

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

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

До захисту допущено:

Зав. кафедри

_____ Андрій МОВЧА-
НЮК

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньою програмою «Інтелектуальні технології радіоелектронної те-
хніки», за спеціальністю 172 «Телекомунікації та радіотехніка»**

**на тему: Програмно-апаратний комплекс для дистанційного монітори-
нгу поведінки домашніх тварин**

Виконав (-ла):

студент (-ка) IV курсу, групи РЕ-22

Стаднік Єлизавета Валеріївна

Прізвище, ім'я, по батькові

Керівник: ст. викладач Адаменко Володимир Олексійович

Посада, науковий ступінь, вчене звання

Прізвище, ім'я, по батькові

Рецензент: доц. к.т.н. каф. РІ, Гусєва Олена Володимирівна

Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові

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

Студент (-ка) _____

Київ – 2026 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/П	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	PE22.404474.001.ТЗ	Завдання на дипломний проєкт	2	
2	A4	PE22.404474.001 ПЗ	Пояснювальна записка	77	
3	A4	PE22.404474.001 Е1	Схема структурна	1	
4	A4	PE22.404474.001 Д1	Схема алгоритму роботи мікроконтролера	1	
5	A4	PE22.404474.001 Д2	Схема алгоритму серверного аналізу даних	1	

				PE22.404474.001		
	ПІБ	Підп.	Дата	Програмно-апаратний комплекс для дистанційного моніторингу поведінки домашніх тварин	Лист	Листів
Розробн.	Стаднік Є.В.				1	1
Керівн.	Адаменко В.О.				КПІ ім. Ігоря Сікорського Каф.ПРЕ, Гр. РЕ-22	
Н/контр.						
Зав.каф.	Мовчанюк А. В					

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 172 «Телекомунікації та радіотехніка»

Освітня програма «Інтелектуальні технології радіоелектронної техніки»

ЗАТВЕРДЖУЮ

Зав. кафедри

_____ Андрій МОВЧА-

НЮК

« ____ » _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студента

Стаднік Єлизаветі Валеріївні

1. Тема проєкту «Програмно-апаратний комплекс для дистанційного моніторингу поведінки домашніх тварин»

2. Керівник проєкту Адаменко Володимир Олексійович ст. викладач, затверджені наказом по університету від « 29» травня 2026 р. №1989-с

Строк подання студентом проєкту 08 червня 2026 року

3. Вихідні дані до проєкту: Напруга живлення від +4,4 до +5,25 В, споживаний струм до 0,1 А, автономне живлення, наявність засобів бездротового підключення до мережі Інтернет.

4. Зміст пояснювальної записки 1) Огляд існуючих рішень та постановка задачі; 2) Розробка структури та апаратної частини комплексу; 3) Розробка програмного забезпечення та алгоритмів; 4) Експериментальні дослідження комплексу в реальних умовах.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): схема електрична структурна комплексу, схема

алгоритму роботи мікроконтролера, схема алгоритму серверного аналізу даних, презентація (10 слайдів).

6. Дата видачі завдання 14 квітня 2026 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд існуючих рішень та постановка задачі	14.04.26-06.05.26	
2	Розробка структури комплексу та вибір елементної бази	07.05.26-10.05.26	
3	Розробка апаратної частини	11.05.26-18.05.26	
4	Розробка вбудованого програмного забезпечення мікроконтролера	19.05.26-23.05.26	
5	Розробка серверної частини та алгоритму аналізу даних	24.05.26-29.05.26	
6	Експериментальні дослідження комплексу	30.05.26-05.06.26	
7	Оформлення текстової та графічної документації	06.06.26-08.06.26	

Студент

Єлизавета СТАДНІК

Керівник

Володимир АДАМЕНКО

АНОТАЦІЯ

Пояснювальна записка до дипломного проєкту: «Програмно-апаратний комплекс для дистанційного моніторингу поведінки домашніх тварин».

Обсяг пояснювальної записки — 77 сторінок, 12 рисунків, 3 таблиці, 13 джерел, 4 додатки.

Об'єкт дослідження — процес дистанційного моніторингу фізіологічних та поведінкових показників домашніх тварин.

Предмет дослідження — апаратно-програмні засоби автоматичного вимірювання маси та частоти відвідувань лотка на основі тензометрії та технологій інтернету речей.

Мета роботи — розробка автономного комплексу для автоматичного моніторингу фізіологічних показників тварини з передаванням даних на сервер та сповіщенням користувача про відхилення у стані здоров'я тварини.

У роботі обґрунтовано структуру комплексу та розроблено його апаратну частину на базі мікроконтролера ESP32-S2 і тензодатчиків (АЦП HX711). Створено енергоощадне вбудоване програмне забезпечення з алгоритмом динамічного зважування, а також серверну частину з Telegram-ботом. Експериментальні дослідження в реальних умовах підтвердили працездатність пристрою. Результати придатні для створення побутових систем моніторингу здоров'я тварин.

КЛЮЧОВІ СЛОВА: МОНІТОРИНГ ДОМАШНІХ ТВАРИН, ТЕНЗОМЕТРІЯ, ESP32, ІНТЕРНЕТ РЕЧЕЙ, ВИМІРЮВАННЯ МАСИ, TELEGRAM-БОТ.

ANNOTATION

Explanatory note to the diploma project: "Hardware and software complex for remote pet behavior monitoring".

The explanatory note consists of 77 pages, 12 figures, 3 tables, 13 references, and 4 appendices.

The object of the study is the process of remote monitoring of physiological and behavioral indicators of pets.

The subject of the study is the hardware and software tools for contactless measurement of weight and litter box visit frequency based on strain gauge technology and the Internet of Things.

The aim of the work is to develop an autonomous complex for automatic monitoring of an animal's physiological indicators, with data transmission to a server and user notification about deviations in its health.

The work substantiates the structure of the complex and develops its hardware based on the ESP32-S2 microcontroller and strain gauges (HX711 ADC). Power-saving embedded software with a dynamic weighing algorithm was created, alongside a server backend with a Telegram bot. Experimental testing under real conditions confirmed the device's operability. The results can be used to create affordable household pet health monitoring systems.

KEYWORDS: PET MONITORING, STRAIN GAUGE, ESP32, INTERNET OF THINGS, WEIGHT MEASUREMENT, TELEGRAM BOT.

**Пояснювальна записка
до дипломного проєкту**

на тему: «Програмно-апаратний комплекс для дистанцій-
ного моніторингу поведінки домашніх тварин»

Київ — 2026 року

ЗМІСТ

Перелік скорочень.....	3
Вступ.....	4
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАДАЧІ	7
1.1 Проблема моніторингу стану здоров'я домашніх тварин	7
1.2 Огляд існуючих рішень на ринку	9
1.3 Методи вимірювання маси.....	11
1.4 Обґрунтування власного підходу та постановка задачі.....	12
2 РОЗРОБКА СТРУКТУРИ ТА АПАРАТНОЇ ЧАСТИНИ КОМПЛЕКСУ	15
2.1 Розробка схеми електричної структурної комплексу	15
2.2 Обґрунтування вибору елементної бази	17
2.2.1 Мікроконтролер	17
2.2.2 Тензометричні датчики	19
2.2.3 Аналого-цифровий перетворювач.....	21
2.2.4 Підсистема живлення	22
2.3 Принцип роботи вимірювального тракту.....	26
2.4 Апаратна частина	29
2.5 Розрахункова частина.....	30
2.5.1 Калібрування вимірювального тракту	30
2.5.2 Оцінний розрахунок часу автономної роботи	31
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АЛГОРИТМІВ .	34
3.1 Загальна архітектура програмного забезпечення	34

					РПЗ1. 404474.001 ПЗ		
ЗМ.	Лист	№ докум.	Підпис	Дата			
Розробив		Стаднік Є.В.			Програмно - апаратний комплекс для дистанційного моніторингу поведінки домашніх	Літ.	Лист
Перевірів		Адаменко В.О					1
Н. Контр.						РЕ-22, РТФ	
Затвер-							

3.2 Вбудоване програмне забезпечення мікроконтролера	35
3.2.1 Загальний алгоритм роботи	35
3.2.2 Енергоощадне керування	36
3.2.3 Алгоритм детекції відвідування та визначення маси.....	37
3.2.4 Захисні механізми	39
3.3 Протокол передавання даних.....	39
3.4 Серверна частина	41
3.4.1 Стек та середовище виконання.....	41
3.4.2 Приймання та зберігання даних	41
3.4.3 Алгоритм аналізу даних	42
3.4.4 Інтерфейс взаємодії з користувачем	44
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	47
4.1 Методика та умови проведення досліджень	47
4.2 Калібрування вимірювального тракту	48
4.3 Дослідження роботи в реальних умовах	49
4.3.1 Зниження результату вимірювання внаслідок заповування..	49
4.3.2 Хибне подвоєння відвідувань через повторні підходи	50
4.4 Перевірка передавання даних та інтерфейсу користувача	51
4.5 Перевірка роботи серверного аналізу даних.....	51
Висновки	55
Додаток А.....	59
Додаток Б	60
Додаток В.....	66
Додаток Г	67

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП — аналого-цифровий перетворювач

ПЗ — програмне забезпечення

ТЗ — технічне завдання

FLUTD — Feline Lower Urinary Tract Disease

GPIO — General-Purpose Input/Output

HTTP — HyperText Transfer Protocol

JSON — JavaScript Object Notation

PGA — Programmable Gain Amplifier

RTC — Real-Time Clock

SPST — Single Pole Single Throw

SoC — System on a Chip

					РЕ22.404474.001 ПЗ	Лис
						3
Зм.	Лис	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Кількість домашніх тварин у світі постійно зростає, а разом із нею зростає й потреба у догляді за їхнім здоров'ям. Однією з ключових проблем такого догляду є те, що тварини не здатні повідомити про погіршення самопочуття, а власник у побутових умовах не завжди може вчасно помітити ранні ознаки захворювання. Водночас саме раннє виявлення є вирішальним для успішного лікування багатьох хвороб.

Двома об'єктивними показниками, що дозволяють виявити погіршення стану тварини на ранній стадії, є маса тіла та частота відвідувань лотка. Поступова втрата маси нерідко є одним із найперших проявів хронічних захворювань: за даними ветеринарних досліджень, у котів із хронічною хворобою нирок втрата маси може починатися за кілька років до встановлення діагнозу, а середня втрата за рік до діагнозу сягає близько 9 % [4] маси тіла. Тому стійке зниження маси тіла понад 10 % вважається клінічно значущою ознакою, що потребує звернення до ветеринара. Так само важливою діагностичною ознакою є зміна частоти сечовипускання: як її збільшення, так і зменшення є характерними для захворювань сечовидільної системи.

Проблема полягає в тому, що в домашніх умовах власник практично не має змоги об'єктивно й систематично відстежувати ці показники. Регулярне зважування дрібної тварини є складним і стресовим для неї, а підрахунок відвідувань лотка вручну неможливий. Наявні комерційні рішення здебільшого або потребують активної участі власника, або є дорогими пристроями із закритою функціональністю. Тому розробка доступного пристрою, який без участі власника відстежує масу тварини та частоту відвідувань лотка, автоматично аналізує ці дані й сповіщає про відхилення, є актуальною науково-практичною задачею.

Мета роботи — розробка програмно-апаратного комплексу для дистанційного моніторингу поведінки домашніх тварин, що забезпечує автоматичне вимірювання маси тварини та частоти відвідувань лотка, передавання даних

					РЕ22.404474.001 ПЗ	Лис
						4
Зм.	Лис	№ докум.	Підпис	Дата		

на сервер, їх аналіз і сповіщення користувача про відхилення, які можуть свідчити про погіршення стану здоров'я.

Для досягнення поставленої мети у роботі розв'язано такі завдання:

1. Проаналізовано предметну область — діагностично значущі поведінкові та фізіологічні показники домашніх тварин і наявні технічні рішення для їх моніторингу.

2. Розроблено структурні схеми апаратної та програмної частин комплексу й обґрунтовано вибір елементної бази апаратного вузла.

3. Розроблено апаратну частину пристрою — вимірювальний тракт на основі тензометричних датчиків для автоматичного зважування тварини під час відвідування лотка.

4. Розроблено вбудоване програмне забезпечення мікроконтролера, що реалізує енергоощадне вимірювання маси, фільтрацію сигналу в умовах динамічних навантажень і передавання даних.

5. Розроблено серверну частину комплексу для приймання, зберігання та аналізу даних і інтерфейс взаємодії з користувачем.

6. Проведено експериментальні дослідження комплексу в реальних умовах, у ході яких виявлено та усунено практичні проблеми його роботи.

Об'єкт дослідження — процес дистанційного моніторингу фізіологічних та поведінкових показників домашніх тварин.

Предмет дослідження — методи та засоби автоматичного вимірювання маси тварини й частоти відвідувань лотка на основі тензометрії з бездротовою передачею та автоматичним аналізом даних.

Методи дослідження. У роботі застосовано тензометричний метод вимірювання маси, методи цифрової обробки сигналів для фільтрації показань в умовах динамічних навантажень, методи проектування вбудованих та енергоощадних систем, а також натурний експеримент для перевірки працездатності комплексу в реальних умовах.

					РЕ22.404474.001 ПЗ	Лис
						5
Зм.	Лис	№ докум.	Підпис	Дата		

Практичне значення одержаних результатів. Розроблений комплекс дозволяє власнику тварини в автоматичному режимі, без додаткових дій і без стресу для тварини, відстежувати її масу та частоту відвідувань лотка й вчасно отримувати попередження про відхилення, що можуть свідчити про захворювання. Це сприяє ранньому виявленню проблем зі здоров'ям і своєчасному зверненню до ветеринара. Пристрій побудований на доступній елементній базі, має низьку собівартість і не потребує спеціального обслуговування, що робить його придатним для широкого застосування в домашніх умовах. Розроблений пристрій орієнтований на домашніх тварин загалом.

Апробація результатів. Розроблений комплекс реалізовано у вигляді діючого макета та випробувано в реальних умовах на прикладі домашнього kota. Підтверджено працездатність усіх основних функцій комплексу: вимірювання маси, детекції відвідувань, передавання даних на сервер, їх зберігання та аналізу з формуванням сповіщень користувачу.

					РЕ22.404474.001 ПЗ	Лис
						6
Зм.	Лис	№ докум.	Підпис	Дата		

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Проблема моніторингу стану здоров'я домашніх тварин

Своєчасне виявлення захворювань домашніх тварин є важливою складовою відповідального утримання. Як об'єкт для експериментальних досліджень у межах роботи обрано домашнього kota. Такий вибір зумовлений кількома причинами: коти є одними з найпоширеніших домашніх тварин; вони привчені користуватися лотком, що робить зважування під час відвідування лотка природним і непомітним для тварини; для котів характерна висока поширеність урологічних захворювань із добре задокументованими діагностичними маркерами. Подальший розгляд проблеми ведеться на прикладі kota, проте запропоновані рішення мають загальний характер і застосовні до інших дрібних домашніх тварин невеликої маси — дрібних порід собак, декоративних кроликів, міні-пігів тощо.

Контроль за станом здоров'я котів ускладнюється фізіологічною особливістю цього виду — здатністю приховувати ознаки нездужання. Як хижаки за походженням, коти інстинктивно маскують слабкість, тому власник часто помічає захворювання вже на пізній стадії, коли тварина потребує тривалого й дорогого лікування. У зв'язку з цим особливого значення набувають об'єктивні фізіологічні показники, які можна відстежувати безперервно, не покладаючись на видимі поведінкові ознаки.

Одним із найінформативніших таких показників є маса тіла. Невмотивована втрата ваги нерідко є однією з перших ознак прихованого захворювання, і її раннє виявлення суттєво підвищує ймовірність успішного лікування. Ветеринарні джерела наголошують, що невимотивована втрата ваги має насторожувати власника й нерідко є раннім симптомом серйозних захворювань, оскільки може супроводжувати хворобу навіть за збереженого апетиту [3]. Тому регулярне зважування й контроль динаміки маси є базовим інструментом раннього виявлення відхилень; зокрема рекомендується зважувати kota щомісяця, адже

					РЕ22.404474.001 ПЗ	Лис
						7
Зм.	Лис	№ докум.	Підпис	Дата		

за щоденного спостереження поступову зміну маси помітити важко [3]. Особливо небезпечним є те, що в багатьох випадках втрата ваги протягом тривалого часу залишається єдиним симптомом, тоді як інші ознаки відсутні.

Окрему групу поширених захворювань становлять патології нижніх сечовивідних шляхів котів (Feline Lower Urinary Tract Disease, FLUTD) — збірне поняття, що охоплює захворювання сечового міхура та уретри. За даними клінічних досліджень, частка котів із ознаками FLUTD серед тварин, які звертаються до ветеринарних клінік, становить від 1,5 % до 8 % залежно від регіону та вибірки [1]. Структура причин FLUTD за результатами ретроспективного аналізу німецької популяції котів (302 тварини) наведена на рисунку 1.1 [2]. Як видно з діаграми, домінуючою причиною є ідіопатичний цистит — на нього припадає 55 % випадків, що більш ніж удвічі перевищує сумарну частку решти діагнозів. Клінічно ці стани проявляються частим і болісним сечовипусканням, помітною зміною режиму користування лотком, а в найтяжчих випадках — повною непрохідністю уретри, що є невідкладним станом, небезпечним для життя тварини.

