

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Механіко-технологічний факультет

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
технічного сервісу та інженерного
менеджменту імені
М. П. Момотенка

_____ Іван РОГОВСЬКИЙ

« _____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему «Розробка технології технічного обслуговування агродронів» _____
Спеціальність 208 «Агроінженерія»

Гарант освітньої програми

_____ Д.Т.Н., професор
(науковий ступінь та вчене звання)

_____ (підпис)

_____ Вячеслав БРАТІШКО
(ПІБ)

Керівник бакалаврської кваліфікаційної роботи

_____ К.Т.Н., доц.

_____ Євтушенко Володимир Дмитрович

Виконав

_____ Довбиш Артем Іванович

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Механіко-технологічний факультет

ЗАТВЕРДЖУЮ

Завідувач кафедри

технічного сервісу та інженерного
менеджменту імені М. П. Момотенка

Іван РОГОВСЬКИЙ

(підпис)

(ПІБ)

« ____ » _____ 2025 р.

ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи студенту

Довбишу Артему Івановичу

(прізвище, ім'я, по батькові)

Спеціальність 208 «Агроінженерія»

(код і назва)

Тема дипломного проєкту бакалавра «Розробка технології технічного обслуговування агродронів» затверджена наказом ректора НУБіП України від «16» лютого 2026 р. № 510 «С»

Термін подання завершеної роботи на кафедру 20 травня 2026 року
(рік, місяць, число)

Вихідні дані до дипломного проєкту бакалавра Матеріали первинного бухгалтерського обліку, нормативно-методичні матеріали, науково-технічна література

Перелік питань, які потрібно розробити:

1. Характеристика господарства «Мрія»
2. Аналіз існуючих методів та засобів технічного обслуговування агродроонів.
3. Розрахунок основних параметрів обслуговуючої майстерні
4. Конструктивна частина
5. Економічне обґрунтування проєкту
6. Охорони праці

Перелік графічних документів: **електронна презентація на 14 слайдів**

Дата видачі завдання «06» січня 2026 р.

Керівник бакалаврської кваліфікаційної роботи _____ Євтушенко

Володимир Дмитрович

(підпис)

(прізвище та ініціали)

Завдання прийняв до виконання

Довбиш Артем Іванович

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота: 79 с., 30 рисунків, 2 таблиці, 27 джерел.

Об'єкт дослідження: агродрон.

Мета роботи: Створити відносно дешеву, працездатну технологію технічного обслуговування агродронів.

Методи: Використання апаратно-програмного забезпечення агродрона для віддаленого користування.

Ключові слова: Arduino, безпілотний літаючий апарат, апаратно-програмне забезпечення, мікроконтролер, технічне обслуговування.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1_ЗАГАЛЬНІ ВІДОМОСТІ	6
1.1. Аналітичний огляд чинних аналогів	6
1.2. Класифікація та різновиди мультикоптерів.....	7
1.3. Складова квадрокоптерів.....	12
1.4. Огляд основних компонентів	16
1.5. Огляд аналогів мікроконтролера	18
1.6. Огляд програмного забезпечення	19
РОЗДІЛ 2_ПРОЕКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АГРОДРОНІВ	21
2.1. Польотна математика квадрокоптера.....	21
РОЗДІЛ 3_РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АГРОДРОНІВ.....	34
3.1. Складання пристрою	34
3.2. Оцінка віброакустичних характеристик гвинтомоторної групи та теплового стану силових елементів (АКБ, ESC, двигуни).....	35
3.3. Розробка алгоритму та покрокової технологічної карти технічного діагностування агродрона.	36
3.4. Висновки до розділу 3.	37
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41
ДОДАТКИ	43

ВСТУП

Застосування безпілотних літальних апаратів (агродронів) – БПЛА, раніше, аж до кінця 20 століття, було виключно в військових цілях. Тільки 10 - 15 років тому, безпілотні апарати міцно почали входити в наш побут і отримали широке застосування в бізнесі, житті держави та населенні. Розглянемо деякі історичні факти зародження дронів [1].

Перша задокументована згадка про безпілотники, це застосування під час облоги Венеції 12 липня 1849 року австрійськими військами. Два запущених по місту аеростата з вибухівку. На цих аеростатах були відсутні пілоти, але були механізми які скинули бомби в потрібний час. У 1910 році американський військовий інженер Чарльз Кеттерінг, запропонував ідею безпіотної бомби з крилами, тим самим, заклав початок розвитку безпіотної ударної авіації [1].

На початку 30 років 20 століття технології побудови повітряних суден відкрили нові горизонти – впровадження автопілотів. У 1935 році англійські інженери створили БПЛА багаторазового використання, який отримує назву "QueenBee". Ця модель використовувалася як мішень в Королівському флоті Великобританії до 1947 року. Вже в середині 20 століття перебували люди, які вірили в те, що ці передові технології будуть використовуватися не тільки у військових але і в мирних цілях. Але роздуми про масове використання БПЛА в мирних цілях, були в категорії наукової фантастики, аж до теперішнього часу [2].

На даний час є велике різноманіття квадрокоптерів від різних виробників, але ціна у більшості дуже не мала.

В рамках даної роботи буде описано створення достатньо дешевого рішення, використовуючи Arduino.

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Аналітичний огляд чинних аналогів

У світі розвиваються технології з використанням дронів, які застосовуються в фактично во всіх сферах життєдіяльності [3].

Вони використовуються для аерофотозйомки, а саме дистанційного обліку тварин в мисливських господарствах, стану нафто– і газопроводу та електромереж, моніторингу лісних ресурсів та охоронюваних об'єктів, в координації пошуково рятувальних операцій. Впроваджуються технології точного землеробства, наприклад, для розрахунку кількості посівного насіння, обсягу польових робіт, вибірковість внесення добрив, виявлення шкідників. Також застосування дронів актуальне в екологічному моніторингу, це єдиний спосіб ефективно контролювати стан навколишнього середовища. Завдяки застосуванню БПЛА вирішуються кур'єрські завдання, наприклад доставка вантажу клієнтам, як в американському магазині Amazon. Одним з перспективних напрямків є створення мережі супутників і дронів, що утворюють інтернет покриття по всій поверхні землі [4].

БПЛА важко класифікувати так як вони мають різні характеристики. Виробники БПЛА поки ще не обмежені ніякими стандартами. З боку авіаційних регуляторів немає загальних вимог, якими повинен бути обладнаний дрон. Вони відзначаються габаритами, функціональністю, дальністю польоту, вагою, типу зльоту, корисним навантаженням, спеціалізацією, рівнем автономності і іншими характеристиками. Безпілотні літальні апарати поступово стають популярні та є головною продукцією багатьох авіаційних фірм. З'являється велика кількість розробників і виробників БПЛА, що займаються виключно безпілотними апаратами і системами. Це відбувається з ряду причин. Самі БПЛА набагато дешевше пілотованих літаків і вертольотів. Також дешевше ніж підготовка льотчика, обходиться і підготовка оператора безпілотної системи. Відсутність

пілота дозволяє виключити бортові системи життєзабезпечення, зменшити 8 масу і габарити БПЛА, а також збільшити діапазон допустимих перевантажень та можливостей [5].

Сучасні моделі дронів можуть мати як просту конструкцію, призначену в основному для розваг, так і складатися зі складних елементів, що дозволяють розширювати їх функціонал в залежності від їх застосування. Незважаючи на те, що внутрішній устрій дрона (мультикоптера) може відрізнятися, його базова складова завжди незмінна. У загальному вигляді кожному дрону можна виділити наступні частини: рама, польотний контролер, двигун, акумулятор, комунікаційний контролер, сенсори. Термін «безпілотний літальний апарат» включає як великі літаки, аналогічні за розміром і складністю пілотованому літаку так і невеликі електронні пристрої для персонального використання. Більшість невеликих дронів зазвичай поставляються, готовими до польоту. Їх використання широко, від розваг до різноманітних професійних напрямків [2].

Як і більшість мобільних пристроїв, дрони або мультикоптери живляться від акумуляторів. Вони поставляються з окремим контролером, який використовується як пульт управління дроном. Хоч і конструктори намагаються спроектувати міцні конструкції та розробити інтелектуальні системи, це не врятує дрона від недосвідченого оператора [1].

1.2. Класифікація та різновиди мультикоптерів

Мультикоптер є безпілотний літальний апарат вертолїтного типу, що має від двох та більше електродвигунів з повітряними гвинтами. У зв'язку з цим рівновага реактивних моментів досягається за рахунок попарно різноспрямованого обертання несучих гвинтів [21].

Підйомна аеродинамічна сила у мультикоптерів створюється, за рахунок обертювих лопатей несучих гвинтів, а не за рахунок крил. Крила або їх відсутність, відіграють допоміжну роль. Зависання в точці і висока маневреність є очевидними перевагами БПЛА вертолїтного типу. Також 9 однією з переваг

мультикоптерів є відсутність спеціально обладнаних аеродромів для їх запусків. Класифікація мультикоптерів по тактико–технічним характеристикам є дуже важливою. Вона дуже сильно спрощує роботу в практично во всіх галузях. За призначення мультикоптери поділяються :

- аматорські, призначені для фото та відеозйомки;
- для бізнесу, доставка пошти або різних товарів;
- для наукових досліджень, спостереження за різними об'єктами або збір інформації;
- наземні (в атмосфері), використання для астрономічних або метеорологічних спостережень;
- для силових структур або військових, патруль прикордонної смуги. В залежності від маси та дальності застосування. У цій класифікації безпілотні апарати коливуються в розмірах від мікрокоптеров менше 10 кг, які знаходяться поблизу від оператора, до важких, здатних підніматися на 20 км і бути в повітрі 24 г:
- мікро– і міні– мультикоптери мають вагу менше 4 кг і здатні віддалятися від оператора на відстань до 30–40 км;
- легкі(або малі) мають вагу до 100 кг і здатні відлітати на відстань до 120 км; -
- середні мають вагу до 310 кг і здатні перебувати на відстані до 1100 км від оператора;
- середньотяжкі мультикоптери мають вагу до 500 кг і здатні відлітати на відстань до 2000 км;
- важкі квадрокоптера мають вагу від 500 кг і здатні віддалятися від оператора на відстань до 3000 км, при цьому вони можуть перебувати в повітрі досить тривалий час.

Класифікація Мультикоптер в залежності від розміру рами:

- мікро– і міні–клас, мають довжину рами менш 250 мм і застосовуються в основному для аматорських цілей; 10 - від 250 до 350 мм, можуть бути обладнані фото– або відеокамерою і здатні розвивати чималу швидкість;
- від 350 до 450 мм, здатні піднімати вагу обладнання, достатній для здійснення прямих відеотрансляцій;
- від 450 до 550 мм, часто застосовуються для здійснення бізнес–діяльності і доставки різних товарів;
- від 550 до 750 мм, застосовуються в основному для здійснення наукової діяльності, так як здатні нести на борту достатню кількість дослідницького обладнання;
- понад 750 мм, входять в розділ важких мульткоптерів і здатні здійснювати військову діяльність.

