Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Електронної та біомедичної інженерії_____
(повна назва)

Кафедра ____Кафедра мікроелектроніки, електронних приладів та пристроїв___
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____перший (бакалаврський)_____

_____Автоматизована система керування дроном_____
_____за допомогою мобільного телефона_____
(тема)

Виконав:
студент _4__ курсу, групи ___МНТМН-21-1___
_____Биков Родіон Ігорович_____
(прізвище, ініціали)

Спеціальність 153 Мікро- та наносистемна техніка_____
(код і повна назва спеціальності)
Тип програми ____ освітньо-професійна_____
Освітня програма: «Мікро та наноелектроніка»_____
(повна назва освітньої програми)

Керівник ст. викл. каф. МЕЕПП_____
_____Васильєв Ю.С._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _Бондаренко І.М._____
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет_____ Електронної та біомедичної інженерії _____

Кафедра_____Мікроелектроніки, електронних приладів та пристроїв_____

Рівень вищої освіти_____перший (бакалаврський)_____

Спеціальність_____153 Мікро- та наносистемна техніка_____
(код і повна назва)

Тип програми _____освітньо-професійна_____

Освітня програма_____Електронні пристрої та системи_____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Бикову Родіону Ігоровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система керування дроном за допомогою мобільного телефона
_____
_____
_____

затверджена наказом університету від _26.05_____2025 р. № __414 Ст_____

2. Термін подання студентом роботи до екзаменаційної комісії _10_ ____06____ 2025 р.

3. Вихідні дані до роботи _____

Рама дрону_____

Бесщіткові мотори (4 шт.)_____

ESC (Electronic Speed Controller, 4 шт.)_____

Пропелери (4 шт.)_____

Акумулятор (Li-Po 3S/4S)_____

Плата керування_____

GPS – модуль_____

Гіроскоп та акселерометр_____

Android Studio_____

4. Перелік питань, що потрібно опрацювати в роботі: 1. Аналіз сучасних безпілотних систем керування. 2. Обґрунтування вибору архітектури автоматизованої системи. 3. Розробка структурної схеми системи. 4. Проєктування та реалізація мобільного додатку. 5. Програмування мікроконтролера дрона. 6. Тестування системи в лабораторних і реальних умовах. 7. Оцінка результатів роботи системи. 8. Виявлення проблем та шляхи їх усунення. 9. Висновки та перспективи застосування

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
_слайд-презентація (15 слайдів), схема електрична структурна.__

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	05.05.2025	Виконано
2	Аналітичний огляд джерел	10.05.2025	Виконано
3	Оформлення пояснювальної записки	14.05.2025	Виконано
4	Вибір структури приладу	18.05.2025	Виконано
5	Розробка та написання програм	21.05.2025	Виконано
6	Оформлення креслеників	26.05.2025	Виконано
7	Підготовка презентації	29.05.2025	Виконано
8	Рецензування, нормоконтроль	30.05.2025	Виконано
9	Захист роботи	13.06.2025	Виконано

Дата видачі завдання _____ _____2025 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи бакалавра містить:

48 сторінок, 22 рисунка, 5 додатків, 12 джерел, 6 таблиц.

АПАРАТНА ЧАСТИНА (ДРОН), ПРОГРАМНА ЧАСТИНА, ЗАСІБ ЗВ’ЯЗКУ МІЖ ДРОНОМ ТА СМАРТФОНОМ, СОФТ ДЛЯ РОЗРОБКИ

Об’єкт дослідження – безпілотний літальний апарат (дрон) як технічна система, здатна до дистанційного керування та автоматизації процесів.

Мета роботи – розробка та реалізація автоматизованої системи керування дроном за допомогою мобільного телефона, що забезпечує стабільне, зручне та безпечне управління польотом у реальному часі.

Метод дослідження – аналітичний метод, проєктно-конструкторський метод, експериментальний метод, моделювання.

Актуальність роботи – На сьогодні безпілотні літальні апарати широко застосовуються в різних сферах: аграрному секторі, відеозйомці, моніторингу територій, рятувальних операціях тощо. Актуальною є потреба у простому та мобільному способі керування такими пристроями, зокрема за допомогою смартфона, який є доступним для більшості користувачів. Розробка автоматизованої системи, що дозволяє керувати дроном за допомогою мобільного телефона, сприятиме спрощенню експлуатації БПЛА, зниженню витрат на обладнання і підвищенню зручності користування.

ABSTRACT

The explanatory note for the bachelor's qualification work contains:
48 pages, 22 figures, 5 appendices, 12 sources, 6 tables.

HARDWARE PART (DRONE), SOFTWARE PART, COMMUNICATION MEANS BETWEEN THE DRONE AND THE SMARTPHONE, SOFTWARE FOR DEVELOPMENT

The object of research is an unmanned aerial vehicle (drone) as a technical system capable of remote control and automation of processes.

The purpose of the work is to develop and implement an automated drone control system using a mobile phone, which provides stable, convenient and safe flight control in real time.

Research method is analytical method, design and construction method, experimental method, modeling.

Relevance of the work - Today, unmanned aerial vehicles are widely used in various fields: the agricultural sector, videography, territory monitoring, rescue operations, etc. There is a pressing need for a simple and mobile way to control such devices, in particular using a smartphone, which is available to most users. The development of an automated system that allows you to control a drone using a mobile phone will simplify UAV operation, reduce equipment costs, and increase user convenien

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API - Application Programming Interface — програмний інтерфейс прикладного програмування

ESC - Electronic Speed Controller — електронний регулятор швидкості обертання двигуна

GPS - Global Positioning System — глобальна система позиціонування

GUI - Graphical User Interface — графічний інтерфейс користувача

HTTP - HyperText Transfer Protocol — протокол передавання гіпертексту

IMU - Inertial Measurement Unit — інерціальний вимірювальний модуль

JSON - JavaScript Object Notation — формат обміну структурованими даними

PID - Proportional-Integral-Derivative — ПІД-регулятор (пропорційно-інтегрально-диференціальний)

PWM - Pulse Width Modulation — широтно-імпульсна модуляція

TCP - Transmission Control Protocol — протокол керування передаванням

UDP - User Datagram Protocol — протокол користувацьких датаграм

UART - Universal Asynchronous Receiver-Transmitter — універсальний асинхронний приймач-передавач

Wi-Fi - Wireless Fidelity — стандарт бездротової локальної мережі

ВСТУП

Сучасний етап розвитку техніки характеризується активним впровадженням автоматизованих систем у різні сфери людської діяльності. Особливу увагу привертають безпілотні літальні апарати (БПЛА), або дрони, які отримали широке застосування в цивільній, промисловій, сільськогосподарській, екологічній, оборонній та науково-дослідницькій галузях. Їх використання дозволяє зменшити участь людини в небезпечних або трудомістких процесах, автоматизувати моніторинг територій, здійснювати аерофотозйомку, доставку вантажів тощо.

У зв’язку з цим зростає потреба в простих, надійних та доступних засобах керування такими апаратами. Одним з перспективних напрямів є використання мобільних пристроїв, зокрема смартфонів, як інтерфейсу для дистанційного керування дроном. Такий підхід забезпечує зручність, мобільність та економічну ефективність, що особливо важливо для широкого кола користувачів.

Однак розробка подібної системи вимагає врахування низки технічних аспектів: вибору способу зв’язку між дроном і мобільним пристроєм, розробки програмного забезпечення, забезпечення стабільного польоту, реалізації базових команд та автоматизації окремих функцій керування.

1 АНАЛІТИЧНИЙ ОГЛЯД. ОГЛЯД ЛІТЕРАТУРИ

1.1 Аналіз сучасних систем керування безпілотними літальними апаратами

Упродовж останнього десятиліття безпілотні літальні апарати (БПЛА), або дрони, знайшли широке застосування у військовій, цивільній та науково-дослідній сферах. Поряд із поширенням апаратної частини зросла і потреба в ефективних засобах керування, серед яких особливу увагу привертає управління за допомогою мобільних пристроїв.

Існують різні підходи до реалізації систем керування дронами:

Традиційні RC-системи (пульти дистанційного керування) забезпечують високу точність та надійність, однак мають обмежені функціональні можливості і не інтегровані з мережею. (Рис. 1.1)

Рисунок 1.1 – Датчик RC-системи

Системи з використанням ПК дозволяють планувати автономні польоти, працювати з телеметрією та даними датчиків, але потребують громіздкого обладнання.