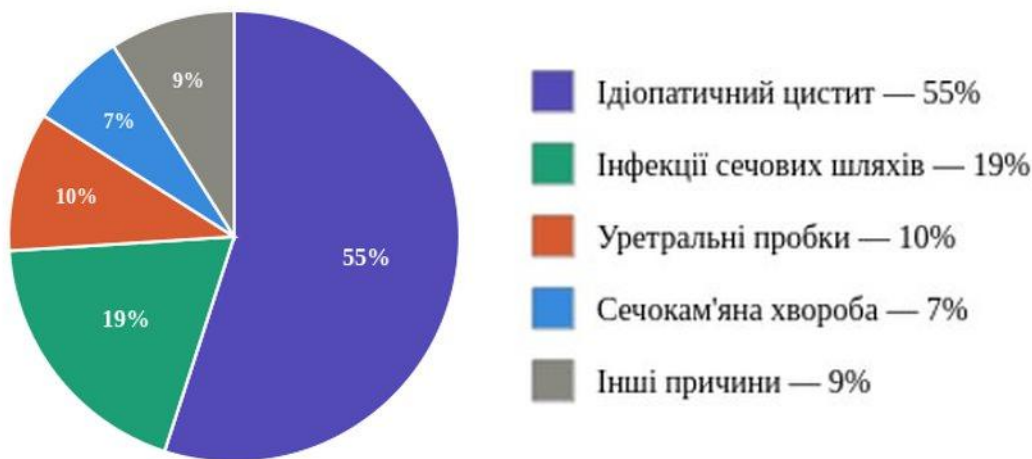


Рисунок 1.1 — Структура причин FLUTD у котів [4]

Таким чином, інформативними діагностичними маркерами стану здоров'я kota є маса тіла та частота відвідувань лотка. Поступова втрата ваги або помітна зміна частоти походів дають змогу запідозрити захворювання на ранній стадії. Проте на практиці власник стикається з низкою труднощів у відстеженні цих показників:

- **Стрес при ручному зважуванні.** Регулярне зважування тварини на побутових вагах є складним через природну реакцію: тварина виривається, чинить опір, що спотворює результат і викликає стрес як у тварини, так і у власника.
- **Неможливість контролю частоти відвідувань.** Власник фізично не може цілодобово фіксувати, скільки разів тварина відвідує лоток, хоча саме ці дані є ключовими для виявлення урологічних розладів.
- **Зайнятість і відсутність систематичності.** Ведення ручної статистики потребує дисципліни й часу, яких у власника, зайнятого роботою, зазвичай немає, тому спостереження має епізодичний характер.
- **Суб'єктивність оцінки.** Візуальна оцінка стану тварини є неточною: незначні, але діагностично важливі зміни маси чи поведінки залишаються непоміченими.

Наведені труднощі обґрунтовують потребу в автоматизованій системі, яка без участі власника й без стресу для тварини, фіксувала б ключові фізіологічні та поведінкові показники й сигналізувала про відхилення від норми.

1.2 Огляд існуючих рішень на ринку

Для визначення місця власної розробки було проаналізовано наявні на ринку засоби, які тією чи іншою мірою вирішують задачу контролю стану домашніх тварин. Умовно їх можна поділити на два класи: побутові ваги для тварин та автоматизовані («розумні») лотки.

Побутові ваги для тварин — це електронні ваги з платформою, розраховані на зважування тварин. Їх перевагами є доступна ціна та висока точність вимірювання. Однак вони мають принципові недоліки в контексті поставленої

					РЕ22.404474.001 ПЗ	Лис
						9
Зм.	Лис	№ докум.	Підпис	Дата		

задачі: зважування є ручним і потребує безпосередньої участі власника, що спричиняє стрес у тварини; ваги не накопичують статистику й не відстежують динаміку показників; вони не дають жодної інформації про частоту відвідувань лотка, тобто не охоплюють важливий діагностичний маркер урологічних захворювань.

Автоматизовані («розумні») лотки. Це складні пристрої з функціями автоматичного прибирання, які додатково можуть вести статистику відвідувань у мобільному застосунку. Їхньою перевагою є автоматизація догляду та збір даних без участі власника. Водночас вони мають суттєві обмеження: висока вартість (від 400 доларів США та вище), значні габарити та шум під час роботи механізму самоочищення, а також закрита конструкція. Останнє є особливо критичним, оскільки закритий простір із рухомими механічними частинами відлякує значну частину котів, через що тварина може відмовитися користуватися таким лотком.

Порівняльний аналіз двох класів пристроїв наведено в таблиці 1.1.

Таблиця 1.1 — Порівняльний аналіз існуючих рішень

Критерій	Побутові ваги	«Розумні» лотки
Вартість	від ~ 1 000 грн	від ~10 000 грн
Участь власника	Потрібна (ручне зважування)	Не потрібна
Стрес для тварини	Високий	Можливий
Збір статистики	Відсутній	Наявний
Контроль частоти відвідувань	Відсутній	Наявний

Аналіз показує, що жоден із наявних класів пристроїв не вирішує поставлену задачу комплексно. Побутові ваги забезпечують точність, але не дають

автоматизації та статистики; «розумні» лотки автоматизують збір даних, але є дорогими, габаритними та некомфортними для тварини. Таким чином, на ринку існує незайнята ніша — недорогий, компактний і непомітний для тварини пристрій, який автоматично відстежував би масу тіла й частоту відвідувань лотка та передавав статистику власнику.

1.3 Методи вимірювання маси

Оскільки центральним вимірюваним параметром системи є маса тіла тварини, було розглянуто можливі методи її визначення. Найпоширенішим для побутових вимірювальних систем є метод на основі тензометричних датчиків.

Принцип дії тензометричного датчика ґрунтується на ефекті зміни електричного опору провідника під час його механічної деформації. Конструктивно датчик являє собою пружний елемент, на якому закріплено тензорезистори — тонкоплівкові резистивні елементи. Під дією прикладеної сили пружний елемент деформується, що спричиняє зміну довжини й, відповідно, опору тензорезисторів. Ця зміна опору пропорційна прикладеному навантаженню [5].

Для перетворення малої зміни опору в електричний сигнал тензорезистори вмикають за схемою мостової вимірювальної схеми Вітстона. Чотири плеча моста (повномостова схема) утворюються тензорезисторами, причому при деформації одні з них розтягуються, а інші стискаються. У стані спокою, коли навантаження відсутнє, міст збалансований і його вихідна напруга дорівнює нулю. Під дією навантаження баланс порушується, і на виході моста виникає диференціальна напруга, пропорційна прикладеній силі. Типова чутливість тензометричних датчиків становить порядку 1–3 мВ на кожен вольт напруги живлення моста, тобто вихідний сигнал має рівень одиниць — десятків мілівольт і потребує подальшого підсилення та оцифрування.

					РЕ22.404474.001 ПЗ	Лис
						11
Зм.	Лис	№ докум.	Підпис	Дата		

Перевагами тензометричного методу є висока точність і лінійність, механічна простота, низька вартість і широка доступність датчиків різних діапазонів навантаження, а також можливість непомітного для тварини розміщення під платформою. Саме тому цей метод обрано як основу для розроблюваної системи. Альтернативні методи (наприклад, оптичні чи ємнісні) є складнішими в реалізації, дорожчими або менш придатними для побутового застосування з тваринами, тому в межах роботи вони не розглядалися як основні.

З огляду на те, що корисний сигнал тензомоста є слабким, для його обробки необхідний спеціалізований аналого-цифровий перетворювач (АЦП) з високою роздільною здатністю та вбудованим підсилювачем, орієнтований саме на роботу з тензодатчиками.

1.4 Обґрунтування власного підходу та постановка задачі

На основі проведеного аналізу проблеми (п. 1.1), огляду існуючих рішень (п. 1.2) та методів вимірювання (п. 1.3) сформульовано концепцію власної розробки — програмно-апаратного комплексу для дистанційного моніторингу поведінки домашніх тварин.

Запропоноване рішення являє собою не самостійний лоток, а вимірювальну платформу, що встановлюється під будь-який стандартний лоток користувача. Такий підхід забезпечує низку переваг порівняно з наявними рішеннями:

- **Універсальність.** Платформа сумісна з будь-яким стандартним лотком, який уже використовує власник, і не потребує його заміни.
- **Низький профіль і комфорт для тварини.** Конструкція має малу висоту й не містить закритих стінок чи рухомих механізмів, які лякають тварину, тому не змінює звичних умов користування лотком.
- **Інтеграція без додаткового програмного забезпечення.** Дані автоматично передаються користувачу у вигляді повідомлень у Telegram-бот, без потреби встановлювати окремий мобільний застосунок.

					РЕ22.404474.001 ПЗ	Лис
						12
Зм.	Лис	№ докум.	Підпис	Дата		

Принцип роботи комплексу полягає в автоматичному зважуванні під час відвідувань лотка: платформа з тензодатчиками фіксує масу тіла тварини під час відвідування лотка, а також самий факт відвідування, що дає змогу обчислювати частоту таких подій. Зібрані дані обробляються мікроконтролером, передаються бездротовим каналом на сервер, накопичуються й аналізуються на предмет відхилень від індивідуальної норми тварини, після чого власник отримує повідомлення. У межах роботи апаратну реалізацію та експериментальні дослідження комплексу виконано для домашнього kota як обраного об'єкта апробації; водночас закладені технічні рішення мають універсальний характер і можуть бути застосовані для інших дрібних домашніх тварин відповідної маси.

Виходячи з цього, метою роботи є розробка програмно-апаратного комплексу для дистанційного моніторингу маси тіла та частоти відвідувань лотка домашньою твариною з метою раннього виявлення можливих відхилень у стані її здоров'я.

З урахуванням вихідних даних завдання до пристрою висуваються такі основні технічні вимоги: напруга живлення в діапазоні від 4,4 до 5,25 В, споживаний струм до 0,1 А, наявність засобів бездротового підключення до мережі Інтернет.

Висновки до розділу 1

1. Обґрунтовано актуальність задачі автоматизованого моніторингу стану здоров'я домашніх тварин на прикладі kota як обраного об'єкта апробації. Показано, що маса тіла та частота відвідувань лотка є інформативними діагностичними маркерами, а їх раннє відстеження ускладнюється здатністю котів приховувати захворювання та практичними труднощами ручного спостереження.
2. Проведено огляд наявних на ринку рішень. Встановлено, що побутові ваги забезпечують точність, але не дають автоматизації та статистики, а «розумні» лотки автоматизують збір даних, проте є

					PE22.404474.001 ПЗ	Лис
						13
Зм.	Лис	№ докум.	Підпис	Дата		

дорогими, габаритними й некомфортними для тварини. Виявлено незайняту ринкову нішу для недорогого компактного пристрою автоматичного моніторингу.

3. Розглянуто методи вимірювання маси й обґрунтовано вибір тензометричного методу на основі мостової схеми Вітстона як оптимального за критеріями точності, вартості та простоти реалізації. Визначено, що слабкий сигнал тензомоста потребує спеціалізованого АЦП високої роздільної здатності.
4. Сформульовано концепцію власної розробки — вимірювальної платформи під стандартний лоток із передаванням даних у Telegram-бот, визначено мету, задачі роботи та основні технічні вимоги до пристрою, що є підставою для подальшого проєктування.

					РЕ22.404474.001 ПЗ	Лис
						14
Зм.	Лис	№ докум.	Підпис	Дата		

2 РОЗРОБКА СТРУКТУРИ ТА АПАРАТНОЇ ЧАСТИНИ КОМПЛЕКСУ

2.1 Розробка схеми електричної структурної комплексу

На основі сформульованих у Розділі 1 мети та задач, а також технічних вимог до пристрою, було розроблено структуру апаратної та програмної частин комплексу. Розроблені схеми наведені на рисунках 2.1 та 2.2.

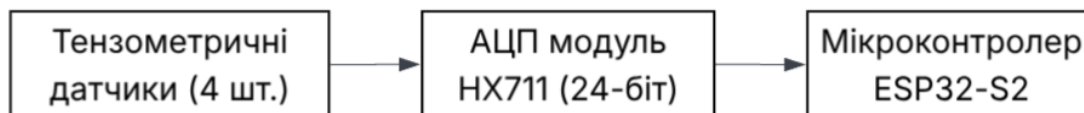


Рисунок 2.1 — Структурна схема апаратної частини



Рисунок 2.2 — Структурна схема програмної частини

Комплекс має розподілену архітектуру і функціонально поділяється на дві частини: апаратний пристрій, розміщений безпосередньо під лотком тварини, та серверну частину, що працює у хмарі. Зв'язок між цими частинами здійснюється каналом бездротового зв'язку Wi-Fi через мережу Інтернет.

Призначення кожного функціонального блоку структурної схеми розглянуто нижче.

Тензометричні датчики (4 шт.) — первинні перетворювачі, що сприймають масу тіла тварини та перетворюють її на аналоговий електричний сигнал. Чотири датчики розміщуються під ніжками платформи й електрично з'єднуються у повну мостову вимірювальну схему (схема Вітстона), завдяки чому корисний сигнал є диференціальним і пропорційним прикладеній силі. Принцип дії та схема з'єднання тензодатчиків детально розглядаються у п. 2.3.

АЦП НХ711 — спеціалізований 24-розрядний аналого-цифровий перетворювач, орієнтований саме на роботу з тензомостами. Він приймає на свій диференціальний вхід слабкий аналоговий сигнал від мосту датчиків, підси-

лює його за допомогою вбудованого програмованого підсилювача та перетворює на цифровий код, який далі передається мікроконтролеру по двопровідному послідовному інтерфейсу. Обґрунтування вибору цього АЦП наведене у п. 2.2.3.

Мікроконтролер ESP32 — обчислювальне ядро пристрою. Він керує процесом вимірювання, виконує алгоритми фільтрації сигналу, формує пакет даних і передає його каналом Wi-Fi на сервер. Окрім обчислювальних задач, мікроконтролер забезпечує енергоощадне керування всім пристроєм через режим глибокого сну (Deep Sleep). Вибір ESP32 обґрунтовано у п. 2.2.1, а реалізовані на ньому алгоритми описані у розділі 3.

Сервер — програмний компонент серверної частини, що приймає вхідні запити від мікроконтролера. Передача даних здійснюється за протоколом HTTP методом POST у форматі JSON.

Модуль аналізу — програмний компонент, що реалізує алгоритм виявлення аномалій. Він порівнює поточні показники з базовими значеннями за попередній період і виявляє відхилення, що можуть свідчити про погіршення стану здоров'я тварини.

Telegram-бот — компонент інтерфейсу взаємодії з користувачем. Він автоматично формує текстові повідомлення про події та надсилає їх користувачу через застосунок Telegram, що дозволяє обійтися без розробки окремого мобільного застосунку.

Смартфон користувача — кінцевий пристрій отримувача інформації, на якому встановлено застосунок Telegram.

Послідовність проходження інформації в комплексі є такою. Маса тіла тварини спричиняє пружну деформацію тензодатчиків і викликає на виході мостової схеми диференціальну напругу. Цей аналоговий сигнал надходить на АЦП HX711, де підсилюється й перетворюється на 24-розрядний цифровий код. Мікроконтролер ESP32-S2 зчитує цей код, виконує обробку, формує пакет даних і за каналом Wi-Fi передає його на вебсервер. Вебсервер записує дані

					<i>PE22.404474.001 ПЗ</i>	Лис
						16
Зм.	Лис	№ докум.	Підпис	Дата		

у сховище, з якого їх отримує модуль аналізу. За результатами аналізу Telegram-бот формує повідомлення й надсилає його на смартфон користувача.

2.2 Обґрунтування вибору елементної бази

У цьому підрозділі обґрунтовано вибір основних компонентів апаратної частини комплексу: мікроконтролера, тензометричних датчиків, аналого-цифрового перетворювача та підсистеми живлення.