Класифікація Мультікоптер в залежності від кількості двигунів. У мультікоптера може бути непарна або парна кількість гвинтів постійного кроку. Кожен гвинт працює від власного двигуна. Як правило, половина гвинтів обертається за годинниковою стрілкою, а друга частина – проти годинникової стрілки. Гвинтів може бути по одному на двигун або коаксіальною (соосною) по два [6].

1) Бікоптери і трікоптери. Вони мають відповідно два (рис 1.1) і три (рис 1.2) мотора, які обертаються в різних напрямках для компенсації обертального моменту. Їх загальною перевагою є відносна простота конструкції, дешевизна, а також висока енергоефективність. Конструкція бікоптера забезпечує найменший рівень шуму в польоті. В свою чергу конструкція рами трікоптера надає апарату здатність поєднувати переваги гвинтокрила і літака, що позначається на його покращених польотних характеристиках, на відміну від квадрокоптера. В основному всі вони знайшли застосування у спортсменів моделістів [6].



Рисунок 1.1 – Вигляд квадрокоптеру

2) Квадрокоптер – чотирьохроторний безпілотник, який має гарні показники автономності а також здатен нести вантаж. Завдяки конструкції дрон менш схильний до вібрації і популярний в якості платформи для зйомки з повітря. Основний недолік такий, як і у попередніх – відмова одного двигуна призводить до падіння. Квадрокоптер є оптимальним рішенням для використання в прикладних цілях, завдяки його невисокій ціні і надійності [6].



Рисунок 1.2 – Вигляд квадрокоптеру

3) Гексакоптери та Октокоптери.– апарати з шістьма та вісьма двигунами – істотно надійніші і дорожчі. Вони одні з найбільш просунутих рішень, на ринку мультикоптерів. Їх великою перевагою є – відмовостійкість, у разі поломки вони здатні здійснити аварійне планування на уцілілих двигунах. Це дорогі, але надійні апарати, яким довіряють цінний вантаж, наприклад, професійну відео– та фототехніку [6].



Рисунок 1.3 – Вигляд октокоптеру та гексакоптеру

4) Многороторний мультикоптер на базі більш ніж восьми моторів створюються для підвищення надійності, маневреності, стабілізації. Літальні апарати, ціна яких істотно вище в порівнянні з розглянутими раніше, призначені для виконання спеціальних завдань [6].



Рисунок 1.4 – Вигляд многороторного мультикоптера

1.3. Складова квадрокоптерів

У теперішній час, моделі дронів можуть мати як різні конструкції (проста конструкція призначена в основному для розваг) так і складатися зі складних складових елементів, що дозволяють перетворити ці літальні апарати посправжньому в вузькоспеціальні пристрої [7].

Внутрішній устрій квадрокоптера може відрізнятися від моделі до моделі, не дивлячись на це, його базові складові завжди незмінні. У загальному вигляді кожному дрону можна виділити наступні частини:

1. Рама
2. Двигуни
3. Регулятори обертів
4. Пропелери
5. Польотний контролер

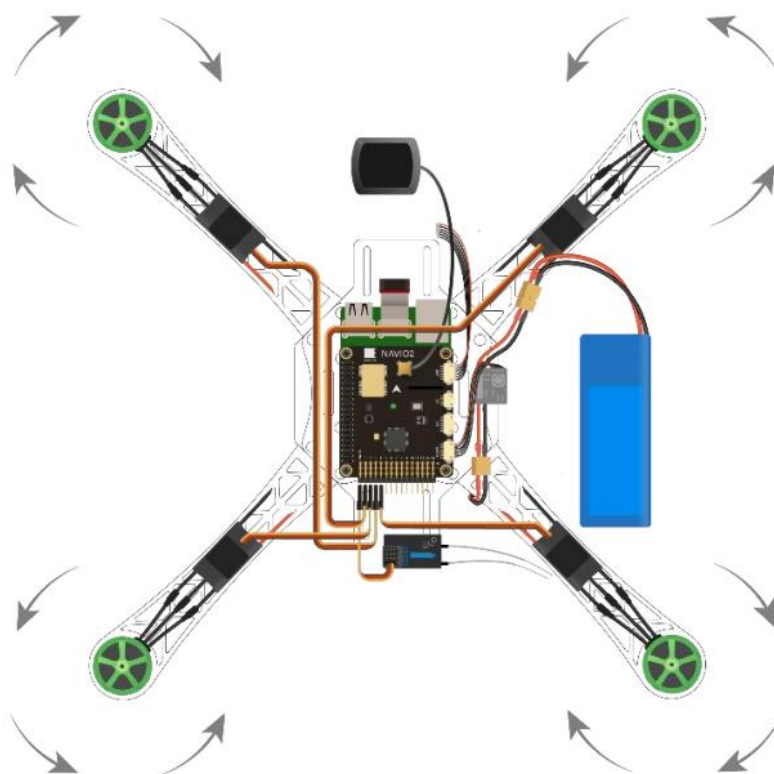


Рисунок 1.5 – Типовий устрій квадрокоптера

Рама. Основою дрона є рама, до якої кріпляться всі складові його елементи. Головне завдання при виготовленні рами, створити пристрій який буде легким, ударостійким, довговічним і також ергономічним. Застосування надлегких матеріалів, дозволяє апарату за невеликий проміжок часу розвивати свою максимальну швидкість польоту. Для того щоб рама була легкою і міцною використовують полімери або легкі сплави. Щоб забезпечити максимальну жорсткість рами, використовують карбон, скловолокно та інші матеріали. Найлегше добитися більшої маневреності і легкості апарату – використовувати раму з багатьох складових деталей. Для зручності монтажу електропроводки, є конструктивні отвори, які з'єднують пролітний контролер з усіма іншими частинами дрона [7].



Рисунок 1.6 – Вигляд типової рами

Двигуни та пропелери. Квадрокоптер має чотири двигуни з пропелерами, які надають рух зі швидкістю, що генерується регуляторами обертів. Залежно від вступників від польотного контролера команд, відбувається регулювання обертів двигуна [7].



Рисунок 1.7 – Вигляд двигуна з пропелером

Найбільш швидкозношуваними частинами дрона є пропелери і двигуни, оскільки при його польоті основне навантаження лягає саме на них. Найчастіше, при аварійних ситуаціях першими ламаються пропелери, які являються відкритими частинами конструкції. Ця проблема вирішується установкою захисних дуг, або кожухів [7].

«Мозок» дрона – це польотний контролер. Завдяки встановленим датчикам він здатний розрізняти сигнали, що надходять з дистанційного пульта оператора. Універсальність дрона характеризується кількістю оброблюваних їм сигналів. Кожен з чотирьох двигунів, з'єднаний шлейфом з контролером, що дозволяє подавати на них сигнали (запрограмовані команди) [7].

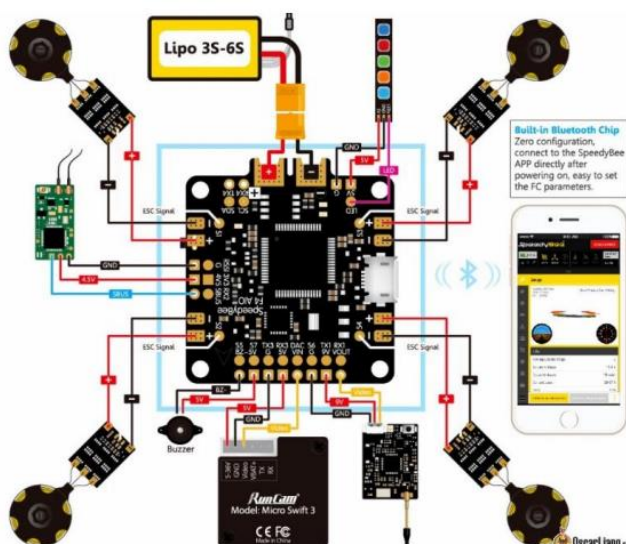


Рисунок 1.8 – Вигляд контролера

Для стабільності роботи польотного контролера, при створенні дронів застосовуються методи, які дозволяють максимально віброізолювати елементи

управління, від яких залежить точність управління польоту. Тому в теперішній час, випускаються дрони у яких достатня віброізоляція. Як правило, чим краще віброізолюван контролер, тим стабільніше буде літати дрон. Польотний контролер складається з різних наборів датчиків (GPS, лазерного далекоміра, барометра, гіроскопу, акселерометра, лідара та інші), які передають йому свої показання. Завдяки передавачу установленому на корпусі, здійснюється зворотний зв'язок. Оператор задає параметри польоту дрона в залежності від показань різних датчиків. Ціна контролера залежить від якості комплектуючих і програмних алгоритмів. Наприклад, контролер може самостійно змінити параметри польоту безпілотної, не вдаючись при цьому до допомоги оператора [8].

Основними технічними параметрами квадрокоптерів є:

1) Габарити. Довжина і висота виробу. Впливає в основному на розмір двигунів і гвинтів, що в свою чергу впливає на максимальну швидкість, швидкість підйому і зниження апарату [8];

2) Акумулятор. Вимірюється ємність батареї в міліампер на годину. Чим показник вище, тим довше квадрокоптер літає на одному заряді [8];

3) Радіус дії. Діаметр, в межах якого зможе літати квадрокоптер [8];

4) Частота передавача / приймача [8].

Впливає на якість прийому і передачі сигналу:

- 900 Mhz. Плюс цієї частоти – сигнал з легкістю проходить крізь дерева, поєднується з апаратурою, що працює на 2.4 Ghz [8];

- 1.2 Ghz. Працюючи на цій частоті дрони, долають довгі дистанції. Сигнал легко проходить через тверді предмети [8];

- 2.4 Ghz. Сигнал гірше проходить через тверді предмети, але дрон відлітає на цій частоті на далекі дистанції [8];

5) Вага. Впливає на керованість, тобто здатність чинити опір вітру. Напрямку залежить від габаритів конструкції [8];

6) Розширення камери. Впливає на деталізацію картинки. При низькій роздільній здатності не вийде схопити багато об'єктів і розгорнути їх на широкоформатному екрані. Іноді якість зйомки у камер, що знімають в низькій роздільній здатності, краще. До базових факторів, що впливають на якість відео, відносять: оптику, оптичну стабілізацію, розмір матриці. Різноманітність характеристик та класифікацій впливає на складність керування дронів. Для найбільш вдалого тренування керуванням БПЛА існують тренажери–симулятори [8].

1.4. Огляд основних компонентів

Основним компонентом приладу є його керуюча плата. В даній роботі використовується програмний мікроконтролер Arduino Mega представлена на рисунку 1.9 [15].

Arduino Mega побудована на мікроконтролері ATmega2560. Плата має 54 цифрових входів/виходів (14 з яких можуть використовуватися як виходи ШІМ), 16 аналогових входів, 4 послідовних порти UART, кварцовий генератор 16 МГц, USB коннектор, роз'єм живлення, роз'єм ICSP і кнопка перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC, або акумуляторною батареєю. Arduino Mega 2560 сумісна з усіма платами розширення, розробленими для платформ Uno або Duemilanove [15].