Мобільні системи керування (смартфони, планшети) є найперспективнішими завдяки компактності, багатофункціональності та доступності. Мобільні додатки дозволяють використовувати GPS,

акселерометр, камеру, та інші сенсори пристрою для розширення можливостей управління. (Рис. 1.2)

Рисунок 1.2 – Приклад керування дроном за допомогою мобільного пристрою

Таким чином, обґрунтованим є вибір мобільної платформи як основного інтерфейсу користувача в системі автоматизованого керування дроном.

1.2 Канали зв’язку та способи передачі команд

Для керування дроном мобільними пристроями найчастіше використовують наступні канали зв’язку:

Bluetooth:

- простий у реалізації, але має обмежену дальність (до 10–20 м);
- підходить для невеликих проєктів у приміщенні;

Wi-Fi:

- поширений варіант із дальністю до 100 м у відкритому просторі;
- дає змогу передавати дані високої швидкості;
- можна створити локальну мережу між дроном і телефоном без доступу до Інтернету;

4G/5G:

- дає можливість керування дроном з будь-якої точки, де є інтернет;
- використовується в комерційних і промислових рішеннях;
- вимагає складнішої інфраструктури та опрацювання затримок;

RF-модулі (наприклад, nRF24, LoRa):

- застосовуються у саморобних або промислових рішеннях для великої дальності передачі команд;
- мають свої особливості в налаштуванні.

Таким чином можна вважати, що для проєкту з мобільним додатком найдоцільніше використання Wi-Fi або Bluetooth залежно від сценаріїв використання.

1.3 Інтерфейси обміну даними та програмне забезпечення для мобільного керування

Серед найбільш популярних протоколів та бібліотек, які використовуються для обміну даними між дроном і контролером:

MAVLink (Micro Air Vehicle Link):

- один із найпоширеніших відкритих протоколів для зв’язку з автопілотами типу Pixhawk або ArduPilot;
- підтримується великою кількістю мобільних SDK. – Має функції телеметрії, передачі команд, обміну даними з датчиків;

Custom протоколи:

- у випадку саморобного дрона на базі, наприклад, Arduino, ESP32, Raspberry Pi – створюються власні прості протоколи (JSON, HTTP-запити, WebSocket, UDP);
- гнучкі, але потребують ретельної реалізації обробки помилок.

Розглянемо кілька готових SDK і бібліотек для реалізації мобільного керування дроном:

DJI Mobile SDK (Android/iOS) – Професійне рішення для роботи з дроном DJI. Має багатий API для контролю камери, польоту, телеметрії. (Рис. 1.3)

Рисунок 1.3 – Вигляд програмного забезпечення DJI Mobile SDK

QGroundControl – Універсальна платформа для роботи з MAVLink-дронами. Підтримує ПК і мобільні пристрої. Можна використовувати або як готове рішення, або як базу для кастомізації. (Рис. 1.4)

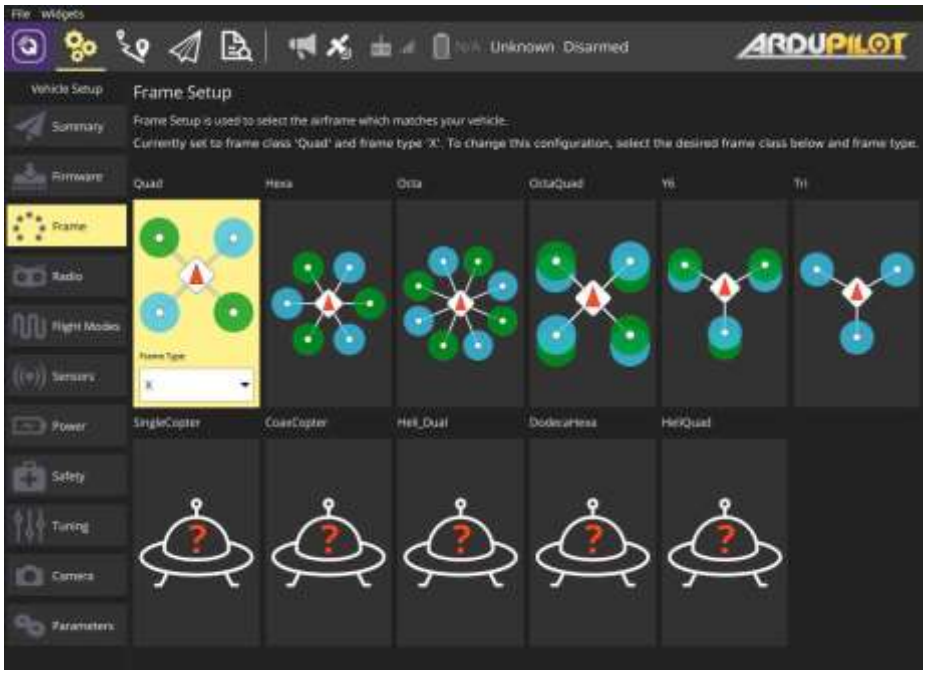

Рисунок 1.4 – Вікно налаштування програми QGroundControl

Android Bluetooth / Wi-Fi API – Стандартні засоби Android SDK для створення зв’язку з апаратною частиною. Гнучкі, але потребують глибшого програмування.

1.4 Аналіз проблем і викликів

Серед основних проблем, що виникають при реалізації систем мобільного керування дроном, слід виділити кілька критичних технічних аспектів, які суттєво впливають на стабільність, надійність та безпечність функціонування всієї системи.

Однією з ключових проблем є затримка передачі команд, що може призводити до запізнілих реакцій дрона на дії оператора, особливо при використанні мереж із високою латентністю або нестабільним з’єднанням. Це є критичним у завданнях, що потребують високої точності або оперативного реагування.

Другою важливою проблемою є перешкоди в радіоканалі, які можуть виникати через інші бездротові пристрої або об'єкти, що заважають передачі сигналу, знижуючи якість зв’язку або призводячи до повної його втрати.

Не менш значущим є питання безпеки з'єднання, особливо при використанні Wi-Fi, де існує ризик несанкціонованого доступу, перехоплення або підміни команд, що може мати небезпечні наслідки як для самого дрона, так і для навколишнього середовища. У разі аварійного відключення зв’язку, наприклад, через втрату сигналу або збій в мережі, надзвичайно важливою є наявність ефективного механізму fail-safe, який забезпечить повернення дрона до точки старту або його безпечну посадку.

Останньою, але не менш важливою проблемою є несумісність пристроїв, яка може виникати при використанні мобільних пристроїв з різними операційними системами, що ускладнює інтеграцію апаратного та програмного забезпечення дрона з керуючими додатками, обмежуючи функціональність або викликаючи збої в роботі.

Усі ці проблеми потребують комплексного технічного вирішення задля забезпечення стабільного та безпечного функціонування систем мобільного керування безпілотними літальними апаратами.

2 ПРОЦЕС РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ

2.1 Мета розробки

Метою дипломної роботи є розробка функціональної, надійної та доступної автоматизованої системи управління безпілотним літальним апаратом (далі – БПЛА або дрон), котра дозволяє здійснювати дистанційне керування польотом з використанням мобільного пристрою на платформі Android. Основною ідеєю є створення альтернативи традиційним пультам управління з розширеними можливостями мобільного інтерфейсу.

Розроблювана система повинна гарантувати:

- інтуїтивно зрозуміле управління дроном – завдяки графічному інтерфейсу на мобільному пристрої, користувач повинен мати можливість виконувати базові команди: зліт, посадка, рух у різних напрямках, стабілізація;
- отримання телеметричних даних у реальному часі, на кшталт: висоти, координат (GPS), рівня заряду акумулятора, швидкості руху, відстані до оператора тощо. Це дає змогу здійснювати базовий моніторинг стану дрона під час польоту;
- безперервний та стабільний зв’язок між дроном та мобільним пристроєм – через бездротовий канал зв’язку (Wi-Fi або Bluetooth), який повинен забезпечувати низьку затримку обміну даними та мінімальну кількість помилок;
- можливість масштабування та розширення системи – розроблена система має бути відкритою для подальшого вдосконалення: додавання алгоритмів автономної навігації, підтримка відеопотоку з камери дрона, інтеграція з хмарними сервісами або мережами IoT;
- доступність та гнучкість реалізації – програмне та апаратне забезпечення має базуватись на широкодоступних компонентах з відкритими або вільно доступними SDK/API, що дозволить у

майбутньому адаптувати систему під інші моделі дронів або нові вимоги.

