НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки
(повне найменування інституту, факультету)

Кафедра комп'ютеризованих систем управління
(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри

_____

«___» _____ 2024 р.

Пояснювальна записка

до кваліфікаційної роботи
на здобуття другого (магістерського) рівня вищої освіти

за спеціальністю 174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»
ОПП - «Комп'ютеризовані системи управління та автоматика»

на тему: Система автоматизованого керування абіотичними факторами у птахівничому комплексі

Виконав студент **6** курсу, **6341м** групи
Топольник Д.В.
(прізвище та ініціали)
Керівник **Герасін О. С.**
(прізвище та ініціали)

Миколаїв - 2024

Національний університет кораблебудування імені адмірала Макарова

ННІ	автоматики та електротехніки
Кафедра	комп'ютеризованих систем управління
Рівень вищої освіти	**другий (магістерський)**
Спеціальність	**174– «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»**
ОПП	**«Комп'ютеризовані системи управління і автоматика»**

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

Черно О.О.

«____» _____2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Топольник Дарія Віталіївна

(прізвище, ім'я, по батькові)

1. Тема роботи: *система автоматизованого керування абіотичними факторами у птахівничому комплексі*

керівник роботи Герасін Олександр Сергійович, к.т.н., доцент

затверджені наказом вищого навчального закладу від "14" жовтня 2024 р. № 1075-уч

2. Строк подання роботи *18.12.2024 р.*

3. Вихідні дані до роботи *при виконанні роботи задіяти плату Rassbery Pi, розробити додаток до смартфону, передбачити засоби машинного навчання для визначення пожеж у курнику.*

4. Мета дослідження *розробка системи автоматизованого керування абіотичними факторами у птахівничому комплексі*

5. Зміст розрахунково-пояснювальної записки (основні задачі дослідження)

Огляд сучасних технічних рішень у автоматизованих системах для керування абіотичними факторами

Проєктування та розробка технічних рішень для автоматизованого управління птахокомплексом

Розробка та випробування програмних рішень для автоматизованого управління пташником

Модель машинного навчання для виявлення пожеж у птахівництві на основі аналізу абіотичного фактору

Охорона праці та навколишнього середовища.

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Актуальність та постановка задач розробки

Система автоматизованого керування абіотичними факторами у птахівничому комплексі

Алгоритми керування абіотичними факторами

Мобільний додаток для автоматизованого керування пташником

Модель машинного навчання для виявлення пожеж

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Охорона праці та навколишнього середовища	*Герасін О.С., доцент каф. КСУ*	25.11.2024	05.12.2024

Дата видачі завдання: «_02__» _____09_____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд сучасних технічних рішень у автоматизованих системах для керування абіотичними факторами.	15.10.2024	Виконано
2	Проєктування та розробка технічних рішень для автоматизованого управління птахокомплексом	04.11.2024	Виконано
3	Розробка та випробування програмних рішень для автоматизованого управління пташником.	18.11.2024	Виконано
4	Модель машинного навчання для виявлення пожеж у птахівництві на основі аналізу абіотичного фактору.	25.11.2024	Виконано
5	Охорона праці.	05.12.2024	Виконано
6	Оформлення пояснювальної записки.	11.12.2024	Виконано

Студент _____ Топольни Д.В.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Герасін О.С.

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота має загальний обсяг 112 сторінок, серед яких міститься 49 рисунків та 34 використаних джерела.

Робота присвячена актуальній задачі розробки автоматизованої системи керування абіотичними факторами в птахокомплексі. У процесі роботи було створено апаратно-програмний комплекс для моніторингу та управління мікрокліматом птахокомплексу, а також забезпечено інтеграцію між пристроями на основі IoT. У першому розділі проведено огляд сучасних технологій автоматизації в аграрному секторі, обґрунтовано вибір компонентів системи, таких як: сенсори, мікроконтролери, реле та дисплеї. У другому розділі представлено проектування та розробку технічних рішень для автоматизованого управління птахокомплексом. Описано вибір мережевого протоколу та методи передачі даних, використання мікрокомп'ютерів для керування системою, вибір сенсорів, а також аналіз використання аналогово-цифрових перетворювачів і дисплеїв на основі LED технології для моніторингу параметрів. У третьому розділі описано розробку апаратної частини системи, підключення сенсорів до мікрокомп'ютера, реалізацію MQTT-брокера для передачі даних та інтеграцію з сервером. Також, розглянуто створення мобільного додатку для моніторингу та управління параметрами мікроклімату. У додатку реалізовано інтерфейс для відображення даних з сенсорів у реальному часі, налаштування параметрів керування системою та отримання сповіщень. Четвертий розділ присвячено розробці моделі машинного навчання для виявлення абіотичного фактора вогню в птахокомплексі. У цьому ж розділі розглянуто математичні моделі, що описують модель CNN. У п’ятому розділі охоплено питання охорони праці, зокрема електробезпеку, захист від механічних і програмних загроз, а також безпеку IoT-рішень та охорону навколишнього середовища.

Ключові слова: АСК, мікроклімат, IoT, Raspberry Pi, MQTT, птахокомплекс, машинне навчання, Android-додаток.

ABSTRACT

The qualification thesis comprises a total of 112 pages, including 49 figures and 34 references.

The work addresses the pressing issue of developing an automated system for controlling abiotic factors in a poultry complex. During the research, a hardware-software complex was created for monitoring and managing the microclimate in the poultry complex, along with ensuring integration between devices based on IoT technologies.

The first chapter provides an overview of modern automation technologies in the agricultural sector, justifies the choice of system components such as sensors, microcontrollers, relays, and displays. The second chapter focuses on the design and development of technical solutions for automated poultry complex management. It includes the selection of a network protocol and data transmission methods, the use of microcomputers for system control, sensor selection, and the analysis of analog-to-digital converters and LED-based displays for parameter monitoring.

The third chapter describes the development of the system's hardware, including connecting sensors to the microcomputer, implementing an MQTT broker for data transmission, and server integration. It also discusses the creation of a mobile application for monitoring and controlling microclimate parameters. The application includes an interface for real-time sensor data visualization, system control parameter settings, and receiving notifications.

The fourth chapter is dedicated to developing a machine learning model for detecting fire as an abiotic factor in the poultry complex. This section examines mathematical models describing the CNN model. The fifth chapter covers occupational safety issues, including electrical safety, protection against mechanical and software threats, IoT security, and environmental protection.

Keywords: ACS, microclimate, IoT, Raspberry Pi, MQTT, poultry complex, machine learning, Android application.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ..... 8

ВСТУП 9

РОЗДІЛ 1 ОГЛЯД СУЧАСНИХ ТЕХНІЧНИХ РІШЕНЬ У АВТОМАТИЗОВАНИХ СИСТЕМАХ ДЛЯ КЕРУВАННЯ АБІОТИЧНИМИ ФАКТОРАМИ 10

1.1 Базові абіотичні фактори, які впливають на якість утримання птаства. 10

1.2 Пташникові системи автоматизованого керування..... 12

1.3 Існуючі системи для підтримки оптимальних умов для птахів. 20

Висновок до 1 розділу 23

РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ТЕХНІЧНИХ РІШЕНЬ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПТАХОКОМПЛЕКСОМ 25

2.1. Мережевий протокол передачі даних. 25

2.2 Мікрокомп’ютер для керування системою. 32

2.3 Вибір дисплея на основі LED технології..... 36

2.4 Вибір аналогових датчиків..... 38

2.5. Аналогово-цифровий перетворювач..... 42

2.6 Інші елементи системи. 45

Висновок до 2 розділу 49

РОЗДІЛ 3 РОЗРОБКА ТА ВИПРОБУВАННЯ ПРОГРАМНИХ РІШЕНЬ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПТАШНИКОМ 51

3.1. Моделювання роботи системи у Proteus..... 51

3.2 Роль брокера у забезпеченні роботи системи автоматизації та його функції..... 61

3.3 Вибір технології, функціонал та принцип роботи мобільного додатка для керування системою 67

Висновок до 3 розділу 73

РОЗДІЛ 4 МОДЕЛЬ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ПОЖЕЖ У ПТАХІВНИЦТВІ НА ОСНОВІ АНАЛІЗУ АБІОТИЧНОГО ФАКТОРУ 75

4.1 Аналіз моделі для машинного навчання. 75

4.2. Створення моделі для виявлення пожежі у птахівництві..... 86

Висновок до розділу 4 94
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА..... 95
5.1 Основна теорія охрони праці 95
5.2. Виробниче середовище та захист працівників від негативного впливу шкідливих виробничих чинників 96
5.3. Поняття ризику як міри небезпеки..... 98
5.4 Електробезпека в процесі роботи з автоматизованими системами. 101
5.5 Огляд питань кібербезпеки в процесі роботи з автоматизованими системами..... 106
5.6. Охорона навколишнього природнього середовища..... 108
ВИСНОВКИ..... 112
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 113

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей (Internet of Things).

MQTT – протокол повідомлень для передачі даних (Message Queuing Telemetry Transport).

Raspberry Pi – малий комп'ютер для розробки систем (Raspberry Pi)

GPIO – входи/виходи для загальних цілей (General Purpose Input/Output).

ВСТУП

В умовах сьогодення автоматизація визначає якість та стабільність галузі птахівництва. Здоров’я і продуктивність птиці безпосередньо залежить від контролю таких абіотичних факторів, як температура, вологість, освітленість та рівень CO_2. Порушення оптимальних умов утримання може призвести до значних втрат як у ефективності**,** так і в здоров’ї поголів’я. Завдяки розвитку АСУ стає можливим створення передових систем для моніторингу та контролю параметрів середовища в пташниках. Використання мікроконтролерів, сенсорів, протоколів передачі даних, а також технологій Інтернету речей (IoT) сприяє впровадженню інтегрованих рішень, які поєднують високу точність і зручність управління. А такі речі, як сучасні методи обробки даних, зокрема машинне навчання, застосовуються для покращення управління системою та додавання додаткових функцій, таких як виявлення небезпек.

В рамках цієї кваліфікаційної роботи проводиться аналіз сучасних технічних рішень для автоматизації управління абіотичними факторами, розробка власного проекту системи автоматизації, а також її програмна реалізація та випробування. Результати роботи спрямовані на створення універсальної автоматизованої системи, яка забезпечить підтримку оптимальних умов у пташниках, підвищить продуктивність птахівничих підприємств та сприятиме зниженню ризиків, пов’язаних із шкідливими факторами та небезпекою.

РОЗДІЛ 1
ОГЛЯД СУЧАСНИХ ТЕХНІЧНИХ РІШЕНЬ У АВТОМАТИЗОВАНИХ СИСТЕМАХ ДЛЯ КЕРУВАННЯ АБІОТИЧНИМИ ФАКТОРАМИ

1.1 Базові абіотичні фактори, які впливають на якість утримання птаства.

Для будь-яких живих істот існують фактори, при яких їх знаходження в середовищі перетворюється на комфортні умови, що зберігають їхнє життя, потомство, здоров'я, і в цілому їхню народжуваність. Фактори навколишнього середовища впливають на людей, тварин і все інше в природі. Екологічний чинник - це явище чи об'єкт, який впливає на організм. Можна навіть сказати, що ці чинники - це все, що оточує нас. Екологічні чинники можна поділити за своїм походженням [1]:

– біотичні чинники (хижацтво, паразити) - вплив живої природи на живу;

– антропогенні (зміна клімату, забруднення атмосфери) - вплив людини на живу природу.

Третій вид факторів – абіотичні, це прямий чи опосередкований вплив неорганічного середовища на живі організми. Цей тип чинника ділиться на фізичні прояви, які включають температуру навколишнього середовища, вологість повітря і ґрунту, освітлення і інтенсивність світла, швидкість і напрям вітру, атмосферний тиск, і фактори хімічного впливу, що включають склад ґрунту, вміст кисню і вуглекислого газу. У птахівництві абіотичні чинники мають прямий вплив на ефективність розведення будь-яких видів свійських птахів. Тому обов'язково потрібно їх брати до уваги, щоб правильно відстежувати стан птахів. Стрес як наслідок може порушити гомеостаз птиці та створити цілу мережу негативних факторів для її організму в рамках птахівництва[2]. Наприклад, навколишнє середовище - температурні коливання, рівень вологості повітря, якість освітлення та атмосфера - все це є

стресорами, які можуть суттєво вплинути на здоров'я, розвиток та продуктивність свійських птахів.

Вентиляція добре допомагає уникнути накопичення шкідливих газів, таких як аміак та вуглекислий газ. Аміак, який виділяється разом з випорожненнями тварин, може дратувати дихальні шляхи та послабити імунітет людини, в той час як вуглекислий газ може викликати гіпоксію та проблеми для птиці. Приміщення важливо провітрювати, використовуючи хорошу вентиляцію, де мешкають птахи. Необхідно стежити за тим, щоб рівень аміаку не перевищував 25 ppm, і рівень вуглекислого газу не перевищував 3000 ppm - такі високі концентрації даних речовин можуть викликати проблеми з диханням.

Також важливим аспектом стає контроль температурного режиму. Для перших, скажімо, одного-двох тижнів, температура має бути близько 32-34 °C, щоб вони могли нормально зростати та розвиватися. Нам потрібно уникати занадто холодного клімату, оскільки він може посилити стрес та хвороби. Ідеальна температура для дорослих птахів має становити 21-23 °C.

Висока вологість повітря провокує велике скупчення аміаку, а також хвороботворних мікроорганізмів. Низька вологість повітря призводить до висихання слизових оболонок та шкіри птахів, що надалі може призвести до погіршення стану птиці. Оптимальна вологість повітря у пташниках становить 50-70%. Занадто висока вологість може призвести до накопичення аміаку та розвитку хвороботворних мікроорганізмів, а надто низька - до сухості слизових оболонок та шкіри, що призведе до респіраторних захворювань та погіршення загального стану птиці.

У пташиних будинках дуже важливим є рівень шуму. Його слід підтримувати на рівні 80-100 дБ, щоб уникнути стресу птахів та зміни їхньої поведінки. Занадто високий рівень шуму здатен спричинити збільшення апетиту, затримку росту, зниження яйценосності.

Кількість світла значно позначається на стані та продуктивності свійської птиці. Для хорошої видимості та активності в перші дні життя курчат

має підтримуватись освітлення не нижче 20 люкс. Пізніше рівень освітлення поступово знижуватиметься до 5 люкс. Слід знати, що та сама довжина світла може вплинути на поведінку і здоров'я птахів. Наприклад, при використанні зеленого, червоного або білого кольору у курок зменшується почуття страху та підвищується денна активність. Завдяки хорошому контролю світла можна контролювати біологічні ритми.

1.2 Пташникові системи автоматизованого керування.

Автоматизація є важливою завдяки об’єднанню нових технологій і механізмів для виконання завдань з мінімальним або відсутнім залученням людей. Передбачається застосування комп'ютерних систем, програмного забезпечення і робототехніки для виконання конкретних функцій, які традиційно виконувались людьми. Це включає в себе зменшення витрат на працю, покращення якості та швидкості виконання завдань, а також зменшення кількості помилок, скоєних людьми.

Автоматизовані системи управління (АСУ) є комплексами програмного та апаратного забезпечення, призначені для автоматизації процесів управління в різних сферах діяльності.

Переваги автоматизованих систем на виробництвах полягають у наступному [3]:

1. Підвищена продуктивність. Автоматизація може бути корисною в багатьох галузях для підвищення продуктивності та ефективності пропонуючи підприємствам нові рішення. Наймання меншої кількості людей за рахунок автоматизації процесів - скорочення витрат. Через який продукт матиме нижчу ціну, і матиме високу конкурентоспроможність.

2. Нові можливості працевлаштування. Також слід відзначити позитивний ефект у вигляді створення нових робочих місць у зв'язку зі зростанням популярності автоматизації, програмування, робототехніки, IT-

забезпеченості та ін. Оскільки зараз потрібні кваліфіковані співробітники для створення та підтримки програмного забезпечення.

3. Підвищення якості. За рахунок автоматизації можливо поліпшити продукт. Як відомо, автоматизація може знизити кількість дефектних продуктів, що виникли внаслідок виробничого процесу без втручання людини.

4. Поліпшення умов праці:

Автоматизація передбачає роботу майже без участі людини, а значить позбавляється потреби від використання фізично статних людей, і уможливлює роботу людей з додатковими потребами. А також дозволяє закінчувати те саме завдання у встановлені терміни.

5. Покращення прийняття рішень. За допомогою технологій автоматизації ми зможемо швидко і точно аналізувати дані, в свою чергу, це дозволить співробітникам робити розумні рішення. Із твердженням можна сказати, що це підвищить ефективність та продуктивність компанії.

6. Покращене обслуговування клієнтів. Також завдяки автоматизації обслуговування клієнтів буде швидше і якісніше, відповідаючи на всі їхні запити.

І все ж, виходячи з усіх переваг, було б помилкою повністю замінити людину, її виключення може призвести до питання гнучкості системи та адаптивності. Рішення, прийняті на людському досвіді, логіки та обліку можливих наслідків спрямовані на досягнення максимальної ефективності системи.

Проте, залишається факт, що АСУ дозволяють знизити ймовірність "людського чинника", через який відбувається зсув параметрів, поломки, неточності у розрахунках. У зв'язку з цим, щоб досягти максимальної оптимізації процесів у системі, важливо визначити баланс між потребами в автоматизації та оперативного людського втручання. Автоматизовані системи управління можуть ефективно виконувати точну роботу та замінювати оператора, для виконання ним складніших завдань[4].

Кожна система складається з певних елементів, які будуть розглянуті з точки зору їх призначення та принципів роботи.

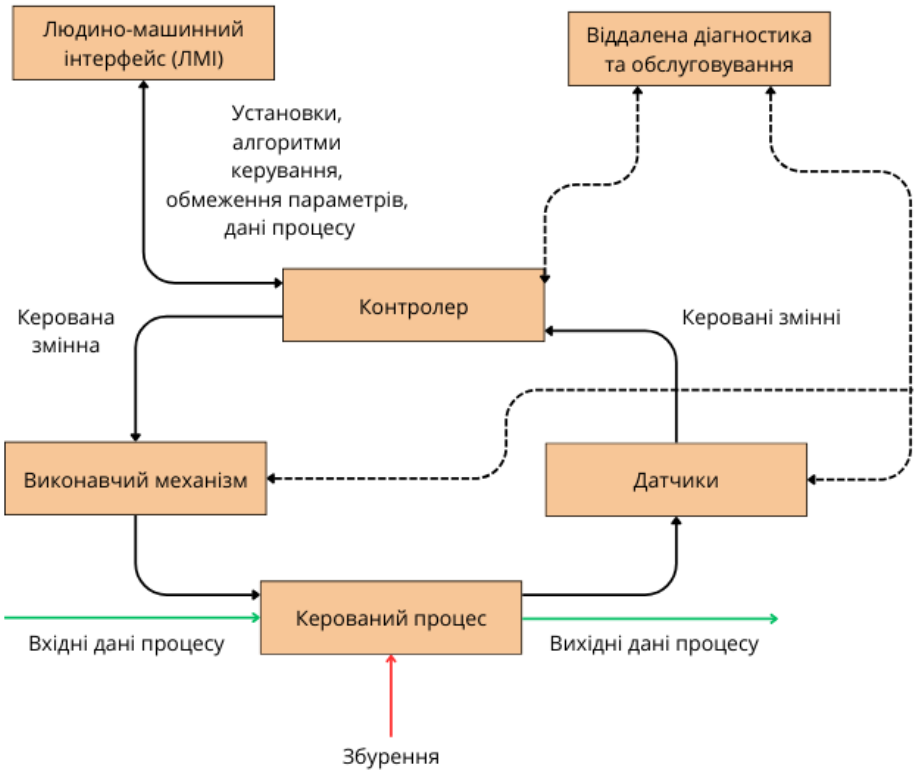

Рисунок 1.1 - Структура АСК [5]

На рисунку 1 показано основні структурні елементи системи автоматизованого керування. Система складається з ряду важливих компонентів, кожен із яких відповідає за виконання певної функції.

1. Контур управління оснащений датчиками, які збирають інформацію з навколишнього середовища, таку як температура, вологість, рівень рідини, освітлення та інші фактори. Вони постійно вимірюватимуть параметри навколо себе і надсилатимуть інформацію в контролер, надаючи можливість аналізувати дані і прийняти правильне рішення.

2. "Мозок" системи - це контролери, включаючи програмовані логічні контролери (ПЛК). Дані з датчиків обробляються та формується сигнал, на основі заздалегідь створених параметрів. ПЛК можна програмувати різними мовами, таких як LD, FBD, ST, для адаптації до конкретних вимог. Крім того,

ПЛК здатен інтегруватися з іншими системами. Саме через цей фактор інтеграція ПЛК у системи може зробити її більш надійною та високопродуктивною складовою.

3. Виконавчі пристрої виконують фізичні дії на основі вхідного сигналу, отриманого від контролера. Наприклад, вода або їжа може контролюватись клапанами, а реле можуть приводити в дію механізми, що використовуються для різних призначень. Ці пристрої механічно реалізують рішення з управління, прийняті системою.

4. Функція систем передачі полягає у встановленні інформаційного зв'язку між датчиками, контролерами та виконавчими механізмами. Існують різні протоколи зв'язку, включаючи дротові та бездротові технології, що дозволяють інтегрувати системи в існуючу інфраструктуру. Надійний і швидкий зв'язок між компонентами забезпечується за рахунок ефективних систем передачі.

5. Через людино-машинний інтерфейс (ЛМІ) оператори та інженери взаємодіють із системою, отримуючи можливість встановлювати параметри, налаштовувати алгоритми управління та коригувати налаштування контролера. ЛМІ також відображає поточний стан процесу та надає доступ до архівних даних, дозволяючи оператору відстежувати зміни та приймати обґрунтовані рішення.

6. Забезпечення належної роботи системи неможливе без віддаленої діагностики та сервісних інструментів. Вони сприяють запобіганню несправностям і їх своєчасному виявленню, а також відновленню функціонування системи, якщо такі проблеми виникають. Ці функції забезпечують надійність та стійкість автоматизованих систем, що особливо важливо у птахівництві, де оптимальні умови вирощування є запорукою здоров'я та продуктивності птиці.

Автоматизовані системи управління у свою чергу діляться на системи з відкритим контуром, які менш складні і дешевші, ніж другий тип системи з

закритим контуром, такі системи із зворотним зв'язком регулюють процес, порівнюючи фактичний результат із бажаним.

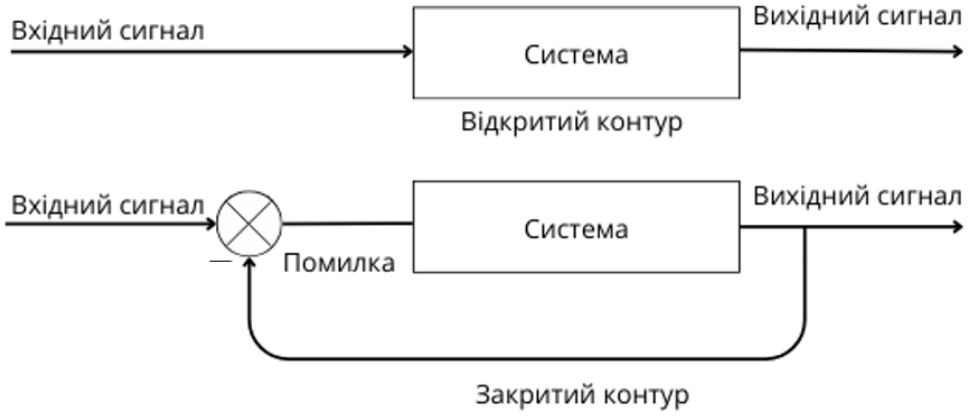

Рисунок 1.2 - Відкритий та закритий контур

У системах АСУ все частіше використовується Інтернет речей (IoT), який пов'язує кожного, де б він не знаходився, і дозволяє підключати безліч різноманітних пристроїв до мережі.

Інтернет речей є взаємодія між фізичним і цифровим світами через безліч датчиків і виконавчих механізмів. У цьому новому світі кожен пристрій підключено до мережі, дозволяючи їм спільно виконувати складні завдання. Пристрої IoT включають датчики, виконавчі механізми, процесори та приймачі.

Датчики збирають інформацію про різні параметри, такі як температура, вологість та рух, і надсилають ці дані для подальшого аналізу. Щоб отримати корисну інформацію, дані необхідно зберігати та обробляти. Цікаво, що в цьому контексті не тільки спеціалізовані пристрої, а й побутові предмети, такі як мобільні телефони або мікрохвильові печі, можуть вважатися «датчиками», якщо вони надають інформацію про свій стан або навколишнє середовище (наприклад, внутрішню температуру чи енергоспоживання).

Актуатори впливають на довкілля з урахуванням даних, отриманих від датчиків. Наприклад, коли датчик виявляє зміну температури в приміщенні, виконавчий механізм (наприклад, контролер кондиціонера) може включати

або вимикати охолоджувальну систему для підтримки комфортної температури.

Загалом кажучи, інтернет речей (IoT) - це мережа, де фізичні пристрої, оснащені датчиками, виконавчими механізмами та процесорами, можуть обмінюватися даними. Ці пристрої можуть містити як просту побутову техніку, так і складні промислові машини, всі вони підключені до інтернету для автоматизації, збору даних та інтелектуального прийняття рішень.

Саму архітектуру IoT можна розділити на трирівневу модель або складнішу п'ятирівневу модель [6]:

Перший рівень - це фізичний рівень, який збирає дані за допомогою датчиків. До них відносяться пристрої, які виявляють зміни у навколишньому середовищі, такі як: температура, вологість та рух.

Мережевий рівень: відповідає за підключення пристроїв Інтернету речей та передачі даних від датчиків на обробні пристрої або сервери. Часто використовуються бездротові технології, такі як Bluetooth, WLAN та Zigbee.

Рівень програм: відповідає за підключення пристроїв Інтернету речей та передачі даних від датчиків на обробні пристрої або сервери.

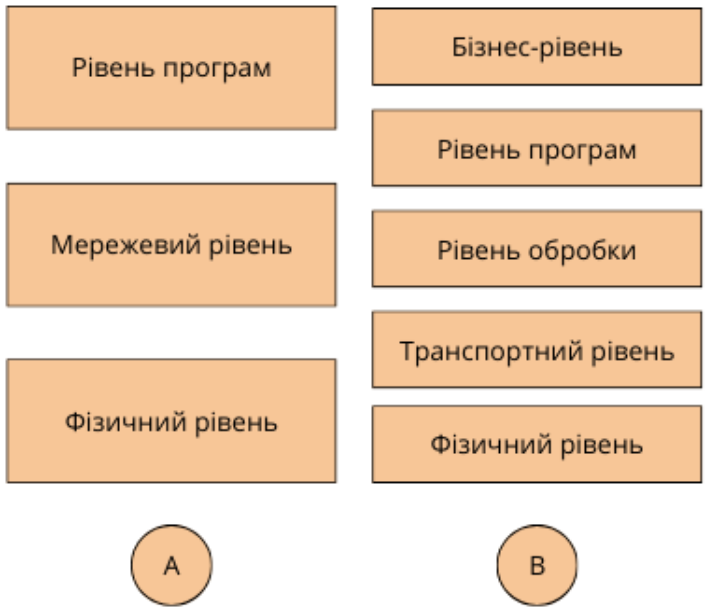

Рисунок 1.3 - Архітектура IoT: а) трирівнева модель; б) п'ятирівнева модель

У п'ятирівневій архітектурі додано такі рівні:

• Рівень обробки: Також називається проміжним рівнем, він обробляє, зберігає та аналізує дані.

• Бізнес-рівень: Цей рівень керує бізнес-аспектами, такими як конфіденційність даних, моделі витрат та взаємодія з користувачем.

Зв'язок між пристроями IoT здійснюється через бездротові технології, і вибір протоколу залежить від таких факторів, як енергоспоживання, дальність дії та розмір мережі. Ключові протоколи включають:

• RFID (радіочастотна ідентифікація): в основному використовується для відстеження об'єктів та реалізації інтелектуальних систем ідентифікації.

