Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

перший (бакалаврський)
(рівень вищої освіти)

Розроблення літаючого макету дрону на базі ARM мікроконтролера для збору та обробки даних у реальному часі

(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ 21-2
Артем КОВАЛЬ
(власне ім'я, прізвище)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно - інтегровані технології
(повна назва освітньої програми)

Керівник проф. Владислав ЄВСЄЄВ
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри КІТАР _____ Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2025 р.

Я , Коваль Артем Русланович, як здобувач(ка) вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(ла) і не одержував(ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував(ла) штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на віповідне джерело.

"04" червень 2025 р. Артем КОВАЛЬ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет АКТ
Кафедра КІТАР
Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми Освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри КІТАР _____
(підпис)
« ___»_____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ковалю Артему Руслановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка ліиаючого макету дрону на базі ARM мікроконтролера для збору та обробки даних у реальному часі

Затверджена наказом по університету від 19.05.2025 р. № 390СТ
2. Термін подання здобувачем роботи до екзаменаційної комісії 16.06.2025 р.
3. Вихідні дані до роботи
3.1 5inc дрон , на ARM мікроконтроллері
3.2 IoT інтеграція
3.3 Отримання даних з давачів у режимі реального часу
3.4 Автоматичне розпізнавання давачів підключених по i2c шині
3.5 Можливість сконфігурувати пін мікроконтролера для роботи з нестандартним давачем
3.6 Можливість виконати калібрування дрону без підключення до комп'ютеру
4. Перелік питань, що потрібно опрацювати в роботі
4.1 Вступ;
4.2 Аналіз сучасних дронів для збору та обробки інформації та їх архітектури;
4.3 Розробка апаратної платформи дрона;
4.4 Розробка програмного забезпечення;
4.5 Проведення експерементального тестування;
4.6 Висновки;

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

Демонстраційні графічні матеріали в Power Point у форматі *pptx формату А4 14 сторинок

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	Термін виконання етапів роботи	Приміт-ка
1	Аналіз сучасних дронів для збору та обробки інформації	28.04.2025	Виконано
2	Розробка апаратної платформи дрону	3.5.2025	Виконано
3	Розробка прошивок для мікроконтролерів	8.5.2025	Виконано
4	Розробка мобільного інтерфейсу користувача	15.5.2025	Виконано
5	Тестування	20.5.2025	Виконано
6	Оформлення пояснювальної записки	25.05.2025	Виконано
7	Подання роботи на нормоконтроль	4.06.2025	Виконано
8	Подання роботи на рецензію	5.06.2025	Виконано
9	Подання роботи на підпис зав. кафедри	9.06.2025	Виконано
11	Подання кваліфікаційної роботи в ЕК	14.06.2025	Виконано

Дата видачі завдання 28.04. 2025 р.

Здобувач _____ Артем Коваль
(підпис) (власне ім'я, прізвище)

Керівник роботи _____ проф. Владислав Євсєєв
(підпис) (посада , власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 85 с., 1 табл., 55 рис., 4 дод.,32 джерел.

ДРОН, БПЛА, ARM МІКРОКОНТРОЛЕР, STM32, ESP32, ЗБІР ДАНИХ, ОБРОБКА ДАНИХ В РЕАЛЬНОМУ ЧАСІ, BETAFLIGHT, СИСТЕМА КЕРУВАННЯ .

Мета роботи – удосконалення процесу збору та автоматизованої обробки даних у реальному часі шляхом розробки інтегрованої системи моніторингу на базі дрона з інтелектуальним модулем.

Об’єкт розробки – процес автоматизованого керування керування дроном на базі мікроконтролера ARM.

Предмет розробки – апаратне, алголритмічне, програмне забезпечення системи керування дроном та розробка макету дрона на базі ARM мікроконтролера.

В даній роботі представлено розробку літаючого макету дрону на базі ARM мікроконтролера. Було розроблено його структурну схему, обрано ключові апаратні компоненти, включаючи основний мікроконтролер та модуль збору даних, та зібрано експериментальний макет. Проведено аналіз стійкості системи керування та розрахунок прогнозованого часу польоту. Також розроблено та налаштовано програмне забезпечення для польотного контролера і модуля збору даних на базі ESP32.

Дана робота відповідає Цілям сталого розвитку ООН 9 та 12, оскільки сприяє технологічним інноваціям, цифровізації та оптимізації ресурсів завдяки моніторингу квадрокоптером.

ABSTRACTS

The explanatory note contains 85p, 1 tables, 55 drawings, 4 pp., 32 sources.

DRONE, UAV, ARM MICROCONTROLLER, STM32, ESP32, DATA COLLECTION, REAL-TIME DATA PROCESSING, BETAFLIGHT, CONTROL SYSTEM.

Purpose of the work – improving the process of collecting and automated data processing in real time by developing an integrated monitoring system based on a drone with an intelligent module.

Object of development – the process of automated drone control based on an ARM microcontroller.

Subject of development – the hardware, algorithmic, and software components of the drone control system, as well as the development of a drone prototype based on an ARM microcontroller.

This work presents the development of a flying drone prototype based on an ARM microcontroller. A structural diagram was designed, key hardware components were selected, including the main microcontroller and data collection module, and an experimental prototype was assembled. The stability of the control system was analyzed, and the estimated flight time was calculated. Additionally, software for the flight controller and data collection module based on the ESP32 was developed and configured.

This work is in line with UN Sustainable Development Goals 9 and 12, as it promotes technological innovation, digitalization, and resource optimization through quadcopter monitoring.

ЗМІСТ

Перелік скорочень 9
Вступ 10
1 Аналіз сучасних дронів для збору та обробки інформації та їх архітектури 13
1.1 Аналіз сучасних безпілотних літальних апаратів як платформ для збору даних 13
1.2 Аналіз мікроконтролерної техніки для вбудованих систем реального часу 17
1.3 Аналіз сенсорних систем для збору даних з дронів 20
1.4 Аналіз методів та технологій обробки даних в режимі реального часу 23
1.5 Аналіз систем керування польотом та джерел живлення 26
1.6 Аналіз систем та протоколів бездротової передачі даних 29
2 Розробка апаратної платформи дрона 31
2.1 Розробка структурних схеми 31
2.2 Вибір компонентів для створення макету дрона для збору та обробки даних 34
2.3. Розробка схеми підключення та збірка макету 41
2.4. Аналіз стійкості системи керування 46
2.6. Розрахунок та аналіз прогнозованого часу польоту 49
3 Розробка програмного забезпечення 50
3.1 Конфігурування прошивки Betaflight для створеного польотного контролера 50
3.2 Розробка прошивки для модуля ESP32 для збору даних та передачі інформації 53

3.3 Розробка OSD інтерфейсу 57

3.4 Розробка Android застосунку 60

4 Проведення експериментального тестування 62

4.1 Тестування розробленого дрона та модуля для обробки даних 62

4.2 Польотні тести 66

4.2 Аналіз лабораторних умов 68

Висновки 70

Перелік джерел посилання 71

Додаток А лістинг функції для розпізнавання підключеного до i2c шини давача 75

Додаток Б лістинг коду для виведення OSD даних 77

Додаток В лістинг коду взаємодії мобільного застосунку та ESP32 80

Додаток Г демонстраційний матеріал у вигляді презентації 85

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач (англ. Analog-to-Digital Converter, ADC)

БПЛА – безпілотний літальний апарат (англ. Unmanned Aerial Vehicle, UAV)

МК – мікроконтролер (англ. Microcontroller, MCU)

НДР – науково-дослідна робота

ПК - польотний контролер

ARM – Advanced RISC Machines (архітектура мікропроцесорів)

CRSF – Crossfire Serial Protocol (протокол приймача Crossfire)

ESC – Electronic Speed Controller (електронний регулятор швидкості)

FC – Flight Controller (польотний контролер, ПЛК)

I2C – Inter-Integrated Circuit (міжінтегральна шина)

IMU – Inertial Measurement Unit (блок інерційних вимірювань, інерційний вимірювальний модуль)

MSP – MultiWii Serial Protocol (протокол послідовного зв'язку MultiWii)

PDB – Power Distribution Board (плата розподілу живлення)

RX – Receiver (приймач)

TX – Transmitter (передавач)

UART – Universal Asynchronous Receiver/Transmitter (універсальний асинхронний приймач-передавач)

OSD – On-Screen Display

ВСТУП

У сучасному світі безпілотні літальні апарати (БПЛА), або дрони, набувають все більшого значення, розширюючи можливості людства в багатьох сферах – від моніторингу довкілля до логістики та безпеки. Їх ефективне застосування вимагає не лише здатності до автономного польоту, але й можливості оперативно збирати та аналізувати інформацію безпосередньо на борту. Це висуває високі вимоги до бортових систем, зокрема щодо обробки даних в обмеженому реальному часі.

Здатність дрона самостійно орієнтуватися, збирати дані за допомогою різноманітних сенсорів, обробляти цю інформацію "на льоту" для прийняття рішень та ефективно використовувати обмежені енергетичні ресурси стає ключовою характеристикою сучасних інтелектуальних дронів.

Дрони, призначені для збору та обробки даних, є складними апаратно-програмними комплексами. Вони об'єднують механічну платформу, навігаційні системи, різноманітні сенсори та потужний обчислювальний блок, часто на базі сучасних ARM мікроконтролерів, що взаємодіють через складне програмне забезпечення. Ці компоненти разом формують систему, здатну до певної міри автономно аналізувати оточення та виконувати поставлені завдання.

Створення експериментального макета дрона на базі ARM мікроконтролера відкриває значні можливості для практичного дослідження реалізації систем збору та обробки даних в режимі реального часу на борту дрона. Аналізуючи продуктивність такої системи, швидкість обробки, точність отримуваних результатів та енергоспоживання, можна виявити її сильні та слабкі сторони в контексті конкретних завдань та визначити шляхи подальшої оптимізації програмного та апаратного забезпечення.

Таким чином, метою даної роботи є удосконалення процесу збору та автоматизованої обробки даних у реальному часі шляхом розробки інтегрованої системи моніторингу на базі дрона з інтелектуальним модулем

Об’єкт розробки – процес автоматизованого керування керування дрону на базі мікроконтролера APM.

Предмет розробки – апаратне, алгоритмічне, програмне забезпечення системи керування дроном та розробка макету дрона на базі ARM мікроконтролера.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз існуючих систем, рішень та сценаріїв використань сучасних дронів;

– провести аналіз компонентів сучасних дронів;

– розробити структурні схеми апаратної платформи дрона;

– здійснити вибір та обґрунтування ключових компонентів для апаратної платформи, включаючи польотний контролер, модуль збору даних, сенсори та інші необхідні елементи;

– дослідити та проаналізувати існуючі дрони та засоби для збору та обробки інформації ,що застосовуються разом з ними;

– скласти експериментальний макет апаратної платформи дрона на основі розроблених схем та обраних компонентів;

– провести аналіз стійкості розробленої системи керування польотом для лінеаризованої моделі дрон;

– сконфігурувати прошивку Betaflight, відповідно до апаратних особливостей створеної платформи;

– розробити програмне забезпечення для модуля ESP32, що забезпечує збір даних з підключених сенсорів, їх попередню обробку та передачу інформації по бездротовому каналу зв'язку;

– розробити OSD інтерфейс для виводу даних з давачів у реальному часі;

– провести тестування системи;

Кваліфікаційна робота виконана згідно ДСТУ 3008 – 15 [1], та керуючись навчальним посібником з дипломного проекту [2] та методичними вказівками [3].

1 АНАЛІЗ СУЧАСНИХ ДРОНІВ ДЛЯ ЗБОРУ ТА ОБРОБКИ ІНФОРМАЦІЇ ТА ЇХ АРХІТЕКТУРИ

1.1 Аналіз сучасних безпілотних літальних апаратів як платформ для збору даних

Безпілотні літальні апарати (БПЛА), широко відомі як дрони, здійснили революцію у багатьох галузях завдяки своїй здатності швидко, ефективно та безпечно збирати різноманітні дані з повітря. Від моніторингу сільськогосподарських угідь до інспекції критичної інфраструктури та картографування місцевості – дрони стали незамінними інструментами, що надають унікальні можливості для отримання просторової та часової інформації. Однак ефективність дронів як платформи для збору даних значною мірою залежить від правильного вибору типу апарату та його характеристик відповідно до конкретного завдання. Метою даного підрозділу є аналіз сучасних типів дронів, їх ключових характеристик та сфер застосування саме в контексті використання як платформ для збору даних.

На рисунку 1.1 зображено приклад сучасного дрона.

Рисунок 1.1 – Приклад сучасного дрону DJI Mavic 3 [4]

Популярними прикладами таких систем є серії DJI Mavic, Phantom, Matrice, а також дрони від Autel та Parrot. Однак, через відносно низьку аеродинамічну ефективність, мультироторні системи зазвичай мають обмежений час польоту, типово 20-40 хвилин, та меншу крейсерську швидкість порівняно з літаковими типами. Це обмежує їх використання для обстеження великих територій за один виліт. Збільшення кількості роторів (гекса-, окто-) підвищує вантажопідйомність та відмовостійкість системи, оскільки дрон може зберегти керованість при відмові одного двигуна, що є важливим фактором при використанні коштовного сенсорного обладнання.

Серед найбільш перспективних напрямів у розвитку платформ для збору даних варто окремо виділити квадрокоптери, побудовані на базі мікроконтролерів із ядром ARM. Цей підхід відкриває нові горизонти в контексті автономності, продуктивності та інтеграції складних обчислювальних алгоритмів безпосередньо на борту літального апарата. Застосування ARM-архітектури у керуючих контролерах дає змогу реалізовувати в режимі реального часу функції фільтрації сенсорних даних, виявлення аномалій, адаптивного керування та навіть базових методів машинного навчання. У ряді досліджень було продемонстровано, що використання 32-бітових ARM Cortex-M контролерів у поєднанні з операційною системою реального часу (наприклад, ChibiOS у PX4) дозволяє істотно скоротити час реакції системи стабілізації та покращити точність динамічного позиціювання апарата.

Іншим важливим класом є БПЛА літакового типу (Fixed-Wing). Ці дрони за конструкцією нагадують традиційні літаки і використовують крила для створення підйомної сили. Це забезпечує значно вищу аеродинамічну ефективність порівняно з мультироторними системами.

Вибір оптимальної платформи БПЛА для конкретного завдання збору даних вимагає ретельного аналізу низки ключових характеристик. Перш за все, це стабільність польоту – здатність дрона утримувати своє положення та орієнтацію в просторі, особливо під впливом вітру. Висока стабільність є

критично важливою для отримання якісних, не розмитих зображень та точних даних з сенсорів. Наступним важливим критерієм є вантажопідйомність, тобто максимальна маса корисного навантаження (сенсори, обчислювальні модулі), яку дрон може нести. Цей параметр визначає, яке саме обладнання для збору даних може бути встановлене на борт. На рисунку 1.2 зображено дрон для автоматичної доставки їжі у Китайської компанії Meituan у спеціальні пункти замовлення.

Рисунок 1.2 – Дрон компанії Meituan для доставки замовлень [5]

Тривалість та дальність польоту безпосередньо впливають на продуктивність місії – площу, яку можна обстежити, або тривалість безперервного моніторингу. Ці параметри залежать від типу платформи, ємності акумуляторів та маси корисного навантаження. Не менш важливою є стійкість до умов навколишнього середовища: здатність дрона працювати в різних погодних умовах (температура, вітер, опади), що визначається його конструкцією та IP рейтингом. Також слід враховувати надійність та відмовостійкість системи, особливо для критичних місій, де може знадобитися

резервування ключових компонентів. На практичний вибір впливають також простота експлуатації та обслуговування, включаючи легкість підготовки до польоту, керування, обробки даних та доступність ремонту. Нарешті, вартість платформи та її експлуатації є невід'ємним фактором при плануванні будь-якого проекту.