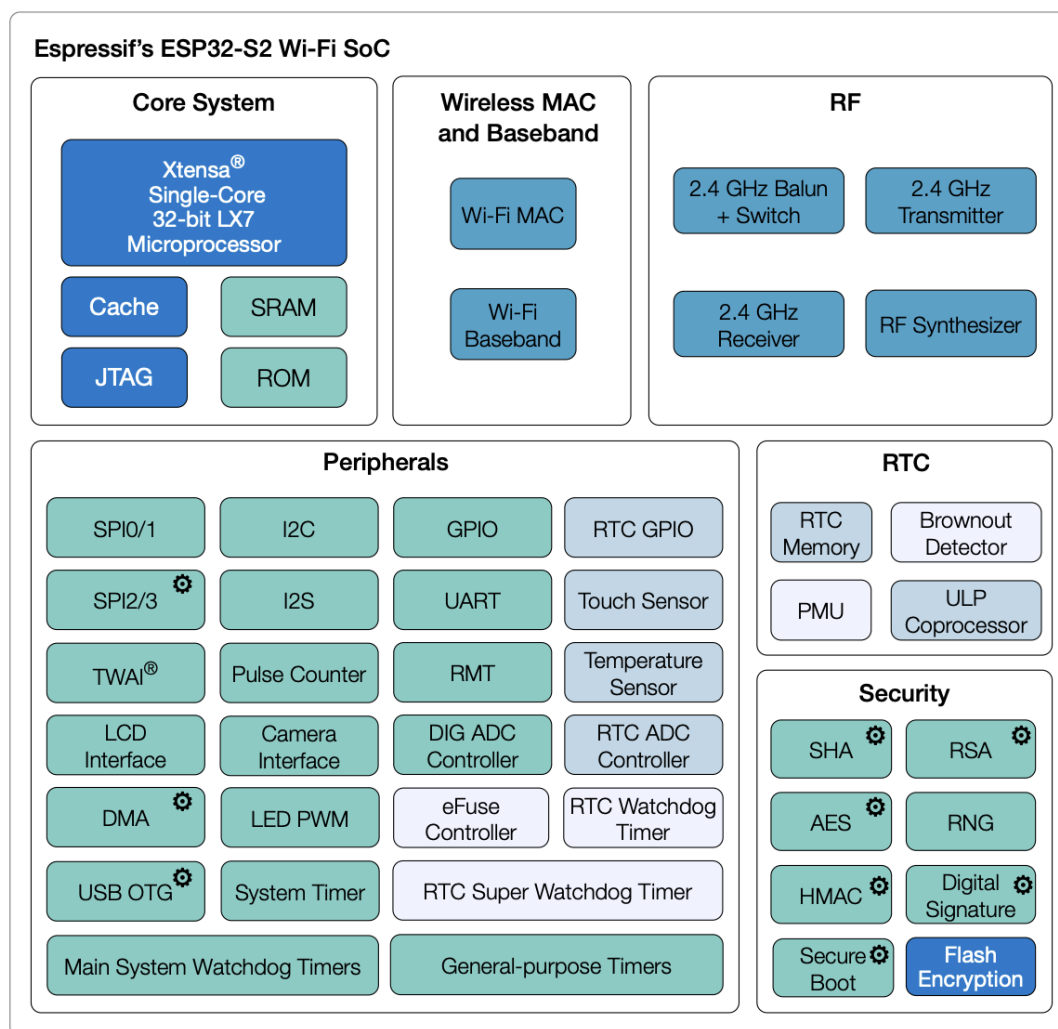
2.2.1 Мікроконтролер

Мікроконтролер є центральним вузлом пристрою — від нього залежить продуктивність обчислень, енергоспоживання та можливості бездротового зв'язку. Виходячи з функціональних вимог до пристрою, до мікроконтролера висуваються такі вимоги:

- наявність вбудованого Wi-Fi-модуля стандарту IEEE 802.11 b/g/n для прямого підключення до домашньої точки доступу;
- достатня обчислювальна потужність для виконання алгоритмів цифрової фільтрації сигналу;
- наявність режиму глибокого сну (Deep Sleep) з наднизьким споживанням;
- наявність енергонезалежної RTC-пам'яті, що зберігається під час глибокого сну;
- широка номенклатура цифрових входів-виходів (GPIO);
- доступність плати, документації та середовища розробки.

На основі цих вимог було обрано мікроконтролер серії ESP32-S2 виробництва компанії Espressif Systems — однокристальну систему з вбудованим Wi-Fi 802.11 b/g/n, орієнтовану на застосування в пристроях Інтернету речей з низьким енергоспоживанням [8]. Функціональна схема мікроконтролера ESP32-S2 наведена на рисунку 2.3.

					<i>PE22.404474.001 ПЗ</i>	Лис
						17
Зм.	Лис	№ докум.	Підпис	Дата		



Modules having power in specific power modes:

- Active
- Active and Modem-sleep
- Active, Modem-sleep, and Light-sleep; optional in Light-sleep
- Active, Modem-sleep, Light-sleep, and Deep-sleep
- all modes

Рисунок 2.3 — Функціональна схема мікроконтролера ESP32-S2 [8]

Основні характеристики ESP32-S2, що визначили його вибір [8]:

- обчислювальне ядро: одноядерний 32-розрядний процесор Xtensa LX7 з тактовою частотою до 240 МГц;
- пам'ять: 128 КБ ROM, 320 КБ SRAM, окрема SRAM у RTC-доміні (зберігається під час Deep Sleep);
- бездротовий зв'язок: інтегрована Wi-Fi-підсистема IEEE 802.11 b/g/n у діапазоні 2,4 ГГц; інтегровані радіотракт, бейсбенд, MAC-рівень, балун і RF-перемикач, що усуває потребу в зовнішніх радіокомпонентах;
- периферія: 43 програмованих GPIO, інтерфейси SPI, I²C, UART;

					PE22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		18

- енергоощадні режими: у режимі Deep-sleep типове споживання близько 25 мкА при увімкненому RTC-таймері та збереженні RTC-пам'яті.

ESP32-S2, на відміну від базової моделі ESP32, не містить модуля Bluetooth. У контексті розроблюваного пристрою це не є недоліком: для передачі даних на сервер використовується виключно Wi-Fi. Тому вибір ESP32-S2 без надлишкової функціональності є виправданим і дозволяє знизити вартість пристрою.

Фізично мікроконтролер реалізовано у вигляді відлагоджувальної плати LOLIN S2 Mini виробництва WEMOS Electronics — компактного модуля у форм-факторі D1 Mini (близько 34×25 мм) на базі чіпа ESP32-S2FN4R2 з 4 МБ flash-пам'яті та 2 МБ PSRAM [8]. Плата має вбудований роз'єм USB Type-C, стабілізатор напруги 3,3 В та виведені на різних 27 цифрових входів-виходів. Компактні розміри дозволяють розмістити її в захищеному корпусі під лотком.

2.2.2 Тензометричні датчики

Тензометричні датчики є первинними перетворювачами вимірювальної системи. Зважаючи на специфіку застосування, до них висуваються такі вимоги:

- сумарний діапазон вимірювання з запасом не менше 4–5-кратного від максимальної очікуваної маси (з урахуванням кінетичних перевантажень при стрибках);
- висока чутливість до малих змін навантаження;
- лінійність характеристики у робочому діапазоні;
- мала висота для збереження низького профілю платформи;
- доступність у вільному продажу.

Для платформи з чотирма точками опори використовуються півмостові тензометричні датчики (half-bridge load cell), у яких на одному пружному елементі розміщуються два тензорезистори. Чотири таких датчики, встановлені

					<i>PE22.404474.001 ПЗ</i>	Лис
						19
Зм.	Лис	№ докум.	Підпис	Дата		

під ніжками платформи, об'єднуються у повну мостову схему Вітстона (детально — у п. 2.3). Це стандартне рішення для побутових ваг, що забезпечує підсумовування навантажень та температурну компенсацію.

На основі цих вимог обрано тензометричні датчики SC902 виробництва Sensor and Control Co., Ltd. з номінальним навантаженням 50 кг на датчик [7]. Зовнішній вигляд комплекту з чотирьох датчиків наведено на рисунку 2.4.



Рисунок 2.4 — Зовнішній вигляд тензометричних датчиків SC902 [7]

Основні технічні характеристики датчика SC902 [7]:

- номінальне навантаження: 50 кг на датчик (сумарне з чотирьох датчиків — до 200 кг);
- номінальний вихідний сигнал: $1,0 \pm 0,15$ мВ/В;
- точність (комплексна похибка): 0,3 % від повної шкали;
- нелінійність: 0,2 %, гістерезис: 0,2 %, повторюваність: 0,1 % від повної шкали;
- вхідний та вихідний опір: 1000 ± 10 Ом;
- напруга живлення: від 3 до 10 В постійного струму;
- робочий діапазон температур: від -10 до $+40$ °С;
- допустиме перевантаження: 150 % від номінального; маса: 22 г.

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						20
Зм.	Лис	№ докум.	Підпис	Дата		

Конструктивно SC902 є плоским датчиком з пружним елементом у вигляді консольної балки. Кожен датчик має три гнучкі вивідні провідники (білий, червоний, чорний) і є півмостовим. Невелика висота датчика (7,8 мм для модифікації на 50 кг) дозволяє створити платформу мінімальної товщини.

Вибір номіналу 50 кг (а не меншого) є компромісом між чутливістю та надійністю: датчики з меншим номіналом мали б вищу чутливість, але швидко виходили б з ладу при кінетичних перевантаженнях під час стрибків тварини, які можуть сягати 8–10 кг. Сумарний діапазон 200 кг забезпечує значний запас міцності.

2.2.3 Аналого-цифровий перетворювач

Сигнал на виході мостової схеми має диференціальний характер і малий рівень — порядку одиниць мілівольт. Цю задачу не вдається розв'язати вбудованим АЦП ESP32-S2 з причин:

- роздільна здатність вбудованого АЦП ESP32-S2 (12–13 бітів) недостатня для розпізнавання змін маси на рівні десятків грамів у діапазоні до 200 кг;
- вбудований АЦП не має диференціального входу та підсилювача з програмованим коефіцієнтом;
- ESP32-S2 не має вбудованого джерела опорної напруги для живлення моста.

Тому потрібен спеціалізований зовнішній АЦП для тензодатчиків. На основі вимог (роздільна здатність не менше 20 бітів, диференціальний вхід з PGA, вбудоване живлення моста, простий інтерфейс, низьке споживання) обрано 24-розрядний прецизійний АЦП HX711 виробництва Avia Semiconductor [6]. Типова схема застосування HX711 наведена на рисунку 2.5.

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						21
Зм.	Лис	№ докум.	Підпис	Дата		

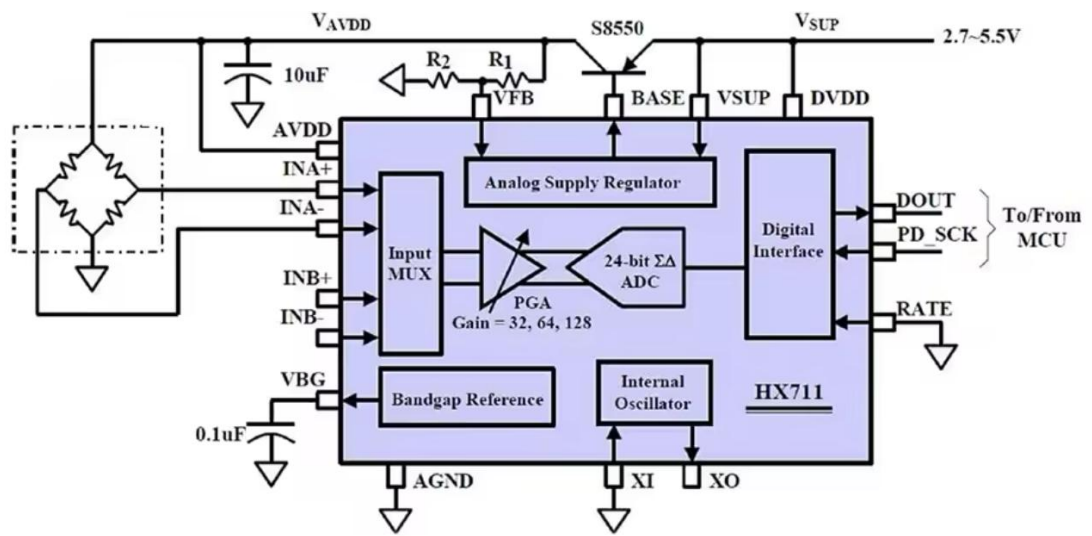


Рисунок 2.5 — Типова схема застосування АЦП HX711 [6]

HX711 містить усі необхідні вузли вимірювального тракту: вхідний мультиплексор, малoshумний програмований підсилювач (PGA) з коефіцієнтами 32, 64, 128, 24-розрядний дельта-сигма АЦП, внутрішній генератор, джерело опорної напруги та стабілізатор живлення моста.

Основні характеристики HX711 [6]: роздільна здатність 24 розряди; два диференціальні канали (A, B); коефіцієнти підсилення PGA 32/64/128; частота вибірок 10 або 80 за секунду; напруга живлення 2,7–5,5 В; струм споживання близько 1,5 мА в активному режимі та менше 1 мкА у режимі вимкнення; двопровідний послідовний інтерфейс (DOUT, PD_SCK).

У пристрої використовується канал А з коефіцієнтом підсилення 128 для максимальної чутливості. При чутливості датчиків 1 мВ/В і напрузі живлення моста близько 3,3 В корисний сигнал на номінальному навантаженні становить близько 3,3 мВ, який підсилюється та оцифровується з достатньою точністю. Режим вимкнення HX711 (споживання менше 1 мкА) керується мікроконтролером і використовується в алгоритмі енергоощадної роботи (Розділ 3).

2.2.4 Підсистема живлення

Підсистема живлення визначає тривалість роботи, зручність обслуговування й безпеку тварини. Згідно з вихідними даними завдання, напруга живлення має бути в діапазоні 4,4–5,25 В, а споживаний струм — не перевищувати

0,1 А. З урахуванням постійного розміщення пристрою в зоні доступу тварини висуваються додаткові вимоги: повне розміщення джерела всередині корпусу, відсутність зовнішніх кабелів, достатня автономність та проста заміна.

Експериментальна перевірка проблеми зовнішнього кабелю. Необхідність відмови від зовнішнього провідного живлення підтверджена експериментально. На ранньому етапі розробки пристрій живився від мережного адаптера через роз'єм USB Type-C. Під час експлуатаційного випробування поряд з твариною було зафіксовано, що тварина гризе живильний кабель: за короткий час ізоляцію було пошкоджено. Це наочно показало, що будь-яке зовнішнє провідне з'єднання в зоні доступу тварини є неприпустимим, оскільки створює загрозу як для працездатності пристрою, так і для здоров'я тварини (травмування ротової порожнини, ризик ураження струмом). Цей результат став визначальним критерієм вибору живлення.

З урахуванням цього розглянуто три варіанти реалізації підсистеми живлення.

Зовнішнє USB-живлення. Технічно найпростіший варіант, проте, як показав експеримент, наявність постійного зовнішнього кабелю в зоні доступу тварини є неприйнятною. Тому варіант відхилено.

Літій-іонний акумулятор (18650). Передбачає банку 18650 з boost-конвертером до 5 В і модулем заряду. Попри високу ємність, заряджання вимагає або зовнішнього роз'єму (повертає проблему кабелю), або знімного акумулятора (ускладнює герметичний корпус); крім того, літій-іонні елементи мають обмежений ресурс і підвищені вимоги до безпеки. Тому варіант не оптимальний.

Лужні батарейки типу АА. Послідовне з'єднання трьох елементів АА (1,5 В кожен) дає 4,5 В — точно посередині діапазону 4,4–5,25 В. Цей варіант обрано як основний, оскільки:

- напруга 4,5 В відповідає ТЗ без перетворювачів;
- батарейки розміщуються всередині корпусу без зовнішніх з'єднань;

					РЕ22.404474.001 ПЗ	Лис
						23
Зм.	Лис	№ докум.	Підпис	Дата		

- вони найдоступніші;
- не потребують схем заряду й захисту;
- мають високу безпеку експлуатації.

Порівняння розглянутих варіантів живлення наведено в таблиці 2.1

Таблиця 2.1 — Порівняння варіантів реалізації підсистеми живлення

Критерій	USB	18650	3×AA
Вихідна напруга	5 В	3,7 В + boost	4,5 В
Відповідність ТЗ (4,4–5,25 В)	так	через перетв.	так
Зовнішні дроти/роз'єми	так	так (заряд)	відсутні
Безпека для тварини	низька	середня	висока
Автономність	необмеж.	висока	достатня
Складність схеми	низька	висока	низька
Додаткові модулі	немає	так	немає

Як видно з таблиці, варіант із трьома лужними батарейками AA є оптимальним за сукупністю критеріїв: він єдиний поєднує відповідність технічному завданню, відсутність зовнішніх з'єднань та просту апаратну реалізацію без додаткових модулів.

Структура підсистеми живлення проста: три батарейки AA з'єднуються послідовно, утворюючи джерело 4,5 В, яке подається на живильний вхід плати ESP32-S2 Mini. Напруга 3,3 В для мікроконтролера формується вбудованим стабілізатором плати, а живлення аналогової частини HX711 — внутрішнім стабілізатором самого АЦП.

У коло живлення між батарейним блоком і платою введено механічний клавішний вимикач. Як вимикач живлення обрано мініатюрний клавішний перемикач KCD11. Зовнішній вигляд вимикача наведено на рисунку 2.6.



