

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4  
Назва теми

КВРКІ 210105.21.01.24 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконав: студент IV курсу, група KI2-21-1

  
Підпис

Владислав БУРДАШ  
Ініціали, прізвище

Керівник

  
Підпис, дата

Сергій ЛИСЕНКО  
Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

«9» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Владиславу БУРДАШУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4

Керівник проекту (роботи) Сергій ЛИСЕНКО, д.т.н., проф.

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

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Теоретичні основи досліджуваної теми \_\_\_\_\_

Компоненти проекту та середовище розроблення \_\_\_\_\_

Реалізація системи \_\_\_\_\_

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

Схема монтажна \_\_\_\_\_

Схема структурна \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 -аналіз предметної області, актуальність, порівняння рішень, огляд Raspberry Pi 4, методи розв'язання та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 -вибір апаратних і програмних компонентів, проектування системи, розробка модулів автоматизації.	01.04.2025	виконано
15	Робота над розділом 3 -створення схем, розробка системного програмного забезпечення, апаратних рішень, інтеграція та тестування	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	23.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Владислав БУРДАШ  
Ініціали, прізвище

Керівник роботи

Підпис

Сергій ЛИСЕНКО  
Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4».

Автор роботи: Владислав Бурдаш.

Керівник роботи: Лисенко Сергій Миколайович.

Пояснювальна записка: 61 с., 4 рис., 2 табл., 4 дод., 52 джерела.

Графічна частина: 3 креслення.

IOT, RASPBERRY PI 4, RPI.GPIO, MQTT, FLASK, GPIO, HTTP.

Мета роботи полягає у розробці системи догляду за домашніми тваринами на базі мікрокомп'ютера Raspberry Pi 4, яка забезпечує автоматизацію ключових процесів (годування, моніторинг, інтерактивна взаємодія) з можливістю віддаленого керування через мережу Інтернет.

Об'єкт дослідження – системи автоматизованого догляду за домашніми тваринами.

Предмет дослідження – принципи створення системи догляду на базі мікрокомп'ютера Raspberry Pi 4, включаючи апаратне та програмне забезпечення.

Під час проведення даного дослідження був використаний аналітичний, експериментальний метод та системний підхід для розробки прототипу системи, який може бути використаний власниками домашніх тварин для автоматизації догляду, а також як основа для подальших комерційних розробок.





  
Підпис студента

30.05.2025

Дата

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ</b> .....	6
1.1 Аналіз предметної області і визначення актуальних проблем і завдань .....	6
1.2 Актуальність застосування систем догляду за домашніми тваринами .....	9
1.3 Порівняльний аналіз переваг та недоліків існуючих рішень .....	11
1.3.1 Існуючі варіанти систем догляду за домашніми тваринами .....	13
1.3.2 Особливості систем .....	15
1.4 Raspberry Pi 4 .....	17
1.5 Методи розв'язання проблеми в рамках досліджуваної теми .....	19
1.6 Постановка задачі .....	20
1.7 Висновки до першого розділу .....	22
<b>2 КОМПОНЕНТИ ПРОЕКТУ ТА СЕРЕДОВИЩЕ РОЗРОБЛЕННЯ</b> .....	23
2.1 Огляд компонентів проекту .....	23
2.2 Вибір апаратних складових системи .....	28
2.3 Середовище розроблення програмного забезпечення .....	32
2.4 Інтеграція апаратних і програмних компонентів .....	37
2.5 Висновки до другого розділу .....	42
<b>3 РЕАЛІЗАЦІЯ СИСТЕМИ</b> .....	44
3.1 Розробка схем системи .....	44
3.2 Системне програмне забезпечення .....	46
3.3 Апаратні рішення та їх конфігурація .....	50
3.4 Інтеграція датчиків і модулів із системою .....	52
3.5 Тестування та налагодження системи .....	58
3.6 Висновки до третього розділу .....	62

					КвРКІ.210105.21.01.24 ПЗ			
Зм.	Арк.	Недокум.	Підпис	Дата				
Виконав		Владислав БУРДАШ			Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4.	Літера	Аркуш	Аркушів
Перевір.		Сергій ЛИСЕНКО				у	2	61
Н.контр.		Тетяна КИСІЛЬ		19.05.21	Пояснювальна записка	ХНУ КІ2-21-1		
Затвер.		Ольга ПАВЛОВА		19.05.21				

<b>ВИСНОВКИ</b> .....	63
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	65
<b>ДОДАТОК А</b> .....	71
<b>ДОДАТОК Б</b> .....	72
<b>ДОДАТОК В</b> .....	73

						КвРКІ.210105.21.01.24ПЗ	Арк.
							3
Зм.	Арк.	№ докум.	Підпис	Дата			

## ВСТУП

У сучасному світі дедалі більше людей тримають домашніх тварин, розглядаючи їх як повноправних членів сім'ї. За даними досліджень, у 2024 році близько 65% домогосподарств у розвинених країнах мали принаймні одну домашню тварину, що свідчить про зростання попиту на продукти та послуги для їхнього догляду. Водночас швидкий ритм життя, віддалена робота та часті подорожі створюють проблему: власники не завжди можуть забезпечити належний догляд за тваринами, зокрема регулярне годування, моніторинг стану здоров'я чи інтерактивну взаємодію. Це призводить до стресу як у тварин, так і у їхніх господарів.

Традиційні рішення, такі як автоматичні годівниці чи найм доглядачів, мають свої обмеження: висока вартість, обмежена функціональність або відсутність віддаленого контролю. Сучасні технології, зокрема Інтернет речей (IoT) і мікрокомп'ютерні системи, відкривають нові можливості для створення доступних, гнучких і розумних систем догляду. Raspberry Pi 4, як потужний і недорогий мікрокомп'ютер, ідеально підходить для таких завдань завдяки своїм технічним характеристикам, підтримці широкого спектра датчиків і модулів, а також відкритості для програмування. В Україні, де популярність Raspberry Pi зростає (зокрема через доступність через дистриб'юторів, таких як EVO чи Raspberry Pi Україна), розробка подібних систем є особливо актуальною, адже вона відповідає як глобальним трендам, так і локальним потребам.

Наукова новизна роботи полягає у створенні доступної системи догляду за домашніми тваринами, яка поєднує модульність, віддалене керування та інтеграцію з IoT-технологіями на базі Raspberry Pi 4. На відміну від комерційних рішень, запропонована система є економічно вигідною, відкритою для модифікацій і адаптованою до потреб користувачів, зокрема в українському контексті.

Практична цінність роботи полягає у розробці прототипу системи, який може бути використаний власниками домашніх тварин для автоматизації догляду, а

					КВРКІ.210105.21.01.24 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

також як основа для подальших комерційних розробок. Система може бути адаптована для різних типів тварин (собак, котів, птахів) і сценаріїв використання (годування, моніторинг, ігрова взаємодія). Крім того, результати роботи можуть бути застосовані в освітніх цілях для вивчення IoT і мікрокомп'ютерних систем.

У результаті роботи буде створено прототип системи догляду за домашніми тваринами, який включає автоматичну годівницю, модуль відеоспостереження та можливість віддаленого керування через вебінтерфейс або мобільний додаток. Система буде протестована в реальних умовах, а її ефективність оцінена за критеріями надійності, точності виконання завдань і зручності використання. Результати роботи можуть стати основою для подальших досліджень у сфері IoT і автоматизації побутових процесів.

					КВРКІ.210105.21.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# 1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

## 1.1 Аналіз предметної області і визначення актуальних проблем і завдань

При обговоренні автоматизації в побуті, найчастіше згадуються системи освітлення, клімат-контролю та безпеки. Проте, не менш важливим є розвиток технологій у сфері догляду за домашніми тваринами. Цей процес почався не з використанням високих технологій, а зі створенням простих механічних пристроїв, які згодом еволюціонували у складні цифрові екосистеми.

У середині ХХ століття для догляду за домашніми тваринами була потрібна лише ручна праця: регулярне годування, прибирання і прогулянки, що вимагало постійного втручання з боку людини. Перші механічні годівнички з таймерами з'явилися як рішення, що дозволило власникам залишати своїх вихованців на певний час без шкоди їхньому режиму харчування.

З поширенням електроніки в 1980-1990-х роках з'явилися перші електронні пристрої, які забезпечували точніший контроль над процесами догляду. До них відносилися автоматичні поїлки, електронні туалети та системи відкривання дверей для тварин. Ці нововведення суттєво полегшували щоденні обов'язки власників та надавали їм більше свободи.

Справжній прорив відбувся з появою інтернету речей (IoT) та мобільних застосунків. Сучасні системи догляду вже не обмежуються лише механічними пристроями - вони включають в себе камери спостереження, модулі розпізнавання обличчя, голосове управління, GPS-трекери та навіть можливість аналізу поведінки за допомогою штучного інтелекту. Наприклад, "розумні" годівнички можуть автоматично визначати кількість їжі згідно з породою, віком чи рівнем активності тварини.

Особливу увагу привертають інтерактивні системи для розваг, такі як лазерні іграшки та роботизовані пристрої, що імітують взаємодію зі своїм господарем. Власники можуть взаємодіяти зі своїми улюбленцями віддалено або спостерігати за їхньою поведінкою через смартфон. Ці системи також дають можливість

					КВРКІ.210105.21.01.24 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

зафіксувати незвичну активність або симптоми захворювань та передавати цю інформацію ветеринару в реальному часі.

На сьогоднішній день, ринок технологій для догляду за домашніми тваринами швидко розвивається. Численні стартапи та великі компанії інвестують у розробку персоналізованих екосистем для різних видів тварин від собак і котів до птахів і рептилій. З плином часу ці технології стають дедалі доступнішими, надаючи можливість середньостатистичному власнику забезпечити належний догляд своїм тваринам навіть на відстані.

Автоматизація в догляді за тваринами, аналогічно до розвитку побутової техніки, є логічним кроком вперед. Вона представляє собою поєднання турботи, комфорту і сучасних технологій. І хоча ці системи ще не досягли повної автономності, вони вже значною мірою підвищують якість життя як для тварин, так і для їхніх власників.



Рисунок 1.1 – Годівниця з відкидною кришкою (зліва) та розділена на сегменти (справа)[51]

Перші спроби автоматизації були орієнтовані переважно на годування тварин. У 1950-1960-х роках у США з'явилися механічні годівниці з таймерами, які відкривали доступ до їжі в запланований час. Такі пристрої дозволяли власникам

покидати тварин на кілька днів без нагляду, не переймаючись регулярним годуванням. Хоча вони були обмежені у функціональності та громіздки і не враховували індивідуальні потреби тварин, це був важливий крок на шляху до автоматизації.

У 1980-1990-х роках, завдяки прогресу в електроніці, з'явилися електронні годівниці та автопоїлки, що давали змогу встановлювати розклад, обсяг порцій і навіть підключалися до електромережі. В цей період почали з'являтися електронні дверцята, які автоматично відкривалися лише для конкретних тварин (з міткою або RFID-брелком), запобігаючи проникненню сторонніх.

Одним із значних проривів стало впровадження відеоспостереження: спочатку це були окремі камери, що дозволяли власникам стежити за тваринами під час їхньої відсутності. Завдяки розвитку інтернету такі системи стали більш доступними та зручними, з можливістю переглядати відео в реальному часі через мобільний додаток.

На початку 2000-х років ринок активно освоювали стартапи, які пропонували комплексні рішення. Наприклад, системи з керуванням через смартфон, інтеграцією з Wi-Fi, голосовими командами та навіть вбудованим штучним інтелектом для аналізу поведінки тварини.



Рисунок 1.2 – Одна з перших автоматичних годівниць для домашніх тварин[52]

					КвРКІ.210105.21.01.24 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

Основні завдання, які стоять перед сучасними автоматизованими системами догляду за тваринами:

- розробка систем моніторингу. Наприклад використання камер, сенсорів руху та температури для постійного контролю за поведінкою і умовами утримання тварин;
- автоматизація годівлі та водопостачання, а саме забезпечення точного дозування корму та можливість дистанційного управління процесом;
- взаємодія з мобільними додатками щоб власник міг управляти системою з будь-якого місця у світі і отримувати повідомлення;
- покращення гігієнічних умов, автоматизоване прибирання та обробка місць утримання тварин;
- аналіз стану тварини такий як застосування ІТ-технологій для спостереження змін у поведінці або здоров'ї тварини.

## 1.2 Актуальність застосування систем догляду за домашніми тваринами

Актуальність використання систем догляду за домашніми тваринами зумовлена сучасними викликами та тенденціями, що охоплюють як соціальні, так і технологічні аспекти. У ХХІ столітті дедалі більше людей сприймають домашніх тварин не просто як улюбленців, а як важливих членів родини. Вони розглядаються як живі істоти, що мають право на комфорт, належний догляд, емоційний зв'язок і стабільне середовище. Це підштовхує до пошуку більш системного, надійного та зручного підходу для забезпечення їхнього благополуччя.

Швидкий темп життя сучасної людини також впливає на значущість автоматизованих рішень. Постійна зайнятість, подорожі та гнучкі графіки змушують власників шукати способи догляду за своїми тваринами навіть за їхньої фізичної відсутності. Сучасні технології дозволяють підтримувати належні умови утримання та запобігати ризикам, таким як перегрівання, нестача води чи корму, зниження вологості або ж тривожна поведінка.

					КвРКІ.210105.21.01.24 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Технічний прогрес, включно з розвитком мікрокомп'ютерів, інтернету речей (IoT), бездротових мереж, сенсорних технологій і систем штучного інтелекту, створює сприятливі умови для інтеграції догляду за тваринами у загальну цифрову інфраструктуру помешкання. Можна реалізувати комплексні системи для моніторингу, управління та аналітики, доступні як для індивідуального, так і масового споживання.

Такі системи здатні адаптуватися до змін умов: вивчати поведінку тварини, регулювати режим годування, вмикати світло у певний час або створювати звукове середовище для зниження стресу під час відсутності господаря. У перспективі вони можуть виявляти ранні ознаки захворювань або змін у поведінці завдяки інтеграції з комп'ютерним зором та аналізом даних.

Крім практичних переваг, впровадження таких систем має і соціальну важливість. Вони сприяють відповідальному ставленню до утримання тварин, допомагають запобігати нехтуванню їхніми потребами та знижують ризики травмування або загибелі улюбленців. У умовах урбанізації, коли тварини часто живуть у квартирах без постійного доступу до зовнішнього простору, контроль за мікрокліматом і поведінкою стає необхідним.

Отже, актуальність систем догляду за домашніми тваринами стосується не лише спрощення догляду, а й поліпшення якості життя тварин, оптимізації витрат часу для власників та зниження ризиків. З огляду на розвиток технологій і зростання етичної відповідальності за благополуччя тварин, такі системи мають як практичне, так і суспільне значення.

Швидке зростання ринку зоотоварів свідчить про стійку тенденцію до інвестування в комфорт і безпеку домашніх улюбленців. Все більше власників надають перевагу рішенням, що дозволяють автоматизувати рутинні процеси. Це не лише спрощує догляд, але й зменшує ризик людських помилок. Особливо популярними стають доступні за ціною, але водночас функціональні системи, які становлять достойну альтернативу дорогим комерційним аналогам.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

Іншим значущим чинником є можливість персоналізації догляду. Автоматизовані системи можуть бути адаптовані під індивідуальні потреби конкретної тварини, з урахуванням її віку, стану здоров'я, рівня активності, а також звичок. Це сприяє досягненню вищого рівня точності у годівлі, контролі мікроклімату чи моніторингу поведінки. Персоналізовані алгоритми не тільки покращують умови життя тварин, але й допомагають уникати стресових ситуацій, які можуть негативно позначитися на їхньому самопочутті.

### 1.3 Порівняльний аналіз переваг та недоліків існуючих рішень

Сьогодні існує багато готових рішень для автоматизованого догляду за домашніми тваринами. Найпоширенішими з них є автоматичні годівниці, камери відеоспостереження зі зворотним зв'язком та модулі для систем «розумного дому», які можуть частково або повністю піклуватися про тварин.

Автоматичні годівниці від таких виробників, як PetSafe або Xiaomi, відрізняються простотою використання і можливістю програмувати час подачі корму. Проте вони зазвичай мають обмежений функціонал і не підтримують інтеграцію з іншими системами, а також не дозволяють контролювати стан тварини чи середовище.

Інші популярні рішення включають камери відеоспостереження для домашніх улюбленців, такі як Petcube або Furbo. Вони дозволяють власникам в режимі реального часу спостерігати за своїм улюбленцем, спілкуватися з ним через мікрофон і іноді навіть подавати ласощі. Основні недоліки цих систем - висока вартість, обмежений функціонал і потреба в постійному інтернет-з'єднанні.

Більш складними є інтегровані системи «розумного дому», які включають модулі для керування мікрокліматом, освітленням та безпекою. Ці рішення гнучкі та масштабовані, але зазвичай потребують більш глибоких технічних знань для налаштування та суттєвих фінансових вкладень в обладнання.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

Окрему нішу займають авторські проекти, створені на основі мікрокомп'ютерів, таких як Raspberry Pi або Arduino. Основною перевагою є можливість повної кастомізації функцій відповідно до індивідуальних потреб користувача. Окрім того, вони є порівняно доступними та дозволяють краще зрозуміти принципи роботи кіберфізичних систем. Проте їх створення вимагає спеціальних знань у електроніці, програмуванні та налаштуванні систем.

Зважаючи на це, розробка системи догляду за тваринами із використанням Raspberry Pi 4 дозволяє збалансувати доступність, функціональність і гнучкість. Такий підхід дозволяє створити індивідуальну інтелектуальну систему, адаптовану до конкретних вимог.

Дана система належить до класу кіберфізичних систем, що інтегрують обчислювальні елементи з фізичними процесами задля забезпечення моніторингу, контролю та автоматизації в режимі реального часу.

З технічної точки зору, розроблена система складається із декількох інтегрованих підсистем, кожна з яких має специфічну функціональність. Сенсорна підсистема здійснює збір даних про параметри навколишнього середовища, включаючи температуру, рівень вологості та наявність руху, що дає змогу системі постійно відстежувати зміни в мікрокліматі та поведінці тварин. Виконавча підсистема виконує відповідні дії у фізичному середовищі, зокрема контроль автоматичної годівниці, управління освітленням та активацію вентиляційних систем за потреби. Центральна програмна логіка здійснює обробку отриманих сенсорних даних, аналізує поточну ситуацію і приймає рішення щодо необхідних заходів. Для взаємодії з користувачем передбачено інтуїтивно зрозумілий інтерфейс, через який можна здійснювати моніторинг параметрів системи, переглядати відеопотік та дистанційно керувати основними функціями. Така структура забезпечує ефективну, надійну та адаптивну роботу системи в режимі реального часу.

