потужних знижувальних DC-DC перетворювачів напруги. Така розподілена топологія надійно ізолює чутливі високочастотні цифрові компоненти (процесор, пам'ять, камеру) від потужних електромагнітних перешкод та катастрофічних падінь напруги, які неминуче виникають під час стартів та зупинок потужних колекторних електродвигунів. Також було розроблено та обґрунтовано механічну концепцію всієї системи сортування, що складається з обертової платформи керування напрямком та похилого лотка скидання. Дана конструкція забезпечує винятково точний та надійний фізичний розподіл розпізнаних відходів на чотири ізольовані категорії.

Створена в процесі інженерного проєктування детальна принципова електрична схема повністю формалізувала технічну архітектуру пристрою. Вона підготувала міцний і перевірений апаратний фундамент для переходу до наступного, не менш важливого етапу виконання дипломної роботи — розробки програмного забезпечення мікроконтролера, інтеграції навчених алгоритмів комп'ютерного зору та налаштування локального вебінтерфейсу для моніторингу працездатності системи.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА АЛГОРИТМІВ КЕРУВАННЯ

На етапі апаратного проєктування було визначено ключові компоненти системи: мікроконтролер ESP32-CAM, ШІМ-контролер PCA9685 та сервоприводи сортувального механізму. Проте апаратна база не здатна виконувати поставлені завдання без відповідного програмного забезпечення. У цьому розділі детально розглядається процес розробки програмного коду пристрою: від вибору середовища розробки та ініціалізації периферії до реалізації логіки фізичного керування сервоприводами, створення локального вебінтерфейсу для моніторингу та підготовки цифрових даних (зображень) для їх подальшої передачі моделі машинного навчання. Відповідно до поставленої задачі, код має працювати виключно на реальній апаратній платформі, ефективно використовувати обмежену оперативну пам'ять та враховувати особливості багатозадачності операційної системи мікроконтролера.

3.1. Вибір середовища розробки та програмних бібліотек

Для програмування мікроконтролера ESP32-CAM було використано екосистему розробки на базі C++. В якості середовища розробки (IDE) було використано PlatformIO [13], оскільки вона містить оптимізовані компілятори (toolchains) для архітектури Xtensa та надає доступ до великої бази готових бібліотек, які абстрагують складну роботу з регістрами процесора.

Для забезпечення коректної роботи всіх апаратних модулів у проєкті було підключено та використано наступні спеціалізовані бібліотеки:

— `<esp_camera.h>` — базова бібліотека від компанії Espressif Systems [8].

Вона містить низькорівневі драйвери для роботи з сенсорами камер (зокрема обраним OV2640), реалізує протокол паралельної передачі даних I2S, керує конфігурацією камери через шину SCCB (аналог I2C) та керує прямим доступом до пам'яті (DMA) для збереження кадрів у зовнішній PSRAM;

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
28

- ``<Wire.h>`` — стандартна бібліотека для реалізації протоколу I2C. Вона необхідна для апаратної генерації тактових імпульсів та передачі даних між мікроконтролером ESP32-CAM і зовнішнім ШІМ-контролером;
- ``<Adafruit_PWMServoDriver.h>`` — спеціалізована бібліотека для керування мікросхемою PCA9685 [10]. Вона абстрагує складні операції запису у внутрішні регістри мікросхеми, дозволяючи розробнику встановлювати частоту роботи та ширину імпульсів за допомогою простих та зрозумілих функцій;
- ``<WiFi.h>`` та ``<WebServer.h>`` — набір мережеских бібліотек, які реалізують роботу мікроконтролера у режимі Wi-Fi станції (STA) та розгортання повноцінного HTTP-сервера для обробки вхідних клієнтських запитів;
- ``<Mono-project-AOL_inferencing.h>`` та ``image.hpp`` — експортований пакет (SDK) платформи Edge Impulse [14]. Ці заголовні файли містять саму нейронну мережу, скомпільовану під архітектуру мікроконтролера, а також допоміжні функції для обробки зображень (обрізання, інтерполяція кольорів тощо).

Використання вищезазначених бібліотек дозволяє значно прискорити процес розробки, оскільки вони беруть на себе реалізацію складних низькорівневих протоколів, дозволяючи зосередитись на створенні високорівневої бізнес-логіки сортування відходів.

3.2. Розробка алгоритму ініціалізації периферійних пристроїв

Життєвий цикл програми на мікроконтролері починається з функції ініціалізації ``setup()``. Першим кроком є запуск послідовного порту (Serial) на швидкості 115200 бод. Цей інтерфейс критично необхідний для виведення діагностичних повідомлень (наприклад, інформації про ініціалізацію, підключення до Wi-Fi та результати класифікації) у термінал розробника.

Наступним, і найбільш складним етапом, є конфігурація камери. Оскільки модулі ESP32-CAM виготовляються за шаблоном "AI Thinker", конфігурація

вимагає жорсткого призначення пінів. Структура ``camera_config_t`` заповнюється масивом констант: визначаються контакти шини даних (від Y2 до Y9), контакти синхронізації (VSYNC, HREF, PCLK), а також контакти керуючої шини (SIOD, SIOC). Важливою частиною конфігурації є встановлення тактової частоти XCLK на рівні 20 мегагерц та налаштування внутрішнього таймера LEDC, який апаратно генерує цей тактовий сигнал.

Формат пікселів встановлюється як ``PIXFORMAT_JPEG``, оскільки апаратне стиснення безпосередньо на камері суттєво заощаджує пропускну здатність шини. Роздільна здатність встановлюється на рівні ``FRAMESIZE_QVGA`` (320 на 240 пікселів), що є оптимальним форматом для подальшої обробки нейронною мережею без надмірного споживання пам'яті. Кількість буферів кадрів (``fb_count``) встановлено рівною 1, а місце їх зберігання примусово призначено у зовнішню пам'ять PSRAM (``CAMERA_FB_IN_PSRAM``).

Після виклику системної функції ``esp_camera_init()``, програмний алгоритм додатково перевіряє ідентифікатор сенсора (PID). Якщо розпізнано альтернативний сенсор OV3660 (код підтримує зворотну сумісність з різними модулями), застосовуються специфічні налаштування: вертикальне віддзеркалення (``set_vflip``), підвищення яскравості та скидання насиченості. Це дозволяє усунути специфічні апаратні артефакти зображення для даного сенсора.

Ініціалізація зовнішнього контролера сервоприводів PCA9685 відбувається за допомогою бібліотеки ``Wire``. Враховуючи, що стандартні піни I2C можуть конфліктувати, розроблений код примусово ініціалізує шину на пінах 14 (SDA) та 15 (SCL) викликом ``Wire.begin(14, 15)``. Після цього мікросхемі PCA9685, що має апаратну адресу ``0x40``, надсилаються команди налаштування тактового генератора (27 мегагерц) та встановлення робочої частоти ШІМ-сигналу на рівні 50 герц.