• NFC (зв'язок ближнього поля): бездротовий зв'язок на короткі відстані, в основному використовується для безпечних програм, таких як платежі.

• Bluetooth Low Energy (BLE): призначений для низького енергоспоживання та короткої дистанції зв'язку, широко використовується у фітнес-трекерах та розумних пристроях для дому.

• Zigbee: Протокол для малопотужного зв'язку з низькою швидкістю передачі даних на середні відстані, використовується в розумних будинках та промислових системах.

• 6LoWPAN: Адаптація IPv6 для малопотужних пристроїв, що дозволяє підтримувати зв'язок через Інтернет, заощаджуючи енергію.

• Wi-Fi призначений для високошвидкісної передачі даних на короткі та середні відстані (до 100-200 метрів), він використовується переважно для підключення пристроїв до Інтернету, наприклад, у домашніх або офісних мережах. Wi-Fi споживає більше енергії та потребує більш потужних пристроїв.

• WSN (Wireless Sensor Networks) - це мережі з безлічі малопотужних сенсорів, які збирають дані та передають їх через інші сенсори або безпосередньо на центральний вузол. Ці мережі зазвичай використовують протоколи з низьким енергоспоживанням, такі як IEEE 802.15.4 (основа Zigbee), щоб продовжити термін служби батарей пристроїв. WSN розраховані працювати з меншими обсягами даних і мають велику гнучкість у топологіях

(наприклад, багатовузлова маршрутизація), що дозволяє використовувати їх для моніторингу великих територій.

Дані пристроїв IoT можуть оброблятися на хмарних платформах або ближче до пристроїв в архітектурах туманних обчислень. Хмарні обчислення дозволяють централізовану обробку з великими ресурсами, але туманні обчислення забезпечують зберігання та обробку даних ближче до пристрою, що знижує затримку та покращує прийняття рішень у реальному часі. Це особливо корисно у додатках, де потрібна швидка реакція, наприклад, у керуванні дорожнім рухом або системах розумних електромереж.

IoT має безліч застосувань у різних галузях [7]:

1. Розумні будинки: автоматизація освітлення, безпеки та побутової техніки для економії енергії та зручності.

2. Охорона здоров'я**:** віддалений моніторинг здоров'я пацієнтів за допомогою пристроїв, які носять пацієнти, що відстежують життєве важливі показники в реальному часі."

3. Промислова автоматизація: моніторинг та управління промисловими процесами з використанням датчиків та виконавчих механізмів для підвищення ефективності.

4. Розумні міста: управління ресурсами, такими як вода та енергія, моніторинг дорожнього руху та підвищення суспільної безпеки за допомогою підключених пристроїв.

Інтернет речей покращує якість життя за рахунок автоматизації повторюваних завдань, підвищення безпеки та надання більш точних даних для прийняття рішень. Це стає невід’ємною частиною автоматизації на різних підприємствах, пропонуючи нові можливості для моніторингу, управління та оптимізації процесів. При цьому зібрані дані можуть бути проаналізовані за допомогою машинного навчання, що сприяє виявленню закономірностей.

1.3 Існуючі системи для підтримки оптимальних умов для птахів.

Завдяки аналізу існуючих рішень можна буде чітко визначити, що потрібно саме для нашого конкретного завдання. Аналіз статей з АСУ наддасть широке розуміння проектування системи.

У [8] описана система, яка використовує бездротові датчики та мобільні мережі для віддаленого моніторингу та контролю умов на птахофабриках. Основна мета системи - підтримання оптимальних параметрів довкілля (температури, вологості, рівня води) задля ефективного зростання птахів. Система використовує датчики передачі даних на мережевий сервер та оповіщення операторів у разі виникнення надзвичайної ситуації. Оператори можуть відповісти на повідомлення за допомогою текстового повідомлення та взяти управління системою під свій контроль, або, якщо протягом заданого часу не буде отримано відповіді, система автоматично здійснить необхідні дії. Зберігання даних здійснюється за допомогою таких технологій, як мікроконтролер CC3200, модуль GSM та таблиці Google.

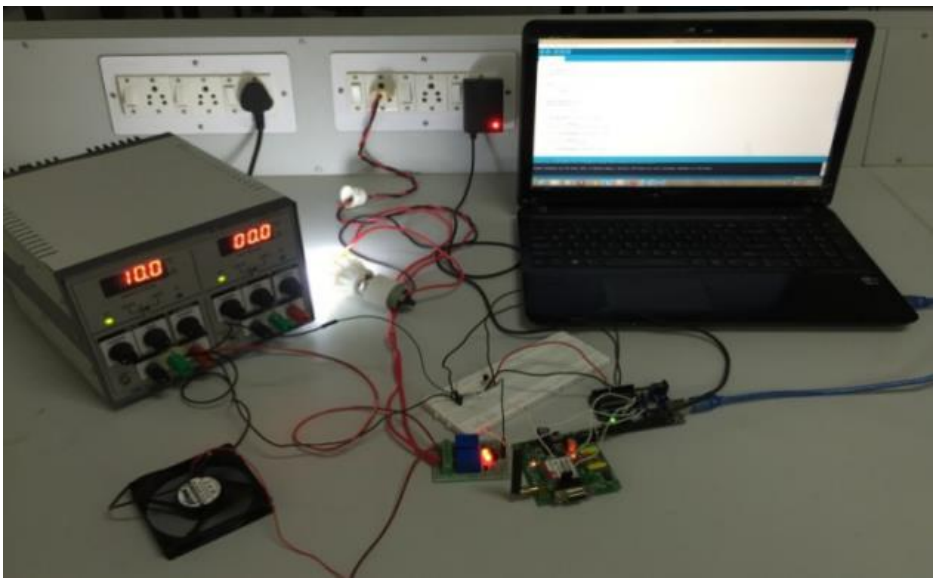

Рисунок 1.4 - Система, налаштована на спрацювання вентилятора.

Однак оператор у цій системі не може віддалено змінити відповідні параметри. Йому доведеться контактувати безпосередньо із самою технікою.

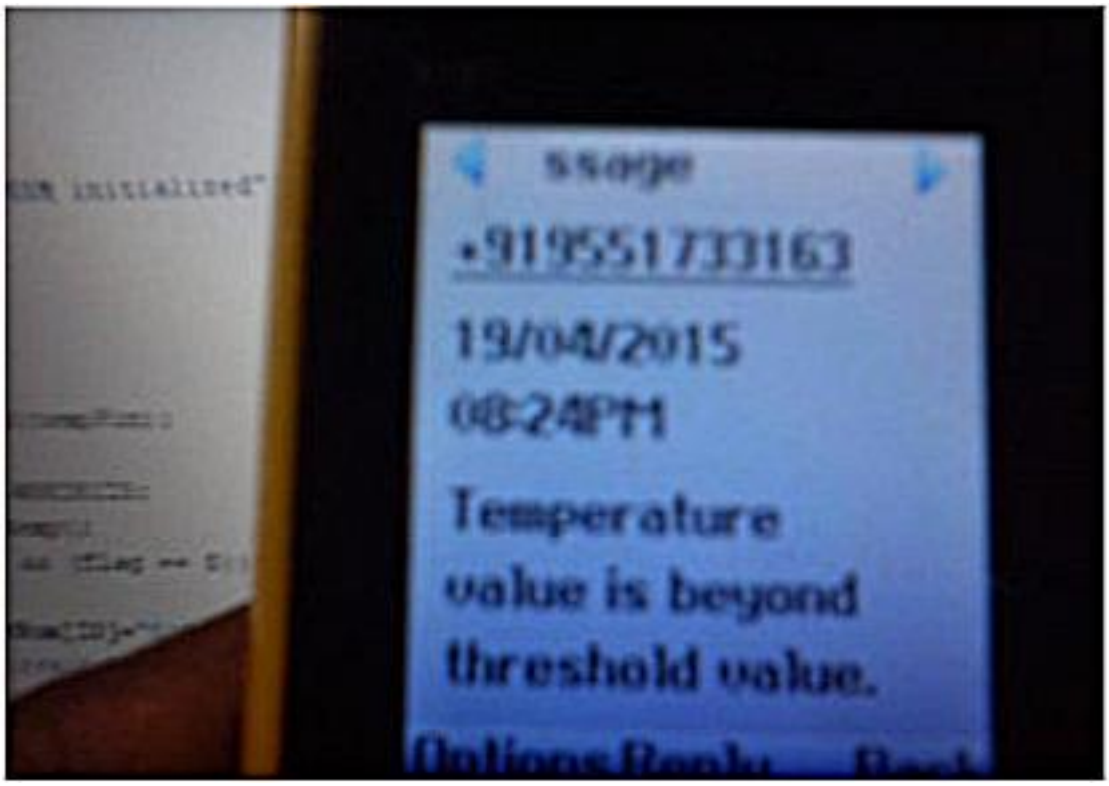

Рисунок 1.5 - Повідомлення про помилку

Стаття [9] присвячена розробці та реалізації інтелектуальної системи управління пташником для підтримки оптимальних умов зростання птахів. У статті описується використання мікроконтролера Arduino Uno для автоматичного контролю температури та вологості в пташнику за допомогою датчика DHT22. Система дозволяє підтримувати комфортний мікроклімат, включаючи та відключаючи вентилятори та нагрівачі залежно від поточних умов. Мета системи – мінімізувати смертність серед птахів, особливо на ранніх стадіях їх зростання, за рахунок автоматизації процесів керування довкіллям. Система має перемикачі для вибору режиму роботи в залежності від віку птахів, що робить управління ще більш ефективним.

Можливість налаштування режиму залежно від віку птахів робить систему гнучкою та застосовною у різних умовах.

Однак через те, що система мікроклімату призначена для певної вікової категорії курчат, програми, вбудовані в мікроконтролер, не підходитимуть іншим віковим групам чи іншим породам. Зростання курчат на ранніх етапах як ніколи важливе, і за такої категорії потрібен пильний моніторинг, тому

інформація, яка виводиться на LED-екран, може бути запізнілою для перегляду людини.

У статті [10] для рішень використовується безліч сучасних технологій та компонентів для створення розумного інкубатора для яєць.

Основним мікроконтролером є Raspberry Pi 3, який управляє всією системою інкубатора, включаючи підключення до інтернету через LAN-модуль для віддаленого моніторингу. Системи контролю температури і вологості залежать від датчика SHT35, який передає інформацію для подальшої обробки.

Як елементи обігріву автор використовує лампи розжарювання, які вмикаються і вимикаються за допомогою реле, а для охолодження інкубатора при перегріві передбачено декілька вентиляторів.

У доповненні до мікроклімату в [10] також йдеться про автоматичне перевертання яєць за допомогою сервомоторів. PIR-сенсори та камера використовуються для фіксації руху, розпізнають процес вилуплення і відправляють повідомлення користувачам через інтернет. Усі кліматичні параметри відображаються на LED-екранах як повідомлення про нормальні або ненормальні параметри системи, що відхиляються від норми.

Рисунок 1.6 - Дисплей розумного інкубатора для яєць.

Рисунок 1.7 - Блок управління розумного інкубатора для яєць.

Система керується за допомогою алгоритму нечіткої логіки Мамдані, який забезпечує адаптивний контроль температури та вологості, дозволяючи інкубатору реагувати на зміни умов реального часу. Однак у ньому є недолік, який полягає в тому, що може бути недостатньо точним за складних і швидких змін умов. Наприклад, стрибки температури чи вологості можуть повністю коригуватися алгоритмами, оскільки система не має чітких моделей. Алгоритм Мамдані функціонуватиме тільки в рамках заданих наперед правил та функцій приналежності, що характеризують поведінку системи. Це говорить про те, що його робота залежить лише від тих умов і сценаріїв, які були визначені в процесі налаштування. Хоча в статті не вказується, наскільки адаптивним є алгоритм у реальних умовах.

Висновок до 1 розділу

Підсумовуючи перший розділ, можна стверджувати наступне: визначено, що абіотичні фактори суттєво впливають на курей, що є важливим аспектом для ефективного управління їхнім утриманням. Розглянуто елементи автоматизації, зокрема технології, що використовуються для управління процесами в сільському господарстві, що підтверджує актуальність впровадження нових рішень у сучасному аграрному секторі. Проведено огляд

статей з управління мікрокліматом та виконавчими механізмами, виявлено їхні переваги та недоліки, що дозволяє визначити перспективи їх використання в галузі та шляхи для подальшої розробки.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ТА РОЗРОБКА ТЕХНІЧНИХ РІШЕНЬ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПТАХОКОМПЛЕКСОМ

2.1. Мережевий протокол передачі даних.

У системі моніторингу мікроклімату курника передбачено віддалений доступ, тому важливим аспектом є ефективна передача даних. Загалом система передачі є ключовим компонентом технологій Інтернету речей (IoT), які активно застосовуються для автоматизації у птахівництві. Для забезпечення функціонування системи використовувалися бездротові мережі сенсорів (WSNs). В основному відсутність необхідності розташовувати проводом обґрунтовано їх ефективними протоколами передачі даних, зручністю експлуатації та економічною вигодою порівняно з провідними мережами.

Переваги бездротових сенсорних мереж над дротовими полягає у наступному.

1) здатності до самовідновлення та самоорганізації;
2) передача інформації великі відстані з низькою потужністю передавачів рахунок можливості ретрансляції;
3) низької вартості вузлів, їх компактності та низького енергоспоживання;
4) можливість живлення від автономних джерел;
5) легкість встановлення без прокладання кабелів;
6) можливості інтеграції у існуючі об'єкти без істотних змін;
7) низька вартість технічного обслуговування.

Для оптимальної роботи бездротової сенсорної мережі її компоненти організовані таким чином, щоб забезпечити збір даних, обробку, передачу та живлення. Зазвичай це сенсори, мікроконтролери, модулі для передачі даних та джерела живлення. Без кожного з цих елементів ефективність роботи мережі була б під загрозою.

Структура типового сенсорного вузла показана на рис. 2.1.

Рисунок 2.1 - Сенсорний вузол

За останні кілька років технології IoT значно розвинулися, як і протоколи WSN. Вони в основному розроблені з точки зору вартості, часу роботи батареї, пропускної здатності, швидкості передачі даних.

У табл. 2.1 представлено широко використовувані протоколи WSN

Таблиця 2.1 - Типи протоколів зв'язку WSN

Типи протоколів зв'язку WSN	Топологія:	Швидкість передачі даних:	Робочий діапазон:	Споживання енергії:
IEEE 802.15.4	Зірка, P2P	20-250 Кбіт/с	10 м	Низьке
Zigbee	Зірка, дерево, сітка	250 Кбіт/с	10-100 м	Низьке

BLE	Зірка, дерево	1-2 Мбіт/с	30 м	Низьке
6LowPAN	Зірка, сітка	0.3-50 Кбіт/с	10-100 м	Низьке
LoRa WAN	2G/3G, Зірка, P2P	< 50 Кбіт/с	5-50 км	Низьке
RFID	P2P	50 Кбіт/с	10-20 см	Дуже низьке
Wi-Fi	Зірка	0.1-54 Мбіт/с	< 92 м	Низьке
Стільниковий	Зірка, P2P	10 Мбіт/с	Кілька км	Високе

В розумних курятниках протокол IEEE 802.15.4 є одним з найпоширеніших для бездротового зв'язку. Він працює на рівнях керування доступом до середовища (MAC) і фізичному (PHY) рівнях низькошвидкісної бездротової персональної мережі (LR-WPAN), що є енергоефективним і економічним рішенням для коротких відстаней. Протокол забезпечує хорошу автономність батареї та є економічним, але для курятника він не підходить через короткий діапазон і повільну швидкість передачі.

Протокол ZigBee, на відміну від інших, використовує бездротову мережу LR-WPAN, як і IEEE 802.15.4, для надійної передачі даних. Мережа ZigBee складається з координатора, маршрутизатора та кінцевих пристроїв, які можуть працювати за трьома основними топологіями: зірка, сітка та дерево кластерів. Висока вартість компонентів ZigBee робить цей протокол мало підходящим для застосування в домашніх курятниках.

Bluetooth Low Energy (BLE) з низьким енергоспоживанням і коротким діапазоном (до 10 м) добре підходить для збору даних з датчиків, однак через низьку безпеку та обмеження на кількість пристроїв він не підходить для систем мікроклімату у курятника.

Протокол стільникового зв'язку 3G, 4G та GSM підходить для високошвидкісної передачі великих обсягів даних через Інтернет, але через

велике енергоспоживання його ефективність знижується для використання в курятниках.

Wi-Fi є найбільш використовуваним протоколом для WSN, що працює за стандартами IEEE 802.11 (включаючи 802.11a, 802.11b, 802.11g та 802.11n), і дозволяє підключення до бездротової локальної мережі, що робить його підходящим для моніторингу кліматичних умов у домашньому курятнику.

Серед усіх перелічених протоколів для системи обрано Wi-Fi, оскільки:

1. Більшість приватних господарств вже підключено до Wi-Fi, що спрощує інтеграцію.
2. Завдяки вже налаштованій інфраструктурі Wi-Fi може передавати дані на високій швидкості.

Буде обрано протокол для обміну інформацією за для налаштування стабільного Wi-Fi з'єднання. Обраний тип протоколу буде важливий для ефективного та надійного з'єднання.

Розглянуто три популярних протоколи, які використовуються для передачі даних по Wi-Fi у табл. 2.2: MQTT [11], HTTP та WebSocket [12].

Таблиця 2.2 - Порівняння протоколів передач даних.

Аспекти	MQTT	HTTP	WebSocket
Модель обміну даними	Публікація-підписка (publish-subscribe)	Класична клієнт-серверна модель	Двостороннє з'єднання
Ефективність	Легкий, мінімальні накладні витрати	Великі заголовки, постійне встановлення з'єднання	Постійне з'єднання, але більше навантаження на ресурси
Масштабо-ваність	Висока: додавання нових пристроїв без змін в архітектурі	Низька: кожен запит ініціює клієнт, що ускладнює масштабування	Середня: підтримка постійних з'єднань з кожним клієнтом
Надійність	Підтримка QoS (3 рівні доставки повідомлень)	Залежить від наявності з'єднання, не	Надійність на рівні з'єднання, немає вбудованої системи повторної

		гарантує доставку	передачі даних при збоях
Протокол безпеки	TLS/SSL (не вбудовано, але доступно)	HTTPS (вбудоване шифрування)	WSS (WebSocket Secure)
Підтримка великих даних	Підходить для передачі малих пакетів даних, оптимальний для IoT	Добре для передачі великих обсягів даних (веб-сторінки, файли)	Добре підходить для передачі малих та середніх даних в реальному часі
Інтеграція з мобільними платформами	Широко підтримується, легко інтегрується в IoT	Широко підтримується, але менш оптимізований для IoT	Добре підтримується в веб-додатках
Енергетична ефективність	Дуже низькі накладні витрати, підходить для пристроїв з обмеженим споживанням енергії	Високі накладні витрати на встановлення з'єднань	Постійне з'єднання може підвищити енергоспоживання
Гнучкість архітектури	Дуже гнучка архітектура, легко додавати нові пристрої	Потрібні зміни в структурі для додавання нових функцій	-
Сумісність з іншими технологіями	Хороша інтеграція з хмарними сервісами та іншими технологіями (Wi-Fi, ZigBee тощо)	Універсально підтримується, але менш гнучкий в контексті IoT	-

MQTT є найкращим варіантом для IoT завдяки своїй ефективності, низькому енергоспоживанню та гнучкості архітектури, забезпечуючи високу масштабованість і надійність з підтримкою QoS та простотою інтеграції. Він підходить для передачі малих даних і має дуже низькі накладні витрати. HTTP є популярним протоколом, але для IoT він не оптимізований. Він підходить для передачі великих обсягів даних, але має значні накладні витрати на

встановлення з'єднань. HTTP не гарантує доставку повідомлень і має меншу гнучкість при розширенні архітектури.

WebSocket забезпечує постійне двостороннє з'єднання і добре підходить для реального часу та веб-додатків, але має середнє навантаження на ресурси і збільшує енергоспоживання через постійне з'єднання. Він підтримує передачу малих і середніх даних і має помірну масштабованість.

Вибір брокера для MQTT-зв'язку визначає, як ефективно та якими методами повідомлення будуть передаватися між сенсорами, Raspberry Pi та користувацькими додатками.

Згідно з аналізом таблиці (табл. 2.3), робиться висновок, що Mosquitto - це простий в установці та гарне рішення для використання на середніх системах. Завдяки високим результатам і мінімальним вимогам до ресурсів, він є ідеальним для IoT-систем з обмеженими ресурсами. Завдяки підтримці стандарту MQTT, Mosquitto зручно збирає інформацію з датчиків, що є важливим для таких проектів. Крім того, процес налаштування системи став легким завдяки простоті інтеграції з іншими пристроями. У той час як HiveMQ та EMQX оптимізовані для більших систем і підтримують кластеризацію, вони потребують більше ресурсів і складніші в налаштуванні. RabbitMQ більше орієнтований на роботу з чергами повідомлень, а VerneMQ – на складні масштабовані рішення. Підтримка TLS і MQTT є у всіх брокерів, проте HiveMQ та EMQX надають розширені функції безпеки, що є важливими для складних систем.

Візуалізація використання мережевих протоколів даних на рис 2.2. MQTT-брокер Mosquitto реалізує з'єднання між сенсорами, Raspberry Pi та користувацькими додатками. MQTT дає можливість в реальному часі відстежувати дані про мікроклімат курятника через мобільний додаток або веб-інтерфейс і керувати системою віддалено.

Таблиця - 2.3. - Порівняння брокерів для MQTT-зв'язка

Параметри	Mosquitto	HiveMQ	EMQX	RabbitMQ	VerneMQ
Легкість встанов-лення	Дуже легкий.	Простий, потребує більше ресурсів	Легкий, але складніше в конфігурації.	Більш складний через додаткові модулі	Широко підтримується, особливо в IoT
Продуктивність	Висока для малих і середніх систем	Оптимізований для великих систем	Підтримує високу пропускну здатність	Менш продуктивний для MQTT, більше для черг	Добре інтегрується з бізнес-системами
Масштабо-ваність	Підходить для локальних систем	Дуже висока, підтримує кластеризацію	Висока, вбудована підтримка кластерів	Низька для MQTT, краще підходить для AMQP	Підходить для IoT та великих мереж
Підтримка протоколів	MQTT 3.1, 3.1.1, 5.0	MQTT 3.1.1, 5.0	MQTT 3.1, 3.1.1, 5.0	MQTT, AMQP, STOMP, HTTP, WebSocket	Орієнтований на мультипротокольні системи
Безпека	Підтримує TLS/SSL	Розширена підтримка безпеки	Вбудовані TLS, ACL, та аутентифікація	Вбудовані інструменти для ACL і шифрування	Найкраще підходить для IoT
Гнучкість	Простий, але мінімальний функціонал	Висока, розширення через плагіни	Гнучкий, багато вбудованих інструментів	Дуже гнучкий, багатофункціональний	Широко підтримується, особливо в IoT

Продовження табл.2.3.

Сумісність	Широко підтримується,особливо в IoT	Добре інтегрується з бізнес-системами	Підходить для IoT та великих мереж	Орієнтований на мультипротокольні системи	Добре інтегруєтся з бізнес-системами
Споживання ресурсів	Низьке,	Помірне	Помірне	Високе	Підходить для IoT та великих мереж
Вартість	Безкоштовний	Платний (є безкоштов-на версія)	Платний (є безкоштовна версія)	Безкоштовний (з обмеженнями)	Орієнтований на мультипротокольні системи

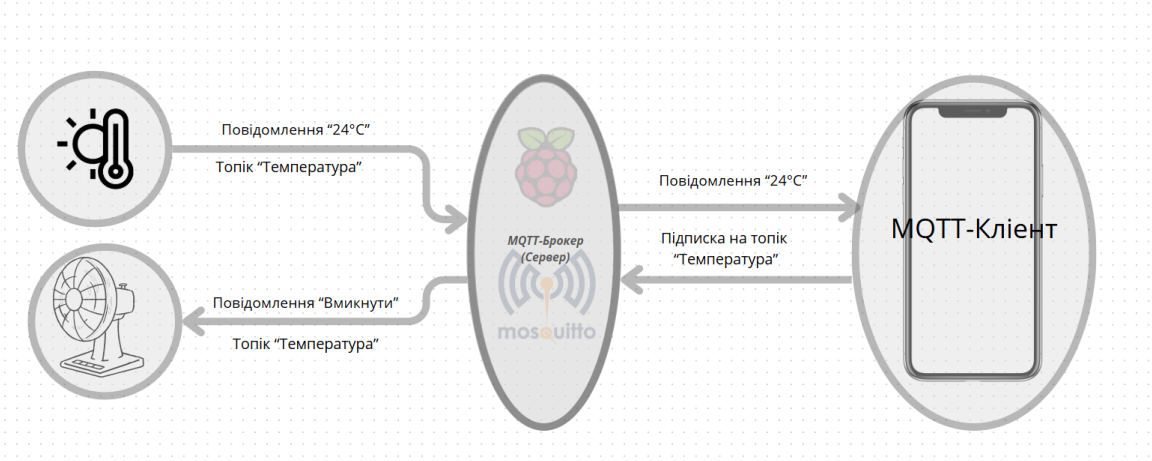

Рисунок 2.2 - Діаграма взаємодії з брокером MQTT

Після того, як буде визначена структура системи, наступний крок полягає у виборі компонентів.

2.2 Мікрокомп'ютер для керування системою.

Raspberry Pi 3 модель B [13] – це мікрокомп'ютер із вбудованою підтримкою Wi-Fi та Bluetooth.

На відміну від звичайних мікроконтролерів, Raspberry Pi 3 має значну обчислювальну потужність, завдяки якій можна використовувати повноцінні операційні системи, такі як Raspbian (засновану на Linux). Завдяки цьому маємо можливість працювати з більш складними проектами: оброблювати дані, застосовувати графічні інтерфейси та підключати до системи різні датчики, виконавчи механізми та інші. Raspberry Pi 3 оснащений процесором Quad-core ARM Cortex-A53 із тактовою частотою 1.2 GHz, 1 GB RAM, вбудованим Wi-Fi 802.11n та Bluetooth 4.2. З 40 пінів GPIO, підтримка яких включає інтерфейси SPI, I2C, UART, PWM та інші, пристрій використовує microSD карту для зберігання даних та має USB порти для підключення зовнішніх пристроїв. Відеовиходи включають HDMI та аудіо виходи, підтримка 1080p відео. Енергоспоживання Raspberry Pi 3 зазвичай становить 2.5W - 3.5W в активному режимі.