Аналіз сучасних дронів показує їх широке застосування як ефективних платформ для збору даних у багатьох галузях. У точному землеробстві дрони з мультиспектральними та тепловізійними камерами дозволяють оцінювати стан здоров'я посівів, виявляти проблемні ділянки та створювати карти для диференційованого внесення добрив. На рис. 1.3 представлена карта NDVI зроблена з дрона.

Рисунок 1.3 – NDVI карта поля зроблена з дрона [6]

У картографії та геодезії використання фотограмметрії та LiDAR дозволяє створювати високоточні ортофотоплани, цифрові моделі місцевості та 3D-моделі об'єктів. На рисунку 1.4 зображена хмара точок отримана з лідару встановленого на дрон.

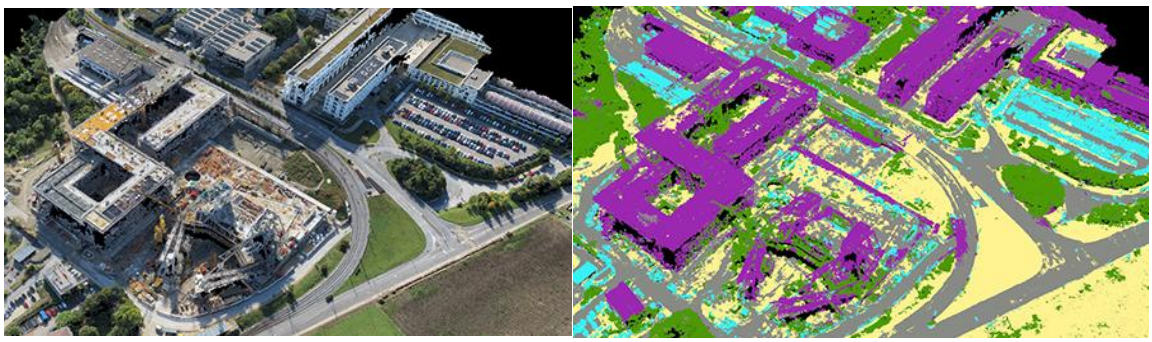

Рисунок 1.4 – Отримана з дрона хмара точок [7]

У моніторингу навколишнього середовища дрони використовуються для оцінки якості повітря та води, відстеження тварин, контролю за вирубкою лісів та оцінки наслідків стихійних лих. Також вони знаходять застосування у сфері безпеки та спостереження для охорони периметрів, моніторингу масових заходів та в пошуково-рятувальних операціях.

1.2 Аналіз мікроконтролерної техніки для вбудованих систем реального часу

Центральним елементом будь-якого сучасного дрона, відповідальним за керування польотом, збір даних з сенсорів та їхню первинну обробку, є мікроконтролер (МК). Вибір відповідного мікроконтролера є одним із ключових практичних завдань при проектуванні бортової електроніки, оскільки від його характеристик залежить продуктивність, функціональність, енергоефективність та вартість усієї системи.

Для дронів, що функціонують в режимі реального часу, особливо тих, що призначені для оперативного збору та обробки даних, вибір мікроконтролера є визначальним етапом проектування. Саме характеристики МК значною мірою диктують можливості, продуктивність та енергоефективність всієї бортової системи.

На сьогоднішній день в сегменті мікроконтролерів для вбудованих систем домінує архітектура ARM. Її популярність обумовлена вдалим

поєднанням продуктивності, енергоефективності та широкої підтримки з боку виробників інструментів розробки.

В контексті розробки дронів особливий інтерес становлять ядра ARM Cortex-M від STmicrotltctronics, у платах STM32. Існує надзвичайно широкий вибір МК – від простих Cortex-M0 до високопродуктивних Cortex-M7. На рисунку 1.5 зображено блок-схему архітектури Cortex-M**.**

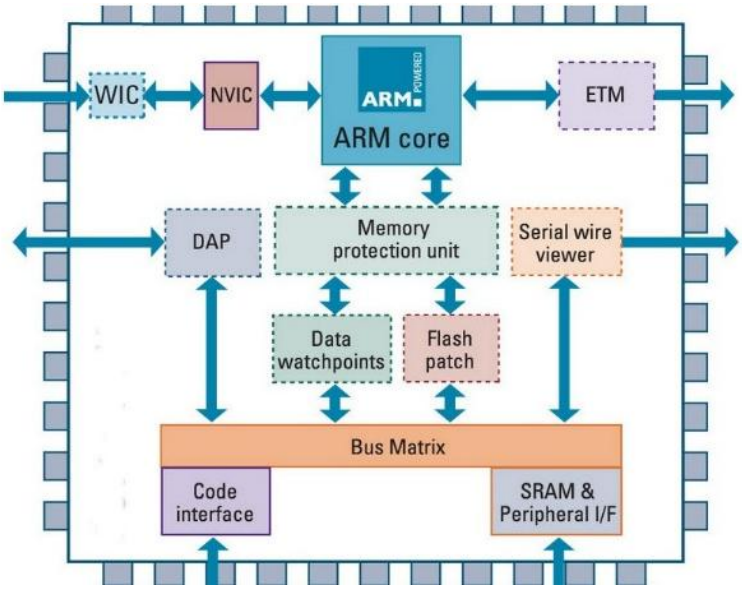

Рисунок 1.5 – Блок схема архітектури ARM Cortex-M [8]

Важливою перевагою STM32 є багатий набір інтегрованої периферії (різноманітні таймери, комунікаційні інтерфейси SPI, I2C, UART, CAN, АЦП, ЦАП), а також потужна та зручна екосистема розробки. Інструменти як STM32CubeMX та бібліотеки HAL/LL значно спрощують та прискорюють процес конфігурування МК та написання коду, що є вагомим практичним плюсом [11]. На рисунку 1.6 зображено приклад типового МК STM32F103C8T6 на архітектурі M1.

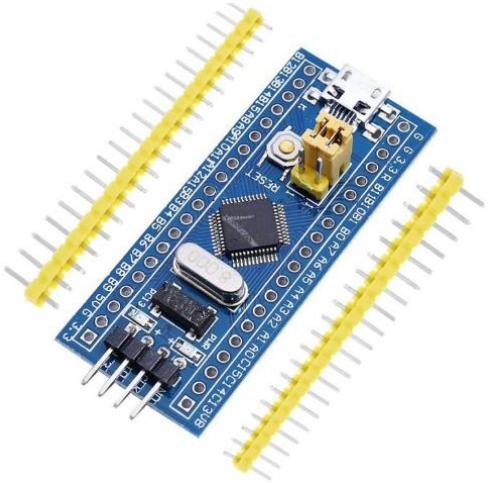

Рисунок 1.6 – Плата розробника STM32F103C8T6 [9]

Іншим популярним вибором, особливо в DIY-сегменті та для завдань, що потребують інтегрованих бездротових інтерфейсів, є мікроконтролери ESP32 (рис. 1.7) від Espressif Systems**.** Хоча класичні ESP32 базуються на ядрах Xtensa, вони мають вбудовані модулі Wi-Fi та Bluetooth, що робить їх ідеальними для реалізації каналів зв'язку, телеметрії або навіть передачі відео низької роздільної здатності. Для завдань, що вимагають гарантованого реального часу, їх використання може потребувати ретельнішого планування програмної архітектури. Новіші чіпи від Espressif вже використовують ядра RISC-V або ARM. Завдяки низькій вартості та потужним комунікаційним можливостям, ESP32 часто використовується як допоміжний процесор або основний контролер у менш вимогливих до жорсткого реального часу системах дронів.

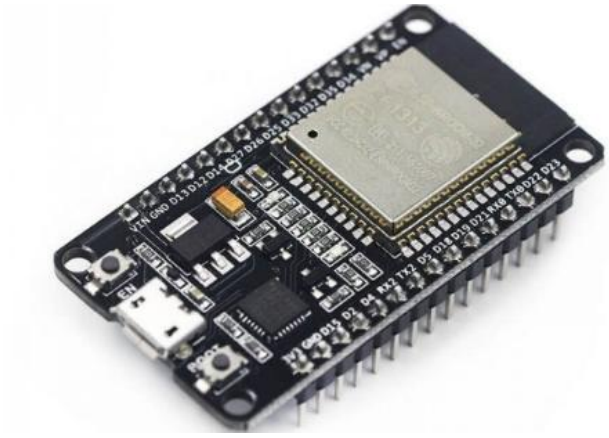

Рисунок 1.7 – Плата розробника ESP32 -WROOM [10]

Також існує Raspberry Pi Pico на базі мікроконтролера RP2040. Цей бюджетний МК має два ядра Cortex-M0+ та унікальну периферію PIO (Programmable I/O), яка дозволяє гнучко реалізовувати нестандартні апаратні інтерфейси для специфічних сенсорів або протоколів. Хоча його обчислювальна потужність поступається STM32 F4/F7, він може бути чудовим вибором для допоміжних завдань або нескладних систем керування.

1.3 Аналіз сенсорних систем для збору даних з дронів

Сенсорні системи є необхідні для навігації та збору даних для безпілотного літального апарату. Вони надають йому критично важливу інформацію про власний стан – положення, орієнтацію, швидкість – а також про навколишнє середовище.

Для дронів, основне призначення яких полягає у зборі даних, сенсори є ключовим компонентом. Саме вони визначають тип та якість інформації, яку можливо отримати під час виконання польотного завдання.

Правильний вибір, подальша інтеграція та ретельне калібрування сенсорів є важливими практичними етапами при проектуванні та створенні макета дрона. Даний підрозділ звіту присвячений аналізу основних типів сенсорів, їх ключових характеристик та аспектів застосування на борту дрона.

Сенсори, що використовуються на сучасних дронах, можна умовно поділити на кілька категорій за їх основним функціональним призначенням.

Перша категорія – це навігаційні сенсори, які формують основу системи керування польотом. До них належать інерційні вимірювальні модулі (IMU). Зазвичай IMU містять акселерометри для вимірювання лінійних прискорень та гіроскопи для визначення кутових швидкостей. Ці дані є критичними для стабілізації польоту дрона.

Також до навігаційних сенсорів відносяться приймачі глобальних навігаційних супутникових систем (GNSS/GPS). Вони надають інформацію про поточне географічне положення апарату.

Барометричні датчики (барометри) дозволяють визначати барометричну висоту польоту, що важливо для її утримання та виконання деяких місій.

Комплексне використання даних з усіх цих сенсорів (IMU, GNSS, барометр) дозволяє системі керування польотом точно та надійно оцінювати стан апарату в просторі. Приклад такої системи на рисунку 1.8.

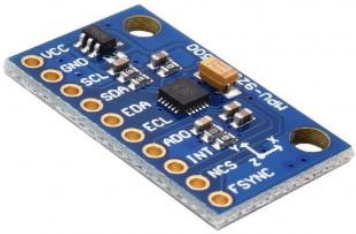

Рисунок 1.8 – Приклад типового IMU MPU6500 [11]

Друга важлива категорія – сенсори сприйняття навколишнього середовища. Ці датчики допомагають дрону орієнтуватися відносно перешкод та інших об'єктів у польоті.

Сюди відносяться далекоміри, такі як лідари. Лідари використовують лазерний промінь для створення детальної тривимірної карти оточення (хмари точок) та вимірюють відстані до об'єктів з високою точністю. Це є ключовим для реалізації автономної навігації та ефективного уникнення зіткнень.

Ультразвукові датчики (сонари) є простішим та дешевшим рішенням. Вони ефективні для виявлення перешкод на відносно близьких відстанях, наприклад, під час посадки або польоту в приміщеннях.

Звичайні камери видимого спектру також можуть використовуватися для навігації (Visual Odometry) та виявлення перешкод, але це потребує складніших алгоритмів комп'ютерного зору.

Третя, і найбільш варіативна, категорія – це сенсори для збору цільових даних місії. Їх вибір повністю залежить від конкретного завдання, яке має виконувати дрон у рамках виробничої практики чи проекту.

Це можуть бути камери високої роздільної здатності видимого спектру (RGB), мультиспектральні та гіперспектральні камери, тепловізійні камери (рис. 1.9).

Рисунок 1.9 – Приклад роботи тепловізійної камери для FPV дрона [12]

Також на борт можуть встановлюватися різноманітні газоаналізатори (рис. 1.10) для моніторингу якості повітря, датчики температури, вологості та інші спеціалізовані вимірювальні прилади, відповідно до мети збору даних.

Рисунок 1.10 – Газоаналізатор для дрона [13]

Важливо пам'ятати, що вантажопідйомність дрона та доступне бортове живлення накладають суттєві практичні обмеження на кількість та тип сенсорів, що можуть бути встановлені та ефективно використані одночасно.

Необхідно також переконатися у сумісності інтерфейсів сенсора (найчастіше це I2C, SPI, UART для простих датчиків; USB, Ethernet, MIPI CSI для складних камер та лідарів) з обраним мікроконтролером чи обчислювальним модулем. Звісно, вартість та надійність сенсора в умовах польової експлуатації також є важливими практичними факторами.

1.4 Аналіз методів та технологій обробки даних в режимі реального часу

Обробка даних в режимі реального часу є невід'ємною складовою функціонування сучасних автономних дронів, особливо тих, що призначені для збору та аналізу оперативної інформації. Це поняття означає обробку даних з сенсорів достатньо швидко, щоб результати цієї обробки могли бути використані для своєчасного прийняття рішень або керування системою.

Для безпілотних літальних апаратів це може включати як негайну реакцію на дані з навігаційних сенсорів для стабілізації польоту, так і оперативний аналіз даних з цільових сенсорів (камер, лідарів тощо) для виконання місії. Важливість обробки в реальному часі зумовлена потребою в автономності, швидкій реакції на зміни в оточенні та часто обмеженою пропускною здатністю каналів зв'язку для передачі великих обсягів сирих даних. Цей підрозділ аналізує ключові підходи та технології, що застосовуються для такої обробки на борту дрону.

Принциповим питанням при проектуванні системи є вибір місця проведення основної обробки даних. Існують два основні підходи: бортова (on-board або edge computing) та позабортова (off-board) обробка.

Бортова обробка передбачає виконання всіх або значної частини обчислень безпосередньо на мікроконтролері чи обчислювальному модулі, встановленому на дроні. Головними перевагами цього підходу є мінімальна затримка між отриманням даних та результатом обробки, що критично для задач керування та швидкої реакції (наприклад, уникнення перешкод).

Також бортова обробка значно зменшує вимоги до пропускної здатності каналу зв'язку, оскільки передавати потрібно лише результати або стислі дані, а не весь сирий потік. Це підвищує рівень автономності дрона, дозволяючи йому функціонувати навіть за умов тимчасової втрати зв'язку з наземною станцією, та може сприяти підвищенню безпеки конфіденційних даних.

Позабортова обробка**,** навпаки, передбачає передачу сирих або мінімально оброблених даних на потужну наземну станцію або у хмарне сховище для подальшого аналізу. Перевагою є доступ до значно більших обчислювальних ресурсів, що дозволяє застосовувати складніші алгоритми, використовувати великі моделі машинного навчання та полегшує процес розробки та відлагодження програмного забезпечення.