В рамках даної мети планується реалізувати повноцінний прототип, що складається з трьох ключових компонентів:

- мобільний застосунок, котрий виконує функцію інтерфейсу користувача і передає керуючі команди до дрона;
- мікроконтролерна система на борту дрона, що приймає команди, взаємодіє з сенсорами та приводами;
- канал бездротового зв’язку, котрий забезпечує обмін даними між пристроями у режимі реального часу.

Реалізація поставленої мети дозволить створити зручну систему управління дроном для застосування у побуті, освіті, дослідженнях чи промислових задачах, де потрібна мобільність та простота використання.

2.2 Умови реалізації проекту

Необхідно врахувати сукупність технічних, програмних, функціональних та середовищних передумов, які впливають на розробку. Ці умови визначають обмеження, можливості й особливості побудови системи.

Апаратне забезпечення системи включає безпілотний літальний апарат (дрон), який має бути обладнаний контролером польоту — таким як Pixhawk, APM або власна плата на основі мікроконтролера типу ESP32 чи STM32. Окрім цього, дрон повинен мати електродвигуни, ESC-регулятори, джерело живлення та навігаційні сенсори, зокрема GPS і IMU. У ролі мікроконтролера або автопілота можуть використовуватися ESP32, Pixhawk чи APM, які забезпечують підтримку бездротового зв’язку та керування виконавчими механізмами. Для реалізації бездротового зв’язку застосовуються модулі Wi-Fi (на базі ESP32 або ESP8266) або Bluetooth (Classic або BLE). Система також передбачає використання мобільного пристрою з операційною системою Android версії 8.0 або вище, що підтримує відповідні типи зв’язку.

Програмне забезпечення системи передбачає використання Android Studio для розробки мобільного застосунка, при цьому основними мовами програмування є Java або Kotlin. Для реалізації обміну даними з дроном можуть використовуватись MAVLink SDK, DJI SDK або власні реалізації відповідних протоколів. Для програмування та прошивки контролера застосовується Arduino IDE або інше сумісне середовище розробки.

Зовнішні умови та середовище експлуатації системи залежать від типу використовуваного зв’язку: для Bluetooth радіус дії становить до 15 метрів, а для Wi-Fi — до 100 метрів. Тестування передбачено проводити на відкритих майданчиках без перешкод, що забезпечує стабільний зв’язок і безпечний політ. Окрім технічних аспектів, необхідно враховувати чинні юридичні обмеження щодо використання дронів, зокрема щодо максимально допустимої висоти польоту та зон, де їх використання заборонене.

Інформаційна база для розробки системи включає документацію до обладнання, яка містить технічні характеристики та інструкції з використання. Важливими джерелами також є відкриті SDK та API, що забезпечують інтеграцію й управління пристроями. Додаткову підтримку надають репозиторії з прикладами реалізацій, розміщені на GitHub або офіційних сайтах виробників. Для поглибленого вивчення теми використовуються наукові статті та обговорення на тематичних форумах.

2.3 Основні вимоги до системи

Функціональні вимоги до системи передбачають, що мобільний додаток має забезпечувати встановлення стабільного з’єднання з дроном, передавання команд для керування його польотом, відображення телеметричних даних у реальному часі, а також обробку помилок зв’язку та повідомлень про стан системи. Зі свого боку, дрон (а саме мікроконтролер, встановлений на ньому) повинен приймати ці команди, обробляти їх та виконувати, генерувати й

надсилати телеметричну інформацію, а також взаємодіяти з сенсорами та приводами, що забезпечують фізичне управління дроном.

Така функціональна структура є логічною та ефективною для базової взаємодії між дроном і користувачем. Вона охоплює ключові аспекти безпечного та стабільного керування, а також дозволяє масштабувати систему — наприклад, додавати підтримку автономного режиму польоту чи інтелектуальної обробки даних із сенсорів у майбутньому.

Нефункціональні вимоги до системи передбачають, що інтерфейс мобільного додатка має бути інтуїтивно зрозумілим і зручним для користувача, що особливо важливо в умовах керування дроном, де швидке реагування має вирішальне значення. Система повинна працювати у режимі реального часу або з мінімальними затримками, щоб забезпечити точне та оперативне керування дроном. Обмін даними між додатком і дроном має бути захищеним від стороннього втручання, бажано з використанням механізмів шифрування, особливо якщо передається конфіденційна або критична інформація. Важливо також, щоб програмний код був модульним, придатним для тестування й подальшого розширення функціональності, що забезпечить легке масштабування проєкту. Окрім того, застосунок повинен бути сумісним з Android-пристроями, починаючи з версії 8.0 (Oreo) і вище, що дозволить охопити широке коло користувачів.

Ці вимоги є добре збалансованими: вони охоплюють як технічні, так і користувацькі аспекти розробки. Особливо актуальним є акцент на реальному часі й модульності — це підвищує надійність і гнучкість системи в умовах динамічного середовища експлуатації.

3 ПРОЄКТУВАННЯ СИСТЕМИ

3.1Архітектура програмного та апаратного забезпечення

Система складається з двох основних частин: клієнтської та серверної (виконавчої):

- клієнтська частина представлена мобільним додатком, який встановлюється на смартфон або планшет користувача та виконує функції надсилання команд, візуалізації телеметричних даних і керування польотом у зручному для користувача форматі;
- серверна або виконавча частина реалізована у вигляді дрона — апаратної системи, що включає мікроконтролер (наприклад, ESP32 або Pixhawk), модулі бездротового зв’язку, навігаційні сенсори (IMU, GPS), виконавчі механізми (ESC та електродвигуни), а також блок живлення, який забезпечує енергозалежність усіх компонентів системи.

Програмне забезпечення реалізується як взаємодія двох прошивок/програм:

- ПЗ мікроконтролера, що опрацьовує команди від мобільного додатку;
- мобільного застосунку, що реалізує інтерфейс і логіку керування.

Для наочності створимо схему за принципом "мобільний пристрій → передача команди → обробка → виконання"(Рис. 3.1), що дозволяє чітко відобразити логіку взаємодії між користувачем і дроном, підвищуючи зрозумілість та ефективність реалізації системи.

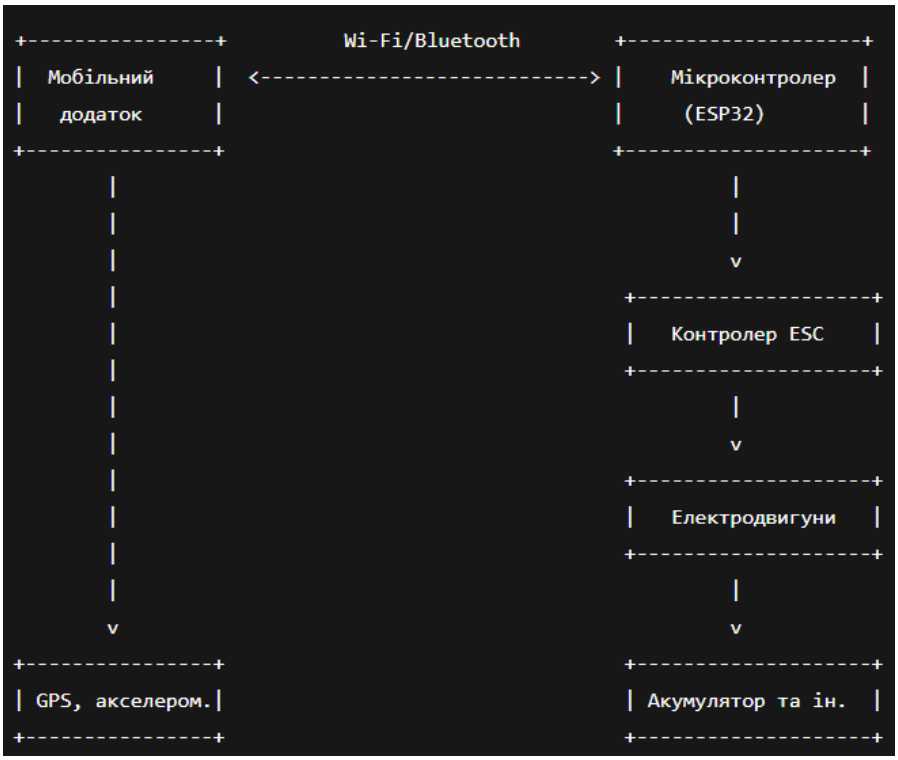

Рисунок 3.1 – Основні компоненти та їх взаємодія

Як видно зі схеми, управління дроном реалізується через чітку послідовність: команда формується на мобільному пристрої, передається по бездротовому каналу, обробляється мікроконтролером дрона та виконується відповідними механізмами. Така структура забезпечує модульність і дає змогу легко масштабувати систему або інтегрувати додаткові функції, зокрема автономну навігацію чи зворотний зв’язок із сенсорів.