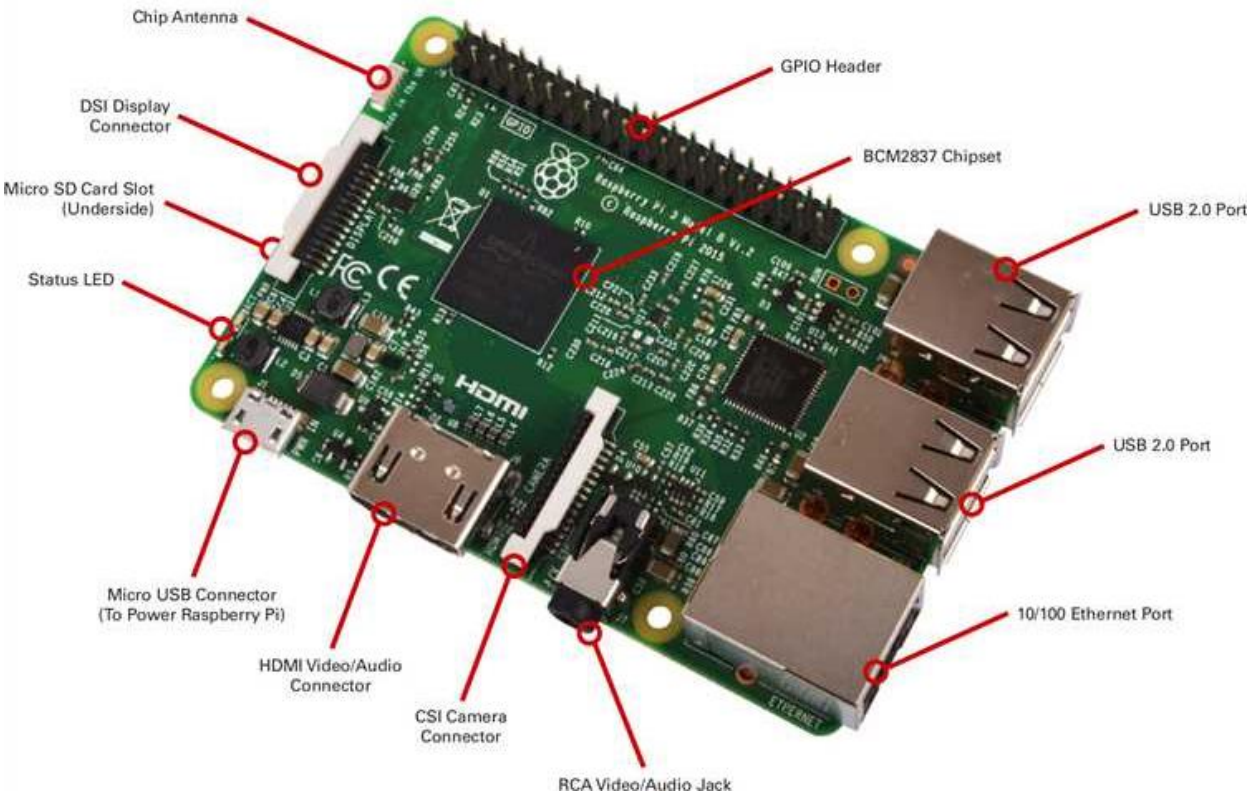

Рисунок 2.3 - Raspberry Pi 3 модель B

Таблиця 2.4 - Опис пінів Raspberry Pi 3

Pin 1, 17	DC Power	3.3V
Pin 2, 4	DC Power	5V живлення.
Pin 6, 9, 14, 20, 25, 39	Ground (GND)	Земля.
Pin 3	I2C SDA (GPIO2)	Лінія даних SDA для I2C.
Pin 5	I2C SCL (GPIO3)	Лінія синхронізації SCL для I2C.
Pin 7, 11, 12, 13, 15, 16, 18, 23, 24, 26, 32, 33, 35, 36, 37, 38, 40, 29, 30	GPIO	Загальні входи/виходи, PWM, для підключення датчиків і пристроїв.
Pin 8	UART TXD (GPIO14)	Передача даних для UART.
Pin 10	UART RXD (GPIO15)	Прийом даних для UART.
Pin 19	SPI MOSI (GPIO10)	Master Out Slave In для SPI.
Pin 21	SPI MISO (GPIO9)	Master In Slave Out для SPI.
Pin 23	SPI CLK (GPIO11)	Сигнал синхронізації для SPI.
Pin 24	SPI CE0 (GPIO8)	Вибір пристрою 0 для SPI.
Pin 26	SPI CE1 (GPIO7)	Вибір пристрою 1 для SPI.
Pin 27, 28	ID_SD, ID_SC	Лінія даних SDA для I2C.

I2C(Inter-Integrated Circuit) - це простий двопровідний інтерфейс для зв’язку між мікроконтролерами, сенсорами.

Основні переваги - проста схема підключення та можливість з'єднання кількох пристроїв на одній шині. Приклади пристроїв: датчик DHT22, АЦП MCP3008, OLED-дисплеї.

Інтерфейс SPI дає змогу передавати дані з великою швидкістю за допомогою чотирьох проводів. Застосовується для підключення дисплеїв, АЦП, модулів пам'яті, та інших пристроїв, які потребують швидкої взаємодії.

Інтерфейс UART (Universal Asynchronous Receiver-Transmitter) підтримує асинхронний обмін даними між мікроконтролерами. Поширений приклад - підключення Raspberry Pi до комп'ютера або модуля Bluetooth для передачі текстових даних.

GPIO (General Purpose Input/Output) - піні, які можна програмувати для роботи як входи чи виходи. Застосовуються для підключення кнопок, індикаторів, реле та інших електронних пристроїв. Наприклад, піни GPIO4 та GPIO27 здатні виконувати функції як сенсорів, так і елементів керування периферією.

Raspberry Pi 3 підходить для курника з кількох причин: по-перше, він здатний обробляти великі обсяги даних та виконувати складні обчислення, що корисно за наявності безлічі датчиків для моніторингу температури, вологості, освітленості та інших параметрів. По-друге, Raspberry Pi 3 може підключати та керувати кількома пристроями через USB порти та GPIO, що важливо для інтеграції різних сенсорів та пристроїв керування, таких як реле, камери та системи моніторингу.

По-третє, наявність вбудованого Wi-Fi та Bluetooth дозволяє Raspberry Pi підключатися до мережі та віддалено керувати системою, що спрощує автоматизацію процесів у курнику.

Linux-оточення на Raspberry Pi 3 дає змогу програмувати за допомогою різних мов, таких як Python і Node.js, для налаштування системи автоматизації. Raspberry Pi підтримує створення віддалених інтерфейсів для управління, зокрема у вигляді веб-інтерфейсів і мобільних додатків. Проте є деякі мінуси. Raspberry Pi 3 споживає значно більше енергії, ніж мікроконтролери, що

створює труднощі при живленні від автономних систем, таких як сонячні панелі.

2.3 Вибір дисплея на основі LED технології.

Технології освітлення зазнали значних змін. Сучасні джерела світла пропонують високу енергоефективність, довговічність та покращені якості освітлення в порівнянні з традиційними лампами. Розглянемо основні технології освітлення з таблиці 2.4 та підіб'ємо підсумки.

Таблиця 2.4. - Порівняння технологій освітлення

Параметр	LED Світлодіодні лампи	OLED (Organic Light Emitting Diodes)	TFT (Thin-Film Transistor)
Енергоефективність	Висока	Середня	Середня
Термін служби	Довгий	Довгий	Середній
Екологічність	Низька	Низька	Середня
Стійкість до ударів	Висока	Середня	Середня
Яскравість	Дуже висока	Середня	Висока
Колірна температура	Широкий діапазон	Широкий діапазон	Середній діапазон
Розміри	Компактні	Компактні	Середні
Ціна	Середня	Висока	Середні
Використання	Освітлення, дисплеї, індикатори	Телевізори, смартфони, дисплеї	Дисплеї, мобільні тел., моніторингу,

Аналіз таблиці дозволяє зробити висновок, що LED є найкращим варіантом для освітлення завдяки своїм характеристикам: економія енергії, надійність і світлова інтенсивність. OLED зручний для дисплеїв, але дорожчий і менш яскравий, ніж LED. TFT використовується в екранах, але поступається в енергоефективності та довговічності. Тому для освітлення та економії енергії краще вибрати LED.

Тому обрано модуль LCD2004 [14] (рис.2.4) - це LED-дисплей використовується для відображення текстових повідомлень та статусів у

реальному часі, надаючи актуальну інформацію про стан системи, повідомлення про критичні ситуації та підтвердження правильного функціонування у курнику. У проекті він є одним із засобів моніторингу інформації.

Завдяки великій площі екрану дисплей здатний вивести більше даних порівняно з маленькими. Він економить піні завдяки використанню I2C інтерфейсу, що дозволяє підключити його лише чотирма проводами, звільняючи місце на платі та спрощуючи підключення. Оскільки дисплей побудований на LED-технології, він економить енергію порівняно з OLED та TFT, при цьому довго служитиме.

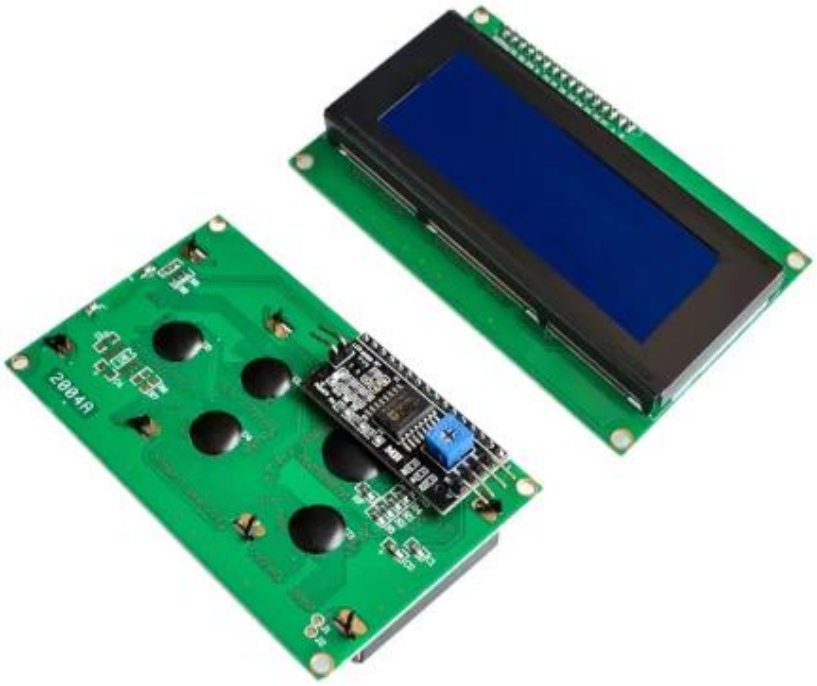

Рисунок 2.4 - Модуль LCD2004

Таблиця 2.5 - Піни модуля LCD2004

Контакт	Опис	функція
VSS	Живлення дисплея (−)	земля (GND), що використовується як загальний мінус для всього пристрою, створює замкнутий ланцюг.
VDD	Живлення дисплея (+)	живлення дисплея (+), подає позитивну напругу роботи дисплея.

VO	Контрастність дисплея	регулювання контрасту, керує яскравістю пікселів дисплея, впливає на контраст зображення.
RS	Вибір регістру: команда / дані	вибір регістру (команда або дані) визначає, що саме передається в дисплей - команда або дані для відображення.
RW	Вибір режиму: запис / читання	вибір режиму (читання або запис), вказує, чи відбуватиметься читання інформації з дисплея або запис даних на дисплей.
E	Лінія синхронізації	тактовий сигнал, синхронізує передачу даних, сигналізуючи дисплею, коли потрібно прийняти чи надіслати дані.
D0-D7	Шина даних	шина даних, через неї передаються дані між контролером та дисплеєм (8 біт інформації).
A	Живлення підсвічування (+)	живлення підсвічування (+), подає позитивну напругу для роботи підсвічування дисплея.
K	Живлення підсвічування (−)	живлення підсвічування (−), подає негативну напругу для підсвічування дисплея, завершує ланцюг підсвічування.

2.4 Вибір аналогових датчиків.

LM35 [15] (рис.2.5) - датчик температури з аналоговим виходом, розроблений Texas Instruments для використання в проектах контролю температури. Діапазон температур від -55°C до +150°C, який покриває LM35, робить його підходящим для численних застосувань. Точність цього датчика становить ±0.5°C в діапазоні від 25°C, що є добрим показником для більшості побутових та промислових додатків.

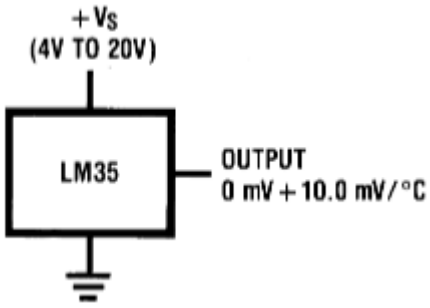

Рисунок 2.5 - Схема датчика LM35

LM35 видає вихідну напругу, що пропорційна температурі, яку він вимірює. Ця напруга становить 10 мВ на кожен градус Цельсія (наприклад, при 25°C вихід буде 250 мВ). LM35 функціонує при напрузі від 4 до 20 В, що робить його сумісним з різними електричними схемами. Його енергоспоживання на дуже низькому рівні, що робить його зручним для автономних систем. Однак датчик вимагає аналогового входу для зчитування значень, що може вимагати додаткових компонентів, таких як АЦП (аналогово-цифровий перетворювач), і не має вбудованого калібрування, що може вимагати точного налаштування більш високої точності.

Аналіз датчика температури LM35 показує, що це прецизійний інтегральний температурний сенсор, який забезпечує вихідну напругу, пропорційну температурі Цельсіях. Це значне перевагу порівняно з датчиками, відкаліброваними в шкалі Кельвіна, оскільки користувачеві не потрібно робити додаткові обчислення для отримання значення зручною шкалою Цельсія.

LM35 не вимагає зовнішнього калібрування, що спрощує його використання та знижує вартість пристрою. Його точність становить ±1/4°C при кімнатній температурі, а на всьому діапазоні від -55°C до +150°C вона залишається в межах ±3/4°C. Ці характеристики роблять датчик придатним до різних додатків, де важлива висока точність вимірювань у широкому температурному діапазоні.

Відмінною рисою LM35 є низький вихідний опір і лінійність вихідного сигналу, що спрощує його підключення до пристроїв, що зчитують і управляють. Крім того, датчик споживає всього 60 мкА, що забезпечує мінімальне самонагрівання (менше 0,1 ° C у спокійному повітрі). Це особливо важливо при тривалих вимірах у стабільних умовах.

Що стосується діапазону температур, LM35 має робочий діапазон від -55°C до +150°C, а його модифікація LM35C обмежена діапазоном від -40°C до +110°C, але з покращеною точністю. LM35 постачається в кількох типах корпусів, зокрема герметичному TO-46, пластиковому TO-92, а також корпусах

для поверхневого монтажу та TO-220, що дає змогу вибрати найбільш підходящий для конкретних умов. LM35 надає точні показники температури, а його простота у використанні робить його чудовим вибором для багатьох електронних проектів.

HIH-5030 [16](рис.2.6) - це датчик вологості від компанії Honeywell, призначений для вимірювання відносної вологості та температури навколишнього середовища. Цей датчик є популярним вибором для використання в системах моніторингу клімату, наприклад, в системах вентиляції, кондиціювання, агротехнологіях, а також в обладнанні для захисту від вологи. HIH-5030 може вимірювати температуру від -40°C до +85°C і вологість від 0 до 100% RH (відносна вологість). Датчик видає аналоговий сигнал, пропорційний рівню вологості та температурі. Наприклад, для вологості 0% RH вихідна напруга буде близько 0.8, а для 100% RH - близько 3.4 В.

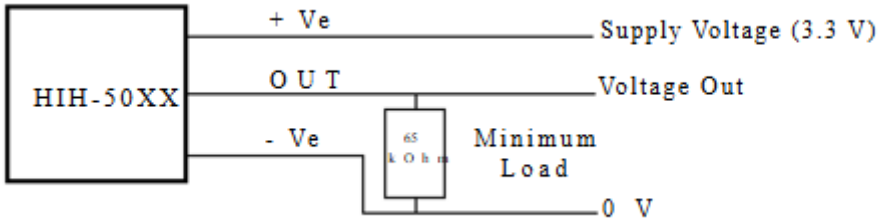

Рисунок 2.6 - Схема датчика вологості HIH-5030

Він працює при напрузі від 3.3 до 5 В, що робить його зручним для роботи з мікроконтролерами, такими як Arduino і Raspberry Pi. Точність вимірювання вологості становить ±3% RH, а точність вимірювання температури - ±2°C. Потужність датчика, що споживається, дуже низька (менше 1 мА), що робить його придатним для автономних проектів.

Однак, як і LM35, HIH-5030 видає аналоговий сигнал, що вимагає наявності АЦП отримання цифрових значень. Однак система, що створюється, передбачає широкість свого використання, можливість вбудовування будь-яких видів датчиків, можливість поєднання старих і нових підходів і

технологій. І саме для такого і були обрані аналогові датчики, щоб показати, що дана створювана система, зможе працювати з усіма елементами.

TEMT6000 [17](рис.2.7) - це аналоговий фотодатчик, який вимірює інтенсивність світла в навколишньому середовищі. Він зазвичай використовується в додатках для вимірювання освітленості, таких як автоматичне регулювання яскравості екрану, системи освітлення для розумних будинків або моніторинг умов навколишнього середовища.

Рисунок 2.7 - Модуль TEMT6000

Принцип роботи: датчик базується на використанні фотодіода, який генерує електричний струм пропорційний кількості світла, що потрапляє на датчик. Цей струм потім перетворюється в аналогову напругу, яка є вихідним сигналом датчика. Чим більше світла потрапляє на датчик, тим вища вихідна напруга. Напруга на виході може бути виміряна за допомогою мікроконтролера або комп'ютера для подальших дій.

TEMT6000 працює з живленням 3.3 В або 5 В і має високий рівень чутливості до світла, з діапазоном від темряви до яскравого денного світла. Програмно зчитане значення з TEMT6000 через АЦП можна використовувати для визначення рівня освітленості, який може бути представлений як цифрове значення. Якщо напруга на датчику велика, це означає, що рівень освітленості високий, і навпаки - якщо напруга низька, значить освітленість низька.

Також TEMT6000 застосовується в багатьох сферах, таких як автоматичне регулювання яскравості екрану або освітлення в розумних будинках, моніторинг умов навколишнього середовища або в проектах IoT для вимірювання рівня освітленості в різних умовах. Цей пристрій є простим у використанні, надійним і підходить для проектів з вимірювання світла.

2.5. Аналогово-цифровий перетворювач.

Щоб передати інформацію з аналогових датчиків, які вимірюють фізичні величини (температура, вологість, світло), у цифровому вигляді, використовується АЦП.

АЦП [17] - це пристрій, який перетворює аналоговий сигнал (безперервний) на цифровий. Така особливість важлива, оскільки обробка даних в комп'ютерах та мікроконтролерах відбувається через цифрові сигнали. Тому АЦП необхідний для того, щоб отримати дані з аналогового датчика, перетворивши його вихідну напругу на цифрову форму, зрозумілу пристроям.

Бітність АЦП - це кількість біт, що використовуються для подання аналогового значення в цифровій формі. При 10 бітній роздільній здатності АЦП аналоговий сигнал перетворюється в одне з 1024 можливих значень. Бітність впливає на точність та діапазон перетворення:

1. Чим більша бітність, тим паче точними будуть перетворення.
2. Дозвіл АЦП визначається кількістю рівнів, які може розрізняти.

Для 10-бітного АЦП роздільна здатність буде

$$D = \frac{\mathrm{V}_{REF}}{1023},$$

де V_{REF} це максимальна вхідна напруга. Це означає, що кожен крок зміни на аналоговому вході складатиме близько

$$\frac{\mathrm{V}_{REF}}{1024}.$$

Формула для розрахунку цифрового значення

$$D = \frac{\mathrm{V}_{in}}{\mathrm{V}_{REF}} \times (2^N - 1),$$

де:

– $2^N - 1$- максимальне значення цифрового коду;

– V_{in} – це аналогова вхідна напруга, яку ви подаєте на АЦП;

– V_{REF} – це максимальна напруга, яка може бути на вході АЦП, і вона задає масштаб для аналогових вимірів.

Таблиця 2.6 - АЦП N-бітні

Бітність АЦП (N)	Макс. цифрове значення $2^N - 1$	Дозвіл $\frac{\mathrm{V}_{REF}}{2^N - 1}$
8 біт	255	0.0196 В (19.6 мВ)
10 біт	1023	0.00489 В (4.89 мВ)
12 біт	4095	0.00122 В (1.22 мВ)
14 біт	16383	0.000305 В (305 мВ)
16 біт	65535	0.0000763 В (76.3 мкВ)

Чим більше бітність АЦП, тим точніше він може вимірювати аналоговий сигнал. Таким чином, з підвищенням бітності, АЦП здатен точно виявляти менші зміни напруги.

Обрано аналогово-цифровий перетворювач MCP3008B [18] системі автоматизації курника використовується безліч аналогових датчиків, таких як температурні, вологі та освітленості, які генерують аналогові сигнали. Однак Raspberry Pi обробляє лише цифрові дані, тому потрібний перетворювач. MCP3008 є оптимальним вибором завдяки своїй багатоканальної (підтримує підключення до 8 аналогових датчиків одночасно) та високій точності (10-розрядна роздільна здатність), що дозволяє отримувати деталізовані та точні дані. MCP3008 інтегрується з Raspberry Pi через інтерфейс SPI, що спрощує підключення та конфігурування. Його компактний розмір та низьке

енергоспоживання роблять його ідеальним для проектів IoT, де важливі економія місця та автономність. Використання MCP3008 дозволяє нашій системі підтримувати сумісність із різними типами датчиків, що збільшує її гнучкість та розширює функціональність.

Основні піни плати: VDD - джерело струму; VSS - Ground; VREF - референсна напруга, що визначає діапазон вимірюваних значень; CH0 - CH7 - аналогові входи, якими подаються сигнали на АЦП; DOUT - цифровий вихід, через який передаються дані з АЦП; DIN - цифровий вхід, через який подається команда на АЦП; CS/SHDN - контроль вибору чіпу або вимкнення АЦП; CLK - тактовий сигнал, який синхронізує роботу АЦП.

MCP3008 може працювати з кількома типами входів. Основні режими роботи:

1. Однофазний (Single-ended) вхід - це коли кожна аналогова напруга вимірюється щодо загального сигналу (землі). Наприклад, ви подаєте сигнал на один контакт, а інший контакт підключаєте до землі.

2. Псевдодиференціальний (Pseudo-differential) вхід- це коли вимірюється різниця між двома входами. Наприклад, ви подаєте сигнал на два контакти, і АЦП вимірює різницю між цими сигналами, що допомагає зменшити шуми та перешкоди.

MCP3008 працює так, що може використовувати 8 однофазних входів або 4 псевдодиференціальні пари. Коли працює псевдодиференціальний режим, один вхід у парі називається IN+, а інший IN-. Напруга на IN+ вимірюється щодо IN-. Це дозволяє знизити вплив шуму, який може бути на обох входах.

Доцільно розглянути приклад роботи.

Якщо напруга на IN+ менша, ніж на IN-, то АЦП видасть результат 0 (код 000h).

Якщо напруга на IN+ набагато більша, ніж на IN-, то АЦП видасть максимальний результат (код 3FFh).

2.6 Інші елементи системи.

Для керування виконавчими механізмами, приладів, що працюють на високий напрузі або струмах потрібно реле. Тому обрано звичайний модуль реле з оптичною ізоляцією на 5V [19]. Він використовується для забезпечення безпеки між низьковольтною та високовольтною стороною, запобігаючи заземлюючим петлям та захищаючи мікроконтролер від пошкоджень. Наприклад, прилади, що працюють на змінному струмі 250В (10А) або постійному струмі 30В (10А). Він працює при напрузі 5В та вимагає струму від 15 до 20 мА для активації реле. Модуль використовує оптопару для ізоляції, що забезпечує безпеку, запобігаючи заземлюючим петлям при підключенні до мікроконтролера і захищає низьковольтну сторону від високої напруги.

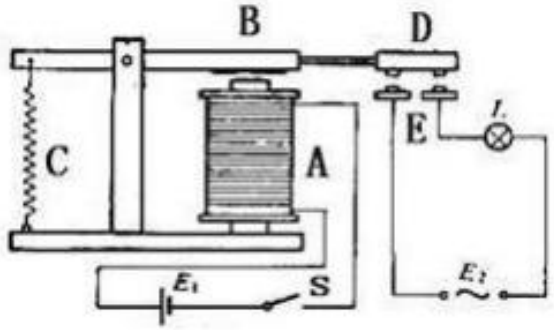

Рисунок 2.8 - Модуль реле з оптичною ізоляцією на 5V

На зображенні показані компоненти пристрою: A - електромагніт, B - якір, C - пружина, D - рухомий контакт, E - нерухомі контакти. Є два нерухомі контакти: нормально замкнутий і нормально відкритий.

Коли котушка не подає напругу, нормально відкритий контакт (NO) залишається в розімкнутому стані, а нормально замкнутий контакт (NC) - замкнутим. долаючи опір пружини, і якір замикає рухливий контакт із нормально відкритим (NO) контактом. нормально замкнутий контакт (NC) розмикається. Після того як котушка знеструмлюється, зникає електромагнітне поле, і якір повертається у вихідне положення, розмикаючи контакт із нормально відкритим (NO) контактом та замикаючи нормально замкнутий

(NC) контакт. включення та вимикання ланцюга. Тобто, щоб активувати реле, його сигнальний вхід подається низький рівень (LOW). Коли вхідний сигнал LOW, реле вмикається, і індикатор на виході загоряється.

Модуль може бути безпосередньо підключений до мікроконтролерів, таких як Arduino, AVR, PIC, ARM, MSP430 і керуватися стандартними TTL логічними сигналами.

При підключенні модуля Raspberry Pi з 3.3В логікою, можливо, потрібно використовувати окреме живлення для реле, щоб забезпечити стабільну роботу модуля, так як деякі реле можуть працювати нестабільно при підключенні до 3.3В сигналів. Модуль також може використовуватися з іншими типами реле, такими як 2-канальні, твердотільні реле або реле з більшою потужністю для більш складних проектів.

Рисунок 2.9 - Реле 5В 10А низького рівня

Для моніторингу курятнику у реальному часі, обрано Camera Module 3 NoIR [20].

Raspberry Pi Camera Module 3 - це компактна камера, що працює з Raspberry Pi, оснащена датчиком Sony IMX708 з роздільною здатністю 12 мегапікселів. Вона має такі особливості, як фазове виявлення фокусу (PDAF), що дозволяє швидко фокусуватися, а також HDR-режим для покращення якості зображень при різних умовах освітлення. Камера підтримує різні варіанти об'єктивів: стандартний, з широким кутом огляду та без інфрачервоного фільтра (NoIR), що дозволяє використовувати її для нічного бачення. Вона здатна знімати фотографії з роздільною здатністю 4608 × 2592

пікселів і відео з роздільною здатністю до 1080p при 50 кадрах в секунду. Камера має високу чутливість до світла завдяки використанню бек-освітленого CMOS сенсора і забезпечує гарну якість зображення навіть за поганого освітлення.

Рисунок 2.10 - Camera Module 3 NoIR

Усі моделі підтримують підключення через CSI (Camera Serial Interface) і використовують кабель з 15 пінів для підключення до Raspberry Pi. Камера також сумісна з усіма моделями Raspberry Pi, за винятком деяких ранніх моделей Raspberry Pi Zero, для яких потрібно використовувати адаптер.

Ця камера підтримує роботу з бібліотекою libcamera, що дозволяє програмістам легко працювати з нею на низькому рівні, включаючи функцію швидкого фокусування. Вона працює при температурі від 0°C до 50°C і має компоновку, що дозволяє використовувати її в різних умовах, зокрема в порівнянні з попередніми моделями, вона має дещо збільшену висоту через поліпшену оптику. Завдяки можливості швидкого фокусування та різним режимам зйомки, Raspberry Pi Camera Module 3 можна налаштувати для автоматичної зйомки фотографій через інтервали часу, наприклад, раз на 5 секунд. Скрипти на Python разом з бібліотекою libcamera дозволяють налаштувати камеру для регулярного знімання фото через визначені інтервали, що корисно для проекту.

Для живлення всієї системи підберемо блок живлення 5V 5A 25Вт (IP20) MC - це універсальний блок, який забезпечує стабільне електроживлення для різних електронних пристроїв. Він має вихідну напругу 5В і здатен подавати струм до 5А, що дозволяє використовувати його для проектів з високою потужністю, таких як Raspberry Pi, LED-освітлення та інші елементи, що потребують стабільного живлення.

Цей блок живлення здатен забезпечити потужність до 25Вт, що робить його придатним для підключення декількох пристроїв одночасно, зокрема для проектів в галузі Інтернету речей, де потрібно живити кілька компонентів від одного джерела. Інтерфейс IP20 означає, що блок живлення має стандартний захист від пилу та води, що робить його підходящим для використання в приміщеннях, не піддаючись впливу вологи чи великої кількості пилу.

До переваг цього блоку живлення можна віднести:

1. Висока потужності пристрою, що потребує великої кількості енергії, працюють безперебійно.

