КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено: «На правах рукопису»
Завідувач кафедри _____ Ігор АНІСІМОВ
« __ » червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«РОЗРОБКА СИСТЕМИ КЕРУВАННЯ КВАДРОКОПТЕРОМ НА ОСНОВІ ARDUINO»

Виконав:
студент 4-го курсу
денної форми навчання
спеціальності 172 - Телекомунікації та радіотехніка
ОП «Інформаційна безпека телекомунікаційних систем і мереж»
Гарбуз Микита Григорович _____

Науковий керівник:
к.ф.-м.н., ас. Фесенко Сергій Олександрович _____

Рецензент:
д.ф.-м.н., проф. Веклич Анатолій Миколайович _____

Засвідчую, що у цій бакалаврській роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від «23» червня 2023 р., протокол № 22.

Завідувач кафедри радіотехніки та радіоелектронних систем,
доктор фіз.-мат. наук, професор
Анісімов Ігор Олексійович _____

Київ – 2023

Зміст:

Вступ

1.1 Постановка проблеми

Квадрокоптери - це багатофункціональні пристрої, які широко використовуються в різних сферах діяльності: від аерофотозйомки (рис. 1.1.1) до перевезення вантажів. Однак, під час їх експлуатації виникає ряд технічних та управлінських проблем. Наприклад, деякі комерційно доступні моделі квадрокоптерів мають обмежені можливості налаштування та керування, що може призвести до недостатньої гнучкості в експлуатації.

Крім того, багато з комерційно доступних систем керування квадрокоптерами є закритими, що ускладнює їх адаптацію до конкретних потреб користувача. В свою чергу, системи, які пропонують більше гнучкості, зазвичай є складними та дорогими.

Ці обмеження підкреслюють необхідність розробки системи керування квадрокоптером, яка була б гнучкою, доступною, та зручною для користувача. Система на основі Arduino представляє великий потенціал для цього завдання, оскільки цей мікроконтролер є доступним, має великі можливості для програмування та підтримує широкий спектр периферійних пристроїв.

Додатково до викладеного вище, виникає ряд специфічних проблем, пов'язаних з керуванням квадрокоптером. По-перше, необхідно розробити ефективні алгоритми стабілізації положення квадрокоптера в повітрі. Це особливо важливо, оскільки незначні зміни в оборотах двигунів можуть призвести до нестабільного польоту або, у гіршому випадку, до падіння квадрокоптера.

По-друге, потрібно врахувати обмеженість живлення. Автономність квадрокоптера є важливим фактором, який залежить від енергоефективності системи керування та використаних компонентів.

По-третє, необхідно врахувати вимоги до безпеки. Випадкові падіння, зіткнення з перешкодами, або неконтрольований політ квадрокоптера можуть призвести до серйозних наслідків. Тому система керування повинна передбачати механізми забезпечення безпеки.

Також варто відзначити, що розробка такої системи вимагає знання в області програмування, електроніки, аеродинаміки, та контрольно-вимірювальних систем.

Таким чином, задача розробки системи керування квадрокоптером на основі Arduino стає важливим і актуальним напрямком дослідження, яке вимагає інтеграції різноманітних знань та вмінь.

Рис.1.1.1. Квадрокоптер для аерофотозйомки DJI Air 2S[7]

1.2 Актуальність теми

Зростаючий інтерес до безпілотних літальних апаратів, зокрема квадрокоптерів, зумовлює постійний пошук нових технологій та методів їх використання. Вони вже активно використовуються в різних сферах, включаючи геодезію, сільське господарство, безпеку, рятувальні операції, розвідку, сферу розваг та багато інших. Проте, забезпечення ефективного керування та висока надійність таких систем залишається викликом.

Платформа Arduino вже успішно застосовується в різних сферах: від освіти до промисловості, завдяки своїй доступності, гнучкості та великій підтримці спільноти. Її використання для розробки системи керування квадрокоптером може бути цікавим і актуальним напрямком дослідження, який може внести вагомий вклад в цю область.

Використання Arduino для керування квадрокоптером дозволяє створити гнучку, доступну та адаптовану до специфічних потреб систему. Крім того, така система може служити основою для подальшого розвитку та модифікації, що сприяє розвитку відкритих інноваційних технологій в цій області.