Завершується етап ініціалізації встановленням сервоприводів механізму сортування у початкову "паркувальну" позицію (90 градусів для сервопривода напрямку та 90 градусів для сервопривода нахилу), що гарантує готовність лотка

до прийому першого об'єкта. Скріншот лістингу функції налаштування камери наведено на рисунку 3.1.

```
static camera_config_t camera_config = {
    .pin_pwdn = PWDN_GPIO_NUM,
    .pin_reset = RESET_GPIO_NUM,
    .pin_xclk = XCLK_GPIO_NUM,
    .pin_sscb_sda = SIOD_GPIO_NUM,
    .pin_sscb_scl = SIOC_GPIO_NUM,
    .pin_d7 = Y9_GPIO_NUM,
    .pin_d6 = Y8_GPIO_NUM,
    .pin_d5 = Y7_GPIO_NUM,
    .pin_d4 = Y6_GPIO_NUM,
    .pin_d3 = Y5_GPIO_NUM,
    .pin_d2 = Y4_GPIO_NUM,
    .pin_d1 = Y3_GPIO_NUM,
    .pin_d0 = Y2_GPIO_NUM,
    .pin_vsync = VSYNC_GPIO_NUM,
    .pin_href = HREF_GPIO_NUM,
    .pin_pclk = PCLK_GPIO_NUM,
    .xclk_freq_hz = 20000000,
    .ledc_timer = LEDC_TIMER_0,
    .ledc_channel = LEDC_CHANNEL_0,
    .pixel_format = PIXFORMAT_JPEG,
    .frame_size = FRAMESIZE_QVGA,
    .jpeg_quality = 12,
    .fb_count = 1,
    .fb_location = CAMERA_FB_IN_PSRAM,
    .grab_mode = CAMERA_GRAB_WHEN_EMPTY,
};
bool ei_camera_init(void) {
    if (is_initialised) return true;
    esp_err_t err = esp_camera_init(&camera_config);
    if (err != ESP_OK) {
        Serial.printf("Camera init failed with error 0x%x\n", err);
        return false;
    }
    sensor_t *s = esp_camera_sensor_get();
    if (s->id.PID == OV3660_PID) {
        s->set_vflip(s, 1);
        s->set_brightness(s, 1);
        s->set_saturation(s, 0);
    }
    is_initialised = true;
    return true;
}
```

Рисунок 3.1 – Скріншот лістингу функції ініціалізації конфігурації камери

3.3. Програмна реалізація вебсервера для віддаленого моніторингу

Для забезпечення зручного відлагодження системи, перевірки кута нахилу камери та умов освітлення всередині корпусу смітника, було розроблено локальний вебсервер. Він дозволяє оператору віддалено, через звичайний інтернет-браузер, переглядати зображення, яке "бачить" мікроконтролер у даний момент.

Процес ініціалізації мережевого підключення відбувається в функції `setup()`. Мікроконтролер викликає метод `WiFi.begin()`, передаючи йому

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		31

жорстко задані в коді облікові дані локальної бездротової мережі (ідентифікатор SSID "librarium" та відповідний пароль). Програма входить у цикл очікування, періодично перевіряючи статус підключення (`WiFi.status()`). Після успішного встановлення з'єднання, локальна IP-адреса пристрою виводиться в послідовний порт, а об'єкт сервера `WebServer server(80)` починає прослуховувати стандартний HTTP-порт.

У програмному коді зареєстровано маршрут обробки запитів для кореневого шляху (`"/`). При зверненні користувача за цим маршрутом викликається функція-обробник `handleCameraView()`. Логіка цієї функції є наступною:

1) відбувається звернення до драйвера камери `esp_camera_fb_get()`, який миттєво захоплює поточний кадр безпосередньо у пам'ять PSRAM. Оскільки формат зображення на етапі конфігурації було задано як JPEG, буфер уже містить стиснений і готовий до відправки файл;

2) у разі успішного захоплення, вебсервер формує правильний HTTP-заголовок з кодом стану 200 (OK) та типом контенту `image/jpeg`;

3) стиснений масив байтів зображення передається клієнту через функцію `server.send_P()`;

4) після відправки обов'язково викликається функція `esp_camera_fb_return()`, яка звільняє виділену під кадр пам'ять, запобігаючи виникненню витоків пам'яті (memory leaks).

```
// Web Server route to see the camera view
void handleCameraView() {
    camera_fb_t *fb = esp_camera_fb_get();
    if (!fb) {
        server.send(500, "text/plain", "Camera capture failed");
        return;
    }
    // Send the JPEG image directly to the browser
    server.send_P(200, "image/jpeg", (const char *)fb->buf, fb->len);
    esp_camera_fb_return(fb);
}
```

Рисунок 3.2 – Скріншот лістингу функції обробки HTTP-запиту та передачі зображення

Даний підхід не реалізує повноцінний відеопотік (JPEG stream), а натомість створює єдиний статичний знімок при кожному оновленні сторінки браузера. Це свідоме архітектурне рішення, яке дозволяє зберегти дефіцитні ресурси процесора для головної задачі — роботи нейронної мережі, надаючи при цьому достатній рівень візуального контролю. Скріншот лістингу коду, що реалізує вебсервер для віддаленого моніторингу, наведено на рисунку 3.2.

3.4. Розробка логіки керування сервоприводами через інтерфейс I2C

Основним механізмом розподілу відходів керує функція `dropTrash(String category)`, яка викликається одразу після успішного розпізнавання об'єкта. Функція приймає строковий аргумент (категорію), який безпосередньо визначає кінематичну поведінку системи.

Оскільки мікросхема PCA9685 не оперує звичними "градусами", програмний код повинен містити конвертер фізичних величин у машинні "тіки". Робоча частота ШІМ встановлена на 50 герц, що відповідає періоду 20 мілісекунд. Мікросхема PCA9685 має 12-бітний лічильник (від 0 до 4095 тіків у межах одного періоду). Стандартні сервоприводи MG996R реагують на імпульс тривалістю від 1 мілісекунди (що відповідає 0 градусів) до 2 мілісекунд (що відповідає 180 градусів). Шляхом математичних розрахунків ці значення тривалості були конвертовані у відповідні межі лічильника. У коді вони представлені константами `SERVO_MIN_TICKS = 102` (мінімальне положення) та `SERVO_MAX_TICKS = 512` (максимальне положення). Функція `angleToTicks()` використовує стандартну математичну операцію `map()` для лінійної інтерполяції заданого кута від 0 до 180 градусів у відповідну кількість тіків для регістру PCA9685.

Логіка фізичного сортування складається з чіткої послідовності кроків (циклограми):

— спочатку аналізується вхідна категорія. Змінній цільового кута `targetPan` присвоюється значення 45 градусів (для Plastic), 135 градусів (для Paper) або 180 градусів (для Metal). Якщо категорія

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		33

невідома, кут залишається на рівні 90 градусів;