2. Великий запас струму до 5А, що є важливим для проектів з багатьма периферійними пристроями.

3. Простоту використання в приміщеннях завдяки захисту від пилу та води (IP20).

4. Компактні розміри та високу енергоефективність, що дозволяє використовувати блок живлення в обмежених просторах.

Цей блок живлення - ідеальне рішення проекту, що працює з низьковольтними електронними проектами, де потрібна стабільність і ефективність.

У ході аналізу створено структурну схему (рис. 2.12) системи, яка включає ключові компоненти та їх взаємодію. Основною метою нашої системи було забезпечення автоматизації збору даних, їх обробки та передачі для інтегрованого управління за допомогою Raspberry Pi та периферійних пристроїв.

Рисунок 2.11 - Перетворювач змінної напруги 220 В постійну 5 В потужністю 25Вт

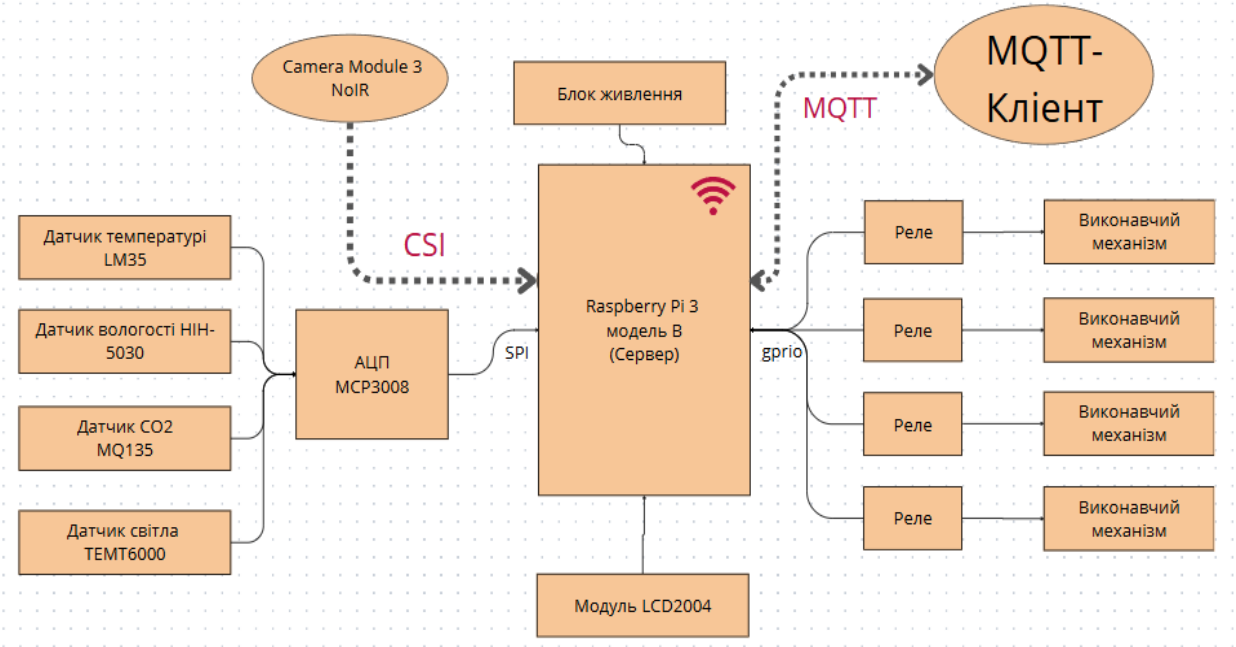

Рисунок 2.12 - Структурна схема проекту

Висновок до 2 розділу

У розділі розглянуто вибір протоколів зв'язку та технологій для автоматизації курника. Для забезпечення бездротового зв'язку між Raspberry Pi та іншими пристроями вибрано Wi-Fi, оскільки він забезпечує стабільне з'єднання з мінімальними енергетичними витратами. Для передачі даних використано протокол MQTT, який є легким і ефективним для обміну

повідомленнями в реальному часі, що дозволяє швидко отримувати та передавати інформацію між пристроями. Для збору та обробки даних обрано сервер Mosquitto, який підтримує стабільний зв'язок через MQTT, мінімізуючи затримки і забезпечуючи надійний обмін даними.

Для зчитування аналогових сигналів від сенсорів використовується АЦП MCP3008, що перетворює аналогові значення в цифрові. Raspberry Pi здатен обробляти інформацію з різних сенсорів, таких як температура, вологість, освітленість та інші, завдяки цьому. Для управління освітленням і вентиляцією в курнику застосовуються реле, що дозволяють автоматизувати включення і вимкнення пристроїв на основі отриманих даних.

Система живиться від блоку живлення 5V 5A 25Вт (IP20), що забезпечує стабільну роботу всіх компонентів, зокрема Raspberry Pi, АЦП, реле та інших периферійних пристроїв. Блок живлення надає необхідну потужність для підтримки безперебійної роботи всієї системи.

РОЗДІЛ 3
РОЗРОБКА ТА ВИПРОБУВАННЯ ПРОГРАМНИХ РІШЕНЬ ДЛЯ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПТАШНИКОМ

3.1. Моделювання роботи системи у Proteus

Proteus - це потужна платформа для моделювання електронних схем та симуляції роботи пристроїв. Програма дозволяє створювати схеми з використанням реальних компонентів, а також тестувати їх у віртуальному середовищі, що допомагає виявити помилки перед тим, як схема буде зібрана в реальному житті. У Proteus можна [21]:

1. Розробляти схеми з компонентами (резистори, транзистори, мікроконтролери, датчики, дисплеї тощо).
2. Моделювати їхню взаємодію.
3. Симулювати роботу мікроконтролерів та інших програмованих пристроїв.
4. Тестувати алгоритми роботи програмного забезпечення.

У рамках розробки автоматизованої системи для птахокомплексу Proteus використовуватиметься для симуляції роботи всієї електроніки перед її реалізацією на реальному устаткуванні. Можна переконатися, що компоненти правильно з'єднані та функціонують один з одним. Протестувати код, який буде працювати на мікроконтролері та переконатися, що все функціонує відповідно до вимог. Ранній етап розробки включає в себе прийняття заходів для попередження помилок, таких як помилки в алгоритмах і неправильне підключення компонентів.

А головне, симуляція дозволяє перевірити схему та код до того, як буде витрачено час та гроші на складання фізичного пристрою.

Схема(рис.3.1) автоматизованого курника на базі Raspberry Pi 3 з погляду взаємодії всіх компонентів. Центральним елементом виступає Raspberry Pi 3, до якого підключений LCD-дисплей через шину I2C (піни SDA та SCL), що

відображає поточні параметри мікроклімату - температуру (T), вологість (H), рівень CO2 та освітленість (Light). Система розділена на чотири функціональні блоки: управління обігрівом, вологістю, вентиляцією та освітленням.

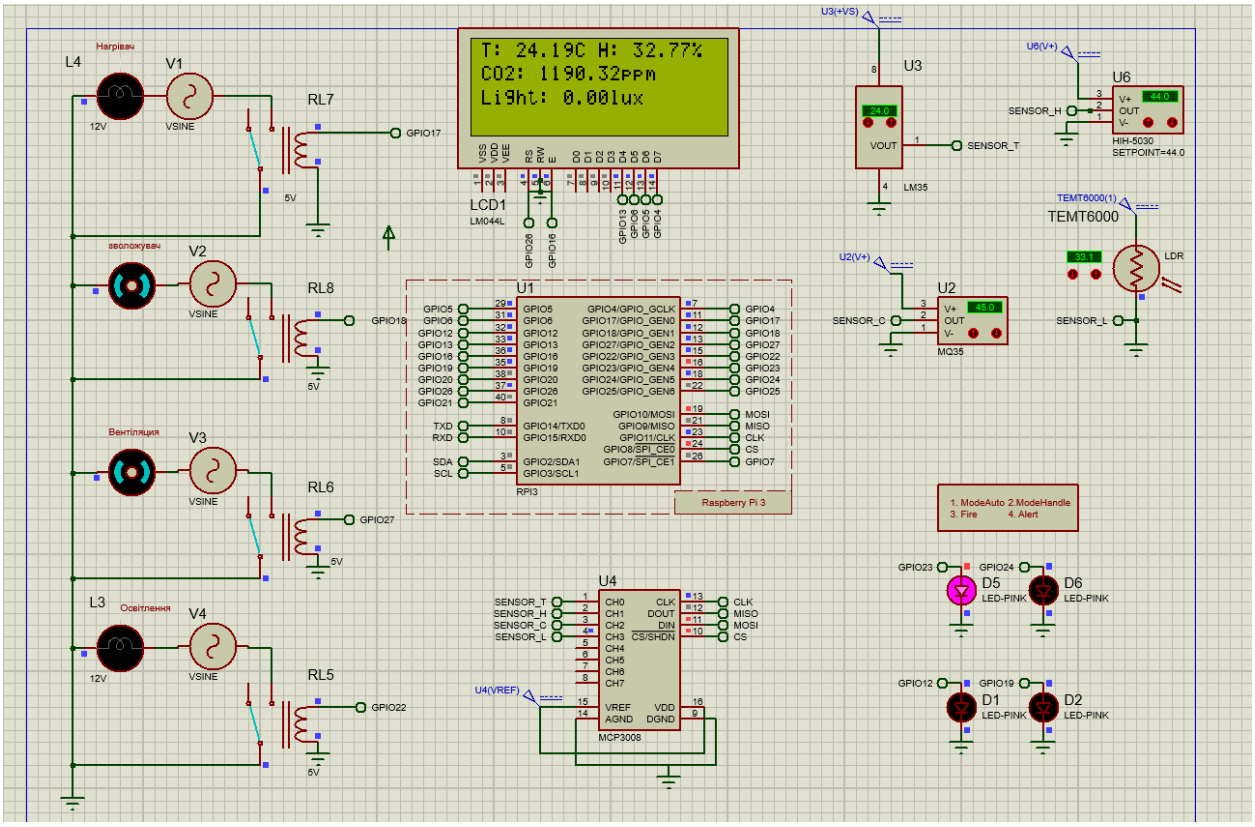

Рисунок 3.1 - Зібрана симуляція схеми АСУ

Для вимірювання параметрів використовуються різні датчики, підключені до MCP3008 (аналогово-цифровий перетворювач) через інтерфейс SPI. MCP3008 перетворює аналогові сигнали з датчиків на цифровий формат для обробки контролером.

Всі релейні модулі підключені до GPIO-пін Raspberry Pi, що дозволяє програмно керувати включенням та вимкненням відповідних пристроїв.

Для захисту від перешкод та стабільної роботи кожен релейний модуль має захисний діод. Система автоматично підтримує задані параметри мікроклімату, включаючи чи вимикаючи відповідні пристрої з урахуванням показань датчиків.

Наприклад, якщо температура опускається нижче заданої, включається обігрівач, при низькій вологості активується зволожувач, за підвищеного рівня CO_2 запускається вентиляція, а при недостатньому природному освітленні включається штучне освітлення.

Важливо відзначити, що всі датчики підключені до загальної шини живлення та землі, а сигнали, що управляють, йдуть на окремі піни GPIO Raspberry Pi через MCP3008, що забезпечує надійне зчитування аналогових показань та їх перетворення в цифровий формат для подальшої обробки.

Розглянуто роботу АЦП більш детально в симуляції (рис.3.2). На осцилограмі виходів АЦП відображається процес перетворення аналогового сигналу (напруги) з датчика цифровий формат, який може обробляти мікроконтролер. Розглянемо детально, що відбувається: бачимо три основні сигнали на осцилограмі - тактовий сигнал синхронізації (жовта лінія вгорі), сигнал вибірки-зберігання (блакитна лінія посередині) та власне вихідні дані АЦП (червона лінія внизу).

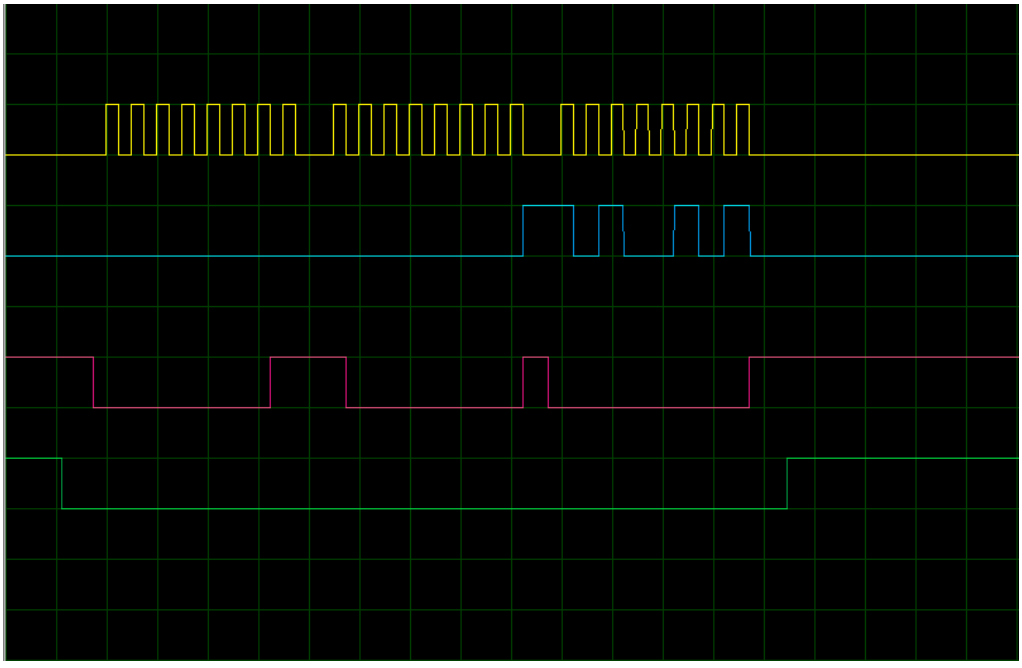

Рисунок 3.2 - Осцилограма виходів АЦП

Тактовий сигнал забезпечує синхронізацію роботи всієї схеми, з його переднього фронту відбувається зчитування даних. В даному прикладі АЦП

перетворює вхідну напругу в число 165, яке в двійковому вигляді представляється як 10100101. Це перетворення відбувається послідовно: спочатку АЦП фіксує вхідну напругу за допомогою схеми вибірки-зберігання (sample-and-hold), що відображається на синій лінії осцилограми, потім починається процес послідовного перетворення, при якому кожен біт результату формується за черги. У цьому значення кожного біта визначається порівнянням вхідної напруги з опорними рівнями всередині АЦП. В цьому випадку, старший біт (MSB) дорівнює 1, що відповідає 128 у десятковій системі ($1 \cdot 2^7$), потім йде 0 ($0 \cdot 2^6$), потім 1 ($1 \cdot 2^5 = 32$) і так далі, доки не сформується повний 8-бітний код 10100101, що у сумі дає 165 (128+32+4+1). Червона лінія на осцилограмі показує, як ці біти з'являються на виході АЦП.

Дані на виході АЦП стають стабільними саме під час високого рівня тактового сигналу, що забезпечує надійне зчитування інформації мікроконтролером. Це перетворення необхідно, оскільки мікроконтролер може працювати лише з цифровими сигналами, а датчики видають аналогові значення напруги [23].

Після перетворення мікроконтролер може використовувати отримане цифрове значення для подальших обчислень, відображення на дисплеї або прийняття рішень у системі керування.

Такий процес перетворення постійно повторюється з певною частотою, щоб система мала актуальні дані з датчиків. Таймінги сигналів на осцилограмі показують, що перетворення відбувається досить швидко, що дозволяє обробляти безліч вимірювань за секунду та забезпечувати високу точність контролю параметрів у системі автоматизації.

Програмна частина проекту зроблена за допомогою мови програмування Python. Цей вибір обумовлено сумісністю Python із Raspberry Pi та його багатими бібліотеками для роботи з камерами, обробки зображень, аналізу даних, взаємодії з датчиками та комунікації з іншими модулями [22].

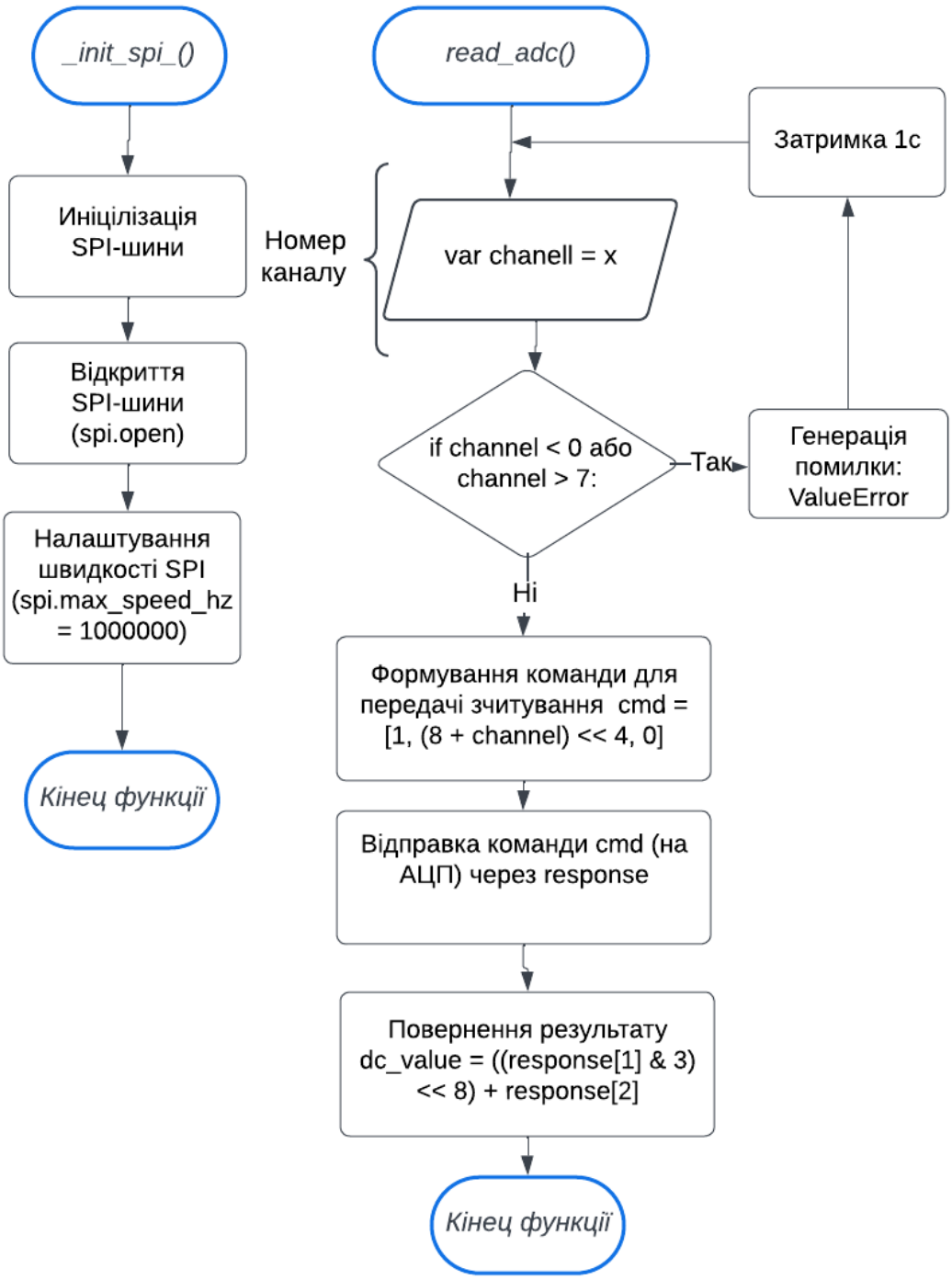

Рисунок 3.3 - Алгоритми функцій всередині классу ADCReader.

На рис. 3.3, представлено алгоритми двух функцій всередині классу ADCReader. Перший алгоритм - Функція _init_spi_() в класі ADCReader. Вона забезпечує ініціалізацію SPI-шини для зв'язку з аналого-цифровим

перетворювачем. На першому етапі виконується підключення до SPI-інтерфейсу, після чого відкривається SPI-шина за допомогою методу spi.open(). Наступним кроком встановлюється максимальна швидкість передачі даних, spi.max_speed_hz = 1000000, що забезпечує необхідну продуктивність для роботи АЦП. Після виконання цих дій SPI-шина готова до використання, і функція завершує свою роботу. Другий алгоритм - функція read_adc(). Вона виконує зчитування даних з указаного каналу аналого-цифрового перетворювача (ADC). Спочатку вона приймає номер каналу, наприклад, channel = x, і перевіряє його на коректність. Якщо номер каналу не входить в діапазон від 0 до 7, генерується помилка ValueError. Якщо канал валідний, формується команда для передачі даних, де cmd = [1, (8 + channel) << 4, 0]. Ця команда відправляється на АЦП через SPI, а результат зчитується у змінну response. Після цього обчислюється цифрове значення результату зчитування, де dc_value = ((response[1] & 3) << 8) + response[2]. Функція завершується поверненням значення dc_value.

Функція _init_spi_() виконується тільки один раз при створенні об’єкту классу ADCReader. Функція read_adc() виконуються циклічно головному циклі системи.

Алгоритм циклічного управління системою керування абіотичними факторами (рис.3.4) починається з зчитування даних із сенсорів, які фіксують значення параметрів. Отримані значення перевіряються відповідно до вибраного режиму роботи: у автоматичному режимі система порівнює їх із заданими порогами, а в ручному використовуються параметри, задані користувачем. Потім дані передаються через протокол MQTT для їхньої обробки або зберігання на сервері.

Потім оновлюється дисплей LCD, на якому відображаються актуальні параметри абіотичних факторів та стану системи. Далі виконується затримка в 1 секунду для забезпечення циклічності роботи. Після цього алгоритм повертається до зчитування даних із сенсорів, і процес повторюється.

Алгоритм керування абіотичними факторами (рис.3.5). Ця блок-схема описує алгоритм управління виконавчими пристроями з урахуванням перевірки параметрів і заданого режиму роботи. Алгоритм починається з виклику функції control_actuators(), після чого система отримує поточний режим роботи та повертає значення параметра mode, який визначає граничні значення або умови роботи. Потім зчитуються поточні параметри X, які можуть містити дані навколишнього середовища, такі як температура, вологість або рівень CO_2.

Далі система перевіряє, чи поточний параметр X відповідає умовам, заданим для режиму mode. Якщо X менше значення mode, то включається виконавчий пристрій, наприклад, нагрівач, вентилятор або зволожувач, і надсилається повідомлення користувачеві або системі про зміну стану, що відбулася. Якщо параметр X більший за значення mode, то теж вмикається специальний виконавчий пристрій, потім система повертається до повторної перевірки параметрів, очікуючи відповідних умов. Цей процес циклічно повторюється, забезпечуючи постійний моніторинг параметрів та автоматичне управління виконавчими пристроями підтримки заданого мікроклімату.

Алгоритм виявлення пожежі(рис.3.6) на основі зображень включає кілька основних етапів, які дозволяють ефективно виявляти ознаки пожежі за допомогою камери та аналізу отриманих зображень. Спочатку починається сама ініціалізация системи, згодом отримаємо зображення з камери, яка розтаашована у курнику, зібрані зображення потім передаються на сервер та до фото курника буде застосовуватися алгоритм машинного навачання, У разі виявлення пожежі система автоматично активує відповідні дії, зокрема відправляє сповіщення на сервер, мобільний телефон. Після цього процес виявлення повторюється із затримкою у 5с для постійного моніторингу і оновлення даних. З таким алгоритмом зменшується рівень небезпеки, тому що система може оперативно зреагувати на загрозу.

Рисунок 3.4 - Алгоритм циклічного управління системою керування абіотичними факторами

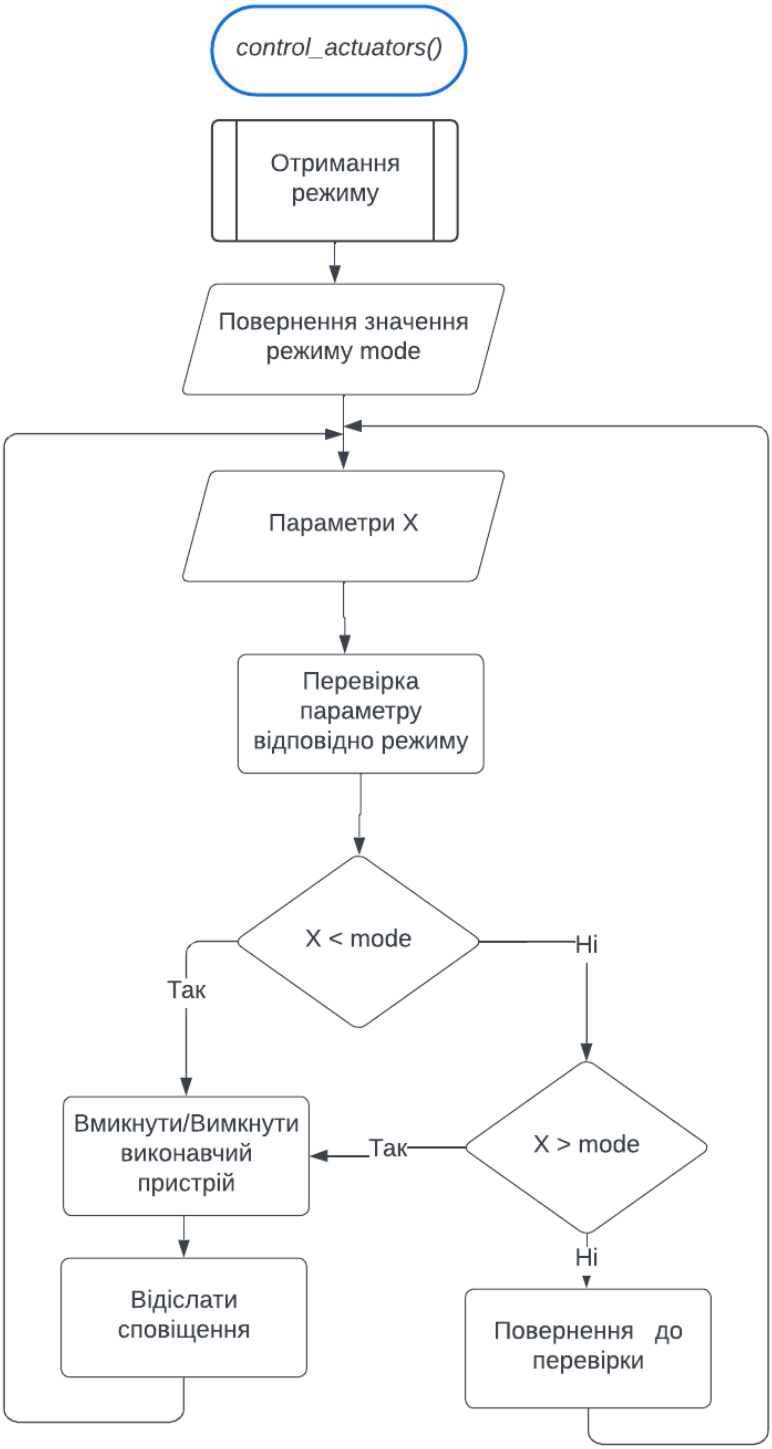

Рисунок 3.5 - Алгоритм керування абіотичними факторами

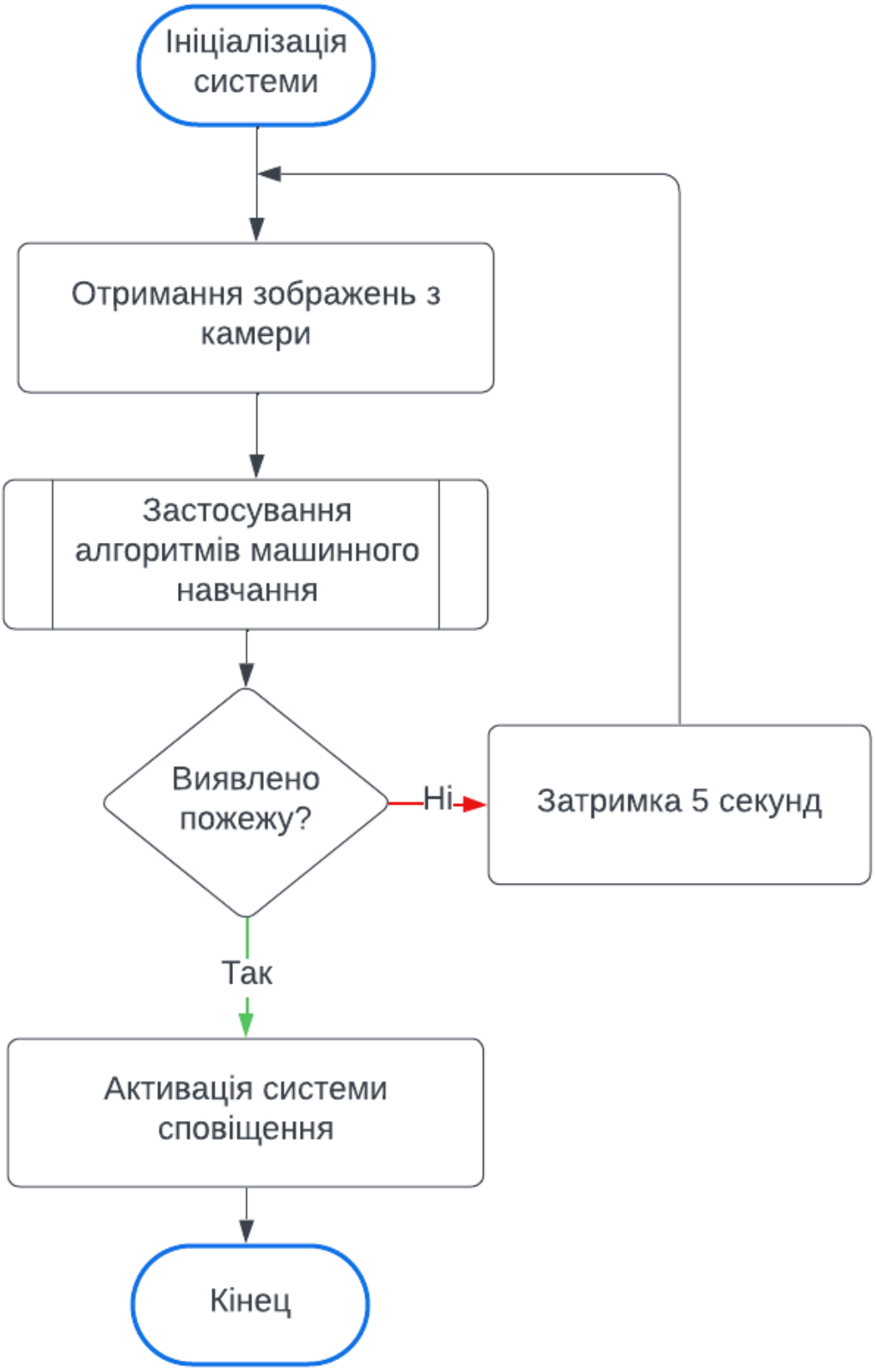

Рисунок 3.6 - Алгоритм виявлення пожежі

3.2 Роль брокера у забезпеченні роботи системи автоматизації та його функції.