Рисунок 2.6 — Зовнішній вигляд клавішного вимикача KCD11

Основні технічні характеристики вимикача KCD11:

- тип: клавішний, однополюсний на одне замикання (SPST);
- кількість положень: 2 (увімкнено/вимкнено) з фіксацією;
- номінальний струм і напруга: 6 А при 125 В / 3 А при 250 В змінного струму;
- опір контактів: не більше 35 мОм;
- розмір монтажного отвору: $13,5 \times 8,3$ мм;
- кількість виводів: 2.

Навантажувальна здатність вимикача багаторазово перевищує робочі параметри кола живлення пристрою (4,5 В, струм до 0,1 А), що гарантує надійну комутацію з великим запасом, а компактні розміри дозволяють вивести клавішу на стінку корпусу для зручного доступу.

Окрім повного знеструмлення пристрою, вимикач виконує важливу експлуатаційну роль: під час обслуговування лотка (чищення, заміна наповнювача) пристрій вимикається, завдяки чому маніпуляції з лотком не сприймаються системою як хибні зважування й не спотворюють накопичену статистику. Повторне ввімкнення ініціює холодний старт із автоматичним таруванням — визначенням нового нульового рівня вже з урахуванням оновленого

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						25
Зм.	Лис	№ докум.	Підпис	Дата		

стану лотка. Таким чином, вимикач одночасно виконує функції знеструмлення та коректного скидання нуля після обслуговування.

2.3 Принцип роботи вимірювального тракту

Вимірювальний тракт перетворює масу тіла тварини на цифровий код. Тракт складається з трьох ланок: тензометричних датчиків у мостовій схемі, АЦП НХ711 та цифрового інтерфейсу зв'язку з мікроконтролером.

В основі роботи датчиків лежить тензорезистивний ефект — зміна електричного опору провідника при механічній деформації. Тензорезистор являє собою тонку металеву фольгу, наклеєну на пружний елемент. Під дією сили елемент деформується разом із тензорезистором, що змінює його опір. Відносна зміна опору пов'язана з деформацією співвідношенням:

$$\frac{\Delta R}{R} = k \times \varepsilon$$

де ΔR — зміна опору;

R — початковий опір;

k — коефіцієнт тензочутливості;

ε — відносна деформація. Це співвідношення відповідає рівнянню (1) джерела [5].

Оскільки деформації малі, зміна опору дуже мала, тому для її точного вимірювання застосовують мостову схему.

Для перетворення малої зміни опору на пропорційну напругу тензорезистори вмикають за схемою моста Вітстона. У пристрої використовуються чотири півмостові датчики SC902, які разом утворюють повний міст із чотирма активними плечима. Схема з'єднання наведена на рисунку 2.7.

					РЕ22.404474.001 ПЗ	Лис
						26
Зм.	Лис	№ докум.	Підпис	Дата		

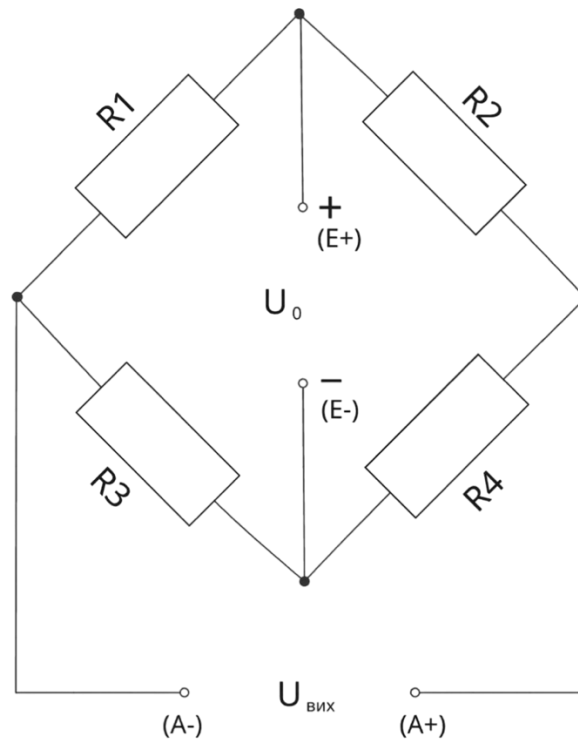


Рисунок 2.7 — Схема з'єднання тензодатчиків у повний міст Вістона

На одну діагональ моста подається напруга живлення, а з іншої знімається вихідний сигнал. Датчики розміщуються під чотирма опорними точками платформи так, щоб під навантаженням опір у протилежних плечах змінювався узгоджено (в одних зростає, в інших зменшується). Це забезпечує підсумовування внеску всіх точок опори, завдяки чому показання не залежать від місця розташування вантажу.

Кожен датчик SC902 має три провідники: червоний (Excitation+), чорний (Excitation-) та білий (сигнал). При з'єднанні чотирьох датчиків у міст утворюються чотири зовнішні виводи: два для живлення (E+, E-) і два для сигналу (S+, S-), які під'єднуються до клем модуля НХ711.

За відсутності навантаження міст збалансований, напруга на вимірювальній діагоналі дорівнює нулю. Під навантаженням баланс порушується, і вихідна напруга стає пропорційною прикладеній силі:

$$\frac{U_{\text{вих}}}{U_{\text{жив}}} = \left(\frac{k}{4}\right) \times (\varepsilon_1 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4).$$

де k — коефіцієнт тензочутливості;

					РЕ22.404474.001 ПЗ	Лис
						27
Зм.	Лис	№ докум.	Підпис	Дата		

$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ — відносні деформації відповідних плечей моста;

$U_{\text{вих}}$ — вихідна напруга вимірювальної діагоналі;

$U_{\text{жив}}$ — напруга живлення моста.

Чутливість датчиків SC902 становить 1,0 мВ/В, тобто при напрузі живлення моста близько 3,3 В вихідний сигнал на номінальному навантаженні становить близько 3,3 мВ. Такий слабкий сигнал потребує попереднього підсилення.

Вихідний диференціальний сигнал моста надходить на канал А АЦП HX711, де підсилюється вбудованим підсилювачем (PGA) з коефіцієнтом 128 і подається на 24-розрядний дельта-сигма АЦП. Висока роздільна здатність у поєднанні з великим коефіцієнтом підсилення забезпечує перетворення навіть малих змін сигналу на цифровий код.

Слід зазначити, що молодші розряди коду завжди зашумлені власними шумами тракту, тому ефективна роздільна здатність є нижчою за номінальну. Проте навіть з урахуванням цього точність значно перевищує потреби задачі: дискретність вимірювання маси становить одиниці грамів, що достатньо для виявлення діагностично значущих змін маси (десятки–сотні грамів).

Цифровий код передається від HX711 до ESP32-S2 по двопровідному послідовному інтерфейсу (лінія даних DOUT та лінія тактових імпульсів PD_SCK). Коли результат готовий, HX711 встановлює низький рівень на DOUT; мікроконтролер формує серію тактових імпульсів на PD_SCK, зчитуючи 24 біти. Кількістю імпульсів додатково задається коефіцієнт підсилення для наступного перетворення. Ця ж лінія використовується для переведення HX711 у режим зниженого споживання (Розділ 3).

Таким чином, вимірювальний тракт забезпечує повне перетворення маси на цифровий код: деформація датчиків змінює опір тензорезисторів, мостова схема перетворює це на напругу, АЦП HX711 підсилює та оцифровує сигнал, а код передається мікроконтролеру для обробки (Розділ 3).

					РЕ22.404474.001 ПЗ	Лис
						28
Зм.	Лис	№ докум.	Підпис	Дата		

2.4 Апаратна частина

Апаратна частина складається з п'яти компонентів: батарейного блока (3 × AA), механічного вимикача живлення KCD11, плати ESP32-S2 Mini, модуля HX711 та групи з чотирьох тензодатчиків у мості.

Живлення. Батарейний блок через механічний вимикач живлення подає напругу 4,5 В на живильний вхід плати ESP32-S2 Mini (додатний полюс — на вивід VBUS, від'ємний — на GND). Вимикач увімкнено послідовно в додатне коло живлення; його призначення розглянуто в п. 2.2.4. Вбудований стабілізатор формує напругу 3,3 В для мікроконтролера та HX711. З'єднання мікроконтролера з АЦП виконано чотирма провідниками:

- вивід VCC модуля HX711 — до виводу 3V3 плати;
- вивід GND модуля HX711 — до виводу GND плати;
- лінія даних DT модуля HX711 — до виводу GPIO5 мікроконтролера;
- лінія тактів SCK модуля HX711 — до виводу GPIO6 мікроконтролера.

Вибір виводів GPIO5 і GPIO6 зумовлений тим, що на платі LOLIN S2 Mini вони є вільними цифровими виводами, не задіяними під інші функції.

З'єднання АЦП з тензодатчиками: виводи живлення моста — до клем E+ та E– модуля, сигнальні виводи моста — до клем A+ та A– модуля (канал А підсилювача HX711).

Таким чином, схема реалізує повний тракт від первинних перетворювачів до мікроконтролера й не містить зовнішніх з'єднань, доступних тварині, що відповідає вимогам безпеки з п. 2.2.4.

Конструктивне виконання пристрою має враховувати специфічні умови експлуатації. Оскільки пристрій розміщується під лотком — у середовищі з підвищеною вологістю та ризиком потрапляння рідини й наповнювача, — корпус електронної частини має бути вологозахисним. Відсік для елементів живлення повинен забезпечувати просту заміну батарейок, водночас лишаючись ізольованим від вологи. Окрему увагу слід приділити жорсткості виміральної платформи: вона не повинна помітно прогинатися під навантаженням,

					РЕ22.404474.001 ПЗ	Лис
						29
Зм.	Лис	№ докум.	Підпис	Дата		

оскільки прогин платформи спотворював би розподіл зусилля між опорними точками й погіршував точність вимірювання маси. Тому платформа виконується з достатньо жорсткого матеріалу, який передає вагу тварини безпосередньо на чотири опорні точки з тензодатчиками, не поглинаючи її у власній деформації.

2.5 Розрахункова частина

2.5.1 Калібрування вимірювального тракту

АЦП НХ711 видає безрозмірний цифровий код, пропорційний навантаженню. Для перетворення його на масу в кілограмах виконують калібрування. Залежність між масою та кодом лінійна:

$$m = \frac{(N - N_0)}{k}$$

де m — маса;

N — поточний код АЦП;

N_0 — код за відсутності навантаження (нуль);

k — калібрувальний коефіцієнт.

Калібрування виконувалося у два етапи. Спочатку визначався нуль тракту (N_0): за відсутності навантаження фіксувалося показання, яке приймалося за початок відліку (функція тарування), а значення зміщення зберігалося в RTC-пам'яті для збереження під час глибокого сну.

На другому етапі визначався коефіцієнт k : на платформу встановлювався еталонний вантаж масою 2 кг, після чого значення коефіцієнта підбиралося ітеративно — коефіцієнт послідовно коригувався доти, доки показання пристрою не збіглося з дійсною масою еталона. Такий метод підбору за еталонною масою є стандартною процедурою калібрування для застосувань з НХ711 і дозволяє врахувати реальні особливості конкретного тракту. Підібране значення для розробленого пристрою становить $k = -21752,5$ одиниці коду на кілограм. Від'ємний знак зумовлений полярністю підключення сигнальних виводів моста й враховується автоматично.

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						30
Зм.	Лис	№ докум.	Підпис	Дата		

Для перевірки калібрування використовувався контрольний вантаж масою 1 кг. Показання пристрою відповідали дійсній масі в межах допустимої похибки, що підтвердило коректність коефіцієнта та лінійність тракту.

2.5.2 Оцінний розрахунок часу автономної роботи

Оскільки пристрій живиться від автономного джерела, важливою характеристикою є час роботи без заміни батарейок. Робота складається з двох режимів: чергового (пристрій більшу частину часу у глибокому сні, прокидаючись кожні 2 с для перевірки навантаження) та активного (тривалий збір даних і передача по Wi-Fi за фактом виявлення тварини).

Вихідні дані для розрахунку (на основі даташитів [6, 9] та профілю програми):

- корисна ємність трьох батарейок AA — близько 1800 мА·год;
- струм у глибокому сні (ESP32-S2 з RTC та HX711 у power-down) — близько 25 мкА;
- струм у активному режимі — близько 40 мА;
- струм передачі по Wi-Fi — близько 120 мА;
- тривалість пробудження — близько 0,1 с;
- період пробуджень — 2 с.

Середній струм у черговому режимі визначається як середньозважене за часом значення струмів активної фази та фази сну й становить близько 2,1 мА. З урахуванням 5 візитів на добу (по 60 с активного режиму та 5 с передачі) сумарний добовий розхід становить близько 54 мА·год. Тоді очікуваний час автономної роботи:

$$T = \frac{1800}{54} \approx 33 \text{ доби}$$

Аналіз структури споживання показує, що основний внесок (понад 90 %) дає черговий режим періодичних пробуджень, тоді як візити тварини споживають незначну частку енергії. Отже, ключовим резервом збільшення автономності є оптимізація періоду опитування платформи. Залежність часу роботи від періоду пробуджень наведена в таблиці 2.2.

					РЕ22.404474.001 ПЗ	Лис
						31
Зм.	Лис	№ докум.	Підпис	Дата		

Таблиця 2.2 — Залежність часу автономної роботи від періоду опитування

Період опитування, с	Середній струм, мА	Очікуваний час роботи
2 (поточний)	2,10	≈ 33 доби
3	1,41	≈ 47 діб
5	0,86	≈ 2,4 місяця
10	0,44	≈ 4,1 місяця
30	0,16	≈ 7,4 місяця
60	0,09	≈ 9,3 місяця

Як видно з таблиці, збільшення періоду опитування з 2 до 10 секунд підвищує час роботи майже вчетверо — до чотирьох місяців, практично не погіршуючи здатність фіксувати відвідування (оскільки тривалість перебування тварини в лотку значно перевищує 10 секунд). Оптимізація періоду опитування є одним із напрямів подальшого вдосконалення пристрою.

Наведений розрахунок є оцінним: реальний час роботи залежить від конкретних струмів плати, якості батарейок, температури та фактичної частоти відвідувань. Проте він дає коректне уявлення про порядок автономності та вказує на основний резерв її підвищення.

Висновки до розділу 2

1. Розроблено структуру програмно-апаратного комплексу з розподіленою архітектурою, що складається з апаратного пристрою та серверної частини, поєднаних каналом Wi-Fi. Побудовано схему електричну структурну комплексу.

2. Обґрунтовано вибір елементної бази: мікроконтролер ESP32-S2 з вбудованим Wi-Fi та режимом глибокого сну; чотири півмостові тензодатчики SC902 (50 кг), з'єднані у повний міст; спеціалізований 24-розрядний АЦП HX711. Вибір кожного компонента підкріплено аналізом вимог і альтернатив.

					PE22.404474.001 ПЗ	Лис
						32
Зм.	Лис	№ докум.	Підпис	Дата		

3. Обґрунтовано вибір автономного живлення від трьох лужних батарейок АА. На основі експериментально виявленої проблеми пошкодження зовнішнього кабелю твариною показано, що джерело має бути автономним і розміщеним усередині корпусу.

4. Розглянуто принцип роботи вимірювального тракту: від тензорезистивного ефекту й мостової схеми Вітстона до підсилення та аналого-цифрового перетворення. Розроблено схему електричну з'єднань пристрою.

5. Виконано калібрування вимірювального тракту та оцінний розрахунок часу автономної роботи. Встановлено, що при поточних параметрах пристрій працює близько 33 діб від одного комплекту батарейок, а основним резервом підвищення автономності є оптимізація періоду опитування платформи.

					РЕ22.404474.001 ПЗ	Лис
						33
Зм.	Лис	№ докум.	Підпис	Дата		

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АЛГОРИТМІВ

3.1 Загальна архітектура програмного забезпечення

Програмне забезпечення комплексу побудовано за клієнт-серверною архітектурою й складається з двох незалежних частин, що працюють на різних апаратних платформах та взаємодіють між собою каналом бездротового зв'язку.

Першою частиною є вбудоване програмне забезпечення мікроконтролера ESP32-S2, написане мовою C++ у середовищі Arduino. Воно виконується безпосередньо на пристрої й відповідає за збирання даних з вимірювального тракту, їх первинну обробку та фільтрацію, енергоощадне керування пристроєм і передавання результатів на сервер. Вбудоване ПЗ працює автономно й не потребує втручання користувача.

Другою частиною є серверне програмне забезпечення, написане мовою Python. Воно виконується на хмарній платформі й відповідає за приймання даних від мікроконтролера, їх постійне зберігання, аналіз накопиченої статистики та взаємодію з користувачем через месенджер Telegram.

Розподіл функцій між двома частинами зумовлений їх різним призначенням. Мікроконтролер є ресурсообмеженим пристроєм з батарейним живленням, тому на нього покладаються лише задачі, що вимагають безпосередньої роботи з апаратурою та мають виконуватися в реальному часі: опитування датчиків, фільтрація сигналу, керування живленням. Натомість задачі, що потребують тривалого зберігання даних та обчислень над накопиченою статистикою, винесено на сервер, який не обмежений за енергоспоживанням та обчислювальними ресурсами. Такий розподіл дозволяє мінімізувати енергоспоживання пристрою й подовжити час його автономної роботи.