- сервоприводу напрямку (який підключений до каналу `PAN\_CH`) відправляється команда на зміну кута (`targetPan`). Обертова платформа починає рух до потрібного відсіку. Для забезпечення достатнього часу на механічне переміщення платформи та гасіння інерційних коливань, система призупиняється на 500 мілісекунд функцією `ei\_sleep()`. Використання саме цієї функції замість стандартної `delay()` обґрунтовано тим, що вона безпечно взаємодіє з перериваннями бібліотек Edge Impulse;
- після стабілізації платформи, сервоприводу нахилу (канал `TILT\_CH`) надсилається команда встановити кут 150 градусів. Лоток різко нахиляється, скидаючи сміття. Система очікує 1000 мілісекунд, даючи достатньо часу для того, щоб предмет впав у бак під дією гравітації;
- наступним кроком є повернення механізмів у вихідний стан. Сервопривід нахилу повертається на кут 90 градусів (горизонтальне положення), після чого з паузою у 500 мілісекунд сервопривід напрямку повертає пустий лоток у центральну позицію (90 градусів).

```
const int SERVO_MIN_TICKS = 102; // 0 degrees
const int SERVO_MAX_TICKS = 512; // 180 degrees
int angleToTicks(float angle) {
    return map((int)angle, 0, 180, SERVO_MIN_TICKS, SERVO_MAX_TICKS);
}
// Physical sorting sequence
void dropTrash(String category) {
    int targetPan = 90; // Default: Unknown bin
    if (category == "Plastic") targetPan = 45;
    else if (category == "Paper") targetPan = 135;
    else if (category == "Metal") targetPan = 180;
    ei_printf("\n--- PHYSICALLY SORTING: %s ---\n", category.c_str());
    // 1. Pan to the selected bin
    pwm.setPWM(PAN_CH, 0, angleToTicks(targetPan));
    ei_sleep(500); // Wait for pan to rotate
    // 2. Tilt tray to drop the trash
    pwm.setPWM(TILT_CH, 0, angleToTicks(150)); // Dump angle
    ei_sleep(1000); // Wait for gravity to do its job
    // 3. Tilt back to flat
    pwm.setPWM(TILT_CH, 0, angleToTicks(90)); // Flat holding angle
    ei_sleep(500);
    // 4. Return Pan to center
    pwm.setPWM(PAN_CH, 0, angleToTicks(90));
    ei_sleep(500);
    ei_printf("--- SORTING COMPLETE ---\n\n");
}
```

Рисунок 3.3 – Скріншот лістингу логіки обчислення ШІМ-імпульсів та сортування

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
34

Даний програмний алгоритм гарантує повну синхронізацію двох моторів, виключаючи ймовірність зіткнення лотка зі стінками корпусу або передчасного скидання відходів мимо обраного відсіку. Скріншот лістингу коду функції сортування наведено на рисунку 3.3.

3.5. Оптимізація процесу захоплення зображення

Безперервний цикл роботи пристрою відбувається у функції ``loop()``. Оскільки модуль ESP32-CAM має вкрай обмежені ресурси пам'яті, процес захоплення та обробки кадрів для нейронної мережі потребує суворої оптимізації та контролю за розподілом оперативної пам'яті (RAM).

Першим кроком у головному циклі є виділення пам'яті для масиву ``snapshot_buf``. Цей масив призначений для зберігання зображення у форматі RGB888, яке є єдиним підтримуваним форматом для вхідного шару більшості згорткових нейронних мереж. Розмір необхідної пам'яті обчислюється множенням ширини кадру (320 пікселів), висоти (240 пікселів) та кількості байт на один піксель кольорового зображення (3 байти для каналів Red, Green та Blue). Сумарно це становить 230 400 байт (або 225 кілобайтів) безперервного масиву пам'яті. У разі невдалого виділення пам'яті функцією ``malloc()``, алгоритм генерує помилку, уникаючи критичного збою мікроконтролера.

Після ініціалізації структури сигналу ``ei::signal_t``, алгоритм викликає власну функцію ``ei_camera_capture()``. Її робота поділяється на кілька життєво важливих етапів:

- 1) виклик драйвера камери ``esp_camera_fb_get()`` для отримання буфера з кадром у стисненому форматі JPEG;
- 2) використання вбудованої апаратно-програмної функції ``fmt2rgb888()`` для декомпресії JPEG-файлу. Функція зчитує стиснені байти та розпаковує їх у раніше виділений "сирий" масив ``snapshot_buf``, перетворюючи стиснений кадр на класичну матрицю кольорових пікселів;

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
35

- 3) обов'язкове повернення вихідного буфера JPEG назад у систему ``esp_camera_fb_return()``, щоб звільнити пам'ять камери для наступного знімка;
- 4) у разі якщо роздільна здатність матриці камери відрізняється від роздільної здатності, з якою тренувалася нейронна мережа, викликається функція ``crop_and_interpolate_rgb888()`` з бібліотеки Edge Impulse, яка оптимізованими математичними методами обрізає краї зображення (робить його квадратним) або масштабує до потрібного розміру.

Підготовлений масив даних передається у функцію-обгортку ``ei_camera_get_data()``. Її роль полягає в перетворенні 8-бітних кольорових компонентів кожного пікселя (від 0 до 255) у формат чисел із рухомою комою (float), який очікує ядро нейромережі.

Після успішного отримання та перетворення даних починається процес класифікації. Відповідно до коду програми, нейронна мережа функціонує в режимі виявлення об'єктів (``EI_CLASSIFIER_OBJECT_DETECTION``). Вона аналізує матрицю зображення і повертає масив знайдених об'єктів (``bounding boxes``) із відповідними мітками та коефіцієнтами впевненості. Алгоритм мікроконтролера циклічно перебирає всі знайдені об'єкти та визначає один з найвищою впевненістю (``best_confidence``). Важливою складовою логіки є використання порогу надійності: система приймає рішення лише в тому випадку, якщо рівень впевненості нейромережі становить не менше 70 відсотків (0.70). Це є захисним механізмом від хибних спрацьовувань та сортування сміття навання.

Після визначення найбільш ймовірного об'єкта, його мітка перетворюється на одну з базових категорій («Paper», «Plastic» або «Metal»). Отримана узагальнена категорія передається як аргумент до функції ``dropTrash()`` для ініціалізації фізичного процесу розподілу.

По завершенню циклу класифікації, життєво важливо звільнити динамічну пам'ять, виділену на початку циклу, викликом функції ``free(snapshot_buf)``. Без цієї оптимізації пристрій вичерпає всю оперативну пам'ять вже на другому кадрі

і перезавантажиться. Останнім кроком циклу `'loop()'` є очікування перед наступним знімком. Замість повної зупинки процесора використовується неблокуючий цикл із тривалістю 3 секунди, всередині якого викликається метод `'server.handleClient()'`. Завдяки цьому оптимізованому підходу, пристрій здатний одночасно відповідати на запити користувачів до вебінтерфейсу і відпочивати перед захопленням нового кадру для класифікації.

3.6. Висновки до розділу 3

В даному розділі було проведено детальний розбір процесу програмної реалізації розумного смітника. Успішно обрано сучасний інструментарій розробки на базі мови C++ та застосовано набір спеціалізованих бібліотек, які дозволили ефективно керувати апаратними ресурсами мікроконтролера ESP32-CAM. Розроблено надійний алгоритм ініціалізації складної периферії: налаштовано сенсор камери на оптимальну роздільну здатність із використанням апаратного стиснення JPEG, а також ініціалізовано шину I2C на нестандартних пінах для стабільного зв'язку з модулем PCA9685.

Успішно реалізовано програмну логіку керування сервоприводами, яка включає конвертацію фізичних кутів обертання у відповідні тривалості ШІМ-імпульсів. Розроблена циклограма руху гарантує плавне і безпомилкове переміщення відходів до обраного відсіку без ризику зіткнень. Також створено локальний вебсервер, який забезпечує зручний віддалений моніторинг у режимі реального часу, дозволяючи оператору бачити те саме, що аналізуватиме штучний інтелект.

Найбільшу увагу було приділено оптимізації процесу захоплення та підготовки зображень для їх подальшого використання моделлю машинного навчання. Розроблений алгоритм ефективно управляє вкрай обмеженою динамічною оперативною пам'яттю (PSRAM), безперебійно перетворюючи стиснені кадри JPEG у необхідний формат RGB888. Таким чином, програмний фундамент повністю готовий до останнього кроку – інтеграції навченої згорткової нейронної мережі та безпосередньої класифікації об'єктів.