Для налаштування брокера Mosquitto для доступу з локальної мережи, виконано установку, конфігурацію та перевірку роботи. Встановлено Mosquitto на Raspberry Pi 3 виконавши команди sudo apt update та sudo apt install mosquitto mosquitto-clients. Після установки сервіс Mosquitto автоматично запускається, і одразу перевіряється його статус(рис.3.7) за допомогою команди sudo systemctl status mosquitto.

```
Имя_службы: mosquitto
        Тип                : 10  WIN32_OWN_PROCESS
        Состояние          : 4  RUNNING
                              (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        Код_выхода_Win32   : 0  (0x0)
        Код_выхода_службы  : 0  (0x0)
        Контрольная_точка  : 0x0
        Ожидание           : 0x0
```

Рисунок 3.7 - Запуск сервера Mosquitto

Служба Mosquitto є брокером повідомлень MQTT, який працює як самостійний процес у системі, що вказано типом WIN32_OWN_PROCESS. Поточний стан служби - RUNNING, що означає, що служба запущена і працює без збоїв. Вона підтримує такі операції: STOPPABLE (службу можна зупинити), NOT_PAUSABLE (не можна призупинити), ACCEPTS_SHUTDOWN (служба завершить роботу при вимкненні системи). Код виходу Win32: 0 (0x0) вказує на відсутність помилок під час виконання операцій Windows. Аналогічно, код виходу служби 0 (0x0) свідчить про те, що служба працює без помилок. Контрольна точка зі значенням 0x0 підтверджує, що процес запуску служби завершений успішно. Очікування також відсутнє, що означає, що немає затримок чи пауз у виконанні операцій. Mosquitto працює безперебійно, готова до прийому підключень клієнтів для передачі даних через MQTT.

Як говорилося раніше, підключення топіків у брокері Mosquitto здійснюється через процес підписки та публікації повідомлень. Брокер Mosquitto дозволяє різним пристроям обмінюватися даними, використовуючи ієрархічну структуру топіків. Кожен топік є міткою, до якої можуть підписуватися клієнти для отримання повідомлень або публікувати дані.

Для підключення до топіків використовуються дві основні утиліти: mosquitto_sub для підписки і mosquitto_pub для публікації повідомлень.

Наприклад, щоб підписатися на топік "sensor / temperature", можна використовувати команду `mosquitto_sub -h 192.168.X.XXX -t sensor / temperature`. Це дозволить клієнту отримувати всі повідомлення, які будуть опубліковані в цей топік. Для публікації повідомлення в той же топік використовується команда `mosquitto_pub -h 192.168.X.XXX -t sensor/temperature-m"23 Градуса". Після виконання цієї команди повідомлення "23 Градуса" буде відправлено в топік "sensor / temperature", і всі клієнти, підписані на нього, отримають це повідомлення.

Таким чином, підключення топіків у брокері Mosquitto здійснюється через процес підписки та публікації повідомлень, що дозволяє ефективно організувати обмін даними між пристроями у системі автоматизації.

Таблиця 3.1 - Поля с сенсорами у проекті.

sensor/temperature	Дані температури	Float (°C)
sensor/humidity	Дані вологості	Float (%)
sensor/co2	Рівень CO_2	Integer (ppm)
sensor/light	Освітленість	Integer (люкс)

Таблиця 3.2 - Поля з виконавчими механізмами в проекті

actuator/temperature	Управління обігрівом	Boolean (ON/OFF)
actuator/humidity	Управління зволожувачем	Boolean (ON/OFF)
actuator/co2	Управління вентиляцією	Boolean (ON/OFF)
actuator/light	Управління освітленням	Boolean (ON/OFF)

Таблиця 3.3 - Поля з режимами

ModeAuto	Перший режим	Boolean (ON/OFF)
ModeHandle	Ручний режим	Boolean (ON/OFF)

Таблиця 3.4 - Поле для аварійних сигналів

Stop	Аварійне оповіщення	Boolean (Yes/No)
Camera/fire	Аварійне оповіщення	Boolean (Yes/No)
Error	Аварійне оповіщення	Boolean (Yes/No)

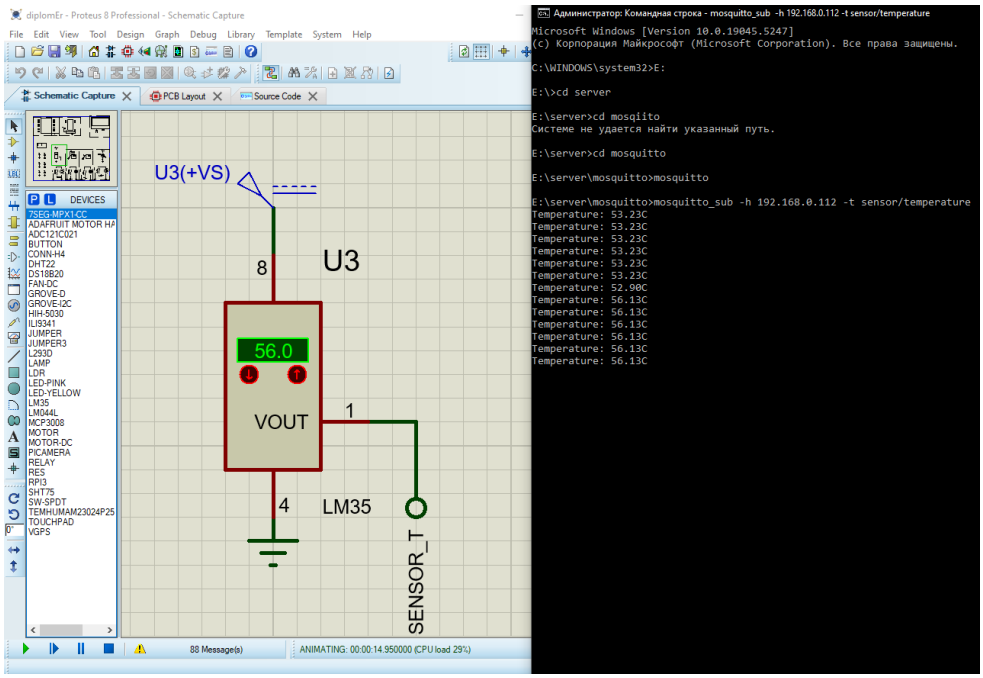

Рисунок 3.8 - Тест роботи сервера.

Для більш безпечної конфігурації підключимо автентифікацію. Щоб увімкнути автентифікацію, створимо файл паролів та введемо дані. Змінимо конфігурацію, щоб відключити анонімний доступ і вказати файл паролів, додавши рядки

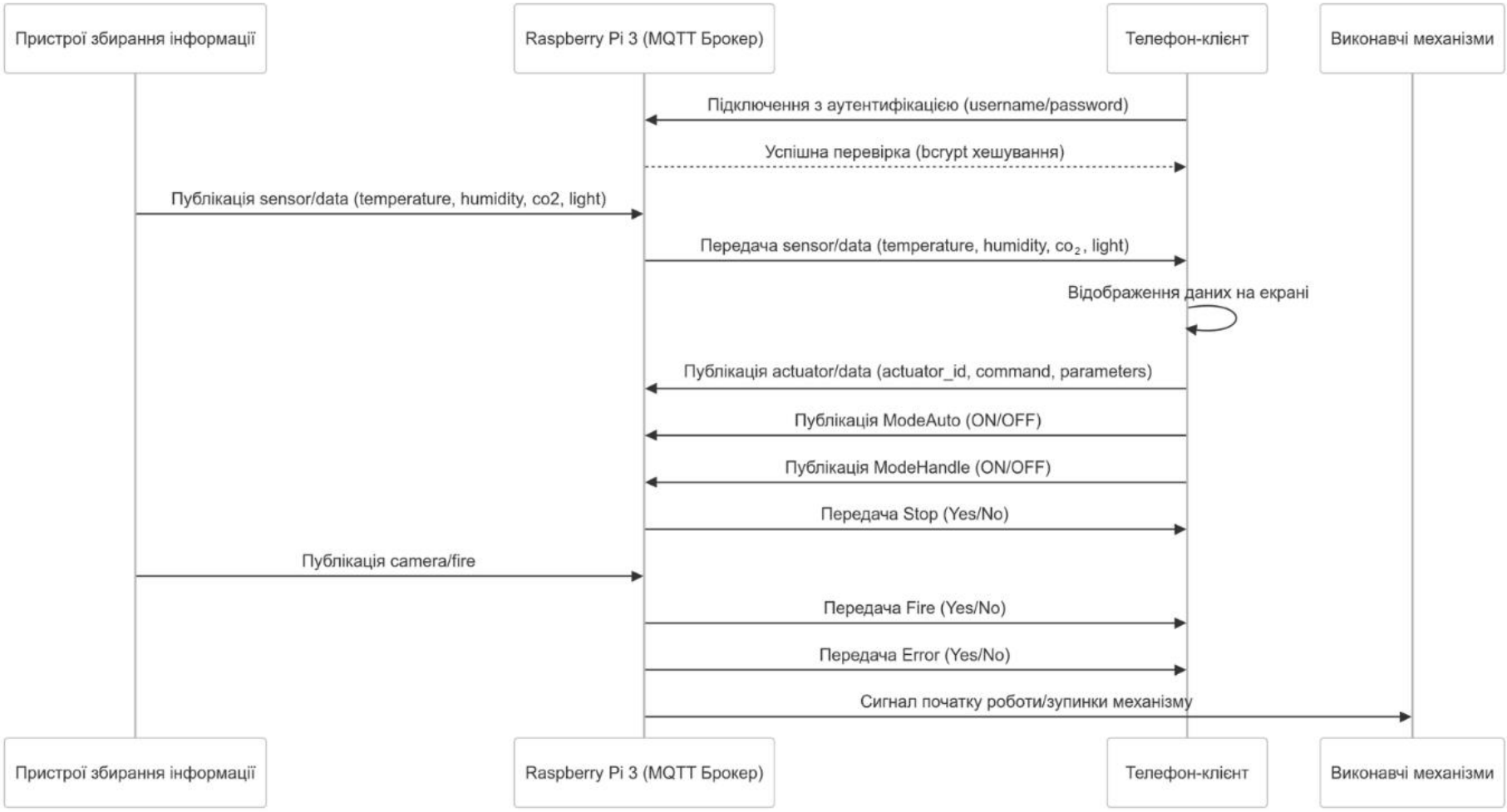

Рисунок 3.9 - Схема взаємодії MQTT-протоколу

Схема(рис.3.9) ілюструє взаємодію компонентів системи автоматизації управління абіотичними факторами завдяки MQTT-протокол. У системі задіяно чотири основні компоненти: пристрої збору інформації (датчики), Raspberry Pi 3 як MQTT-брокер, клієнтський додаток на телефоні та виконавчі механізми.

Датчики збирають дані про навколишнє середовище. Ці дані надсилаються на Raspberry Pi 3 через протокол MQTT. Raspberry Pi передає отримані дані клієнтському додатку, де вони відображаються на екрані користувача.

Підключення клієнтського додатку до MQTT-брокера вимагає автентифікації за допомогою імені користувача та пароля, перевірка яких здійснюється за алгоритмом хешування bcrypt. Після успішної автентифікації клієнт може взаємодіяти із системою.

Користувач через додаток може керувати виконавчими механізмами, надсилаючи команди через MQTT. Команди містять ідентифікатор механізму, дію (наприклад, увімкнення чи вимкнення) та параметри. Також передбачено два режими роботи: автоматичний (ModeAuto), у якому система самостійно регулює параметри, та ручний (ModeHandle), що дозволяє користувачу керувати механізмами напряму.

У разі виникнення позаштатних ситуацій система надсилає сповіщення. Наприклад, у випадку виявлення пожежі публікується повідомлення через канал camera/fire, а у разі помилки надсилається сповіщення через канал Error. Ці повідомлення обробляються клієнтським додатком та передаються виконавчим механізмам для виконання відповідних дій

Додатково передбачено сигнали для зупинки або запуску роботи механізмів, які також передаються через MQTT. Таким чином, схема описує цикли збору даних, управління системою та реагування на події в автоматизованому середовищі.

Розглядаючи безпечну конфігурацію у схемі вище, розберемо роботу автентифікації.

Щоб увімкнути автентифікацію, створимо файл паролів та введемо дані. Змінимо конфігурацію, щоб відключити анонімний доступ і вказати файл паролів, додавши рядки

#allow_anonymous false

#password_file E:\server\mosquitto\passwd

У папці, як видно, знаходиться логін та зашифрований пароль. Він згенерований за допомогою алгоритму хешування bcrypt. Та тепер для доступу до сервера треба доповнювати виклик топіка mosquitto_sub -h 192.168.0.112 -t "test/topic" -v -u "dariia" -P "1234".

Рисунок 3.10 - Логін та хешуванням пароля для сервера MQTT

Цей криптографічний алгоритм хешування паролів [34] розроблений для забезпечення високого рівня безпеки. Його основна мета - уповільнити процес хешування, щоб зробити атаки грубою силою непрактичними навіть із використанням сучасних обчислювальних потужностей. Алгоритм має адаптивну складність, яка дозволяє збільшувати час виконання хешування в майбутньому для протидії зростаючій потужності атакуючих. Він автоматично генерує випадкову сіль для кожного пароля. Сіль у цьому випадку - це випадкова послідовність байтів, яка додається до пароля перед його хешуванням. Вона забезпечує унікальність кожного хешу, навіть якщо паролі збігаються. Це захищає від атак на основі попередньо обчислених таблиць (rainbow tables - кольорова таблиця обчислень хешу, використовуються для підбору пароля).

Коли створюється новий пароль на сервері Russbery - BCrypt генерує випадкову сіль довжиною 128 біт (16 байтів). Далі задається параметр складності (cost factor), який контролює кількість повторень алгоритму. Цей

параметр зазвичай задається у вигляді ступеня двійки (наприклад, значення 10 означає, що алгоритм буде виконуватися $2^{10} = 1024$ рази). Пароль об’єднується із сіллю, після чого застосовується алгоритм на основі шифрування Blowfish. Процес включає виконання етапів розширення ключа та шифрування, з повтореннями, що залежать від параметра складності. Результатом роботи є хеш довжиною 184 біти (23 байти), який містить інформацію про сіль, параметр складності та сам хеш пароля.

BCrypt зберігає сіль у самому хеші разом із параметром складності, що дозволяє автоматично витягувати ці дані під час перевірки пароля. Формат збереженого хешу включає префікс алгоритму (наприклад, $2b$), параметр складності, сіль (22 символи) і хеш пароля. Завдяки цьому немає потреби зберігати сіль окремо, оскільки вона вже вбудована в результат хешування. Ця риса значно спрощує процес використання алгоритму. Користувач при введенні пароля передає його для перевірки, і алгоритм автоматично витягує сіль і параметр складності з хешу, обчислює новий хеш і порівнює його з збереженим.

Найвища надійність цього алгоритму досягається завдяки додаванню "солі" та налаштовуваний складності через кількість ітерацій. Це робить його стійким до атак методом підбору та радіусним таблицям.

3.3 Вибір технології, функціонал та принцип роботи мобільного додатка для керування системою

Для створення інтуїтивно зрозумілого інтерфейсу управління птахофермою було обрано мову програмування Kotlin. Відзначено певні переваги вибраного рішення. По-перше, Kotlin має дуже простий і зрозумілий синтаксис, що сприяє покращенню читання коду та забезпеченню зручної підтримки. Крім того, мова забезпечує безпеку типів, а це дозволяє уникнути розповсюджених питань із NullPointerException. Тіже, Kotlin має функціональне програмування та надає засоби для роботи з даними, що в свою

чергу дозволяє швидко нескладно інтегрувати IoT-пристрої. Через повну сумісність з Java, можна використати будь-яку Java-бібліотеку, що пришвидшить роботу. Таким чином, Kotlin зручна та проста мова програмування для створення інтерфейсу керування та моніторингу птахокомплексом. Розробка інтерфейса буде здійснюватися на операційну систему Android, яка має безліч переваг. Це система, яка залишається однією з найбільш потрібних серед мобільних користувачів. Підтримує запуск програмного забезпечення як на смартфонах, так і на планшетах. Та є доступна багата екосистема інструментів, що сприяє легкій розробці та інтеграції з IoT-пристроями. Зазначимо, що Android OS підтримує Kotlin та Java, що дає розробникам гнучкість у виборі технологій. Що може дозволити адаптувати програму під різні проекти. Це дозволяє власникам птахоферм та іншим людям легко керувати своїми системами за допомогою звичних пристроїв.

Працювати над проектом обрано у Android Studio. Середовище розробки підтримує різні API, спрощуючи взаємодію додатків з пристроями інтернету речей. Наприклад, розробники можуть легко інтегрувати API для роботи з Bluetooth, Wi-Fi та іншими протоколами зв'язку, що дозволяє підключати та керувати такими IoT-пристроями, як датчики температури та вологості.

Варто зазначити, що проект працює з Git [24] - системою керування версіями. Вона дозволяє розробникам відстежувати зміни в коді, управляти різними версіями файлів та ефективно співпрацювати з іншими учасниками проекту.

Основні переваги Git полягають у його здатності зберігати повну історію змін, що дає можливість легко повертатися до попередніх версій, а також у можливості працювати з локальними репозиторіями, що забезпечує високу швидкість і гнучкість. Git функціонує завдяки кільком деяким концепціям. 1. По суті, репозиторій є місцем для зберігання всіх версій проекту. Та коміти являють собою знімки стану проекту в певний момент часу.

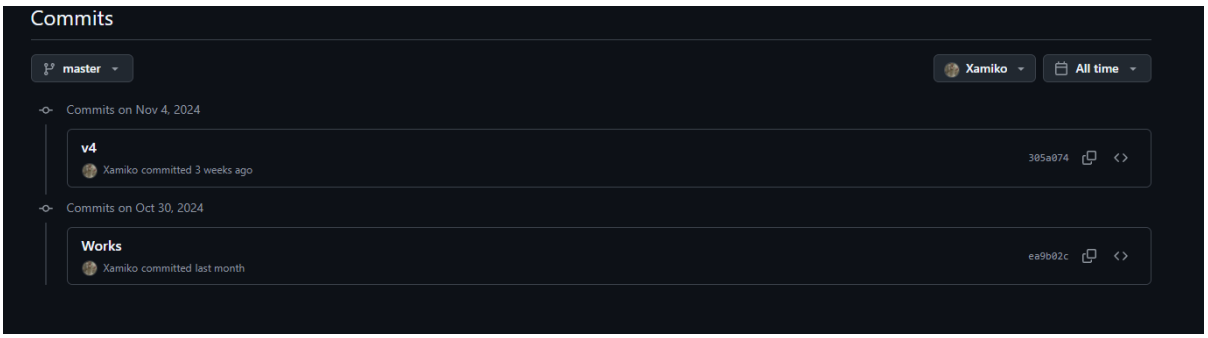

Рисунок 3.11 - Репозіторій проекту на GitHub

Проект додатка буде розміщений у цьому репозиторії. Заохочується участь усіх зацікавлених у його розвитку: пропозиції щодо модернізації, внесення нового функціоналу, виправлення помилок і активність у розділі Issues. Будь-який внесок сприятиме покращенню та вдосконаленню проекту.

У проекті використано Jetpack Compose для створення інтерфейсу користувача. Jetpack Compose дозволяє розробникам зосередитися на декларативному підході до створення інтерфейсу користувача, де кожен компонент визначається як функція. Це спрощує підтримку коду, адже зміни в стані автоматично відображаються в інтерфейсі без потреби в ручному оновленні. Завдяки цьому підходу зменшується кількість шаблонного коду, а розробка стає більш зрозумілою та структурованою.

У проекті використано ключові компоненти Compose, такі як Scaffold, TopAppBar, LazyColumn та інші, що забезпечують зручне створення структурованих інтерфейсів. Також інтегровано системи управління станом, такі як State та LiveData, для забезпечення реактивного оновлення даних у реальному часі. Це особливо важливо для таких функцій, як динамічний контроль за параметрами середовища, моніторинг показників сенсорів або отримання повідомлень через MQTT.

Додатково, використання Material Design компонентів дозволило створити сучасний, естетичний вигляд програми, що відповідає вимогам до мобільних додатків. Теми, стилі та кастомізація компонентів були реалізовані для забезпечення унікального стилю програми, що адаптується під світлий та темний режими.

Крім того, Jetpack Compose значно полегшує інтеграцію з іншими бібліотеками Jetpack, такими як Navigation для керування екранною навігацією, ViewModel для збереження стану додатку під час зміни конфігурацій, та Hilt для впровадження залежностей. Це забезпечує модульність та легкість у подальшій підтримці і масштабуванні проекту.

Загалом, використання Jetpack Compose не лише прискорило процес розробки, але й зробило інтерфейс більш гнучким, адаптивним та легким у підтримці. Завдяки цьому проект має потенціал до ефективної роботи навіть при змінних вимогах та збільшенні функціоналу.

У проекті застосовано бібліотеку MQTT [26] для ефективного обміну даними між пристроями. MQTT. Завдяки цій бібліотеці вдалося легко інтегрувати різні пристрої, такі як датчики температури й вологості.

На блок-схемі (рис.3.12) представлено детальний процес налаштування та роботи з MQTT-клієнтом, який забезпечує зв’язок між пристроями в IoT-системі. Алгоритм розпочинається зі старту, де ініціалізується робота системи. Першим кроком є конфігурація MQTT-клієнта за допомогою функції setupMqttClient. На цьому етапі налаштовуються параметри з’єднання, такі як адреса брокера, порт, параметри авторизації та інші необхідні параметри, що забезпечують стабільну роботу клієнта.

Наступний етап включає спробу підключення до MQTT-сервера за допомогою функції connect. Цей процес є критично важливим, оскільки саме з’єднання з сервером дозволяє клієнту обмінюватися повідомленнями в реальному часі. Після цього алгоритм переходить до перевірки стану підключення. Якщо з’єднання успішно встановлено, система виконує підписку на топік за допомогою функції subscribeToTopic. Підписка дозволяє клієнту отримувати повідомлення з певного каналу зв’язку, які надсилаються іншими пристроями або сервісами.

У випадку, якщо з’єднання не вдалося встановити, викликається функція Error, яка відображає повідомлення про помилку. Це інформування

користувача про проблему через UI. Після цього алгоритм повертається до повторної перевірки з’єднання, щоб забезпечити стабільний зв’язок.

Коли клієнт отримує повідомлення з підписаного топіка (через функцію onReceived), наступним кроком є оновлення інтерфейсу користувача за допомогою функції updateUI. Так ми оновимо візуалізацію отриманих даних, таких як поточні показники сенсорів або статус пристроїв. Після цього здійснюється публікація повідомлення через функцію publishMessage. Публікація дозволяє передати оновлені дані або команди іншим підключеним пристроям чи сервісам. Інтерфейс знову оновлюється для відображення актуального стану системи.

На завершальному етапі, після успішного виконання всіх дій або при зупинці алгоритму, процес припиняється. Цей підхід забезпечує надійну роботу системи, гнучкість у реагуванні на події та зручність у використанні для кінцевого користувача.

Розроблено додаток (рис.3.13) для моніторингу та керування абіотичними факторами в курнику.

Додаток дозволяє додавати декілька курятників для моніторингу і управління їх умовами, що дозволяє контролювати різні об'єкти одночасно, зручно перемикаючись між ними через інтерфейс додатку. Це дозволяє вести ефективне управління декількома об'єктами на різних локаціях, підвищуючи зручність і ефективність роботи. Для роботи відбувається підключення до брокера, що забезпечує роботу в реальному часі завдяки Інтернету.

Цей додаток особлив для великих господарств, де курятники розташовані на різних локаціях, або для підприємств, які займаються птахівництвом на віддалених майданчиках. Користувач може легко відстежувати стан усіх курятників, оперативно реагувати на зміни умов та приймати рішення, що підвищує загальну ефективність роботи.