3.2 Вибір модулів зв’язку

Вибір модуля зв’язку є критичним аспектом проєкту, оскільки саме він визначає ефективність і надійність обміну даними між компонентами системи. Основними критеріями при виборі є: дальність дії — впливає на можливість забезпечення зв’язку на необхідній відстані; стабільність зв’язку — гарантує безперебійність і стійкість передачі даних у різних умовах; споживання енергії — критично важливо для автономних пристроїв із живленням від

батарей; затримка сигналу — визначає швидкість реакції системи, що особливо важливо для реального часу.

З огляду на ці параметри, нижче представлено порівняльну таблицю (Таблиця 3.1) популярних технологій бездротового зв’язку, які можуть бути використані в проєкті.

Таблиця 3.1 – Порівняльна характеристика

Технологія	Радіус дії	Швидкість	Затримка	Переваги	Недоліки
Bluetooth	~10–15 м	До 2 Мбіт/с	Низька	Простота реалізації	Малий радіус, перешкоди
Bluetooth Low Energy (BLE)	~10 м	До 1 Мбіт/с	Дуже низька	Енергоефективність	Обмежена передача даних
Wi-Fi	~50–100 м	До 150 Мбіт/с	Середня	Висока швидкість, гнучкість	Більше енергоспоживання
ILoRa	~1–2 км+	До 50 Кбіт/с	Висока	Дальність, стійкість	Дуже низька пропускна здатність

У даному випадку оптимальним вибором для реалізації бездротового зв’язку є використання модуля Wi-Fi на базі ESP32 (Рис. 3.2), оскільки він забезпечує прийнятну дальність дії для побутових або тестових сценаріїв.

Рисунок 3.2 – Модуль Wi-Fi на базі ESP32

Крім того, ESP32 дозволяє створити стабільне з’єднання без потреби у зовнішньому маршрутизаторі, оскільки може працювати в режимі точки

доступу. Ще однією перевагою є повна сумісність з Android-пристроями завдяки підтримці стандартних бібліотек, що значно спрощує процес розробки мобільного додатка та налагодження взаємодії між клієнтською та виконавчою частинами системи.

3.3 Інтерфейс мобільного додатка

Інтерфейс мобільного додатка є одним із ключових елементів системи керування дроном, оскільки саме через нього користувач здійснює повноцінну взаємодію з апаратом (Рис. 3.3). Через цей інтерфейс передаються команди, отримуються телеметричні дані, відображаються повідомлення про стан системи, а також реалізуються всі ключові функції, необхідні для безпечного й ефективного польоту.

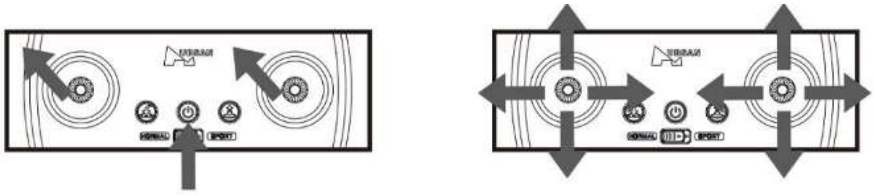

Рисунок 3.3 – Інтерфейс керування дроном

Головний екран застосунку виступає центральною панеллю управління, на якій розташовані віртуальні джойстики або керуючі кнопки для зміни напрямку руху, висоти, а також активації зльоту, посадки та стабілізації. Основні елементи керування мають бути великими, чіткими та зручно розміщеними для комфортного використання навіть у стресових умовах або на відкритому повітрі.

Панель статусу забезпечує постійне оновлення критично важливих даних — рівень заряду акумулятора, висота польоту, координати GPS, поточний режим роботи, а також стан зв’язку між пристроями (Рис. 3.4). Усе це дозволяє оператору контролювати ситуацію в реальному часі.

Рисунок 3.4 – Показники даних

Додатково, застосунок може включати модуль інтеграції з картографічними сервісами (наприклад, Google Maps API), що дозволяє виводити поточну позицію дрона на мапі, фіксувати маршрут польоту, визначати домашню точку або планувати траєкторію у випадку реалізації автономного режиму. Живий журнал подій фіксує всі ключові події та повідомлення, серед яких — втрати зв’язку, помилки, зміни режиму, попередження про низький заряд батареї тощо.

Меню налаштувань забезпечує гнучкість і адаптацію до потреб користувача(Рис. 3.5- 3.6). Тут можна обрати тип з’єднання (Wi-Fi або Bluetooth), вказати параметри підключення, налаштувати чутливість керування, активувати або вимкнути карту, змінити мову інтерфейсу чи зовнішній вигляд (світла/темна тема).

Рисунок 3.5 – Основне меню налаштування

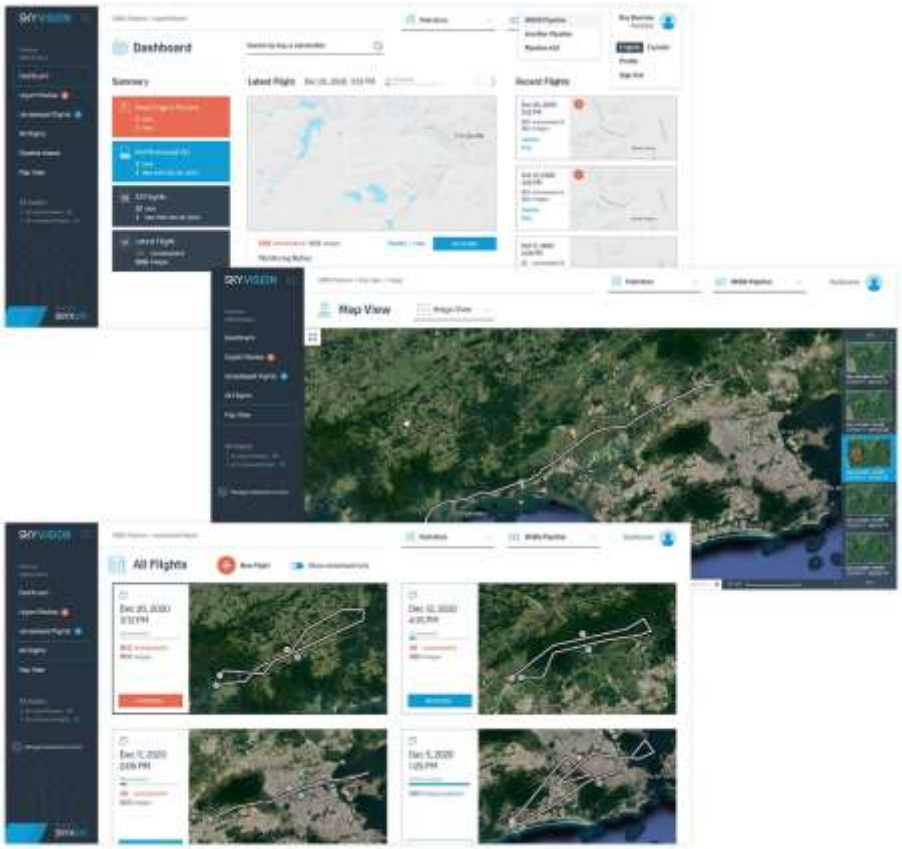

Рисунок 3.6 – Налаштування карти

Окрему увагу слід приділити дизайну інтерфейсу. Він має бути мінімалістичним, з акцентом на швидкий доступ до найважливіших функцій. Візуальні елементи повинні залишатися контрастними й добре помітними навіть при яскравому сонячному освітленні. Також необхідно передбачити автоматичне перепідключення при втраті зв’язку, що забезпечує безперервність управління. Зворотний зв’язок — вібрація, анімації натискання, звукові сигнали — покращує сприйняття дій користувача і підвищує точність керування.