Взаємодія між частинами відбувається в односторонньому напрямку: мікроконтролер за фактом зафіксованого відвідування формує пакет даних і надсилає його на сервер за протоколом HTTP. Сервер приймає, зберігає й оброб-

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						34
Зм.	Лис	№ докум.	Підпис	Дата		

ляє ці дані, а результати аналізу надає користувачу за його запитом або у вигляді сповіщень. Детально вбудоване ПЗ розглянуто у п. 3.2, протокол передавання даних у п. 3.3, а серверну частину у п. 3.4.

3.2 Вбудоване програмне забезпечення мікроконтролера

3.2.1 Загальний алгоритм роботи

Вбудоване програмне забезпечення реалізує циклічний алгоритм роботи пристрою, побудований навколо режиму глибокого сну для забезпечення мінімального енергоспоживання. Загальний алгоритм роботи мікроконтролера наведено в додатку А.

Робота пристрою організована як послідовність коротких циклів. Після кожного пробудження мікроконтролер виконує таку послідовність дій. Спочатку визначається причина пробудження: якщо це холодний старт (перше ввімкнення живлення), виконується тарування вимірювального тракту — визначення та запам'ятовування нульового рівня. Якщо пробудження відбулося за сигналом таймера після глибокого сну, відновлюється раніше збережений нульовий рівень.

Далі мікроконтролер вмикає АЦП, вимірює поточну вагу й порівнює її з нульовим рівнем. Якщо різниця не перевищує порогового значення 1 кг, лоток вважається порожнім, і пристрій одразу повертається у глибокий сон до наступного циклу. Якщо ж різниця перевищує поріг, фіксується факт присутності тварини, і пристрій переходить у активний режим, у якому виконуються вимірювання маси, фільтрація сигналу та, за результатами, передавання даних на сервер. Після завершення обробки пристрій знову повертається у глибокий сон.

Такий цикл повторюється безперервно. Завдяки тому, що більшу частину часу пристрій перебуває у глибокому сні, прокидаючись лише на короткі інтервали для перевірки, досягається низьке середнє енергоспоживання. Ключові механізми алгоритму розглянуто детальніше нижче.

					РЕ22.404474.001 ПЗ	Лис
						35
Зм.	Лис	№ докум.	Підпис	Дата		

3.2.2 Енергоощадне керування

Основою енергоощадної роботи пристрою є режим глибокого сну мікроконтролера ESP32-S2. У цьому режимі вимикається основне обчислювальне ядро та більшість периферії, а активним залишається лише таймер реального часу, що забезпечує пробудження через заданий інтервал. Інтервал між пробудженнями встановлено рівним 2 с, що задає періодичність опитування платформи.

Перед кожним переходом у сон мікроконтролер також переводить АЦП HX711 у режим зниженого споживання, завдяки чому в стані сну обидва основні компоненти споживають мінімальний струм. Увімкнення таймера пробудження та перехід у сон виконуються відповідними системними викликами:

```
esp_sleep_enable_timer_wakeup(2000000); // інтервал 2 с  
esp_deep_sleep_start();
```

Важливою особливістю режиму глибокого сну є те, що під час сну вміст звичайної оперативної пам'яті втрачається — мікроконтролер після пробудження фактично виконує програму спочатку. Це створює проблему збереження стану між циклами: зокрема, нульовий рівень ваги, визначений під час тарування, має зберігатися протягом усього часу роботи пристрою. Для розв'язання цієї задачі використано енергонезалежну RTC-пам'ять, вміст якої зберігається під час глибокого сну. Змінні, які мають зберегтись після сну, оголошуються зі спеціальним атрибутом:

```
RTC_DATA_ATTR float m_base = 0.0; // нульовий рівень (тара)
```

```
RTC_DATA_ATTR long rtc_offset = 0; // зміщення нуля АЦП
```

У RTC-пам'яті зберігаються нульовий рівень тари та апаратне зміщення нуля бібліотеки роботи з АЦП. Останнє необхідне тому, що бібліотека HX711 після кожного перезапуску втрачає внутрішнє значення зміщення, тому воно зберігається окремо й примусово відновлюється після кожного пробудження.

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		36

Такий підхід дозволяє виконати тарування лише один раз при ввімкненні пристрою, а в подальшому використовувати збережене значення, не повторюючи цю операцію щоразу.

Для компенсації повільного температурного дрейфу нуля передбачено механізм плавного оновлення тари: коли лоток порожній і поточне відхилення дуже мале, нульовий рівень повільно коригується за принципом експоненційного згладжування. Це дозволяє утримувати точність вимірювань протягом тривалого часу без повторного тарування.

3.2.3 Алгоритм детекції відвідування та визначення маси

Ключовою задачею вбудованого ПЗ є коректне визначення маси тварини в умовах динамічних навантажень. Особливість полягає в тому, що тварина не стоїть на платформі нерухомо: вона застрибує, рухається, копає наповнювач, спирається на бортики лотка. Усе це створює значні коливання сигналу, через які просте усереднення показань дало б грубу похибку. Для розв'язання цієї задачі розроблено алгоритм пошуку максимальної стабільної ваги.

Після виявлення присутності тварини, за рахунок перевищення порогу 1 кг, пристрій переходить у активний режим і починає опитувати датчики раз на секунду. На кожній ітерації виконується перевірка стабільності сигналу: якщо поточне значення ваги перевищує поріг присутності та водночас мало відрізняється від попереднього вимірювання, таке значення вважається стабільним і враховується в накопиченні. Умову стабільності в коді реалізовано так:

```
if ((current_weight - m_base >= 1.0) &&  
    (abs(current_weight - previous_weight) <= 0.08)) {  
    if (current_weight > max_stable_weight) {  
        max_stable_weight = current_weight;  
    }  
}
```

					РЕ22.404474.001 ПЗ	Лис
						37
Зм.	Лис	№ докум.	Підпис	Дата		

Серед усіх стабільних значень за сесію алгоритм запам'ятовує найбільше. Такий вибір саме максимуму, а не середнього, є принциповим. Тварина не завжди стоїть на платформі повною вагою: під час закопування наповнювача вона спирається передніми лапами на бортик лотка, через що частина її ваги передається на бортик, а не на платформу, і виміряне значення занижується. Натомість у момент, коли тварина стоїть на платформі всіма лапами, платформа сприймає її повну вагу — і це значення є максимальним за сесію. Тому максимальне стабільне значення найкраще відповідає дійсній масі тварини. Підсумкова маса обчислюється як це максимальне значення за вирахуванням нульового рівня.

Завершення відвідування визначається за падінням ваги нижче порогу присутності. Проте просте спрацювання за миттєвим падінням ваги є ненадійним з двох причин. По-перше, коли тварина спирається передніми лапами на бортик лотка, вага на платформі тимчасово падає, хоча тварина фактично не вийшла. По-друге, характерною є поведінка, коли тварина виходить із лотка й через кілька секунд повертається, щоб закопати наповнювач, — без захисту це сприймалося б як два окремі відвідування. Щоб уникнути обох ситуацій, застосовано таймер виходу: перехід до завершального стану відбувається лише тоді, коли вага залишається нижчою за поріг безперервно протягом 20 с. Якщо протягом цього часу вага знову зростає, лічильник скидається, і відвідування продовжує вважатися єдиною сесією. Це забезпечує фіксацію одного відвідування як однієї події з єдиним значенням маси.

Після підтвердження виходу тварини пристрій переходить у завершальний стан, у якому очікує стабілізації показань протягом 10 с. Це очікування потрібне, щоб платформа заспокоїлася після руху тварини, а також щоб урахувати можливу зміну ваги порожнього лотка. Після стабілізації вимірюється новий рівень порожнього лотка, який стає новим значенням тари для наступних вимірювань.

					PE22.404474.001 ПЗ	Лис
						38
Зм.	Лис	№ докум.	Підпис	Дата		

3.2.4 Захисні механізми

Для підвищення надійності роботи в алгоритм закладено кілька захисних механізмів.

Перший — захист від «застрягання» в активному режимі. Якщо з якоїсь причини вага утримується вище порогу присутності надто довго, пристрій не має залишатися в активному режимі нескінченно, оскільки це швидко розрядило б батарею. Тому передбачено таймаут: якщо активний режим триває понад 10 хвилин, пристрій надсилає на сервер службове повідомлення про нештатну ситуацію, скидає тару й повертається у звичайний цикл роботи.

Другий — фільтр коротких відвідувань. Тварина може лише на мить заскочити на платформу й одразу зіскочити, або пройти повз, торкнувшись її. Такі події не є повноцінними відвідуваннями й не повинні фіксуватися як дані. Тому застосовано фільтр за тривалістю: якщо тривалість зафіксованого відвідування менша за 6 с, результат відкидається й на сервер не передається. Це відсікає випадкові короткочасні спрацювання й підвищує достовірність накопичуваної статистики.

Завдяки цим механізмам пристрій стійко працює в реальних умовах експлуатації, відрізняючи справжні відвідування тварини від випадкових подій і захищаючи себе від нештатних ситуацій. Повний текст вбудованого програмного забезпечення мікроконтролера наведено в додатку Б

3.3 Протокол передавання даних

Передавання даних від пристрою на сервер здійснюється бездротовим каналом Wi-Fi за протоколом прикладного рівня HTTP. Вибір саме цих технологій зумовлений їх повсюдною поширеністю: будь-яка домашня точка доступу Wi-Fi придатна для роботи пристрою, а протокол HTTP є стандартним і просто реалізується як на стороні мікроконтролера, так і на стороні сервера.

Важливою особливістю реалізації є те, що модуль Wi-Fi вмикається лише на час передавання даних. Радіомодуль є одним з найбільш енергоємних

					РЕ22.404474.001 ПЗ	Лис
						39
Зм.	Лис	№ докум.	Підпис	Дата		

вузлів мікроконтролера, тому тримати його ввімкненим постійно неприпустимо для пристрою з батарейним живленням. У звичайному циклі роботи Wi-Fi вимкнений, і лише за фактом зафіксованого відвідування, коли потрібно передати дані, мікроконтролер вмикає радіомодуль, встановлює з'єднання з точкою доступу, надсилає пакет даних і одразу вимикає Wi-Fi знову. Такий підхід суттєво зменшує середнє енергоспоживання.

Дані передаються методом HTTP POST у форматі JSON — компактному текстовому форматі, зручному для обміну структурованими даними. Кожне зафіксоване відвідування формується у вигляді JSON-пакета з такими полями:

```
{  
  "event_type": "cat",  
  "duration": 45,  
  "cat_weight": 4.250  
}
```

Поля пакета мають таке призначення:

- event_type — тип події. Значення «cat» відповідає зафіксованому відвідуванню тварини, службове значення «error_timeout» — нештатній ситуації перевищення таймауту активного режиму;
- duration — тривалість відвідування в секундах;
- cat_weight — обчислена маса тварини в кілограмах (після фільтрації).

Послідовність передавання даних така: мікроконтролер переводить Wi-Fi у режим станції, підключається до заданої точки доступу й очікує встановлення з'єднання протягом обмеженого часу (з контролем тайм-ауту, щоб не «зависнути» за відсутності мережі). У разі успішного підключення формується JSON-пакет, який надсилається POST-запитом на заздалегідь визначену адресу серверного застосунку. Після отримання відповіді сервера з'єднання роз-

					РЕ22.404474.001 ПЗ	Лис
						40
Зм.	Лис	№ докум.	Підпис	Дата		

ривається, а радіомодуль вимикається. Якщо підключитися до мережі не вдалося протягом відведеного часу, передавання скасовується, а пристрій повертається у звичайний цикл, щоб уникнути зайвих витрат енергії.

3.4 Серверна частина

Серверна частина комплексу реалізована мовою Python і виконується на хмарній платформі. Її задачі — приймання даних від мікроконтролера, постійне зберігання історії відвідувань, аналіз накопиченої статистики на предмет відхилень від норми та взаємодія з користувачем через месенджер Telegram.

3.4.1 Стек та середовище виконання

Серверне програмне забезпечення побудоване на трьох основних компонентах:

- aiohttp [10] — асинхронна бібліотека Python, що реалізує вебсервер для приймання HTTP-запитів від мікроконтролера на маршрут /api/sensor;
- aiogram [11] — асинхронна бібліотека для роботи з API Telegram, що реалізує бот-інтерфейс взаємодії з користувачем;
- asyncio — стандартна бібліотека асинхронного виконання, що дозволяє вебсерверу й Telegram-боту працювати паралельно в межах одного процесу.

Серверний застосунок розгорнуто на хмарній платформі Railway [12], що забезпечує безперервне виконання, автоматичний перезапуск у разі збоїв та постійний публічний URL для прийому даних від мікроконтролера. Для зберігання історії відвідувань підключено постійний том (Volume), на якому розміщується файл історії. Використання тому є принциповим: дані на ньому зберігаються між перезапусками й оновленнями застосунку, тоді як вміст контейнерної файлової системи скидається при кожному перезавантаженні.

3.4.2 Приймання та зберігання даних

Вебсервер приймає від мікроконтролера POST-запити з JSON-пакетами, описаними у п. 3.3. Після приймання пакета сервер виконує таку обробку: пе-

					РЕ22.404474.001 ПЗ	Лис
						41
Зм.	Лис	№ докум.	Підпис	Дата		

ревіряє коректність структури JSON, визначає тип події (звичайне відвідування або службове повідомлення про таймаут), у разі звичайного відвідування — дописує запис у файл історії та надсилає користувачу повідомлення в Telegram про факт зафіксованого візиту з вказанням маси.

Файл історії має простий формат — масив JSON-об'єктів, кожен з яких містить два поля: позначку часу та значення маси. Тривалість окремого візиту в історії не зберігається, оскільки для подальшого аналізу вона не є інформативною; натомість кількість походів за день автоматично обчислюється як кількість записів за відповідний день.

Свідомо обрано саме файлове сховище, а не повноцінну систему керування базою даних. Це рішення обґрунтоване характером і обсягом даних: за рік експлуатації при середній частоті 3–5 відвідувань на добу обсяг історії становитиме лише близько 1500 записів, що не перевищує сотні кілобайтів і повністю обробляється у пам'яті за мілісекунди. Запровадження повноцінної СУБД (наприклад, PostgreSQL) у такому випадку було б надмірним ускладненням, що потребувало б додаткових ресурсів і обслуговування без жодного виграшу в продуктивності або зручності. Файлове сховище натомість є простим, надійним і легко переноситься між середовищами.

3.4.3 Алгоритм аналізу даних

Алгоритм аналізу призначений для виявлення стійких змін маси тварини відносно її ж індивідуальної норми. Оскільки абсолютні значення маси в кожній тварини різні, аналіз будується не на порівнянні з фіксованою нормою, а на відстеженні динаміки власних показників тварини за різні за тривалістю проміжки часу. За кожним запитом аналізу виконується така послідовність дій. Блок-схему цього алгоритму наведено в додатку В.

Спочатку перевіряється, чи достатньо в історії накопичених даних для аналізу. Якщо обсягу даних недостатньо (наприклад, пристрій встановлено нещодавно й триваліших періодів ще не накопичено), формується звіт лише з

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						42
Зм.	Лис	№ докум.	Підпис	Дата		

поточними даними — останньою виміряною масою та наявною короткостроковою статистикою, — без визначення тенденції.

За наявності достатнього обсягу даних обчислюється середнє значення маси за три проміжки часу — останні 7, 30 і 90 днів. Коротший проміжок (7 днів) відображає поточний стан тварини, а триваліші (30 і 90 днів) слугують базовою лінією, відносно якої оцінюється відхилення. Такий набір періодів дозволяє відрізнити випадкові короткочасні коливання маси від стійкої спрямованої зміни — тенденції.

Далі за результатами порівняння середніх значень за періодами визначається тенденція зміни маси: чи зростає, спадає, чи залишається стабільною маса тварини в часі. Спрямоване зниження середньої маси за короткий період відносно триваліших розцінюється як тенденція до втрати маси.

Виявлена зміна порівнюється з пороговим значенням, що задає межу клінічно значущого відхилення. У межах розробленої системи прийнято поріг зниження маси 10 %, обґрунтований медичними даними з Розділу 1: стійка втрата котом понад 10 % маси тіла є визнаною підставою для звернення до ветеринара [4]. Медіана втрати маси за рік до встановлення діагнозу хронічної хвороби нирок становить близько 9 %. Саме тому відстеження маси за триваліші проміжки (30 і 90 днів) є принциповим: поступова втрата маси, характерна для прихованих хронічних захворювань, дає малопомітну зміну від тижня до тижня, але стає очевидною при порівнянні з базовою лінією за місяць чи квартал.