Також слід зазначити, що квадрокоптери та інші безпілотні літальні апарати залишаються предметом важливих правових дискусій та регулювань у багатьох країнах. Розробка безпечних, надійних і контрольованих систем керування є ключовим фактором, що впливає на правову рамку та соціальне прийняття цих технологій.

Окрім того, у сучасних військових конфліктах важливу роль відіграють безпілотні аеронавтичні системи, включаючи квадрокоптери. Вони можуть виконувати широкий спектр завдань, від розвідки і спостереження до доставки вантажів і, в деяких випадках, прямої участі в бойових діях.

Керування такими системами стає критично важливим елементом їх ефективності. Здатність до автономного керування, розуміння та адаптації до

навколишнього середовища може значно покращити їх здатність до виконання завдань, а також знизити ризики для людського персоналу.

Розробка системи керування квадрокоптером на основі Arduino має значну актуальність в такому контексті. Arduino - це відносно дешева і гнучка платформа, яка може бути адаптована для великої кількості задач.

Навички, отримані під час розробки такої системи, можуть бути безпосередньо застосовані до більш складних і потужних систем, що використовуються у військових цілях. Це включає здатність розуміти та розробляти алгоритми керування, працювати з датчиками та актуаторами, інтегрувати різні компоненти в єдину систему та вирішувати проблеми, пов'язані з реальним світом.

Крім того, можливість розробки власних систем керування може дати стратегічну перевагу у військових ситуаціях, оскільки воно зменшує залежність від іноземних постачальників і забезпечує більший контроль над технологією.

Таким чином, в умовах війни, розробка системи керування квадрокоптером на основі Arduino стає надзвичайно актуальною і має безпосереднє застосування. Крім безпосереднього використання на полі бою, дрони можуть також забезпечити важливі можливості для:

1. Розвідки та спостереження: автономні дрони можуть слідкувати за ворожими позиціями, виявляти рухи ворога, контролювати ситуацію на лінії фронту або навіть спостерігати за важливими інфраструктурними об'єктами.
2. Картографування: дрони можуть робити повітряні знімки територій, що дозволяє створювати детальні карти, які можуть бути використані для планування оперативних дій або оцінки шкоди від бойових дій.
3. Доставка вантажів: квадрокоптери можуть доставляти невеликі вантажі, такі як медичні засоби, продовольство або навіть важливу інформацію, в недоступні або небезпечні місця.

4. Рятувальні операції: в ситуаціях, коли людське життя є в небезпеці, дрони можуть допомогти рятувальникам шукати загублених людей або доставляти життєво важливі ресурси в зони надзвичайних ситуацій.
5. Безпосередня участь у бойових діях: дрони-камікадзе (рис. 1.2.1) представляють собою новітню військову технологію, яка використовується для точкових ударів з великою точністю. Це безпілотні літальні апарати, які здатні відшукати, відстежити та вразити ціль, використовуючи вбудований вибуховий заряд. Основною метою такого дрона є самознищення з максимальним шкодою для ворога.

Рис. 1.2.1. Зовнішній вигляд дрона-камікадзе українського виробництва [8]

Отже, актуальність даної теми полягає в необхідності розробки ефективних та доступних систем керування для квадрокоптерів, що можуть

бути модифіковані та адаптовані для різних потреб. Розробка такої системи на основі Arduino має значний потенціал для внесення вкладу в розвиток цієї області.

1.3 Мета і завдання дослідження

Головною метою цієї роботи є розробка та реалізація надійної, ефективної та гнучкої системи керування квадрокоптером на базі мікроконтролера Arduino.

Завдання дослідження:

1. Аналіз і дослідження: Це включає аналіз існуючих технологій та підходів до керування квадрокоптерами, а також вивчення можливостей платформи Arduino.
2. Розробка вимог: Опрацювання вимог до системи керування, які враховують критичні аспекти, такі як стабільність польоту, безпека, енергоефективність та адаптивність до різних сценаріїв використання.
3. Проектування системи: На основі аналізу та вимог буде створено проект системи керування, який включає архітектуру системи, алгоритми керування та схему взаємодії з користувачем.