Безпека є ще одним важливим аспектом, тому критичні функції, такі як аварійна зупинка, мають бути захищені від випадкового натискання (наприклад, через підтвердження дії). Інтерфейс також повинен показувати попередження при виявленні помилок або загроз, надаючи можливість оператору швидко зреагувати. Окрім цього, мобільний додаток має бути

розроблений з урахуванням можливого розширення: підтримка відеопотоку, запис телеметрії, підключення до хмарних сервісів, підтримка профілів користувача — усе це має бути враховано на рівні структури інтерфейсу та архітектури застосунку.

Отже, інтерфейс мобільного додатка — це не просто візуальне оформлення, а повноцінний інструмент управління, який напряму впливає на якість польоту, зручність взаємодії, безпеку експлуатації та загальний досвід користувача. Його розробка повинна базуватись на принципах ергономіки, швидкої реакції, адаптивності до умов експлуатації та логічної організації всіх функцій.

3.4 Надійність управління

Заходи безпеки в системі управління дроном є критично важливими для забезпечення стабільної, контрольованої та безпечної експлуатації як для користувача, так і для оточення. Одним із основних напрямів є фізичні обмеження, які реалізуються програмно через встановлення максимально допустимої висоти та дальності польоту. Це дозволяє уникнути виходу дрона за межі дозволеного радіусу дії, а також відповідати чинним нормативним обмеженням.

У разі виникнення аварійної ситуації, наприклад, втрати зв’язку з мобільним пристроєм або досягнення критичного рівня заряду акумулятора, система повинна автоматично ініціювати посадку. Така функція дозволяє мінімізувати ризик падіння дрона або його втрати. На рівні мікроконтролера передбачена перевірка коректності команд, що надходять із мобільного застосунку — це запобігає виконанню помилкових або потенційно небезпечних інструкцій, які можуть спричинити аварійну ситуацію. Для додаткового захисту передбачене опціональне шифрування з’єднання, зокрема через використання протоколів WPA2 у випадку передачі даних через Wi-Fi, що знижує ризик несанкціонованого доступу до управління дроном.

Надійність системи також забезпечується за рахунок дублювання ключових функцій — наприклад, наявність окремої аварійної кнопки для екстреної посадки або зупинки дрона. Перед виконанням будь-яких команд система проходить валідацію стану, зокрема перевірку, чи перебуває дрон у повітрі, чи готові всі системи до дій, і лише після цього виконується команда. Для запобігання зависанням або збоям у роботі мікроконтролера використовуються watchdog-таймери, які здійснюють автоматичне перезавантаження системи в разі втрати стабільності.

Усі ці заходи в сукупності формують багаторівневу систему безпеки, яка забезпечує контроль над критичними параметрами польоту, стійкість до відмов та захист від людських і технічних помилок, що є обов’язковими вимогами для сучасних безпілотних систем.

4 РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі представлено практичну реалізацію автоматизованої системи керування дроном, яка включає мобільний додаток, вбудоване програмне забезпечення дрона та засоби їх взаємодії. Детально описано етапи розробки, використані технології та специфіку інтеграції складових системи.

4.1 Опис розробленого мобільного додатку

Основні функції застосунку зосереджені на забезпеченні повноцінної взаємодії користувача з дроном у режимі реального часу.

З'єднання з дроном:

- wi-fi - точка доступу (ESP32) для передачі MAVLink-телеметрії.

Керування польотом:

- віртуальні кнопки: зліт, посадка, стабілізація, аварійна зупинка.

Телеметрія:

- індикатори: рівень заряду, висота, GPS-координати, швидкість.

Статус зв’язку:

- іконки чи смуги сигналу біля кнопок керування або у шапці додатку для швидкого контролю зв’язку.

Логування подій:

- журнал: стрічка або список подій (наприклад, "Зліт", "Втрачено зв’язок", "Посадка") — у цьому UI може бути вкладка або вікно “Logs”.

Подивимось на рисунок 4.1, та побачимо наглядно усі процеси функцій мобільного додатка.

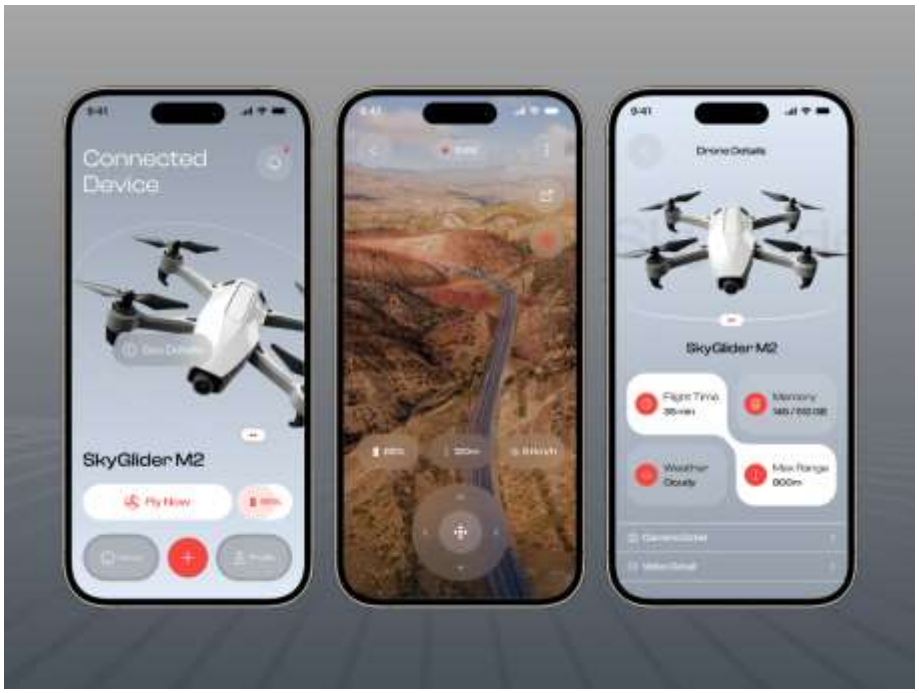

Рисунок 4.1 – Інтерфейс управління дроном

Структура додатку:

а) головний екран:

- дисплей відео/телеметрії;
- віртуальні кнопки польоту;
- індикація зв’язку;

б) налаштування:

- вибір типу з’єднання (Wi-Fi), IP-адреси ESP32;
- параметри контролю: чутливість стиків, формат клавіш, аварійна зупинка;

Телеметрія:

- окремий екран з потоковими графіками: висота, швидкість, заряд акумулятора, GPS;
- логування даних для відтворення або аналізу.

На рівні користувацького інтерфейсу (UI) взаємодія з дроном реалізується через продумане розміщення основних елементів, що забезпечують зручність та інформативність під час керування. Головний екран мобільного застосунку містить праворуч ілюстрацію або схему інтерфейсу, де

основні віртуальні кнопки керування розташовані в нижній частині екрана для зручного доступу великим пальцем. У верхній частині екрану відображаються ключові телеметричні дані — висота, заряд батареї, координати GPS, швидкість тощо. Такий підхід дозволяє оператору тримати всі важливі параметри під контролем без відволікання від процесу керування.

Меню налаштувань винесене у вигляді іконки "гамбургер" або як окрема вкладка з назвою «Settings». У цьому розділі користувач може вручну ввести IP-адресу дрона або вибрати її з доступного списку, налаштувати чутливість керування за допомогою селекторів, увімкнути чи вимкнути додаткові функції (наприклад, автоматичну посадку або карту), а також зберегти профілі підключення для різних сценаріїв використання.

Вкладка «Telemetry» надає доступ до розширених графіків, які можуть виводитися у повноекранному режимі або у вигляді компактних карток із ключовими метриками. Це дозволяє аналізувати поведінку дрона в реальному часі або під час післяпольотного огляду. Така структура UI забезпечує баланс між функціональністю, доступністю та наочністю, що є особливо важливим для ефективного польоту та своєчасного реагування на зміни в стані системи.