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		37

РОЗДІЛ 4. РОЗРОБКА ТА ІНТЕГРАЦІЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ

У сучасному світі створення систем розумного сортування відходів неможливе без впровадження алгоритмів штучного інтелекту. Хоча апаратна та програмна інфраструктури, розглянуті у попередніх розділах, забезпечують фізичне переміщення сміття, "мозком" системи, що приймає рішення, є модель машинного навчання. У цьому розділі детально розглядається процес створення нейронної мережі: від вибору спеціалізованої платформи для вбудованих систем і підготовки набору даних до безпосереднього навчання, квантизації та програмної інтеграції згенерованої моделі у мікроконтролер ESP32-CAM. Головною метою цього етапу є досягнення високої точності розпізнавання при суворих обмеженнях на використання оперативної пам'яті та обчислювальної потужності.

4.1. Вибір платформи для створення моделі машинного навчання

Створення моделей штучного інтелекту для мікроконтролерів (так званий напрямок TinyML) кардинально відрізняється від традиційного машинного навчання. Звичайні моделі, створені за допомогою фреймворків TensorFlow або PyTorch, орієнтовані на запуск на потужних графічних процесорах (GPU) або багатоядерних центральних процесорах, маючи при цьому обсяг у десятки та сотні мегабайтів. Мікроконтролер ESP32-CAM має лише кілька мегабайтів оперативної пам'яті (включно із зовнішньою PSRAM), що виключає можливість використання класичних підходів.

В якості основного інструменту для розробки та навчання нейронної мережі було обрано платформу Edge Impulse [14]. Дана платформа є індустріальним стандартом у сфері TinyML, оскільки вона надає повний цикл розробки (pipeline) в єдиному хмарному інтерфейсі: від збору сирих даних з пристроїв до розгортання оптимізованого коду C++ без жодних зовнішніх залежностей.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
38

Вибір платформи Edge Impulse обґрунтовано наступними ключовими перевагами:

- наявність вбудованих оптимізаторів (EON Compiler), які компілюють нейронну мережу не як статичний граф операцій (як у TensorFlow Lite), а як нативний виконуваний код мовою C++, що значно зменшує споживання оперативної пам'яті та прискорює час інференсу;
- підтримка процесу пост-тренувальної квантизації (Post-Training Quantization), яка перетворює 32-бітні числа з рухомою комою (float32) у 8-бітні цілі числа (int8) без суттєвої втрати точності розпізнавання;
- вбудовані інструменти обробки цифрових сигналів (DSP), які дозволяють застосувати алгоритми перетворення кольорів, нормалізації та зміни розміру зображень безпосередньо у згенерованій бібліотеці, гарантуючи, що дані, подані на мікроконтролері, повністю збігатимуться з форматом даних, на якому навчалась модель;
- можливість експорту готової моделі у вигляді стандартної бібліотеки Arduino/PlatformIO, яка легко імпортується у будь-який існуючий проєкт без необхідності ручного переписування низькорівневих математичних операцій.

Завдяки цим перевагам, використання Edge Impulse дозволило абстрагуватися від складного процесу налаштування середовища навчання (Python, CUDA, оптимізатори) і повністю зосередитись на архітектурі моделі та якості набору даних.

4.2. Збір та підготовка набору даних для класифікації сміття

Фундаментом будь-якої моделі штучного інтелекту є набір даних (dataset), на якому вона навчається. Оскільки завданням проєкту є розпізнавання сміття, набір даних повинен репрезентативно відображати ті об'єкти, з якими система зіткнеться у реальних умовах експлуатації.

Згідно з дослідженнями Adedeji та Wang у статті "Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network" [15],

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
39

обсяг, різноманітність та якість розмітки набору даних є найвпливовішими факторами, що дозволяють згортковим нейронним мережам досягати точності понад 87 відсотків. Відповідно до цих рекомендацій, було зібрано та підготовлено набір зображень побутових відходів.

Збір даних здійснювався з урахуванням специфіки роботи базової камери OV2640. Зображення робилися безпосередньо з мікроконтролера через розроблений раніше вебінтерфейс. Це дозволило отримати кадри з тим самим кутом огляду, фокусною відстанню та умовами освітлення, які будуть присутні під час реального використання розумного смітника.

Дані були поділені на три основні концептуальні категорії: пластик (пляшки, упаковки, пластикові кришечки), папір (зібгані аркуші, серветки, картонні коробки) та метал (алюмінієві бляшанки, металеві кришки). В процесі підготовки зображення, завантажені у платформу Edge Impulse, пройшли процедуру анотування.

Оскільки в даному проєкті використовується не проста класифікація зображень (Image Classification), а виявлення об'єктів (Object Detection), кожне зображення було розмічене за допомогою обмежуючих рамок (bounding boxes). Навколо кожного значущого об'єкта було виділено прямокутну зону та присвоєно відповідну текстову мітку. Для забезпечення високої точності розпізнавання, об'єкти були анотовані з використанням стандартних англomовних міток: "paper" (папір), "plastic" (пластик) та "metal" (метал).

Після анотування весь набір даних був автоматично поділений платформою на тренувальну вибірку (близько 80 відсотків зображень) та тестову вибірку (близько 20 відсотків), яка ніколи не показується нейромережі в процесі навчання і слугує виключно для об'єктивної перевірки її здатності до узагальнення.

4.3. Навчання нейронної мережі для розпізнавання об'єктів

Наступним кроком після підготовки набору даних стало формування обчислювального графа (Impulse design) та безпосереднє тренування нейронної

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		40

мережі. Архітектура обробки даних (Pipeline) складалася з трьох основних блоків: блоку вхідних даних (роздільна здатність зображення), блоку обробки сигналу (DSP) та блоку машинного навчання.

Вхідна роздільна здатність була встановлена на рівні 96 на 96 пікселів. Незважаючи на те, що камера захоплює кадри у розмірі 320 на 240, нейронна мережа не здатна обробляти такі великі матриці в режимі реального часу на ESP32. Блок обробки сигналу (Image block) був налаштований на автоматичне обрізання та масштабування вхідного зображення до розміру 96 на 96 пікселів з перетворенням у формат RGB, що значно знизило обчислювальне навантаження.

В якості архітектури нейронної мережі було обрано алгоритм виявлення об'єктів (Object Detection) на базі архітектури FOMO (Faster Objects, More Objects) [14]. Ця архітектура, розроблена Edge Impulse, спеціально створена для мікроконтролерів. На відміну від класичних алгоритмів типу YOLO або MobileNet SSD, які вимагають значних обчислювальних ресурсів для розрахунку перекриття рамок (Anchor boxes), FOMO генерує теплову карту ймовірностей, вказуючи координати центрів об'єктів. Це дозволяє досягти швидкості у понад 10-15 кадрів на секунду навіть на слабких процесорах Cortex-M або Xtensa.

Процес навчання мережі тривав 60 епох. Швидкість навчання (Learning rate) була підібрана емпіричним шляхом і встановлена на рівні 0.001. Під час тренування платформа безперервно коригувала вагові коефіцієнти моделі, мінімізуючи функцію втрат.

По завершенню навчання модель автоматично пройшла етап 8-бітної квантизації. Це дозволило зменшити розмір моделі з кількох мегабайтів до приблизно 250 кілобайтів у флеш-пам'яті, при цьому використання динамічної оперативної пам'яті під час інференсу скоротилось до безпечних меж (близько 150-200 кілобайтів).