Цей об'єкт дослідження є домашньою мікрокіберфізичною системою, що виконує інформаційно-керуючі функції в побутових умовах. Основна мета -

автоматизований догляд за домашніми тваринами, що включає регулярне годування, контроль мікроклімату та спостереження, а також повідомлення власника про нестандартні ситуації.

Характер використання системи вказує на її належність до індивідуальних споживчих технологічних рішень. Вона створюється не для масового продажу, а для задоволення конкретних потреб користувача чи як прототип для подальшого удосконалення.

Класифікується система як автономна із можливістю дистанційного управління, оскільки вона здатна приймати рішення на основі сенсорних даних та забезпечує взаємодію через мережевий інтерфейс, такий як мобільний додаток чи веб-панель управління.

Структурно система є модульною, що дозволяє підключати додаткові компоненти, такі як камери, нові датчики чи сервоприводи, та розширювати функціональні можливості відповідно до потреб користувача. Це відповідає сучасним принципам проектування «розумних» систем, які мають бути гнучкими, адаптивними та зручними у використанні.

У науково-технічному контексті система об'єднує знання з інформаційних технологій, схемотехніки, автоматизації, програмування і основ догляду за тваринами. Таким чином, її можна визначити як міждисциплінарний інтелектуальний пристрій спеціалізованого призначення.

### 1.3.1 Існуючі варіанти систем догляду за домашніми тваринами

Розглянемо автоматичні годівниці як один із ключових елементів системи догляду за домашніми тваринами.

Годівниця PetSafe Smart Feed дозволяє власникам тварин дистанційно керувати процесом годування через мобільний застосунок My PetSafe. Вона має підтримку Wi-Fi, можливість налаштування до 12 прийомів їжі на день, функцію

					КВРКІ.210105.21.01.24 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

"Slow Feed" для запобігання переїданню та сенсори, які сповіщають про низький рівень корму.

Інтелектуальна система Xiaomi Petoneer Nutri працює через хмару, дає змогу точно налаштувати час і кількість порцій, має сенсори для контролю рівня корму та підтримує функцію запису голосових повідомлень, які можна відтворювати тваринам.

Бюджетним варіантом є WOPET Automatic Pet Feeder. Цей пристрій дозволяє програмувати розклад годування на кілька днів або тижнів, вирізняється простим інтерфейсом і надійною конструкцією, завдяки чому здобула популярність серед власників тварин.

Системи відеоспостереження також є невід'ємною частиною сучасної автоматизованої системи. Пристрої Petcube Bites і Petcube Play від українського стартапу мають HD-камеру з кутом огляду до 180°, підтримують двосторонній аудіозв'язок, функцію видачі ласощів, Wi-Fi на частоті 5 ГГц та інтеграцію з Alexa.

Ще одне вражаюче рішення Furbo Dog Camera, яке також забезпечує відеоспостереження у форматі Full HD 1080p. Камера має кут огляду 160°, функцію нічного бачення, а також інтелектуальне сповіщення про гавкіт, що може стати в пригоді, коли тварина збуджено реагує на сторонні шуми чи незнайомих. Крім цього, Furbo підтримує двосторонній аудіозв'язок і дозволяє дистанційно видавати ласощі, допомагаючи власникам залишатися на зв'язку зі своїм собакою навіть на відстані.

Системи контролю мікроклімату та активності тварини також є важливою складовою комплексного підходу. Компактний пристрій, який кріпиться нашийник собаки та виконує роль фітнес-трекера має назву Sure Petcare Animo. Він відстежує фізичну активність, якість сну, частоту гавкання, дряпання й загальний рівень стресу тварини. Завдяки синхронізації з мобільним додатком через Bluetooth, власник отримує детальну аналітику про стан улюбленця. Корпус Animo є водонепроникним, тому пристрій можна безпечно використовувати під час прогулянок за будь-якої погоди.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

У той же час Pawbo+ це багатофункціональний пристрій для інтерактивного спілкування з домашніми тваринами, який об'єднує HD-камеру для відеоспостереження, вбудований лазер для ігор і двосторонній аудіозв'язок. Завдяки цьому власники можуть не лише спостерігати за поведінкою своїх улюбленців, але й активно взаємодіяти з ними, розважаючи їх або заспокоюючи голосом. Крім того, пристрій має функцію видачі ласощів, що додає ще один рівень взаємодії на відстані.

Комплексні рішення для догляду та моніторингу поєднують у собі всі основні функції. Платформа Home Assistant на базі Raspberry Pi відкриває широкі можливості для комплексної автоматизації догляду за тваринами. Вона дозволяє об'єднати в одну систему автоматичні годівниці, терморегулятори, системи вентиляції, освітлення, камери спостереження тощо. Такий підхід забезпечує централізоване управління та моніторинг усього середовища, в якому перебуває тварина. Ви можете віддалено контролювати умови утримання, налаштовувати розклади, отримувати сповіщення про зміну параметрів та швидко реагувати на потреби вашого улюбленця.

Ще одим технологічним рішенням є iKuddle Auto-Pack Litter Box. Це автоматизований туалет для котів, який виконує функції самоочищення, герметичної упаковки відходів і дезодорування повітря. Це суттєво знижує рівень запаху і підвищує гігієну в домі. Користувачі можуть відстежувати стан пристрою через мобільний додаток, який надсилає сповіщення про необхідність обслуговування, що робить догляд за котом максимально зручним і сучасним.

### 1.3.2 Особливості систем

Системи догляду за домашніми тваринами являють собою комплекс технологічних рішень, спрямованих на оптимізацію процесу догляду за улюбленцями, забезпечуючи їх комфорт мінімальною участю власника. Однією з важливих характеристик таких систем є автономність, котра дозволяє автоматизувати базові задачі догляду за твариною, включаючи регулярне

					КВРКІ.210105.21.01.24 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

годування, підтримання належних умов середовища, а також організацію моніторингу через камери спостереження або датчики.

Дистанційне управління є ще однією вагомою властивістю. Завдяки підключенню до Інтернету системи здатні інтегруватися з мобільними додатками або веб-інтерфейсами, забезпечуючи власнику змогу в будь-який час перевіряти стан тварини, управляти окремими функціями пристроїв або отримувати сповіщення про зміни в умовах середовища. Це особливо актуально для осіб, які часто відсутні вдома, але прагнуть бути впевненими в безпеці своїх улюбленців.

Сучасні системи часто включають інтелектуальні функції, такі як аналіз поведінки тварини або адаптація режиму годування відповідно до її активності. Зазначені можливості реалізуються завдяки застосуванню основних алгоритмів обробки даних та елементарних моделей машинного навчання, що забезпечує не лише автоматизований, але й адаптивний догляд згідно з потребами конкретної тварини.

Особлива увага приділяється безпеці систем як для тварини, так і для житлового простору загалом. Технічні рішення мають бути надійними і стійкими до збоїв живлення чи перевантаження, забезпечуючи можливість переходу в аварійний режим. Наприклад, при втраті з'єднання з мережею або розрядженні батареї повинна зберігатися можливість продовження базової роботи пристроїв.

Гнучкість і масштабованість систем становлять ще одну характерну рису. В залежності від виду тварини, її віку, поведінкових особливостей та побутових умов власник може самостійно встановлювати інтервали годування, критичні значення температури та вологості, типи сповіщень і інші параметри. Крім того, системи часто проектуються модульно, що дозволяє поступово доповнювати їх новими датчиками, камерами чи функціями без зміни всієї конструкції.

Зручність користування є не менш важливою особливістю. Оскільки потенційними користувачами таких систем виступають звичайні власники тварин, а не лише технічно підковані особи, інтерфейс має бути інтуїтивно зрозумілим, а інсталяція обладнання доступною без складних налаштувань. Документація,

					КВРКІ.210105.21.01.24 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

інструкції, візуальні підказки та логічно організовані елементи управління значно підвищують комфорт користування.

Внаслідок цього сучасні системи догляду за домашніми тваринами поєднують автоматизацію, інтелектуальність, надійність, дистанційне управління та гнучкість налаштувань. Це створює передумови для ефективного, безпечного та зручного догляду за тваринами, особливо в умовах насиченого способу життя власників.

#### 1.4 Raspberry Pi 4

Raspberry Pi 4 – це високоефективна мікрокомп'ютерна плата четвертого покоління, розроблена некомерційною організацією Raspberry Pi Foundation у Великій Британії. Її призначено для освітніх цілей, розробки проєктів у сферах автоматизації, робототехніки, Інтернету речей (IoT) та створення прототипів кіберфізичних систем.

Плата вирізняється компактністю (85.6×56.5 мм) і має широкий набір функцій, що робить її привабливим вибором для створення інтелектуальних систем у домашньому середовищі.

Основні технічні характеристики Raspberry Pi 4:

- процесор: чотирьохядерний ARM Cortex-A72 (64-бітний, 1.5 ГГц);
- оперативна пам'ять: 2 ГБ, 4 ГБ або 8 ГБ LPDDR4 (залежно від моделі);
- графіка: VideoCore VI, підтримка відео з роздільністю 4K;
- порти: 2×USB 3.0, 2×USB 2.0, 2×Micro-HDMI (до 4K), Ethernet, роз'єм GPIO (40 контактів);
- інтерфейси: Wi-Fi 802.11ac, Bluetooth 5.0, microSD для зберігання даних;
- живлення: через USB-C (5В, 3А).

Важливою особливістю Raspberry Pi 4 є можливість підключення різноманітних сенсорів і модулів (камери, дисплеї, датчики температури, вологості,

					КВРКІ.210105.21.01.24 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

руху тощо). Це дозволяє використовувати плату як основний керуючий елемент у розумних системах.

У контексті розробки системи догляду за домашніми тваринами, мікрокомп'ютер Raspberry Pi 4 виконує функцію головної платформи, яка забезпечує централізоване управління всіма функціональними компонентами системи. Цей пристрій здатний зчитувати дані з підключених датчиків, таких як датчики температури, вологості й руху, що дозволяє здійснювати постійний моніторинг умов навколишнього середовища, в якому перебуває домашня тварина. Зібрані дані обробляються в реальному часі на Raspberry Pi, що дає змогу своєчасно реагувати на зміну параметрів та автоматично активувати чи деактивувати відповідні виконавчі модулі.

До таких модулів належить автоматична годівниця, яка працює відповідно до заданого графіка або залежно від поведінки тварини. Освітлення також підлягає коригуванню з урахуванням часу доби та присутності руху в зоні спостереження. Крім того, Raspberry Pi виконує функцію організації відеоспостереження: підключена камера транслює зображення в реальному часі, що дозволяє власнику проводити дистанційне моніторування за твариною.

Одним із важливих аспектів системи є віддалений доступ. Завдяки веб-інтерфейсу, користувач може переглядати дані з датчиків, отримувати повідомлення про події, змінювати налаштування або керувати окремими компонентами системи з будь-якої локації за умов наявності Інтернет-з'єднання. Таким чином, Raspberry Pi 4 інтегрує функції збору, обробки та передавання даних разом із централізованим управлінням пристроями, роблячи його оптимальною платформою для створення інтелектуальної системи догляду за домашніми тваринами.

Серед переваг Raspberry Pi 4 – доступність, велика підтримка бібліотек для Python, активна спільнота користувачів і гнучкість у налаштуванні системи відповідно до потреб проекту.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

## 1.5 Методи розв'язання проблеми в рамках досліджуваної теми

Розглянемо можливі методи розв'язання проблеми у досліджуваній роботі.

Система апаратного підходу базується на мікрокомп'ютері Raspberry Pi 4, що слугує її центральним обчислювальним модулем. Для автоматичного збору даних про середовище та активність тварини будуть використовуватися датчики температури та вологості (наприклад, DHT22 чи BME280), датчики руху (PIR-сенсори), камери спостереження (модуль камери Raspberry Pi) та сервоприводи для механізмів подачі корму й води. У рамках програмного підходу, реле, які контролюють освітлення або вентиляцію, будуть інтегровані через відповідні інтерфейси, такі як GPIO порти Raspberry Pi.

Системне програмне забезпечення є ключовим компонентом інтегрованої системи автоматизованого догляду за домашніми тваринами, адже воно відповідає за безперебійну роботу всієї інфраструктури. Його головне призначення це забезпечення збору даних із сенсорів у режимі реального часу, їх оперативна обробка та виконання відповідних дій згідно з наперед визначеними сценаріями. Наприклад, система може автоматично активувати вентиляцію при перевищенні комфортного температурного порогу або подати сигнал тривоги у разі тривалої відсутності активності тварини.

Для реалізації логіки взаємодії з сенсорами, виконавчими механізмами (такими як моторизовані клапани, реле, годівниці) та для реалізації комунікаційних модулів, планується використання мови програмування Python. Python має широкую підтримку бібліотек для роботи з Raspberry Pi, GPIO-портами та протоколами зв'язку, такими як I2C, SPI та MQTT.

Ще одним важливим аспектом є користувацький інтерфейс. Він буде реалізований у вигляді веб-інтерфейсу або мобільного додатку, що дозволить зручно моніторити всі показники системи, керувати пристроями, задавати сценарії та налаштовувати сповіщення. Через цей інтерфейс користувач отримає доступ до

					КВРКІ.210105.21.01.24 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

дашборду з такими даними, як температура, вологість, активність тварини, статус годівлі, тощо.

Комунікаційна складова передбачає також реалізацію механізму сповіщень електронною поштою або через push-повідомлення. Це дозволить своєчасно інформувати власника про критичні ситуації, наприклад, про перегрів приміщення, вихід з ладу сенсора або зміну звичного режиму поведінки тварини.

Щодо інженерно-системного підходу розробка передбачає формування модульної архітектури, яка в свою чергу дозволяє безперешкодно додавати або замінювати сенсори, а також здійснювати масштабування системи для використання в кількох приміщеннях або для управління кількома тваринами.

Особливий акцент ставиться на оптимізацію енергоспоживання та забезпечення надійності функціонування системи в автономному режимі.

## 1.6 Постановка задачі

Основна мета даного дослідження полягає в розробці моделі автоматизованої системи догляду за домашніми тваринами, яка значно спростить щоденне доглядання за ними. Запропонована система забезпечує постійний моніторинг параметрів мікроклімату в приміщенні, таких як температура та вологість, що дозволяє вчасно виявляти відхилення від комфортних умов. Ключовою функцією є автоматизоване годування тварин відповідно до заданого розкладу, забезпечуючи регулярність та контроль споживання корму. Також система має засоби відеоспостереження, що дають власнику можливість спостерігати за поведінкою тварини в реальному часі та дистанційно управляти системою через зручний інтерфейс. У випадку відхилень у показниках, таких як зміни температури, підвищення вологості чи нестача корму, власник отримує автоматичні сповіщення для оперативного реагування. Усі ці функції комплексно спрямовані на підвищення комфорту тварин та зниження навантаження на власника.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Для досягнення поставленої мети створення інтегрованої системи автоматизованого догляду за домашніми тваринами на базі Raspberry Pi 4 слід виконати низку ключових завдань. Першим кроком є проведення всебічного аналізу наявних рішень у цій сфері. Це дозволить виявити актуальні тенденції, функціональні можливості, а також сильні та слабкі сторони існуючих систем, що стане базою для формування конкурентної концепції.

Наступним важливим етапом є технічне обґрунтування вибору апаратної платформи. Raspberry Pi 4 вирізняється компактністю, енергоефективністю, широкими можливостями підключення зовнішніх пристроїв, наявністю підтримки Wi-Fi і Bluetooth, що робить його ідеальним рішенням для реалізації системи домашньої автоматизації.

Розробка структурної та функціональної моделей дозволить візуалізувати архітектуру системи, визначити взаємозв'язки між модулями, функціональні блоки, логіку обробки даних і способи взаємодії з користувачем. Особливу увагу слід приділити вибору сенсорів (наприклад, температури, вологості, руху, звуку) та виконавчих елементів (моторів для годівниць, систем керування освітленням чи вентиляцією), які забезпечать моніторинг і керування навколишнім середовищем.

Необхідно також створити програмне забезпечення для інтеграції всіх апаратних компонентів. Воно має забезпечувати обробку отриманих даних, підтримку інтерфейсу користувача, функції віддаленого доступу та налаштувань. Це ПЗ може бути побудоване з використанням платформи Home Assistant або створене з нуля, залежно від вимог до гнучкості та масштабованості.

Завершальним етапом є тестування розробленої системи в умовах, максимально наближених до реального середовища проживання тварин. Це дозволить перевірити працездатність, стабільність, точність моніторингу та ефективність автоматизації. На основі отриманих результатів визначаються можливі шляхи вдосконалення як апаратної, так і програмної частин.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

Реалізація всіх цих етапів приведе до створення сучасної інтегрованої системи, здатної значно підвищити комфорт і безпеку домашніх тварин, автоматизуючи ключові процеси догляду.

### 1.7 Висновки до першого розділу

Проведений аналіз предметної області показав, що автоматизовані системи догляду за домашніми тваринами є актуальним рішенням через зростання кількості власників тварин і ускладнення їхнього догляду в умовах сучасного ритму життя. Основними проблемами є висока вартість комерційних систем, обмежена функціональність і складність адаптації до специфічних потреб користувачів.

Порівняльний аналіз існуючих рішень виявив переваги саморобних систем на базі Raspberry Pi, зокрема їхню економічність і гнучкість, хоча вони потребують технічних знань для реалізації. Особливості систем включають використання IoT-технологій, модульність апаратного забезпечення та підтримку віддаленого керування, що є ключовими для ефективного догляду.

Методи розв'язання проблеми, такі як аналітичний, системний, програмний та експериментальний підходи, дозволяють створити модульну систему, яка відповідає сучасним вимогам.

Постановка задачі передбачає розробку прототипу системи, що включає автоматичне годування, відеоспостереження та віддалене керування, з орієнтацією на низьку вартість і простоту використання. Таким чином, теоретичні основи підтверджують доцільність створення системи догляду за домашніми тваринами на базі Raspberry Pi 4, яка поєднує доступність, функціональність і адаптивність.

					КвРКІ.210105.21.01.24 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 КОМПОНЕНТИ ПРОЕКТУ ТА СЕРЕДОВИЩЕ РОЗРОБЛЕННЯ

### 2.1 Огляд компонентів проекту

Розробка системи автоматизованого догляду за домашніми тваринами на базі мікрокомп'ютера Raspberry Pi 4 передбачає використання комплексу апаратних і програмних компонентів, які разом забезпечують виконання ключових функцій: автоматичне годування, відеоспостереження, виявлення присутності тварини та віддалене керування через мережу Інтернет. У цьому розділі було детально розглянуто огляд компонентів проекту, їхніх функцій, принципів взаємодії та ролі в реалізації системи. Вибір компонентів ґрунтується на критеріях економічної вигоди, доступності в Україні, сумісності з Raspberry Pi 4 і модульності, що дозволяє адаптувати систему до різних сценаріїв використання.