Недоліки цього підходу пов'язані з високими вимогами до каналу зв'язку – він повинен бути стабільним та мати достатню пропускну здатність. Затримки передачі даних можуть бути неприйнятними для задач, що вимагають негайної реакції. Крім того, постійна передача даних споживає значну кількість енергії та робить систему залежною від наявності зв'язку.

Для реалізації обробки даних в реальному часі на борту дрона використовуються різноманітні методи та алгоритми. Фільтрація даних є необхідною для видалення шумів та похибок, властивих показанням більшості фізичних сенсорів (IMU, GPS, аналогові датчики). Для цього застосовуються цифрові фільтри, такі як фільтри ковзного середнього, медіанні фільтри, а для навігаційних задач – складніші алгоритми на кшталт розширеного фільтра Колмана (EKF), який одночасно фільтрує дані та об'єднує показники з різних сенсорів (sensor fusion).

Виділення ознак (feature extraction) та детектування подій/об'єктів дозволяють зменшити обсяг даних, залишаючи лише найважливішу інформацію. Наприклад, в обробці зображень це може бути виявлення кутів, країв, контурів об'єктів або розпізнавання певних патернів. Для інших типів сенсорів це може бути детектування перевищення порогових значень,

виявлення певних сигнатур у сигналах тощо. Приклад розпізнавання дроном обьєктів зображено на рисунку 1.11.

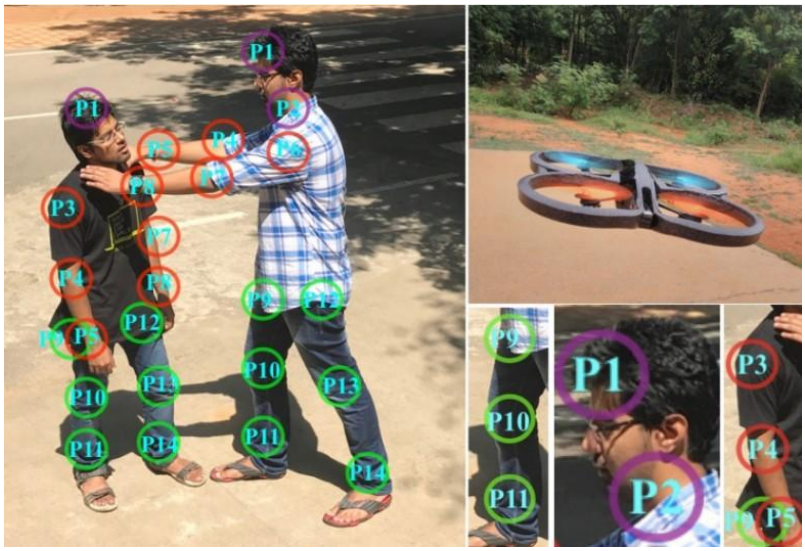

Рисунок 1. 11 – Розпізнавання дроном людей [14]

Компресія даних використовується для зменшення обсягу інформації перед її збереженням на борту або передачею через канал зв'язку. Можуть застосовуватися як методи стиснення без втрат (для критично важливих даних), так і методи з контрольованими втратами (наприклад, для відео чи зображень), які забезпечують вищий ступінь стиснення.

У задачах, пов'язаних з комп'ютерним зором, можуть застосовуватись алгоритми розпізнавання та відстеження об'єктів. Через високу обчислювальну складність на борту зазвичай використовуються полегшені та оптимізовані моделі, наприклад, на основі згорткових нейронних мереж (CNN), адаптовані для роботи на мікроконтролерах чи вбудованих системах.

Ефективне керування паралельним виконанням різних задач (збір даних з сенсорів, їх обробка, керування двигунами, комунікація) на одному мікроконтролері часто потребує використання операційної системи реального часу (RTOS). Популярні RTOS, такі як FreeRTOS, Zephyr, ChibiOS, надають механізми для планування задач за пріоритетами, їх синхронізації (через семафори, м'ютекси, черги) та керування ресурсами.

Для прискорення розробки та підвищення продуктивності обробки даних існують спеціалізовані програмні бібліотеки та інструменти. Фреймворки типу TensorFlow Lite for Microcontrollers або uTVM дозволяють запускати оптимізовані моделі машинного навчання на мікроконтролерах. Для задач комп'ютерного зору можуть використовуватись полегшені версії бібліотек типу OpenCV.

Важливим аспектом практичної реалізації є оптимізація продуктивності програмного коду та використання апаратних можливостей мікроконтролера. Це включає вибір ефективних алгоритмів, написання оптимізованого коду на C/C++, використання можливостей компілятора, активне застосування апаратних прискорювачів (якщо є FPU, DSP-інструкції), а також використання контролера прямого доступу до пам'яті (DMA) для передачі даних між периферією та пам'яттю без завантаження центрального процесора.

1.5 Аналіз систем керування польотом та джерел живлення

Стабільний та керований політ, а також достатній час роботи від акумулятора є фундаментальними вимогами для будь-якого безпілотного літального апарату. Особливо це стосується дронів, що використовуються для збору даних, адже якість інформації та можливість виконання місії безпосередньо залежать від надійності системи керування польотом та ефективності системи енергоживлення. Цей підрозділ коротко аналізує ключові компоненти та аспекти цих двох критично важливих підсистем.

Система керування польотом є, логічним центром дрона, що відповідає за його стабілізацію в повітрі та навігацію. Сучасні польотні контролери являють собою плату, що містить МК, IMU з акселерометрами та гіроскопами, а також барометр для визначення висоти. ПЗ реалізує складні алгоритми, включаючи ПІД-регулятори для стабілізації кутового положення,фільтри,наприклад фільтр Калмана для точної оцінки стану дрона, а також логіку для виконання

різних польотних режимів – від ручного керування до повністю автономної навігації за заданими точками.

Типовий контролер польоту зображено на рисунку 1.12.

Рисунок 1.12 – Типовий польотний контролер [15]

Система енергоживлення забезпечує дрон необхідною енергією для роботи двигунів, польотного контролера, сенсорів та корисного навантаження. Основним джерелом енергії для більшості сучасних дронів є літієві акумуляторні батареї, найчастіше літій-полімерні (LiPo) або літій-іонні (Li-Ion). LiPo батареї (рис. 1.13) пропонують високу густину енергії та здатність віддавати великі струми (високий C-рейтинг), що важливо для динамічних польотів, але вони вимагають обережного поводження.

Рисунок 1.13 – Li-Po батарея для квадрокоптера [16]

Li-Ion батареї (рис. 1.14) зазвичай мають трохи нижчу густину енергії та струмовіддачу, але можуть запропонувати довший термін служби та вищий рівень безпеки. Ключовими параметрами батареї є напруга (визначається кількістю послідовно з'єднаних комірок, позначається як "S", наприклад, 3S, 4S, 6S), ємність (вимірюється в міліампер-годинах, mAh, або ампер-годинах), та максимальний струм розряду (C-rating).

Рисунок 1.14 – Li-ion батарея для квадрокоптера [17]

Окрім самої батареї, система живлення включає електронні регулятори швидкості (ESC), які отримують команди від польотного контролера (через протоколи PWM, OneShot, DShot) і керують швидкістю обертання безколекторних двигунів. Також часто використовується плата розподілу живлення (PDB) для зручного підключення ESC та інших компонентів до батареї, яка може містити стабілізатори напруги (BEC) для живлення електроніки. Для багатокоміркових Li-Ion збірок може використовуватись система керування батареєю (BMS) для моніторингу стану кожної комірки та захисту акумулятора.

Енергоефективність є критично важливим аспектом, оскільки вона безпосередньо визначає максимальний час польоту дрона. На тривалість польоту впливає багато факторів: загальна злітна маса апарату (включаючи батарею та корисне навантаження), аеродинамічна досконаліть конструкції, ефективність рушійної системи (пара "двигун-пропелер"), ємність та стан

("здоров'я") акумуляторної батареї, а також стиль польоту (плавні рухи споживають менше енергії, ніж агресивні маневри) та погодні умови (сильний вітер збільшує енерговитрати). Оптимізація цих параметрів є важливим практичним завданням при проектуванні.

Таким чином, система керування польотом забезпечує стабільність та керованість дрона в повітрі, виконуючи складні обчислення на основі даних з сенсорів. Система енергоживлення, що базується переважно на літієвих акумуляторах, надає необхідну енергію для всіх бортових систем. Практичний вибір компонентів та їх ретельна інтеграція для обох систем є життєво важливими для забезпечення надійності, безпеки та достатньої тривалості польоту дрона при виконанні завдань збору даних.

1.6 Аналіз систем та протоколів бездротової передачі даних

Бездротовий зв'язок є невід'ємною частиною функціонування будь-якого дрона. Він забезпечує канал для керування апаратом, отримання даних про його стан.

Для передачі даних від приймача радіосигналу, встановленого на дроні, до польотного контролера використовуються різні протоколи. Застарілим вважається аналоговий протокол PPM, який передає всі канали по одному дроту, але є чутливим до завад та має обмежену точність і швидкість.

Для безпосереднього керування на невеликих відстанях часто використовуються стандартні пульти радіокерування (RC), що працюють на частоті 2.4 ГГц та використовують протоколи на кшталт SBUS або PPM. Пульт для керування дроном зображено на рисунку 1.15.

Рисунок 1.15 – Пульт для керування дроном Radio Master Boxer [18]

Для забезпечення надійного керування та отримання телеметрії на значних відстанях (кілька кілометрів і більше) застосовуються Long-Range Systems (LRS), наприклад, на базі технологій Crossfire або ELRS, що часто працюють на субгігагерцових частотах (868/915 МГц). На рис. 1.16 наведено приклад ELRS модуля.

Рисунок 1.16 – Приклад ELRS модулів [19]

Технологія LoRa (та мережі LoRaWAN) пропонує надзвичайно велику дальність зв'язку при дуже низькому енергоспоживанні. Однак її пропускна здатність вкрай мала, що робить її придатною переважно для передачі рідких телеметричних даних або показників з повільних сенсорів.

2 РОЗРОБКА АПАРАТНОЇ ПЛАТФОРМИ ДРОНА

2.1 Розробка структурних схеми

Логічна схема на рисунку 2.1 ілюструє потік керування основного борту квадрокоптера. Команди оператора через пульт та приймач ELRS надходять на польотний контролер (STM32). Контролер, використовуючи дані з IMU (MPU6500), розраховує сигнали для системи ESC та двигунів, що створюють рух. Цей рух фіксується IMU, надаючи дані зворотного зв'язку для корекції польоту. Схема також відображає зв'язок польотного контролера з портом модуля ESP32.

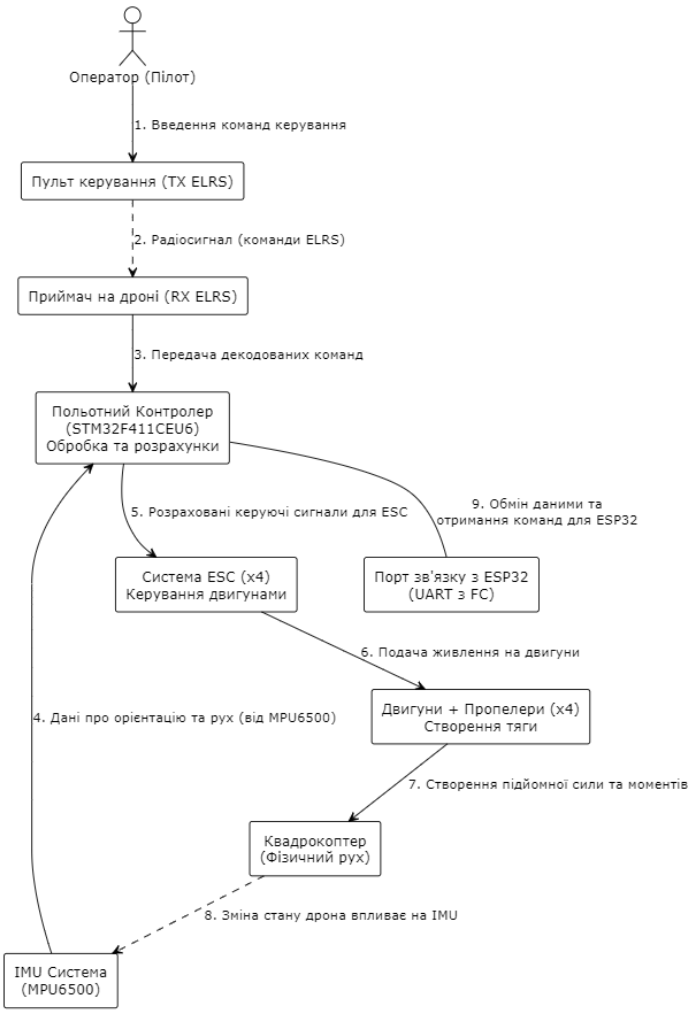

Рисунок 2.1 – Структурна схема керування квадрокоптером

Після розгляду структурної організації та логічного потоку роботи основного борту квадрокоптера, перейдемо до аналізу поведінки підключеного модуля на базі ESP32. Цей модуль, як було зазначено, відповідає за збір та запис даних з додаткових датчиків, керуючись сигналами, отриманими від основного польотного контролера (зокрема, станом кнопки на пульті). Для візуалізації логіки роботи цього модуля та його режимів залежно від зовнішніх подій (сигналу кнопки) зручно використати діаграму станів. Побудована діаграма станів зображена на рисунку 2.2.

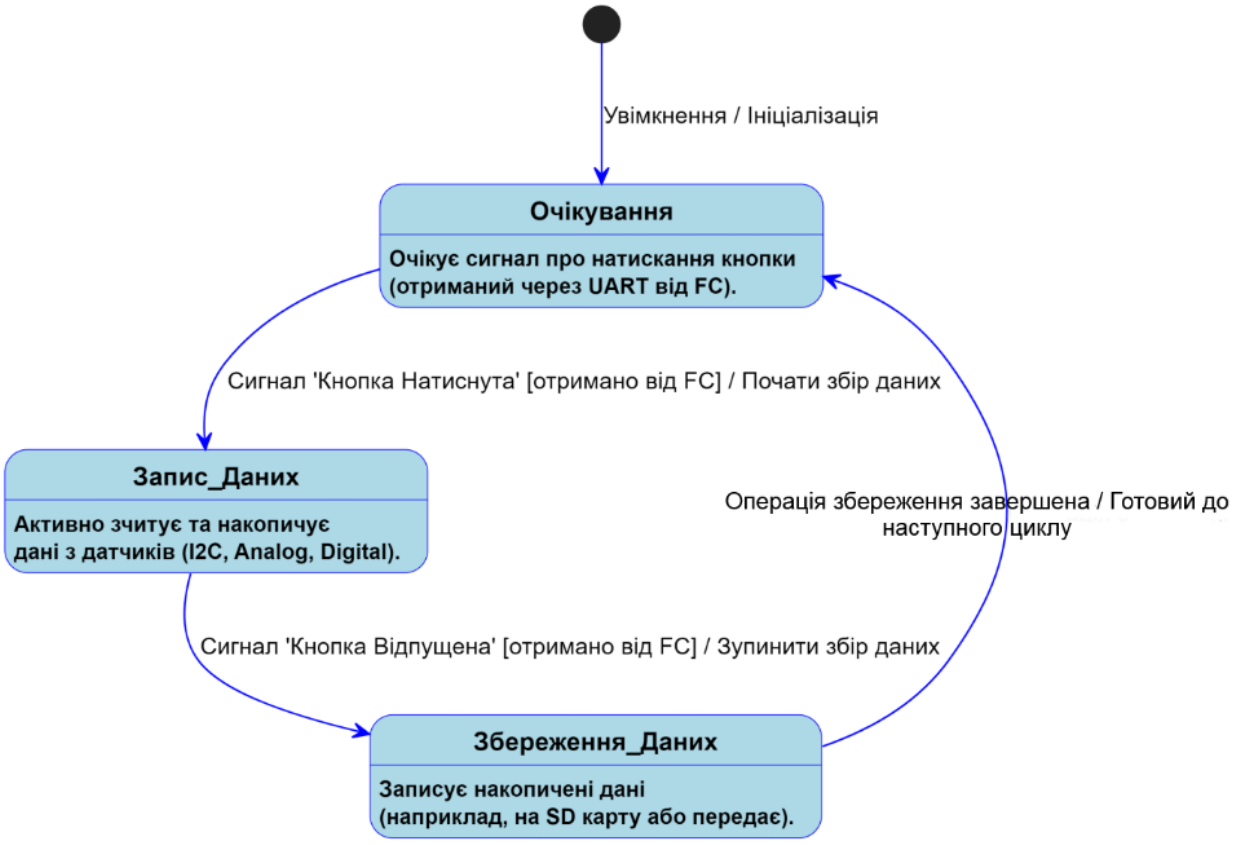

Рисунок 2.2 – Діаграма станів модуля на базі ESP32

На рисунку 2.3 представлена структурна блок-схема основного борту квадрокоптера. Центральний модуль керування (STM32F411CEU6 та IMU MPU6500), отримує команди від приймача ELRS 2.4GHz та дані орієнтації від IMU. На основі цих даних польотний контролер генерує керуючі сигнали для силової установки, яка складається з регуляторів ходу (ESC) та двигунів.