Якщо стійке зниження середньої маси перевищує встановлений поріг, формується відповідне попередження. Якщо тенденція до зниження відсутня або зміни не виходять за межі норми, підсумок констатує, що стан тварини перебуває в межах норми. У будь-якому разі за результатами аналізу формується звіт, який надсилається користувачу. Частота відвідувань лотка, хоча й фіксується системою при кожному візиті та зберігається в історії, у межах розроб-

					РЕ22.404474.001 ПЗ	Лис
						43
Зм.	Лис	№ докум.	Підпис	Дата		

леного алгоритму свідомо не використовується як критерій автоматичного попередження. Це обґрунтовано двома причинами. По-перше, вимірювальна платформа фіксує лише факт відвідування й не розрізняє сечовипускання та дефекацію, тоді як діагностично значущою є саме частота сечовипускання. По-друге, частота відвідувань є слабким діагностичним маркером: середнє добове число відвідувань у хворих котів відрізняється від здорових несуттєво, а відповідна площа під ROC-кривою не перевищує 0,6, що свідчить про низьку роздільну здатність цього показника [13]. Тому основним критерієм автоматичного аналізу обрано масу тіла як надійніший і кількісно обґрунтований маркер, а частота лишається допоміжною інформацією, доступною власнику через миттєві сповіщення про кожен візит.

3.4.4 Інтерфейс взаємодії з користувачем

Взаємодія з користувачем здійснюється через Telegram-бот. Такий підхід обрано з кількох міркувань: користувач отримує повідомлення на пристрій, яким він користується щодня, без потреби встановлювати окремий мобільний застосунок; інтерфейс месенджера є знайомим і не потребує навчання; реалізація бот-API є простою й безкоштовною.

Бот підтримує два режими взаємодії — автоматичне інформування про події та звіти за запитом.

Автоматичне інформування полягає в тому, що при кожному зафіксованому відвідуванні бот надсилає користувачу коротке повідомлення з даними цього візиту. Це дає змогу відстежувати активність тварини в реальному часі.

Звіт за запитом формується за натисканням відповідної кнопки в інтерфейсі бота. У відповідь бот виконує алгоритм аналізу (п. 3.4.3) на накопиченій історії та надсилає користувачу зведений звіт, що містить:

- значення маси, виміряне під час останнього відвідування;
- середню масу за останні 7, 30 і 90 днів, що відображає динаміку маси за різні проміжки часу;

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						44
Зм.	Лис	№ докум.	Підпис	Дата		

- виявлену тенденцію зміни маси та її величину у грамах і у відсотках відносно базової лінії;

- підсумкове повідомлення про стан — «стан у межах норми» або попередження з вказанням виявленого відхилення.

Таке зведення дає користувачу швидку оцінку стану тварини за тиждень в одному короткому повідомленні. Інтерфейс через Telegram повністю покриває потреби моніторингу й не потребує додаткових засобів — ні веб-кабінету, ні мобільного застосунку. Повний текст програмного коду серверної частини наведено в додатку Г.

Висновки до розділу 3

1. Розроблено архітектуру програмного забезпечення комплексу, що складається з двох частин — вбудованого ПЗ мікроконтролера ESP32-S2 (мова C++) та серверної частини (мова Python). Обґрунтовано розподіл функцій між частинами, що дозволяє мінімізувати енергоспоживання пристрою.

2. Розроблено вбудоване ПЗ мікроконтролера, що реалізує циклічний алгоритм роботи з режимом глибокого сну. Запропоновано та реалізовано алгоритм пошуку максимальної стабільної ваги, який забезпечує коректне визначення маси тварини в умовах динамічних навантажень і дозволяє уникнути заниження результату під час накопчування наповнювача.

3. Закладено механізми енергоощадного керування (глибокий сон ESP32-S2 і вимкнення HX711 між циклами, увімкнення Wi-Fi лише на час передавання) та захисні механізми (таймер виходу 20 с, що об'єднує повторні підходи в одну сесію; таймаут активного режиму; фільтр коротких відвідувань).

4. Розроблено протокол передавання даних на основі Wi-Fi та HTTP POST у форматі JSON. Визначено структуру пакета даних, що передається від мікроконтролера на сервер.

					РЕ22.404474.001 ПЗ	Лис
						45
Зм.	Лис	№ докум.	Підпис	Дата		

5. Розроблено серверну частину комплексу, що приймає дані від мікроконтролера, зберігає їх у файловому сховищі на постійному томі та виконує аналіз накопиченої статистики. Алгоритм аналізу побудований на обчисленні середньої маси за періоди 7, 30 і 90 днів та визначенні тенденції її зміни, що порівнюється з пороговим значенням, обґрунтованим медичними даними.

6. Реалізовано інтерфейс взаємодії з користувачем через Telegram-бот, що поєднує автоматичне інформування про події та звіти за запитом, забезпечуючи моніторинг стану тварини без потреби в додаткових застосунках чи веб-інтерфейсах.

					РЕ22.404474.001 ПЗ	Лис
						46
Зм.	Лис	№ докум.	Підпис	Дата		

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Методика та умови проведення досліджень

Метою експериментальних досліджень була перевірка працездатності розробленого програмно-апаратного комплексу в реальних умовах експлуатації та оцінка коректності виконання його основних функцій: вимірювання маси, детекції відвідувань, передавання даних на сервер і відображення їх користувачу.

Дослідження проводилися в домашніх умовах із реальним об'єктом — домашнім котом. Зібраний макет пристрою, що складається з вимірювальної платформи на чотирьох тензодатчиках, модуля АЦП НХ711 та плати мікроконтролера ESP32-S2, розміщувався під стандартним котячим лотком. Серверна частина розгорталася на хмарній платформі, а повідомлення приймалися в Telegram-бот на смартфоні. Зовнішній вигляд зібраного експериментального стенду наведено на рисунку 4.1.

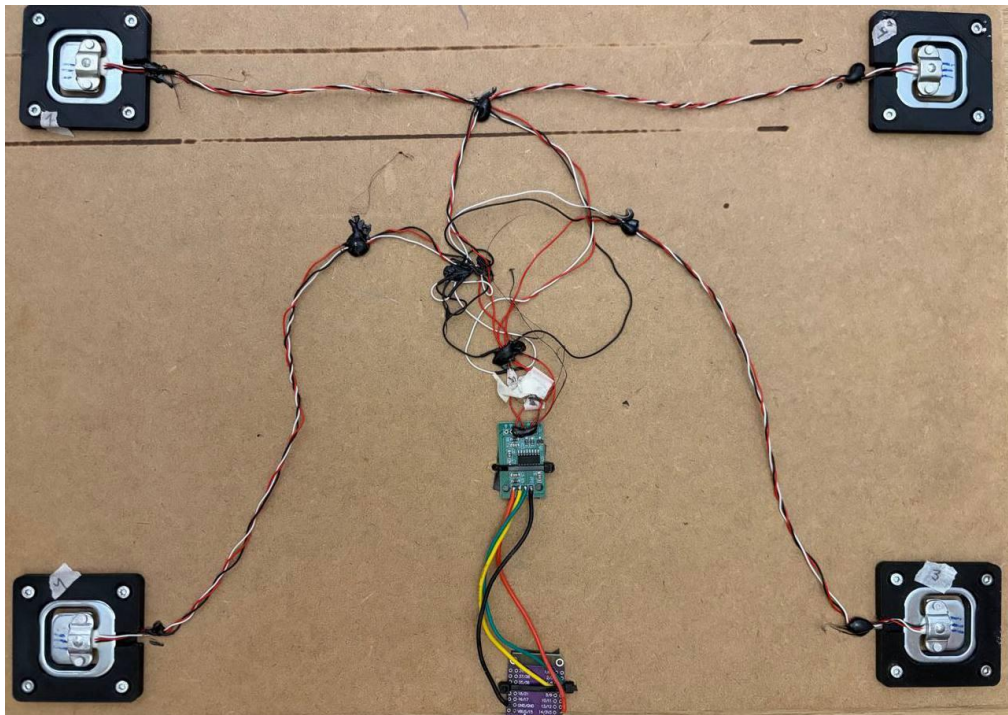


Рисунок 4.1 — Зовнішній вигляд експериментального макета комплексу

Перевірці підлягали такі аспекти роботи комплексу:

- коректність калібрування вимірювального тракту й відтворюваність результатів вимірювання;

					РЕ22.404474.001 ПЗ	Лис
						47
Зм.	Лис	№ докум.	Підпис	Дата		

- здатність системи правильно визначати масу тварини в реальних умовах, коли тварина рухається, копає наповнювач і змінює позу;
- стійкість алгоритму детекції до характерних поведінкових патернів тварини (зокрема повторних підходів до лотка);
- надійність передавання даних бездротовим каналом і їх відображення користувачу.

Оскільки дослідження проводилися з реальною твариною в природних для неї умовах, вони мали характер натурних випробувань, у ході яких виявлялися практичні проблеми, не очевидні на етапі проектування. Виявлення та усунення цих проблем становить основний зміст експериментальної частини роботи й описане в наступних підрозділах.

4.2 Калібрування вимірювального тракту

Першим етапом досліджень було калібрування вимірювального тракту, оскільки без нього показання АЦП не відповідають реальним значенням маси. Методику калібрування описано у п. 2.5.1; тут наведено її практичну реалізацію та результати.

Калібрування виконувалося методом підбору калібрувального коефіцієнта за еталонною масою. На вимірювальну платформу встановлювався еталонний вантаж відомої маси 2 кг, після чого значення калібрувального коефіцієнта в програмі послідовно коригувалося доти, доки показання пристрою не збіглося з дійсною масою еталона. Підібране таким чином значення коефіцієнта було записане у програму мікроконтролера як постійний параметр.

Для перевірки правильності калібрування використовувався контрольний вантаж іншої маси — 1 кг, що не застосовувався під час калібрування. Показання пристрою з контрольним вантажем відповідали його дійсній масі в межах допустимої похибки, що підтвердило коректність визначеного калібрувального коефіцієнта та лінійність характеристики вимірювального тракту в робочому діапазоні. Лінійність є очікуваною властивістю тензометричного

					PE22.404474.001 ПЗ	Лис
						48
Зм.	Лис	№ докум.	Підпис	Дата		

тракту, і її підтвердження контрольним вантажем дозволяє застосовувати єдиний калібрувальний коефіцієнт для всього робочого діапазону мас.

4.3 Дослідження роботи в реальних умовах

Після калібрування комплекс було випробувано в реальних умовах експлуатації — із встановленням під лоток, яким користується тварина. Натурні випробування виявили низку практичних проблем, пов'язаних із природною поведінкою тварини, які не були очевидними на етапі проектування. Нижче описано виявлені проблеми та запропоновані й реалізовані рішення.

4.3.1 Зниження результату вимірювання внаслідок закопування

Під час перших випробувань було виявлено, що значення маси, які фіксує пристрій, систематично занижені та помітно різняться від походу до походу. Аналіз поведінки тварини показав причину: кіт не стоїть на платформі нерухомо всією вагою. Значну частину часу перебування в лотку тварина закопує наповнювач, спираючись передніми лапами на бортик лотка. У ці моменти частина ваги тварини передається на бортик, а не на вимірювальну платформу, через що виміряне значення маси виявляється заниженим. Оскільки тривалість і характер закопування щоразу різні, занижене значення також щоразу відрізнялося, що й давало розкид результатів.

Початковий алгоритм обчислював масу як середнє значення зі стабільних вимірювань за час відвідування. Проте за наявності тривалих фаз закопування таке середнє враховувало й занижені значення, через що підсумкова маса виявлялася меншою за дійсну.

Для усунення цієї проблеми алгоритм визначення маси було вдосконалено: замість усереднення стабільних значень застосовано пошук максимального стабільного значення за час відвідування. Обґрунтування полягає в тому, що момент, коли тварина стоїть на платформі всіма лапами (не спираючись на бортик), відповідає найбільшому виміряному значенню — саме воно є найближчим до дійсної маси тварини. Закопування ж, навпаки, завжди дає занижені

					РЕ22.404474.001 ПЗ	Лис
						49
Зм.	Лис	№ докум.	Підпис	Дата		

значення, які тепер не впливають на результат. Детально вдосконалений алгоритм описано у п. 3.2.3.

Після впровадження цього вдосконалення повторні випробування показали, що значення маси, які фіксуються для різних відвідувань, стали стабільними й узгодженими між собою, тоді як раніше вони мали значний розкид. Це підтвердило, що перехід до максимального стабільного значення усунув залежність результату від фази заковування.

4.3.2 Хибне подвоєння відвідувань через повторні підходи

Другою виявленою проблемою було подвоєння подій: одне фактичне відвідування лотка інколи фіксувалося як два окремі. Аналіз показав, що причиною є характерна поведінка тварини — кіт виходить із лотка, а через кілька секунд повертається, щоб закопати наповнювач. Початковий алгоритм фіксував завершення відвідування одразу після короткочасного падіння ваги нижче порогу, тому вихід і повернення сприймалися як два різні візити з двома окремими (до того ж різними за значенням) результатами маси.

Для усунення цієї проблеми було збільшено час підтвердження виходу тварини. Якщо в початковій версії завершення сесії фіксувалося вже після 1,5 с відсутності навантаження, то у вдосконаленому алгоритмі перехід до завершення відбувається лише за умови, що вага залишається нижчою за поріг безперервно протягом 20 с. Якщо протягом цього часу тварина повертається, відвідування продовжує вважатися єдиною сесією. Таким чином, короткочасний вихід для повернення й заковування більше не розриває відвідування на дві події.

Вибір інтервалу 20 с достатній, щоб охопити типову паузу між виходом і поверненням тварини, але не настільки великий, щоб два справді окремі відвідування злилися в одне. Аналіз впливу цього інтервалу на енергоспоживання показав, що його збільшення є несуттєвим для автономності, оскільки додатковий час активного режиму припадає лише на завершення кожного відвідування, а не на основний черговий режим роботи.

					PE22.404474.001 ПЗ	Лис
						50
Зм.	Лис	№ докум.	Підпис	Дата		

4.4 Перевірка передавання даних та інтерфейсу користувача

Окремо перевірялася робота каналу передавання даних і відображення інформації користувачу. За фактом кожного зафіксованого відвідування мікроконтролер вмикав модуль Wi-Fi, підключався до точки доступу, передавав сформований JSON-пакет на сервер і вимикав радіомодуль. Сервер успішно приймав надіслані дані, після чого Telegram-бот надсилав користувачу повідомлення про зафіксоване відвідування із зазначенням часу та виміряної маси. Приклад такого повідомлення наведено на рисунку 4.2.



Рисунок 4.2 — Приклад повідомлення Telegram-бота про зафіксоване відвідування

Випробування підтвердили, що ланцюг передавання даних від мікроконтролера до користувача працює коректно: дані, виміряні пристроєм, доходять до сервера, зберігаються в історії та відображаються користувачу у вигляді зрозумілого повідомлення. Перевірено також збереження даних на постійному томі сервера — кожне відвідування додається до файлу історії, що є основою для подальшого аналізу накопиченої статистики.

4.5 Перевірка роботи серверного аналізу даних

Окремо перевірялася робота алгоритму аналізу накопиченої статистики (п. 3.4.3). Оскільки накопичення достатнього обсягу реальних даних потребує кількох тижнів безперервної експлуатації, для перевірки алгоритму було використано підготовлену тестову вибірку, що імітує два характерні сценарії — нормальний стан тварини та стан із тривожною зміною маси. Такий підхід дозволяє перевірити коректність роботи алгоритму, не очікуючи тривалого періоду накопичення реальних даних.

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		51

Перший сценарій імітує здорову тварину зі стабільною масою: значення маси протягом усього періоду спостереження коливаються в межах природної варіативності близько сталого рівня (середня маса близько 3,3 кг). Зміна середньої маси за останній тиждень відносно базової лінії становить лише +2,0 %, що відповідає природному коливанню. У цьому випадку алгоритм коректно визначає стан як нормальний — тенденція до зниження відсутня, а зміна не перевищує порогового значення.

Результат звіту для цього сценарію наведено на рисунку 4.3.



Рисунок 4.3 — Звіт системи для сценарію нормального стану

Другий сценарій імітує поступову втрату маси: середня маса тварини за короткий період (7 днів) знизилася на 11,7 % відносно базової лінії за триваліші періоди — з близько 3,26 кг (середня за 90 днів) до 2,90 кг. У цьому випадку алгоритм коректно виявляє тенденцію до зниження (послідовне зменшення середніх за 90, 30 і 7 днів) та перевищення порогового значення 10 % і формує відповідне попередження користувачу.

Результат звіту для цього сценарію наведено на рисунку 4.4.