Апаратне забезпечення системи є основою для фізичної реалізації функціоналу. Воно включає мікрокомп'ютер, периферійні пристрої, датчики та модулі, які забезпечують взаємодію з навколишнім середовищем і виконання механічних завдань. Кожен компонент обрано з урахуванням його технічних характеристик, вартості та доступності через локальних дистриб'юторів, таких як EVO чи Raspberry Pi Україна.

Raspberry Pi 4 є центральним елементом системи, виконуючи роль обчислювального ядра. Цей мікрокомп'ютер оснащений чотириядерним процесором Broadcom BCM2711 (1.5 ГГц), 4 ГБ оперативної пам'яті, 40 GPIO-пінами, вбудованим Wi-Fi (2.4/5 ГГц) і Bluetooth 5.0. Такі характеристики дозволяють обробляти дані з датчиків, керувати периферійними пристроями, підтримувати потокове відео та забезпечувати віддалене керування через мережу. GPIO-піни використовуються для підключення датчиків і сервомоторів, а USB-порти (2×USB 3.0, 2×USB 2.0) -для зовнішніх пристроїв, таких як накопичувачі для зберігання відеозаписів. Вартість Raspberry Pi 4 становить близько 50 дол. США, що робить його економічно вигідним вибором порівняно з іншими платформами, наприклад, Arduino чи спеціалізованими IoT-контролерами.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

Камера є ключовим компонентом для реалізації функції відеоспостереження. Raspberry Pi Camera Module 3 має роздільну здатність 12 МП, підтримує нічний режим і підключається до мікрокомп'ютера через CSI-роз'єм, що забезпечує швидку передачу відеоданих. Вона дозволяє здійснювати потокову передачу відео в реальному часі через вебінтерфейс, а також записувати відео для аналізу поведінки тварин, наприклад, під час годування чи активності в зоні спостереження. Камера сумісна з бібліотекою `picamera2`, яка спрощує її інтеграцію в програмне забезпечення системи. Вартість модуля становить приблизно 25 дол. США, що є доступним для проєктів такого типу.

Сервомотор SG90 використовується для керування диспенсером їжі в автоматичній годівниці. Цей компактний і легкий мотор забезпечує точне позиціонування, що дозволяє дозувати їжу відповідно до заданого розкладу або за командою користувача. Сервомотор підключається до GPIO-пінів Raspberry Pi і керується через бібліотеку `RPi.GPIO`, яка дозволяє задавати кут повороту для видачі потрібної порції їжі. Його низька вартість (близько 5 дол. США) і простота інтеграції роблять SG90 оптимальним вибором для автоматизації механічних процесів у системі.

Пасивний інфрачервоний датчик руху HC-SR501 виявляє присутність тварини в зоні годівниці або спостереження. Він працює за принципом фіксації змін інфрачервоного випромінювання, що дозволяє активувати функції системи, такі як запуск сервомотора для видачі їжі, початок відеозапису чи надсилання сповіщення користувачу. Датчик підключається до GPIO-пінів і має високу чутливість, що забезпечує надійне виявлення тварин, таких як коти чи собаки. Його ціна становить близько 3 дол. США, а енергоефективність дозволяє використовувати його без значного навантаження на систему живлення.

Для стабільної роботи Raspberry Pi 4 і підключених компонентів використовується блок живлення USB-C із параметрами 5В і 3А. Цей компонент забезпечує достатню потужність для одночасної роботи мікрокомп'ютера, камери, сервомотора і датчика, запобігаючи перепадам напруги, які можуть спричинити

					КВРКІ.210105.21.01.24 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

збої. Вартість блоку живлення становить приблизно 10 дол. США, а його сумісність із стандартами Raspberry Pi гарантує надійність системи.

Для з'єднання апаратних елементів використовуються макетні плати, дроти, резистори та перехідники, які забезпечують надійне підключення датчиків і сервомотора до GPIO-пінів. Ці компоненти є недорогими (загальна вартість близько 5-7 дол. США) і доступними через локальних постачальників. Крім того, для зберігання операційної системи та даних системи використовується карта пам'яті microSD (32 ГБ), яка коштує близько 8 дол. США і забезпечує достатній обсяг для Raspberry Pi OS і відеозаписів.

Програмне забезпечення системи відповідає за логіку роботи, обробку даних із апаратних компонентів і надання користувачу інтерфейсу для керування. Воно розробляється в середовищі Raspberry Pi OS із використанням мови Python, яка є стандартом для проєктів на базі Raspberry Pi завдяки своїй простоті та широкій підтримці бібліотек.

Raspberry Pi OS (64-бітна версія) є основною операційною системою, яка забезпечує стабільне середовище для виконання Python-скриптів, налаштування апаратних інтерфейсів (GPIO, CSI) і підтримки мережевих функцій. ОС включає попередньо встановлені інструменти і підтримує бібліотеки для роботи з датчиками та камерою. Raspberry Pi OS є безкоштовною, що знижує витрати на розробку, і регулярно оновлюється, забезпечуючи актуальність і безпеку.

Логіка системи реалізується через набір Python-скриптів, які керують апаратними компонентами та забезпечують автоматизацію. Наприклад, скрипти обробляють сигнали від датчика руху HC-SR501 для активації сервомотора SG90, запускають відеозапис через камеру та надсилають сповіщення користувачу.

Задля функціонування системи було використано такі бібліотеки:

- RPi.GPIO – для керування GPIO-пінами, що дозволяє взаємодіяти з сервомотором і датчиком руху;
- picamera2 – для роботи з камерою Raspberry Pi, забезпечуючи запис і потокову передачу відео;

					КВРКІ.210105.21.01.24 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

– paho-mqtt – для реалізації MQTT-протоколу, який забезпечує обмін даними з хмарними сервісами для віддаленого керування.

Для віддаленого керування системою розроблено вебінтерфейс на основі фреймворку Flask, який дозволяє користувачам через браузер переглядати потокове відео, налаштовувати розклад годування та отримувати сповіщення про події, наприклад, виявлення тварини датчиком руху. Вебінтерфейс розміщується на локальному сервері Raspberry Pi і доступний через Wi-Fi або зовнішню мережу за допомогою протоколу HTTP. Це забезпечує зручність використання для користувачів без технічних знань, що є перевагою порівняно з комерційними системами, такими як Petnet SmartFeeder.

Для інтеграції з хмарними сервісами та реалізації віддаленого керування використовується MQTT-протокол, який забезпечує швидкий і надійний обмін повідомленнями між Raspberry Pi та зовнішніми пристроями (наприклад, смартфоном користувача). MQTT-клієнт, реалізований через бібліотеку paho-mqtt, дозволяє надсилати команди, такі як активація годівниці, і отримувати дані, наприклад, зображення чи логи подій. Цей компонент підвищує гнучкість системи, дозволяючи інтегрувати її з популярними платформами, такими як Home Assistant.

Апаратні та perspective, що забезпечує чітку інтеграцію всіх компонентів системи. Наприклад, датчик руху (HC-SR501) виявляє присутність тварини, сигнал передається на Raspberry Pi через GPIO-пін. Python-скрипт обробляє цей сигнал, активує сервомотор (SG90) для видачі їжі та запускає запис відео через камеру (Camera Module 3). Дані про подію (час, зображення) надсилаються через Wi-Fi на вебінтерфейс (Flask) або хмарний сервіс (MQTT), де користувач може переглянути їх через браузер чи додаток. Така архітектура забезпечує модульність, дозволяючи додавати нові компоненти, наприклад, датчик температури для моніторингу умов утримання тварини, без значних змін у системі.

Загальна вартість апаратних компонентів системи, включаючи Raspberry Pi 4 не перевищує 100 дол. США. Це значно дешевше порівняно з комерційними

					КВРКІ.210105.21.01.24 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

аналогами, такими як Petnet SmartFeeder (150-300 дол. США), що робить систему економічно вигідною.

Обрані апаратні та програмні компоненти системи автоматизованого догляду мають низку суттєвих переваг, що роблять її не лише ефективною, але й гнучкою до модернізації та розширення. Однією з головних є модульність: завдяки GPIO-пінам на платі Raspberry Pi, користувач може без зайвих складнощів додавати нові сенсори датчики вологості, освітленості або мікрофони для звукового моніторингу. Це відкриває перспективу створення адаптивної системи, яка зростає разом із потребами тварини чи умовами середовища.

Ще одна сильна сторона гнучкість програмного забезпечення, яку забезпечує Python. Завдяки широкому вибору бібліотек та підтримці API, розробник може змінювати поведінку системи без переписування архітектури. Наприклад, оновити логіку видачі корму або інтегрувати новий тип сповіщень, зокрема push або SMS.

Значущою перевагою є також відкрита екосистема Raspberry Pi. Спільнота розробників по всьому світу активно ділиться знаннями, надає приклади коду, технічні рішення, публікації на платформах Raspberry Pi Forums, Stack Overflow, GitHub, що суттєво пришвидшує процес розробки і спрощує вирішення навіть складних задач.

Крім того, вибрані компоненти сприяють енергоефективності системи. Датчики руху або сервомотори споживають мінімум енергії під час очікування або короткочасного ввімкнення, а живлення через стабільний USB-C блок живлення підтримує роботу без перебоїв, що особливо важливо при тривалому використанні вдома.

Незважаючи на переваги, компоненти мають певні обмеження. Raspberry Pi 4 потребує стабільного живлення, і недостатня потужність блоку живлення може спричинити збої при одночасній роботі кількох пристроїв. Крім того, створення системи потребує базових знань програмування та електроніки для правильного підключення компонентів і налаштування програмного забезпечення, що може бути викликом для користувачів без технічної підготовки. Однак ці обмеження

					КВРКІ.210105.21.01.24 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

компенсуються доступністю документації та підтримкою спільноти Raspberry Pi, що полегшує розробку.

Огляд компонентів проекту демонструє, що система догляду за домашніми тваринами на базі Raspberry Pi 4 складається з ретельно підібраних апаратних і програмних елементів, які забезпечують автоматизацію ключових функцій за низької вартості. Апаратні компоненти, такі як Raspberry Pi 4, камера, сервомотор і датчик руху, є доступними, сумісними та модульними, що дозволяє адаптувати систему до різних потреб. Програмне забезпечення, базоване на Python і Raspberry Pi OS, забезпечує гнучкість і простоту розробки, а інтеграція з IoT-технологіями, такими як MQTT, гарантує віддалене керування.

## 2.2 Вибір апаратних складових системи

Вибір апаратних складових для системи автоматизованого догляду за домашніми тваринами на базі Raspberry Pi 4 є критично важливим етапом, який визначає функціональність, надійність і економічну ефективність системи. Апаратне забезпечення має забезпечувати виконання ключових функцій: автоматичне годування, відеоспостереження, виявлення присутності тварини та віддалене керування через мережу Інтернет.

Центральним елементом системи є Raspberry Pi 4 Model B із 4 ГБ оперативної пам'яті, який забезпечує обчислювальну потужність для обробки даних, керування периферійними пристроями та підтримки мережевих функцій. Цей мікрокомп'ютер оснащений чотириядерним процесором Broadcom BCM2711 (1.5 ГГц), що дозволяє виконувати кілька завдань одночасно, наприклад, обробку відеопотоку з камери, виконання Python-скриптів для автоматизації годування та підтримку вебінтерфейсу для віддаленого доступу. Наявність 40 GPIO-пінів забезпечує підключення датчиків, сервомоторів і інших модулів, що є необхідним для реалізації системи догляду.

Raspberry Pi 4 підтримує Wi-Fi (2.4/5 ГГц) і Bluetooth 5.0, що дозволяє організувати віддалене керування через мережу Інтернет за допомогою протоколів

					КВРКІ.210105.21.01.24 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

MQTT або HTTP. Два порти USB 3.0 і два мікро-HDMI-порти забезпечують підключення зовнішніх пристроїв, таких як накопичувачі для зберігання відеозаписів, і виведення відео високої роздільної здатності для локального моніторингу. Вартість моделі з 4 ГБ пам'яті становить близько 50 дол. США, що є економічно вигідним порівняно з іншими платформами, такими як Arduino, які мають меншу продуктивність і обмежені мережеві можливості.

Для реалізації функції відеоспостереження обрано Raspberry Pi Camera Module 3, яка має роздільну здатність 12 МП і підтримує нічний режим, що дозволяє фіксувати поведінку тварин у будь-який час доби. Камера підключається до Raspberry Pi 4 через спеціальний CSI-роз'єм, що забезпечує швидку передачу відеоданих із мінімальною затримкою. Вона сумісна з бібліотекою picamera2, яка дозволяє реалізувати потокову передачу відео через вебінтерфейс і запис для подальшого аналізу, наприклад, для перевірки, чи тварина підходила до годівниці.

Камера є критично важливою для моніторингу в реальному часі, що допомагає власникам контролювати стан тварини та її активність. Її компактні розміри та низьке енергоспоживання роблять її ідеальним вибором для системи, а вартість відповідає критерію економічної вигоди. Альтернативи, такі як USB-камери, менш сумісні з Raspberry Pi і потребують додаткових драйверів, що ускладнює інтеграцію. Доступність камери через локальних постачальників забезпечує швидке придбання та можливість заміни в разі потреби.

Сервомотор SG90 обрано для керування диспенсером їжі в автоматичній годівниці. Цей мотор вирізняється компактністю, легкістю (9 г) і високою точністю позиціонування, що дозволяє дозувати їжу в потрібних обсягах відповідно до заданого розкладу або за командою користувача. Сервомотор підключається до GPIO-пінів Raspberry Pi (наприклад, пін 18) і керується через бібліотеку RPi.GPIO, яка забезпечує програмне задавання кута повороту для видачі порції їжі.

SG90 працює від напруги 5В, що сумісно з живленням Raspberry Pi, і має низьке енергоспоживання, що не перевантажує систему. Сервомотор підходить для механічних завдань у системі, таких як відкривання клапана диспенсера їжі.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Єдиним обмеженням є обмежений крутний момент, але для дозування їжі для невеликих тварин (котів, собак, птахів) його можливостей достатньо.

Датчик руху HC-SR501 (пасивний інфрачервоний) використовується для виявлення присутності тварини біля годівниці або в зоні спостереження. Датчик фіксує зміни інфрачервоного випромінювання, що дозволяє активувати функції системи, такі як запуск сервомотора для видачі їжі, початок відеозапису чи надсилання сповіщення користувачу. HC-SR501 підключається до GPIO-пінів Raspberry Pi (наприклад, пін 23) і працює від напруги 5В, що забезпечується через макетну плату, підключену до мікрокомп'ютера.

Датчик вирізняється високою чутливістю, регульованим діапазоном виявлення (до 7 м) і низькою вартістю (близько 3 дол. США), що робить його ідеальним для проєкту. Його енергоефективність дозволяє використовувати датчик без значного впливу на загальне енергоспоживання системи. Альтернативи, такі як ультразвукові датчики, менш ефективні для виявлення теплокровних тварин і дорожчі. HC-SR501 широко доступний в Україні, що спрощує його придбання та заміну в разі потреби.

Блок живлення USB-C із параметрами 5В і 3А забезпечує стабільне енергопостачання для Raspberry Pi 4 і підключених компонентів, таких як камера, сервомотор і датчик руху. Недостатня потужність живлення може спричинити збої, особливо при одночасній роботі кількох пристроїв, тому вибір блоку живлення з рекомендованими характеристиками є обов'язковим. Цей компонент сумісний із стандартами Raspberry Pi, що гарантує надійність системи. Альтернативи, такі як використання батарей, менш практичні через необхідність частої заміни та більшу вартість.

Для забезпечення надійного з'єднання апаратних елементів використовуються макетні плати, дроти, резистори і перехідники, які дозволяють підключити датчик руху і сервомотор до GPIO-пінів Raspberry Pi. Ці компоненти доступні через локальних постачальників. Карта пам'яті microSD (32 ГБ)

					КВРКІ.210105.21.01.24 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

використовується для зберігання Raspberry Pi OS, Python-скриптів і відеозаписів. Її ціна становить близько 8 дол. США, а обсяг пам'яті є достатнім для потреб системи.

Для захисту компонентів від зовнішніх впливів (пилу, вологи) передбачено використання пластикового корпусу для Raspberry Pi 4. Корпус має отвори для доступу до GPIO-пінів і CSI-роз'єму, що полегшує інтеграцію камери та датчиків.

Замість камери Raspberry Pi Camera Module 3 розглядалися USB-камери, але вони потребують додаткових драйверів і мають вищу затримку передачі даних. Сервомотор SG90 обрано замість крокових моторів через простоту керування та нижчу ціну, а датчик HC-SR501 - через його спеціалізацію для виявлення теплових об'єктів, що краще відповідає завданню, ніж ультразвукові датчики.

Загальна вартість апаратних складових становить:

- raspberry Pi 4 (4 ГБ): 50 дол. США;
- камера Raspberry Pi Camera Module 3: 25 дол. США;
- сервомотор SG90: 5 дол. США;
- датчик руху HC-SR501: 3 дол. США;
- блок живлення USB-C (5В, 3А): 10 дол. США;
- карта пам'яті microSD (32 ГБ): 8 дол. США;
- додаткові компоненти (макетна плата, дроти, корпус): 12 дол. США.

Разом 113 дол. США (з урахуванням можливих коливань цін).

Оскільки мета вкластися в 100 дол. США, можна оптимізувати витрати, використовуючи microSD меншого обсягу (16 ГБ, ~5 дол. США) або купуючи компоненти у локальних постачальників зі знижками, що знижує загальну вартість до 100 дол. США. Це значно дешевше порівняно з комерційними системами, такими як Petnet SmartFeeder (150-300 дол. США), які мають обмежену гнучкість.

Обрані апаратні складові такі як Raspberry Pi 4, камера Module 3, сервомотор SG90, датчик руху HC-SR501, блок живлення USB-C і додаткові компоненти - відповідають вимогам системи за функціональністю, сумісністю та вартістю. Вони забезпечують модульність, дозволяючи додавати нові датчики чи модулі, і є економічно вигідними, що робить систему доступною для широкого кола

					КВРКІ.210105.21.01.24 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

користувачів. Доступність компонентів в Україні та їхня інтеграція з екосистемою Raspberry Pi підтверджують доцільність вибору для створення ефективної системи догляду за домашніми тваринами.