Система живлення забезпечує енергією всі вузли. Схема також відображає наявність інтерфейсу для зв'язку з зовнішнім модулем ESP32. Таким чином, діаграма ілюструє ключові апаратні блоки та основні шляхи передачі сигналів і живлення в системі.

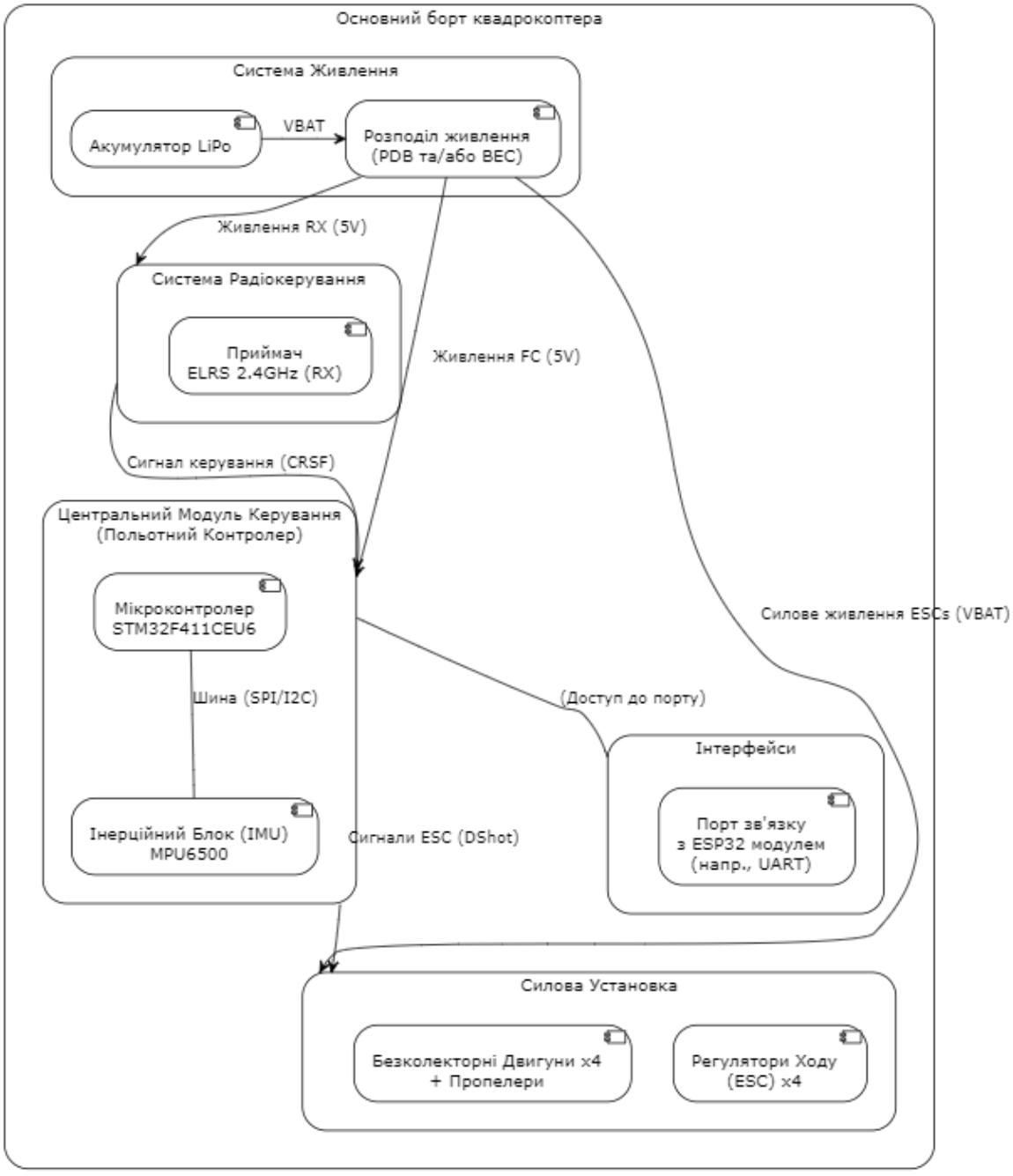

Рисунок 2.3 – Блок схема функціонування квадрокоптера

2.2 Вибір компонентів для створення макету дрона для збору та обробки даних

Ключовим етапом проектування квадрокоптера є обґрунтований вибір компонентів, що формують його несучу систему та силову установку. Ці компоненти визначають основні льотні характеристики, вантажопідйомність, час польоту та загальну надійність апарата.

Для даного проекту було обрано раму 5-дюймового класу типової X-конфігурації. Такий розмір є популярним стандартом, що забезпечує хороший баланс між маневреністю, стабільністю та доступністю компонентів. X-подібна конфігурація рами забезпечує симетричний розподіл мас та тяги, що сприяє більш передбачуваній та стабільній поведінці в польоті, особливо при виконанні маневрів. Як матеріал рами використовується вуглецеве волокно (карбон) завдяки його високій міцності, жорсткості та малій вазі, що є критично важливим для безпілотних літальних апаратів. Обрана рама зображена на рисунку 2.4.

Рисунок 2.4 – Карбонова рама MarkIV

Враховуючи розмір рами (5 дюймів) та заплановане використання 4S LiPo акумулятора, було обрано безколекторні електродвигуни з

характеристикою 2500 KV (рис. 2.5). Таке значення KV забезпечує оптимальну швидкість обертання для 5-дюймових пропелерів на 4S напрузі, дозволяючи досягти як достатньої максимальної тяги, так і прийнятної ефективності в крейсерських режимах. Типовий розмір статора для моторів типу 2306 забезпечує необхідний крутний момент для ефективної роботи з обраними пропелерами.

Рисунок 2.5 – Обрані двигуни

Було обрано пропелери (рис. 2.6) типорозміру 51466, що означає діаметр приблизно 5.1 дюйма та крок 4.66 дюйма. Діаметр 5.1 дюйма відповідає розміру рами та можливостям моторів. Крок 4.66 дюйма є компромісом між високою тягою на низьких обертах, максимальною швидкістю та енергоефективністю, що добре підходить для універсального використання. Матеріал пропелерів – полікарбонат, який забезпечує достатню міцність та гнучкість.

Рисунок 2.6 – Обрані пропелери

Для керування безколекторними двигунами обрано ESC (рис. 2.7), що підтримують сучасний цифровий протокол DShot600. Цей протокол забезпечує високу швидкість передачі керуючого сигналу, хорошу роздільну здатність та стійкість до перешкод порівняно з аналоговими протоколами. Номінальний струм ESC обирався з запасом відносно максимального струму споживання обраної мотоустановки (мотор+пропелер) на 4S напрузі.

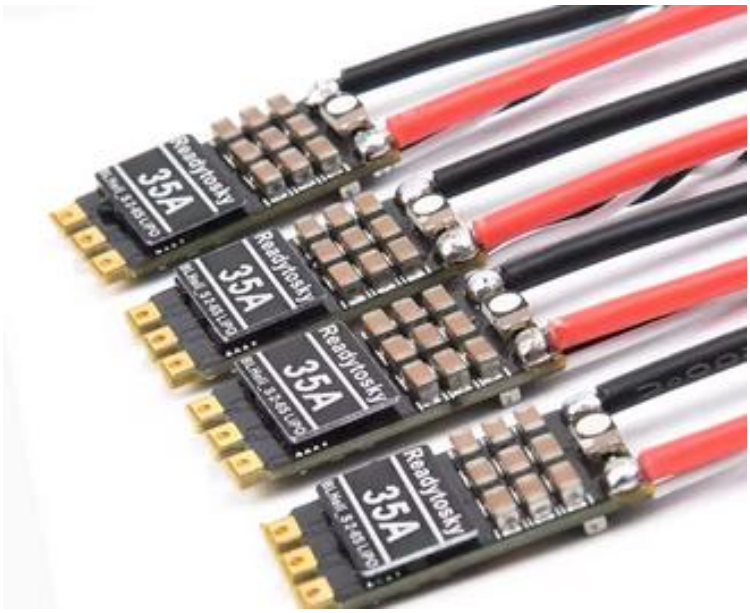

Рисунок 2.7 – Обрані регулятори швидкості [20]

Для даного проекту було вирішено використовувати мікроконтролер STM32 виробництва STMicroelectronics, оскільки вони широко розповсюджені в індустрії дронів, мають розвинену екосистему, хорошу продуктивність та підтримуються основними прошивками для польотних контролерів (Betaflight, INAV, ArduPilot). Для вибору конкретної моделі було проведено порівняння кількох популярних серій мікроконтролерів STM32, що використовуються в сучасних польотних контролерах. У таблиці 2.1 наведено характеристику поширених представників серій STM32, які часто використовуються у польотних контролерах.

Таблиця 2.1 – Порівняння мікроконтролерів на базі STM32 для польотних контролерів

Характеристика	STM32F411 [21]	STM32F405 [22]	STM32F7xx (F722/F745) [23]	STM32H7xx (H743) [24]
Ядро	Cortex-M4	Cortex-M4	Cortex-M7	Cortex-M7
Макс. частота	100 МГц	168 МГц	216 МГц	400+ МГц
Flash	512 КБ	1 МБ	512 КБ	1-2 МБ
SRAM	128 КБ	192 КБ	256-512 КБ	~1 МБ
Кількість UART	до 3	до 6	4-8	4-8

Для створення власного польотного контролера було обрано плату розробника STM32F411CEU6 (рис. 2.8) оскільки вона пропонує оптимальний для даного проекту баланс між обчислювальною потужністю (ядро Cortex-M4, 100 МГц), обсягом пам'яті та набором периферійних пристроїв.

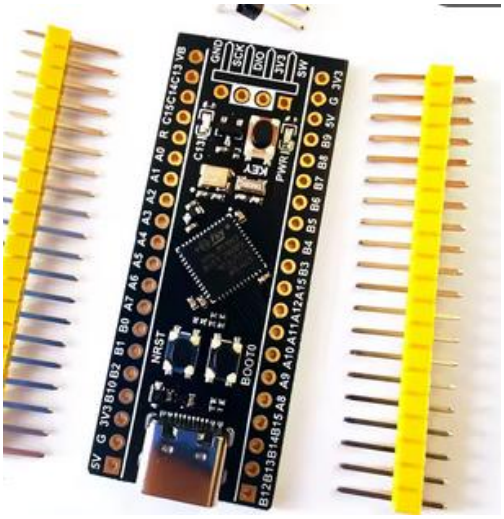

Рисунок 2.8 – Плата розробника STM32F411CEU6 [25]

Після вибору основного мікроконтролера STM32F411CEU6, наступним критично важливим елементом польотного контролера є інерційний вимірювальний модуль (IMU).

Для даного проекту було обрано IMU MPU6500 (рис. 2.9) оскільки він інтегрує 3-осьові гіроскоп та акселерометр (6-DOF), надаючи повну інформацію про рух апарату.

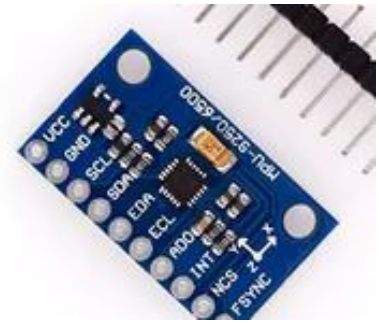

Рисунок 2.9 – IMU MPU6500

Для надійного радіокерування квадрокоптером обрано систему ExpressLRS (ELRS) на частоті 2.4 ГГц (рис. 2.10). Ця відкрита система забезпечує наднизьку затримку, велику дальність та стійкість зв'язку, що є критично важливим для керування безпілотними апаратами. Обрано діапазон 2.4 ГГц як збалансований варіант за дальністю та розміром антени. Відповідний ELRS-приймач підключається до польотного контролера STM32 через UART за швидкісним протоколом CRSF.

Рисунок 2.10 – Обраний приймач SpeedyBee Nano 2.4G ELRS [26]

Для забезпечення візуального контролю та передачі відеосигналу в реальному часі на квадрокоптері встановлено аналогову FPV-систему (рис. 2.11). В якості камери використовується RunCam Phoenix 2, яка відома своєю якістю зображення в різних умовах освітлення. Відеосигнал з камери передається за допомогою відеопередавача (VTX) Rush Tank Ultimate V2, що працює на частоті 5.8 ГГц та забезпечує потужність передачі до 800 мВт для стабільного зв'язку на значні відстані.

Рисунок 2.11 – Обрана FPV система

Для реалізації окремого модуля збору даних з датчиків та забезпечення можливостей IoT [27] (керування записом, передача даних, налаштування) було обрано модуль ESP32-WROOM.Його перевагами є потужний двоядерний процесор, інтегровані Wi-Fi та Bluetooth [28], велика кількість периферійних інтерфейсів для підключення різноманітних датчиків (I2C, UART, ADC, GPIO) та широка підтримка спільнотою розробників. Це робить ESP32-WROOM гнучким та ефективним рішенням для створення автономного сенсорного вузла, що взаємодіє з основним польотним контролером STM32 через інтерфейс UART.

Для розподілу живлення від акумулятора до силової установки (ESC) та інших бортових систем (польотний контролер, приймач, модуль ESP32 тощо) використовується плата розподілу живлення PDB (рис. 2.12).

Рисунок 2.12 – Плата розподілу живлення

Для живлення використовується акумулятор LiPo 4S 1500mAh, що забезпечує оптимальний баланс часу польоту та ваги для даної конфігурації, а висока струмовіддача (100C) гарантує стабільну роботу під навантаженням.

Керування здійснюється за допомогою компактної апаратури Radiomaster Pocket (рис. 2.13) з підтримкою протоколу ELRS та гнучкої прошивки EdgeTX/OpenTX.