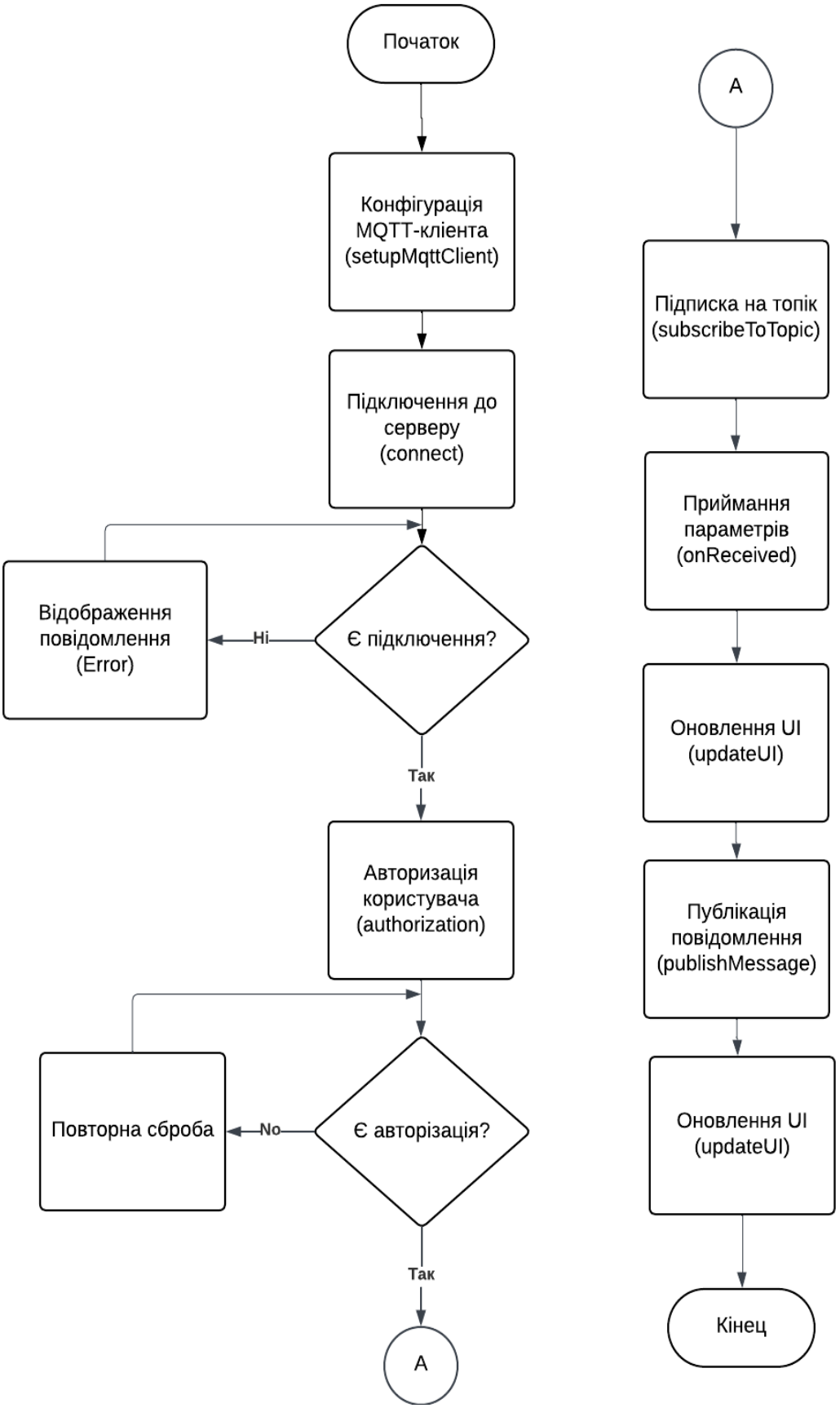

Рисунок 3.12 - Приклад роботи MQTT у додатку.

Передбачається кілька режимів роботи: ручний, автоматичний. У ручному режимі користувач може безпосередньо регулювати параметри середовища, такі як температура, вологість, вентиляція і освітлення, вручну змінюючи налаштування кожного пристрою. Автоматичний режим забезпечує автономну роботу системи, коли всі параметри підтримуються в межах оптимальних значень, з урахуванням заданих критеріїв.

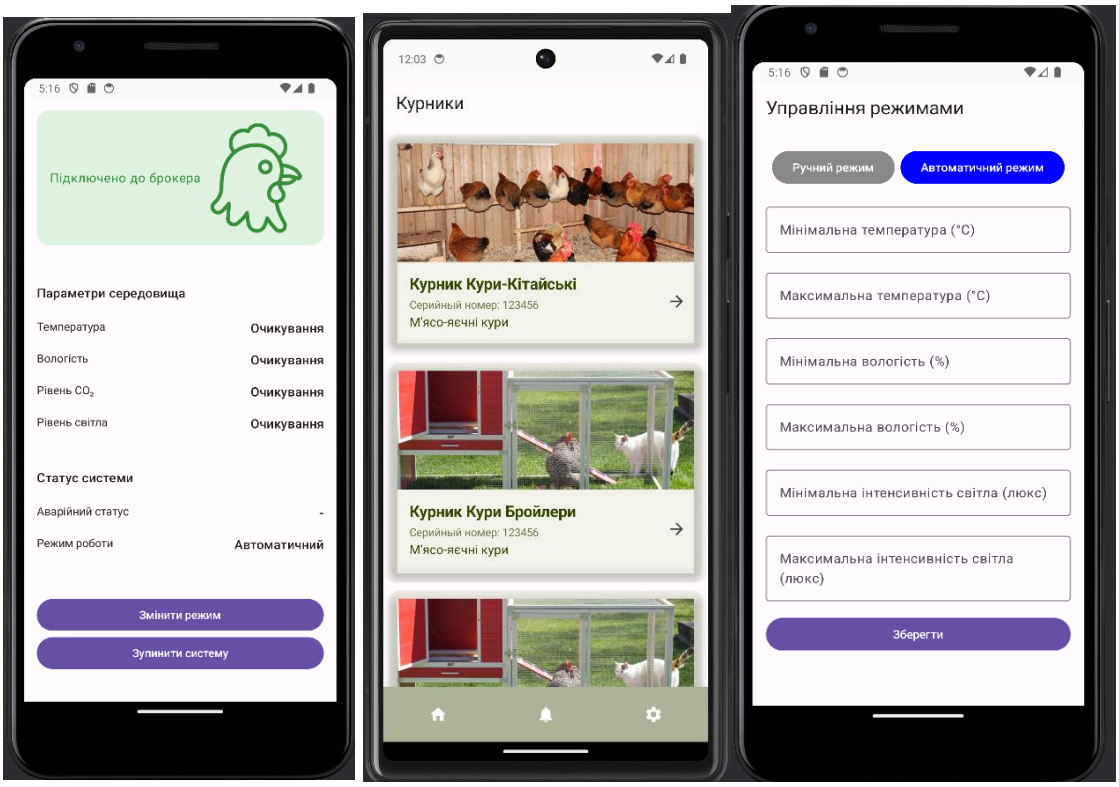

Рисунок 3.13 - Додаток для моніторингу та керування абіотичними факторами у курнику

Висновок до 3 розділу

У цьому розділі детально розглянута розробка та тестування програмного забезпечення для автоматизованого управління пташником з використанням різних компонентів і технологій.

Спочатку здійснено симуляцію в середовищі Proteus, де була створена схема системи управління пташником, зокрема підключення датчиків, реле,

мікроконтролера та мікрокомп'ютера, що дозволило протестувати взаємодію компонентів перед їх фізичним впровадженням.

Наступним етапом стало налаштування брокера Mosquitto для комунікації між Raspberry Pi, датчиками та іншими пристроями через протокол MQTT, що забезпечило ефективну передачу даних у локальній мережі. Основне програмне забезпечення для Raspberry Pi було розроблене для зчитування даних з датчиків, таких як температура, вологість і освітленість, а також для управління реле з метою регулювання умов у пташнику. Для програмування використано Python. Для зберігання та версійного контролю програмного коду був використаний GitHub, що забезпечило зручну співпрацю та контроль за змінами.

Окрім того, було створено Android додаток на Kotlin за допомогою Jetpack Compose у середовищі Android Studio для моніторингу та управління пташником через мобільний пристрій. Додаток надає користувачеві можливість відображати дані, що надходять від Raspberry Pi, та передавати команди для управління. Ці етапи разом утворюють цілісну автоматизовану систему для ефективного моніторингу та управління пташником.

РОЗДІЛ 4
МОДЕЛЬ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ПОЖЕЖ У ПТАХІВНИЦТВІ НА ОСНОВІ АНАЛІЗУ АБІОТИЧНОГО ФАКТОРУ

4.1 Аналіз моделі для машинного навчання.

Машинне навчання виступає потужним інструментом для автоматизації розпізнавання, аналізу та прогнозування в реальному часі. У птахівництві застосування машинного навчання для розпізнавання пожеж є ключовим елементом для забезпечення безпеки та здоров'я птиці. Це дозволяє своєчасно виявляти пожежі, здатні спричинити великі руйнування серед поголів'я та пошкодження інфраструктури.

Пожежі відносяться до найнебезпечніших абіотичних факторів, які здатні негативно позначитися на здоров'ї тварин і птахів. В результаті пожежі виникає висока температура і продукти горіння, серед яких токсичні гази і дим. Це може призвести до фізіологічного стресу, дихальних проблем, а в найгіршому випадку - до загибелі птиці. Чим швидше вдається виявити пожежу, тим більше шансів на її локалізацію і запобігання серйозним наслідкам.

Традиційні методи виявлення пожеж можуть бути недостатньо оперативними, особливо в умовах великих птахофермерських комплексів або вночі, коли пожежа може розвиватися без візуальних ознак. Автоматизовані системи, засновані на машинному навчанні, можуть миттєво аналізувати зображення та сигналізувати про наявність пожежі, що дозволяє вчасно вжити заходів. Використовуючи алгоритми машинного навчання, можна не лише визначити факт наявності пожежі, а й оцінити її масштаб і локалізацію, що дасть змогу ефективно спланувати евакуацію птахів або прийняття інших заходів для їх захисту. Виявлення пожежі на ранніх етапах дозволяє знизити

ризики фінансових втрат, оскільки пошкодження будівель, техніки та загибель поголів'я може призвести до значних витрат.

Автоматизовані системи також знижують навантаження на персонал, оскільки алгоритми машинного навчання можуть працювати безперервно, без необхідності постійного контролю з боку людини. Моделі машинного навчання можуть бути навчанні на датасетах, що містять зображення пожеж або диму. Після тренування такі моделі здатні автоматично розпізнавати ознаки пожежі, навіть за слабких або часткових ознак, які можуть бути непомітними для звичайного спостерігача.

Для створення ефективної моделі машинного навчання, здатної виявляти пожежі в умовах птахівницького комплексу, було використано математичні формули та алгоритми для обробки та аналізу зображень. Ці формули забезпечують нормалізацію вхідних даних, а також трансформації, необхідні для збільшення різноманітності навчального набору даних. Завдяки цьому модель здатна краще адаптуватися до різних умов, що можуть виникнути в реальному середовищі

Згорткові нейронні мережі (CNN) [27] - це тип штучних нейронних мереж (ANN)**,** спеціально розроблений для обробки даних з сітчастою топологією, таких як зображення. CNN є надзвичайно ефективними в завданнях комп'ютерного зору, оскільки вони можуть автоматично вивчати та виділяти важливі ознаки з вхідних зображень з мінімальним попереднім обробленням.

Основна перевага CNN над традиційними повнозв'язними мережами (FCN) полягає в тому, що CNN використовують загальні ваги для всіх нейронів, що значно зменшує кількість параметрів і допомагає уникати перенавчання, покращуючи загальну здатність до узагальнення.

Як було зазначено раніше, CNN складається з кількох основних складових (так званих шарів архітектури).

Згортковий шар є найважливішим компонентом будь-якої архітектури CNN. Він містить набір згорткових ядер (фільтрів), які застосовуються до вхідного зображення для створення вихідної карти ознак (feature map).

Ядро можна описати як матрицю чисел, де кожне значення є вагою цього ядра. Спочатку всі ваги ініціалізуються випадковими числами, а під час тренування моделі вони налаштовуються для витягнення корисних ознак із зображення. Приклад ядра 2 × 2 (4.1).

$$\begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix}, \tag{4.1}$$

У CNN вхідні дані зазвичай представлені у вигляді багатоканальних зображень (наприклад, для RGB зображення буде 3 канали - червоний, зелений і синій).

Операція згортки (4.2) полягає в тому, що ядро накладається на зображення, і обчислюється скалярний добуток між значеннями пікселів в зображенні та вагою ядра. Потім ці значення сумуються, утворюючи елемент у вихідній карті ознак. Цей процес повторюється для кожної частини зображення.

$$Y\,(i,j) = \sum_m \sum_n X\,(i+m, j+n) * K\,(m,n), \tag{4.2}$$

де $Y\,(i, j)$ - значення результату свертки на позиції (i, j) у вихідній матриці Y; $X\,(i+m, j+n)$ - елемент матриці X, зміщений на m рядків і n стовпців; $K\,(m, n)$ - елемент ядра свертки K, де індекси m і n відповідають позиціям у ядрі.

Сумування відбувається по всіх елементах ядра K в межах вікна розміром $h \times w$. Для кожної позиції (i,j) в результативній матриці Y ми беремо підматрицю з вихідної матриці X, яка покриває область, відповідну розміру ядра. Ця підматриця множиться поелементно на ядро K, результати поелементного множення сумуються, і отримана сума стає значенням у матриці Y на позиції (i,j).

Приклад: якщо задано зображення 4x4, і ядро 2x2, операція згортки виглядатиме так

$$\text{Вхідна матрця X} = \begin{bmatrix} 1 & 0 & -2 & 1 \\ -1 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \text{ Ядро K} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}, \quad (4.3)$$

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} = 1 \cdot 0 + 0 \cdot 1 + (-1) \cdot (-1) + 0 \cdot 2 = 0 + 0 + 1 + 0 = 1$$

$$\begin{bmatrix} 0 & -2 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} = 0 \cdot 0 + (-2) \cdot 1 + 0 \cdot (-1) + 1 \cdot 2 = 0 - 2 + 0 + 2 = 0$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} = (-1) \cdot 0 + 0 \cdot 1 + 0 \cdot (-1) + 2 \cdot 2 = 0 + 0 + 0 + 4 = 4$$

І так далі.

$$\begin{bmatrix} 1 & 0 & 4 \\ 4 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \quad (4.4)$$

Формула для розрахунку розміру вихідної карти ознак є ключовою при аналізі операції згортки у нейронних мережах. Вона дозволяє визначити розмір результату після застосування ядра згортки *(f×f)* до вхідного зображення *(h×w)*, з урахуванням паддінгу *(p)* та кроку *(s)*.

Розмір вихідної карти ознак *(h×w)* обчислюється за такими формулами (4.5).

$$h = \left\lfloor \frac{h - f + p}{s} + 1 \right\rfloor, \quad (4.5)$$

$$w = \left\lfloor \frac{w - f + p}{s} + 1 \right\rfloor, \quad (4.6)$$

де h – висота вхідної матриці (зображення); w – ширина вхідної матриці; f – розмір ядра згортки (фільтра), зазвичай квадратний *(f×f)*; p – заповнення (кількість пікселів, доданих навколо вхідної матриці); s – крок (кількість пікселів, на які зсувається ядро під час обчислень).

Пулінгові шари служать для зменшення розмірів карт ознак, що виникають після згортки. Іншими словами, вони зменшують розміри карт ознак, зберігаючи найважливішу інформацію з кожного регіону під час процесу зменшення розміру.

Завдання пулингу полягають у наступному: зменшення розмірів карт ознак, що знижує обчислювальні витрати та кількість параметрів моделі, а також допомагає уникнути перенавчання. Пулинг сприяє збереженню найбільш домінуючих ознак, що дозволяє зберегти ключову інформацію, важливу для розв'язання задачі, незалежно від точного розташування цих ознак на вихідній карті.

$$S_i = \max_{i \in R_j} h_i \ , \tag{4.7}$$

- S_i - вихідне значення для підрегіону.
- h_i - значення пікселя в підрегіоні.

Матриця (4.3), розмір її пула складає 2×2 з кроком 1. Це означає, що з кожного регіону 2×2 вибирається максимальне значення, і це значення зберігається в вихідній карті ознак.

Приклад пулінга (4.8).

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \xrightarrow{\max(1,0,-1,0)} 1 \begin{bmatrix} 0 & -2 \\ 0 & 1 \end{bmatrix} \xrightarrow{\max(0,-2,0,1)} 1 \begin{bmatrix} -2 & 1 \\ 1 & 2 \end{bmatrix} \xrightarrow{\max(-2,1,1,2)} 2$$

$$\begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} \xrightarrow{\max(-1,0,0,2)} 2 \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \xrightarrow{\max(0,1,2,1)} 2 \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \xrightarrow{\max(1,2,1,0)} 2 \tag{4.8}$$

$$\begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix} \xrightarrow{\max(0,1,2,0)} 2 \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} \xrightarrow{\max(2,0,1,0)} 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \xrightarrow{\max(1,0,1,0)} 1$$

$$\begin{bmatrix} 1 & 0 & -2 & 1 \\ -1 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{Після пулінгу}} \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

Розглянемо активаційні функції. Основна задача будь-якої активаційної функції в нейронній мережі - це перетворення вхідних значень у вихідні. Вхідні значення обчислюються як зважена сума входів нейрона з додаванням зміщення (якщо воно є). Іншими словами, активаційна функція вирішує, чи "запрацює" нейрон для даного вхідного значення, видаючи відповідний вихід.

В архітектурі згорткових нейронних мереж (CNN) після кожного навчального шару (шари з вагами, такі як згорткові та повнозв'язні шари) використовуються нелінійні активаційні шари. Нелінійна поведінка цих шарів дозволяє CNN навчатися більш складним залежностям і ефективно відображати вхідні дані у вихідні. Важливою властивістю активаційної функції є її диференційованість, що необхідно для алгоритму зворотного поширення помилки, який використовується для навчання моделі.

У проекті використовуються ReLU (рис.4.1) (Rectified Linear Unit), яка працює за принципом обрізання всіх від'ємних значень, залишаючи лише позитивні. Математично це виглядає так

$$F(x)_{ReLU} = max(0,x), \tag{4.9}$$

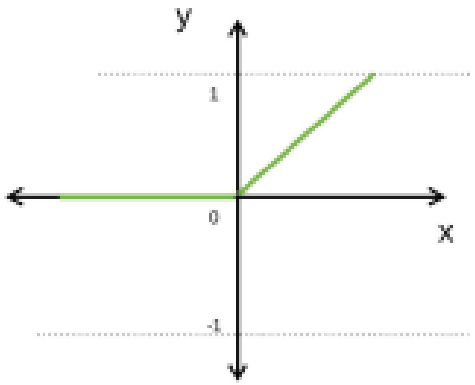

Рисунок 4.1 - Функція активації ReLU (Rectified Linear Unit)

Функція ReLU є дуже популярною завдяки своїй простоті та ефективності, а також тому, що вона не страждає від проблеми "вибуху" або "згасання" градієнта для позитивних значень входу. Це дозволяє ReLU значно прискорити процес навчання порівняно з іншими функціями активації. В

цілому, це функція застосовуються для додавання нелінійності в модель, що дозволяє нейронній мережі навчатися більш складним залежностям.

Повністю зв'язаний (FC) шар - це останній етап в архітектурі сверточної нейронної мережі (CNN), зазвичай використовуваний для класифікації. У FC-шарі кожен нейрон з'єднаний з усіма нейронами попереднього шару, а останній FC-шар служить вихідним шаром для класифікації. FC-шари реалізують принцип багатошарового перцептрона (MLP), класичної архітектури нейронних мереж. Вони отримують вхід від останнього сверточного чи пулінгового шару у вигляді матриць, таких як карти ознак. Ці матриці перетворюються в одномірний вектор через процес розгладжування (Flattening), після чого вектор подається до FC-шару, де кожен нейрон з'єднаний з усіма нейронами попереднього шару. На кожному з'єднанні обчислюється взважена сума входів, яка потім проходить через активаційну функцію. На основі результатів останнього FC-шару генерується остаточний результат, наприклад, ймовірності класів для задачі класифікації. У FC-шарах немає локальних зв'язків, як у сверточних, і вони мають велику кількість параметрів, що робить їх обчислювально дорогими, особливо в глибоких мережах. Ці шари використовуються для прийняття остаточних рішень на основі видобутих ознак.

Рисунок 4.2 ілюструє загальну архітектуру FC-шарів, де відбувається з'єднання всіх нейронів у шарах мережі.

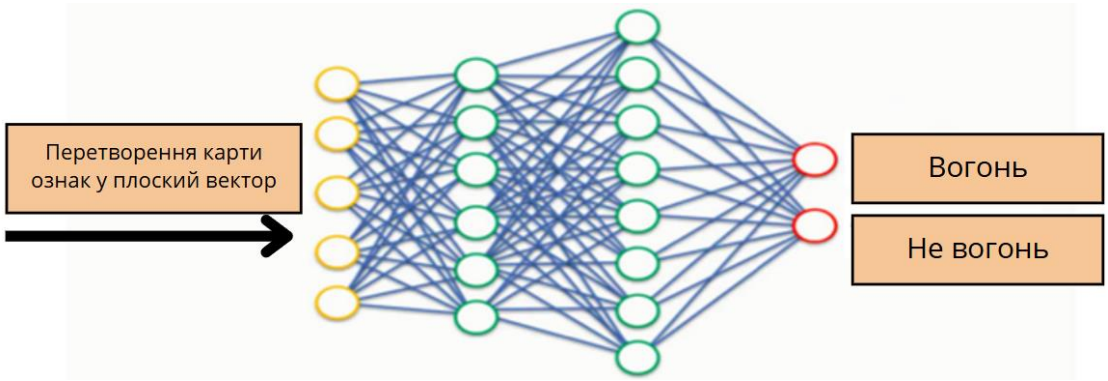

Рисунок 4.2 - Архітектура повністю зв'язанного шару

Попередня обробка даних є важливим етапом, оскільки вона забезпечує якість та узгодженість вхідних даних для моделі, що впливає на її точність та стабільність. Попередня обробка зображень включає нормалізацію значень пікселів, а також можливе застосування аугментації для збільшення кількості даних, зокрема використання формул для масштабування або трансформацій. Це буде забезпечувати коректне сприйняття вхідної інформації моделлю та покращує якість навчання.

Нормалізація значень пікселів: усі зображення складаються з пікселів, значення яких зазвичай знаходяться в діапазоні від 0 до 255 (8-бітне представлення). Нормалізація дозволяє зменшити ці значення до діапазону від 0 до 1. У результаті нормалізації визначається нормалізоване значення інтенсивності пікселя *x′*, яке обчислюється за формулою нижче.

$$x' = x / 255.0, \tag{4.10}$$

де x − початкова інтенсивність пікселя, x′ − нормалізоване значення інтенсивності пікселя, що належить діапазону [0;1].

Нормалізовані значення допомагають алгоритмам машинного навчання працювати стабільніше та швидше, оскільки більшість моделей чутливі до масштабу даних. Зменшення масштабу значень запобігає числовим переповненням під час виконання обчислень у моделі.

Зазначимо, що *T(x)* — це функція трансформації, яка описує зміну, що застосовується до вхідного зображення xxx. Іншими словами, *T(x)* відображає нове зображення після застосування певної операції аугментації, такої як обертання, масштабування, зміщення або віддзеркалення.

Аугментація використовується для збільшення кількості та різноманітності навчальних даних. Це особливо корисно, якщо вихідний набір зображень невеликий. Формула (4.11) виражає перетворення вхідного зображення *x* в аугментоване зображення *x′* за допомогою набору трансформацій *T(x)*

$$x' = T(x), \qquad (4.11)$$

де x - це початкове значення пікселя (або набір пікселів у зображенні), $T(x)$ - це функція, яка визначає тип трансформації, що застосовується до пікселів або всього зображення. (наприклад: обертання на 10 градусів або зміщення на 5 пікселів), x' - результат трансформації, тобто нове значення пікселя після обробки.

Ця величина x' додається до навчального набору даних і використовується для підвищення стійкості моделі до різних варіацій вхідних даних.

До основних методів аугментації належать:

1. Обертання: змінює кут нахилу зображення $T(x)=R\ (x, \theta)$, де θ - кут обертання.

2. Масштабування: збільшує або зменшує розмір зображення. $T(x) =S(x, \alpha)$ де α- коефіцієнт масштабування.

3. Зміщення: переміщує зображення по осі x або y. $T(x)=M\ (\Delta x, \Delta y)$, де Δx і Δy - зміщення по відповідних осях.

4. Віддзеркалення: горизонтальне або вертикальне відображення. $T(x)=H(x)$, де $H(x)= (x_{max} - x, y)$.

Аугментація створює нові версії існуючих зображень шляхом застосування зазначених трансформацій. Це допомагає моделі навчитися розпізнавати об'єкти у різних умовах (з різними розташуваннями, розмірами чи орієнтацією).

Це використається для покращення моделі, бо трансформовані зображення допомагають моделі навчитися розпізнавати об'єкти в різних умовах. Росте кількість даних, тощо використання $T(x)$ - дозволяє створити нові варіанти даних без необхідності збирати додаткові зображення. Підвищення стійкості моделі: трансформації забезпечують адаптацію моделі до варіативності в реальному середовищі, наприклад, до зміни освітлення, розташування або масштабу об'єктів.

У задачі двокласової класифікації, використовується бінарна крос-ентропія (4.12) як функцію втрат для вимірювання якості передбачень моделі. Основна мета — мінімізувати цю величину, щоб покращити точність класифікації.

Функція крос-ентропії вимірює різницю між істинним розподілом ймовірностей *(y)* та передбаченим розподілом *(p).* Вона дає уявлення про те, наскільки добре чи погано модель передбачає ймовірність для кожного приклада. Чим менше значення крос-ентропії, тим краща модель.

Якщо *y=1,* то функція втрат буде залежати від значення ймовірності p. Якщо модель правильно передбачає наявність пожежі (p близько до 1), то втрата буде мала.

Якщо *y= 0*, то функція втрат буде залежати від того, наскільки точно модель передбачає відсутність пожежі (p близько до 0).

Ця функція втрат використовується для того, щоб модель навчалася давати правильні ймовірності для кожного класу, зменшуючи різницю між передбаченнями та реальними мітками.

$$H(p, y) = -(y * \log(p) + 1(1-y) * \log(1-p)), \quad (4.12)$$

- *p* - ймовірність, що зображення належить до класу 1 (наприклад, "пожежа"),
- *y* - реальна мітка класу (0 або 1), де 1 означає наявність об'єкта (пожежі), а 0 - відсутність,

Для оцінки моделі використовуються різні метрики [28], серед яких важливим інструментом є матриця плутанини (Confusion Matrix) - це таблиця, яка використовується для оцінки ефективності класифікаційної моделі, особливо в задачах бінарної або багатокласової класифікації. Вона допомагає зрозуміти, скільки прикладів кожного класу було правильно чи неправильно класифіковано моделлю.

Для бінарної класифікації (вогонь / не вогонь) матриця плутанини має вигляд (4.13).

$$\begin{bmatrix} \text{True Positives (TP)} & \text{False Positives (FP)} \\ \text{False Negatives (FN)} & \text{True Negatives (TN)} \end{bmatrix} \quad (4.13)$$

- True Positives (TP) - кількість правильних передбачень класу "вогонь" (модель вірно передбачила, що на зображенні є вогонь).
- False Positives (FP) - кількість неправильних передбачень, коли модель помилково передбачила "вогонь", хоча насправді це не так.
- False Negatives (FN) - кількість неправильних передбачень, коли модель помилково передбачила "не вогонь", хоча насправді це вогонь.
- True Negatives (TN) - кількість правильних передбачень класу "не вогонь" (модель вірно передбачила, що на зображенні немає вогню).

Ці величини допомагають у подальшому обчисленні різних метрик ефективності моделі.

Слід зазначити, що результат з формул вказує на деякі частки правильно класифікованих і неправильно класифікованих прикладів серед усіх передбачених даних. Тому у результатах можуть виходити десяткові числа, які відображають частку правильних передбачень. Ці числа зручно використовувати у вигляді відсотків, щоб легше сприймати результат. Щоб перетворити десяткове число у відсотки, необхідно помножити його на 100.

Точність (Accuracy) — це метрика, яка вимірює, скільки правильно класифікованих прикладів було серед всіх передбачень моделі. Вона визначає, яку частку всіх прогнозів модель зробила правильно. Формула для обчислення точності виглядає так

$$Accurasy = \frac{TP+TN}{TP+FP+TN+FN}, \quad (4.15)$$

де:

- TP- кількість вірно передбачених пожеж,
- TN - кількість вірно передбачених не-пожеж,
- FP - кількість помилково передбачених пожеж,

– FNF - кількість помилково передбачених не-пожеж.