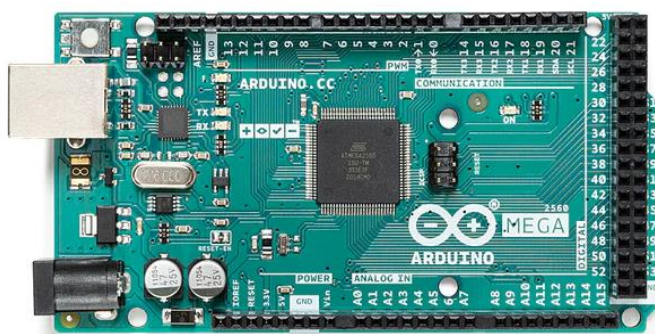


Рисунок 1.9 – Вигляд мікроконтролера

Далі розглянемо датчики. Для побудови будуть використовуватися датчик гіроскопу моделі ITG3200 та акселерометр моделі BMA180, представлені на рисунках 1.10 та 1.11 відповідно. Гіроскоп – пристрій, що здатний реагувати на зміну орієнтації основи, на якій його встановлено.

Акселерометр — прилад для вимірювання сили реакції індукованої прискоренням або гравітацією. Одно- та багато-вісні моделі можуть визначати величину та напрям прискорення у вигляді векторної величини і тому можуть бути використані для визначення орієнтації, вібрації й ударів [23].

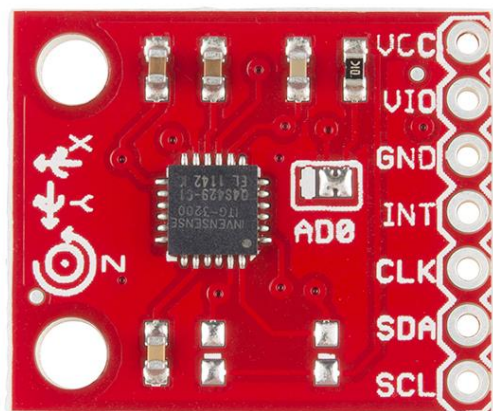


Рисунок 1.10 – Вигляд гіроскопу

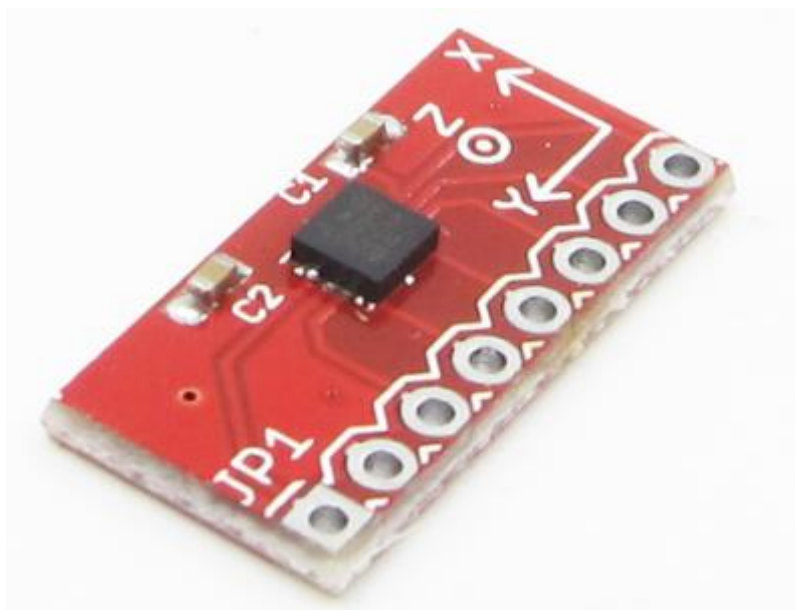


Рисунок 1.11 – Вигляд акселерометру

Інші елементи розглянемо в сукупності. В роботі використовуються 4 двигуна моделі 2212 KV1400 та пропелери 1047 для них. Також використовуюється контролер регуляторів швидкості Brushless Speed Controller та джерело живлення моделі Turnigy 2200mAh 3S 30C Lipo. В якості апаратури керування використовується пульт моделі Hobby King 2.4Ghz 4Ch Tx & Rx V2. Всі ці компоненти кріпляться на рамі моделі F450 Frame FPV.

Використовуються дані компоненти через їх вартість та якість. Сам контролер обраний Arduino, через вартість відкритість та легкість налаштування.

1.5. Огляд аналогів мікроконтролера

1.5.1 Raspberry Pi

Raspberry Pi — одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Головне призначення — сприяти вивченню базових комп'ютерних навичок школярами представлено на рисунку 1.12. У жовтні 2013 розробники оголосили про продаж більше двох мільйонів цих плат, що закріпило за Raspberry Pi звання найпопулярнішої платформи для ентузіастів [24].

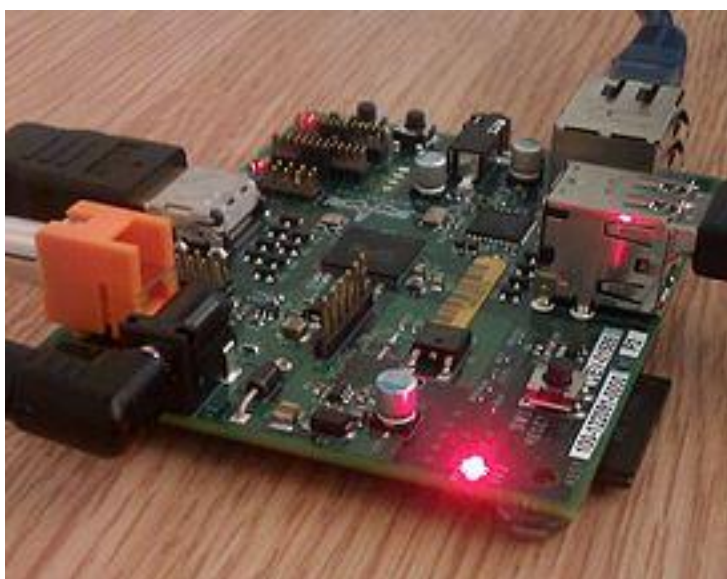


Рисунок 1.12 – Вигляд плати Raspberry Pi

1.5.2 Intel Galileo.

Intel Galileo - це перша в лінійці сертифікованих Arduino Development плат на основі архітектури Intel X86, розроблена для виробників та освіти. Intel випустив дві версії Galileo, які називаються Gen 1 та Gen 2. Ці плати іноді називають "платами прориву" представлено на рисунку 1.13 [25].



Рисунок 1.13 – Вигляд плати Intel Galileo

1.6. Огляд програмного забезпечення

Інтегроване середовище розробки Arduino це багатоплатформовий додаток на Java, що включає в себе редактор коду, компілятор і модуль передачі прошивки в плату. Середовище розробки засноване на мові програмування Processing та спроектоване для програмування новачками, не знайомими близько з розробкою програмного забезпечення. Мова програмування аналогічна мові Wiring. Загалом, це C++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюються за допомогою AVR-GCC [27].

Програми Arduino пишуться на мові програмування C або C++. Середовище розробки Arduino поставляється разом із бібліотекою програм «Wiring» (бере

початок від проекту Wiring, який дозволяє робити багато стандартних операцій вводу/виводу набагато простіше). Користувачам необхідно визначити лише дві функції для того, щоб створити програму, яка буде працювати за принципом циклічного виконання [27]:

- `setup()`: функція виконується лише раз при старті програми і дозволяє задати початкові параметри
- `loop()`: функція виконується періодично, доки плата не буде вимкнена

Також використовується програмне забезпечення ArduPilot. Це набір програмного забезпечення, що складається з навігаційного програмного забезпечення (зазвичай називається вбудованим програмним забезпеченням, якщо воно скомпільоване в двійкову форму для цілей апаратного забезпечення мікроконтролера), що працює на транспортному засобі (вертоліт, літак, марсохід, AntennaTracker або Sub), а також програмне забезпечення для керування наземною станцією включаючи Mission Planner, APM Planner, QGroundControl, MavProху, Tower та інші. Вихідний код ArduPilot зберігається та керується на GitHub майже 400 учасниками [26].

Набір програмного забезпечення автоматично створюється щовечора з безперервною інтеграцією та модульним тестуванням, наданим Travis CI, а також середовищем збирання та компіляції, включаючи кросплатформний компілятор GNU і Waf. Попередньо скомпільовані двійкові файли, що працюють на різних апаратних платформах, доступні для завантаження користувачем із суб-сайтів ArduPilot [26].

РОЗДІЛ 2

ПРОЕКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АГРОДРОНА

2.1. Польотна математика агродрона

Квадрокоптер складається з двох пар обертових роторів і гвинтів, розташованих у вершині квадратної рами. Він здатний виконувати вертикальний зліт і посадку (VTOL), подібно до типових гвинтокрилам. Однак система управління між гвинтокрилами та квадрокоптерами суттєво відрізняється від одного до іншого за рахунок динаміки польоту відповідно [22].

Використовуючи цю систему координат в якості основи нашого квадрокоптера, як показано на рисунку 2.1, двигуни фізичний розподіл двигунів був прикріплений за принципом «×» конфігурації. Весь квадрокоптер складається з двох двигунів, що обертаються за годинниковою стрілкою, і двох двигунів, що обертаються проти годинникової стрілки, на вершині квадратного каркаса, як видно на рисунку 2.1 [22].

Рух у просторі досягається варіацією кінцевої сили і крутного моменту на кожній осі. Для досягнення правильної зміни крутного моменту та сили в системі, мотори налаштовані на роботу в парах, як показано в таблиці 2.2 M1, M2, M3 і M4 відповідно вважаються Мотор – 1, Мотор – 2, Мотор – 3 і Мотор – 4 [22].

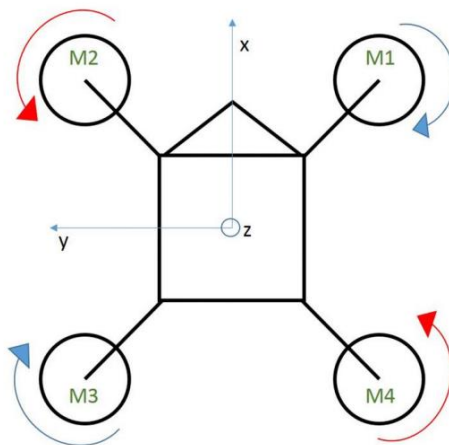


Рисунок 2.1 – Складова агродрона

Таблиця 2.1 – Парна робота двигунів

	Газ				Крен				Тангаж				Рискання			
Мотори	1	2	3	4	1	2	3	3	1	2	3	4	1	2	3	4

Поступальний рух квадрокоптера вимагає нахилу платформи до потрібної осі. Опорний кут квадрокоптера задається креном ϕ , тангажем θ , рисканням ψ умовами руху в трьох вимірах (3D), див. Рис. 2.2 [9].