Рисунок 2.13 – Апаратура Radiomaster Pocket

2.3. Розробка схеми підключення та збірка макету

Розроблено схему підключення компонентів (рис. 2.14), а саме визначено виводи STM32 для приймача ELRS, сигналів ESC, живлення та для зв'язку з модулем ESP32 (UART) та MPU 6500.

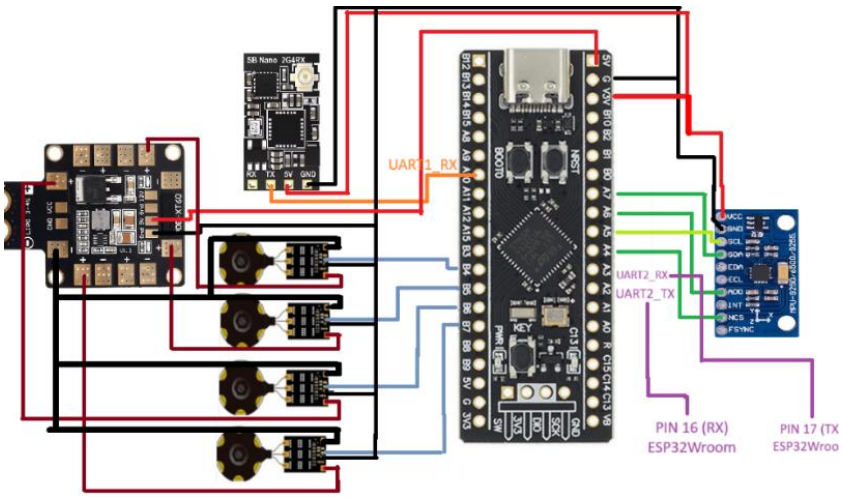

Рисунок 2.14 – Розроблена схема підключення

З огляду на те, що в польоті важлива надійність з'єднань та стійкість до вібрацій, основні компоненти, мікроконтролер STM32F411CE та інерційний модуль MPU6500, розпаюються безпосередньо на макетній платі (рис. 2.15).

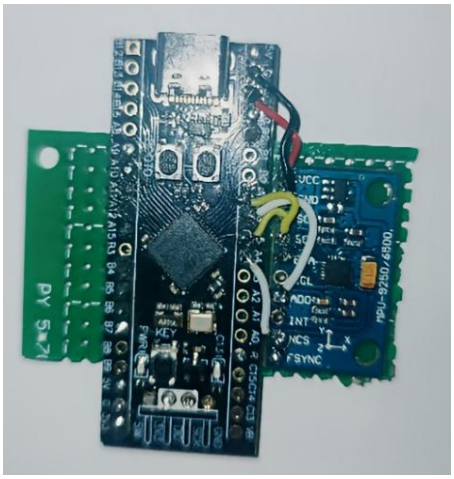

Рисунок 2.15 – Розпаяний модуль польотного контролеру STM32F411CEU6 та MPU6500

Наступним кроком виконується збірка рами дрона та встановлення двигунів на промені. На рисунку 2.16 зображено встановлений на промінь двигун. На рисунку 2.17 зображено результат виконаної роботи, а саме встановлені двигуни та зібрана рама.

Рисунок 2.16 – Встановлений двигун на промінь

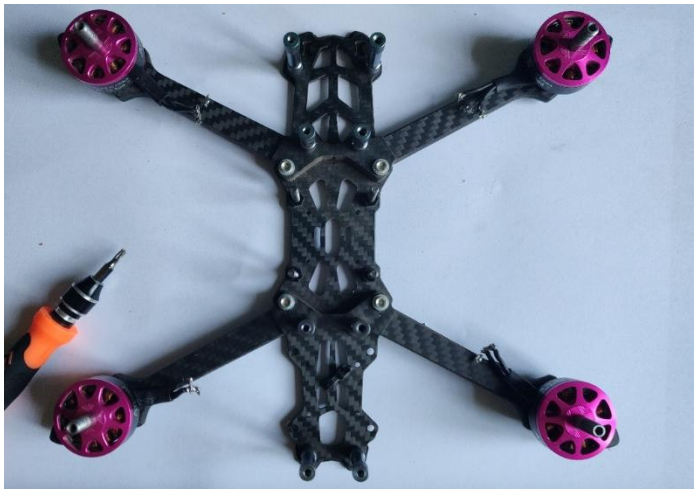

Рисунок 2.17 – Результат, зібрана рама з встановленими двигунами

Наступним кроком виконано розпайку силової установки. Фазні дроти від кожного двигуна були припаяні до відповідних контактних площадок ESC, до плати PDB. Також припаяно силовий провід для акумулятора, замість стандартного, фіксованого у роз’ємі. Результат на рисунку 2.18.

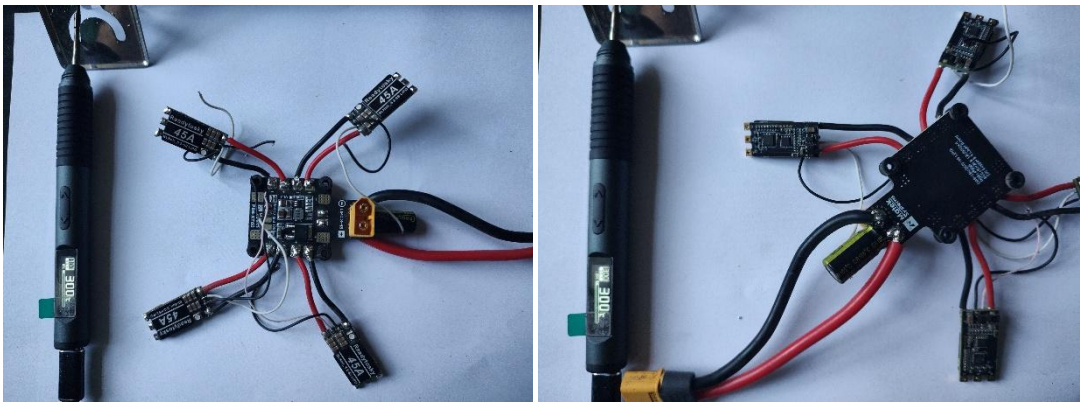

Рисунок 2.18 – Результат розпайки силової установки та ESC

Далі було здійснено підключення моторів до ESC (рис. 2.19).

Рисунок 2.19 – Монтаж та підключення моторів

Наступним кроком виконано монтаж зібраного польотного контролера (ПК) та розпайку до нього відповідних сигнальних проводів від моторів і ELRS-приймача. Результат представлено на рисунку 2.20.

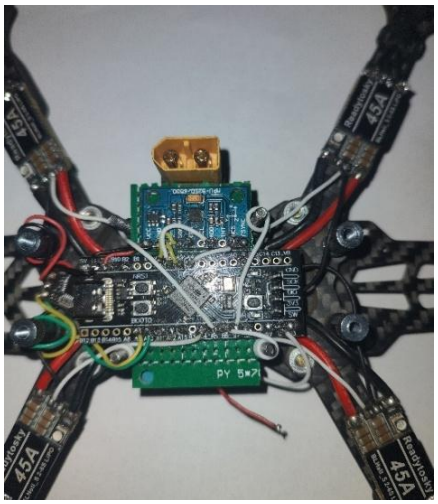

Рисунок 2.20 – Монтаж польотного контролера на раму

Після завершення робіт з силовою установкою, здійснено монтаж FPV камери та VTX відеопередавач на раму. Таким чином етап збірки базової FPV платформи завершено.

Наступним кроком виконано підключення модуля датчиків на базі ESP32. Схема підключення має наступний вигляд (рис. 2.21). З'єднання між польотним контролером та модулем ESP32 було реалізовано за допомогою інтерфейсу UART згідно розробленої раніше схеми підключення.

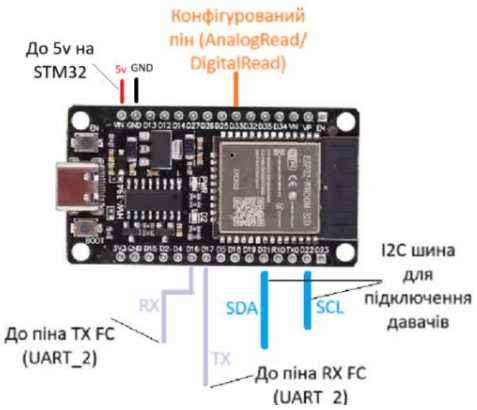

Рисунок 2.21 – Схема підключення модуля ESP32

Модуль розміщено на макетній платі та встановлено на квадрокоптер. Результат приведено на рисунку 2.22.

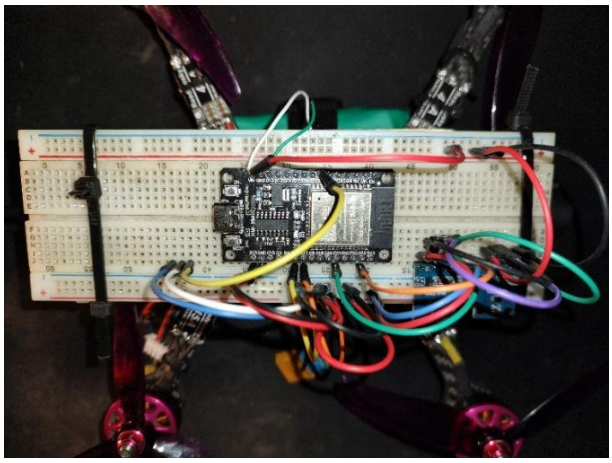

Рисунок 2.22 – Встановлення модуля на дрон

Таким чином, виконавши вибір та підготовку всіх необхідних компонентів, розробивши схему їхнього з'єднання, розпаявши компоненти польотного контролера на макетній платі, зібравши силову установку на рамі та підготувавши модуль ESP32 на безпайковій платі, було здійснено збірку та комутацію основних вузлів квадрокоптера. Результат роботи на рисунку 2.23.

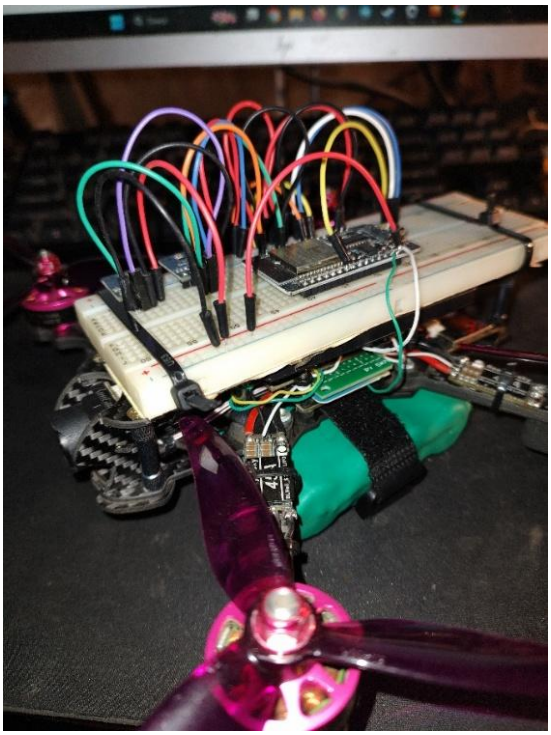

Рисунок 2.23 – Зібраний макет квадрокоптера з модулем для обробки даних

2.4. Аналіз стійкості системи керування

Квадрокоптер – це нелінійна динамічна система. Однак, для аналізу стійкості та проектування системи керування, особливо для режиму стабілізації поблизу точки зависання, використовуємо лінеаризовану модель, яка є наближенням реальної системи при малих відхиленнях.

Спочатку розраховую передавальну функцію системи керування. Передавальна функція об'єкта керування (квадрокоптер + двигуни) від керуючого сигналу Uϕ(s) до кута крену Φ(s) визначається як:

$$Gp(s) = \frac{\Phi(s)}{U_\phi(s)} = \frac{K_{act}}{I_{xx} \cdot s^2 \cdot (\tau_m \cdot s + 1)} \quad (2.1)$$

Підставляючи значення параметрів у формулу (2.1), отримаємо:

$$Gp(s) = \frac{1}{0.005 \cdot s^2 \cdot (0.01 \cdot s + 1)}$$

Завданням системи керування є стабілізація відхиленнях від стану рівноваги. У даному проекті для цього використовується ПІД-регулятор.

Передавальна функція ПІД-регулятора:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d \cdot s \quad (2.2)$$

Тепер складаємо характеристичне рівняння замкненої системи:

Передавальна функція розімкненої системи дорівнює добутку передавальних функцій регулятора та об'єкта (2.3).

$$G_{ol}(s) = G_c(s) \cdot G_p(s)\,, \quad (2.3)$$

Характеристичне рівняння замкненої системи отримуємо з умови:

$$1 + G_{ol}(s) = 0 \tag{2.4}$$

Отриманий характеристичний поліном:

$$P(s) = 1s^4 + 100s^3 + 300000s^2 + 600000s + 400000.$$

Отримавши в результаті попередніх розрахунків характеристичний поліном замкненої системи, ми можемо виконати аналіз стійкості за критерієм Рауса-Гурвіца. Матриця Рауса–Гурвіца для визначення стійкості системи має вигляд:

s^4	1	300000	400000
s^3	100	600000	0
s^2	294000	400000	0
s^1	599864	400000	0
s^0	203966	0	0

Елементи першого стовпця: 1, 100, 294000, 599864, 203966. Всі елементи першого стовпця є позитивними.

Оскільки всі елементи першого стовпця матриці Рауса позитивні, система є стійкою для обраних параметрів.

Однак, сам факт стійкості не дає повної картини щодо якості керування та робастності системи. Для кількісної оцінки ступеня цієї стійкості – тобто визначення запасів по фазі та амплітуді – та аналізу частотних властивостей системи, тепер проведемо аналіз частотних характеристик розімкненої системи за допомогою діаграм Боде.

На основі параметрів системи та ПІД-регулятора, отримуємо частотну передавальну функцію:

$$G_{ol}(j\omega) = \frac{(20-15\omega^2)+j\cdot(30\omega)}{0.00005\omega^4-j\cdot(0.005\omega^3)}$$

Амплітуда в децибелах (dB) для ЛАЧХ:

$$M_{dB}(\omega) = 20 \cdot lg\left|\left(Gol(j\omega)\right)\right| \quad (2.5)$$

Фаза Gol(j*ω) (в градусах) для ЛФЧХ:

$$\phi(\omega) = \frac{(\phi_N(\omega)-\phi_D(\omega))\cdot 180^\circ}{\pi} \quad (2.6)$$

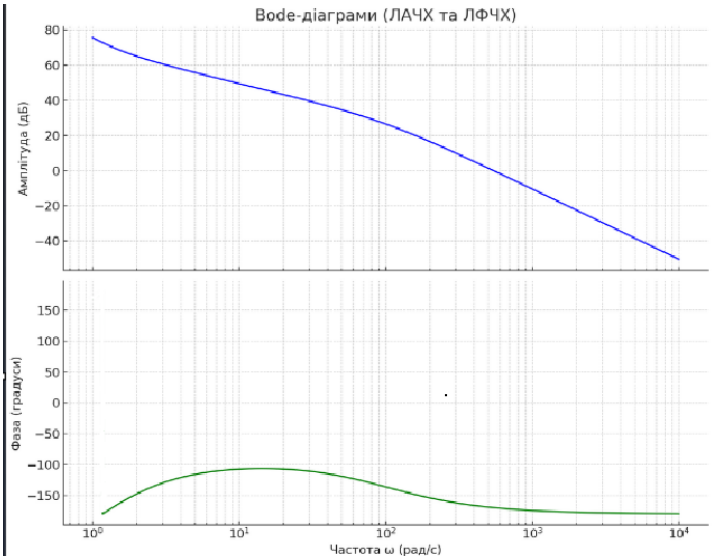

Рисунок 2.24 – Побудовані діаграми Боде розімкненої системи

В результаті аналізу частотних характеристик розімкненої системи керування (Діаграми Боде, представлені на рис. 2.24) встановлено, що частота зрізу wc≈20 рад/с (де ЛАЧХ перетинає 0 дБ), а фазовий зсув (за ЛФЧХ) на цій частоті складає ≈−110°, що забезпечує запас по фазі PM ≈70°. Оскільки фазова характеристика не досягає -180°, запас по амплітуді GM є нескінченно великим. Таким чином, значні запаси стійкості (PM ≈70°, GM ≈ ∞ дБ) підтверджують стійкість спроектованої системи керування для лінеаризованої

моделі. Аналіз за критеріями Боде та Рауса-Гурвіца остаточно підтверджує стійкість система керування.