Матриця плутанини також дозволяє оцінити *повноту (Recall)*, яка показує, наскільки добре модель виявляє пожежі. Це особливо важливо для систем безпеки, де пропуск позитивних випадків може бути критичним:

$$Recall = \frac{TP}{TP+FN}, \quad (4.16)$$

Точність (Precision) вимірює, наскільки передбачені "вогонь" відповідають реальності, тобто яка частина передбачених пожеж є справжніми пожежами:

Для забезпечення збалансованості між точністю і повнотою можна використовувати *F1-міру*, яка є гармонійним середнім між точністю і повнотою

$$Precision = \frac{TP}{TP+FP}, \quad (4.17)$$

Ці математичні концепції дозволять реалізувати систему автоматичного виявлення пожежі в птахівницькому комплексі, використовуючи можливості машинного зору та обробки даних безпосередньо на Raspberry Pi.

4.2. Створення моделі для виявлення пожежі у птахівництві.

У процесі розробки моделі для виявлення пожеж у птахівницькому комплексі використовуються спеціалізовані набори даних [29], які містять зображення пожеж у різних умовах. Ці датасети служать основою для навчання моделі, що дозволяє розпізнавати характерні ознаки пожежі, такі як зміни в кольорі, диму та інших візуальних ознак, що з'являються при загораннях. Завдяки такому підходу, система буде здатна в реальному часі виявляти та сигналізувати про пожежу, що є критично важливим для забезпечення безпеки птахів і збереження комплексу.

Датасет поділено на 2 папки: папка fireimages містить 755 зображень на відкритому повітрі з вогнем(рис.4.3), деякі з яких містять густий дим, а інша

папка non-fireimages містить 244 зображення природи(рис.4.4) (ліс, дерева, трава, річка, люди, туманний ліс, озеро, тварини, дорога, водоспад). Взято за мету створення модель класифікації, яка може виявити пожежу на зображеннях. Яка буде працювати на мікрокомп'ютере та відправляти аварійні сигнали.

Рисунок 4.3 - Зображення з вогнем

Рисунок 4.4 - Зображення без вогня

Спочатку було застосовано метод розкиду(рис.4.5), який допомагає візуалізувати не тільки розподіл міток у наборі даних, але й виявляти потенційні проблеми в даних (наприклад, дисбаланс класів або неправильні мітки). Це важливий крок в аналізі даних, особливо при підготовці набору даних для навчання моделі машинного навчання.

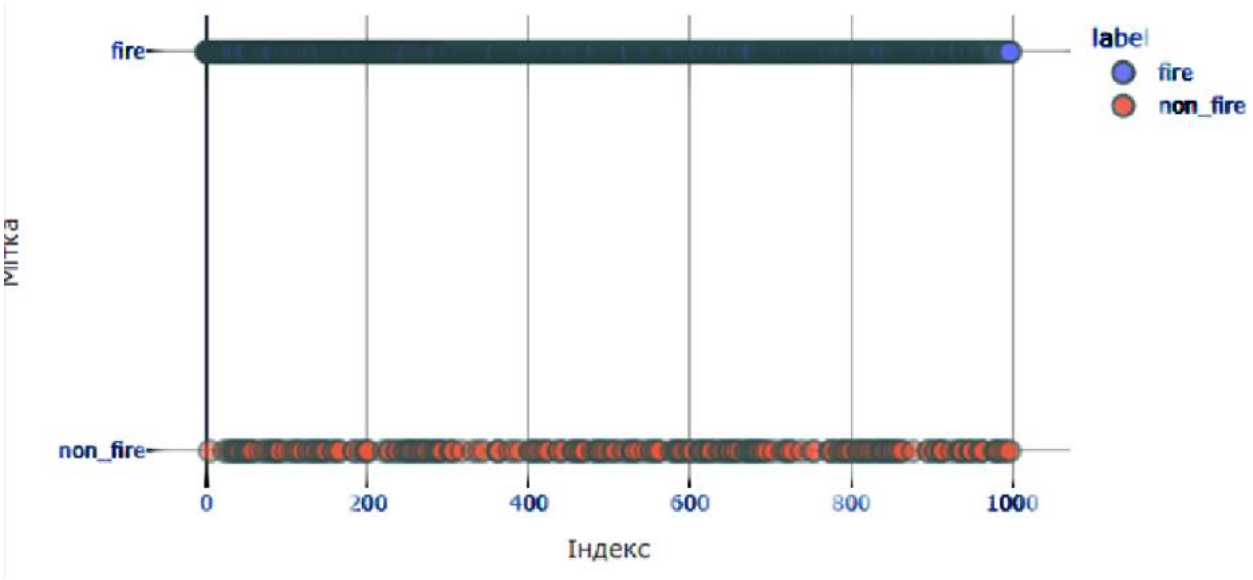

Рисунок 4.5 - Візуалізація розподілу міток у наборі даних

За допомогою ImageDataGenerator було реалізовано реальну аугментацію зображень(рис.4.6): обертання, зміщення по ширині/висоті, зсув, і збільшення масштабу.

Рисунок 4.6 - Аугментацію зображень

Протягом 15 епох навчання моделі були зафіксовані такі метрики, що відображають ефективність моделі машинного навчання(рис.4.7). Точність моделі, яка показує частку правильно класифікованих прикладів, на початок навчання складала 96.53% для навчальних даних і 95.48% для валідаційних. До кінця навчання точність на навчальних даних зросла до 97.93%, тоді як на валідаційних даних вона трохи знизилася до 94.47%. Це свідчить про можливе

перенавчання, коли модель надто добре адаптується до навчальних даних, але менш ефективно працює на нових даних.

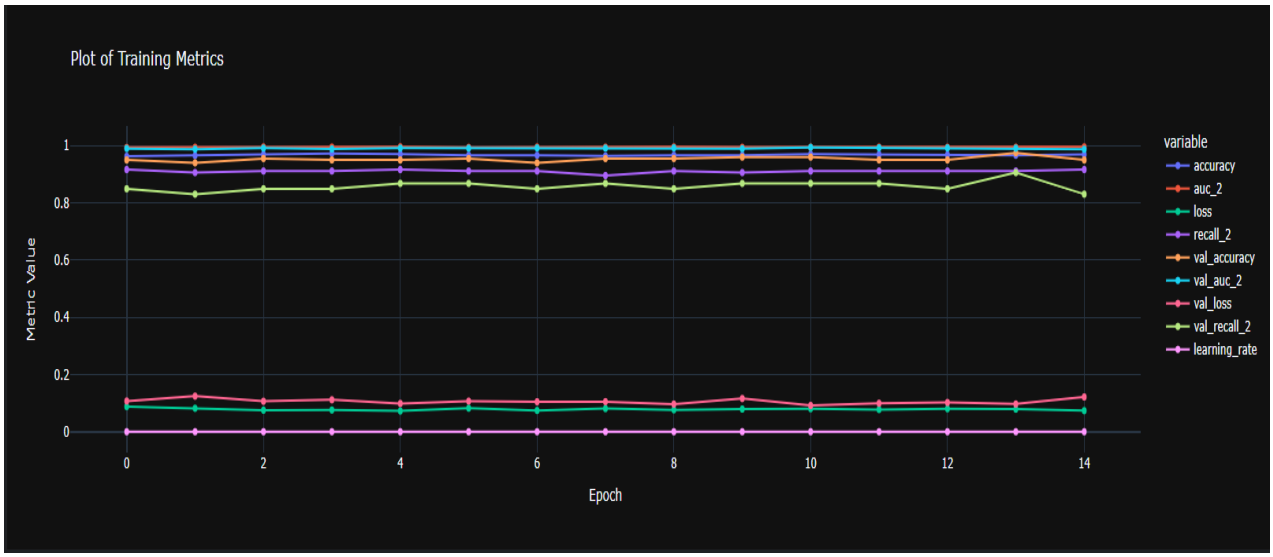

Рисунок 4.7- Графік ефективності машинного навчання

AUC, що вимірює здатність моделі розрізняти класи, залишалося високим протягом усього навчання. На навчальних даних AUC почалося з 0.9957 і досягло 0.9974, тоді як на валідаційних воно залишалося трохи нижчим, але все одно дуже високим - близько 0.9910. Ці значення вказують на те, що модель добре відокремлює класи навіть на нових даних.

Втрати, що відображають помилку моделі, на початок складали 0.0763 для навчальних даних і 0.0991 для валідаційних. Поступово втрати на навчальних даних зменшувалися і досягли 0.0628 до останньої епохи, тоді як втрати на валідаційних даних залишалися майже незмінними, становлячи 0.1016. Це демонструє стабільність моделі, але також може свідчити про недостатню узагальнюючу здатність.

Повнота, яка показує частку правильно знайдених позитивних прикладів, наприклад, зображень пожеж, на навчальних даних зросла з 91.23% до 95.06%. На валідаційних даних повнота була нижчою і знизилася з 84.91% до 83.02%, що може бути наслідком дисбалансу класів або складності даних для перевірки.

Швидкість навчання залишалася фіксованою на рівні $1*10^{-5}$, що забезпечувало стабільний процес оптимізації без різких змін у метриках.

Попри те, що цифри вказують на високу продуктивність моделі, графік ефективності навчання виглядає гірше. Це може бути пов'язано з кількома факторами:

– невелика різниця між метриками навчальних і валідаційних даних може вказувати на перенавчання.

– модель може бути чутливою до дисбаланса класів, коли один із класів переважає у валідаційних даних, що погіршує загальну ефективність.

Загальні метрики після навчання моделі:

1. Точність (accuracy): 0.95.
2. AUC: 0.99.
3. Втрати (loss): 0.10.
4. Повнота (recall): 0.87.

Для тестування моделі завантажено реальне зображення пожежі у курнику, змінили його розмір і передали через модель для класифікації (рис.4.8-4.9).

Рисунок 4.8 - Зображення курника з курками

Рисунок 4.9 - Зображення курника з курками в пожежі

На зображенні(рис. 4.8) прогнозоване значення складає 0, що вказує на дуже високу ймовірність того, що об'єкт не є вогнем (мітка "non_fire"). У другому випадку(рис.4.9), прогнозоване значення ймовірності становить 0.99389, що відповідає мітці "fire" та вказує на дуже високу ймовірність вогню, і насправді являється зображення з вогнем.

Щоб оцінити модель, використовується матриця плутанини, яка дозволяє порівняти передбачені мітки з фактичними мітками та оцінити ефективність моделі в різних аспектах.

Матриця плутанини (рис.4.10) показує, що з 146 реальних зразків класу "Fire" модель правильно класифікувала 113, а 33 були помилково віднесені до класу "Non-Fire". З 53 зразків класу "Non-Fire" модель правильно класифікувала 36, а 17 помилково віднесла до класу "Fire".

Загальна точність моделі згідно з матрицею плутанини становить 75%. Дає уявлення точності для обох класів разом (не відображає по кожному)

$$Accurasy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{113 + 36}{113 + 17 + 36 + 33} = 0.75$$

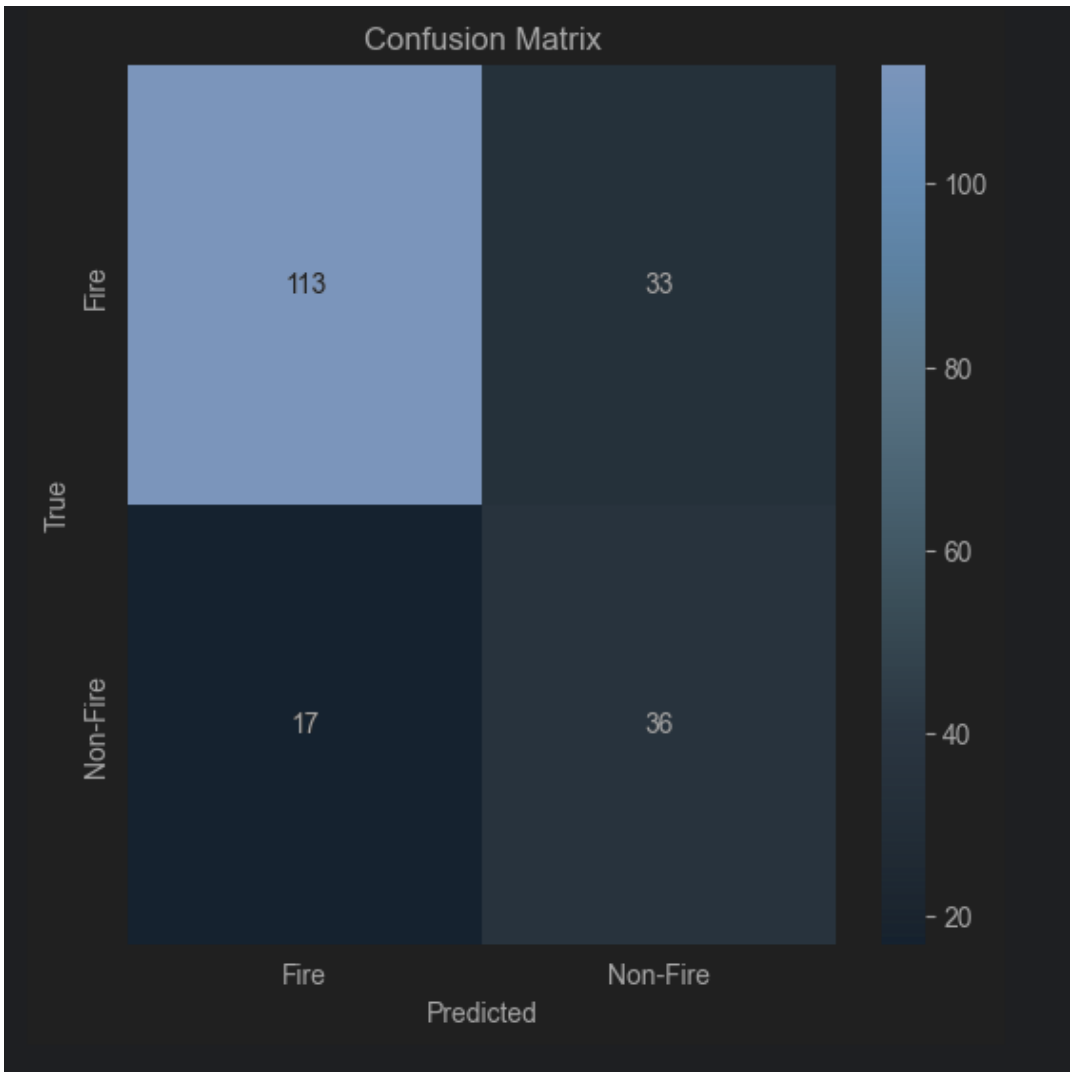

Рисунок 4.10 - Матриця плутанини

Метрики для класу "Fire" показують хороші результати: точність 0.77 означає, що 77% з передбачених "Fire" зразків дійсно є "Fire", а повнота 0.87 вказує на те, що модель правильно знайшла 87% усіх реальних зразків цього класу. F1-Score для класу "Fire" становить 0.82, що є гармонійним середнім між точністю та повнотою

$$Precision = \frac{TP}{TP+FP} = \frac{113}{113+17} = 0.87 \text{ (для класу "}Fire\text{")}$$

$$Recall = \frac{TP}{TP + FN} = \frac{113}{113 + 33} = 0.77 \text{ (для класу "}Fire\text{")}$$

$$F1 - Score(Fire) = 2\frac{Precision * Recall}{Precision + Recall} = 2 * \frac{0.87 * 0.77}{0.87 + 0.77}$$

$$= 0.82 \text{ (для класу "Fire")}$$

Для класу "Non-Fire" точність складає 0.68, що означає, що з усіх передбачених "Non-Fire" зразків, 68% дійсно не є вогнем, але повнота цього класу лише 0.52, що вказує на те, що модель не так добре знаходить зразки "Non-Fire". F1-Score для цього класу низький - 0.59, оскільки точність і повнота значно відрізняються. Загальна точність моделі становить 0.75, що означає, що 75% усіх зразків були правильно класифіковані.

$$Precision = \frac{TT}{TN + FN} = \frac{36}{36 + 33} = 0.52 \text{ (для класу "}Non - Fire\text{")}$$

$$Recall = \frac{TN}{TN + FP} = \frac{36}{36 + 17} = 0.68 \text{ (для класу "Non-Fire")}$$

$$F1 - Score(Non - Fire) = 2\frac{Precision * Recall}{Precision + Recall} = 2 * \frac{0.52 * 0.68}{0.52 + 0.68}$$

$$= 0.59 \text{ (для класу "}Non - \text{Fire")}$$

Загальний звіт з метрик для обох класів показує: для класу "Fire" точність 0.87, повнота 0.77, F1-Score 0.82, а для класу "Non-Fire" точність 0.52, повнота 0.68, F1-Score 0.59. Враховуючи дисбаланс класів, точність і F1-Score для "Fire" значно кращі, ніж для "Non-Fire".

Ці результати свідчать про те, що модель добре класифікує клас "Fire", але має труднощі з класом "Non-Fire", що, ймовірно, пов'язано з дисбалансом в кількості зразків класів у навчальній вибірці. Не дивлячись на це, відомо, що моделі, які мають загальну точність більше ніж 70%, зазвичай вважаються достатньо ефективними для багатьох практичних задач, особливо коли йдеться про складні або реальні умови, де дані можуть бути шумними або неповними.

Ця модель машинного навчання буде завантажуватися на Raspberry Pi, використовуючи TensorFlow Lite. Навчено модель на більш потужному комп'ютері, а потім перетворимо її у форматі. tflite за допомогою TensorFlow. Це дозволить оптимізувати модель для роботи на Raspberry Pi з обмеженими ресурсами. На Raspberry Pi буде встановлено TensorFlow Lite, що дозволить запустити модель та виконати необхідні обчислення. Завдяки цьому підходу,

можемо використовувати модель для виявлення пожеж в реальному часі, обробляючи зображення, отримані за допомогою камери на пристрої.

Висновок до розділу 4

У цьому розділі було описано модель, створену на основі машинного навчання, яка аналізує зображення для моніторингу курників і запобігання пожежам. Система обробляє фото, що отримуються, наприклад, кожні 5 секунд, для виявлення потенційно небезпечних ситуацій, таких як задимлення, перегрів або інші аномалії. Також було проведено аналіз архітектури моделі, яка побудована з використанням згорткових нейронних мереж (CNN), а її математичні основи розглянуто для кращого розуміння механізмів роботи та точності прогнозів. Це забезпечує швидке реагування на загрози та підвищує ефективність моніторингу.

РОЗДІЛ 5
ОХОРОНА ПРАЦІ ТА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

5.1 Основна теорія охрони праці

Охорона праці [31] - це система заходів і засобів, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні та лікувально-профілактичні аспекти, спрямованих на забезпечення безпечних умов праці, збереження життя, здоров’я та працездатності працівників у процесі їх професійної діяльності. Ця система охоплює різні напрямки: від створення безпечних умов праці до профілактики професійних захворювань і травм, а також надання медичної допомоги і компенсацій за втрату працездатності.

В Україні охорона праці регулюється на державному рівні. Основним документом, що регулює питання охорони праці, є Закон України «Про охорону праці», який був прийнятий у 1992 році. З моменту його прийняття і до сьогодні в країні здійснюється правове регулювання у сфері охорони праці, включаючи обов'язкові стандарти безпеки та гігієнічні вимоги на робочих місцях. Крім того, Україна є частиною міжнародних організацій, таких як Міжнародна організація праці (МОП), що також впливає на законодавство та нормативні акти у сфері охорони праці.

Охорона праці тісно пов'язана з економікою країни та підприємства. Створення безпечних і здорових умов праці сприяє підвищенню продуктивності, зниженню рівня професійних захворювань і травматизму. Ефективна організація охорони праці дозволяє значно скоротити втрати робочого часу, який раніше витрачався на відновлення після травм, лікування захворювань та усунення наслідків нещасних випадків. Це, у свою чергу, сприяє збільшенню ефективності праці та зниженню рівня плинності кадрів.

Крім того, зниження рівня травматизму та професійних захворювань має економічний ефект і для самого роботодавця. Оскільки виплати по

страхуванню від нещасних випадків і компенсації по професійних захворюваннях виплачуються за рахунок Фонду соціального страхування, зниження цих виплат дозволяє зменшити фінансове навантаження на підприємство. Це також сприяє зниженню страхових тарифів, що, у свою чергу, зменшує загальні витрати на охорону праці на підприємстві.

Важливим аспектом охорони праці є профілактика. Профілактичні заходи включають правильну організацію робочих місць, використання індивідуальних і колективних засобів захисту, а також регулярні перевірки стану здоров'я працівників і технічного стану обладнання. Однією з ключових завдань є мінімізація впливу шкідливих факторів на здоров'я працівників, таких як шум, пил, токсичні речовини і вібрація.

Також у контексті охорони праці важливо навчати працівників безпечним методам праці, правильному використанню обладнання і засобів захисту, а також реагуванню в разі надзвичайних ситуацій. Навчання має бути доступним і систематичним, охоплюючи всі категорії працівників - від молодшого персоналу до керівників, оскільки всі вони повинні бути поінформовані про ризики на виробництві та методи їх запобігання.

Основна мета навчання з охорони праці - не лише знання теорії, а й уміння застосовувати ці знання на практиці. Здатність правильно організувати трудовий процес і взаємодіяти з колегами в безпечних умовах сприяє формуванню культури безпеки на виробництві, де кожен працівник розуміє свою роль у забезпеченні безпеки і здоров'я на робочому місці.

5.2. Виробниче середовище та захист працівників від негативного впливу шкідливих виробничих чинників

Виробниче середовище - це сукупність умов, в яких здійснюється трудова діяльність. Воно включає в себе фізичні, хімічні, біологічні та соціально-економічні фактори, які можуть впливати на здоров’я та безпеку працівників. Забезпечення безпечних умов праці є необхідною умовою для

запобігання травмам та професійним захворюванням, а також для підвищення ефективності виробничих процесів.

Шкідливі виробничі чинники - це фактори, які можуть викликати травмування, захворювання або інші небажані наслідки для здоров'я працівників. Вони поділяються на кілька основних груп:

Фізичні чинники:

1. Шум - надмірний рівень шуму на виробництві може призвести до погіршення слуху, підвищення стомлюваності, порушень сну та нервових розладів.

2. Вібрація - тривала дія вібрацій може спричинити захворювання кровоносних судин, нервової системи, суглобів.

3. Температурний режим - надмірна спека чи холод можуть призвести до теплових або холодових ударів, зниження працездатності, а також сприяти виникненню хронічних захворювань.

4. Іонізуюче випромінювання - вплив радіації може призвести до серйозних захворювань, таких як рак, мутації, опіки.

Хімічні чинники:

5. Вибухонебезпечні та токсичні речовини - їх випаровування або потрапляння на шкіру може викликати отруєння, алергічні реакції, шкідливі впливи на органи дихання та інші системи організму.

6. Пил та аерозолі - дрібні частки пилу можуть потрапляти в дихальні шляхи і сприяти розвитку професійних захворювань, таких як пневмоконіози.

7. Газоподібні токсини - деякі виробничі процеси спричиняють виділення газів, які можуть бути отруйними, наприклад, оксиди азоту, сірки, чадний газ.

8. Біологічні чинники:

9. Мікроорганізми - працівники, які працюють в аграрній, харчовій або медичній галузях, можуть бути піддані ризику зараження патогенними бактеріями, вірусами та грибками.

10. Паразити - при роботі в умовах з підвищеною вологістю або у контакті з тваринами працівники можуть піддаватися впливу паразитів, таких як кліщі, комарі.

Психофізіологічні чинники, зокрема психоемоційне навантаження - стреси, тривожність та перевантаження можуть викликати нервові та психічні захворювання, знижувати працездатність і ефективність праці.

5.3. Поняття ризику як міри небезпеки

Ризик - це імовірність заподіяння шкоди з урахуванням її тяжкості. Він є кількісною мірою здатності певних подій завдавати шкоду людині. Поняття ризику використовується для прогнозування і оцінки стану умов та безпеки праці на робочому місці, ефективності впровадження заходів і засобів з охорони праці, доцільності робіт з підвищення безпечності обладнання, впровадження нових безпечних технологічних процесів тощо.

Ризик є важливим інструментом в управлінні безпекою праці, оскільки дозволяє оцінити ймовірність виникнення негативних наслідків при різних умовах та чинниках. Поняття ризику є актуальним як для *стохастичних*, так і для *детерміністичних ефектів*.

До *стохастичних ефектів* відносяться ті, ймовірність виникнення яких існує при будь-якій кількості випадків впливу небезпечного чи шкідливого чинника. Імовірність виникнення таких ефектів зростає при збільшенні кількості впливів, тоді як тяжкість наслідків не залежить від їх кількості чи рівня. Прикладом таких ефектів є аварії, пов'язані з обваленням гірських порід, раптові викиди газу, травмування під час виконання небезпечних робіт, а також виникнення злоякісних новоутворень чи генетичних змін внаслідок дії іонізуючого випромінювання.

До *детерміністичних ефектів* відносяться ті, що завжди настають при певних подіях чи перевищенні певного рівня чинника. Тяжкість наслідків таких подій залежить від величини цього чинника. Наприклад, виникнення

легеневих захворювань при накопиченні в легенях певної кількості пилу, розрив барабанної перетинки при перевищенні певного рівня звукового тиску, променева хвороба при перевищенні певної дози опромінення.

Для подій, що викликають стохастичні ефекти, ризик визначається як ймовірність виникнення травм, захворювань або смертельних випадків за певний проміжок часу, пов'язаних з впливом того чи іншого виробничого чинника. Ризик можна визначити за такою формулою

$$r = \frac{n}{N}, \tag{5.1}$$

де n - кількість осіб, які отримали травми або захворіли внаслідок дії на них чинника за певний проміжок часу (наприклад, травми, спричинені обваленням гірських порід);

N - загальна кількість осіб, на яких впливав цей чинник (наприклад, працівники, що працювали в шахті).

Ризик може бути розрахований для всіх працівників (загальний ризик) або для певної групи працівників, наприклад, залежно від професії, віку чи статі. Це дозволяє більш точно оцінити ймовірність виникнення негативних наслідків у різних категорій працівників.

Оцінка індивідуального ризику визначається для конкретної особи, яка піддається повторному впливу одного і того ж чинника. Індивідуальний ризик можна обчислити за такою формулою

$$r_i = \frac{1}{P}, \tag{5.2}$$

де P - середня кількість випадків дії чинника на один випадок пошкодження.

Оптимізація ризику передбачає зниження рівня ризику до найнижчого можливого рівня, забезпечуючи безпеку при мінімальних витратах. Це здійснюється в межах граничного ризику, що має верхню і нижню межі, де нижня межа визначається як знехтуваний ризик. При оцінці ризику важливо враховувати поняття знехтуваного ризику, прийнятного ризику та верхньої границі індивідуального ризику. Наприклад, для опромінення іонізуючим

випромінюванням рівень знехтуваного ризику становить 10^{-6}, а для персоналу, що працює з джерелами випромінювання, прийнятний ризик може бути 10^{-4}.

Технічні заходи з охорони праці - це комплекс дій і рішень, спрямованих на створення безпечних і здорових умов праці для працівників. Вони реалізуються через технологічні, конструктивні та організаційні рішення з метою мінімізації або усунення шкідливих та небезпечних чинників, що можуть впливати на працівників. Основною метою технічних заходів є забезпечення безпеки та зручності праці, а також запобігання нещасним випадкам.

Одним із основних технічних заходів є вдосконалення технологічних процесів і змін у обладнанні. Це включає впровадження нових безпечних технологій, автоматизацію та механізацію трудомістких і небезпечних процесів, а також заміну шкідливих хімічних сполук на безпечніші. Крім того, застосовуються нові матеріали, які є нетоксичними і менш займистими, а також мають менший рівень шуму і вібрацій. Важливо також впроваджувати системи моніторингу та контролю технологічних процесів, що дозволяють своєчасно виявляти та усувати можливі відхилення, які можуть призвести до аварій.