Тестування квантованої моделі на відкладеній вибірці показало, що показник точності F1-Score (який враховує як повноту, так і точність виявлення) досяг прийняттого для побутового використання рівня.

4.4. Інтеграція навченої моделі в мікроконтролер

Після успішного завершення етапу тренування, розгорнута модель була експортована з платформи Edge Impulse у вигляді стандартної бібліотеки C++. Дана бібліотека містить оптимізований вихідний код нейронної мережі та всі необхідні математичні функції для підготовки вхідних даних. Бібліотеку було інтегровано в структуру проєкту під назвою `<Mono-project-AOL_inferencing.h>`.

Для ініціалізації процесу логічного висновку (інференсу) в мікроконтролері використовується спеціальна структура `ei::signal_t`. Оскільки модель приймає на вхід зображення як одновимірний масив кольорових значень, структура `signal_t` пов'язується з функцією-обгорткою `ei_camera_get_data`. Ця функція послідовно читає дані зі створеного нами раніше буфера `snapshot_buf` та перетворює їх на масив чисел з рухомою комою, необхідних для роботи внутрішніх шарів мережі.

Сам запуск класифікації здійснюється викликом єдиної функції `run_classifier`, яка інкапсулює у собі всі складні математичні операції згортки та активації нейронів. Функція приймає підготовлений сигнал та записує результати у структуру `ei_impulse_result_t`. Скріншот лістингу коду, що реалізує налаштування сигналу та запуск класифікатора на мікроконтролері, наведено на рисунку 4.1.

```
ei::signal_t signal;
signal.total_length = EI_CLASSIFIER_INPUT_WIDTH * EI_CLASSIFIER_INPUT_HEIGHT;
signal.get_data = &ei_camera_get_data;
ei_printf("Running inference...\n");
ei_impulse_result_t result = {};
// Run the classifier
EI_IMPULSE_ERROR err = run_classifier(&signal, &result, debug_nn);
if (err != EI_IMPULSE_OK) {
    ei_printf("ERR: Failed to run classifier (%d)\n", err);
    free(snapshot_buf);
    return;
}
```

Рисунок 4.1 – Скріншот лістингу запуску класифікатора на мікроконтролері

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

42

4.5. Обробка результатів розпізнавання та прийняття рішень

Отримання результатів роботи нейронної мережі є лише першим етапом. Нейронна мережа не сортує сміття, вона лише повертає масив ймовірностей (confidence scores) та обмежуючих рамок для кожного об'єкта, який їй вдалося виявити у кадрі. Завданням прикладної програми є парсинг цих результатів, відсіювання "шуму" та прийняття остаточного рішення про те, в який саме відсік слід скинути відходи.

Враховуючи, що в лоток смітника користувач кидає лише один предмет за раз, програмна логіка реалізує пошук єдиного об'єкта з найвищим показником впевненості. За допомогою циклу перебирається масив `result.bounding_boxes`. Кожного разу, коли значення `bb.value` (ймовірність знаходження об'єкта в даній рамці) перевищує поточне найбільше значення `best_confidence`, система запам'ятовує нову мітку `best_rawLabel`.

Для запобігання хибним спрацьовуванням (наприклад, коли система намагається розпізнати сміття на порожньому лотку), запроваджено жорсткий поріг ймовірності (threshold). Будь-які результати, де рівень впевненості нейромережі становить менше ніж 0.70 (70 відсотків), повністю ігноруються.

Якщо ж знайдено об'єкт, який проходить поріг впевненості, його сира мітка (з якою навчалася модель) проходить через умовний оператор, що здійснює логічне перетворення (mapping) на одну з трьох цільових категорій фізичного механізму:

- мітка "paper" перетворюється на категорію "Paper";
- мітка "plastic" перетворюється на категорію "Plastic";
- мітка "metal" перетворюється на категорію "Metal".

Лише після успішного перетворення фінальна категорія передається до функції фізичного сортування `dropTrash()`, яка була розроблена у попередньому розділі. Якщо ж поріг не пройдено або об'єкт відсутній, лоток переміщується у категорію "інше". Скріншот лістингу програмного коду, який реалізує обробку результатів та прийняття рішень, наведено на рисунку 4.2.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
43

```

float best_confidence = 0.0;
String best_rawLabel = "";
String best_category = "Unknown";
// Find the single highest confidence detection in the entire image
for (uint32_t i = 0; i < result.bounding_boxes_count; i++) {
    ei_impulse_result_bounding_box_t bb = result.bounding_boxes[i];
    if (bb.value > best_confidence) {
        best_confidence = bb.value;
        best_rawLabel = String(bb.label);
    }
}
// Only accept it if the AI is at least 70% sure
if (best_confidence >= 0.70) {
    // Програмна логіка перетворення сирої мітки (best_rawLabel)
    // у відповідну базову категорію (best_category)
    // ...
    ei_printf("Detected [%s] -> Sorting into: %s (Confidence: %.2f)\n",
        best_rawLabel.c_str(), best_category.c_str(), best_confidence);
    // Physically move the motors to sort the trash
    dropTrash(best_category);
} else {
    ei_printf("No trash detected (or confidence too low).\n");
    dropTrash("unknown");
}

```

Рисунок 4.2 – Скріншот лістингу обробки результатів інференсу та прийняття рішень

4.6. Висновки до розділу 4

У четвертому розділі було здійснено комплексне проєктування та практичне впровадження алгоритмів штучного інтелекту, які становлять основу інтелектуальної складової розробленого пристрою. Перехід від класичних методів машинного навчання до парадигми граничних обчислень (TinyML) дозволив наділити малопотужний мікроконтролер ESP32-CAM повноцінним комп'ютерним зором.

Обґрунтовано та успішно застосовано спеціалізовану платформу Edge Impulse, яка надала повний цикл розробки TinyML-моделей в єдиному середовищі. Для забезпечення максимальної репрезентативності та адаптації моделі до реальних умов експлуатації збір даних здійснювався безпосередньо через оптичний сенсор OV2640 розробленого пристрою. Зібраний набір зображень побутового сміття був ретельно розмічений методами обмежуючих рамок (bounding boxes) для трьох ключових цільових категорій (папір, пластик, метал).

Вирішальним фактором успіху стало використання архітектури FOMO (Faster Objects, More Objects), спеціально адаптованої для мікроконтролерів. На

					ІАЛЦ.466120.004 ПЗ	Арк.
						44
Змін.	Арк.	№ докум.	Підпис	Дата		

відміну від класичних алгоритмів (наприклад, YOLO), які вимагають значних обчислень для розрахунку перекриття рамок, FOMO генерує ефективні теплові карти ймовірностей. Завдяки попередній обробці сигналу (масштабування до роздільної здатності 96x96 пікселів) та застосуванню алгоритмів 8-бітної пост-тренувальної квантизації, розмір кінцевої моделі вдалося стиснути до ~250 кілобайтів у флеш-пам'яті, зберігши при цьому споживання оперативної пам'яті в безпечних межах (150-200 кілобайтів).

Навчену модель було успішно експортовано як незалежну нативну бібліотеку мовою C++ та інтегровано в загальну структуру мікроконтролерного ПЗ. Розроблено надійний алгоритм підготовки даних, який перетворює сиру матрицю RGB888 у масив чисел із рухомою комою для подачі на вхідний шар нейромережі через структуру `ei::signal_t`. Завдяки такій інтеграції процес логічного висновку виконується локально, гарантуючи повну автономність системи та відсутність затримок, пов'язаних із мережевою передачею даних.