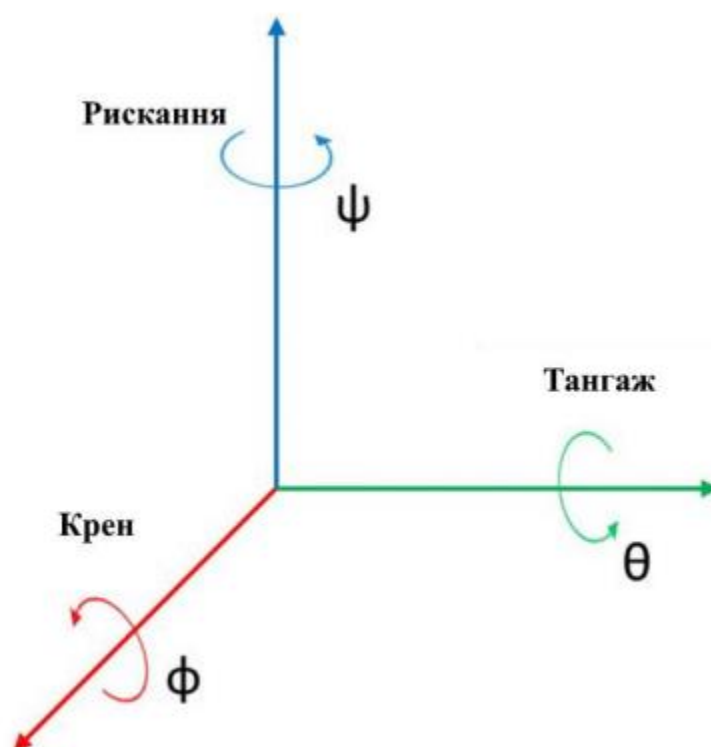


Рисунок 2.2 – Позначення для руху в 3D

Виходячи з розташування моторів, описаного в табл. 2.1, просто потрібно змінити швидкість руху одной з пар двигунів, щоб викликати рух у шести ступенях свободи (DOF). Це причина, що дозволяє квадрокоптеру рухатися в шести DOF і керувати, використовуючи лише чотирима входами [9].

2.1.1 Пересування дрона

Динаміка польоту квадрокоптера заснована на співвідношенні між кутовою швидкістю двигунів, тобто силою підйому (Ft) і кінцевим крутним моментом (τt), виробленою двигунами і прикладеною до квадрокоптера [22].

Чотири рухи досягаються при зміні Ft і τt :

1. По вертикалі 'z' Він утворюється шляхом підсумовування всіх сил ротора на осі z.

2. Тангаж ' θ ' Цей рух досягається маніпулюванням остаточною крутним моментом навколо осі y. Це робиться за допомогою зміни швидкості передніх роторів проти задніх роторів (збільшення або зменшення швидкості відповідно до бажаного руху) [22].

3. Крен ' ϕ ' Принцип цього руху такий самий, як і для тангажа, але нова опорна вісь – x. Лівий і правий ротори забезпечують рух з креном [22].

4. Рискання ' ψ ' Рух рискання створюється реактивним моментом роторів. Цей рух вводиться різними швидкостями обертання, застосованими до пари двигунів проти обертання. Це викликає різницю крутного моменту між ними і надає остаточною крутний момент навколо осі z [22].

2.1.2 Математичне моделювання

Зазвичай для розробки математичної моделі більшості мобільних роботів використовуються два підходи [10]:

- 1) Рівняння Ньютона – Ейлера.
- 2) Рівняння Лангранжа.

У цій роботі було зосереджено на класичному математичну модель, заданому законами Ньютона; тому ми розробляємо нашу модель на основі рівнянь Ньютона – Ейлера. Виходячи з цих теорій, квадрокоптер повинен бути описаний як жорстке тіло, яке рухається через різні системи відліку на основі нерухомої системи координат (рисунк 2.3). Вони допомагають нам описати положення орієнтації літаючої платформи в просторі [10].

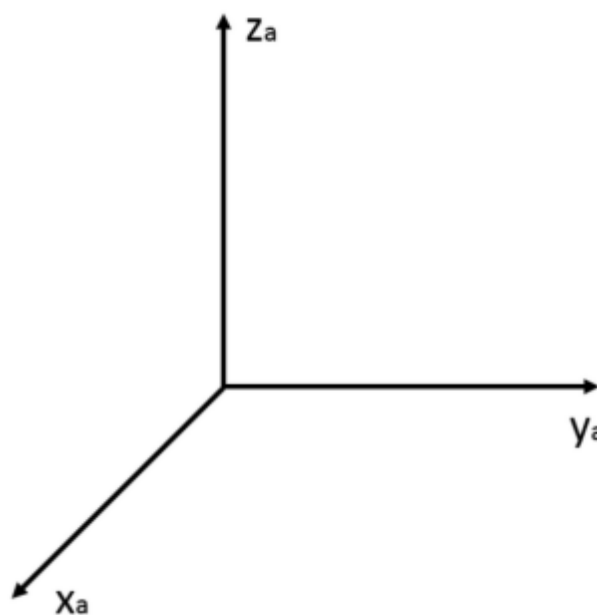


Рисунок 2.3 – Базова система координат

2.1.3 Координатна система

По–перше, слід розуміти, як рухається квадрокоптер у просторі (див. пункт 2.1.1) і які переміщення пов'язані з кожною еталонною системою координат. Вони допомагають нам зрозуміти, як об'єкти рухаються в просторі і як їх математично представляти. Далі представлені координатні системи, що описують динаміку квадрокоптера [10].

- Інерційна система $F i$: нерухома система координат.
- Система апарата $F v$: Початок координат є центром маси квадрокоптера і воно вирівняне з $F i$.
- Система тіла $F b$: $F v$ повертається на ϕ +, див. рівняння 2.4. Вектори на системі апарата $F v$ описуються матрицею, наведеною у рівнянні (2.1.)

$$F^v(1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

Вектори системи координат апарату відповідають матриці $F v$ (див. Рис. 2.3), вважаються основою для аналізу наших контрольних системах. Початкова частина цього аналізу полягає в обертанні векторної системи ' $F v$ ' 24 стосовно якоїсь конкретної осі, щоб створити наші математичні посилення для аналізу рухів у 3D–просторі [10].

- Система апарату–1 $F v1$: Базується на $F v$, але повертається у $\psi +$.

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

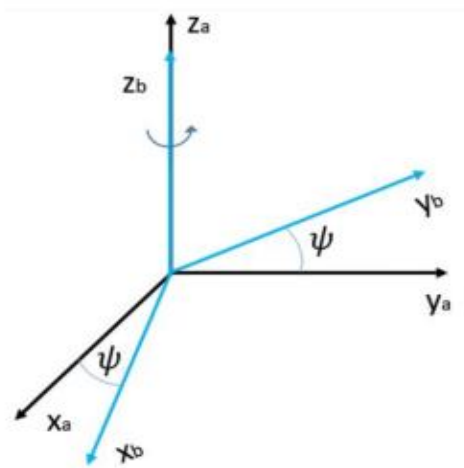


Рисунок 2.4 – Система апарату-1

- Система апарату–2 $F v2$: Базується на $F v1$, але обертається у $\theta +$.

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (2.3)$$

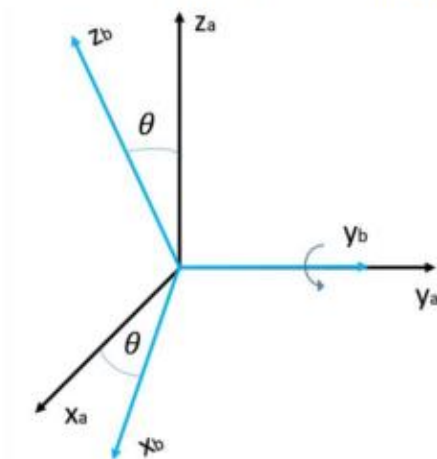


Рисунок 2.5 – Система апарату–2

- Система тіла Fb : заснований на $F v2$, але обертається у $\phi +$.

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (2.4)$$

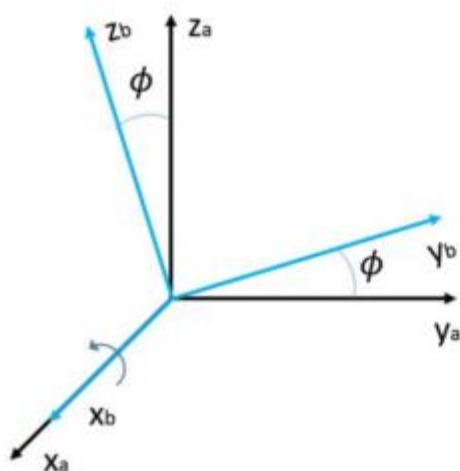


Рисунок 2.6 – Система апарату–1

Після того, як координатні системи були визначені, можна зобразити рухи від одної системи до іншої за допомогою двох операцій: обертання та трансляції [11].

2.1.4 Кінематичний та динамічний аналіз

Першим кроком є визначення лінійної швидкості в координатній системі апарату, як показано у рівнянні. 2.5.

$$\frac{d}{dt} \begin{pmatrix} x^i \\ y^i \\ z^i \end{pmatrix} = R_b^v \begin{pmatrix} v_x^b \\ v_z^b \\ v_z^b \end{pmatrix} = \begin{pmatrix} v_x^i \\ v_z^i \\ v_z^i \end{pmatrix} \quad (2.5)$$

По-друге, ми визначаємо кутову швидкість типу $wb = (p \ q \ r) T$ і вважаємо, що діапазон виведення кута буде меншим або рівним ± 20 градусів; отже, можна використовувати критерії наближення малого кута. Поєднуючи визначення описаної вище кутової швидкості та поняття, можна отримати кутову швидкість, виконавши кроки, задані рівняннями. 2.6, 2.7).

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = R_{v2}^b(\phi) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{v2}^b(\phi) R_{v1}^{v2}(\dot{\theta}) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{v2}^b(\phi) R_{v1}^{v2}(\dot{\theta}) R_{vv}^{v1}(\dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \quad (2.6)$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (2.7)$$

2.1.5 Динаміка твердого тіла

Другий закон Ньютона визначається рівнянням (2.8) для поступального руху.

$$m \frac{d^2 x}{dt^2} = m \frac{dv}{dt_i} = m a = F \quad (2.8)$$

Застосування ефекту Коріоліса до рівняння (2.9), отримуємо:

$$m \frac{dv}{dt_i} = m \left(\frac{dv}{dt_i} + w_{b/i} \times v \right) = F \quad (2.9)$$

Для обчислення кутового прискорення в системі координат тіла необхідно застосувати 2-й закон Ньютона для обертального руху і включити ефект Коріоліса.

$$\frac{dH^b}{dt_i} = \frac{dH^b}{dt_i} + w_{b/i} \times H^b = \sum \tau^b \quad (2.10)$$

Для обчислення інертів вважається ідеальною моделлю, як спостерігається на рис. 2.8. Квадрокоптер передбачається сферичним щільним центром з масою M і радіусом R . Мотори моделюються як чотири точки мас, розташованих на відстані l від центру масою m . Якщо ці параметри зрозумілі, тепер можна обчислити відповідну інерцію всієї системи (рис. 2.7).

$$J_x = J_y = \frac{2MR^2}{5} + 2ml^2 \quad (2.11)$$

$$J_z = \frac{2MR^2}{5} + 4ml^2 \quad (2.12)$$

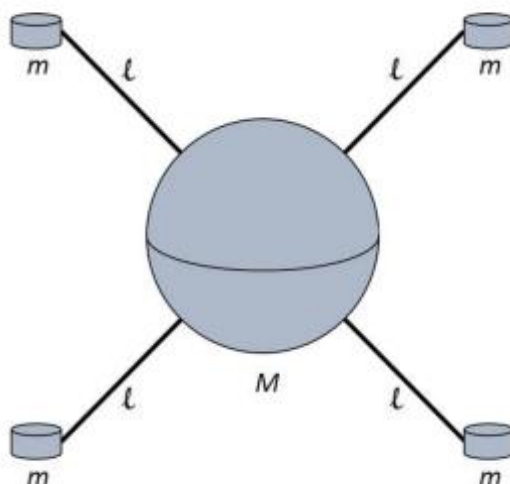


Рисунок 2.7 – Ідеальна модель квадрокоптера

2.1.6 Сили та моменти

Сили та моменти на квадрокоптері, описані на рисунку 2.8, зумовлені, в першу чергу, силою тяжіння та чотирма гвинтами.