4.2 Реалізація взаємодії з дроном

Комунікація між мобільним пристроєм і дроном здійснюється за допомогою Wi-Fi-з'єднання (ESP32 працює як точка доступу, мобільний пристрій підключається до неї). Протокол обміну команд виглядає наступним чином (Рис. 4.2):

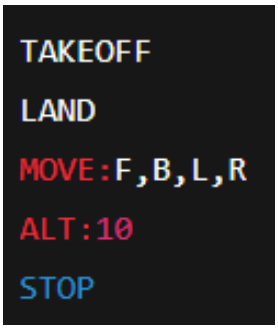

Рисунок 4.2 – Використання простого протоколу через UDP

Дані надсилаються у вигляді окремих текстових рядків через протокол UDP, що дозволяє передавати інформацію швидко та з мінімальними затримками. Кожен рядок містить завершену команду або фрагмент телеметрії, що спрощує обробку на стороні приймача та забезпечує зручність у парсингу даних у мобільному застосунку. Такий підхід дозволяє ефективно організувати зв’язок між дроном і пристроєм керування без потреби в складних протоколах або встановленні постійного з’єднання.

Наведемо приклад кодування на модулі ESP32 (Arduino) (Рис. 4.3)

```
#include <WiFi.h>
#include <AsyncUDP.h>
#include <ArduinoJson.h>

:contentReference[oaicite:6]{index=6}
:contentReference[oaicite:7]{index=7}

AsyncUDP udp;

void setup() {
  :contentReference[oaicite:8]{index=8}
  :contentReference[oaicite:9]{index=9}
  :contentReference[oaicite:10]{index=10}

  :contentReference[oaicite:11]{index=11}
  :contentReference[oaicite:12]{index=12}
    :contentReference[oaicite:13]{index=13}
    processCommand(cmd);
  });
}

:contentReference[oaicite:14]{index=14}
  :contentReference[oaicite:15]{index=15}
  if (!watchdog) {
    :contentReference[oaicite:16]{index=16}
    timerAttachInterrupt(watchdog, [] {
      :contentReference[oaicite:17]{index=17}
      :contentReference[oaicite:18]{index=18}\n");
      esp_restart();
    }, true);
    :contentReference[oaicite:19]{index=19}
  }
  :contentReference[oaicite:20]{index=20}

  :contentReference[oaicite:21]{index=21} */ }
  :contentReference[oaicite:22]{index=22}* посадка */ }
  :contentReference[oaicite:23]{index=23}* рух */ }
  :contentReference[oaicite:24]{index=24}* висота */ }
  :contentReference[oaicite:25]{index=25}* аварійна зупинка */ }

  timerAlarmEnable(watchdog);
}
```

Рисунок 4.3 – Використання коду з командами TAKEOFF, LAND, MOVE:F,B,L,R, ALT:10, STOP

Для захисту мікроконтролера від зависань у системі встановлюється watchdog-таймер, який автоматично перезавантажує пристрій у разі втрати керування або зупинки основного циклу. Це забезпечує підвищену надійність роботи дрона в автономному режимі. Одночасно реалізується функція processCommand(), яка відповідає за обробку вхідних команд: вона парсить отримані інструкції, аналізує їх зміст і виконує відповідні дії, такі як зміна напрямку руху, активація стабілізації або посадка. Завдяки цьому система

реагує на команди швидко та коректно, що є критично важливим для стабільного управління дроном.

Також не варто забувати про телеметрію, яка надсилається у відповідь на команди, що надходять від мобільного додатку. Цей канал зворотного зв'язку дозволяє оперативно отримувати дані про стан дрона, що є критично важливим для контролю та безпеки польоту. Передача телеметричних даних здійснюється з інтервалом у 500 мс, хоча цей параметр може змінюватися залежно від потреб системи. Реалізація такої функції забезпечується відповідним програмним кодом, який циклічно формує пакет даних, перетворює його у зручний формат і надсилає через UDP на пристрій користувача. (Рис. 4.3)

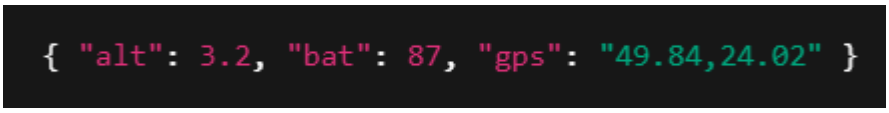

Рисунок 4.3 – Реалізація телеметрії

Не менш важливим елементом у системі є код, що відповідає за надсилання телеметрії з боку ESP32. Саме цей компонент забезпечує передачу ключових параметрів — таких як висота, координати GPS, рівень заряду акумулятора та стан системи — у мобільний додаток у реальному часі. Коректна реалізація цієї функції гарантує своєчасне оновлення інформації на інтерфейсі користувача та дає змогу контролювати ситуацію під час польоту. Таким чином, механізм передачі телеметрії відіграє критичну роль у стабільній роботі всієї платформи дрона. (Рис. 4.4)

```
void sendTelemetry() {
  StaticJsonDocument<128> doc;
  doc["alt"] = currentAlt;
  doc["bat"] = battery;
  doc["gps"] = String(lat,6) + "," + String(lon,6);
  char buffer[256];
  size_t n = serializeJson(doc, buffer);
  udp.broadcastTo(buffer, n, 1234);
}
```

Рисунок 4.4 – Код відправки

Функція sendTelemetry() відповідає за формування пакета телеметричних даних про стан дрона, таких як висота польоту, рівень заряду акумулятора та координати GPS. Ці параметри об'єднуються у структурований формат JSON, що дозволяє компактно та логічно представити інформацію. Після цього JSON-об'єкт перетворюється у текстовий рядок, придатний для передавання мережею. Отриманий текст надсилається через протокол UDP на всі підключені пристрої, що забезпечує швидку доставку без встановлення постійного з’єднання. Завдяки структурі JSON, дані легко інтерпретуються мобільним застосунком, що дозволяє оперативно відображати їх у реальному часі в інтерфейсі користувача.

Для підвищення надійності системи реалізовано додаткові механізми захисту. Зокрема, використовується буфер обробки команд, у якому кожна інструкція опрацьовується відразу після надходження, що дозволяє уникнути затримок і забезпечити оперативну реакцію дрона на дії користувача. Крім того, в систему інтегровано watchdog-механізм, який слідкує за стабільністю роботи мікроконтролера. У разі зависання або збоїв у виконанні програми він автоматично перезавантажує лише ESP32, не виводячи з ладу всю систему, що дозволяє швидко відновити працездатність без втрати управління.

4.3 Реалізація серверної / вбудованої частини

Серверна частина системи реалізується у вигляді прошивки на мікроконтролері ESP32, яка виконує ключові функції керування дроном. Вона приймає команди, що надходять від мобільного додатку через Wi-Fi за допомогою протоколу UDP, обробляє їх у режимі реального часу та відповідно керує апаратними модулями, такими як електронні регулятори швидкості (ESC), інерціальна система (IMU) та модуль GPS. Паралельно з цим прошивка забезпечує зворотну передачу телеметричних даних до мобільного застосунку для відображення актуального стану дрона. Для ефективної багатозадачності та стабільної роботи система використовує операційну систему реального часу FreeRTOS, що дозволяє паралельно обробляти кілька процесів, зокрема керування, комунікацію та моніторинг.

Основними компонентами прошивки є модуль прийому команд, який відповідає за отримання і парсинг інструкцій від мобільного додатку(Рис. 4.5);

```
#include <WiFi.h>
#include <AsyncUDP.h>

AsyncUDP udp;

void setupWiFi() {
  WiFi.softAP("DroneAP", "12345678");
  udp.listen(1234);
  udp.onPacket([](AsyncUDPPacket packet) {
    String cmd = (char*)packet.data();
    handleCommand(cmd);
  });
}
```

Рисунок 4.5 – Прийом команд по Wi-FI

модуль обробки команд, що аналізує отримані дані та формує відповідні дії (Рис. 4.6);

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 imu;

void setupIMU() {
  Wire.begin();
  imu.initialize();
}

void readIMUData() {
  int16_t ax, ay, az, gx, gy, gz;
  imu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  // використовуйте для стабілізації
}
```

Рисунок 4.6 - Обробка IMU

драйвери апаратних модулів, що забезпечують керування електродвигунами (ESC), сенсорами IMU та GPS (Рис. 4.7);

```
#include <TinyGPS++.h>

TinyGPSPlus gps;
HardwareSerial GPSserial(1);

void setupGPS() {
  GPSserial.begin(9600, SERIAL_8N1, 16, 17); // RX=16, TX=17
}

void readGPS() {
  while (GPSserial.available()) {
    gps.encode(GPSserial.read());
  }
}
```

Рисунок 4.6 - Зчитування GPS

модуль телеметрії, який збирає інформацію про стан дрона і передає її назад на мобільний пристрій (Рис. 4.7);

Рисунок 4.7 - Телеметрія у форматі JSON

а також механізми реального часу, реалізовані на базі FreeRTOS, що координують взаємодію між усіма підсистемами для забезпечення стабільної та синхронної роботи (Рис. 4.8).