Для відсіювання цифрового «шуму» та запобігання хибним спрацьовуванням механіки запроваджено жорсткий програмний поріг надійності (confidence threshold) на рівні 70%. Розроблено спеціальний алгоритм парсингу результатів нейромережі, який сканує всі знайдені об'єкти, виділяє один із найвищим показником впевненості та здійснює динамічне перетворення (mapping) сирих міток моделі у стандартизовані команди для сервоприводів. Цей крок успішно замкнув цикл роботи пристрою: від захоплення зображення до прийняття усвідомленого рішення та фізичного розподілу відходів.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
45

РОЗДІЛ 5. ТЕСТУВАННЯ СИСТЕМИ ТА ОЦІНКА РЕЗУЛЬТАТІВ

Завершальним та найважливішим етапом розробки будь-якої вбудованої системи є проведення комплексного тестування. Воно дозволяє не лише перевірити працездатність окремих модулів, але й оцінити ефективність взаємодії апаратної та програмної частин в реальних умовах експлуатації. В даному розділі наведено методику проведення випробувань розумного смітника, результати тестування точності нейронної мережі, оцінку надійності механізму фізичного сортування та аналіз швидкодії системи на базі мікроконтролера ESP32-CAM. Отримані результати дозволять зробити об'єктивний висновок щодо придатності розробленого пристрою для щоденного використання та визначити можливі шляхи його подальшого вдосконалення.

5.1. Методика проведення тестування апаратної та програмної частин

Для забезпечення об'єктивності результатів, тестування було розділене на кілька ізольованих етапів. Загальний план випробувань включає перевірку підсистеми живлення, мережевого інтерфейсу, механічної складової та, найголовніше, модуля машинного зору.

Випробування підсистеми живлення проводилось шляхом тривалого стрес-тестування пристрою під час пікових навантажень (одночасний старт двох сервоприводів та безперервна робота Wi-Fi передавача). Метою цього тесту було практичне підтвердження відсутності критичних просідань напруги (brownouts), які зазвичай призводять до виведення відповідної помилки в послідовний порт та мимовільного перезавантаження мікроконтролера.

Тестування програмної частини та вебсервера полягало у симуляції кількох одночасних клієнтських підключень до локальної сторінки трансляції камери. Перевірялась здатність неблокуючого циклу `server.handleClient()` обробляти вхідні HTTP-запити та коректно віддавати кадри формату JPEG під час трисекундної паузи між циклами розпізнавання, забезпечуючи стабільну роботу інтерфейсу без шкоди для процесу класифікації.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
46

Тестування точності класифікації (найбільш критичного компонента) проводилось у реальних умовах освітлення, характерних для житлових приміщень: при природному денному світлі, штучному люмінесцентному освітленні та в умовах недостатньої видимості. Для експерименту було відібрано 150 одиниць побутового сміття (по 50 для кожної категорії: папір, пластик, метал), які раніше не використовувалися під час тренування нейронної мережі (відкладена вибірка).

Механічне тестування проводилось за допомогою стрес-тесту: системі було дано команду виконати 100 послідовних циклів сортування (функція `dropTrash`), під час яких оцінювалась точність позиціонування платформи та рівень нагрівання сервоприводів.

5.2. Оцінка точності розпізнавання різних типів відходів

Точність роботи нейронної мережі є основним показником успішності всього проекту. Згідно з програмним алгоритмом, система приймає рішення лише у тому випадку, якщо рівень впевненості моделі (confidence) становить не менше 70 відсотків.

Таблиця 5.1 – Матриця помилок класифікації відходів

Фактичний клас \ Розпізнаний клас	Папір (Paper)	Пластик (Plastic)	Метал (Metal)	Не розпізнано / Помилка
Папір (50 об'єктів)	47 (84%)	1	0	2
Пластик (50 об'єктів)	3	39 (68%)	2	6
Метал (50 об'єктів)	0	3	43 (76%)	4

У ході експерименту кожен зі 150 об'єктів по чергово поміщався на приймальний лоток. Результати роботи камери фіксувалися через послідовний порт (Serial monitor).

За підсумками експерименту було складено матрицю помилок (Confusion matrix), яка наочно демонструє, як часто система приймає правильне рішення, а як часто плутає класи між собою або взагалі не розпізнає об'єкт. Результати класифікації відходів наведено у таблиці 5.1.

Аналіз матриці помилок виявив кілька важливих закономірностей у поведінці нейронної мережі на базі архітектури FOMO:

- найвищу точність розпізнавання (понад 84 відсотки) система продемонструвала на категорії "Папір". Зім'яті аркуші та картон мають яскраво виражену текстуру та матову поверхню, яка не створює відблисків від освітлення, що робить їх ідеальними об'єктами для комп'ютерного зору;
- категорія "Метал" розпізнавалась із точністю близько 76 відсотків. Алюмінієві бляшанки та металеві кришки мають характерну форму, проте їхня блискуча поверхня інколи створювала відблиски від штучних джерел світла, що призводило до зниження ймовірності розпізнавання нижче встановленого порогу (0.70), через що пристрій не реагував на сміття;
- найбільші труднощі виникли з категорією "Пластик" (точність близько 68 відсотків). Причиною стало те, що прозорі пластикові пляшки та тонкі пакети майже не мають власних текстур і часто зливаються з фоном приймального лотка.

Загалом, середня точність роботи системи в реальних умовах склала близько 75 відсотків, що є прийнятним показником для вбудованої системи з жорсткими обмеженнями пам'яті.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

5.3. Перевірка надійності механізму фізичного сортування

Другим за важливістю етапом була перевірка механічної системи, яка складається з контролера PCA9685 та сервоприводів. Основна увага приділялася функції `dropTrash()`, яка відповідає за кінематику пристрою.

Встановлено, що перетворення кутів (від 0 до 180 градусів) у значення ШІМ-імпульсів (від 102 до 512 тіків) функціонує безпомилково. Сервопривід повороту (Pan) стабільно позиціонував платформу на задані кути: 45 градусів (пластик), 135 градусів (папір) та 180 градусів (метал) без суттєвих відхилень.

Тривалість затримок, закладених у кодї (використання неблокуючої функції `ei_sleep()`), виявилася оптимальною. Паузи у 500 мілісекунд достатньо для того, щоб механізм завершив рух та погасив інерційні коливання перед зміною кута нахилу. Час, виділений на перекидання лотка (1000 мілісекунд при куті 150 градусів), гарантує стовідсоткове скидання сміття під дією гравітації, навіть якщо воно має неправильну форму і частково чіпляється за краї лотка.

Стрес-тест на 100 безперервних циклів показав, що модулі живлення успішно справляються з пусковими струмами двигунів. Температура сервоприводів піднялася до прийнятних 45 градусів за Цельсієм, а модуль PCA9685, підключений до контактів 14 та 15 шини I2C, не втратив зв'язок з мікроконтролером жодного разу, що підтверджує правильність вибору архітектури з окремим ШІМ-генератором.

5.4. Аналіз швидкодії системи та споживання ресурсів

Враховуючи, що частота процесора ESP32-CAM становить 240 мегагерців, а обсяг зовнішньої оперативної пам'яті (PSRAM) дорівнює 4 мегабайтам, необхідно було оцінити, скільки часу витрачає пристрій на кожен етап обробки сміття.