РОЗДІЛ 2. ТЕОРИТИЧНА ЧАСТИНА

2.1 Огляд сучасних технологій керування квадрокоптерами

Системи керування квадрокоптерами продовжують розвиватися впродовж останніх років. Вони включають в себе різні технології, що відповідають за стабілізацію, навігацію, маршрутизацію та інтерфейс користувача. Ось декілька ключових аспектів:

Стабілізація та управління польотами: Основна функція системи керування - забезпечити стабілізацію квадрокоптера під час польоту. Для цього використовуються алгоритми керування, як-от PID (proportional-integral-derivative), що регулюють швидкість обертання кожного з чотирьох роторів на основі даних з датчиків, таких як гіроскопи та акселерометри.

Навігація та маршрутизація: Для навігації та обрання маршруту квадрокоптери використовують різні технології, включаючи GPS для зовнішньої навігації, оптичну відстань та визначення положення для навігації в приміщенні, а також технології обробки зображень для виявлення та уникнення перешкод.

Автономність: сучасні квадрокоптери все більше переходять до автономного керування, використовуючи алгоритми штучного інтелекту та машинного навчання. Це включає в себе автоматичне виконання завдань, таких як взліт, посадка, слідування за маршрутом та навіть виконання складних маневрів, таких як обхід перешкод.

Інтерфейс користувача: Системи керування квадрокоптерами також включають інтерфейси, що дозволяють людям керувати дронами. Це можуть бути радіоуправління, мобільні додатки, а також більш складні системи, що використовують жести, очі або навіть мозкові сигнали.

Платформа Arduino: Arduino - це популярна відкрита платформа для створення DIY проектів, включаючи квадрокоптери. Вона пропонує велику кількість сенсорів і модулів, що можуть бути використані для розробки

власної системи керування квадрокоптером, включаючи гіроскопи, акселерометри, GPS модулі, модулі радіоуправління і багато іншого.

Цей огляд демонструє, що сучасні технології керування квадрокоптерами є дуже розмаїтими і продовжують розвиватися, включаючи нові та інноваційні методи управління, що поєднують алгоритми штучного інтелекту, обробку зображень і інші технології.

2.2 Вивчення основних принципів керування квадрокоптерами

Управління квадрокоптером - це складний процес, що вимагає розуміння ряду ключових принципів. Завдяки ним, квадрокоптери можуть виконувати різноманітні завдання в різних умовах з високою точністю і ефективністю.

Цей розділ присвячений детальному розгляду цих принципів, включаючи стабілізацію і управління польотами, навігацію, безпеку та уникнення перешкод, автономне керування, керування енергоспоживанням та зв'язок та передачу даних. Вивчення цих принципів має важливе значення для розробки ефективної системи керування квадрокоптером на основі Arduino.

1. Стабілізація і управління польотами: Це базовий принцип управління квадрокоптером. Швидкість обертання кожного з роторів може бути індивідуально регульована для забезпечення руху вгору/вниз, вліво/вправо, вперед/назад або для здійснення обертання навколо вертикальної осі.
2. Управління шляхом польоту: Керування шляхом польоту квадрокоптера, як правило, здійснюється через зміну швидкості обертання окремих роторів. Це включає в себе розрахунок та імплементацію відповідного алгоритму керування, здатного реагувати на зовнішні фактори, як-от вітер, та забезпечувати стабільність польоту.

3. Навігація: Для навігації квадрокоптери використовують різні технології, включаючи GPS, компаси, акселерометри та гіроскопи. Ці датчики дозволяють квадрокоптеру визначати своє місцезнаходження та орієнтацію в просторі, а також слідкувати за заданими маршрутами польоту. Дальність польоту можна покращити за допомогою ретрансляторів сигналу, лідери ринку пропонують подібні рішення для професійного використання. Подібне рішення можна побачити на Рис. 2.2.1.

Рис. 2.2.1. Зовнішній вигляд ретранслятора DJI D-RTK 2[9]

4. Безпека та уникнення перешкод: Це важливий аспект керування квадрокоптерами, особливо при польотах в міській місцевості або в інших складних умовах. Сучасні технології, такі як LIDAR або камери, використовуються для виявлення і уникнення перешкод.
5. Автономне керування: Автономне керування включає в себе використання алгоритмів штучного інтелекту та машинного навчання для автоматичного виконання завдань, таких як зліт, посадка, слідування за маршрутом та навіть виконання складних маневрів.
6. Керування енергоспоживанням: Ефективне керування енергоспоживанням важливе для максимізації часу польоту та оптимізації роботи системи.