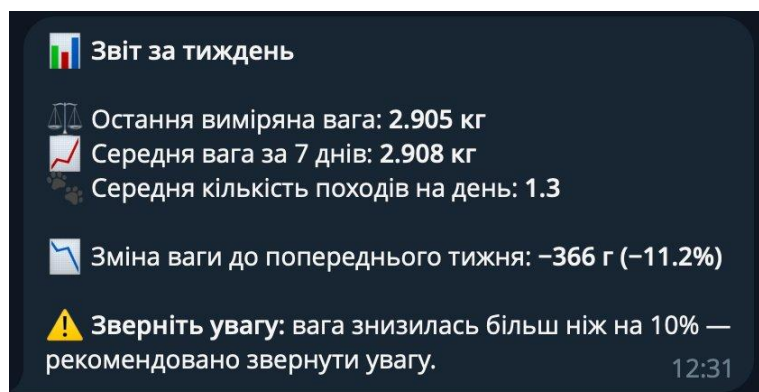


Рисунок 4.4 — Звіт системи для сценарію тривожної зміни маси

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		52

Випробування підтвердили, що алгоритм аналізу коректно розрізняє нормальний і тривожний стани: за стабільної маси система повідомляє про відсутність відхилень, а за зниження маси понад встановлений поріг — формує попередження. Це підтверджує працездатність реалізованого механізму аналізу тенденції маси за періодами 7, 30 і 90 днів.

Висновки до розділу 4

1. Проведено експериментальні дослідження розробленого комплексу в реальних умовах із домашнім котом. Виконано калібрування вимірювального тракту методом підбору за еталонною масою 2 кг та перевірено його коректність контрольним вантажем 1 кг, що підтвердило лінійність і правильність вимірювань.

2. У ході натурних випробувань виявлено проблему систематичного заниження маси внаслідок закопування наповнювача, коли тварина спирається на бортик лотка. Проблему усунено вдосконаленням алгоритму — переходом від усереднення до пошуку максимального стабільного значення. Повторні випробування показали, що результати вимірювання маси для різних відвідувань стали стабільними та узгодженими.

3. Виявлено проблему хибного подвоєння відвідувань через повторні підходи тварини до лотка. Проблему усунено збільшенням часу підтвердження виходу до 20 с, завдяки чому короткочасний вихід із наступним поверненням фіксується як одне відвідування.

4. Перевірено роботу каналу передавання даних та інтерфейсу користувача. Підтверджено, що дані коректно передаються від пристрою на сервер, зберігаються в історії та відображаються користувачу через Telegram-бот.

5. Перевірено роботу алгоритму аналізу даних на підготовленій тестовій вибірці. Підтверджено, що система коректно розрізняє нормальний стан тварини та стан із тривожною зміною маси, формуючи у другому випадку попередження користувачу.

					<i>РЕ22.404474.001 ПЗ</i>	Лис
						53
Зм.	Лис	№ докум.	Підпис	Дата		

6. Результати випробувань підтвердили працездатність комплексу й водночас дозволили вдосконалити алгоритми його роботи на основі реальної поведінки тварини, що є важливим результатом саме натурних, а не лабораторних досліджень.

					РЕ22.404474.001 ПЗ	Лис
						54
Зм.	Лис	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломному проєкті розроблено програмно-апаратний комплекс для дистанційного моніторингу поведінки домашніх тварин, що забезпечує автоматичне вимірювання маси тварини та частоти відвідувань лотка, передавання даних на сервер, їх аналіз і сповіщення користувача про відхилення. Поставлену мету досягнуто, усі завдання роботи виконано.

За результатами роботи зроблено такі висновки.

1. Проаналізовано предметну область і встановлено, що маса тіла та частота відвідувань лотка є об'єктивними показниками, які дозволяють виявити погіршення стану тварини на ранній стадії. Обґрунтовано, що стійке зниження маси понад 10 % є клінічно значущою ознакою, а зміна частоти сечовипускання характерна для захворювань сечовидільної системи. Показано, що систематичне відстеження цих показників у домашніх умовах потребує автоматизованого засобу.

2. Розроблено структурні схеми апаратної та програмної частин комплексу з розподіленою клієнт-серверною архітектурою. Апаратна частина представлена автономним вимірювальним пристроєм під лотком, а програмна охоплює хмарний сервер, модуль аналізу та клієнтський інтерфейс (Telegram-бот), що взаємодіють між собою через бездротовий зв'язок Wi-Fi. Обґрунтовано вибір елементної бази апаратного вузла: мікроконтролера ESP32-S2 із вбудованим Wi-Fi та режимом глибокого сну, чотирьох півмостових тензодатчиків, аналого-цифрового перетворювача HX711 та автономного живлення від лужних батарейок.

3. Розроблено апаратну частину пристрою — вимірювальний тракт на основі чотирьох тензодатчиків, з'єднаних у повну мостову схему. Виконано калібрування тракту та оцінено час автономної роботи. Запропоновано конструктивні рішення, що враховують специфіку експлуатації поряд із твариною, зокрема відмову від зовнішніх кабелів живлення з міркувань безпеки.

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		55

4. Розроблено вбудоване програмне забезпечення мікроконтролера, що реалізує енергоощадний циклічний алгоритм роботи з режимом глибокого сну. Запропоновано алгоритм пошуку максимальної стабільної ваги, який забезпечує коректне визначення маси тварини в умовах динамічних навантажень і дозволяє уникнути заниження результату під час руху тварини.

5. Розроблено серверну частину комплексу, що приймає та зберігає дані у файловому сховищі й виконує їх аналіз шляхом обчислення середньої маси за періоди 7, 30 і 90 днів та визначення тенденції її зміни з порівнянням із пороговим значенням, обґрунтованим медичними даними. Реалізовано інтерфейс взаємодії з користувачем через Telegram-бот із функціями автоматичного інформування та формування звітів за запитом.

6. Проведено експериментальні дослідження комплексу в реальних умовах на прикладі домашнього кота. У ході натурних випробувань виявлено та усунуено практичні проблеми роботи (заниження маси під час закопування та хибне подвоєння відвідувань), а також підтверджено працездатність усіх основних функцій комплексу, зокрема коректне розрізнення нормального та тривожного станів за результатами аналізу даних.

Практична цінність роботи полягає в тому, що розроблений комплекс дозволяє в автоматичному режимі, без участі власника та без стресу для тварини, відстежувати її стан і вчасно виявляти ознаки можливого захворювання.

Подальший розвиток роботи можливий у таких напрямках: розробка завершеного вологозахищеного корпусу з жорсткою вимірною платформою та зручним відсіком для заміни елементів живлення; оптимізація енергоспоживання для збільшення часу автономної роботи; розширення переліку контрольованих показників; адаптація конструкції для інших видів домашніх тварин (дрібних собак, кролів); удосконалення алгоритмів аналізу даних на основі довготривалих статистичних спостережень.

					РЕ22.404474.001 ПЗ	Лис
						56
Зм.	Лис	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hostutler R. A., Chew D. J., DiBartola S. P. Recent concepts in feline lower urinary tract disease. Veterinary Clinics of North America: Small Animal Practice. 2005. Vol. 35, No. 1. P. 147–170.
2. Dorsch R., Remer C., Sauter-Louis C., Hartmann K. Feline lower urinary tract disease in a German cat population: a retrospective analysis of demographic data, causes and clinical signs. Tierärztliche Praxis Ausgabe K: Kleintiere/Heimtiere. 2014. Vol. 42, No. 4. P. 231–239.
3. Warning signs of cancer : вебсайт. Cornell Feline Health Center, Cornell University College of Veterinary Medicine. URL: <https://www.vet.cornell.edu/departments-centers-and-institutes/cornell-feline-health-center/health-information/feline-health-topics/warning-signs-cancer> (дата звернення: 18.05.2026).
4. Freeman L. M., Lachaud M.-P., Matthews S., Rhodes L., Zollers B. Evaluation of weight loss over time in cats with chronic kidney disease. Journal of Veterinary Internal Medicine. 2016. Vol. 30, No. 5. P. 1661–1666.
5. The Versatile Strain Gauge Load Cell : Knowledge Base. Tacuna Systems. URL: <https://tacunasystems.com/knowledge-base/the-versatile-strain-gauge-load-cell/> (дата звернення: 19.05.2026).
6. HX711 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales : Datasheet. Avia Semiconductor. URL: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf (дата звернення: 29.05.2026).
7. SC902 Half Bridge Micro Load Cell, 50 kg : Datasheet. Sensor and Control Co., Ltd. URL: https://akizukidenshi.com/goodsaffix/SC902-50kg_half_bridge_load_cell.pdf (дата звернення: 29.05.2026).
8. ESP32-S2 Series Datasheet. Version 1.8. Espressif Systems, 2025. URL: https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf (дата звернення: 06.06.2026).

					PE22.404474.001 ПЗ	Лис
						57
Зм.	Лис	№ докум.	Підпис	Дата		

9. S2 Mini : документація. WEMOS Electronics. URL: https://www.wemos.cc/en/latest/s2/s2_mini.html (дата звернення: 13.05.2026).

10. aiohttp Documentation : Asynchronous HTTP Client/Server for asyncio and Python. URL: <https://docs.aiohttp.org> (дата звернення: 29.05.2026).

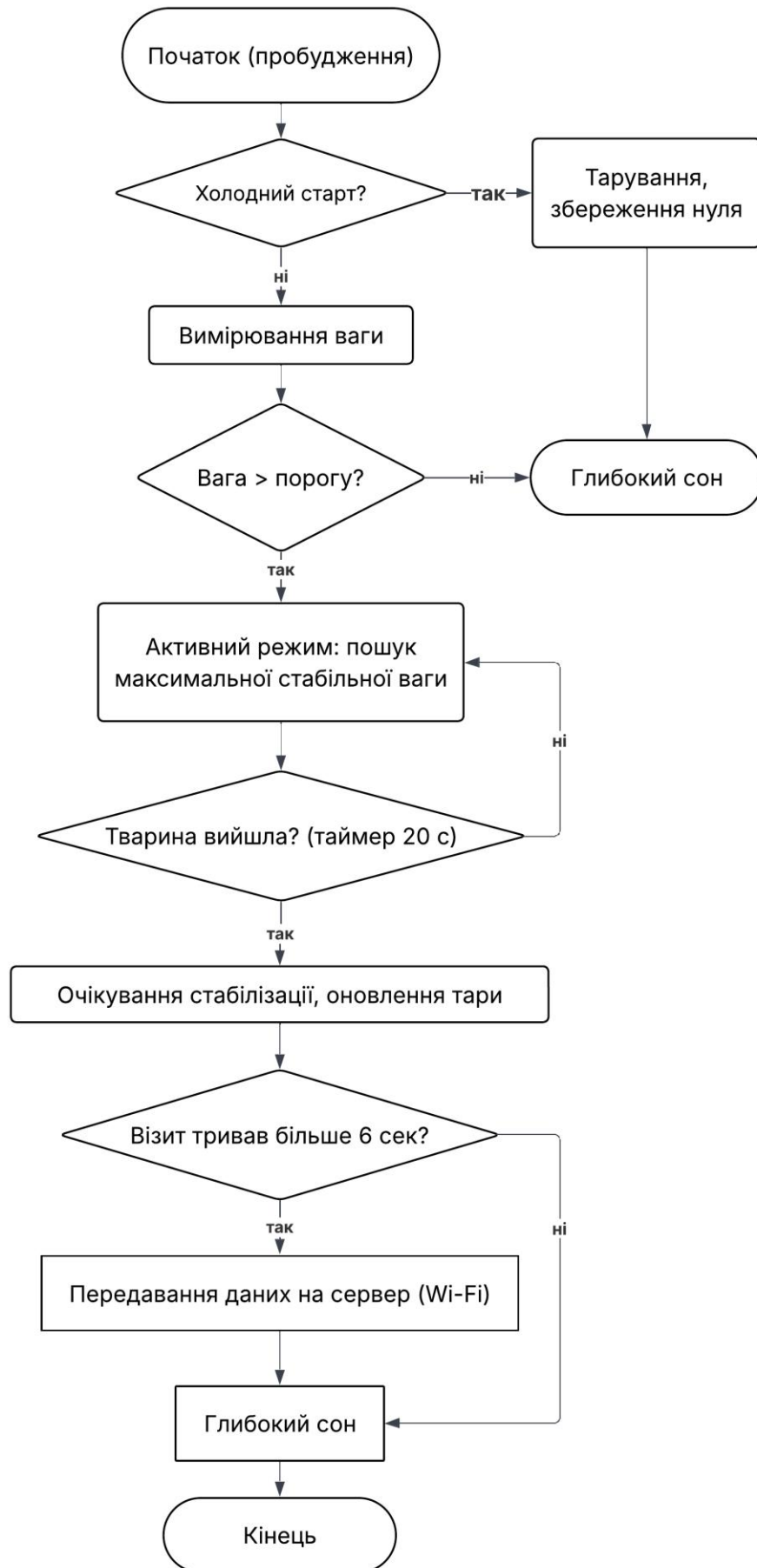
11. aiogram Documentation : Modern and fully asynchronous framework for Telegram Bot API. URL: <https://docs.aiogram.dev> (дата звернення: 29.05.2026).

12. Railway Documentation : Cloud application hosting platform. URL: <https://docs.railway.app> (дата звернення: 29.05.2026).

13. Shin J.-W., Han S.-W., Kweon S.-H., Kang M., Kim J.-H., Choi C.-G., Chae J.-S. Comparison of Frequency and Stay Time between Normal and Abnormal Elimination Behavior of Cats Using a Litter Box with Automatic Sensor. Journal of Veterinary Clinics. 2024. Vol. 41, No. 2. P. 71–78.

					РЕ22.404474.001 ПЗ	Лис
						58
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК А



ДОДАТОК Б

```
#include <WiFi.h>
#include <HTTPClient.h>
#include "HX711.h"

// Налаштування мережі та вимірювального тракту
const char* ssid = "<WIFI_SSID>";
const char* password = "<WIFI_PASSWORD>";
const char* serverUrl = "<SERVER_URL>";
const int DOUT_PIN = 5; // лінія даних DT модуля HX711
const int SCK_PIN = 6; // лінія тактів SCK модуля HX711
HX711 scale;

float calibration_factor = -21752.50; // калібрувальний коефіцієнт тракту
// Змінні в RTC-пам'яті (зберігаються під час глибокого сну)
RTC_DATA_ATTR float m_base = 0.0; // нульовий рівень (тара)
RTC_DATA_ATTR long rtc_offset = 0; // апаратне зміщення нуля бібліо-
теки HX711

// Стан активного режиму (кіт у лотку)
enum State { CAT_INSIDE, SETTLING };
State currentState = CAT_INSIDE;
unsigned long event_start_time = 0;
unsigned long settling_start_time = 0;
unsigned long last_sensor_read = 0;
float previous_weight = 0.0;
float max_stable_weight = 0.0; // максимальне стабільне значення за від-
відування

unsigned long cat_exit_timer = 0;
void sendDataToServer(String type, int duration, float cat_w);
void setup() {
    Serial.begin(115200);
```

					РЕ22.404474.001 ПЗ	Лис
						60
Зм.	Лис	№ докум.	Підпис	Дата		

```

WiFi.mode(WIFI_OFF); // радіо вимкнене у черговому режимі
scale.begin(DOUT_PIN, SCK_PIN);
scale.set_scale(calibration_factor);
esp_sleep_wakeup_cause_t wakeup_reason =
esp_sleep_get_wakeup_cause();

// Холодний старт (увімкнення): виконуємо тарування
if (wakeup_reason != ESP_SLEEP_WAKEUP_TIMER) {
    scale.power_up();
    delay(50);
    scale.tare();
    rtc_offset = scale.get_offset(); // зберігаємо апаратний нуль у RTC-па-
м'ять

    m_base = 0.0;
    scale.power_down();
    esp_sleep_enable_timer_wakeup(2000000); // інтервал опитування 2 с
    esp_deep_sleep_start();
} else {
    // Пробудження за таймером: повертаємо збережений нуль у бібліо-
теку

    scale.set_offset(rtc_offset);
}

// Швидка перевірка ваги після пробудження
scale.power_up();
delay(50);
float current_weight = scale.get_units(3);

// Лоток порожній: повільне коригування тари і повернення у сон
if (current_weight - m_base <= 1.0) {
    if (current_weight - m_base > -0.1 && current_weight - m_base < 0.1) {
        m_base = (m_base * 0.9) + (current_weight * 0.1); // експоненційне
згладжування тари
    }
}

```

					РЕ22.404474.001 ПЗ	Лис
						61
Зм.	Лис	№ докум.	Підпис	Дата		

```

scale.power_down();
esp_sleep_enable_timer_wakeup(2000000);
esp_deep_sleep_start();
}
// Вага перевищує поріг присутності: переходимо в активний режим
event_start_time = millis();
last_sensor_read = millis();
previous_weight = current_weight;
max_stable_weight = 0.0;
cat_exit_timer = 0;
currentState = CAT_INSIDE;
}

void loop() {
    if (millis() - last_sensor_read >= 1000) { // опитування раз на секунду
        float current_weight = scale.get_units(3);
        last_sensor_read = millis();
        switch (currentState) {
            case CAT_INSIDE:
                // Захист від застрягання в активному режимі: таймаут 10 хв
                if (millis() - event_start_time > 600000) {
                    sendDataToServer("error_timeout", 600, 0);
                    scale.tare();
                    rtc_offset = scale.get_offset();
                    m_base = 0.0;
                    esp_sleep_enable_timer_wakeup(2000000);
                    esp_deep_sleep_start();
                }
                // Пошук максимального стабільного значення ваги
                if ((current_weight - m_base >= 1.0) &&
                    (abs(current_weight - previous_weight) <= 0.08)) {
                    if (current_weight > max_stable_weight) {