Конструктивні зміни в обладнанні та будівлях є ще одним важливим елементом технічних заходів. Вони включають розробку та впровадження безпечних конструкцій машин і механізмів, що зменшують ймовірність аварій та травм. Покращення ергономічних характеристик робочих місць, правильне розташування обладнання і врахування особливостей роботи людей допомагає знизити фізичне навантаження і запобігти травмам. Використання захисних огороджень і бар'єрів дозволяє мінімізувати ризик випадкових контактів з небезпечними частинами механізмів або шкідливими викидами.

Автоматизація процесів і впровадження дистанційних систем управління дозволяють зменшити людський фактор у небезпечних роботах. Це знижує кількість контактів людини з шкідливими та небезпечними чинниками, дозволяє проводити моніторинг процесів в реальному часі та

оперативно реагувати на небезпеки. Роботизовані системи можуть виконувати небезпечні операції, такі як в хімічних виробництвах або в умовах високих температур.

Колективні засоби захисту включають системи вентиляції та очищення повітря, що знижують концентрацію шкідливих газів і пилу в робочих приміщеннях. Автоматичні системи пожежогасіння та системи протидії вибухам, а також спеціальні огородження в зонах підвищеного ризику, забезпечують безпеку працівників.

Індивідуальні засоби захисту є необхідними для забезпечення безпеки працівників при виконанні робіт, що супроводжуються небезпечними чинниками. До таких засобів належать охоронні каски, захисні окуляри, респіратори, спецодяг, рукавички, а також системи захисту від шуму та протипожежне обладнання.

Для запобігання поломкам і аваріям важливо здійснювати планово-попереджувальний ремонт і технічний огляд обладнання. Перевірка обладнання на відповідність стандартам безпеки і встановлення датчиків для автоматичного виявлення потенційних проблем дозволяють своєчасно виявляти і локалізувати можливі загрози.

Таким чином, технічні заходи є важливим елементом системи безпеки праці, допомагаючи зменшити ризики та запобігати нещасним випадкам. Вони повинні бути розроблені на основі комплексного аналізу небезпек і відповідати сучасним вимогам охорони праці.

5.4 Електробезпека в процесі роботи з автоматизованими системами.

Електробезпека [32] - це система заходів, спрямованих на забезпечення захисту людей від небезпечного впливу електричного струму, електричних полів, електромагнітних полів та статичної електрики. Травми, які спричинені

дією на організм людини електричного струму і електричної дуги, називають електротравмами.

Електротравматизм порівняно з іншими видами травматизму має деякі відмінні особливості. Електротравматизм складає близько 1% від загальної кількості усіх нещасних випадків на виробництві. Але серед нещасних випадків зі смертельними наслідками електротравми складають близько 40%, посідаючи одне з перших місць, причому близько 90% смертельних уражень електричним струмом трапляється в електроустановках з напругою 127-380 В.

Основними причинами нещасних випадків, пов'язаних з обслуговуванням електричних мереж і електроустановок, вважають:

1) допуск до роботи осіб, які не мають кваліфікаційної групи з електробезпеки;

2) допуск до роботи осіб, які не знають приміщень і зовнішніх установок за ступенем небезпеки ураження електрострумом;

3) роботу на електроустановках і електроінструментом без заземлення, занулення, без перевірки опору ізоляції в мережах споживачів електроструму;

4) роботу без зняття напруги, без засобів колективного та індивідуального захисту;

5) роботу без наряду-допуску;

6) нерегулярне навчання та переатестацію персоналу, який обслуговує електромережі та електроустановки;

7) допуск до роботи осіб без медичного огляду;

8) початок роботи без попередньої перевірки відсутності напруги, розширення зони робочого місця.

До нещасних випадків призводить також застосування в особливо небезпечних приміщеннях і приміщеннях підвищеної небезпеки напруги понад 42 В.

Особливості електротравматизму:

1) організм людини не наділений властивістю, за допомогою якої можна було б визначити наявність електроструму;

2) електротравма може виникнути без безпосереднього контакту зі струмопровідниками, частинами устаткування (ураження через електричну дугу, крокову напругу тощо);

3) електричний струм, проходячи через тіло людини, діє не тільки в місцях контактів і на шлях проходження через організм, а й на центральну нервову систему, що спричинює ураження внутрішніх органів (порушення нормальної діяльності серця, зупинку дихання тощо).

Дія електричного струму на організм людини може бути дуже небезпечною і супроводжуватись різними фізіологічними та механічними ураженнями. Основні наслідки, які виникають під час контакту з електричним струмом представлено у табл 5.1.

Таблиця 5.1 - Види впливу електричного струму на організм людини

Тип дії	Опис
Термічна дія	Викликає опіки на окремих ділянках тіла, ушкодження кровоносних судин, нервових клітин, серця, мозку. Це може призвести до серйозних функціональних розладів.
Електролітична дія	Розклад органічних рідин та крові, що порушує їх фізико-хімічний склад.
Біологічна дія	Подразнення і збудження живих тканин, що супроводжується мимовільним скороченням м'язів.

Чим довше струм проходить через організм, тим більша ймовірність важких уражень або летального результату.

Людина потрапляє під дію електричного струму:

– у разі випадкового дотикання до струмопровідних частин електроустановки або наближення до цих частин на неприпустимо близьку відстань;

– під час виникнення в електроустановці аварійного режиму (ушкодження ізоляції, обрив проводів тощо).

Аналізуючи умови виникнення електричного кола через тіло людини, розрізняють безпосередній контакт людини зі струмопровідними частинами і опосередкований. Непрямий контакт настає під час пробою ізоляції на корпусі обладнання. Найчастішим і характерним випадком дотикання людини до струмопровідних частин є однофазне дотикання.

За однофазного дотикання у мережі з глухозаземленою нейтраллю (рис. 16) струм, що проходить через тіло людини, піде колом: фаза А тіло людини підлога (грунт) - заземлювач нейтралі - нейтраль (нульова точка джерела живлення).

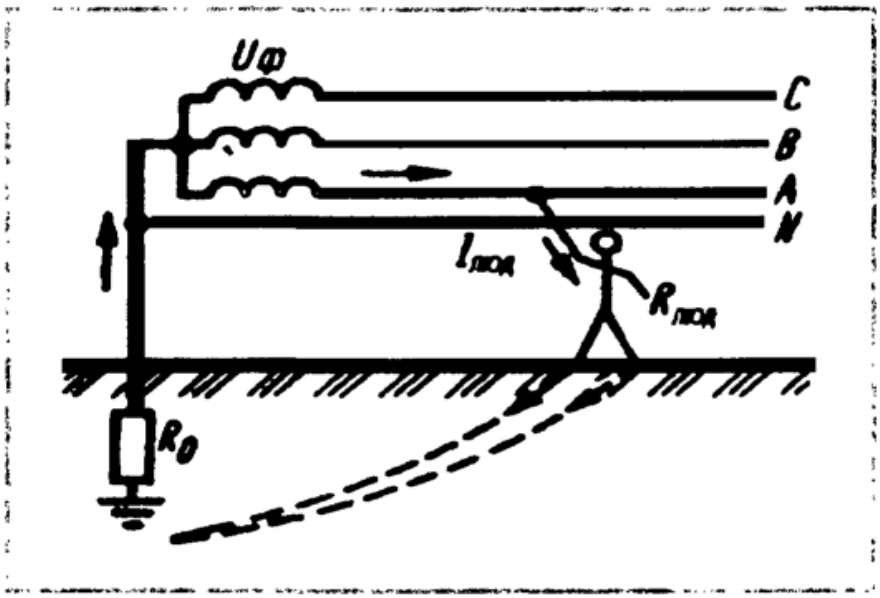

Рисунок 5.1 - Однофазне дотикання в мережі з глухозаземленою нейтраллю

За законом Ома

$$I_{\text{люд}} = \frac{U_{\text{ф}}}{R_{\text{люд}}+R_{\text{п}}+R_{\text{о}}}, \qquad (5.3)$$

де $U_{\text{ф}}$ – фазна напруга, яка діє на людину; $R_{\text{люд}}$- опір людини; $R_{\text{п}}$ опір підлоги; $R_{\text{о}}$ - опір заземлення нейтралі.

Якщо підлога струмопровідна, то $R_{\text{п}}$= 0. Тоді $U_{\text{ф}}$ = 220 В, $R_{\text{люд}}$ = 1000 Ом. Враховуючи, що $R_0 << R_{\text{п}}$ отримаємо

$$I_{\text{люд}} = \frac{U_{\text{ф}}}{R_{\text{люд}}} = \frac{220}{1000} = 0.22A = 220\text{мА}\,, \qquad (5.4)$$

таке дотикання вкрай небезпечне.

Якщо трапився нещасний випадок від дії електричного струму:

– потерпілого звільняють від дії струму;

– надають йому першу допомогу.

Під час ураження електричним струмом потрібно використовувати такі безпечні методи:

– вимикати напругу рубильником або вимикачем;

– забезпечити безпеку захисним вимиканням аварійної ділянки або мережі повністю.

Якщо вимикання не може бути виконано досить швидко, треба терміново звільнити потерпілого від дії струмопровідних частин, до яких він доторкається. При цьому особа, яка надає допомогу, повинна пам'ятати, що не можна доторкатися до потерпілого, бо це небезпечно для життя рятівника. Особі, яка надає допомогу, треба також бути обережним, щоб не доторкнутися до струмопровідної частини і не опинитися під напругою. Для звільнення потерпілого від струмопровідних частин або проводу до 1000 В користуються сухою палицею, дошкою або іншим сухим діелектричним предметом (рис. 5.2).

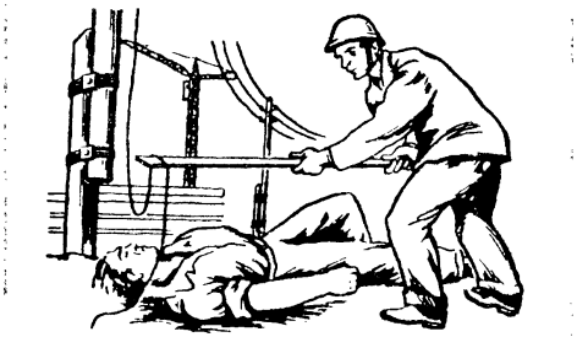

Рисунок 5.2 - Звільнення потерпілого від дії струму в установках до 1000 В відкиданням проводу дошкою.

5.5 Огляд питань кібербезпеки в проце сі роботи з автоматизованими системами

Однак із розвитком таких технологій зростає й кількість програмних загроз, що можуть вплинути на безпеку цих об'єктів. Програмні загрози, як-от віруси, хакерські атаки, або несанкціонований доступ до даних, можуть призвести до серйозних наслідків, зокрема до порушення роботи критичних систем, втрати даних або навіть до фізичних пошкоджень обладнання [29].

Захист від таких загроз вимагає комплексного підходу, що включає використання різноманітних методів і технологій для забезпечення безпеки.

Програмні загрози для автоматизованих об'єктів можуть бути різноманітними, і вони зазвичай виникають через вразливості в програмному забезпеченні або неправильну конфігурацію систем.

Таблиця 5.2 - Типів програмних загроз.

Тип загрози	Опис
Віруси та шкідливі програми	Віруси та інші шкідливі програми можуть потрапити в систему через підключення до мережі або через зовнішні носії інформації. Вони можуть змінювати або видаляти критичні дані, що викликає неполадки в роботі системи.
Атаки типу «відмова в обслуговуванні» (DoS)	Атаки DoS і DDoS (розподілені відмови в обслуговуванні) спрямовані на перевантаження мережевих ресурсів, що призводить до блокування доступу до сервісів або зниження продуктивності автоматизованих об'єктів.
Атаки типу «відмова в обслуговуванні» (DoS)	Атаки DoS і DDoS (розподілені відмови в обслуговуванні) спрямовані на перевантаження мережевих ресурсів, що призводить до блокування

	доступу до сервісів або зниження продуктивності автоматизованих об'єктів.
Неавторизований доступ	Зловмисники можуть отримати несанкціонований доступ до критичних систем і даних через вразливості в програмному забезпеченні або через слабкі паролі. Такий доступ може призвести до змін в операціях, знищення даних або саботажу.
Шкідливе ПЗ (Ransomware)	Витягування грошей від організації шляхом блокування доступу до даних або важливих систем через шифрування файлів.

Таблиця 5.3 - Методи захисту від програмних загроз.

Метод захисту	Опис
Шифрування даних	Використання шифрування на всіх етапах обробки та зберігання даних для забезпечення конфіденційності і запобігання несанкціонованому доступу.
Системи виявлення та запобігання вторгнень (IDS/IPS)	Виявлення підозрілих дій і блокування їх до того, як вони завдадуть шкоди системам АОТ. Це може бути інтегровано на рівні мережі або пристроїв.
Регулярні оновлення програмного забезпечення та патчі	Постійне оновлення програмного забезпечення та застосування патчів для усунення вразливостей, які можуть бути використані для атак.
Мультифакторна аутентифікація (MFA)	Використання багатофакторної аутентифікації для обмеження доступу лише для авторизованих

	користувачів, знижуючи ймовірність несанкціонованого доступу.
Контроль доступу на основі ролей (RBAC)	Управління доступом, де права доступу до даних та функцій надаються залежно від ролі користувача, мінімізуючи ризики несанкціонованих змін у системах.
Моніторинг та аудит системи	Регулярний моніторинг і аналіз системи для виявлення аномалій або спроб несанкціонованого доступу, а також створення історії подій для аудиту.
Використання фаєрволів та VPN	Захист від атак ззовні за допомогою фаєрволів та віртуальних приватних мереж (VPN), що забезпечують безпеку комунікацій і обмежують небажаний доступ.

5.6. Охорона навколишнього природнього середовища

Охорона навколишнього середовища [32] — це система заходів, спрямованих на раціональне використання природних ресурсів, збереження унікальних природних комплексів і забезпечення екологічної безпеки. Вона включає державні, правові, економічні та соціальні ініціативи, які обмежують негативний вплив людської діяльності на природу. Охорона довкілля є важливою умовою сталого розвитку країни та забезпечення здоров'я людей, через збереження безпечного середовища та раціональне використання ресурсів.

Біосфера — це природна підсистема географічної оболонки, яка являє собою глобальну екосистему, населена живими організмами. Вона складається з компонентів, які формуються під впливом діяльності живих організмів на Землі. Господарська діяльність людини впливає на навколишнє середовище, погіршуючи його якість, що спричиняє серйозні проблеми, такі як зміна клімату, забруднення води та повітря, деградація земель та зменшення генетичної різноманітності. Сучасна епоха, яка характеризується розвитком

науки і техніки, а також неконтрольованим зростанням населення, висвітлює важливість взаємозв'язку людини і природи. Проблеми забруднення, надмірного використання ресурсів і антропогенного впливу стали об'єктами уваги міжнародних організацій, таких як ООН, ЮНЕСКО, ЮНЕП, ВООЗ. Найбільший вплив на біосферу мають промисловість, енергетика, сільське господарство та транспорт.

Вплив промисловості на навколишнє середовище: Промисловість є важливою галуззю, без якої неможливий розвиток цивілізації, однак її діяльність завдає значної шкоди навколишньому середовищу. Найбільший негативний вплив мають гірничо-металургійні підприємства, які спричиняють кислотні дощі, забруднюють повітря і ґрунти токсичними речовинами, такими як миш'як, фосфор, свинець, ртуть і смоли. Підприємства кольорової металургії викидають в атмосферу пил, сірчаний ангідрид, оксиди азоту та свинець. Також важливою проблемою є звалища, які щорічно поглинають значні площі землі, забруднюючи її важкими металами та токсичними речовинами.

Вплив енергетики на навколишнє середовище: Енергетика є основою розвитку цивілізації, але вона також є головним джерелом забруднення атмосфери. Приблизно 80% всіх забруднень повітря є наслідком енергетичних процесів, зокрема спалювання вугілля, переробки нафти та видобутку матеріалів. Ці процеси призводять до викидів у атмосферу шкідливих сполук, таких як діоксид сірки, оксид вуглецю, оксиди азоту та різноманітні аерозолі, що спричиняють кислотні дощі і забруднення повітря.

Забруднення атмосфери може бути природним (виверження вулканів, пожежі) та антропогенним (пов'язане з діяльністю людини). Природні джерела, як правило, не завдають значної шкоди, але при інтенсивному поширенні можуть змінювати тепловий баланс Землі. Штучне забруднення виникає через викиди від виробництв і спалювання палива, що змінює склад атмосфери і може мати негативний вплив на навколишнє середовище та здоров'я людини.

Забруднення атмосфери поділяється на тверді, рідкі та газоподібні речовини, причому останні становлять 90% забруднюючих агентів. Серйозну шкоду завдають хімічні сполуки, зокрема сірчисті сполуки, оксиди азоту, хлор. Проблеми забруднення виникають через промисловість, транспорти, а також викиди від чорної металургії та радіоактивні речовини, зокрема після вибухів атомних бомб у 1945 році. Зокрема, оксиди азоту сприяють утворенню фотохімічного смогу, а свинець забруднює повітря в містах.

Таблиця 5.4 - Вплив різних галузей промисловості в забруднення атмосфери [33].

Галузь	Основні компоненти забруднюючих викидів
Теплоенергетика (25,7%)	Дим, що містить оксиди карбону, сульфуру та нітрогену, вуглеводні, пари води.
Чорна металургія (23,4%)	SO_2, пил, сполуки P, As, Hg, Pb, Mn.
Кольорова металургія (11,1%)	SO_2, CO_2, пил.
Нафтова і нафтохімічна промисловість (13,7%)	Аерозолі, неорганічний пил, оксиди нітрогену та карбону, SO_2, NH_2, H_2S, сполуки F.
Транспорт (11,6%)	CO, вуглеводні, оксиди нітрогену та сульфуру, сажа.
Будівельна промисловість (3,4%)	Пил, органічні сполуки, оксиди карбону, нітрогену та сульфуру.
Целюлозно-паперова промисловість	Органічні речовини, оксиди сульфуру, H_2S, Cl_2.
Гірничо-видобувна промисловість (7,1%)	Пил і гази, які утворюються під час вибухів.

Висновок до розділу 5

За допомогою охорони праці можна зменшити травматизм людини під час виконання роботи. Охорона праці встановлює норми та вимоги з техніки безпеки під час роботи.

Електробезпека - це система організаційних та технічних заходів і засобів, які спрямовані на захист людей від шкідливої та небезпечної дії електричного струму, електричної дуги, електричного поля та статичної електрики.

Корпуси електричних машин, джерел живлення, електричні апарати і інші частини, виконані з струмопровідних матеріалів, можуть опинитися під напругою при замиканні їх струмоведучих частин на корпус. Якщо корпус при цьому не має контакту із землею, дотик до нього так само небезпечне, як і дотик до фази.

Також розглянуто вплив промисловості, енергетики та транспорту на навколишнє середовище, зокрема їх роль у забрудненні атмосфери через викиди різних хімічних сполук. Важливими джерелами забруднення є теплоенергетика, чорна та кольорова металургія, а також транспорт і нафтова промисловість. Усі ці фактори можуть сприяти змінам клімату, кислотним дощам та іншим серйозним екологічним проблемам, що вимагають комплексного підходу до охорони довкілля та сталого розвитку.

ВИСНОВКИ

Підсумовуючи основні результати дослідження, можна зазначити наступне:

1. Розроблено автоматизовану систему управління абіотичними факторами в птахокомплексах, що забезпечує створення оптимальних умов для утримання птаства, підвищуючи ефективність і безпеку процесу. Система включає інтеграцію різних пристроїв на базі IoT для моніторингу і контролю мікроклімату.

2. Створено апаратно-програмний комплекс для автоматизованого управління птахокомплексом, що включає мікроконтролери, сенсори, дисплеї та реле, які забезпечують точне і безперервне управління умовами. Окрема увага приділена вибору сенсорів для визначення фізичних параметрів, що впливають на здоров'я птахів.

3. Розроблено мобільний додаток для моніторингу та управління мікрокліматом птахокомплексу, який дозволяє користувачам зручно налаштовувати та контролювати систему, змінювати параметри в реальному часі і отримувати сповіщення про критичні зміни.

4. Створено модель машинного навчання для виявлення пожеж у птахівництві, що дозволяє своєчасно реагувати на загрози та підвищує безпеку птахокомплексу, аналізуючи абіотичні фактори, що можуть вказувати на ризик виникнення пожежі.

5. Забезпечено охорону праці ы навколишнього середовища, а також безпеку в роботі з автоматизованими системами, зокрема розглянуто питання електробезпеки, захисту від механічних і програмних загроз, а також забезпечення кібербезпеки для IoT-рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аніщенко, В. О. Основи екології: [навч. посіб. для вищ. навч. закл.]. Київ: Кондор, 2009.
2. Ncho, C. M., Berdos, J., Gupta, V., & Rahman, A. Abiotic stressors in poultry production: A comprehensive review. *Journal of Animal Physiology and Animal Nutrition*. 2024. DOI: https://doi.org/10.1111/jpn.14032.
3. Лимкар, К. Р., & Тамболи, Ф. А. Влияние автоматизации. *Міжнародний журнал наукових досліджень у сфері сучасної науки і технологій*. 2024. № 3(8). С. 13-17. DOI: https://doi.org/10.59828/ijsrmst.v3i8.243.
4. Haight, J. M., & Kecojevic, V. Automation vs. Human intervention: What is the best fit for the best performance? *Production and Operations Management*. 2005. Т. 24, № 1. С. 45-51. DOI: https://doi.org/10.1002/prs.10050.
5. Stouffer, K., Falco, J., & Kent, K. Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security (Special Publication 800-82). *National Institute of Standards and Technology*. 2006. URL: https://www.researchgate.net/publication/246350235.
6. Sethi, P., & Sarangi, S. R. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*. 2017. Article ID 9324035. DOI: https://doi.org/10.1155/2017/9324035.
7. Gurusamy, M., Uma Swarupa, P., Gahlot, B., Ratnesh, K., & Kalra, G. Application of Internet of Things (IoT) in Various Industries. *European Journal of Molecular & Clinical Medicine*. 2020. Т. 7, № 2. С. 3277-3284. DOI: https://doi.org/10.13140/RG.2.2.23697.66403.
8. Goud, K. S., & Sudharson, A. Internet based Smart Poultry Farm. *Indian Journal of Science and Technology*. 2015. Т. 8, № 19. DOI: https://doi.org/10.17485/ijst/2015/v8i19/76227.
9. Idowu, P. O., Adegbola, O. A., Solomon, I. D., Adeagbo, M. A., & Ojo, J. A. Design and implementation of smart-controlled poultry farm management system. *International Journal of Enhanced Research in Science, Technology &*

Engineering. 2021. Т. 10, № 9. С. 16. URL: https://www.researchgate.net/publication/357092162_Design_and_Implementation_of_Smart-Controlled_Poultry_Farm_Management_System.

10. Wahyuni, R., Irawan, Y., Febriani, A., Nurhadi, N., Saputra, H. T., & Andrianto, R. Smart egg incubator based on IoT and AI technology for modern poultry farming. *ILKOM Jurnal Ilmiah*. 2024. Т. 16, № 2. С. 134-144. DOI: 10.33096/ilkom.v16i2.1957.134-144.

11. Кашьяп, М., Дев, А. К., Шарма, В. Реализация и анализ брокера EMQX для протокола MQTT в Интернете вещей. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*. 2024. DOI: https://doi.org/10.1016/j.prime.2024.100846.

12. Srinivasan, L., Scharnagl, J., & Schilling, K. Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation. *IFAC Proceedings Volumes*. 2013. Т. 46, № 29. С. 83-88. DOI: 10.3182/20131111-3-kr-2043.00032.

13. Інтернет-ресурс. RS Components Ltd. Product datasheet: Raspberry Pi 3 Model B+. *RS Components*. 2018. URL: https://us.rs-online.com/m/d/4252b1ecd92888dbb9d8a39b536e7bf2.pdf.

14. Інтернет-ресурс. SHENZHEN EONE ELECTRONICS CO., LTD. 2004A LCD Module Specification Ver1.0. 14 стор. Отримано з: https://www.example-link.com.

15. Інтернет-ресурс. RCS Components. SNIS159B - 3.3V/5V Logic-Level Interface IC. 14 стор. Отримано з: https://www.rcscomponents.kiev.ua/datasheets/snis159b-186967.pdf.

16. Honeywell. HIH-5030 Humidity Sensor. Datasheet, версія 1.0. доступно на: www.alldatasheet.com, доступ 21 листопада 2024 року.

17. Никамин, В. А. Аналого-цифровые и цифро-аналоговые преобразователи. Санкт-Петербург: КОРОНА принт; Альтекс-А, 2003. 224 с.

18. Інтернет-ресурс. Arduino.ua. 10-разрядний АЦП MCP3008 для Raspberry Pi і Arduino. *Arduino.ua*. URL: https://arduino.ua/ru/prod1931-10-razryadnii-acp-mcp3008-dlya-raspberry-pi-i-arduino.

19. Інтернет-ресурс. Hands-On Technology. 1-Channel Relay Module Datasheet. *Hands-On Technology*. URL: https://handsontec.com/dataspecs/relay/1Ch-relay.pdf.

20. Медвідь, В. Р., Пісьціо, В. П. Проектування та аналіз електричних схем в програмному середовищі Proteus VSM. Методичні вказівки до самостійної роботи студентів курсу "Проектування мікропроцесорних систем керування технологічними процесами". Тернопіль: ТНТУ, 2018. 26 с.

21. Бредбері, А., Еверард, Б. Learning Python with Raspberry Pi. Packt Publishing, Limited, 2022. 248 с.

22. Mitra, R., Ganiga, R. A novel approach to sensor implementation for healthcare systems using internet of things. *ResearchGate*. 2024. URL: https://www.researchgate.net/publication/337664623_A_novel_approach_to_sensor_implementation_for_healthcare_systems_using_internet_of_things.

23. Інтернет-ресурс. GitHub. Hello World. *GitHub Docs*. URL: https://docs.github.com/ru/get-started/start-your-journey/hello-world, доступ 6 грудня 2024 року.

24. Künneth, T. Android UI Development with Jetpack Compose: Bring declarative and native UIs to life quickly and easily on Android using Jetpack Compose. Packt Publishing, Limited, 2022.

25. Інтернет-ресурс. Eclipse Paho. MQTT Java Client. *GitHub*. URL: https://github.com/eclipse-paho/paho.mqtt.java.

26. Fundamental Concepts of Convolutional Neural Network. In: Recent Trends and Advances in Artificial Intelligence and Internet of Things. Intelligent Systems Reference Library, 2020. DOI: 10.1007/978-3-030-32644-9_36.

27. Akgül, İ. A Pooling Method Developed for Use in Convolutional Neural Networks. *Computer Modeling in Engineering & Sciences*. 2024. С. 1-10. DOI: https://doi.org/10.32604/cmes.2024.052549.

28. Інтернет-ресурс. Kaggle. Fire Dataset. *Kaggle*. URL: https://www.kaggle.com/datasets/phylake1337/fire-dataset.

29. Li, K.-C., Gupta, B. B., & Agrawal, D. P. Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS). CRC Press, 2021. ISBN 9780367220655.

30. Künneth, T. Android UI Development with Jetpack Compose: Bring declarative and native UIs to life quickly and easily on Android using Jetpack Compose. Packt.

31. Голінько, В. І. Основи охорони праці: підручник / В. І. Голінько; М-во освіти і науки України; Нац. гірн. ун-т. - 2-ге вид. - Д.: НГУ, 2014. - 271 с.

32. Закон України "Про охорону навколишнього природного середовища" № 1264-XII від 25 червня 1991 року

33. Хроленко І.А. Основи екології: курс лекцій. Ел. варіант.

34. Інтернет-ресурс. Хешування паролів: використання солі та bcrypt. URL: drukarnia.com.ua/articles/kheshuvannya-paroliv-vikoristannya-soli-ta-bcrypt-fsme-.