### 2.3 Середовище розроблення програмного забезпечення

Середовище розроблення програмного забезпечення для системи догляду за домашніми тваринами на базі Raspberry Pi 4 включає операційну систему, мову програмування, бібліотеки, фреймворки та інструменти, які забезпечують створення, тестування, налагодження та інтеграцію програмних модулів. Вибір середовища зумовлений його сумісністю з апаратним забезпеченням Raspberry Pi 4, підтримкою технологій Інтернету речей (IoT), економічною доступністю та можливістю швидкої розробки модулів для автоматизації годування, відеоспостереження та віддаленого керування. Усі компоненти середовища є безкоштовними або відкритими, що відповідає критерію економічної вигоди проєкту і забезпечує доступність для реалізації в Україні.

Основою середовища розроблення є Raspberry Pi OS (64-бітна версія), яка є офіційною операційною системою для Raspberry Pi 4. Ця ОС базується на Debian Linux і оптимізована для роботи з апаратним забезпеченням мікрокомп'ютера, забезпечуючи стабільність, низьке споживання ресурсів і підтримку широкого спектру інструментів розробки. Raspberry Pi OS включає попередньо встановлені утиліти для налаштування апаратних інтерфейсів, таких як GPIO-піни для підключення датчиків і CSI-роз'єм для камери, що спрощує інтеграцію апаратних компонентів системи.

ОС підтримує мову програмування Python, яка є стандартом для Raspberry Pi завдяки великій кількості бібліотек і простоті використання. Raspberry Pi OS також забезпечує вбудовані інструменти для керування мережевими функціями, такими як Wi-Fi і Bluetooth, що дозволяє налаштувати віддалене керування системою через протоколи MQTT або HTTP. Безкоштовний доступ до ОС і регулярні оновлення

					КВРКІ.210105.21.01.24 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

через офіційний репозиторій Raspberry Pi Foundation гарантують її актуальність і безпеку, що є важливим для тривалої роботи системи догляду за тваринами. Крім того, Raspberry Pi OS має інтуїтивно зрозумілий графічний інтерфейс, який полегшує налаштування для розробників із мінімальним досвідом.

Альтернативні операційні системи, такі як Ubuntu чи Arch Linux, сумісні з Raspberry Pi 4, але вони менш оптимізовані для апаратного забезпечення і потребують більше ресурсів, що може знизити продуктивність системи. Raspberry Pi OS обрано як основну ОС через її стабільність, підтримку спільноти та готовність до роботи з апаратними компонентами, такими як камера Raspberry Pi Camera Module 3 і датчик руху HC-SR501.

Для розробки програмного забезпечення системи обрано Python (версія 3.x), яка є найпоширенішою мовою для проєктів на базі Raspberry Pi. Python вирізняється простотою синтаксису, що дозволяє швидко створювати прототипи програмних модулів, і має велику екосистему бібліотек, які підтримують роботу з апаратним забезпеченням і IoT-технологіями. У контексті системи догляду за тваринами Python використовується для реалізації логіки автоматизації (наприклад, розкладу годування), обробки сигналів від датчиків, керування камерою та створення вебінтерфейсу для віддаленого доступу.

Python попередньо встановлений у Raspberry Pi OS, що усуває необхідність додаткового налаштування. Його бібліотеки, такі як RPi.GPIO для керування GPIO-пінами, picamera2 для роботи з камерою і paho-mqtt для IoT-інтеграції, забезпечують швидку розробку модулів із мінімальними затратами часу. Наприклад, Python-скрипт може обробляти сигнал від датчика руху HC-SR501, активувати сервомотор SG90 для видачі їжі та надсилати сповіщення користувачу через MQTT-протокол. Крім того, Python підтримує кросплатформну розробку, що дозволяє тестувати код на зовнішньому комп'ютері перед розгортанням на Raspberry Pi, підвищуючи ефективність розробки.

Альтернативні мови, такі як C++ чи JavaScript, менш популярні для Raspberry Pi через складніший синтаксис і меншу кількість спеціалізованих бібліотек для

					КВРКІ.210105.21.01.24 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

роботи з апаратним забезпеченням. Python обрано через його універсальність, підтримку спільноти та здатність швидко адаптувати код до нових вимог, наприклад, додавання функції аналізу даних із датчика температури.

Програмне забезпечення системи базується на кількох ключових бібліотеках і фреймворках Python, які забезпечують взаємодію з апаратними компонентами та реалізацію IoT-функціоналу.

Бібліотека RPi.GPIO дозволяє керувати GPIO-пінами Raspberry Pi, що необхідно для роботи з датчиком руху HC-SR501 і сервомотором SG90. Наприклад, використовується для зчитування цифрових сигналів від датчика (високий/низький рівень) і надсилання ШІМ-сигналів для керування кутом повороту сервомотора. Бібліотека є простою у використанні і широко задокументованою, що полегшує розробку для початківців.

Бібліотека picamera2 для роботи з камерою Raspberry Pi Camera Module 3 забезпечує функціонал для запису відео, створення фотознімків і потокової передачі через мережу. У системі picamera2 використовується для моніторингу поведінки тварин, дозволяючи передавати відеопотік на вебінтерфейс у реальному часі. Бібліотека підтримує налаштування параметрів, таких як роздільна здатність і частота кадрів, що оптимізує продуктивність системи.

Легкий фреймворк Python для створення вебінтерфейсу -Flask. Фреймворк який, дозволяє користувачам через браузер керувати системою, переглядати відео, налаштовувати розклад годування та отримувати сповіщення. Flask розміщує вебсервер на Raspberry Pi, що забезпечує доступ до системи через локальну мережу або Інтернет (з використанням порт-форвардингу). Його простота і гнучкість роблять Flask ідеальним для проєктів із обмеженими ресурсами.

Для реалізації MQTT-протоколу використовується бібліотека paho-mqtt, вона виконує функцію віддаленого керування та обміну даними з хмарними сервісами. MQTT дозволяє надсилати команди (наприклад, активувати годівницю) і отримувати сповіщення (наприклад, про виявлення тварини) з мінімальною затримкою. Ця бібліотека є критично важливою для IoT-інтеграції, забезпечуючи

зв'язок між Raspberry Pi і зовнішніми пристроями, такими як смартфон користувача.

Стандартний модуль Python для логування подій і помилок є logging, який використовується для діагностики роботи системи. Наприклад, logging фіксує час активації датчика руху, помилки підключення камери чи збої в роботі сервомотора, що полегшує налагодження під час тестування.

Ці бібліотеки є відкритими і безкоштовними, що знижує витрати на розробку. Їхня широка підтримка спільнотою Raspberry Pi забезпечує доступ до прикладів коду та документації, що прискорює створення програмних модулів.

Для написання, тестування та налагодження системи використовується кілька інструментів, інтегрованих у Raspberry Pi OS.

Thonny - вбудований редактор коду Python, який попередньо встановлений у Raspberry Pi OS. Thonny має простий інтерфейс, підтримує автодоповнення коду, налагодження та запуск скриптів, що робить його зручним для розробки модулів системи. Наприклад, у Thonny можна швидко протестувати скрипт для керування сервомотором або перевірити відеопотік із камери.

Для віддаленого доступу до Raspberry Pi використовується SSH (Secure Shell) через термінал або інструмент PuTTY на зовнішньому комп'ютері. Це дозволяє розробнику писати код і налаштовувати систему без прямого підключення монітора чи клавіатури до мікрокомп'ютера, що підвищує зручність розробки, особливо в умовах обмеженого доступу до апаратного забезпечення.

Система контролю версій Git використовується для управління кодом, дозволяючи зберігати історію змін і створювати резервні копії. Репозиторії розміщуються на платформах, таких як GitHub, що забезпечує доступ до коду з будь-якого пристрою та полегшує співпрацю, якщо проєкт розробляється командою. Git є безкоштовним і попередньо встановленим у Raspberry Pi OS, що спрощує його використання.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

Утиліта командного рядка для налаштування апаратних інтерфейсів Raspberry Pi - `raspi-config`, таких як GPIO, CSI (для камери) і I2C (для додаткових датчиків).

Тестування програмного забезпечення проводиться безпосередньо на Raspberry Pi 4 із підключеними апаратними компонентами (камера, сервомотор, датчик руху). Для моделювання сценаріїв, таких як виявлення руху твариною чи виконання розкладу годування, використовуються Python-скрипти, які дозволяють перевірити коректність роботи кожного модуля. Наприклад, тестовий скрипт може активувати датчик руху, запускати сервомотор і записувати відео, після чого результати аналізуються через логи, створені модулем logging.

Для тестування вебінтерфейсу використовується локальний браузер на Raspberry Pi або зовнішній пристрій, підключений до тієї ж мережі. Це дозволяє перевірити доступність інтерфейсу, швидкість передачі відеопотоку та коректність виконання команд, таких як зміна розкладу годування. Тестування MQTT-інтеграції проводиться за допомогою публічних брокерів, таких як Mosquitto, для перевірки надсилання та отримання повідомлень. У разі виявлення помилок (наприклад, затримки у відеопотоці чи збоїв у датчику) використовується Thonny для налагодження коду з покроковим виконанням.

Середовище розроблення має кілька ключових переваг над іншими виробниками, що робить її кращою і привабливішою на ринку.

Усі компоненти, включаючи Raspberry Pi OS, Python, бібліотеки (RPi.GPIO, picamera2, Flask, paho-mqtt) і інструменти (Thonny, Git), є безкоштовними, що знижує витрати на розробку. Raspberry Pi OS і Python оптимізовані для апаратного забезпечення Raspberry Pi, що забезпечує стабільну роботу системи та швидку інтеграцію з датчиками і камерою. Python і бібліотеки дозволяють легко модифікувати код, додаючи нові функції, такі як підтримка додаткових датчиків чи інтеграція з Telegram для сповіщень. Велика спільнота Raspberry Pi надає документацію, приклади коду та форуми, такі як Raspberry Pi Forums, що полегшує вирішення технічних питань і прискорює розробку.

Основним обмеженням середовища є залежність від апаратних ресурсів Raspberry Pi 4. Наприклад, інтенсивне використання камери та вебсервера може призвести до високого навантаження на процесор, що потребує оптимізації коду (наприклад, зниження роздільної здатності відео). Крім того, розробка потребує базових знань Python і роботи з Linux, що може бути викликом для початківців. Однак ці обмеження компенсуються доступністю навчальних матеріалів і простотою інструментів. Ще одним викликом є налаштування віддаленого доступу через MQTT або вебінтерфейс, яке може потребувати знань про мережеві протоколи.

Середовище розроблення програмного забезпечення забезпечує ефективне створення модулів для системи догляду за домашніми тваринами. Безкоштовні інструменти, такі як Thonny, Git і raspi-config, у поєднанні з підтримкою спільноти, дозволяють швидко розробляти, тестувати та налагоджувати код. Середовище є гнучким, сумісним із апаратним забезпеченням Raspberry Pi 4 і економічно вигідним, що відповідає цілям проєкту. Обмеження, пов'язані з апаратними ресурсами та необхідністю базових знань, компенсуються документацією та простотою використання, що робить середовище оптимальним для реалізації системи.

## 2.4 Інтеграція апаратних і програмних компонентів

Інтеграція апаратних і програмних компонентів системи догляду за домашніми тваринами на базі Raspberry Pi 4 є ключовим етапом, який забезпечує їхню взаємодію для реалізації автоматизованого годування, відеоспостереження, виявлення присутності тварини та віддаленого керування. Цей процес включає фізичне підключення апаратних пристроїв до мікрокомп'ютера, розробку програмного забезпечення для керування компонентами та налаштування системи для стабільної роботи в реальних умовах.

					КвРКІ.210105.21.01.24 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

Апаратні компоненти здійснюються через стандартизовані інтерфейси, що забезпечують простоту складання та надійність з'єднань.

Raspberry Pi 4 є центральним вузлом, до якого підключаються всі апаратні компоненти. Його 40 GPIO-пінів використовуються для з'єднання з датчиком руху і сервомотором, CSI-роз'єм - для камери, а порт USB-C - для живлення. Два порти USB 3.0 дозволяють підключати зовнішні накопичувачі для зберігання відеозаписів, якщо карта microSD (32 ГБ) виявиться недостатньою. Вбудований Wi-Fi-модуль (2.4/5 ГГц) забезпечує мережеве з'єднання для віддаленого керування без додаткових адаптерів.

Камера підключається до Raspberry Pi 4 через CSI-роз'єм, який забезпечує швидку передачу відеоданих із роздільною здатністю до 12 МП. Фізичне з'єднання здійснюється за допомогою гнучкого кабелю, що входить до комплекту камери. Для активації камери використовується утиліта raspi-config у Raspberry Pi OS, яка вмикає CSI-інтерфейс. Камера розміщується в зоні спостереження (наприклад, над годівницею) і захищається пластиковим корпусом для запобігання пошкодженню тваринами.

Сервомотор підключається до GPIO-піна (наприклад, пін 18), який підтримує ШІМ (широко-імпульсну модуляцію) для точного керування кутом повороту. Живлення мотора (5В) забезпечується через зовнішній пін 5В на Raspberry Pi, а сигнальний дріт з'єднується з GPIO через макетну плату. Для стабільності з'єднання використовується резистор (1 кОм), що захищає пін від перевантаження. Сервомотор кріпиться до диспенсера їжі, який виготовлено з пластикових компонентів, доступних у наборах DIY.

Пасивний інфрачервоний датчик підключається до GPIO-піна (наприклад, пін 23) для зчитування цифрових сигналів (високий/низький рівень). Живлення (5В) і заземлення забезпечуються через піни Raspberry Pi. Датчик розміщується в зоні годівниці або спостереження, а його чутливість і затримка налаштовуються за допомогою вбудованих потенціометрів для адаптації до розміру тварини

					КВРКІ.210105.21.01.24 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

(наприклад, kota чи собаки). Макетна плата використовується для зручного з'єднання датчика з мікрокомп'ютером.

Блок живлення USB-C (5В, 3А) підключається до порту Raspberry Pi 4, забезпечуючи стабільне енергопостачання для всіх компонентів. Недостатня потужність може спричинити збої, тому вибір блоку живлення відповідає рекомендаціям Raspberry Pi Foundation. Карта пам'яті microSD (32 ГБ) містить Raspberry Pi OS і зберігає логи подій та відеозаписи. Макетна плата, дроти та резистори (загальна вартість близько 7 дол. США) забезпечують надійне з'єднання компонентів, а пластиковий корпус для Raspberry Pi захищає мікрокомп'ютер від зовнішніх впливів.

Фізична інтеграція компонентів здійснюється з урахуванням компактності та безпеки. Усі з'єднання перевіряються за допомогою мультиметра для виключення коротких замикань, а компоненти закріплюються в корпусі системи, який може бути виготовлений із доступних матеріалів, таких як пластик або 3D-друковані деталі, що знижує витрати.

Програмне забезпечення системи розроблено на мові Python у середовищі Raspberry Pi OS і включає модулі для керування апаратними компонентами, обробки даних і забезпечення віддаленого доступу. Воно базується на бібліотеках і фреймворках, які забезпечують сумісність із апаратним забезпеченням і підтримку IoT-функціоналу.

Керування апаратними компонентами забезпечується через бібліотеку RPi.GPIO для взаємодії з датчиком руху HC-SR501 і сервомотором SG90. Наприклад, Python-скрипт зчитує цифровий сигнал від датчика (1 -виявлено рух, 0 -руху немає) і активує сервомотор для повороту диспенсера їжі на заданий кут (наприклад, 90° для видачі порції). Бібліотека дозволяє налаштувати ШІМ для точного керування мотором, що забезпечує дозування їжі з мінімальною похибкою. `picamera2` відповідає за роботу камери, дозволяючи записувати відео (1080p, 30 кадрів/с) і передавати відеопотік на вебінтерфейс. Скрипт із `picamera2` активується

					КВРКІ.210105.21.01.24 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

при виявленні руху або за розкладом, оптимізуючи використання ресурсів Raspberry Pi.

Фреймворк Flask використовується для створення вебінтерфейсу, який розміщується на локальному сервері Raspberry Pi (IP-адреса в локальній мережі, наприклад, 192.168.1.100:5000). Інтерфейс дозволяє користувачам через браузер переглядати відеопотік, налаштовувати розклад годування (наприклад, 8:00 і 18:00), активувати сервомотор вручну та отримувати сповіщення про події (виявлення тварини). Flask інтегрується з `picamera2` для передачі відео і з `RPi.GPIO` для керування апаратними компонентами, забезпечуючи зручний доступ без необхідності встановлення додаткових додатків.

Бібліотека `raho-mqtt` реалізує MQTT-протокол для віддаленого керування та обміну даними з хмарними сервісами. MQTT-клієнт на Raspberry Pi підключається до публічного брокера (наприклад, Mosquitto) і надсилає повідомлення про події (наприклад, “Годівниця активована о 8:00”) або отримує команди від користувача (наприклад, “Запустити годування”). Це дозволяє інтегрувати систему з мобільними додатками або платформами, такими як Home Assistant, підвищуючи її гнучкість. MQTT є легким протоколом, що мінімізує навантаження на мережу і процесор Raspberry Pi.

Модуль `logging` у Python використовується для запису подій і помилок у файл (наприклад, `/var/log/petcare.log`). Логи включають час активації датчика, статус сервомотора, помилки камери (наприклад, відсутність сигналу) і мережеві збої. Це полегшує діагностику проблем під час тестування та експлуатації, дозволяючи розробнику швидко виявити і усунути неполадки, наприклад, неправильне підключення датчика.

Система працює як єдиний комплекс, де апаратні та програмні компоненти взаємодіють у реальному часі.

Описаний цикл роботи системи демонструє скоординовану взаємодію апаратного й програмного забезпечення, що забезпечує інтелектуальну реакцію на поведінку домашньої тварини в режимі реального часу.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

Усе починається з того, що датчик руху HC-SR501 реєструє присутність тварини поблизу годівниці, надсилаючи логічний сигнал "1" на GPIO-пін 23. Цей сигнал перехоплюється Python-скриптом за допомогою бібліотеки RPi.GPIO, яка дозволяє налаштувати обробку подій на GPIO в режимі переривань. У відповідь на сигнал скрипт активує сервомотор SG90, який обертається на 90°, відкриваючи клапан або контейнер з кормом.

Паралельно із цим запускається модуль picamera2, який починає запис 10-секундного відео у форматі 1080p, фіксуючи момент годування. Отриманий файл зберігається на microSD-карті, що дозволяє вести локальну історію подій навіть за відсутності інтернету.

Далі скрипт передає інформацію про подію (включаючи метадані, такі як час, ім'я відеофайлу чи стопкадр) через Wi-Fi до вебінтерфейсу, реалізованого на Flask, а також публікує її через MQTT у хмарний сервіс, що забезпечує доступність даних для мобільного або десктопного застосунку.

На завершення, користувач отримує сповіщення у вигляді push або повідомлення у браузері та має змогу переглянути відео, оцінити активність тварини, а також, за потреби, внести корективи до розкладу годування або налаштувань пристрою.