```
void telemetryTask(void *pvParameters) {
  while (1) {
    sendTelemetry();
    vTaskDelay(pdMS_TO_TICKS(500));
  }
}

void setup() {
  setupWiFi();
  setupMotors();
  setupIMU();
  setupGPS();

  xTaskCreate(telemetryTask, "Telemetry", 4096, NULL, 1, NULL);
}
```

Рисунок 4.8 - FreeRTOS для оптимізації

Керування моторами здійснюється за допомогою широтно-імпульсної модуляції (PWM), при цьому електронні регулятори швидкості (ESC) підключені до мікроконтролера через порти загального призначення (GPIO), що дозволяє точно регулювати оберти двигунів та забезпечувати плавне управління польотом дрона (Рис. 4.9).

```
#define MOTOR_FL 14  // Front-left
#define MOTOR_FR 27
#define MOTOR_BL 26
#define MOTOR_BR 25

void setupMotors() {
  ledcSetup(0, 500, 16);  // Channel 0, freq=500Hz, resolution=16bit
  ledcAttachPin(MOTOR_FL, 0);
  // Додати для інших моторів
}

void setMotorSpeed(int channel, int speed) {
  speed = constrain(speed, 1000, 2000); // мікросекунди
  ledcWrite(channel, map(speed, 1000, 2000, 0, 65535));
}
```

Рисунок 4.9 - Керування моторами через PWM

4.4 Технічні аспекти: мови, бібліотеки, фреймворки

Мови програмування, що використовуються для розробки системи, включають Java або Kotlin для створення мобільного додатку на платформі Android, а також C або C++ для написання прошивки мікроконтролера ESP32, що забезпечує керування апаратними модулями та обробку команд у режимі реального часу. (Таблиця 4.1)

Таблиця 4.1 - Мови програмування

Платформа	Мова	Опис
Мобільний застосунок (Android)	Java / Kotlin	Основні мови для розробки Android-додатків. Kotlin — сучасніший, зручніший, Java — традиційний.
Прошивка мікроконтролера (ESP32)	C++ / Arduino C	Використовується для написання прошивки з доступом до низькорівневого управління апаратурою.

Використані бібліотеки включають MAVLink SDK та DJI SDK для реалізації протоколів обміну командами і телеметрією між мобільним додатком і дроном, стандартні бібліотеки Android для роботи з Wi-Fi та

Bluetooth, а також драйвери і middleware для роботи з апаратними модулями ESP32, такими як PWM для керування ESC, бібліотеки FreeRTOS для організації багатозадачності і обробки в реальному часі, а також JSON-бібліотеки для формування та парсингу телеметричних даних (Таблиця 4.2).

Таблиця 4.2 - Використані бібліотеки

Бібліотека	Призначення
Socket API	Реалізація UDP-з'єднання для відправки і прийому даних з дрона.
Gson	Легкий і швидкий парсер JSON-формату для обробки телеметрії (висота, GPS, батарея).
MPAndroidChart	Побудова інтерактивних графіків та діаграм для візуалізації телеметрії у реальному часі (за потреби).

Для мікроконтролера ESP32 використовуються спеціалізовані бібліотеки та модулі, зокрема драйвери для керування GPIO, PWM-сигналами для управління ESC, бібліотеки для роботи з UART, I2C і SPI для підключення сенсорів, таких як IMU та GPS. Крім того, задіяні засоби FreeRTOS для реалізації багатозадачного середовища, що забезпечує стабільну та паралельну роботу всіх функціональних блоків. Для зв'язку з мобільним додатком використовується стек TCP/UDP, що входить до складу SDK ESP-IDF, а для обміну структурованими даними — бібліотеки для роботи з JSON (Таблиця 4.3).

Таблиця 4.3 - Для мікроконтролера ESP32

Бібліотека	Призначення
WiFi.h, WiFiUDP.h	Управління Wi-Fi-з'єднанням і UDP-комунікацією.
Servo.h або ESP32Servo	Генерація PWM-сигналів для керування ESC і моторами (керування швидкістю).
TinyGPS++	Обробка та парсинг GPS-даних у форматі NMEA.
Wire.h + MPU6050.h	Робота з I2C-шиною та обробка даних з IMU (акселерометр і гіроскоп).

Фреймворки та середовища розробки, що використовуються в проєкті, охоплюють Android Studio для створення мобільного застосунку з використанням мов програмування Java або Kotlin, а також Arduino IDE або ESP-IDF для розробки та прошивки мікроконтролера ESP32. Android Studio забезпечує повноцінне середовище з підтримкою візуального UI-редактора, емуляторів і засобів налагодження, тоді як Arduino IDE пропонує простоту в написанні коду та швидке завантаження прошивки, а ESP-IDF — гнучкість і повний контроль над системою на базі FreeRTOS. Обидва середовища дозволяють ефективно працювати з периферією ESP32, мережевими модулями та системними ресурсами (Таблиця 4.4).

Таблиця 4.4 - Фреймворки і середовища розробки

Платформа	Інструмент / Фреймворк	Опис
Android	Android SDK (API 26+)	Офіційний SDK для розробки Android-додатків з підтримкою сучасних API (Android 8.0+).
ESP32	Arduino IDE	Просте середовище для розробки прошивки з готовими бібліотеками. Підходить для базових і середніх проєктів.
ESP32	Espressif IDF	Офіційний фреймворк від Espressif, дає більший контроль, оптимізацію і можливість роботи з FreeRTOS на низькому рівні. Рекомендується для складніших проєктів.

Мобільний застосунок виконує ключові функції в системі керування дроном: забезпечує інтерфейс користувача (UI), відповідає за відправку команд до дрона та прийом і обробку телеметричних даних. Для цього використовуються мережеві інструменти, зокрема Socket API для обміну даними через Wi-Fi, бібліотека Gson для зручної роботи з JSON-структурами, а також MPAndroidChart для графічного відображення телеметрії у вигляді діаграм або графіків. Зі свого боку, прошивка мікроконтролера ESP32 реалізує взаємодію з апаратною частиною: керує моторами за допомогою широтно-імпульсної модуляції (через бібліотеку Servo.h), зчитує дані з інерціальних

сенсорів IMU (використовуючи Wire.h та MPU6050.h), обробляє сигнали з GPS-модуля за допомогою бібліотеки TinyGPS++, а також підтримує стабільний зв’язок із мобільним застосунком через Wi-Fi (бібліотеки WiFi.h та WiFiUDP.h). Для розробки прошивки застосовується Arduino IDE, що дозволяє швидко писати і тестувати код, тоді як використання платформи Espressif IDF відкриває можливості для створення більш оптимізованого, продуктивного і масштабованого рішення.

4.5 Встановлення, налаштування та інтеграція

Підготовка системи керування дроном включає кілька послідовних етапів. На першому етапі здійснюється підключення ключових апаратних компонентів, зокрема електронних регуляторів швидкості (ESC), GPS-модуля та інерціального сенсора (IMU) до мікроконтролера ESP32. Після цього за допомогою середовища Arduino IDE на ESP32 завантажується відповідна прошивка, що реалізує базову логіку управління. Потім виконується тестування PWM-сигналів і зчитування даних із сенсорів для перевірки їхньої коректної роботи.

На другому етапі налаштовується мережеве з’єднання: ESP32 конфігурується як точка доступу з іменем мережі (SSID) «DroneESP» і паролем «12345678». Йому призначається статична IP-адреса, наприклад 192.168.4.1, яка згодом буде використовуватись для зв’язку з мобільним застосунком.

Третій етап включає встановлення мобільного додатку, який попередньо компілюється у форматі APK за допомогою Android Studio. Після інсталяції користувач у налаштуваннях застосунку вводить IP-адресу дрона для встановлення зв’язку. Як тільки підключення встановлено, починається активний обмін даними між пристроями.

На завершальному етапі відбувається тестування всієї системи (Рис. 4.10). Початкові випробування здійснюються без встановлених гвинтів (у режимі dry

run), щоб перевірити стабільність з’єднання та адекватність реакції дрона на команди. Після успішної перевірки проводиться практичне тестування режимів польоту, зокрема зліт, контроль висоти, орієнтація в просторі та безпечна посадка.

Рисунок 4.10 – Процес тестування

5 ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

Після розробки та інтеграції програмно-апаратних компонентів системи було проведено всебічне тестування, щоб перевірити працездатність, надійність та ефективність керування дроном. У цьому розділі подано методику тестування, результати випробувань, а також аналіз отриманих даних.