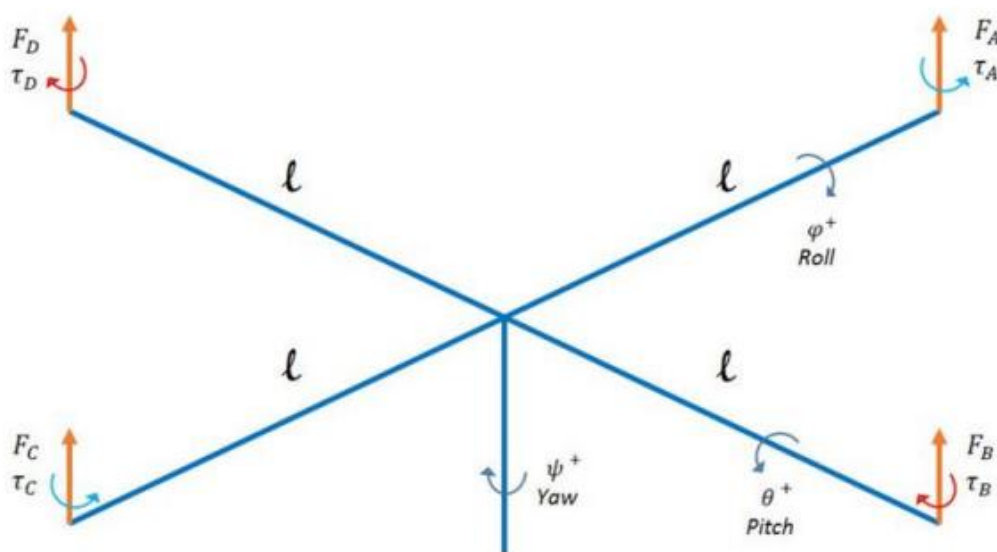


Рисунок 2.8 – Сили та момент на квадрокоптері (ідеальні)

Кожен двигун створює силу вгору F і крутний момент τ . Таким чином, загальна сила квадрокоптера задається сумою всіх сил ($F_t = \sum_{i=1}^4 F_i$). Цей же принцип застосовується і для крутного моменту ($\tau_t = \sum_{i=1}^4 \tau_i$). Враховуючи це, визначаються крутні моменти кочення, розбивання та похитування.

$$\begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} l(F_d - F_b) \\ l(F_d - F_b) \\ (\tau_b + \tau_d) - (\tau_a + \tau_c) \end{pmatrix} \quad (2.13)$$

Сила тяжіння також чинить силу на систему координат транспортного засобу F_v квадрокоптера, і вона просто впливає на напрямок z . Однак, щоб співставити цю силу з еталоном, наведеним на корпусі F_b , необхідно помножити на відповідну матрицю обертання.

$$F_g^v = R_v^b F_g^b \rightarrow F_g^v = \begin{pmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\theta) \cos(\phi) \end{pmatrix} \quad (2.14)$$

2.1.7 Дизайн системи управління PID

Для систем контролера квадрокоптера важливо врахувати, наскільки міцною повинна бути система щодо зовнішніх сил, таких як вітер або зіткнення. Зазвичай сила, викликана вітром, може бути постійною або змінною (поступова чи ні). З цієї причини важливо врахувати, наскільки швидкою є реакція квадрокоптера щодо зовнішніх сил, що діють за короткий проміжок часу [17].

Перший контролер відповідає за стабілізацію кута. Ідея полягає в тому, щоб мати контролер якомога швидше і стабільніше. Для цього реалізовано контролер P [17].

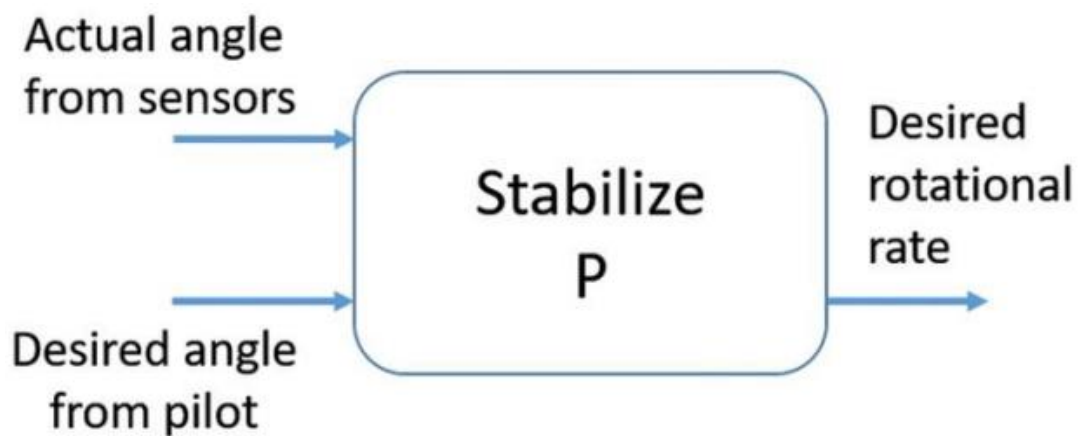


Рисунок 2.9 – Контролер P

Опорний сигнал для контролера подається пілотом (або функцією) і вважається бажаним кутом руху на основі системи координат квадрокоптера. Вхід зворотного зв'язку задається датчиками, у цьому випадку значенням 29 зворотного зв'язку вхідного сигналу є фактичний кут квадрокоптера. Вихід буде розглядатися як бажана швидкість обертання, яка буде входом для наступної частини керуючої системи [17].

Наступний крок – зробити систему стабільною проти зовнішніх сил. З цієї причини розглядаються зміни швидкості обертання квадрокоптера з метою компенсації небажаних змін (поступових чи ні). PID-контролер вибирається завдяки перевагам, які задаються інтегральною та похідною частинами, та простоті в майбутній реалізації коду. Введений вхід цього контролера – це прямий вихід контролера стабільності [17].

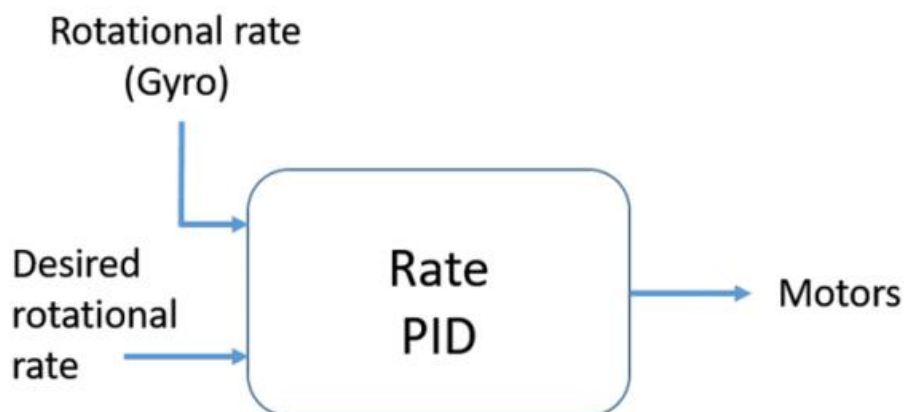


Рисунок 2.10 – Контролер PID

Мета – повернути квадрокоптер з певною швидкістю (заданою градусами в секунду) у будь-якому з можливих напрямків (котиться ϕ , крок θ , позіхає ψ). Вхід зворотного зв'язку подається безпосередньо з датчиків. Ця реалізація вимагає, щоб введення зворотного зв'язку було в градусах за секунду, щоб відповідати бажаній частоті обертання, яка має цю одиницю. Вихід контролера направляється безпосередньо на двигуни. Весь контролер, рис. 2.11, – це склад регулятора стійкості та швидкості в каскадному режимі (одна гілка на кут). Ідея полягає в тому, що коли перший контролер досягне потрібного кута, його вихід буде нульовим; таким чином, вхід для контролера швидкості буде дорівнює

нулю, і квадрокоптер не матиме обертального руху. З іншого боку, коли різниця першого контролера по 30 відношенню до керуючого сигналу не дорівнює нулю, другий контролер матиме ненульовий вхід, а квадрокоптер матиме обертальний рух [17].

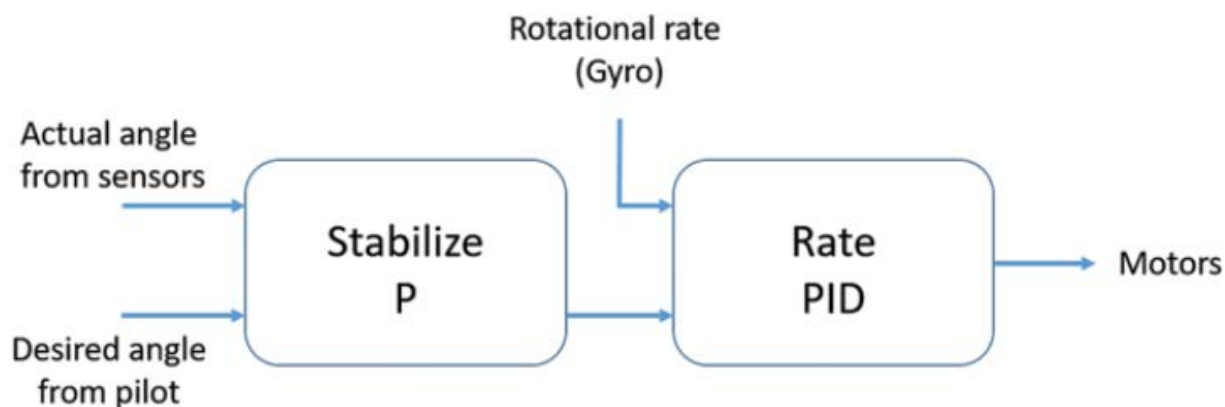


Рисунок 2.11 – Зв'язок контролерів P та PID

Контролер стабілізації (рис. 2.9) вимагає реалізації P-контролера, тоді як регулятор швидкості (рис. 2.10) вимагає PID-контролера. Або P, або PID-контролер, обидва мають однакову структуру коду. Різниця задається параметрами контролера, де k_p , k_i і k_d визначають тип контролера (P-PID-PID) [17].