7. Зв'язок та передача даних: Системи керування квадрокоптером також мають забезпечувати надійний зв'язок з дроном та передачу даних, таких як відеозображення або телеметрична інформація.

2.3 Дослідження можливостей та характеристик Arduino у використанні для керування квадрокоптерами

Arduino - це відкрита платформа, що включає апаратне та програмне забезпечення, розроблена для створення інтерактивних проектів. Завдяки своїй гнучкості та доступності, Arduino широко використовується у різних областях, включаючи розробку систем керування для квадрокоптерів. Далі на Рис. 2.3.1. представлена найбільш розповсюджена модель Arduino Uno.

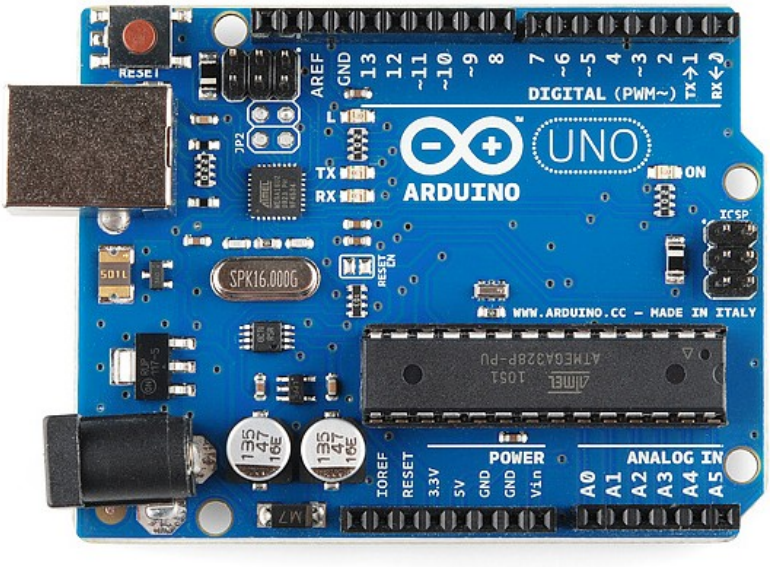

Рис. 2.3.1. Зовнішній вигляд плати Arduino Uno[10]

1. Апаратні можливості Arduino: Arduino може використовуватися для контролю моторів, читання даних з сенсорів, забезпечення зв'язку з пультом управління та інших важливих завдань, необхідних для керування квадрокоптером. Arduino Uno, наприклад, має 14 цифрових вхід/вихідних пінів, 6 з яких можна використовувати як

PWM виходи для керування швидкістю моторів, а також 6 аналогових входів для читання даних з сенсорів.

2. Програмне забезпечення Arduino: Arduino IDE дозволяє легко розробляти програмне забезпечення для Arduino. Використовуючи мову програмування, подібну до C++, можна розробляти складні алгоритми керування для квадрокоптера.
3. Зовнішні модулі: Arduino може використовувати різноманітні зовнішні модулі, такі як GPS, компаси, акселерометри, гіроскопи, радіомодулі для покращення своїх можливостей та виконання специфічних завдань.
4. Гнучкість і масштабованість: Завдяки своїй модульній структурі та відкритому коду, Arduino може бути легко адаптована для різних завдань та вимог. Це дозволяє використовувати Arduino для створення кастомних систем керування для квадрокоптерів різного розміру та функціональності.

Вивчення цих характеристик та можливостей Arduino є важливим кроком у розробці системи керування квадрокоптером на її основі.

2.4 Розробка концепції системи керування

Після вивчення основних принципів керування квадрокоптерами та характеристик платформи Arduino, наступний крок - розробка концепції системи керування.

1. Основні компоненти: Концепція передбачає використання плати Arduino в якості основи системи керування, яка буде контролювати мотори, читати дані з сенсорів та забезпечувати зв'язок з пультом управління або наземною станцією.