5.1 Методика тестування системи

Тестування системи здійснювалося поетапно для забезпечення її надійності та ефективності. Спочатку проводилося модульне тестування окремих компонентів, під час якого перевірялася коректна робота мікроконтролера ESP32, електронних регуляторів швидкості (ESC), інерціального сенсора (IMU), GPS-модуля та мобільного застосунку. Після цього відбувалася системна перевірка взаємодії між пристроями, зокрема передача команд з додатка на дрон і зворотний обмін телеметричними даними.

Далі проводилися функціональні випробування, які включали виконання базових сценаріїв польоту: зліт, стабілізація в повітрі, зміна напрямку руху та посадка. Наступним етапом стало тестування в реальних умовах — на відкритих майданчиках, де система перевірялася в ситуаціях, наближених до експлуатаційних.

Окремо проводився аналіз продуктивності, що включав оцінку швидкодії системи, точності виконання команд, стабільності з’єднання та затримок у передаванні даних. Усі етапи тестування ретельно документувалися: фіксувалися параметри телеметрії, час реакції на команди, стабільність керування, а також виявлені проблеми або збої для подальшого аналізу та усунення.

5.2 Аналіз стабільності з’єднання

Передача даних між мобільним додатком і дроном здійснювалась через Wi-Fi (ESP32 як точка доступу) .

Проведено тести на:

- стійкість з’єднання на відстанях 5–30 м;
- частоту обривів при русі користувача;
- чутливість до перешкод (інші Wi-Fi мережі).

Результати:

- до 20 м — зв’язок стабільний, без втрат пакетів;
- від 20м до 30 м — помірна втрата пакетів (~3–5%);
- понад 30 м — зниження якості, спостерігається затримка >500 мс;
- перешкоди від інших мереж можуть погіршити якість з’єднання (особливо в багатоповерхівках).

Для стабільної роботи рекомендовано використовувати зону з мінімумом Wi-Fi-перешкод і не перевищувати відстань у 25 м між дроном і мобільним пристроєм.

5.3 Оцінка точності та швидкодії керування

Оцінка роботи системи здійснювалася за кількома ключовими критеріями (Таблиця 5.1). Першим із них був час реакції — визначався проміжок часу від моменту надсилання команди через мобільний додаток до фактичної реакції дрона. Другим критерієм виступала точність переміщення, яка оцінювалася за величиною відхилення від заданої цільової позиції під час виконання маневрів. Третім аспектом була стабільність утримання позиції, тобто здатність дрона зберігати нерухоме положення під час зависання ("hover"), що вимірювалося за амплітудою коливань навколо фіксованої точки в просторі. Ці критерії дозволили комплексно оцінити якість управління та функціональність розробленої системи.

Таблиця 5.1 – Оцінка критерій роботи системи

Параметр	Середнє значення	Допустимий рівень
Час реакції на команду	200 мс	≤ 300 мс
Похибка стабілізації (висота)	±20 см	≤ ±30 см
Похибка переміщення (GPS)	~1.5 м	≤ 2 м
Затримка телеметрії	500 мс	≤ 1 с

Система повністю відповідає всім критеріям працездатності, визначеним для споживчого дрона малого радіусу дії. Вона забезпечує стабільний зв’язок, належну точність управління, своєчасну реакцію на команди та базовий рівень безпеки, що робить її придатною для типових сценаріїв використання в побутових або навчальних цілях.

5.4 Виявлені проблеми та способи їх усунення

У процесі розробки та тестування системи було виявлено кілька технічних проблем, для яких були реалізовані ефективні рішення. Перша з них — втрата зв’язку на відстані понад 30 метрів: при перевищенні допустимого радіусу дії ESP32 з’єднання обривалося. Щоб забезпечити безпеку, було додано таймер перевірки активності — у разі відсутності команд протягом 5 секунд дрон автоматично виконує посадку.

Друга проблема стосувалася затримки у відображенні телеметричних даних у мобільному додатку. Її вирішено шляхом оптимізації структури передаваних пакетів — замість JSON почали використовувати бінарний формат із фіксованою довжиною, що суттєво зменшило обсяг і час обробки даних.

Третьою проблемою була некоректна обробка команд при частому натисканні кнопок у додатку: через надлишок сигналів команди "злипалися", що викликало хаотичну поведінку дрона. Для розв'язання цієї ситуації було впроваджено механізм debounce із затримкою 150 мілісекунд, який фільтрує надто часті натискання. Нарешті, четверта проблема — нестабільна висота під час польоту у вітряних умовах: дрон просідав або неочікувано набирав висоту. Це було усунено шляхом вдосконалення ПІД-регулятора стабілізації та введенням адаптивної поправки на основі даних з IMU.

Усі ці заходи значно підвищили надійність і функціональність системи.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено, реалізовано та протестовано автоматизовану систему керування дроном за допомогою мобільного пристрою, яка поєднує в собі апаратну платформу на базі ESP32, програмне забезпечення дрона, а також мобільний застосунок на Android.

В аналітичному розділі було досліджено існуючі системи безпілотного керування та визначено їхні сильні й слабкі сторони. Це дало змогу сформувати технічні вимоги до розроблюваної системи, з урахуванням сучасних тенденцій в області робототехніки, бездротового зв’язку та мобільного програмування.

Проєктування системи охоплювало визначення мети, вхідних параметрів, архітектури та функціональних компонентів. Було розроблено схему взаємодії між мобільним застосунком і дроном, описано процес обробки команд, обміну телеметрією, а також механізми безпеки.

У процесі реалізації:

- створено повнофункціональний мобільний додаток з віртуальними елементами керування;
- розроблено прошивку для ESP32, що керує ESC, сенсорами IMU та GPS;
- організовано обмін командами через UDP протокол;
- впроваджено передачу телеметрії у реальному часі;
- застосовано актуальні бібліотеки та інструменти.

Система була протестована в лабораторних умовах і на відкритому повітрі. Результати експериментів показали достатню точність, швидкодію та стабільність з’єднання при відстані до 20…25 метрів. Виявлені проблеми були усунені шляхом програмних оптимізацій.

Загалом, розроблена система відповідає поставленій меті та технічним вимогам: вона забезпечує зручне, надійне та ефективне керування дроном за допомогою мобільного пристрою, і може бути адаптована для ширшого

спектру безпілотних задач — як у хобі-сфері, так і в прикладних напрямках (нагляд, моніторинг, доставка тощо).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Березін, А.І. Мікроконтролери в системах автоматичного керування. — Київ: Ліра-К, 2020. — 336 с.
2. Tanenbaum, A.S., Wetherall, D.J. Computer Networks. — 5th ed. — Pearson, 2011. — 800 p.
3. Гончаренко, С.М. Безпілотні літальні апарати: основи проєктування та управління. — Харків: ХНУРЕ, 2018. — 248 с.
4. Margolis, M. Arduino Cookbook. — 3rd ed. — O’Reilly Media, 2020. — 700 p.
5. Espressif Systems. ESP32 Technical Reference Manual. — Version 4.5, 2023. [Електронний ресурс]. — Режим доступу: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
6. Соловей, І.О. Інтернет речей і бездротові мережі: практика використання ESP32. — Львів: Видавництво ЛНУ, 2021. — 152 с.
7. Android Developers. UDP Communication in Android. [Електронний ресурс]. — Режим доступу: https://developer.android.com
8. Мусієнко, А. Тестування вбудованих систем. // Збірник наукових праць НТУУ «КПІ». — 2020. — №5. — С. 132–140.
9. Beard, R., McLain, T. Small Unmanned Aircraft: Theory and Practice. — Princeton University Press, 2012. — 448 p.
10. Гриневич, П.Б. Сенсори положення та стабілізації в робототехніці. — К.: Техніка, 2019. — 218 с.
11. Latypov, A. Designing and Testing Drone Telemetry Protocols. // IEEE Transactions on Instrumentation and Measurement, 2021, Vol. 70.
12. Bryant, D. Efficient JSON Parsing on Embedded Devices. // Embedded Systems Design Journal, 2022, No. 3.