2.1.8 Робота PID регулятора

PID – це функція в польотному контролері. Ця функція зчитує дані з датчиків і передає двигунів, як швидко їм потрібно обертатися. В кінцевому підсумку, саме так досягається стабільність і ідеальність польоту квадрокоптера. PID позначає похідну пропорційного інтеграла. ППД-регулятор являє собою замкнуту систему управління, яка намагається отримати фактичний результат ближче до бажаного результату, внівши корективи у вихідні дані, які відправляються двигунів. Якщо відбувається помилка, вона повертається в початок і цикл повторюється. ППД-регулятор обчислює значення «помилка» як різниця між вимірюваною величиною і бажаної величиною. Контролер намагається звести до мінімуму помилку, відрегулювати надходять значення

управління. Працює це так: в квадрокоптера PID отримує дані з датчиків і порівнює їх з поданими даними [17].

Різниця між цими дані називається «помилка» або «error» по-англійськи і намагається зменшити в подальшому цю помилку. В ПД-регулятор є три функції: P, I і D рисунок 2.10. Ці значення можуть бути інтерпретовані з точки зору часу:

- P обробляє помилку, яка відбувається в даний момент – чим далі вона від заданого значення, тим сильніше вона коригується
- D – коригує майбутні помилки, дивлячись на те, як швидко ви наближаєтеся до заданої точки та наскільки відбувається протидія P при наближенні до заданої точки
- I – Аналізує минулі помилки, які відбуваються протягом якогось часу (наприклад, якщо вісь постійно зміщується від заданого значення через вітер) і протидіє цій силі збільшуючи швидкість потрібних двигунів.

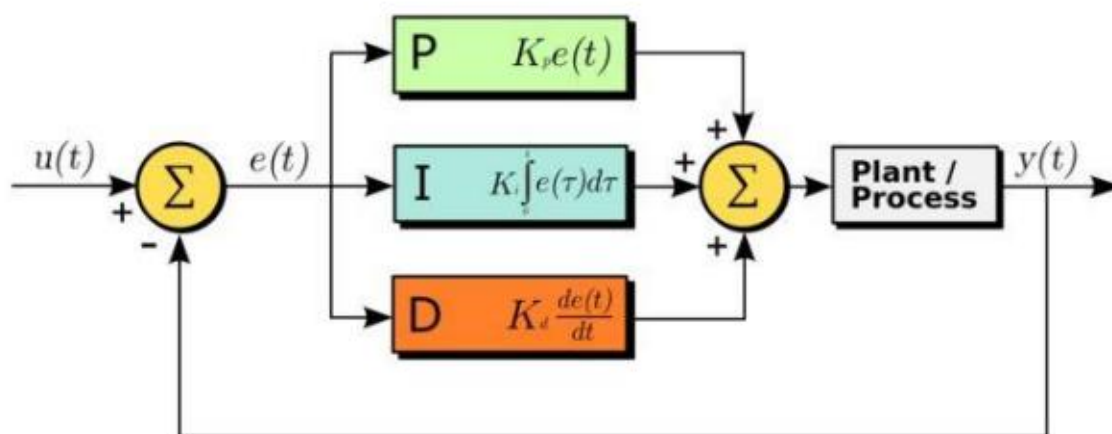


Рисунок 2.12 – Схема PID регулятора

Теоретичні методи аналізу системи з ПД-регулятором рідко застосовуються на практиці. Основна складність практичного застосування – незнання характеристик об'єкта управління. Крім того, істотну проблему представляють не лінійність і не стаціонарність системи. Практичні регулятори працюють в обмеженому зверху і знизу діапазоні, тому в принципі нелінійні. У

зв'язку з цим набули поширення методи експериментальної настройки регулятора, підключеного до об'єкта управління. Пряме використання формованої алгоритмом керуючої величини також має свою специфіку. Для ручної настройки PID – параметрів потрібно багато практики. Є різноманітні аналітичні методи їх обчислення і аналізу, але вони потребують кваліфікованої технічної підготовки та предметне знання великої кількості параметрів конкретної системи, яка настраюється. Різними дослідниками пропонується численний ряд емпіричних методів – середнє між аналітичним обчисленням та ручним підбором [17].

РОЗДІЛ 3

РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АГРОДРОНА

3.1. Складання пристрою

3.1.1 Збирання елементів на платі

Зібравши всі потрібні компоненти можна приступити до збирання. Спочатку зберемо мізки. Для зручності монтажу все, що треба можна розмістити на платі прототипування, як зображено на рисунку 3.1.

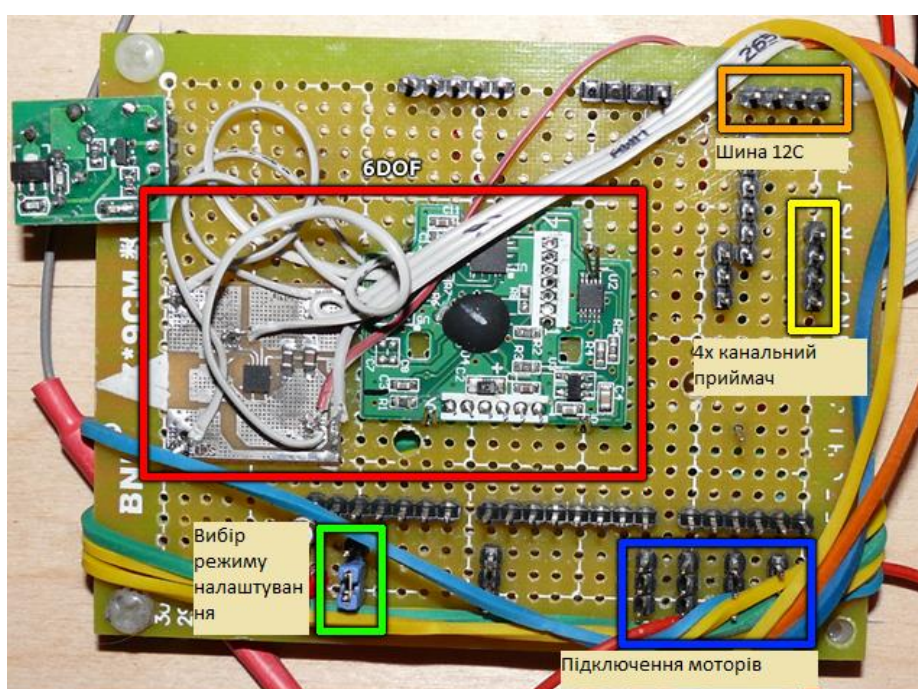


Рисунок 3.1. – Приклад зібраних компонентів на платі прототипування

Таблиця 3.1. - Розпинання підключення

Двигуни	
Передній лівий	2
Передній правий	5
Задній лівий	6
Задній правий	3
Приймач	
Throttle	A8

Продовження таблиці 3.1.

Roll	A9
Pitch	A10
Yaw	A11
Живлення мізків та приймача	
Arduino можна живити від контролера швидкості (якщо він має перетворювач живлення), підпаявши вихід потужності до USB та VIN, іншого варіанту - живити безпосередньо від акумулятора за допомогою спеціального роз'єму потужності на платі	
Записуємо приймач від Arduino через відповідні виходи +5 В та GND	
Інше	
Перехід до режиму командного рядка	A5 замкнути з GND
Світлодіоди	13,30,31

3.1.2 Налаштування прошивки та завантаження

Для налаштування потрібно виконати ряд пунктів:

- 1) Для початку можливо завантажити архів з прошивкою з сайту gscorp.com. Але будемо використовувати власно написану за допомогою інтегрованого середовища розробки Arduino. Частину розробленого програмного забезпечення можна переглянути у додатку А.
- 2) Тепер в папці `libraries/AP_ADC` в файлі `AP_ADC_ADS7844.cpp` потрібно обрати тип датчика: розкоментувати один із варіантів `#define FFIMU`, `#define ALLINONE` або одночасно `#define ALLINONE` та `#define BMA_020` для варіанта с датчиками ITG3200 и BMA020.
- 3) Якщо не має барометра, потрібно закоментувати код роботи з шиною I2C с цим датчиком в файлі `libraries/APM_BMP085/APM_BMP085.cpp`.
- 4) Відкрити через Arduino IDE любий з файлів в папці MegaPirate.

- 5) На вкладці APM_Config.h обрати тип коптера, для цього після `#define FRAME_CONFIG` вставимо один из варіантів: `QUAD_FRAME`, `TRI_FRAME`, `HEXA_FRAME`, `Y6_FRAME`, `OCTA_FRAME`, а також тип орієнтації рами `#define FRAME_ORIENTATION`, виставивши `PLUS_FRAME` або `X_FRAME`.
- 6) Якщо не має GPS, потрібно на початку змінити значення `#define GPS_PROTOCOL` на `GPS_PROTOCOL_NONE`, щоб не чекати 10-15 секунд ініціалізації GPS (коли його не має).
- 7) Тепер можемо підключити Arduino Mega та загрузити скетч в контролер.

3.1.3 Налаштування коптера

Після успішного завантаження скетча-прошивки в Arduino IDE завантажити програму налаштування та замкнути контакт A5 та GND. Після запуску програми в меню Options обираємо COM-порт Arduino як зображено на рисунку 3.2.

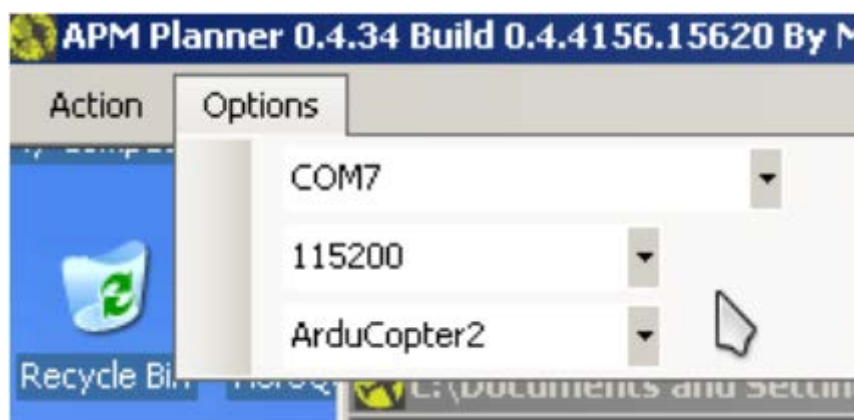


Рисунок 3.2. – Меню Options

В меню Action обираємо AC2 Setup як зображено на рисунку 3.3.



Рисунок 3.3. – Меню Actions

Після натискання AC2 Setup бачимо величезну кнопку, на яку потрібно натиснути і дотримуватися інструкцій, особливо важливо не пропустити перше віконце з купою тексту, в якому попросять рухати ручки пульта до їх мінімальних і максимальних значень, а також інше віконце, в якому попросять переконатися, що коптер стоїть рівно для калібрування датчиків.

Після завершення налаштування можна розмикати A5 з GND і в меню вибрати пункт AC2 Sensor, щоб перевірити адекватність роботи датчиків на вкладці Raw Sensor.