2.6. Розрахунок та аналіз прогнозованого часу польоту

Для прогнозування часу польоту 5-дюймового квадрокоптера (тип рами: X-рама, двигуни: 2500 KV, пропелери: 51466) з загальною масою m=0.5 кг та акумулятором 4S ємністю 1.5 А·год (струмовіддача 100C), розрахунок починається з визначення тяги, необхідної для зависання:

$$W = m * g \tag{2.7}$$

Необхідна тяга на кожен з чотирьох двигунів:

$$T_{motor} = \frac{W}{4} \tag{2.8}$$

Згідно формулі (2.8):

$$T_{motor} = \frac{4.905\ \text{N}}{4} \approx 1.226\ \text{N}$$

Наступним кроком розраховано корисну ємність акумулятора:

$$C_{\text{Корисний}} = 1.5\ \text{А·год} \cdot 0.80 = 1.2\ \text{А·год}\ ,$$

Прогнозований час польоту:

$$t_{\text{Год}} = \frac{1.2\ \text{А·год}}{12\ \text{А}} = 0.1\ \text{год}; t_{\text{хв}} = 0.1\ \text{год} \cdot 60\ \text{хв/год} = 6\ \text{хвилин}$$

Таким чином час польоту в режимі зависання становить ≈ 6 хвилин.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Конфігурування прошивки Betaflight для створеного польотного контролера

Для забезпечення роботи зібраного польотного контролера під управлінням прошивки Betaflight [29] необхідно виконати два ключові етапи: адаптацію прошивки до конкретної апаратної конфігурації шляхом визначення цілі (target) [30] та подальше налаштування параметрів польоту через Betaflight Configurator [31].

Оскільки зібрана плата польотного контролера є унікальною, стандартні targets Betaflight не повністю відповідають її розпіновці. Тому було обрано підхід створення власного targets [32] для мікроконтролера STM32F411.

Для цього створено файл Black.config та відредаговано його:

```
# Target Definition
TARGET_NAME Black_FC   # Назва FC
MCU_FAMILY STM32F4          # Сімейство MCU
MCU_VARIANT STM32F411xE      # Варіант MCU (CEU6 -> 'E')
board_name BlackPill_FC    # Назва плати
```

Далі вмикаємо основні апаратні можливості та драйвери, які будуть скомпільовані в прошивку. А саме гіроскоп та акселерометр, зокрема драйвери для мікросхем MPU6xxx, підключених через SPI:

```
#define USE_GYRO            # Увімкнути гіроскоп
#define USE_GYRO_SPI_MPU6000 # Використовувати драйвер MPU6xxx для гіроскопа на SPI
```

```
#define USE_ACC             # Увімкнути акселерометр
#define USE_ACC_SPI_MPU6000  #
#define USE_VCP             # Увімкнути USB Virtual COM Port
```

Далі призначаємо піни:

IMU (MPU6500 на SPI1): Піни PA5, PA6, PA7 призначаються для SCK, MISO, MOSI шини SPI1, а PA4 – як Chip Select для гіроскопа (GYRO_CS 1).

```
resource SPI_SCK 1 A05     # SPI1 SCK
resource SPI_MISO 1 A06    # SPI1 MISO
resource SPI_MOSI 1 A07    # SPI1 MOSI
resource GYRO_CS 1 A04     # SPI1 CS для IMU
resource GYRO_EXTI 1 NONE  # Переривання IMU не підключено
```

Мотори**:** Піни PB4, PB5, PB6, PB7 призначаються для виходів сигналів на ESC для моторів 1-4 відповідно. Ресурси для невикористовуваних моторів 5 та 6 встановлюються в NONE:

```
resource MOTOR 1 B04        # Мотор 1 -> PB4
resource MOTOR 2 B05        # Мотор 2 -> PB5
resource MOTOR 3 B06        # Мотор 3 -> PB6
resource MOTOR 4 B07        # Мотор 4 -> PB7
resource MOTOR 5 NONE
resource MOTOR 6 NONE
```

Послідовні порти (UART): Призначаються піни для UART1 для приймача ELRS та UART2 для модуля ESP32:

```
resource SERIAL_TX 1 A09     # UART1 TX -> ELRS
resource SERIAL_RX 1 A10     # UART1 RX -> ELRS
```

```
resource SERIAL_TX 2 A02     # UART2 TX -> ESP32
resource SERIAL_RX 2 A03     # UART2 RX -> ESP32
```

Для всіх інших функцій (PPM, LED_STRIP, I2C, ADC_BATT, ADC_CURR, OSD_CS, BEEPER тощо), які не використовуються в даній апаратній реалізації, ресурс встановлюється в NONE , щоб звільнити піни та ресурси мікроконтролера.

Встановлємо призначення для стандартного світлодіода PC13 та детектування USB PC15 для плати stm32.

Далі налаштовуємо таймери та DMA для моторів:

```
timer B04 AF2          # TIM3_CH1
timer B05 AF2          # TIM3_CH2
timer B06 AF2          # TIM4_CH1
timer B07 AF2          # TIM4_CH2
dma pin B04 0          # DMA1 Stream 4 Channel 5
dma pin B05 0          # DMA1 Stream 5 Channel 5
dma pin B06 0          # DMA1 Stream 0 Channel 2
dma pin B07 0          # DMA1 Stream 3 Channel 2
```

Налаштовуємо основні програмні функції, які будуть увімкнені в прошивці:

```
feature RX_SERIAL              # Увімкнення підтримки послідовного приймача
feature TELEMETRY              # Увімкнення телеметрії (важливо для ELRS)
feature AIRMODE            # Увімкнення режиму Airmode
```

Також задаємо параметри майстра які будуть встановлені в Betaflight за замовчуванням після прошивки:

```
master
set motor_pwm_protocol = DSHOT600  # Протокол ESC
set serialrx_provider = CRSF        # Протокол приймача (ELRS використовує CRSF)
set gyro_1_bustype = SPI        # Тип підключення IMU
set gyro_1_spibus = 1         # Номер шини SPI для IMU
# ВАЖЛИВО: Перевірити та встановити правильну орієнтацію IMU!
set gyro_1_sensor_align = CW270 # Приклад! Замініть на реальне значення!
# Вимкнення невикористовуваних датчиків
set baro_hardware = NONE
set mag_hardware = NONE
set current_meter = NONE
set battery_meter = NONE
```

3.2 Розробка прошивки для модуля ESP32 для збору даних та передачі інформації

Для реалізації функцій збору даних з зовнішніх датчиків та взаємодії з основним польотним контролером (FC) була розроблена спеціалізована прошивка для модуля ESP32-WROOM. Розробка велася в середовищі Arduino IDE з використанням мови C++ та відповідних бібліотек для ESP32.

Зв'язок між ESP32 та польотним контролером здійснюється через апаратний інтерфейс UART (Serial2 на ESP32, підключений до UART2 на FC) на швидкості 115200 бод. Для обміну даними використовується протокол MSP (MultiWii Serial Protocol), поширений у прошивках типу Betaflight.

Ключовою функцією взаємодії є отримання стану перемикача на пульті оператора, який використовується як тригер для старту/зупинки запису даних.

Модуль ESP32 періодично надсилає запит MSP_RC (код 105) до FC, щоб отримати поточні значення всіх радіоканалів. Це робиться у функції handleMspCommunication().

```
void handleMspCommunication() {
   unsigned long currentTime = millis();
   // ... (запит MSP_ATTITUDE) ...
   // Періодичний запит RC каналів
      lastMspRcRequestTime = currentTime;
      sendMSPRequest(MSP_RC); // Відправка запиту MSP_RC (код 105)
```

Важливою особливістю модуля є здатність автоматично визначати наявність підтримуваних датчиків на шині I2C (див. додаток А) під час старту системи. Це реалізовано у функції scanAndDetectSensors().

```
void scanAndDetectSensors() {
   Serial.println("Сканування шини I2C...");
   byte error, address;
   int nDevices = 0;
     bh1750_detected = false;
  for(address = 1; address < 127; address++ ) {
      Wire.beginTransmission(address);
      if (error == 0) { // Пристрій знайдено
```

Функція scanAndDetectSensors() викликається з setupSensorsAndFs() в модулі sensor_handler.cpp. Спочатку ініціалізується шина I2C, а потім відбувається сканування всіх можливих адрес.

Для забезпечення зв'язку між дроном (ESP32) та мобільним застосунком використовується комбінація протоколів: WiFi Access Point: ESP32 розгортає власну точку доступу WiFi, до якої може підключитися користувач, HTTP Server: ESP32 запускає HTTP-сервер для обслуговування веб-сторінки інтерфейсу та обробки RESTful API запитів та HTTP Server: ESP32 запускає HTTP-сервер для обслуговування веб-сторінки інтерфейсу та обробки RESTful API запитів.

Завдяки WebSocket, веб-сторінка отримує та відображає поточні значення крену (Roll), тангажу (Pitch) та курсу (Yaw) дрона без необхідності постійного оновлення сторінки.

```
void sendAttitudeDataViaWebSocket() {
  StaticJsonDocument<128> jsonDoc;
  jsonDoc["roll"] = round(attitudeRoll * 10.0) / 10.0;
  jsonDoc["pitch"] = round(attitudePitch * 10.0) / 10.0;
  jsonDoc["yaw"] = round(attitudeYaw);
  String jsonString;
  serializeJson(jsonDoc, jsonString);
  webSocket.broadcastTXT(jsonString);
```

Користувач може ініціювати процес калібрування акселерометра дрона натисканням відповідної кнопки на веб-сторінці. При цьому відправляється HTTP POST запит на сервер ESP32, який, у свою чергу, передає команду калібрування польотному контролеру через MSP протокол.

```
function calibrateAcc() {
  fetch('/calibrate', { method: 'POST' })
    .then(response => response.text())
    .then(text => {
      document.getElementById('calibrateStatus').textContent = text;
```

Список датчиків, виявлених ESP32 тримується через HTTP GET запит, на який ESP32 відповідає JSON-об'єктом зі списком назв виявлених сенсорів.

```
function fetchActiveSensors() {
    fetch('/active_sensors')
        .then(response => response.json())
        .then(data => {
            if (data.detected_sensors && data.detected_sensors.length > 0) {
                document.getElementById('sensorList').textContent =
data.detected_sensors.join(', ');
```

Користувач може запросити статистику зібраних датчиками даних (мінімальне, максимальне, середнє значення), надіславши HTTP GET запит. ESP32 у відповідь надсилає JSON-файл зі збереженою статистикою (якщо такий файл існує).

Веб-інтерфейс надає можливість встановлювати тип датчика (цифровий або аналоговий), підключеного до конфігурованого піна ESP32. Це здійснюється через HTTP POST запит з відповідним параметром.

```
function setSensorType(type) {
    fetch('/set_sensor_type', {
        method: 'POST',
        headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
        body: `type=${type}`
    })
    .then(response => response.text())
    .then(text => {
        alert(text); // Або відобразити статус іншим способом
    })
```

3.3 Розробка OSD інтерфейсу

У рамках створення системи виводу телеметричних даних на екран було реалізовано інтерфейс на базі чіпа MAX7456, що відповідає за накладання графічної інформації поверх відеосигналу.

На відміну від типових конфігурацій, у яких обмін даними з OSD здійснюється через контролери типу Arduino або полетні контролери з підтримкою MSP-протоколу, в даній реалізації OSD-модуль безпосередньо приймає телеметричні дані по UART від мікроконтролера ESP32. Такий підхід забезпечує гнучкість у форматуванні та відображенні даних, зокрема показників напруги живлення та даних давачів.

Передача даних з ESP32 до OSD здійснюється через UART з використанням простого текстового протоколу, де кожен тип повідомлення ідентифікується за префіксом. Наприклад: $V це напруга акумулятора в вольтах, $D1 це дані першого давача, $D2 це дані другого давача.

Чіп MAX7456 конфігурується у режим PAL. Для оновлення інформації на екрані використовується попереднє очищення символів на відповідних позиціях з подальшим виведенням нових значень. Це дозволяє уникнути артефактів при частому оновленні даних. Таким чином виводжу на екран дані з датчиків у реальному часі.

Спочатку, у головному циклі loop(), після отримання даних через послідовний порт, значення asd та asd2 (які представляють дані від "Davach1" та "Davach2" відповідно) перетворюються з цілочисельного типу (int) у строковий формат (char*) за допомогою функції itoa:

```
itoa(asd, bufffer, 10);
itoa(asd2, bufffer2, 10);
```

Далі, для кожного значення створюється буфер фіксованої довжини (outputBuffer та outputBuffer2), який спочатку заповнюється пробілами. Це

необхідно для того, щоб при оновленні значення на екрані попередній текст повністю замінювався новим, запобігаючи залишкам старих символів.

```
char outputBuffer[10] = "          ";
char outputBuffer2[10] = "          ";
strncpy(outputBuffer, bufffer, 9);
strncpy(outputBuffer2, bufffer2, 9);
```

Після підготовки строкових представлень значень, вони виводяться на екран OSD за допомогою функції osd_print. Для ідентифікації кожного значення також виводяться відповідні мітки "Davach1:" та "Davach2:"

```
osd_print(name1, strlen(name1), 1, 10);
osd_print(outputBuffer, strlen(outputBuffer), 9, 10);
osd_print(name2, strlen(name2), 1, 11);
osd_print(outputBuffer2, strlen(outputBuffer2), 9, 11);
```

Також виводимо напругу акумулятора для моніторингу розряду. Основна логіка виведення напруги інкапсульована у функції readvolts, яка приймає як аргументи ідентифікатор джерела напруги (input), позицію виведення по осі X (xpos) та по осі Y (ypos).

```
voltToDisplay = ina219_voltage;
len = floattochar(buffer, voltToDisplay);
osd_print(buffer, len + 1, VOLT1XPOS, VOLT1YPOS);
```

Для моніторингу часу польоту створюється простий системний таймер:

```
unsigned long elapsedTime = (millis() - startTime) / 1000; // Час у секундах
int minutes = elapsedTime / 60;
```

int seconds = elapsedTime % 60;

Для виводу таймеру на оверлей:

osd_print(timerBuffer, strlen(timerBuffer), 13, 1);

Повний код функції виведення OSD наведено у додатку Б.