```

					PE22.404474.001 ПЗ	Лис
						62
Зм.	Лис	№ докум.	Підпис	Дата		

```

        max_stable_weight = current_weight;
    }
}
previous_weight = current_weight;
// Підтвердження виходу: вага нижча за поріг безперервно 20 с
if (current_weight - m_base < 1.0) {
    if (cat_exit_timer == 0) {
        cat_exit_timer = millis();
    } else if (millis() - cat_exit_timer > 20000) {
        currentState = SETTling;
        settling_start_time = millis();
        cat_exit_timer = 0;
    }
} else {
    cat_exit_timer = 0; // вага повернулася — це та сама сесія
}
break;
case SETTling:
    // Очікування стабілізації платформи після виходу тварини
    if (millis() - settling_start_time > 10000) {
        float m_new = scale.get_units(10);
        int duration_sec = (settling_start_time - event_start_time) / 1000;
        // Підсумкова маса — максимальне стабільне значення за вираху-
        ванням тари
        float pure_cat_weight = 0.0;
        if (max_stable_weight > 0.0) {
            pure_cat_weight = max_stable_weight - m_base;
        } else {
            pure_cat_weight = previous_weight - m_base;
        }
        // Фільтр коротких відвідувань: коротші за 6 с не передаємо

```

					РЕ22.404474.001 ПЗ	Лис
						63
Зм.	Лис	№ докум.	Підпис	Дата		

```

        if (duration_sec >= 6) {
            sendDataToServer("cat", duration_sec, pure_cat_weight);
        }
        m_base = m_new; // новий нульовий рівень порожнього лотка
        scale.power_down();
        esp_sleep_enable_timer_wakeup(2000000);
        esp_deep_sleep_start();
    }
    break;
}
}
}

void sendDataToServer(String type, int duration, float cat_w) {
    // Wi-Fi вмикається лише на час передавання даних
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    int timeout_counter = 20; // тайм-аут підключення ~10 с
    while (WiFi.status() != WL_CONNECTED && timeout_counter > 0) {
        delay(500);
        timeout_counter--;
    }
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverUrl);
        http.addHeader("Content-Type", "application/json");
        // Формування JSON-пакета (поля за п. 3.3)
        String jsonPayload = "{}";
        jsonPayload += "\"event_type\": \"" + type + "\", ";
        jsonPayload += "\"duration\": " + String(duration) + ", ";
        jsonPayload += "\"cat_weight\": " + String(cat_w, 3);
        jsonPayload += "}";
    }
}

```

					<i>PE22.404474.001 ПЗ</i>	Лис
						64
Зм.	Лис	№ докум.	Підпис	Дата		

```

http.POST(jsonPayload);
http.end();
}
// Розрив з'єднання і вимкнення радіо одразу після передачі
WiFi.disconnect(true);
WiFi.mode(WIFI_OFF);
}

```

					РЕ22.404474.001 ПЗ	Лис
						65
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК В



Зм.	Лис	№ докум.	Підпис	Дата

PE22.404474.001 ПЗ

ДОДАТОК Г

```
import asyncio
import logging
import os
import json
from datetime import datetime, timedelta
from aiohttp import web
from aiogram import Bot, Dispatcher, Router, F, types
from aiogram.filters import CommandStart
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
BufferedInputFile

# Конфігурація
BOT_TOKEN = "<BOT_TOKEN>"
ADMIN_ID = 0 # Telegram ID власника (підставити свій)
WEB_SERVER_PORT = 8080
PERIOD_SHORT = 7 # поточний стан
PERIOD_MID = 30 # базова лінія
PERIOD_LONG = 90 # базова лінія
WEIGHT_DROP_THRESHOLD = 10.0 # поріг тривоги: % зниження
маси (Розділ 1)
MIN_BASELINE_RECORDS = 3 # мінімум записів у базі для аналізу
тенденції
DATA_DIR = os.environ.get("DATA_DIR", "/app/data") # постійний том
Railway
HISTORY_FILE = os.path.join(DATA_DIR, "history.json")
logging.basicConfig(level=logging.INFO)
bot = Bot(token=BOT_TOKEN)
dp = Dispatcher()
router = Router()
```

					РЕ22.404474.001 ПЗ	Лис
						67
Зм.	Лис	№ докум.	Підпис	Дата		

```

# Робота з файлом історії
def load_history() -> list:
    if not os.path.exists(HISTORY_FILE):
        return []

    try:
        with open(HISTORY_FILE, "r", encoding="utf-8") as f:
            return json.load(f)
    except (json.JSONDecodeError, ValueError):
        logging.warning("Файл історії пошкоджено або порожній.")
        return []

def save_visit(weight: float):
    try:
        os.makedirs(DATA_DIR, exist_ok=True)
        history = load_history()
        record = {
            "timestamp": datetime.now().isoformat(timespec="seconds"),
            "weight": round(float(weight), 3)
        }
        history.append(record)
        with open(HISTORY_FILE, "w", encoding="utf-8") as f:
            json.dump(history, f, ensure_ascii=False, indent=2)
        logging.info(f"Візит збережено: {record}. Усього записів:
{len(history)}")
    except Exception as e:
        logging.error(f"Помилка збереження візиту: {e}")

# Допоміжні функції аналізу
def _parse_records(history: list):
    parsed = []
    for rec in history:
        try:

```

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		68

```

        ts = datetime.fromisoformat(rec["timestamp"])
        w = float(rec["weight"])
    except (KeyError, ValueError, TypeError):
        continue
    parsed.append((ts, w))
parsed.sort(key=lambda x: x[0])
return parsed

def _avg_in_window(records, now: datetime, days: int):
    cutoff = now - timedelta(days=days)
    vals = [w for ts, w in records if ts >= cutoff]
    if not vals:
        return None, 0
    return sum(vals) / len(vals), len(vals)

def _avg_baseline(records, now: datetime, days: int):
    # база — записи віком (7; days] днів, без перетину з поточним тижнем
    start = now - timedelta(days=days)
    end = now - timedelta(days=PERIOD_SHORT)
    vals = [w for ts, w in records if start <= ts < end]
    if not vals:
        return None, 0
    return sum(vals) / len(vals), len(vals)

# Алгоритм аналізу (середня за 7/30/90 днів + тенденція)
def build_report() -> str:
    history = load_history()
    if not history:
        return "    <b>Звіт</b>\n\nІсторія порожня — ще немає збережених  
візитів."
    records = _parse_records(history)
    if not records:
        return "    <b>Звіт</b>\n\nКоректних записів не знайдено."

```

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		69

```

now = datetime.now()
last_ts, last_weight = records[-1]
oldest_ts = records[0][0]
span_days = (now - oldest_ts).days

avg_7, visits_7 = _avg_in_window(records, now, PERIOD_SHORT)
avg_30, visits_30 = _avg_in_window(records, now, PERIOD_MID)
avg_90, visits_90 = _avg_in_window(records, now, PERIOD_LONG)

# середня частота на день = візити / фактична к-сть днів (захист від ді-
лення на 0)

days_7 = min(PERIOD_SHORT, span_days + 1)
days_30 = min(PERIOD_MID, span_days + 1)
days_90 = min(PERIOD_LONG, span_days + 1)
freq_7 = visits_7 / days_7 if days_7 else 0.0
freq_30 = visits_30 / days_30 if days_30 else 0.0
freq_90 = visits_90 / days_90 if days_90 else 0.0

lines = ["    <b>Звіт</b>\n"]

lines.append(f"    Остання вимірюна вага: <b>{last_weight:.3f}</b> кг")

baseline_days = PERIOD_LONG if span_days > PERIOD_MID else
PERIOD_MID

base_avg, base_n = _avg_baseline(records, now, baseline_days)

enough_for_trend = (
    avg_7 is not None
    and base_avg is not None
    and base_n >= MIN_BASELINE_RECORDS
    and span_days > PERIOD_SHORT
)

if not enough_for_trend:
    # гілка «недостатньо даних»: звіт лише з поточними даними
    lines.append(f"    Середня вага за 7 днів: <b>{avg_7:.3f}</b> кг")

```

					РЕ22.404474.001 ПЗ	Лис
						70
Зм.	Лис	№ докум.	Підпис	Дата		

```

lines.append(f"    Відвідувань на день (7 дн): <b>{freq_7:.1f}</b>")
lines.append("")
lines.append("i <i>Даних поки недостатньо для визначення тенден-
ції. "

        "Звіт сформовано лише з поточними даними; аналіз дина-
міки "

        "стане доступним після накопичення статистики за тривалі-
ший період.</i>")

return "\n".join(lines)

lines.append(f"    Середня вага: 7 дн — <b>{avg_7:.3f}</b> кг</b>")
if avg_30 is not None:
    lines.append(f"        30 дн — <b>{avg_30:.3f}</b> кг</b>")
if avg_90 is not None and span_days > PERIOD_MID:
    lines.append(f"        90 дн — <b>{avg_90:.3f}</b> кг</b>")
lines.append(f"\n    Відвідувань на день: 7 дн — <b>{freq_7:.1f}</b>")
if avg_30 is not None:
    lines.append(f"        30 дн — <b>{freq_30:.1f}</b>")
if avg_90 is not None and span_days > PERIOD_MID:
    lines.append(f"        90 дн — <b>{freq_90:.1f}</b>")
# тенденція: поточний тиждень відносно базової лінії
diff_g = (avg_7 - base_avg) * 1000.0
diff_pct = (avg_7 - base_avg) / base_avg * 100.0 if base_avg > 0 else 0.0
sign = "+" if diff_g >= 0 else "-"
if diff_pct <= -3.0:
    trend_word = "тенденція до зниження"
elif diff_pct >= 3.0:
    trend_word = "тенденція до зростання"
else:
    trend_word = "стабільна"

```

					РЕ22.404474.001 ПЗ	Лис
						71
Зм.	Лис	№ докум.	Підпис	Дата		

```

lines.append(
    f"\n    Динаміка маси ({trend_word}) відносно базової лінії за "
    f"{baseline_days} дн: <b>{sign} {abs(diff_g):.0f} г
({sign} {abs(diff_pct):.1f}%)</b>"
)
# порівняння з порогом за масою
lines.append("")
if diff_pct <= -WEIGHT_DROP_THRESHOLD:
    lines.append(
        f"□ <b>Зверніть увагу:</b> вага знизилась більш ніж на "
        f"{WEIGHT_DROP_THRESHOLD:.0f}% — рекомендовано звер-
нути увагу."
    )
else:
    lines.append("✓<b>Стан у межах норми.</b>")
return "\n".join(lines)
# Інтерфейс Телеграм-бота
def get_main_menu() -> ReplyKeyboardMarkup:
    kb = [
        [KeyboardButton(text="    Звіт")],
        [KeyboardButton(text="    Експорт даних"), KeyboardButton(text="
Статус")],
    ]
    return ReplyKeyboardMarkup(keyboard=kb, resize_keyboard=True)
@router.message(CommandStart())
async def cmd_start(message: types.Message):
    if message.from_user.id != ADMIN_ID:
        await message.answer("Доступ заборонено.")
    return
    await message.answer(

```

					РЕ22.404474.001 ПЗ	Лис
						72
Зм.	Лис	№ докум.	Підпис	Дата		

```

"Привіт, Лізо! Система моніторингу активована.\n\n"
"• <b>Звіт</b> — динаміка маси за 7 / 30 / 90 днів і тенденція\n"
"• <b>Експорт даних</b> — отримати файл історії\n"
"• Щоб <b>оновити дані</b> — просто надішли боту виправлений
файл history.json\n"
"• <b>Статус</b> — кількість записів",
reply_markup=get_main_menu(),
parse_mode="HTML"
)
@router.message(F.text == "    Статус")
async def check_status(message: types.Message):
    if message.from_user.id != ADMIN_ID:
        return
    count = len(load_history())
    await message.answer(
        "Система в режимі очікування. Дані від ESP32 оновлюються авто-
матично.\n"
        f"Збережено записів у історії: {count}"
    )
@router.message(F.text == "    Звіт")
async def send_report(message: types.Message):
    if message.from_user.id != ADMIN_ID:
        return
    await message.answer(build_report(), parse_mode="HTML")
@router.message(F.text == "    Експорт даних")
async def export_data(message: types.Message):
    if message.from_user.id != ADMIN_ID:
        return
    history = load_history()
    if not history:

```

					РЕ22.404474.001 ПЗ	Лис
						73
Зм.	Лис	№ докум.	Підпис	Дата		

```

        await message.answer("Історія порожня — немає що експортувати.")
        return
content = json.dumps(history, ensure_ascii=False, indent=2).encode("utf-
8")

file = BufferedInputFile(content, filename="history.json")
await message.answer_document(
    file,
    caption="    Поточна історія візитів.\nВідредагуй за потреби й наді-
шли файл назад, щоб оновити дані."
)
@router.message(F.document)
async def import_data(message: types.Message):
    if message.from_user.id != ADMIN_ID:
        return
    doc = message.document
    if not doc.file_name.lower().endswith(".json"):
        await message.answer("Очікується файл .json (history.json). Файл не
прийнято.")
        return
    try:
        file_info = await bot.get_file(doc.file_id)
        file_bytes = await bot.download_file(file_info.file_path)
        raw = file_bytes.read()
        data = json.loads(raw.decode("utf-8"))
        if not isinstance(data, list):
            await message.answer("Помилка: файл має містити список записів
(масив JSON). Дані не змінено.")
            return
        valid = []
        for rec in data:

```

					РЕ22.404474.001 ПЗ	Лис
						74
Зм.	Лис	№ докум.	Підпис	Дата		

```

        if isinstance(rec, dict) and "timestamp" in rec and "weight" in rec:
            valid.append({
                "timestamp": str(rec["timestamp"]),
                "weight": round(float(rec["weight"]), 3)
            })
    os.makedirs(DATA_DIR, exist_ok=True)
    with open(HISTORY_FILE, "w", encoding="utf-8") as f:
        json.dump(valid, f, ensure_ascii=False, indent=2)
    await message.answer(f"✓Історію оновлено. Прийнято записів:
{len(valid)}.")
    logging.info(f"Історію замінено імпортом. Записів: {len(valid)}")
except json.JSONDecodeError:
    await message.answer("Помилка: файл не є коректним JSON. Дані не
змінено.")
except Exception as e:
    logging.error(f"Помилка імпорту: {e}")
    await message.answer(f"Помилка обробки файлу: {e}. Дані не змі-
нено.")

# Вебсервер (приймання даних від ESP32)
async def handle_sensor_data(request: web.Request):
    try:
        data = await request.json()
        event_type = data.get("event_type", "unknown")
        duration = data.get("duration", 0)
        cat_weight = data.get("cat_weight", 0.0)

        logging.info(f"Отримано дані від ESP32: Подія={event_type}, Трива-
лість={duration}с, Вага={cat_weight}кг")

        if event_type == "cat":
            save_visit(cat_weight)

```

					РЕ22.404474.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		75

```

report_text = (
    "    <b>Новий візит зафіксовано!</b>\n\n"
    f"    □ Тривалість: {duration} сек\n"
    f"    □ Вага котика: {cat_weight:.3f} кг"
)

await bot.send_message(chat_id=ADMIN_ID, text=report_text,
parse_mode="HTML")

elif event_type == "error_timeout":
    report_text = (
        "    □ <b>Увага!</b> Вага понад 1 кг тримається більше 10 хви-
лин.\n"
        "    Лоток примусово обнулено і переведено в режим очіку-
вання.\n"
        "    Перевірте, чи не залишили ви щось важке на вагах."
    )
    await bot.send_message(chat_id=ADMIN_ID, text=report_text,
parse_mode="HTML")
else:
    logging.info(f"Невідомий тип події ({event_type}), повідомлення
не надіслано.")

    return web.json_response({"status": "success", "message": "Data
processed"})
except Exception as e:
    logging.error(f"Помилка обробки даних від ESP32: {e}")
    return web.json_response({"status": "error", "message": str(e)},
status=400)

# Паралельний запуск вебсервера та бота
async def main():

```

					РЕ22.404474.001 ПЗ	Лис
						76
Зм.	Лис	№ докум.	Підпис	Дата		

```

dp.include_router(router)
app = web.Application()
app.router.add_post('/api/sensor', handle_sensor_data)
runner = web.AppRunner(app)
await runner.setup()
site = web.TCPSite(runner, '0.0.0.0', WEB_SERVER_PORT)
logging.info(f'Локальний сервер запущено. Очікування даних на порту
{WEB_SERVER_PORT}...')
await site.start()
logging.info("Telegram-бот запущено...")
await dp.start_polling(bot)
if __name__ == "__main__":
    asyncio.run(main())

```

					РЕ22.404474.001 ПЗ	Лис
						77
Зм.	Лис	№ докум.	Підпис	Дата		