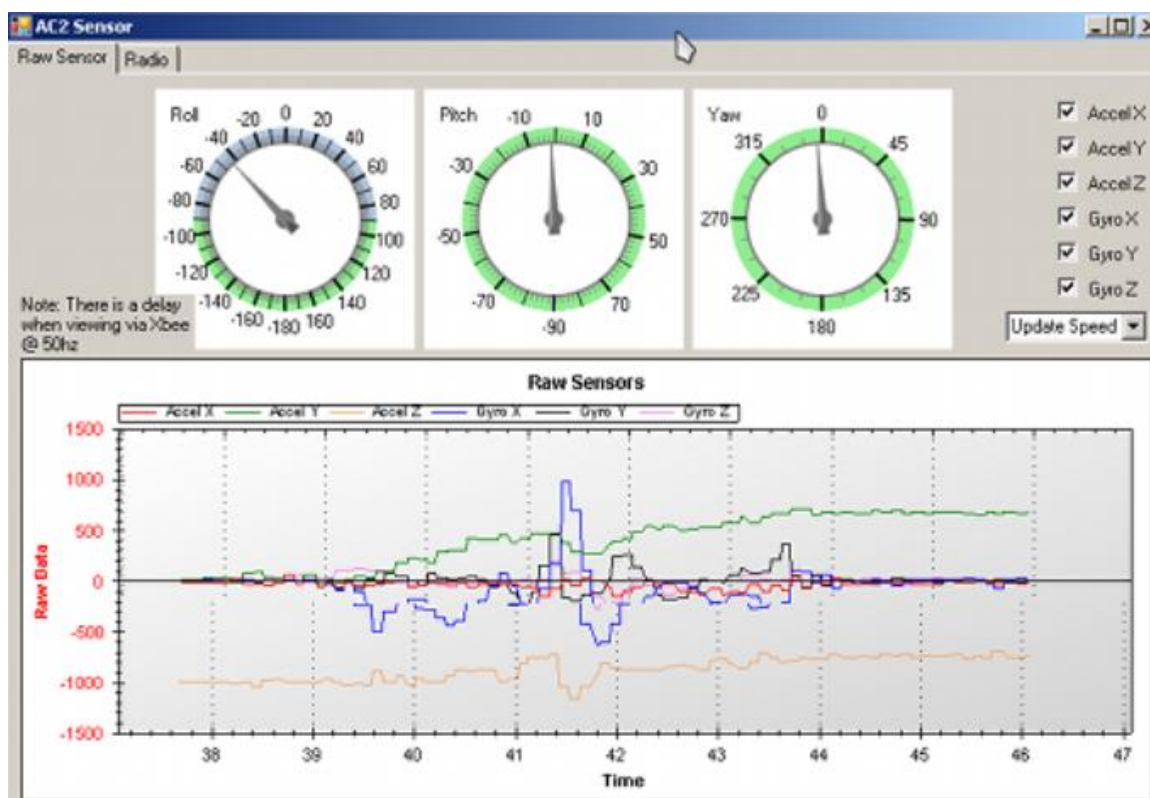


Рисунок 3.4. – Вкладка Raw sensor

Також відразу варто перевірити роботу приймача в сусідній вкладці і якщо рівні рухаються як треба, то відхиляємо джойстик газу (в пульті Mode 2 лівий) вниз вправо на 2 секунди, світлодіод на контролері повинен почати блимати. А тепер повільно рухаємо ручку газу вгору, причому в лівому стовпці рівні повинні приблизно однаково збільшуватися.

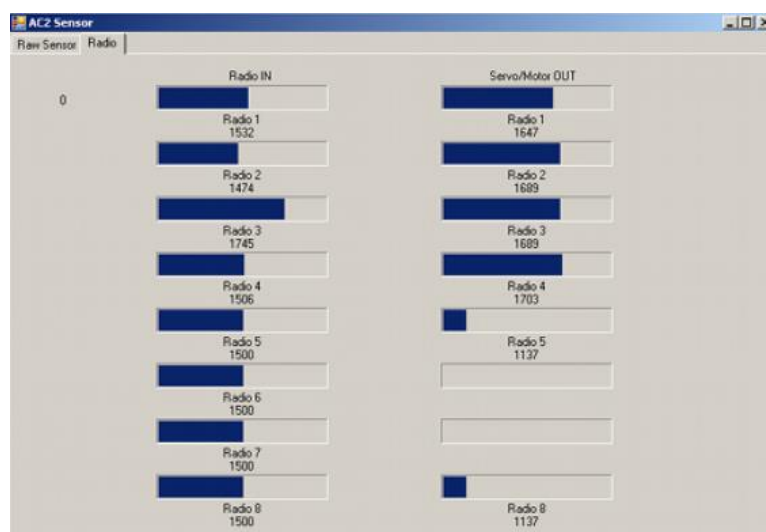


Рисунок 3.5. – Вікно перевірки роботи приймача

ВИСНОВКИ

У роботі вирішено актуальне завдання, що полягає у розробці технології технічного діагностування агродронів на основі комплексного аналізу віброакустичних, теплових та електричних параметрів їх конструктивних елементів. Актуальність дослідження обумовлена стрімким розвитком технологій точного землеробства та широким впровадженням безпілотних літальних апаратів у процеси внесення засобів захисту рослин, моніторингу посівів і виконання інших агротехнологічних операцій. Зростання інтенсивності експлуатації агродронів підвищує вимоги до їхньої надійності, безпеки та ефективності технічного обслуговування.

У ході дослідження проведено аналіз сучасних конструкцій агродронів, їх класифікації, особливостей будови та сфер застосування в агропромисловому комплексі. Встановлено, що найбільш відповідальними вузлами, від технічного стану яких залежить безпечна експлуатація БПЛА, є гвинтомоторна група, акумуляторна батарея, електронні регулятори швидкості (ESC), система живлення та навігаційне обладнання. Саме ці елементи визначають льотно-технічні характеристики та надійність виконання технологічних операцій.

Виконано аналіз типових відмов і дефектів конструктивних елементів агродронів. Встановлено, що найбільш поширеними причинами несправностей є дисбаланс пропелерів, зношування підшипників електродвигунів, деградація літій-полімерних акумуляторів, перегрів силових електроніки, механічна втома елементів рами та порушення роботи електричних з'єднань. Доведено, що своєчасне виявлення зазначених дефектів дозволяє значно підвищити рівень експлуатаційної надійності безпілотних літальних апаратів.

Проведено огляд сучасних методів технічного сервісу та діагностування БПЛА. Встановлено, що найбільш перспективним напрямом є комплексне використання вібраційної, теплової та електричної діагностики із застосуванням цифрової обробки сигналів та алгоритмів прогностичного технічного обслуговування (Predictive Maintenance). Такий підхід дозволяє перейти від планово-попереджувальної системи ремонту до обслуговування за фактичним технічним станом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрущенко В.М. Основи технічної діагностики безпілотних літальних апаратів. – Київ: НУБіП України, 2022. – 256 с.
2. Бондаренко О.П. Сучасні агродрони у точному землеробстві. – Харків: ХНТУСГ, 2021. – 198 с.
3. Василенко І.В. Безпілотні авіаційні системи в агропромисловому комплексі України. – Київ: Аграрна наука, 2023. – 214 с.
4. Гнатюк П.М. Технічне обслуговування та ремонт БПЛА. – Львів: Сполом, 2020. – 243 с.
5. Даниленко С.О. Системи технічної діагностики машин і обладнання. – Київ: Ліра-К, 2021. – 318 с.
6. Дорошенко А.В. Вібраційна діагностика машин та механізмів. – Київ: КПІ ім. Ігоря Сікорського, 2019. – 276 с.
7. Єременко М.О. Інтелектуальні системи моніторингу технічного стану БПЛА. – Дніпро: ДНУ, 2022. – 205 с.
8. Жук В.М. Основи експлуатації безпілотних авіаційних комплексів. – Київ: НАУ, 2020. – 290 с.
9. Захарченко І.П. Технології точного землеробства в Україні. – Харків: Факт, 2021. – 184 с.
10. Іванченко Р.В. Автоматизовані системи керування агродронами. – Київ: Аграрна освіта, 2023. – 241 с.
11. Карпенко О.О. Методи діагностування електромеханічних систем. – Одеса: ОДАУ, 2020. – 211 с.
12. Кириленко В.В. Безконтактні методи технічної діагностики. – Київ: Політехніка, 2019. – 227 с.
13. Коваленко А.М. Сучасні електронні системи безпілотних літальних апаратів. – Львів: Новий Світ-2000, 2022. – 236 с.
14. Козак Ю.П. Тепловізійний контроль технічного стану обладнання. – Київ: Основа, 2021. – 187 с.
15. Лисенко В.П. Віброакустичний контроль технічного стану машин. – Київ: Видавництво НУБіП, 2022. – 264 с.
16. Марченко С.В. Діагностика електроприводів мобільних машин. – Харків: ХНАДУ, 2021. – 208 с.
17. Мельник І.В. Основи надійності технічних систем. – Київ: Центр учбової літератури, 2020. – 312 с.
18. Мороз О.Г. Безпілотні технології у сільському господарстві. – Вінниця: ВНАУ, 2023. – 218 с.
19. Нестеренко П.О. Інформаційні технології в аграрному виробництві. – Київ: Урожай, 2021. – 246 с.
20. Олійник С.М. Дистанційний моніторинг агроєкосистем за допомогою БПЛА. – Київ: Аграрна наука, 2022. – 192 с.
21. Павленко Д.В. Методи контролю технічного стану безпілотних систем. – Київ: НАУ, 2021. – 233 с.

22. Петренко В.О. Технічний сервіс аграрної техніки. – Харків: ХНТУСГ, 2019. – 301 с.
23. Поліщук А.М. Теорія технічної діагностики машин. – Київ: Ліра-К, 2020. – 279 с.
24. Приходько Ю.С. Алгоритми оцінювання технічного стану БПЛА. – Дніпро: ДНУ, 2023. – 194 с.
25. Романенко О.В. Системи автоматичного керування агродронами. – Київ: КПІ, 2022. – 215 с.
26. Савченко М.П. Основи технічної експлуатації літальних апаратів. – Київ: НАУ, 2020. – 286 с.
27. Семенюк В.О. Методи прогнозування технічного ресурсу машин. – Київ: Освіта України, 2021. – 248 с.
28. Сидоренко Л.В. Цифрові технології точного землеробства. – Київ: Аграрна освіта, 2023. – 205 с.
29. Соловйов П.М. Інтелектуальні системи підтримки прийняття рішень у технічній діагностиці. – Харків: ХПІ, 2022. – 229 с.
30. Степаненко В.В. Методи неруйнівного контролю технічних систем. – Київ: Політехніка, 2020. – 261 с.
31. Ткаченко І.О. Діагностика електронних систем мобільної техніки. – Київ: Центр учбової літератури, 2021. – 224 с.
32. Усенко П.В. Надійність безпілотних літальних апаратів. – Київ: НАУ, 2022. – 210 с.
33. Федоренко М.М. Технології внесення засобів захисту рослин агродронами. – Харків: ХНАУ, 2023. – 187 с.
34. Харченко В.П. Авіаційні інформаційні системи. – Київ: НАУ, 2019. – 334 с.
35. Хоменко С.В. Контроль технічного стану електричних машин. – Львів: Львівська політехніка, 2020. – 213 с.
36. Цимбалюк І.О. Вібраційний аналіз у технічній діагностиці. – Київ: КПІ, 2021. – 238 с.
37. Черненко О.В. Системи навігації безпілотних літальних апаратів. – Київ: НАУ, 2022. – 219 с.
38. Шаповал В.М. Безпілотні технології у сучасному агробізнесі. – Київ: Аграрна наука, 2023. – 196 с.
39. Шевченко І.П. Методи технічного сервісу високотехнологічної техніки. – Харків: ХНАДУ, 2020. – 242 с.
40. Щербак О.М. Технічна експлуатація мобільних роботизованих систем. – Київ: Освіта України, 2022. – 207 с.
41. Ющенко П.В. Комплексна діагностика мехатронних систем. – Дніпро: ДНУ, 2021. – 231 с.
42. Яценко В.М. Основи експлуатації електронних систем агродронів. – Київ: НУБіП України, 2023. – 214 с.
43. Ярошенко С.О. Інноваційні технології використання безпілотних літальних апаратів у рослинництві. – Київ: Аграрна освіта, 2024. – 225 с.