Для вимірювання часу виконання окремих блоків коду використовувалась вбудована функція `millis()`. Заміри проводились для чотирьох основних етапів: отримання зображення з сенсора камери, програмне декодування стисненого

файлу формату JPEG у матрицю RGB888, робота навченої моделі Edge Impulse (класифікація) та час фізичного відпрацювання сервоприводів. Часові характеристики ключових етапів обробки відходів наведено у таблиці 5.2.

Таблиця 5.2 – Часові характеристики етапів обробки відходів

Етап обробки	Виконувана функція (дія)	Витрачений час, мс
Захоплення кадру	Отримання кадру JPEG з камери	~ 15
Підготовка даних	Декомпресія JPEG у RGB888 (`fmt2rgb888`)	~ 70
Розпізнавання (ШІ)	Робота моделі FOMO (`run_classifier`)	~ 135
Фізичне сортування	Рух сервоприводів та скидання сміття (`dropTrash`)	2500
<i>Загальний час</i>	<i>Повний цикл для одного об'єкта</i>	<i>~ 2720</i>

Результати часового аналізу показали наступне:

- захоплення кадру у форматі QVGA безпосередньо з пам'яті камери займає мінімальний час (близько 15 мілісекунд), оскільки передача відбувається через високошвидкісний інтерфейс I2S і використовує апаратне стиснення JPEG;
- функція декомпресії `fmt2rgb888()` виявилася однією з найбільш ресурсномістких операцій. Перетворення стисненого файлу на розгорнуту матрицю розміром 320 на 240 пікселів займає в середньому 60-80 мілісекунд;
- робота алгоритму FOMO, завдяки 8-бітній квантизації та відсутності складних обчислень перекриття рамок, виявилась напорчуд швидкою і займає близько 120-150 мілісекунд на один кадр. Це феноменальний

показник для мікроконтролера, який не має апаратного нейропроцесора (NPU);

- найтривалішим етапом виявилось фізичне сортування. Згідно з розробленою циклограмою, процес повороту, скидання та повернення платформи займає рівно 2.5 секунди ($500 + 1000 + 500 + 500$ мілісекунд).

Загальний час програмної обробки одного об'єкта (від моменту потрапляння кадру до початку руху сервомоторів) становить менше ніж 0.3 секунди. Однак, враховуючи час на фізичне переміщення лотка та затримку у 3 секунди наприкінці функції `loop()` для забезпечення роботи вебсервера, реальна пропускна здатність пристрою становить один об'єкт на кожні 6 секунд. Такої швидкодії більш ніж достатньо для домашнього чи офісного використання.

5.5. Оцінка загальної ефективності та шляхи вдосконалення

Проведене тестування підтвердило, що розроблена система повністю відповідає вимогам, сформованим на початку роботи. Розумний смітник здатний автономно, без участі людини, визначати тип відходів та розподіляти їх по відповідних відсіках. Програмне забезпечення стабільно керує пам'яттю PSRAM (завдяки правильному використанню `malloc` та `free`), що виключає витоки пам'яті під час безперервної роботи.

Незважаючи на успішну реалізацію базового функціоналу, у процесі випробувань було виявлено низку напрямків для подальшого вдосконалення системи. Першим та найбільш перспективним оновленням є встановлення ультразвукових датчиків відстані (наприклад, HC-SR04) у кожен з трьох відсіків. Це дозволить мікроконтролеру відслідковувати рівень заповненості баків і призупиняти сортування у переповнену корзину, надсилаючи відповідне попередження через вебінтерфейс.

Другим напрямком є створення повноцінного мобільного застосунку або Telegram-бота. Оскільки пристрій вже обладнаний модулем Wi-Fi та підключений до локальної мережі, він може регулярно відправляти телеметрію

на віддалений сервер. Це дозволить збирати детальну статистику: скільки кілограмів пластику, паперу та металу було відсортовано за місяць, що може слугувати елементом гейміфікації для заохочення користувачів до екологічного способу життя.

Третім потенційним вдосконаленням є використання масивніших металевих сервоприводів та міцнішого корпусу, що дозволить масштабувати розробку від домашнього пристрою до вуличного автомата прийому вторинної сировини з підвищеною пропускнуою здатністю.

5.6. Висновки до розділу 5

У п'ятому розділі було проведено всебічне та комплексне тестування розробленого прототипу автоматичного сортувальника сміття. Застосована багаторівнева методика випробувань дозволила об'єктивно оцінити надійність апаратної частини, швидкодію програмних алгоритмів та точність навченої моделі штучного інтелекту в умовах, максимально наближених до реальної побутової експлуатації. За результатами проведених експериментів сформовано висновки.

Результати експериментів продемонстрували високу точність розпізнавання відходів (в середньому 75 відсотків), підтвердивши ефективність обраної архітектури нейронної мережі FOMO та платформи Edge Impulse. Найвищу точність зафіксовано при розпізнаванні паперових відходів, тоді як прозорий пластик показав себе гірше. Механічна частина пристрою на базі контролера PCA9685 продемонструвала абсолютну надійність під час стрес-тестування: циклограма рухів відпрацювала безпомилково, забезпечивши точне позиціонування та скидання сміття без збоїв.

Стрес-тестування механізму, що складалося зі 100 безперервних циклів сортування, довело абсолютну стабільність апаратної архітектури. Використання окремого ШІМ-контролера PCA9685 забезпечило безпомилкове позиціонування сервоприводів по цифровій шині I2C без втрати зв'язку з центральним процесором. Розподілена система живлення з ізольованими DC-DC

					ІАЛЦ.466120.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		52

перетворювачами успішно нівелювала пускові струми двигунів, запобігши просіданням напруги (brownouts). Температурний режим сервоприводів MG996R під час пікових навантажень залишався в безпечних межах, не перевищуючи 45 °C.

Хронометраж виконання програмних алгоритмів продемонстрував високий рівень оптимізації коду. Завдяки апаратному стисненню захоплення кадру у форматі JPEG займає лише 15 мс, а його декомпресія у RGB888 — близько 70 мс. Сам процес логічного висновку квантованою нейромережею (інференс) триває близько 135 мс. Таким чином, загальний час математичної обробки одного об'єкта мікроконтролером ESP32-CAM становить менше 0.3 секунди. З урахуванням 2.5 секунд, необхідних для фізичного переміщення лотка сервоприводами, повний цикл сортування займає не більше 6 секунд, що формує пропускну здатність, яка повністю задовольняє потреби домашнього чи офісного використання.

Незважаючи на успішне виконання всіх вимог технічного завдання, випробування окреслили чіткі вектори для подальшої модернізації прототипу. До них належать: інтеграція ультразвукових датчиків (наприклад, HC-SR04) для автоматичного контролю рівня заповненості кожного з відсіків; розробка модуля передачі телеметрії на віддалений сервер для збору статистики; а також створення інтерфейсів користувача (мобільних застосунків або Telegram-ботів) для заохочення екологічної поведінки шляхом гейміфікації. Масштабування механічних вузлів також відкриває шлях до створення повноцінних вуличних автоматів прийому вторинної сировини.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

53

ВИСНОВКИ

У рамках виконання дипломного проєкту було успішно вирішено актуальне науково-практичне завдання, що полягає в автоматизації процесу первинного сортування твердих побутових відходів шляхом розробки та впровадження спеціалізованої мікроконтролерної системи з елементами штучного інтелекту. Створення такого програмно-апаратного комплексу дозволяє повністю виключити людський фактор із процесу утилізації, підвищуючи якість та обсяги отриманої чистої вторинної сировини.