Одним з ключових завдань даної роботи була інтеграція системи OSD для відображення важливої телеметричної інформації безпосередньо на відеопотоці. Це дозволяє оператору в режимі реального часу отримувати дані про стан системи, такі як напруга живлення, значення з датчиків та час польоту, що значно підвищує інформативність та зручність моніторингу. На рисунку 3.1 зображено вивід інформації, що відображається на екрані OSD після успішної реалізації даного етапу роботи.

Рисунок 3.1 – Відображення OSD даних у реальному часі

Як видно на представленому зображенні, система OSD успішно відображає в режимі реального часу ключові параметри. У верхній частині екрана відображається таймер, а нижче представлені значення, отримані з давачів "Davach1" та "Davach2. Також відображається напруга акумулятора у нижньому лівому куті.

3.4 Розробка Android застосунку

У рамках даної роботи було розроблено мобільний застосунок для платформи Android, у середовищі Android Studio на мові програмування Kotlin.Головна мета застосунку спростити налаштування польотного контролера та забезпечити зручну роботу з телеметричними давачами.

Застосунок складається з двох основних екранів, кожен з яких виконує свою функціональну роль.Код взаємодії з дроном наведено у додатку В.

На першому екрані відбувається налаштування польотного контролера. На даному етапі розробки реалізовано функціонал калібрування. Користувач має можливість ініціювати процес калібрування ПК без необхідності підключення до персонального комп’ютеру. Інтерфейс цього екрану розроблено таким чином, щоб процес калібрування був інтуїтивно зрозумілим. На рисунку 3.2 зображено створений інтерфейс.

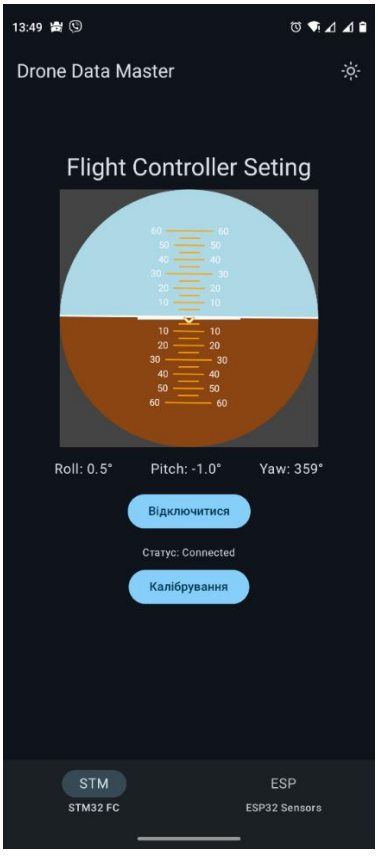

Рисунок 3.2– Інтерфейс калібрування дрона з мобільного застосунку

Другий єкран застосунку (рис. 3.3) призначений для взаємодії з давачами, підключеними до польотного контролера. Його функціональність включає відображення активних датчиків.На екрані в реальному часі відображається перелік давачів, які на даний момент передають дані. Це дозволяє користувачеві швидко отримати інформацію про те, які сенсори функціонують. А також можливість сконфігурувати окремий пін для роботи з аналоговим або цифровим давачем. Також користувач має можливість ініціювати процес запису даних з активних датчиків у локальний журнал. Це може бути корисним для подальшого аналізу телеметричної інформації після польоту. Також реалізовано збереження даних з давачів у журнал даних.

Кожен запис у журналі містить часову мітку завантаження та значення, отримані від активних датчиків на момент запису. Це забезпечує можливість відстеження змін параметрів протягом певного періоду часу.

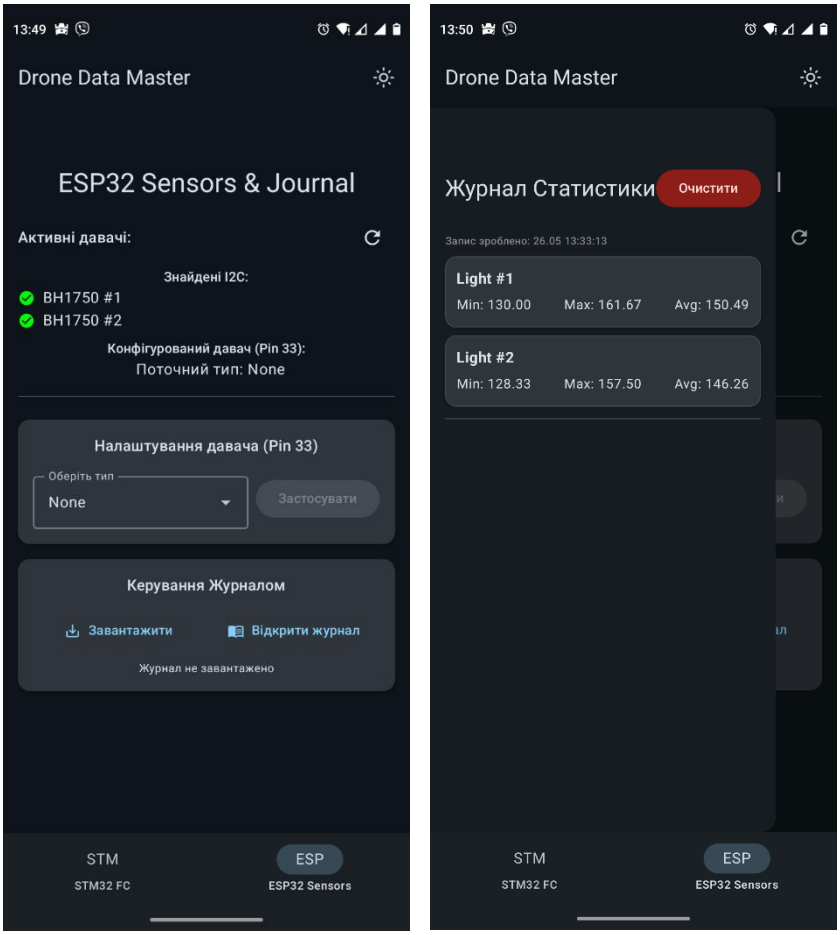

Рисунок 3.3 – Екран керування давачами

4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНОГО ТЕСТУВАННЯ

4.1 Тестування розробленого дрона та модуля для обробки даних

Спочатку протестовано інтерфейс калібрування дрона. Для цього з мобільного застосунку відправлено відповідну команду до модуля на базі ESP32, який, у свою чергу, передасть дані до ПК. Результат до та після калібрування представлено на рисунку 4.1.

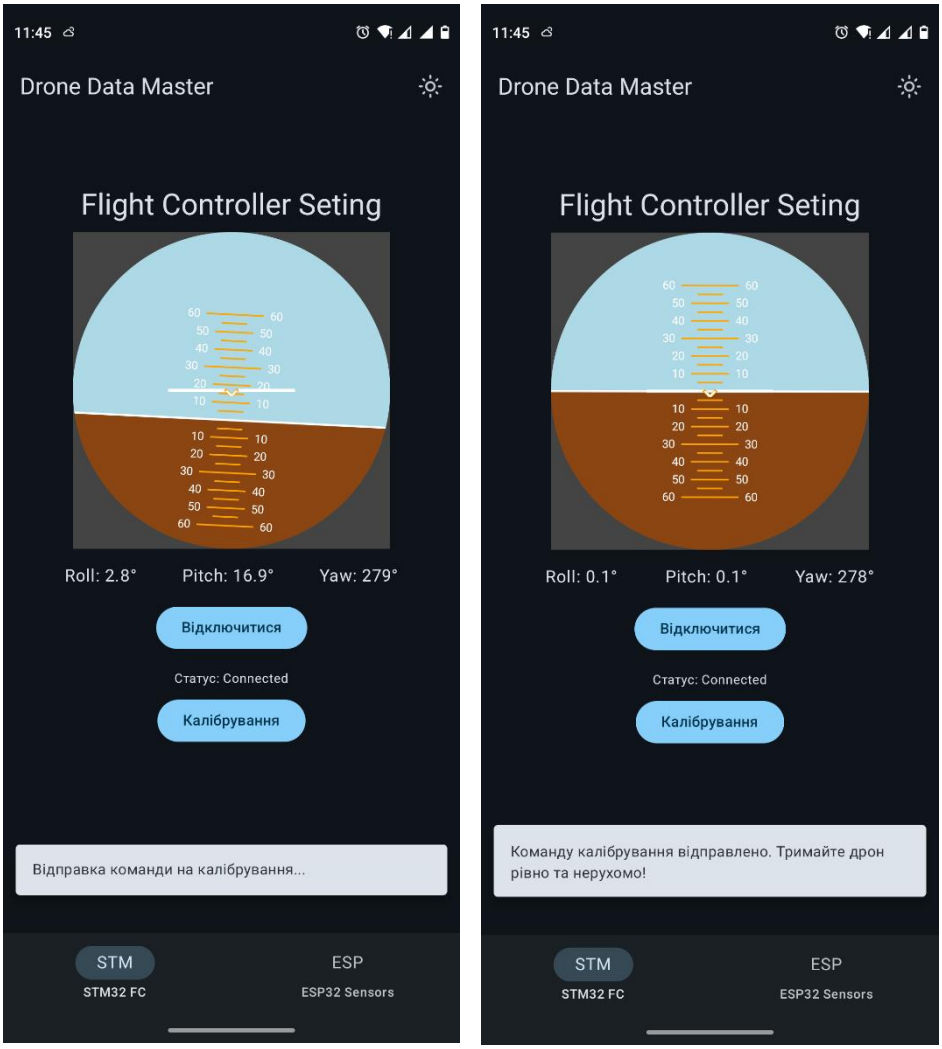

Рисунок 4.1– Проведення калібрування дрону

Наступним кроком протестовано автоматичне розпізнавання давачів, підключених по I2C-шині. Спочатку під'єднано два давачі. На рисунку 4.2 зображено два підключених давачі, а на рисунку 4.3 їх відображення у мобільному застосунку.

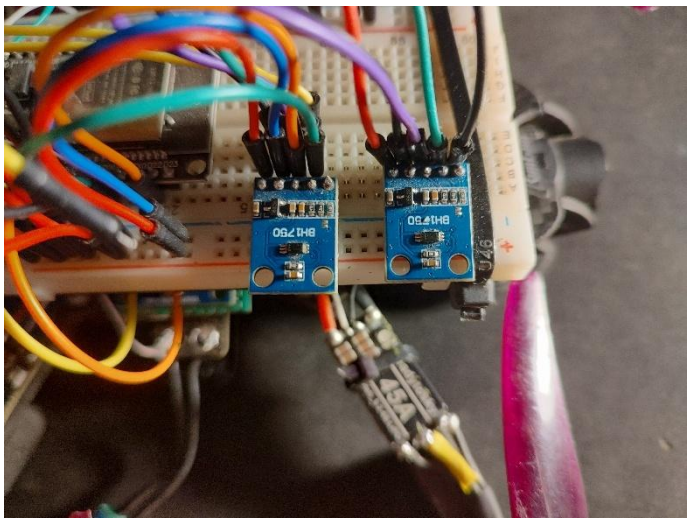

Рисунок 4.2 – Під’єднання двох давачів до модуля з esp32

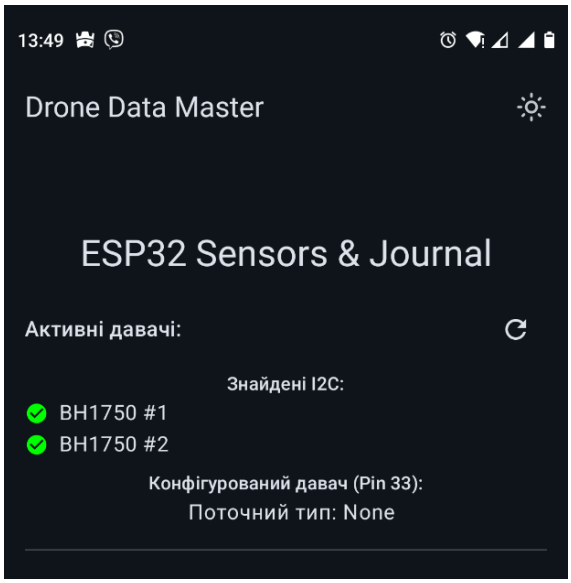

Рисунок 4.3 – Ініціалізація двох давачів згідно підключення

Тепер від'єднано давач "#1" та оновлено застосунок. На рисунку 4.4 зображено підключений давач, а на рисунку 4.5 оновлений застосунок.

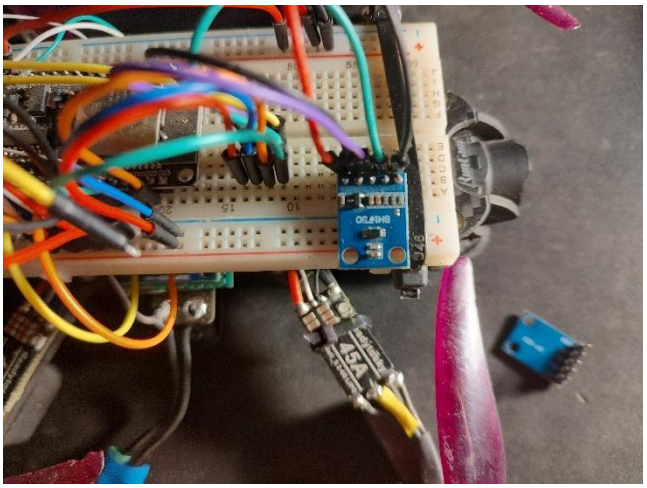

Рисунок 4.4 – Підключення лише другого давача

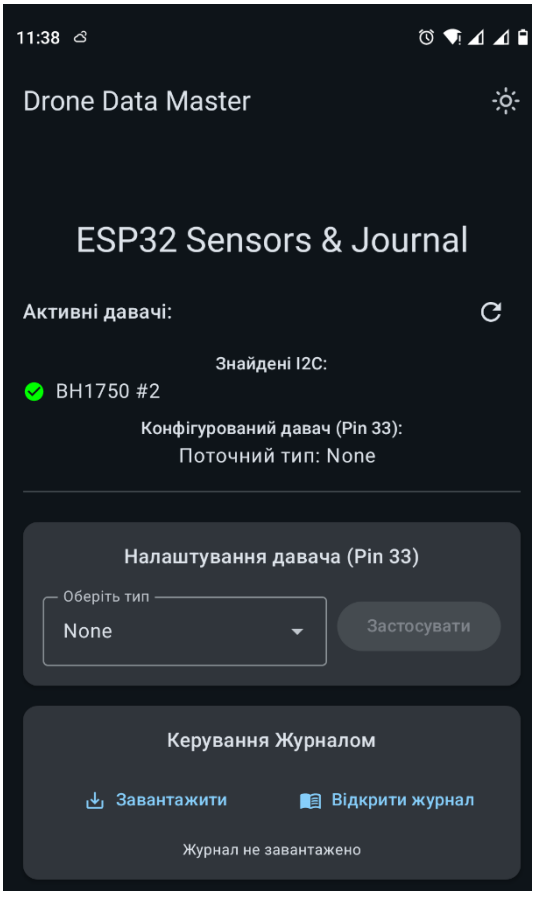

Рисунок 4.5 – Відображення підключених давачів

Аналогічно якщо від’єднати давач #2 залишиться лише перший (рис. 4.6).