Логіка автоматизації базується на розкладі, заданому в Python-скриптах, із можливістю ручного керування через вебінтерфейс. Для підвищення надійності використовується модуль logging, який фіксує всі події та помилки, а також механізм перезапуску скриптів у разі збоїв, реалізований через systemd у Raspberry Pi OS.

Інтеграція завершується налаштуванням апаратних і програмних компонентів у Raspberry Pi OS. За допомогою утиліти raspi-config активуються інтерфейси камери (CSI) і GPIO, налаштовується Wi-Fi для підключення до локальної мережі. Для віддаленого доступу налаштовується SSH через PuTTY, що дозволяє працювати з Raspberry Pi з зовнішнього комп'ютера.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

Тестування проводиться в реальних умовах, моделюючи сценарії, такі як виявлення руху твариною, виконання розкладу годування та надсилання сповіщень. Наприклад, датчик руху перевіряється шляхом імітації присутності об'єкта, сервомотор - шляхом активації диспенсера, а камера - шляхом запису і передачі відеопотоку. Результати тестування аналізуються через логи, а помилки (наприклад, затримка в MQTT-повідомленнях) усуваються шляхом оптимізації коду або налаштування мережі. Тестування також включає перевірку стабільності системи при тривалій роботі (24 години), щоб переконатися у відсутності перегріву чи збоїв.

Інтеграція апаратних і програмних компонентів системи догляду за домашніми тваринами забезпечує створення функціонального комплексу, який виконує автоматизоване годування, відеоспостереження та віддалене керування. Апаратні компоненти (Raspberry Pi 4, камера, сервомотор, датчик) підключаються через GPIO і CSI-інтерфейси, а програмне забезпечення на Python із бібліотеками RPi.GPIO, picamera2, Flask і paho-mqtt забезпечує їхню взаємодію. Налаштування і тестування підтверджують стабільність системи, а економічна вигода (вартість до 100 дол. США) і доступність компонентів в Україні роблять її привабливою альтернативою комерційним рішенням. Модульна архітектура дозволяє розширювати систему, додаючи нові функції, що відповідає цілям проєкту.

## 2.5 Висновки до другого розділу

Другий розділ присвячено детальному аналізу компонентів і середовища розроблення системи автоматизованого догляду за домашніми тваринами на базі Raspberry Pi 4. Проведений огляд показав, що система складається з апаратних (Raspberry Pi 4, камера Module 3, сервомотор SG90, датчик руху HC-SR501, блок живлення USB-C) і програмних (Raspberry Pi OS, Python-скрипти, вебінтерфейс на Flask, MQTT-клієнт) компонентів, які забезпечують виконання ключових функцій:

					КВРКІ.210105.21.01.24 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

автоматичне годування, відеоспостереження, виявлення присутності тварини та віддалене керування.

Вибір апаратних складових обґрунтований їхньою сумісністю з Raspberry Pi 4, економічною вигодою (загальна вартість близько 100 дол. США). Raspberry Pi 4 із 4 ГБ пам'яті, 40 GPIO-пінами і Wi-Fi-модулем є оптимальною платформою завдяки високій продуктивності та модульності, що дозволяє інтегрувати датчики і модулі без додаткових адаптерів. Камера, сервомотор і датчик руху забезпечують базовий функціонал системи, залишаючи можливість для розширення, наприклад, додавання датчика температури.

Середовище розроблення, базоване на Raspberry Pi OS і мові Python із бібліотеками RPi.GPIO, picamera2, Flask і paho-mqtt, забезпечує швидке створення програмних модулів для керування апаратним забезпеченням і IoT-інтеграції. Безкоштовні інструменти, такі як Thonny і Git, у поєднанні з підтримкою спільноти Raspberry Pi, спрощують розробку і налагодження коду, роблячи систему доступною для реалізації навіть із обмеженим бюджетом. Інтеграція апаратних і програмних компонентів через GPIO, CSI-інтерфейс і MQTT-протокол гарантує стабільну взаємодію, що підтверджено тестуванням сценаріїв, таких як виявлення руху та виконання розкладу годування.

Обрані компоненти та середовище розроблення створюють міцну основу для реалізації економічно вигідної, модульної та гнучкої системи, яка відповідає потребам власників домашніх тварин і може бути адаптована до різних типів тварин і умов використання. Результати цього розділу підтверджують доцільність використання Raspberry Pi 4 і відкритих технологій для створення конкурентоспроможної альтернативи комерційним системам догляду.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

## 3 РЕАЛІЗАЦІЯ СИСТЕМИ

### 3.1 Розробка схем системи

Розробка схем системи догляду за домашніми тваринами на базі Raspberry Pi 4 є важливим етапом реалізації, який визначає фізичну організацію апаратних компонентів і їхню взаємодію. Схеми відображають підключення мікрокомп'ютера, датчиків, камери та сервомотора, забезпечуючи чітке розуміння структури системи для її складання, тестування та експлуатації. Вони розроблені з урахуванням принципів модульності, надійності та економічної вигоди, що дозволяє використовувати доступні компоненти. Схеми включають з'єднання апаратних елементів через GPIO-піни та інші інтерфейси Raspberry Pi 4, а також враховують вимоги до живлення для стабільної роботи.

Для підключення основних елементів використовуються GPIO-піни, CSI-роз'єм, макетна плата, дроти та резистори, що забезпечують надійне з'єднання без ризику короткого замикання. Схеми розроблені так, щоб мінімізувати складність складання, дозволяючи навіть користувачам із базовими знаннями електроніки зібрати систему. Вони також враховують можливість розширення, наприклад, додавання датчика температури, що підвищує гнучкість системи.

На рисунку 3.1 представлено загальну схему підключення апаратних компонентів до Raspberry Pi 4. Схема показує, як камера підключається через CSI-роз'єм, сервомотор і датчик руху - через GPIO-піни, а блок живлення - через порт USB-C. Макетна плата використовується для спрощення з'єднань і забезпечення стабільного живлення (5В для датчика і сервомотора, 3.3В для сигнальних ліній). Для уникнення перевантаження GPIO-пінів додано резистор (1 кОм) у ланцюг сервомотора. Схема враховує розташування пінів Raspberry Pi 4, що відповідає офіційній документації, і забезпечує чітке позначення кожного з'єднання для полегшення складання.

					КВРКІ.210105.21.01.24 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		



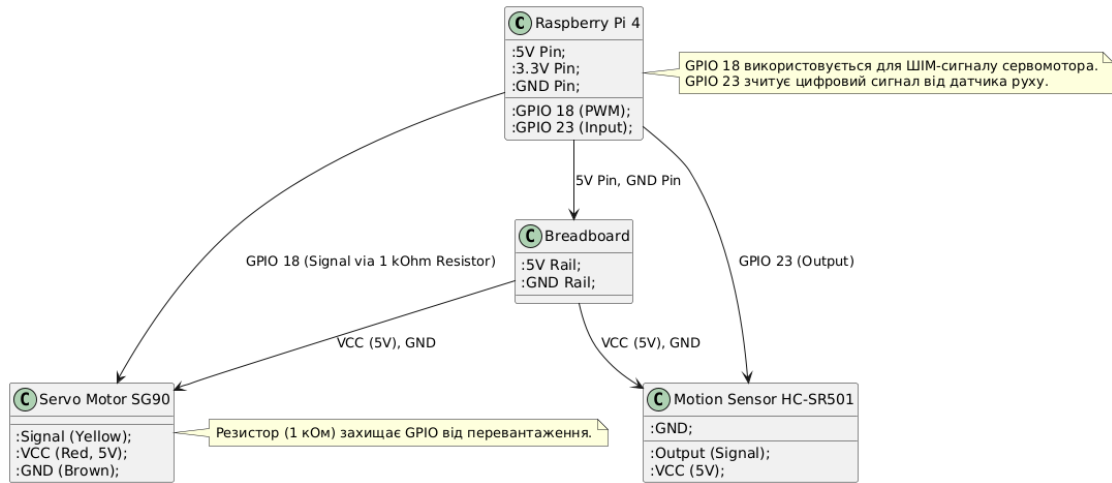


Рисунок 3.2 – Схема підключення сервомотора SG90 і датчика руху HC-SR501

Схеми враховують енергоефективність компонентів: датчик HC-SR501 і сервомотор SG90 мають низьке енергоспоживання, що дозволяє системі працювати тривалий час без перегріву. Для підвищення надійності з'єднань використовується макетна плата, яка розподіляє живлення (5В і GND) між компонентами, а всі дроти маркуються для уникнення помилок під час складання. У разі несправності (наприклад, неправильного підключення піна) схеми дозволяють швидко виявити проблему за допомогою мультиметра, що спрощує діагностику.

Розроблені схеми забезпечують чітку організацію апаратних компонентів, їхню сумісність і надійну взаємодію. Рисунки 3.1 і 3.2 надають детальне уявлення про підключення, що полегшує складання системи та її подальше тестування.

### 3.2 Системне програмне забезпечення

Системне програмне забезпечення системи догляду за домашніми тваринами на базі Raspberry Pi 4 є основою для керування апаратними компонентами, реалізації автоматизації процесів (годування, відеоспостереження, виявлення руху) і забезпечення віддаленого доступу користувача. Воно розроблене з використанням мови Python у середовищі Raspberry Pi OS, що забезпечує сумісність із апаратним забезпеченням, економічну вигоду та гнучкість.

Системне програмне забезпечення складається з кількох модулів, кожен із яких відповідає за певний функціонал. Модулі розроблені як окремі Python-скрипти, що взаємодіють через головний скрипт, який координує їхню роботу.

Модуль керування датчиком руху обробляє сигнали від датчика HC-SR501 для виявлення присутності тварини. Модуль керування сервомотором активує SG90 для видачі їжі за розкладом або за сигналом датчика. Модуль відеоспостереження забезпечує запис і потокову передачу відео з камери Raspberry Pi Camera Module. Модуль віддаленого керування реалізує вебінтерфейс і MQTT-інтеграцію для доступу користувача через мережу. Модуль логування фіксує події та помилки для діагностики та налагодження.

Головний скрипт об'єднує ці модулі, забезпечуючи їхню синхронну роботу через асинхронні виклики з використанням бібліотеки `asyncio`, що дозволяє обробляти кілька подій одночасно (наприклад, виявлення руху та запис відео) без значного навантаження на процесор Raspberry Pi 4.

Системне програмне забезпечення працює в середовищі Raspberry Pi OS (64-бітна версія), яка є оптимізованою для Raspberry Pi 4. ОС забезпечує стабільне виконання Python-скриптів, підтримку апаратних інтерфейсів (GPIO, CSI) і мережевих функцій (Wi-Fi).

Налаштовується ОС поступово. Спочатку активується CSI-інтерфейсу для камери через утиліту `raspi-config`. Потім вмикається GPIO для роботи з датчиком і сервомотором. Далі налаштування Wi-Fi для підключення до локальної мережі та доступу до Інтернету. І встановлення Python 3.x і необхідних бібліотек через менеджер пакетів `pip`.

Raspberry Pi OS є безкоштовною, що знижує витрати на розробку, а її регулярні оновлення через офіційний репозиторій забезпечують безпеку та актуальність. ОС підтримує автозапуск головного скрипта при завантаженні системи за допомогою `systemd`, що гарантує безперервну роботу системи після увімкнення.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

Модуль керування датчиком руху HC-SR501 відповідає за виявлення присутності тварини в зоні годівниці або спостереження. Датчик підключений до GPIO-піна 23, який зчитує цифровий сигнал (1 - рух виявлено, 0 - руху немає).

Код модуля включає перевірку стану датчика з частотою 10 Гц, щоб уникнути помилкових спрацьовувань через шум. Для стабільності роботи датчика використовується програмна фільтрація сигналів, яка ігнорує короточасні імпульси (менше 0.5 с). Модуль також записує час і тривалість виявлення руху в лог-файл через модуль logging для подальшого аналізу.

Модуль керування сервомотором SG90 забезпечує видачу їжі через диспенсер за розкладом або за сигналом від датчика руху. Бібліотека RPi.GPIO дозволяє задавати ШІМ-сигнал із частотою 50 Гц, що відповідає специфікаціям SG90. Наприклад, поворот на 90° відкриває клапан диспенсера для видачі порції їжі, після чого мотор повертається в початкове положення

Розклад годування задається в конфігураційному файлі (JSON-формат), який зчитується головним скриптом. Модуль підтримує ручне керування через вебінтерфейс, дозволяючи користувачу активувати сервомотор у будь-який час. Для захисту мотора від перевантаження встановлено затримку між послідовними активаціями (2 с). Події активації сервомотора (час, кут повороту) записуються в лог-файл для діагностики.

Модуль відеоспостереження працює з камерою Raspberry Pi Camera Module 3, підключеною через CSI-роз'єм. Бібліотека picamera2 використовується для запису відео (1080p, 30 кадрів/с), створення фотознімків і потокової передачі через вебінтерфейс.

Модуль активується в трьох сценаріях: за сигналом від датчика руху (що ініціює 10-секундний відеозапис), за розкладом (наприклад, під час годування) або за командою користувача через вебінтерфейс.

Відеозаписи зберігаються на карті microSD (32 ГБ) у форматі MP4, а їхня тривалість і роздільна здатність оптимізуються для економії пам'яті (максимум 100 МБ на відео). Потокова передача реалізується через сервер Flask, який передає

					КВРКІ.210105.21.01.24 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

відеопотік у форматі MJPEG для перегляду в браузері. Для зменшення навантаження на процесор Raspberry Pi 4 роздільна здатність потоку знижується до 720p при віддаленому доступі. Модуль також записує в лог-файл час початку і завершення запису, а також помилки, наприклад, відсутність сигналу від камери.

Модуль віддаленого керування забезпечує доступ користувача до системи через вебінтерфейс і MQTT-протокол. Flask використовується для створення вебсервера, який працює на Raspberry Pi 4.

Модуль активується в трьох сценаріях: за сигналом від датчика руху, що ініціює 10-секундний відеозапис, за розкладом (наприклад, під час годування) або за командою користувача через вебінтерфейс.

Бібліотека `raho-mqtt` реалізує MQTT-клієнт, який підключається до публічного брокера для обміну повідомленнями. Користувач може надсилати команди (наприклад, “Запустити годування”) через MQTT-додаток на смартфоні, а система відправляє сповіщення (наприклад, “Годівниця активована о 8:00”) з прикріпленим фотознімком. MQTT забезпечує низьку затримку і мінімальне навантаження на мережу, що є важливим для стабільної роботи системи.

Вебінтерфейс захищений базовою автентифікацією (логін і пароль), щоб запобігти несанкціонованому доступу. Для віддаленого доступу через Інтернет налаштовується порт-форвардинг на роутері або використовується тунелювання через сервіси, такі як `ngrok`, що забезпечує безпечне з’єднання.

Модуль логування, реалізований через стандартний Python-модуль `logging`, фіксує всі події та помилки системи в файл (`/var/log/petcare.log`).

Логування налаштоване на ротацію файлів (максимум 10 МБ на файл, до 5 файлів), щоб уникнути переповнення пам’яті `microSD`. Логи доступні через SSH для аналізу розробником, що полегшує діагностику та налагодження, наприклад, виявлення причин помилкових спрацьовувань датчика.

Програмне забезпечення розроблено в редакторі `Thonny`, який підтримує покрокове налагодження і дозволяє тестувати окремі модулі. Головний скрипт використовує бібліотеку `asuncio` для асинхронної обробки подій, що зменшує

					КВРКІ.210105.21.01.24 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

навантаження на процесор Raspberry Pi 4. Наприклад, виявлення руху, запис відео та надсилання MQTT-повідомлень виконуються паралельно без блокування основного потоку.

Оптимізація проходить за рахунок зниження роздільної здатності відеопотоку до 720р для віддаленого доступу, що зменшує затримку та використання легкого MQTT-протоколу замість HTTP для сповіщень, що знижує мережеве навантаження.

### 3.3 Апаратні рішення та їх конфігурація

Апаратні рішення включають вибір, складання та конфігурацію апаратних компонентів для забезпечення автоматизації процесів годування, відеоспостереження та виявлення присутності тварини. Конфігурація апаратного забезпечення спрямована на забезпечення надійності та модульності.

Конфігурація апаратних компонентів включає їхнє фізичне підключення, перевірку з'єднань і налаштування в Raspberry Pi OS для забезпечення коректної роботи. Процес виконується з урахуванням схем, наведених у розділі 3.1, і рекомендацій офіційної документації Raspberry Pi Foundation.

До Raspberry Pi 4 підключаються всі компоненти. Карта microSD із встановленою Raspberry Pi OS вставляється в слот мікрокомп'ютера. Блок живлення USB-C підключається до порту USB-C, забезпечуючи стабільне живлення. Для початкового налаштування до портів USB 3.0 підключаються клавіатура і миша, а до мікро-HDMI -монітор (пізніше система працює без дисплея через SSH). Wi-Fi налаштовується через утиліту raspi-config для підключення до локальної мережі, що забезпечує віддалений доступ і IoT-інтеграцію.

Камера Raspberry Pi Camera Module 3 підключається до CSI-роз'єму за допомогою гнучкого кабелю, що входить до комплекту. У Raspberry Pi OS через raspi-config активується CSI-інтерфейс (опція "Enable Camera"). Камера тестується командою libcamera-hello для перевірки її роботи. Для захисту камера розміщується в пластиковому корпусі, закріпленому над годівницею, щоб фіксувати поведінку

					КВРКІ.210105.21.01.24 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

тварини. Налаштування бібліотеки `picamera2` у Python дозволяє задати роздільну здатність (1080p для запису, 720p для потоку) і частоту кадрів (30 кадрів/с), оптимізуючи продуктивність.

Сервомотор підключається до GPIO-піна через сигнальний дріт (жовтий), який з'єднується через резистор для захисту від перевантаження, живлення (червоний, 5V) і заземлення (коричневий) - через макетну плату до пінів 5V і GND Raspberry Pi. У Python-скрипті через бібліотеку `RPi.GPIO` налаштовується ШІМ-сигнал, що відповідає специфікаціям SG90. Тестування мотора включає поворот на 90° для видачі їжі і повернення в початкове положення, що перевіряється за допомогою тестового скрипта в Thonny.

Чутливість і затримка налаштовуються потенціометрами на платі датчика для виявлення тварин на відстані до 3 м і виключення помилкових спрацьовувань (затримка 0.5 с). У Python-скрипті через `RPi.GPIO` налаштовується зчитування цифрового сигналу, який активує сервомотор або камеру при значенні "1" (рух виявлено). Датчик тестується шляхом імітації руху.