ДОДАТКИ

Технологічна схема діагностування



Рисунок Д.1 – Алгоритм технічного діагностування XAG P40

Таблиця 1 – Технологічна карта ТО

№	Операція	Метод контролю	Інструмент	Норматив
1	Огляд корпусу	Візуальний	Ліхтар	Відсутність тріщин
2	Перевірка рами	Візуальний	Шаблон	Без деформацій
3	Контроль кріплень	Механічний	Динамометричний ключ	Надійна фіксація
4	Огляд пропелерів	Візуальний	Лупа	Без сколів
5	Балансування пропелерів	Віброаналіз	Балансир	Допустимий дисбаланс
6	Перевірка двигунів	Пробний запуск	Тахометр	Стабільні RPM

7	Контроль ESC	Тепловізійний	Тепловізор	Без локального перегріву
8	Перевірка АКБ	Електричний	Мультиметр/BMS	Нормальна напруга
9	Контроль кабелів	Візуальний	Огляд	Без пошкоджень
10	Контроль форсунок	Функціональний	Вода	Рівномірне розпилення
11	Перевірка насосів	Функціональний	Витратомір	Номінальна подача
12	Перевірка RTK/GPS	Програмний	ПЗ	Стійкий FIX
13	Контроль радара	Самодіагностика	ПЗ	Без помилок
14	Перевірка IMU	Калібрування	ПЗ	Допуск
15	Контроль телеметрії	Пробний запуск	Контролер	Стабільний зв'язок
16	Формування висновку	Комплексний	Протокол	Придатність до польоту

Етап 1. Візуальна діагностика

Перевіряються:

- несуча рама;
- промені;
- посадкові опори;
- корпус;
- кріплення;
- антени;
- бак;
- форсунки.

ПРОТОКОЛ ТЕХНІЧНОГО КОНТРОЛЮ ТО АГРОДРОНА XAG P40

Загальна інформація

Дата діагностування _____

Час _____

Модель БПЛА XAG P40

Серійний номер _____

Оператор (ПІБ) _____

Місце проведення контролю _____

Контрольний чек-лист

№	Контрольований вузол	Стан	Примітка
1	Несуча рама	☐ Справна ☐ Несправна	_____
2	Пропелерна група	☐ Справна ☐ Несправна	_____
3	Електродвигуни	☐ Справні ☐ Несправні	_____
4	ESC (регулятори швидкості)	☐ Справні ☐ Несправні	_____
5	Акумуляторна батарея (АКБ)	☐ Справна ☐ Несправна	_____
6	Кабелі та електричні з'єднання	☐ Справні ☐ Несправні	_____
7	Система розпилення	☐ Справна ☐ Несправна	_____
8	Насоси та форсунки	☐ Справні ☐ Несправні	_____
9	RTK/GPS система	☐ Справна ☐ Несправна	_____
10	IMU та компас	☐ Справні ☐ Несправні	_____
11	Телеметрія та зв'язок	☐ Справні ☐ Несправні	_____
12	Пробний запуск	☐ Виконано ☐ Не виконано	_____

Результати діагностування
Загальний технічний стан

- ☐ Повністю справний
- ☐ Потребує планового технічного обслуговування
- ☐ Потребує ремонту
- ☐ Експлуатація заборонена до усунення несправностей

Виявлені дефекти

Рекомендовані роботи

- ☐ Балансування пропелерів
- ☐ Заміна АКБ
- ☐ Перевірка ESC
- ☐ Заміна електродвигуна
- ☐ Калібрування RTK/GPS
- ☐ Перевірка системи розпилення
- ☐ Інше: _____

Виконавець діагностики _____

Підпис _____

Дата _____

ДОДАТОК

PID_v1.cpp

```

/*****
*****
* Arduino PID
*****
*****/

#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#include <PID_v1.h>

/*Constructor (...)* *****/
*   The parameters specified here are those for for which we can't set up
*   reliable defaults, so we need to have the user set them.
*****/
PID::PID(double* Input, double* Output, double* Setpoint,
         double Kp, double Ki, double Kd, int ControllerDirection)
{

    myOutput = Output;
    myInput = Input;
    mySetpoint = Setpoint;
    inAuto = false;

    PID::SetOutputLimits(0, 255);           //default output limit corresponds to
                                            //the arduino pwm
    limits

    SampleTime = 100;                       //default Controller
    Sample Time is 0.1 seconds

    PID::SetControllerDirection(ControllerDirection);
    PID::SetTunings(Kp, Ki, Kd);

    lastTime = millis()-SampleTime;
}

```

```

/* Compute()
*****
*   This, as they say, is where the magic happens.  this function should be called
*   every time "void loop()" executes.  the function will decide for itself whether a new
*   pid Output needs to be computed.  returns true when the output is computed,
*   false when nothing has been done.

*****
***/
bool PID::Compute()
{
    if(!inAuto) return false;
    unsigned long now = millis();
    unsigned long timeChange = (now - lastTime);
    if(timeChange>=SampleTime)
    {
        /*Compute all the working error variables*/
        double input = *myInput;
        double error = *mySetpoint - input;
        ITerm+= (ki * error);
        if(ITerm > outMax) ITerm= outMax;
        else if(ITerm < outMin) ITerm= outMin;
        double dInput = (input - lastInput);

        /*Compute PID Output*/
        double output = kp * error + ITerm- kd * dInput;

        if(output > outMax) output = outMax;
        else if(output < outMin) output = outMin;
        *myOutput = output;

        /*Remember some variables for next time*/
        lastInput = input;
        lastTime = now;
        return true;
    }
    else return false;
}

/* SetTunings(...)*****
*   This function allows the controller's dynamic performance to be adjusted.
*   it's called automatically from the constructor, but tunings can also
*   be adjusted on the fly during normal operation

```

```

*****/
void PID::SetTunings(double Kp, double Ki, double Kd)
{
    if (Kp<0 || Ki<0 || Kd<0) return;

    dispKp = Kp; dispKi = Ki; dispKd = Kd;

    double SampleTimeInSec = ((double)SampleTime)/1000;
    kp = Kp;
    ki = Ki * SampleTimeInSec;
    kd = Kd / SampleTimeInSec;

    if(controllerDirection ==REVERSE)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
}

/* SetSampleTime(...) *****
 * sets the period, in Milliseconds, at which the calculation is performed
 *****/
void PID::SetSampleTime(int NewSampleTime)
{
    if (NewSampleTime > 0)
    {
        double ratio = (double)NewSampleTime
            / (double)SampleTime;
        ki *= ratio;
        kd /= ratio;
        SampleTime = (unsigned long)NewSampleTime;
    }
}

/* SetOutputLimits(...)*****
 * This function will be used far more often than SetInputLimits. while
 * the input to the controller will generally be in the 0-1023 range (which is
 * the default already,) the output will be a little different. maybe they'll
 * be doing a time window and will need 0-8000 or something. or maybe they'll
 * want to clamp it from 0-125. who knows. at any rate, that can all be done
 * here.
 *****/
void PID::SetOutputLimits(double Min, double Max)
{

```

```

    if(Min >= Max) return;
    outMin = Min;
    outMax = Max;

    if(inAuto)
    {
        if(*myOutput > outMax) *myOutput = outMax;
        else if(*myOutput < outMin) *myOutput = outMin;

        if(ITerm > outMax) ITerm= outMax;
        else if(ITerm < outMin) ITerm= outMin;
    }
}

/* SetMode(...)*****
 * Allows the controller Mode to be set to manual (0) or Automatic (non-zero)
 * when the transition from manual to auto occurs, the controller is
 * automatically initialized
 *****/
void PID::SetMode(int Mode)
{
    bool newAuto = (Mode == AUTOMATIC);
    if(newAuto == !inAuto)
    { /*we just went from man

PID_v1.h

#ifndef PID_v1_h
#define PID_v1_h
#define LIBRARY_VERSION    1.0.0

class PID
{

public:

    //Constants used in some of the functions below
#define AUTOMATIC    1
#define MANUAL    0
#define DIRECT    0
#define REVERSE    1

    //commonly used functions
*****
    PID(double*, double*, double*,    // * constructor. links the PID to the Input, Output, and

```

```

double, double, double, int); // Setpoint. Initial tuning parameters are also set here

void SetMode(int Mode); // * sets PID to either Manual (0) or Auto (non-0)

bool Compute(); // * performs the PID calculation. it should be
                // called every time loop() cycles. ON/OFF and
                // calculation frequency can be set using SetMode
                // SetSampleTime respectively

void SetOutputLimits(double, double); //clamps the output to a specific range. 0-255 by default,
but
//it's likely the user will want to
change this depending on
//the application

//available but not commonly used functions
*****
void SetTunings(double, double, // * While most users will set the tunings once in the
double); // constructor, this function gives the user the option
// of changing tunings during runtime for Adaptive control
void SetControllerDirection(int); // * Sets the Direction, or "Action" of the controller.
DIRECT
// means the output will
increase when error is positive. REVERSE
// means the opposite. it's
very unlikely that this will be needed
// once it is set in the
constructor.
void SetSampleTime(int); // * sets the frequency, in Milliseconds, with which
// the PID calculation is performed. default is 100

//Display functions
*****
double GetKp(); // These functions query the pid for
interal values.
double GetKi(); // they were created mainly for the pid
front-end,
double GetKd(); // where it's important to know what is
actually
int GetMode(); // inside the PID.
int GetDirection(); //

```

```

private:
void Initialize();

double dispKp;                // * we'll hold on to the tuning parameters in user-
entered
double dispKi;                //  format for display purposes
double dispKd;                //

double kp;                    // * (P)roportional Tuning Parameter
double ki;                    // * (I)ntegral Tuning Parameter
double kd;                    // * (D)erivative Tuning Parameter

int controllerDirection;

double *myInput;              // * Pointers to the Input, Output, and Setpoint variables
double *myOutput;             //  This creates a hard link between the variables and the
double *mySetpoint;           //  PID, freeing the user from having to constantly tell us
                              //  what these values are.  with pointers we'll just know.

unsigned long lastTime;
double ITerm, lastInput;

unsigned long SampleTime;
double outMin, outMax;
bool inAuto;
};
#endif

```