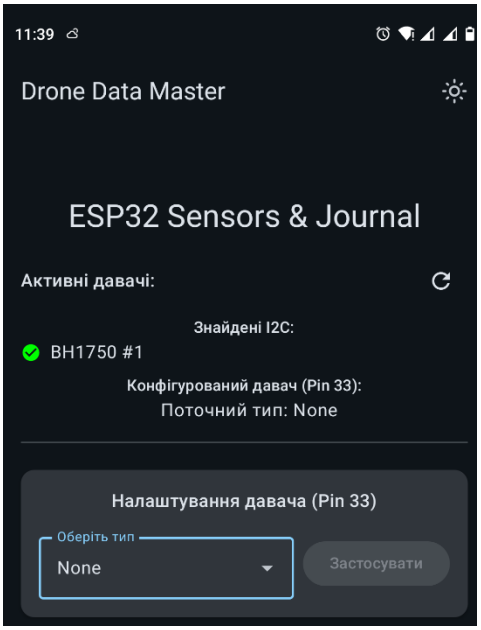

Рисунок 4.6 – Оновлення екрану застосунку

Далі протестовано конфігурований пін. Тип піну встановлено як "Digital". Результат представлено на рисунку 4.7.

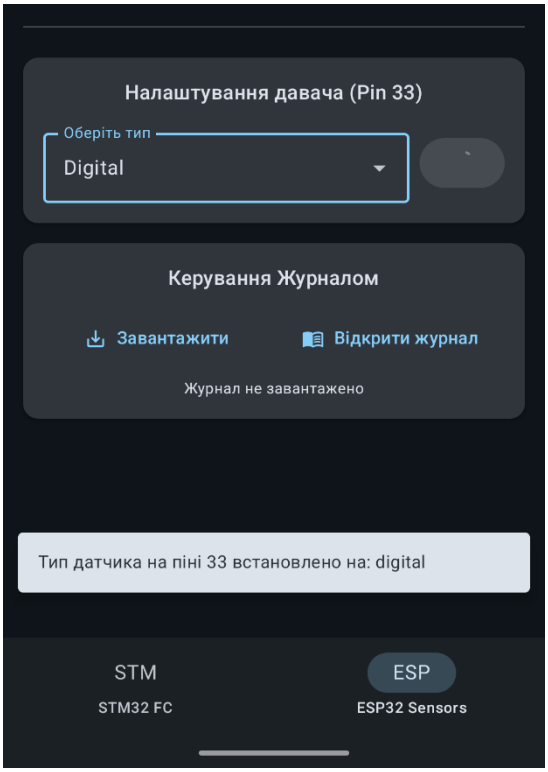

Рисунок 4.7 – Сконфігуровано пін 33 для цифрового читання

4.2 Польотні тести

Розроблений макет дрона встановлено на асфальтне покриття (рис. 4.8).

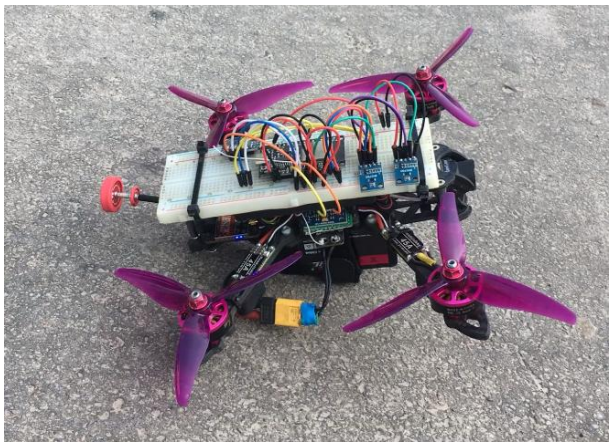

Рисунок 4.8 – Розроблений макет дрона

Виконано команду армінгу для активації гвинтів (рис. 4.9).

Рисунок 4.9 – Виконання команди ARM

Результат роботи макету зображено на рисунку 4.10.

Рисунок 4.10 – Зліт дрона

Дані з давачів успішно виводяться на єкран у режимі OSD, поверх зображення з камери у режимі реального часу. Також інтерфейс відображае поточну напругу акумулятора і напругу 5v лінії живлення. Результат працюючого OSD інтерфейсу представлено на рисунку 4.11.

Рисунок 4.11 – Отримання даних з давачів у реальному часі

Після польоту можна завантажити загальну статистику до мобільного застосунку (рис. 4.12).

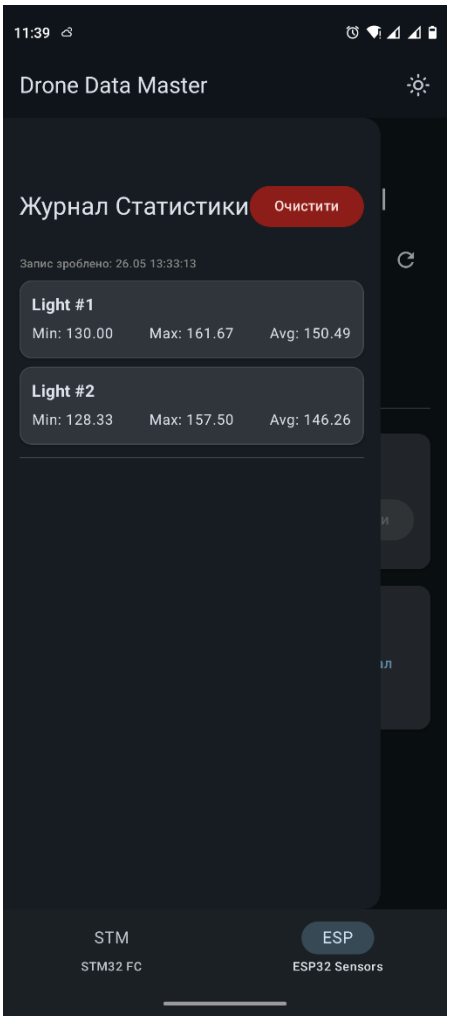

Рисунок 4.12 – Завантажені дані до журналу після польоту

4.2 Аналіз лабораторних умов

У процесі створення дрону для збору та обробки інформації, особлива увага приділяється безпечним умовам праці при проєктуванні, програмуванні, збиранні та експлуатації системи.

Робоче місце було організовано із дотриманням державних санітарних норм і правил. Забезпечено достатній рівень освітлення (не менше 300 лк), природну та примусову вентиляцію, ергономічність розміщення обладнання.

Усі електротехнічні прилади, що використовувались під час розробки дрона, проходили регулярну перевірку на справність.

При виконанні монтажних і налагоджувальних робіт використовувалися безпечні джерела живлення з низькою напругою (5–16.8 В), що відповідає I класу електробезпеки. Додатково вживалися заходи щодо захисту від коротких замикань за допомогою запобіжників, заземлення металевих частин обладнання та використання інструменту з ізольованими ручками.

У проекті використано радіозв’язок у діапазоні 2.4 ГГц, що відповідає міжнародним стандартам безпеки для користувача.

При розробці та тестуванні автоматизованих алгоритмів управління застосовувалися методи моделювання та симуляції в середовищі MATLAB/Simulink та Arduino IDE. Це дозволило мінімізувати ризики некоректної поведінки системи під час реального запуску.

При експлуатації дрону враховуються вимоги чинного законодавства України щодо: заборони польотів у заборонених зонах (поблизу аеропортів, військових об’єктів). Збору та обробки інформації відповідно до Закону України «Про захист персональних даних».

ВИСНОВКИ

В ході виконання кваліфікаційної роботи, було здійснено розробку макету дрону на базі ARM-мікроконтролера для збору та обробки даних у реальному часі.

Виконано наступні завдання:

– проведено аналіз існуючих систем дронів;

– розроблено структурні схеми дрона, що визначають архітектуру системи та взаємозв'язки її ключових компонентів;

– здійснено вибір основних компонентів для апаратної платформи;

– складено експериментальний макет апаратної платформи дрона;

– проведено аналіз стійкості розробленої системи керування польотом для лінеаризованої моделі дрона, який підтвердив її працездатність та наявність достатніх запасів стійкості (зокрема, запас по фазі PM ≈ 70° та нескінченний запас по амплітуді GM);

– розроблено та сконфігуровано програмне забезпечення для польотного контролера на базі прошивки Betaflight, адаптоване до апаратних особливостей створеної платформи та завдань керування;

– розроблено програмне забезпечення для модуля збору даних на базі ESP32, яке реалізує збір інформації з сенсорів, її попередню обробку та передачу по бездротовому каналу зв'язку.

– проведено експериментальне тестування розробленого макету дрона, у результаті якого підтверджено його працездатність: дрон здійснює політ, коректно передає дані в режимі реального часу, а мобільний застосунок успішно взаємодіє з пристроєм.

У результаті було досягнуто основної мети кваліфікаційної роботи, а саме, було удосконалено процес збору та автоматизованої обробки даних у реальному часі шляхом розробки інтегрованої системи моніторингу на базі дрона з інтелектуальним модулем на базі ESP 32.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. - 29 с.

2. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми.

3. Кафедра комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки. Про нас. Офіційний сайт кафедри КІТАМ ХНУРЕ. Режим доступу: https://tapr.nure.ua/golovna/pro-nas. Дата доступу: (1.05.2025).

4. Квадрокоптер DJI Mavic 3 Fly [Електронний ресурс] – Режим доступу: https://vistrel.kiev.ua/uk/kvadrokoptery/kvadrokopter-dji-mavic-3-fly-more-combo-dron-hasselblad-20mp-kameroyu-do-30-km-do-46-hvilin-sumka/ (дата звернення: 28.04.2025).

5. Drone delivery transporting. [Електронний ресурс] – Режим доступу: https://www.euroshop-tradefair.com/en/euroshopmag/drone-delivery-transporting-goods-in-the-retail-sector (дата звернення: 15.04.2025).

6. Visual and NDVI field maps. [Електронний ресурс] – Режим доступу: https://drone.ua/en/pages/visual-and-ndvi-field-maps
(дата звернення: 16.04.2025).

7. Point cloud classification. [Електронний ресурс] – Режим доступу: https://www.pix4d.com/blog/construction-surveys-point-cloud-classification/ (дата звернення: 17.04.2025).

8. Broadcast Equipment. [Електронний ресурс] – Режим доступу: https://uk.fmuser.net/content/?10492.html (дата звернення: 18.04.2025).

9. CS32 MCU is an STM32 clone, and the Blue Pill board sells for under [Електронний ресурс] – Режим доступу: https://www.cnx-software.com/2019/02/10/cs32- mcu-stm32-clone-bluepill-board/
(дата звернення: 18.04.2025).

10. Плата розробника ESP32-WROOM-32 (WiFi+Bluetooth) [Електронний ресурс] – Режим доступу: https://www.aliexpress.com/item/1005006456519790.html.
(дата звернення: 20.04.2025).

11. KNACRO Nine-axis Module Acceleration Three-axis Gyroscope. [Електронний ресурс] – Режим доступу: https://www.amazon.sg/KNACRO-Nine-axis-Acceleration-Three-axis-Accelerometer/dp/B01ID63R1G
(дата звернення: 20.04.2025).

12. FLIR Drone Thermal Cam. [Електронний ресурс] – Режим доступу: https://www.flir.com/suas/?srsltid=AfmBOop2eQ60Md2-rv0aGApb5qIOGBBXI4zQmiqCAbiJQSz_P91YJaGK
(дата звернення: 20.04.2025).

13. M300 Third-Party Payload. [Електронний ресурс] – Режим доступу: https://enterprise-insights.dji.com/blog/m300-third-party-payloads
(дата звернення: 20.04.2025).

14. Drones predict violent individuals from sky. [Електронний ресурс] – Режим доступу: https://www.digitaltrends.com/cool-tech/drones-predict-violent-individuals-from-sky/ (дата звернення: 1.04.2025).

15. Matek F405-AIO. [Електронний ресурс] – Режим доступу: https://flymod.net/item/matek_f405_aio
(дата звернення: 20.04.2025).

16. POI BATTERU For. [Електронний ресурс] – Режим доступу: https://polydrone.com.ua/ua/p1934074986-poi-batteru-for.html
(дата звернення: 20.04.2025).

17. PowerSmart® 3400mAh Li-ion Akku für Gardena 9676473-03. [Електронний ресурс] – Режим доступу: https://www.amazon.de/PowerSmart%C2%AE-3400mAh-Li-ion-Gardena-9676473-03/dp/B09126FP18 (дата звернення: 20.04.2025).

18. Radiomaster Boxer ELRS. [Електронний ресурс] – Режим доступу: https://prom.ua/ua/m-17098352937039537 44-radiomaster-boxer-elrs.html (дата звернення: 10.04.2025).

19. Ranger Nano 2.4GHZ ExpressLRS RF Starter Combo. [Електронний ресурс] – Режим доступу: https://www.mahmoudishop.de/Ranger-Nano-24GHZ-ExpressLRS-RF-Starter-Combo (дата звернення: 20.04.2025).

20. Регулятор швидкості Readytosky 35A/45A BLHeli_S ESC [Електронний ресурс] – Режим доступу: https://www.aliexpress.com/item/1005004793689505.html. (дата звернення: 20.04.2025).

21. STMicroelectronics. STM32F411 Datasheet [Електронний ресурс] – Режим доступу: https://www.st.com/en/microcontrollers-microprocessors/stm32f411.html. (дата звернення: 20.04.2025).

22. STMicroelectronics. STM32F405xx/STM32F407xx Datasheet [Електронний ресурс] – Режим доступу: https://www.st.com/resource/en/datasheet/dm00037051.pdf. (дата звернення: 20.04.2025).

23. STMicroelectronics. Серія STM32F7 Datasheet [Електронний ресурс] – Режим доступу: https://www.st.com/en/microcontrollers-microprocessors/stm32f7-series.html (дата звернення: 20.04.2025).

24. STMicroelectronics. Серія STM32H7 [Електронний ресурс] – Режим доступу: https://www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html (дата звернення: 15.04.2025).

25. Плата розробника WeAct BlackPill STM32F411CEU6 [Електронний ресурс] Режим доступу: https://aliexpress.com /1005001456186625.html. (дата звернення: 20.04.2025).

26. SpeedyBee ELRS 2G4 RX Nano 2.4GHz RX ExpressLRS Receiver Manual. [Електронний ресурс] – Режим доступу: https://manuals.plus/speedybee/elrs-2g4-rx-nano-2-4ghz-rx-expresslrs-receiver-manual (дата звернення: 20.04.2025).

27. Nevliudov, I., Tsymbal, O., Bronnikov, A., & Mordyk, O. (2020). Internet of things for robotic projects. innovative technologies and scientific solutions for industries, (3 (13)), 58–64. URL: https://journals.uran.ua/itssi/article/view/ITSSI.2020.13.058.
(дата звернення: 010.05.2025). DOI: https://doi.org/10.30837/ITSSI.2020.13.058

28. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 [Електронний ресурс] : монографія / І. Ш. Невлюдов, В. В. Євсєєв, А. О. Андрусевич, С. С. Максимова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2023. – 321 с. (дата звернення: 1.05.2025).

29. Betaflight Flight Controller Firmware [Електронний ресурс] – Режим доступу: https://github.com/betaflight/betaflight.
(дата звернення: 15.05.2025).

30. Betaflight Configurator [Електронний ресурс] – Режим доступу: https://github.com/betaflight/betaflight-configurator
(дата звернення: 20.05.2025).

31. Репозиторій файлів конфігурації цілей (targets) Betaflight [Електронний ресурс] – Режим доступу: https://github.com/betaflight/config. – (дата звернення: 20.05.2025).

32. Розробка та збірка прошивки Betaflight. Документація для розробників [Електронний ресурс] – Режим доступу: https://betaflight.com/docs/development. (дата звернення: 21.05.2025).