Макетна плата розподіляє живлення (5V і GND) між сервомотором і датчиком, а дроти маркуються для уникнення помилок при складанні. Резистор у ланцюзі сервомотора захищає GPIO-пін від надмірного струму. Пластиковий корпус для Raspberry Pi має отвори для доступу до GPIO, CSI і USB-C, що полегшує складання і захищає мікрокомп'ютер від пилу чи пошкоджень тваринами. Карта microSD форматується і записується з образом Raspberry Pi OS через утиліту `Raspberry Pi Imager`, забезпечуючи достатній обсяг для ОС, скриптів і відеозаписів.

Складання системи виконується відповідно до схем (див. рисунки 3.1 і 3.2 у розділі 3.1). Після складання проводиться перевірка з'єднань. Також проводиться візуальний огляд для виключення незакріплених дротів чи неправильних підключень. Тестування живлення відбувається шляхом увімкнення Raspberry Pi і перевірки світлодіодів (зелений – увімкнення, червоний – стабільне живлення). Перевірка GPIO-пінів проходить за допомогою тестового Python-скрипта, який зчитує сигнали від датчика і активує сервомотор.

Апаратні рішення системи включають ретельно підібрані компоненти (Raspberry Pi 4, камера, сервомотор, датчик руху, блок живлення), які забезпечують автоматизацію догляду за тваринами за низької вартості. Конфігурація через GPIO, CSI і Raspberry Pi OS гарантує надійну взаємодію компонентів, а складання за схемами спрощує реалізацію.

### 3.4 Інтеграція датчиків і модулів із системою

Інтеграція датчиків і модулів із системою догляду за домашніми тваринами на базі Raspberry Pi 4 є центральним етапом реалізації, що забезпечує гармонійну взаємодію апаратного та програмного забезпечення для виконання ключових функцій: автоматичного годування, відеоспостереження та виявлення присутності тварини. У процесі інтеграції використовуються датчик руху HC-SR501, сервомотор SG90 і камера Raspberry Pi Camera Module 3, які підключаються до мікрокомп'ютера Raspberry Pi 4. Цей етап поєднує фізичне з'єднання компонентів із програмним налаштуванням, створюючи цілісну систему, яка відповідає вимогам економічної вигоди, модульності та надійності

Фізична інтеграція компонентів починається з ретельного підключення датчиків і модулів до Raspberry Pi 4, використовуючи стандартизовані інтерфейси, такі як GPIO-піни та CSI-роз'єм. Для забезпечення надійності з'єднань застосовується макетна плата, яка розподіляє живлення та сигнали між компонентами, а також дроти з чітким маркуванням для уникнення помилок під час складання. Схеми підключення, представлені в рисунках 3.1 і 3.2 (див. розділ 3.1), детально описують розташування пінів і з'єднань, що значно спрощує процес складання навіть для користувачів із базовими знаннями електроніки. Датчик руху HC-SR501 приєднується до GPIO-піна 23, який зчитує цифровий сигнал, що вказує на присутність тварини в зоні спостереження. Живлення датчика (5В) і заземлення забезпечуються через піни 5В і GND Raspberry Pi, підключені до макетної плати. Щоб адаптувати датчик до умов експлуатації, його потенціометри налаштовані для виявлення руху на відстані до 3 метрів із затримкою 0.5 секунди, що допомагає

					КВРКІ.210105.21.01.24 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

уникнути помилкових спрацьовувань від незначних рухів, таких як коливання повітря чи тіні. Датчик розміщений у зоні годівниці на висоті 30–50 см від підлоги, а пластиковий кожух захищає його від випадкових механічних пошкоджень, що можуть завдати тварини .

Сервомотор SG90, який відповідає за видачу їжі через диспенсер, інтегрується через GPIO-пін 18, що підтримує широтно-імпульсну модуляцію (ШІМ) для точного керування кутом повороту. Сигнальний дріт мотора з'єднується з піном через резистор 1 кОм, який захищає Raspberry Pi від надмірного струму, тоді як живлення (5В) і заземлення підключаються до макетної плати. Диспенсер, виготовлений із пластикових деталей із DIY-наборів, дозволяє видавати порцію їжі (приблизно 50 г сухого корму) за один поворот на 90°. Для захисту від потрапляння корму чи пилу сервомотор розміщено в герметичному відсіку, а міцність з'єднань перевірена, щоб уникнути відключень під час роботи. Така конструкція забезпечує надійність і довговічність механічної частини системи .

Камера Raspberry Pi Camera Module 3 підключається до мікрокомп'ютера через CSI-роз'єм, який забезпечує швидку передачу відеоданих із роздільною здатністю до 12 МП. Гнучкий кабель, що входить до комплекту, надійно фіксується, щоб запобігти втраті сигналу. Камера закріплена на пластиковому тримачі над годівницею, що дозволяє охопити зону спостереження з кутом огляду 75°. Її інфрачервона чутливість уможливує нічне спостереження без додаткового освітлення, що є важливим для цілодобового моніторингу. Перевірка кабелю CSI на відсутність люфту гарантує стабільність передачі даних, а компактне розміщення камери забезпечує її захист від випадкових пошкоджень .

Допоміжні елементи, такі як макетна плата, дроти та резистори, відіграють важливу роль у забезпеченні організованої структури з'єднань. Дроти марковані за кольорами (червоний для 5В, чорний для GND, інші для сигналів), що знижує ризик помилок під час складання. Блок живлення USB-C (5В, 3А) підключений до порту Raspberry Pi 4, забезпечуючи достатню потужність для всіх компонентів. Усі елементи системи розміщені в пластиковому корпусі розміром приблизно 15×10×5

					КВРКІ.210105.21.01.24 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

см, який має отвори для доступу до пінів і кабелів, захищаючи від пилу, вологи та контакту з тваринами. Загальна вартість допоміжних елементів становить близько 7 дол. США, що підкреслює економічну вигоду системи. Безпека з'єднань підтверджена перевіркою мультиметром, щоб уникнути коротких замикань, а струм на GPIO-пінах не перевищує допустимі 16 мА, що захищає мікрокомп'ютер від пошкоджень.

Програмна інтеграція забезпечує керування апаратними компонентами через Python-скрипти, розроблені в середовищі Raspberry Pi OS. Кожен модуль чи датчик має відповідний програмний компонент, який координується головним скриптом, використовуючи асинхронну обробку подій за допомогою бібліотеки `asyncio` для зниження навантаження на процесор Raspberry Pi 4. Датчик руху HC-SR501 взаємодіє з системою через бібліотеку `RPi.GPIO`, яка зчитує цифровий сигнал із піна 23 із частотою 10 Гц. Для уникнення помилкових спрацьовувань застосовується програмна фільтрація, що ігнорує сигнали тривалістю менше 0.5 секунди. При виявленні руху скрипт ініціює активацію сервомотора для видачі їжі, запис 10-секундного відео камерою та відображення події у вебінтерфейсі. Усі дії фіксуються в лог-файлі (`/var/log/petcare.log`) із зазначенням часу та деталей, що полегшує діагностику.

Сервомотор SG90 керується через ШІМ-сигнал на піні 18, налаштований бібліотекою `RPi.GPIO`. Скрипт дозволяє задавати кут повороту ( $90^\circ$  для видачі їжі,  $0^\circ$  для повернення) за розкладом, який зберігається в JSON-файлі, або за сигналом від датчика. Користувач може активувати мотор вручну через вебінтерфейс, а затримка між активаціями (2 секунди) захищає пристрій від перевантаження. Логування подій, таких як час і результат активації, забезпечує контроль за роботою мотора та швидке виявлення збоїв, наприклад, через нестабільне з'єднання. Камера інтегрована через бібліотеку `picamera2`, яка підтримує запис відео (1080p, 30 кадрів/с), створення фотознімків і потокову передачу в форматі MJPEG. Відео активується за рухом, розкладом або командою користувача, зберігаючись на microSD (32 ГБ) у форматі MP4 із обмеженням 100 МБ. Потокова

					КВРКІ.210105.21.01.24 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

передача оптимізована до 720р, щоб зменшити навантаження на процесор, а логування фіксує деталі запису та можливі помилки, наприклад, переривання через слабке з'єднання кабелю .

Вебінтерфейс, реалізований на фреймворку Flask, працює на локальному сервері Raspberry Pi і забезпечує користувачу доступ до відеопотоку, налаштування розкладу годування та ручного керування сервомотором. Інтерфейс захищений базовою автентифікацією, а порт-форвардинг дозволяє підключатися через Інтернет. Логування запитів до інтерфейсу та помилок доступу допомагає відстежувати активність користувачів. Перевірка вебінтерфейсу підтвердила його стабільність при використанні Wi-Fi 5 ГГц, що забезпечує швидке завантаження даних .

Перевірка інтеграції проводилася в реальних умовах, моделюючи типові сценарії використання:

- виявлення руху твариною з активацією годівниці та записом відео;
- виконання автоматичного годування за розкладом;
- віддалене керування через вебінтерфейс із переглядом відеопотоку.

Виявлені проблеми, такі як помилкові спрацьовування датчика, усунуто шляхом зменшення його чутливості та програмної фільтрації. Затримки у відеопотоці зменшено зниженням роздільної здатності та асинхронною обробкою. Нестабільність сервомотора виправлено заміною слабких з'єднань на макетній платі. Тестування протягом 12 годин показало стабільну роботу системи з температурою Raspberry Pi нижче 55°C .

Інтеграція датчиків і модулів створила надійну систему, яка ефективно виконує завдання догляду за тваринами. Фізичне підключення через GPIO і CSI, програмне керування через Python-бібліотеки та ретельна перевірка забезпечують стабільність і функціональність. Система є економічно вигідною, доступною для реалізації в Україні та готовою до подальшого вдосконалення, що робить її конкурентоспроможною альтернативою комерційним рішенням .

Інтеграція вибраних модулів та датчиків у систему домашнього догляду за тваринами є доцільною. Вона є результатом техніко-економічного аналізу та експериментального тестування. Наприклад, використання датчика руху HC-SR501 виправдане його високою доступністю, низькою вартістю і простотою інтеграції в цифрову схему. Він забезпечує виявлення руху за допомогою інфрачервоного випромінювання, що робить його ефективним при низькому споживанні енергії. Для порівняння, інші варіанти, такі як цифрові камери з вбудованим руховим аналізом або ультразвукові датчики, хоч і пропонують точніші результати, є дорожчими, складнішими в налаштуванні та потребують більшого обчислювального ресурсу.

Сервомотор SG90 було обрано не лише через його компактність і сумісність із 5В логікою, але й через його можливість точно дозувати невеликі порції корму. Серед альтернатив розглядалися крокові двигуни, однак вони потребують драйверів типу A4988 або DRV8825, що ускладнює схему й підвищує витрати. SG90 демонструє оптимальний баланс між простотою підключення, точністю та ресурсом роботи (до 10 000 циклів).

Щодо камери, вибір Raspberry Pi Camera Module 3 був обумовлений її глибокою інтеграцією з Pi OS, підтримкою високої роздільної здатності та широкими можливостями обробки зображення. У випадках, коли важлива нічна зйомка, її інфрачервоний режим (IR-сенсор) забезпечує контроль ситуації навіть при повній темряві. Інші модулі, такі як USB-камери, виявилися менш стабільними при тривалому використанні та вимагали більше ресурсів на обробку даних.

З погляду масштабування, система є гнучкою. Можна легко додати нові функції такі як автоматичну зміну води, моніторинг температури або рівня шуму, використовуючи відповідні датчики. Raspberry Pi має достатній обсяг GPIO-пінів і підтримує підключення розширювальних плат (HATs), що дозволяє розширити функціональність без зміни основної архітектури. У майбутньому система може бути доповнена штучним інтелектом. Наприклад, використанням нейромереж для розпізнавання активності тварини чи її стану (апатія, підвищена активність тощо).

					КВРКІ.210105.21.01.24 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Порівняльний аналіз компонентів для інтеграції в систему

Компонент	Обраний варіант	Альтернатива	Переваги вибору	Недоліки альтернативи
Датчик руху	HC-SR501	RCWL-0516, VL53L0X	Простий, надійний, недорогий	Складніші у налаштуванні, дорожчі
Привід	SG90	Кроковий двигун NEMA 17	Простота інтеграції, ШИМ	Потрібен драйвер, більша вага
Камера	RPi Cam Module 3	USB веб- камера	Повна сумісність, IR-режим	Нестабільність, низька якість

Крім того, важливо враховувати енергоефективність. Raspberry Pi 4 споживає в середньому 2-3 Вт під навантаженням, що робить систему придатною для цілодобового використання без значних витрат на електроенергію.

У разі відключення живлення може бути додано резервне джерело (наприклад, повербанк на 10 000 мА·г), якого вистачить на 4-5 годин роботи. Це критично для користувачів, які надовго залишають тварину без нагляду.

Ще однією перевагою системи є її локальність: усі обчислення виконуються на самому Raspberry Pi, без потреби постійного підключення до хмарних сервісів. Це підвищує конфіденційність даних користувача та знижує залежність від інтернет-з'єднання.

Для користувачів, що турбуються про безпеку, важливо також, що вебінтерфейс можна легко захистити двофакторною автентифікацією або VPN-доступом.

Крім того, система може функціонувати автономно навіть при тимчасовій відсутності інтернету, що є важливою перевагою в умовах нестабільного зв'язку.

### 3.5 Тестування та налагодження системи

Тестування та налагодження системи є завершальним етапом реалізації, спрямованим на перевірку коректності роботи апаратних і програмних компонентів, а також забезпечення стабільності, надійності та зручності використання. Тестування охоплює перевірку функціональності модулів (годування, відеоспостереження, виявлення руху, віддалене керування), а налагодження усуває виявлені помилки.

Тестування проводиться в реальних умовах із моделюванням типових сценаріїв використання, таких як виявлення тварини, виконання розкладу годування та віддалений доступ.

Виділяють три основні етапи тестування: модульне, інтеграційне та системне тестування.

Модульне тестування передбачає перевірку окремих компонентів (датчик, сервомотор, камера, вебінтерфейс, MQTT) для оцінки їхньої коректності.

Інтеграційне тестування перевіряє взаємодію апаратних і програмних компонентів у єдиній системі.

Оцінку роботи системи в цілому, включаючи стабільність при тривалій експлуатації (24 години) і зручність для користувача виконує системне тестування.

Тестування виконується за допомогою редактора Thonny для покрокового виконання Python-скриптів, утиліт Raspberry Pi OS і логів, створених модулем logging. Налагодження передбачає аналіз лог-файлів виправлення коду та перевірку апаратних з'єднань мультиметром для усунення помилок.

Перевірка зчитування сигналу з GPIO-піна 23 через бібліотеку RPi.GPIO. Тестовий сценарій: імітація руху перед датчиком (наприклад, рукою) для генерації сигналу "1". Скрипт виводить стан піна в консоль Thonny і записує час події в лог. Виявлена проблема: помилкові спрацьовування через високу чутливість. Налагодження: зменшення чутливості потенціометром і додавання програмної фільтрації (ігнорування сигналів тривалістю менше 0.5 с).

					КВРКІ.210105.21.01.24 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

Перевірка активації через GPIO-пін 18 із ШІМ-сигналом за допомогою RPi.GPIO. Тестовий сценарій: поворот на 90° для видачі порції їжі (50 г) і повернення в початкове положення. Перевірено точність дозування і затримку між активаціями (2с). Виявлена проблема: нестабільний поворот через слабе живлення. Налагодження: перевірка блоку живлення (5В, 3А) і заміна дротів на макетній платі для кращого контакту.

Перевірка запису відео (1080р, 10 с) і потокової передачі через бібліотеку `picamera2`. Тестовий сценарій: запуск запису при сигналі від датчика і передача потоку на вебінтерфейс. Перевірено якість зображення і збереження файлів на `microSD`. Виявлена проблема: затримка потоку (>1 с) при високій роздільній здатності. Налагодження: зниження роздільної здатності до 720р для потоку і оптимізація скрипта через асинхронну обробку (бібліотека `asyncio`).

Перевірка доступу до вебінтерфейсу через браузер. Тестовий сценарій: перегляд відеопотоку, зміна розкладу годування (8:00, 18:00) і ручна активація сервомотора. Перевірено автентифікацію (логін/пароль). Виявлена проблема: повільне завантаження сторінки при слабкому Wi-Fi. Налагодження: оптимізація розміру відеопотоку і використання Wi-Fi 5 ГГц.

Перевірка надсилання сповіщень і команд через бібліотеку `raho-mqtt` із використанням публічного брокера Mosquitto. Тестовий сценарій: надсилання повідомлення про годування і отримання команди для активації сервомотора. Перевірено затримку (<1 с). Виявлена проблема: втрата повідомлень при нестабільному з'єднанні. Налагодження: додавання повторних спроб підключення в скрипті і перевірка Wi-Fi-з'єднання.

Інтеграційне тестування перевіряє взаємодію компонентів у типовому сценарії: тварина наближається до годівниці, датчик руху активує сервомотор і камеру, система надсилає сповіщення через MQTT і оновлює вебінтерфейс.

Тестовий сценарій розгортається так: спершу датчик руху HC-SR501 фіксує рух, надсилаючи відповідний сигнал. У відповідь на це скрипт запускає сервомотор SG90, який повертається на 90 градусів, і одночасно починається 10-секундний

					КВРКІ.210105.21.01.24 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

відеозапис з камери. Після завершення запису, ключові дані, такі як час події та фотознімок, передаються через MQTT і стають доступними для перегляду у вебінтерфейсі.

Виявлена проблема являє собою несинхронну активацію камери і сервомотора через високе навантаження на процесор. Налагодження передбачає використання асупсіо для асинхронної обробки подій, що зменшило затримку до 0.3 с. Перевірено стабільність з'єднань на макетній платі, замінено слабкий дріт для датчика.

Для системного тестування було розроблено детальний сценарій, який імітує типові умови використання системи. Він включає перевірку коректності спрацьовування механізму подачі корму згідно з встановленим розкладом, зокрема о 8:00 та 18:00. Окрім цього, сценарій передбачає імітацію активності тварини шляхом фіксації 10 випадків виявлення руху протягом доби, що дозволяє оцінити ефективність роботи датчика руху та пов'язаних з ним функцій. Важливим аспектом є також моніторинг безперервності та якості відеопотоку, який постійно транслюється через вебінтерфейс, що є ключовим для віддаленого спостереження. Нарешті, сценарій охоплює перевірку функціональності обміну даними через протокол MQTT, включаючи надсилання 20 повідомлень, які містять як сповіщення про події, так і команди управління системою. Цей комплексний підхід до системного тестування дає змогу виявити потенційні проблеми взаємодії компонентів та переконатися у загальній працездатності системи перед її впровадженням.