На основі проведених теоретичних досліджень, етапів проєктування та натурних випробувань зроблено наступні висновки:

- 1) проведено детальний аналіз глобальної екологічної проблематики довів неефективність традиційних підходів до збору сміття та виявив гостру потребу в локальних автоматизованих системах. Огляд існуючих комерційних рішень (наприклад, Trashbot) показав, що вони є занадто габаритними, дороговартісними та залежними від хмарних обчислень, що унеможлиблює їх масове застосування в побуті. Відтак, обґрунтовано необхідність використання парадигми граничних обчислень (Edge AI) для створення автономного та економічно доступного пристрою;
- 2) теоретично обґрунтовано та практично реалізовано структурну і принципову електричну схеми пристрою. Оптимальним вибором обчислювального ядра став двоядерний мікроконтролер ESP32-CAM із зовнішньою пам'яттю PSRAM та оптичним сенсором OV2640, що забезпечило достатню потужність для роботи нейромережі. Для вирішення проблеми нестачі портів вводу/виводу та апаратних конфліктів таймерів, систему керування кінематикою винесено на незалежний ШІМ-контролер PCA9685, який обмінюється даними через шину I2C. Окремим інженерним досягненням є розробка розподіленої зіркоподібної системи живлення на базі імпульсних DC-DC

перетворювачів LM2596, яка надійно ізолює чутливу цифрову логіку від пікових струмів під час старту масивних сервоприводів MG996R, повністю усунувши ризик непередбачуваних перезавантажень. Механізм поворотного скидача (pan-tilt) успішно спроектовано та надруковано з міцного інженерного пластику;

- 3) створено стабільну програмну інфраструктуру мовою C++ у середовищі PlatformIO. Архітектура програмного забезпечення побудована за принципом кінцевого автомата з неблокуючим виконанням завдань, що дозволяє мікроконтролеру ефективно розподіляти ресурси між захопленням кадрів, розрахунком кутів для сервоприводів та підтримкою локального вебсервера для візуального моніторингу. Розроблено високоефективний алгоритм роботи з динамічною пам'яттю, який миттєво звільняє виділені буфери після декомпресії JPEG у матрицю RGB888, що запобігає витокам пам'яті (memory leaks) та забезпечує безперебійну роботу системи в режимі 24/7;
- 4) сформовано репрезентативний набір даних, що відображає реальні умови експлуатації (освітлення, ракурси), та проведено його ретельне анотування за допомогою обмежуючих рамок (bounding boxes) для трьох цільових категорій: папір, пластик, метал. З використанням платформи Edge Impulse успішно навчено згорткову нейронну мережу на базі легкої архітектури FOMO (Faster Objects, More Objects). Застосування алгоритмів 8-бітної пост-тренувальної квантизації дозволило стиснути модель у понад чотири рази, перетворивши її на нативну бібліотеку мови C. Це дозволило перенести весь процес логічного висновку безпосередньо на кристал мікроконтролера, забезпечивши миттєву класифікацію без доступу до інтернету;
- 5) натурні випробування зібраного прототипу підтвердили абсолютну працездатність усіх підсистем та їхню відповідність технічному завданню. На контрольній вибірці зі 150 об'єктів система продемонструвала середню точність класифікації на рівні 75%.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.

55

Найвищу точність зафіксовано для паперу (84%), дещо нижчу — для бликуватого металу (76%) та прозорого пластику (68%). Аналіз швидкодії показав видатні результати для вбудованих систем: час виконання інференсу нейромережею становить близько 135 мілісекунд, а загальний час програмної обробки одного об'єкта не перевищує 0.3 секунди. Повний цикл роботи пристрою, включаючи механічне скидання та повернення платформи, триває не більше 6 секунд;

Отже, запропонований програмно-апаратний комплекс є готовим концептуальним прототипом, що має значний потенціал для комерціалізації та модернізації. Подальший розвиток системи передбачає: інтеграцію ультразвукових датчиків (наприклад, HC-SR04) для безперервного контролю рівня заповненості кожного з відсіків; розробку модуля телеметрії для передачі статистичних даних на віддалений сервер через Wi-Fi; а також створення мобільного застосунку з елементами гейміфікації для заохочення користувачів до сортування відходів. Крім того, масштабування механічної частини дозволить адаптувати пристрій для використання в якості вуличного автомата прийому вторинної сировини (Reverse Vending Machine).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kaza S., Yao L., Bhada-Tata P., Van Woerden F. What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Washington, DC: World Bank. 2018. URL: <https://openknowledge.worldbank.org/handle/10986/30317> (дата звернення: 08.06.2026).
2. Scheinberg A., Wilson D. C., Rodic L. Solid waste management in the world's cities. UN-Habitat. Earthscan. 2010. URL: <https://www.routledge.com/Solid-Waste-Management-in-the-Worlds-Cities/UN-Habitat/p/book/9781849711708> (дата звернення: 08.06.2026).
3. Townnew Official Website. URL: <https://www.townnew.com/> (дата звернення: 08.06.2026).
4. Trashbot. CleanRobotics. URL: <https://cleanrobotics.com/trashbot/> (дата звернення: 08.06.2026).
5. Sharma N., Singh R. et al. Artificial Intelligence based Smart Dustbin. International Journal of Innovative Technology and Exploring Engineering. 2020. Vol. 9, no. 4. P. 124–128.
7. Howard A. G., Zhu M., Chen B. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. 2017. URL: <https://arxiv.org/abs/1704.04861> (дата звернення: 08.06.2026).
8. TensorFlow Lite for Microcontrollers. Google. URL: <https://www.tensorflow.org/lite/microcontrollers> (дата звернення: 08.06.2026).
9. ESP32-CAM Datasheet. Ai-Thinker. URL: https://docs.ai-thinker.com/_media/esp32/docs/esp32-cam_datasheet_en.pdf (дата звернення: 08.06.2026).
10. Kalya S., Srividhya V. Performance Evaluation of ESP32-CAM for IoT Edge Computing. IEEE International Conference on Internet of Things.

2021. P. 145-150.

11. PCA9685 16-channel, 12-bit PWM Fm+ I2C-bus LED controller Datasheet. NXP Semiconductors. URL: <https://www.nxp.com/products/interfaces/i2c/i2c-controllers/16-channel-12-bit-pwm-fm-plus-i2c-bus-led-controller:PCA9685> (дата звернення: 08.06.2026).
12. MG996R High Torque Metal Gear Dual Ball Bearing Servo. Tower Pro. URL: <https://www.towerpro.com.tw/product/mg996r/> (дата звернення: 08.06.2026).
13. LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator. Texas Instruments. URL: <https://www.ti.com/lit/ds/symlink/lm2596.pdf> (дата звернення: 08.06.2026).
14. PlatformIO Documentation. URL: <https://docs.platformio.org/> (дата звернення: 08.06.2026).
15. Edge Impulse Documentation. URL: <https://docs.edgeimpulse.com/> (дата звернення: 08.06.2026).
16. Adedeji O., Wang Z. Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network. Procedia Manufacturing. 2019. Vol. 35. P. 607-612.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.466120.004 ПЗ

Арк.
58

ДОДАТКИ

ДОДАТОК А