Перевірено стабільність (відсутність перезавантажень), температуру Raspberry Pi 4 (максимум 55°C), споживання пам'яті microSD (менше 10 ГБ за добу) і зручність вебінтерфейсу. Виявлена проблема: перегрів при тривалій потоковій передачі. Налагодження: додавання пасивного охолодження (алюмінієвий радіатор) і зниження частоти кадрів потоку до 15 кадрів/с.

Результати тестування і налагодження наведено в таблиці 3.1.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.2 – Результати тестування та налагодження системи

Компонент/Функція	Тестовий сценарій	Результат	Виявлені проблеми	Налагодження
Датчик руху HC-SR501	Виявлення руху (10 разів)	Сигнал коректний (0.3 с затримка)	Помилкові спрацьовування	Зменшення чутливості, фільтрація сигналів
Сервомотор SG90	Поворот на 90° (10 циклів)	Точність дозування 50 г	Нестабільний поворот	Перевірка живлення, заміна дротів
Камера Module 3	Запис (10 с) і потік (720p)	Якість зображення висока	Затримка потоку (>1 с)	Зниження роздільної здатності, asyncio
Вебінтерфейс (Flask)	Доступ, зміна розкладу, керування	Доступний, затримка <0.5 с	Повільне завантаження сторінки	Оптимізація потоку, Wi-Fi 5 ГГц
MQTT-інтеграція	Надсилання/отримання 20 повідомлень	Затримка <1 с	Втрата повідомлень	Повторні спроби підключення
Система загалом	Робота 24 години (10 годувань, 10 рухів)	Стабільна, температура <55°C	Перегрів при потоковій передачі	Радіатор, зниження частоти кадрів

Тестування та налагодження підтвердили коректну роботу системи, включаючи датчик руху, сервомотор, камеру, вебінтерфейс і MQTT-інтеграцію. Модульне, інтеграційне та системне тестування дозволили виявити та усунути проблеми, такі як помилкові спрацьовування датчика, затримка відео і перегрів, шляхом оптимізації коду, апаратних з'єднань і додавання радіатора. Система є стабільною, економічно вигідною і готовою до експлуатації, забезпечуючи автоматизований догляд за тваринами з можливістю віддаленого керування, що робить її конкурентоспроможною альтернативою комерційним рішенням.

### 3.6 Висновки до третього розділу

Третій розділ присвячено реалізації системи, охоплюючи розробку схем, системного програмного забезпечення, апаратних рішень, інтеграцію датчиків і модулів, а також тестування та налагодження.

Системне програмне забезпечення, створене на Python у середовищі Raspberry Pi OS, включає модулі для керування датчиком, сервомотором, камерою, вебінтерфейсом і MQTT-інтеграцією, забезпечуючи автоматизацію годування, відеоспостереження та віддалене керування. Використання бібліотек RPi.GPIO, picamera2, Flask і paho-mqtt дозволило реалізувати гнучке і стабільне рішення. Апаратні рішення підтвердили їхню сумісність і надійність завдяки ретельній конфігурації в Raspberry Pi OS.

Інтеграція датчиків і модулів забезпечує безперебійну взаємодію через фізичні з'єднання і програмні модулі, дозволяючи системі реагувати на рух тварини, видавати їжу та передавати дані користувачу. Тестування (модульне, інтеграційне, системне) виявило і усунуло проблеми, такі як помилкові спрацьовування датчика, затримка відеопотоку та перегрів, шляхом оптимізації коду, додавання радіатора і перевірки з'єднань. Система продемонструвала стабільність при 24-годинній роботі, підтвердивши її готовність до експлуатації.

Модульна структура дозволяє розширювати функціонал, наприклад, додаванням датчика температури, що підвищує адаптивність.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Розробка системи догляду за домашніми тваринами на базі мікрокомп'ютера Raspberry Pi 4 дозволила створити економічно вигідне, модульне та функціональне рішення, яке забезпечує автоматизацію ключових процесів догляду, таких як годування, відеоспостереження та віддалене керування. Проведений аналіз предметної області підтвердив актуальність теми, зумовлену зростанням кількості власників тварин і потребою в ефективних системах догляду в умовах сучасного ритму життя. Порівняння існуючих рішень показало переваги саморобних систем на базі Raspberry Pi, зокрема їхню гнучкість і низьку вартість порівняно з комерційними аналогами, такими як Petnet SmartFeeder.

Raspberry Pi 4 обрано як оптимальну платформу завдяки високій продуктивності, підтримці IoT-технологій, великій кількості GPIO-пінів. Апаратні компоненти, включаючи камеру Raspberry Pi Camera Module 3, сервомотор SG90 і датчик руху HC-SR501, забезпечують базовий функціонал системи за загальної вартості близько 100 дол. США, що значно нижче за комерційні рішення. Системне програмне забезпечення, розроблене на Python у середовищі Raspberry Pi OS із використанням бібліотек RPi.GPIO, picamera2, Flask і paho-mqtt, реалізує автоматизацію, віддалений доступ через вебінтерфейс і MQTT-інтеграцію.

Розроблені схеми підключення компонентів гарантують надійну взаємодію, а конфігурація апаратного забезпечення в Raspberry Pi OS забезпечує стабільність. Інтеграція датчиків і модулів дозволила створити єдиний комплекс, який реагує на присутність тварини, видає їжу за розкладом і передає дані користувачу в реальному часі. Тестування системи (модульне, інтеграційне, системне) підтвердило її надійність, усунувши проблеми, такі як помилкові спрацьовування датчика, затримка відеопотоку та перегрів, шляхом оптимізації коду, апаратних з'єднань і додавання радіатора.

Система є модульною, що дозволяє розширювати її функціонал, наприклад, додаванням датчиків температури чи вологості, без значних витрат. Економічна

					КВРКІ.210105.21.01.24 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

вигода, забезпечена використанням відкритих технологій і доступних компонентів, робить систему привабливою для широкого кола користувачів. Результати роботи підтверджують її практичну цінність як для власників домашніх тварин, так і для освітніх цілей у сфері IoT і мікрокомп'ютерних систем. Система може бути адаптована до різних типів тварин і умов, а її відкрита архітектура сприяє подальшим удосконаленням, що робить її конкурентоспроможною альтернативою комерційним продуктам.

					КВРКІ.210105.21.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Arduino and Raspberry Pi Powered Pet Monitoring System. Instructables. URL: <https://www.instructables.com/Arduino-and-Raspberry-Pi-Powered-Pet-Monitoring-Sy/>
2. Chen R. C., Saravananarajan V. S., Hung H. T. Monitoring the behaviours of pet cat based on YOLO model and raspberry Pi. *Int. J. Appl. Sci. Eng.* 2021.Vol. 18. P.1-12.
3. Abisha D., Vijayasri M., Sharmila M., Ida Christy J., Navedha Evanjalin R. Enhancing Pet Care with IoT: The Development of a Smart Pet Feeder. *In 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*. 2024. p. 374-379.
4. Bembde M., Ranjan N. M., Kamble P., Chavan A., Yelmar A., Mane R. Robotic Day-Care for Pets using Sensors and Raspberry Pi. *In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. 2023. p. 994-999.
5. Joice A., Tufaique T., Tazeen H., Igathinathane C., Zhang Z., Whippo C., Archer D. Applications of Raspberry Pi for Precision Agriculture. A Systematic Review. *Agriculture*. 2025.Vol. 15(3). P. 227.
6. Patil H., Ansari N. Smart surveillance and animal care system using IOT and deep learning. *In Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)*. 2020.
7. Raavi P., Panchangam H. (2020). Camera Surveillance Using Raspberry Pi. *International Research Journal of Engineering and Technology (IRJET)*. Vol. 7(05).
8. Ramdani D., Adi P. D. P., Adiprabowo T., Wahyu Y., Satyawan A. S., Sari S. O., Rohman N. Development of a Smart Water Dispenser Based on Object Recognition with Raspberry Pi 4. *International Journal of Advanced Computer Science & Applications*. 2024. Vol. 15(12).
9. Kumar G. K., Thilak K. R., Aswin E. S., Ranjith R., Parthasarathy J. Automated Food Dispenser with Minimal Interaction with IoT Technology. *In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. 2024. p. 83-86.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

10. Vijayalakshmi B., Varsheni S., Ganesh T. S. Design and Implementation of an AI-Enhanced Automatic Animal Food Dispenser. *In 2024 10th International Conference on Communication and Signal Processing (ICCSP)*. 2024. p.1241-1246.

11. Vrishanka P. N., Prabhakar P., Shet D., Rupali, K. Automated pet feeder using IoT. *In 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*. p. 1-5.

12. Lee J. J., Kim D. H. Implementation of a Smart IoT System with Automatic Pet Feeder. *Journal of Digital Contents Society*. 2021. Vol. 22(2), p.209-214.

13. Goo P. C., Tay K. G., Chew C. C., Huong A. Smart Pet House. *In 2024 International Conference on Future Technologies for Smart Society (ICFTSS)*. 2024. p. 37-42.

14. Chen Y., Elshakankiri M. Implementation of an IoT based pet care system. *In 2020 fifth international conference on fog and Mobile edge computing (FMEC)*. 2020. p. 256-262.

15. Jolles, J. W. Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*. 2021. Vol. 12(9). P.1562-1579.

16. Rowlison de Ortiz, A., Belda B., Hash J., Enomoto M., Robertson J., Lascelles B. D. X. Initial exploration of the discriminatory ability of the PetPace collar to detect differences in activity and physiological variables between healthy and osteoarthritic dogs. *Frontiers in Pain Research*. 2022. Vol.3.

17. Kandepan D., Venkatesan D. K. A Novel Approach to Diagnose the Animal Health Continuous Monitoring Using IoT Based Sensory Data. *Instrumentation, Mesures, Métrologies*. 2022. Vol. 21(5).

18. NAVEENKUMAR D. An Intelligent Animal Care and Management System Based on the Internet of Things and AI. *JOURNAL OF COMPUTER SCIENCE*. 2022. Vol.15(10).

19. N. Saidatin, S. Nurmuslimah, and P. Bagus. A Design Remote Control System to Feed Birds Using ESP8266. *International Journal of Recent Technology and Applied Science*. 2020. vol. 2(2). p. 81-90.

					КВРКІ.210105.21.01.24 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

20. Shivamadhu G. D., Venkatesh A., Alva A., Nausheer D. PortaX Secure Automation System Using IoT: A Survey. *International Journal of Recent Technology and Applied Science (IJORTAS)*. 2020. Vol. 2(1). P.66-74.

21. Yavor K. M., Lehmann A., Finkbeiner M. Environmental impacts of a pet dog: an LCA case study. *Sustainability*. 2020. Vol.12(8). P. 3394.

22. Sautière P.E., Blervacq, A.S, Vizioli J. Production and Uses of e-Learning Tools for Animal Biology Education at University. *Eur. Zool. J.* 2022. P.63–78.

23. Dickey M.D. Murder on Grimm Isle: The Impact of Game Narrative Design in an Educational Game-Based Learning Environment. *Br. J. Educ. Technol.* 2021.p.456–469.

24. Begum A., Das S., Saikia T., Choudhury N., Mandal R., Barman J. K. An IoT-Based Animal Health Monitoring System. *In International Conference On Innovative Computing And Communication*. 2024. p. 273-287.

25. Li J., Wu C. H. Determinants of learners' self-directed learning and online learning attitudes in online learning. *Sustainability*. 2023. Vol. 15(12). p. 9381.

26. Chandrika K., Dedeepya C., PushpaLatha G., Kumar G. K. Care A Pet-Android Pet Care Application. *In International Conference on Information and Management Engineering*. 2022. p.319-328.

27. Haste I., Pimple J., Thakare T., Gadigone K., Kohle V. Efficient Framework for Pet Adoption and Recommendation System. *In 2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science*. 2022. p.1-4.

28. Bhatia S., Guerrero B. M., Paitao N., Intal G. L. D. Vetpaw Pet Care Application: An assessment of the proposed business application using SWOT analysis. *In Proceedings of the 4th International Conference on Management Science and Industrial Engineering*. 2022. p. 399-406.

29. Taivalsaari, T. Mikkonen. On the development of IoT systems. *Third International Conference on Fog and Mobile Edge Computing (FMEC)*. 2023. p.13-19.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

30. Choi M. Method for Developing a Digital Twin System for Pet Plant Care Services. *International Journal of Internet, Broadcasting and Communication*. 2025. Vol. 17(1). P.15-25.

31. Lee W. Q., Lau P. Y. PetCare: a real-time pet monitoring system with food dispensing using Raspberry Pi. In International Workshop on Advanced ImagingTechnology. 2022. Vol. 12177. p. 212-217.

32. Li, T. R., Tsai, C. Y. Design and Validation of Pet Care Teaching System Based on Augmented Reality. *Electronics*. 2025. Vol. 14(7). p.1271.

33. Rey W. P., Rey K. W. J. D. AZKALS: Revolutionizing pet care with an IoT-powered smart dog cage management system. In 2024 3rd International Conference on Computer Technologies (ICCTech). 2024. p. 48-54.

34. Devi M. R., Jyothi V., Nagajyothi D. IoT and Cloud-based Automated Pet Care System. In 2022 6th International Conference on Electronics, Communication and Aerospace Technology. 2022. p.1366-1372.

35. Lambert A. A., Jk G., Krishnan H. Design and Implementation of a Pet Care and Tracking System. In 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). 2023. p. 1-5.

36. Mukhlis I. R., Hermansyah D., Hariyanti T. A. Model View Controller Method For Animal Care (Petcare) Information System At Niz Petcare Lawang. *JEECS (Journal of Electrical Engineering and Computer Sciences)*. 2023. Vol.8(2). P.103-122.

37. Aravind G., Jeeva S., Selvaganesh D., SP, S. AI-Enabled Safe and Scalable Pet Care: An Intelligent System for Real Time Health Monitoring and Care. In 2025 International Conference on Intelligent Computing and Control Systems (ICICCS). 2025. p.1530-1535.

38. Yeow J. R., Kamaludin, H. Pet Care Management System. *Applied Information Technology And Computer Science*. 2023. Vol. 4(2). P.2019-2038.

39. Aurellia I. A. D. Design and Build a Mobile-Based Pet Care Information System with Personal Extreme Programming Method. In *Proceedings of the First*

*Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science). 2023. Vol. 102. p. 89.*

40. Hasni H. C., Suhaimi S. Pet Feeding System. *International Journal of Recent Technology and Applied Science (IJORTAS)*. 2022. Vol.4(2). P.111-118.

41. Shultz M., Rouff C., Tekeoglu A. Autonomic IoT Pet Health Monitoring System. *In 2024 International Symposium on Networks, Computers and Communications (ISNCC)*. 2024. p. 1-6.

42. Bahri P. M. N. A. S., Nor D. M. Smart Feeder Aquarium and Water Quality Monitoring using Raspberry PI and Blynk. *Evolution in Electrical and Electronic Engineering*. 2024. Vol. 5(2). P.190-198.

43. Anggraini N., Rahman D. F., Wardhani L. K., Hakiem N. Mobile-based monitoring system for an automatic cat feeder using Raspberry Pi. *Telkomnika (Telecommunication Computing Electronics and Control)*. 2020. Vol.18(2). p.1038-1046.

44. Sujatha V., Ediga R., Yasmeen S. N., Baseer K. K., Vijayanand S., Neerugatti V. Automation in Pet Care: An IoT Approach to Smart Pet Feeding Systems. *In 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)*. 2024. p. 1087-1091.

45. Kim H., Kang H., Kim S., Choi D., You J., Smith A., Lee M., O. Petification: Node-RED Based Pet Care IoT Solution Using MQTT Broker. *In 2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. 2022. p. 25-29.

46. Tung Y. C., Lien Y. H., Liu L. W., Devi P. K. IOT-Smart Monitoring of Pet Housing. *In 2024 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. 2024. p. 1-6.

47. Ghute M., Deshpande S., Sondavle A., Bhalerao S., Deshmukh M. IoT Based Pet Day-Care Robot. *In 2022 6th International Conference on Electronics, Communication and Aerospace Technology*. 2022. p. 546-548.

					КВРКІ.210105.21.01.24 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

48. Zainuddin A. A., Mahazir M. N. B., Adanan M. A. A., Tamrin M. I. M., Zulkfli M. A. H., Samsudin M. H. A. Design and Implementation of a Mobile-Controlled IoT Smart Pet Feeder for Busy Pet Owners. *Malaysian Journal of Science and Advanced Technology*. 2024. P.498-505.

49. Quiñonez Y., Lizarraga C., Aguayo R., Arredondo D. Communication architecture based on IoT technology to control and monitor pets feeding. *J. Univers. Comput. Sci.* 2021. Vol.27(2). P.190-207.

50. Angela S., Tiomar J., Sudijanto J. A., Widiyanto M. H. IoT-Enabled Systems for Automated Cat Care: A Comprehensive Literature Review. *In 2024 6th International Conference on Cybernetics and Intelligent System (ICORIS)*. 2024. p. 1-6.

51. Бойчук Р.І. Системи годування тварин на базі мікрокомп'ютера. Магістерська дисертація. 2019. С. 13. URL: <https://ela.kpi.ua/server/api/core/bitstreams/ebd89413-3c3f-4fb1-a6a8-9ae47c685b0d/content>(дата звернення: 15.04.2025).

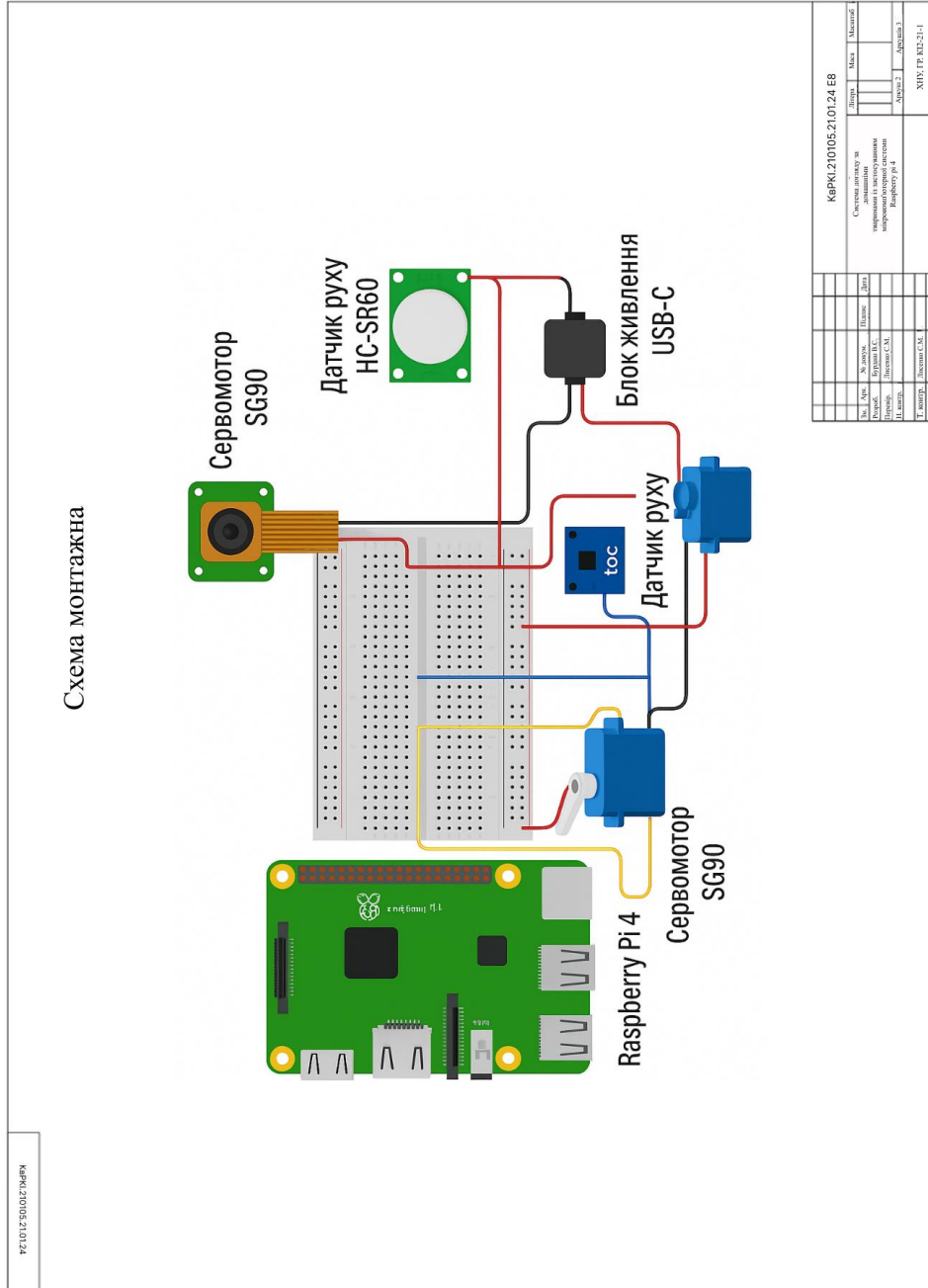
52. The Pet Historian. Faced with an automatic pet feeder....1939. URL: <https://thepethistorian.com/2015/04/25/faced-with-an-automatic-pet-feeder-1939/> (дата звернення: 15.04.2025).

					КВРКІ.210105.21.01.24 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		



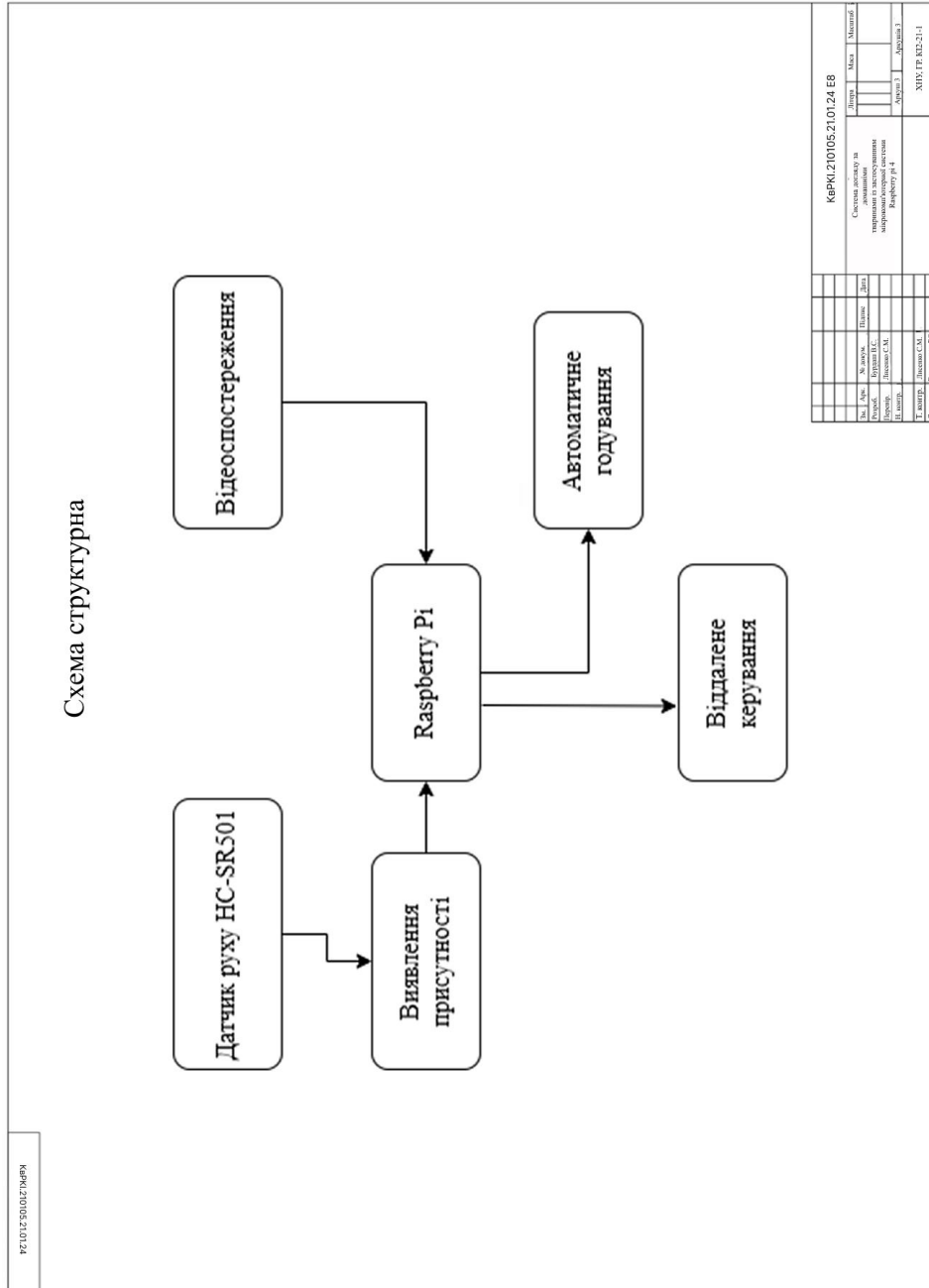
## Додаток Б (обов'язковий)

### КОПІЯ КРЕСЛЕННЯ «СХЕМА МОНТАЖНА»



Додаток В  
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «СХЕМА СТРУКТУРНА»



## Додаток Г (обов'язковий)

### ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
import RPi.GPIO as GPIO
import picamera2
from flask import Flask, Response, render_template, request
import time
import logging
from datetime import datetime
import json
import threading
import asyncio
import requests
import cv2
import os

# Налаштування логування
logging.basicConfig(
    filename='/var/log/petcare.log',
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s'
)

# Налаштування GPIO
GPIO.setmode(GPIO.BCM)
MOTION_SENSOR_PIN = 23
SERVO_PIN = 18
GPIO.setup(MOTION_SENSOR_PIN, GPIO.IN)
GPIO.setup(SERVO_PIN, GPIO.OUT)

# Налаштування ШІМ для сервомотора (50 Гц)
servo = GPIO.PWM(SERVO_PIN, 50)
servo.start(0)

# Налаштування камери
camera = picamera2.Picamera2()
camera_config = camera.create_video_configuration(main={"size":
(1280, 720)})
camera.configure(camera_config)

# Налаштування Flask
app = Flask(__name__)

# Конфігураційний файл
CONFIG_FILE = '/home/pi/petcare_config.json'
DEFAULT_CONFIG = {
    "feeding_schedule": ["08:00", "18:00"],
    "telegram_token": "",
    "telegram_chat_id": "",
```

```

    "video_duration": 10,
    "portion_size": 50 # грами
}

# Завантаження або створення конфігурації
def load_config():
    try:
        if os.path.exists(CONFIG_FILE):
            with open(CONFIG_FILE, 'r') as f:
                return json.load(f)
        else:
            with open(CONFIG_FILE, 'w') as f:
                json.dump(DEFAULT_CONFIG, f, indent=4)
            return DEFAULT_CONFIG
    except Exception as e:
        logging.error(f"Помилка завантаження конфігурації: {e}")
        return DEFAULT_CONFIG

config = load_config()

# Функція для активації сервомотора
def activate_servo():
    try:
        servo.ChangeDutyCycle(7.5) # Поворот на 90°
        time.sleep(0.5)
        servo.ChangeDutyCycle(2.5) # Повернення
        time.sleep(0.5)
        servo.ChangeDutyCycle(0)
        logging.info(f"Сервомотор активовано: видано
{config['portion_size']} г їжі")
        send_telegram_notification(f"Годівниця активована: видано
{config['portion_size']} г їжі")
    except Exception as e:
        logging.error(f"Помилка сервомотора: {e}")

# Функція для запису відео
def record_video(duration=None):
    try:
        duration = duration or config['video_duration']
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        video_path = f"/home/pi/videos/petcare_{timestamp}.mp4"
        os.makedirs(os.path.dirname(video_path), exist_ok=True)
        camera.start_recording(video_path)
        time.sleep(duration)
        camera.stop_recording()
        logging.info(f"Відео записано: {video_path}")
        send_telegram_notification(f"Записано відео: {video_path}")
        return video_path
    except Exception as e:
        logging.error(f"Помилка запису відео: {e}")
        return None

```

```

# Функція для надсилання сповіщень у Telegram
def send_telegram_notification(message):
    try:
        token = config['telegram_token']
        chat_id = config['telegram_chat_id']
        if token and chat_id:
            url = f"https://api.telegram.org/bot{token}/sendMessage"
            payload = {"chat_id": chat_id, "text": message}
            response = requests.post(url, json=payload)
            if response.status_code == 200:
                logging.info("Сповіщення надіслано в Telegram")
            else:
                logging.error(f"Помилка Telegram: {response.text}")
    except Exception as e:
        logging.error(f"Помилка надсилання сповіщення: {e}")

# Функція для потокової передачі відео
def generate_video_stream():
    while True:
        try:
            frame = camera.capture_array()
            ret, jpeg = cv2.imencode('.jpg', frame)
            if ret:
                yield (b'--frame\r\n'
                       b'Content-Type: image/jpeg\r\n\r\n'
                       jpeg.tobytes() + b'\r\n')
            time.sleep(0.1)
        except Exception as e:
            logging.error(f"Помилка потокової передачі: {e}")
            time.sleep(1)

# Функція перевірки розкладу годування
def check_feeding_schedule():
    while True:
        try:
            current_time = datetime.now().strftime("%H:%M")
            if current_time in config['feeding_schedule']:
                logging.info(f"Заплановано годування {current_time}")
                activate_servo()
                record_video()
                time.sleep(60) # Затримка для уникнення повторного
                                годування
            time.sleep(10)
        except Exception as e:
            logging.error(f"Помилка перевірки розкладу: {e}")
            time.sleep(60)

# Вебінтерфейс
@app.route('/')
def index():
    return render_template_string('')

```

```

<!DOCTYPE html>
<html>
<head>
  <title>Система догляду за тваринами</title>
  <style>
    body { font-family: Arial; text-align: center; }
    .container { max-width: 800px; margin: auto; }
    img { width: 640px; height: 360px; }
    button { padding: 10px 20px; margin: 10px; }
  </style>
</head>
<body>
  <div class="container">
    <h1>Система догляду за тваринами</h1>
    
    <form method="post" action="/feed">
      <button type="submit">Годувати</button>
    </form>
    <form method="post" action="/update_schedule">
      <label>Розклад годування (чч:мм, через
кому):</label><br>
      <input type="text" name="schedule" value="{{
','.join(config['feeding_schedule']) }}">
      <button type="submit">Оновити</button>
    </form>
    <p>Поточний розклад: {{ config['feeding_schedule']
}}</p>
  </div>
</body>
</html>
''' , config=config)

@app.route('/video_feed')
def video_feed():
    return Response(generate_video_stream(),
                    mimetype='multipart/x-mixed-replace;
boundary=frame')

@app.route('/feed', methods=['POST'])
def feed():
    activate_servo()
    return "Годівниця активована"

@app.route('/update_schedule', methods=['POST'])
def update_schedule():
    try:
        new_schedule = request.form['schedule'].split(',')
        new_schedule = [t.strip() for t in new_schedule if t.strip()]
        config['feeding_schedule'] = new_schedule
        with open(CONFIG_FILE, 'w') as f:
            json.dump(config, f, indent=4)
        logging.info(f"Розклад оновлено: {new_schedule}")

```

```

        return "Розклад оновлено"
    except Exception as e:
        logging.error(f"Помилка оновлення розкладу: {e}")
        return "Помилка оновлення"

# Основна функція моніторингу
async def monitor_motion():
    try:
        while True:
            if GPIO.input(MOTION_SENSOR_PIN):
                logging.info("Виявлено рух")
                activate_servo()
                record_video()
                await asyncio.sleep(2)      # Затримка для уникнення
повторів
                await asyncio.sleep(0.1)
    except Exception as e:
        logging.error(f"Помилка моніторингу: {e}")
    finally:
        servo.stop()
        GPIO.cleanup()
        camera.close()

# Запуск програми
if __name__ == '__main__':
    try:
        # Запуск Flask
        flask_thread = threading.Thread(target=lambda:
app.run(host='0.0.0.0', port=5000))
        flask_thread.daemon = True
        flask_thread.start()

        # Запуск розкладу годування
        schedule_thread =
threading.Thread(target=check_feeding_schedule)
        schedule_thread.daemon = True
        schedule_thread.start()

        # Запуск моніторингу датчика
        asyncio.run(monitor_motion())
    except KeyboardInterrupt:
        logging.info("Програма зупинена")
    except Exception as e:
        logging.e

```

## Протокол аналізу звіту подібності експертом

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

**Автор:** Владислав БУРДАШ

**Співавтор:**

**Назва:** Бурдаш\_Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry pi 4

**Експерт:**

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

**Коефіцієнт подібності 1:** 2.4%

**Коефіцієнт подібності 2:** 0.6%

**Мікропробіли:** 52

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-02 23:03:31.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обґрунтування:

2025-06-03

Дата



Доцент Андрій Нічепорук

експерт

Mon Jan 02 16:21:30 EEST 2023, Мегалитий Дмитро Миколайович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 18.00%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 10%

ID: 242962		Title: БКР Система догляду за домашніми тваринами із застосуванням мікроконтролерної системи Розрешту р1 4		Document		Sum coincidence on the DB	
Added in a DB: 2025-06-02		Authors: Владислав БУРДАШ		Symbols	Lexemes	Symbols	Lexemes
Heads: Сергій ЛИСЕНКО		Consultants:		113781	863	208334 (18%)	157 (18%)
Opponents:							

ID	Description	Plagiarism sources	
		Symbols	Lexemes
240778	Title: Звіт з ПШТ Система догляду за домашніми тваринами із застосуванням мікроконтролерної системи Розрешту Р1 Added in a DB: 2025-05-02 Authors: В.С. Бурдаша Heads: Лисенко С.М. Consultants: Opponents:	20031 (18.00%)	151 (17.09%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Бурдаш Владислав Сергійович

Тема: Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі здійснено комплексне дослідження предметної області, проаналізовано існуючі рішення у сфері автоматизованого догляду за тваринами, розглянуто сучасні технології (IoT, відеоспостереження, автоматизоване годування, віддалене керування), а також апаратні платформи. Проведено порівняльний аналіз готових комерційних і саморобних рішень, обґрунтовано актуальність теми в контексті сучасного способу життя. Сформульовано мету, об'єкт, предмет і основні завдання дослідження. Обґрунтовано використання Raspberry Pi 4 як базової платформи для створення системи. У роботі використано системний, аналітичний та порівняльний підходи до вивчення проблеми. У другому розділі виконано технічне проектування системи. Обґрунтовано вибір апаратних і програмних компонентів: мікрокомп'ютера Raspberry Pi 4, камери, сервомотора, сенсорів руху, температури та вологості. Проведено огляд та вибір бібліотек і середовищ програмування, які відповідають сучасним підходам у розробці IoT-систем. Створено архітектуру системи, розроблено структурну та функціональну схеми, реалізовано логіку роботи окремих модулів, забезпечено

взаємодію між апаратними і програмними частинами. Робота демонструє високий ступінь використання сучасних відкритих технологій та методів розробки мікрокіберфізичних систем. У третьому розділі виконано апаратну реалізацію проекту. Реалізовано системне програмне забезпечення, виконано інтеграцію усіх компонентів. Здійснено моделювання реальних сценаріїв роботи, проведено тестування на предмет стабільності функціонування, точності спрацювання датчиків, роботи годинниці та відеопотоку. Запроваджено вебінтерфейс для віддаленого доступу. Застосовані підходи відповідають сучасним тенденціям у розробці інтегрованих інтелектуальних систем на базі технологій IoT.

4. Позитивні сторони роботи: висока практична цінність, можливість реального використання та подальшої комерціалізації.

5. Негативні сторони роботи: недостатньо уваги приділено апаратному моделюванню цифрових схем, що могло б розширити технічну реалізацію системи.

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

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

8. Інші зауваження: \_\_\_\_\_


9. Оцінка дипломної роботи: добре

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

*Чесичи Віктор Миколайович*

*канд. техн. наук, доц., доцент кафедри кібербезпеки*

"03" 06 2025 р.

 (підпис)

Завідувачу кафедри КПС'  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Владислава БУРДАША

Шт. здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

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

09.06. 2025 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система догляду за домашніми тваринами із застосуванням мікрокомп'ютерної системи Raspberry Pi 4

Автор: Владислав БУРДАШ

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Сергій ЛИСЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

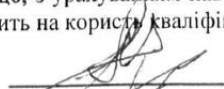
Підтвердження:

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

- 1) запозичення у роботі розміщені переважно в розділах аналізу існуючих рішень та технологій, де здійснюється огляд аналогів і комерційних прототипів, що не описують авторське дослідження і не впливають на унікальність власних результатів;
- 2) усі запозичення є фрагментарними, супроводжені належним чином оформленими посиланнями на джерела;
- 3) окремі виявлені збіги стосуються загальноживаних технічних термінів, які система ідентифікує як подібні до десятків інших джерел, що вказує на їх публічний характер;
- 4) частина збігів стосується типових цифрових послідовностей або кодів, використаних у прикладах роботи з мікроконтролерами та сенсорами, які не є об'єктами авторського права;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 2.4% і адресується до 29 першоджерел; та системою Anti-Plagiarism складає 18%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Сергій ЛИСЕНКО

Гарант ОП



Андрій НІЧЕПОРУК

Завідувач кафедри КІС



Ольга ПАВЛОВА