Плати Arduino в основному збудовані на мікроконтролерах компанії Atmel. Різні моделі Arduino використовують різні мікроконтролери, тому характеристики можуть варіюватися в залежності від моделі. Ми будемо

використовувати контролер ATmega328P. Цей мікроконтролер є основою для Arduino Uno і Arduino Nano. Він працює на частоті до 20 MHz, має 32 KB флеш-пам'яті для програм, 2 KB оперативної пам'яті та 1 KB EEPROM для зберігання даних. ATmega328P має 23 вхідно-вихідних порти, включаючи 6 портів, які можуть генерувати PWM-сигнали, та 6 аналогових входів.

2. Сенсори: Система керування буде включати різні сенсори для забезпечення стабілізації та навігації. Це можуть бути гіроскопи, акселерометри, GPS, сенсори висоти та інші. Для забезпечення стабілізації ми використаємо модуль MPU6050 - це мікроелектромеханічна система (MEMS), яка включає в себе 3-осний акселерометр і 3-осний гіроскоп. Це допомагає нам вимірювати прискорення, швидкість, орієнтацію, переміщення та багато інших параметрів руху системи або об'єкта.
3. Алгоритми керування: Будуть розроблені алгоритми для обробки даних з сенсорів та керування швидкістю обертання моторів. Це включає алгоритми PID для стабілізації польоту та алгоритми для керування шляхом польоту.

Типовий квадрокоптер має чотири мотори з фіксованими кутами, вони надають квадрокоптеру чотири вхідні сили, які є тягою, що забезпечується кожним пропелером, як показано на Рис.2.4.1.

Існує дві можливі конфігурації для більшості конструкцій квадрокоптера «+» і «×» . Квадрокоптер X-конфігурації вважається більш стабільним порівняно з конфігурацією +, яка є більш акробатичною конфігурацією. Пропелери 1 і 3 обертаються за годинниковою стрілкою (CW), 2 і 4 обертаються проти годинникової стрілки (CCW). Таким чином, квадрокоптер може підтримувати рух вперед (назад), збільшуючи (зменшуючи) швидкість передніх (задніх) роторів і одночасно зменшуючи (збільшуючи) задню (передню) швидкість ротора, що означає зміну кута нахилу. Цей процес необхідний для компенсації ефекту дії/реакції (третій

закон Ньютона). Гвинти 1 і 3 мають протилежний крок відносно 2 і 4, тому всі тяги мають однаковий напрямок.

Рис. 2.4.1. Базове представлення типового квадрокоптера[1]

Положення квадрокоптера визначається в інерціальній системі осей X,Y,Z. Стан, тобто кут положення, визначається в інерціальній системі з трьома кутами Ейлера η. Кут нахилу θ визначає обертання квадрокоптера навколо осі Y. Кут крену φ визначає поворот навколо осі X і кут ψ повороту навколо осі Z.[1]

4. Зв'язок та передача даних: Система буде містити модуль для бездротового зв'язку з пультом управління. В нашому випадку модулем бездротового зв’язку буде виступати NRF24L01. Це однокристальний радіоприймач ISM-діапазону 2,4–2,5 ГГц. До складу трансивера входять: повністю інтегрований синтезатор частоти, підсилювач потужності, кристалічний генератор та модулятор. Канали вихідної потужності та частоти легко програмуються за допомогою 3-провідного послідовного інтерфейсу. Споживаний струм дуже низький, лише 10,5 мА при

вихідній потужності -5 дБм і 18 мА в режимі прийому. Вбудовані режими вимкнення живлення дозволяють досягти високого рівня енергоефективності.

5. Безпека і автономність: Система може включати функції безпеки, такі як автоматичне повернення додому при втраті зв'язку, і автономні функції, як-от автоматичний зліт та посадка.

Кінцевою метою є створення системи керування, яка буде надійною, витривалою, гнучкою та легкою для налаштування і модифікації.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

3.1 Вибір та обґрунтування вибору компонентів для квадрокоптера

Основним елементом системи керування обрано Arduino Uno - через їх доступність, гнучкість та велику підтримку спільноти. Вони мають достатню кількість входів/виходів для управління моторами, сенсорами та комунікаційними модулями.

Мотори: Вибір моторів залежить від ваги та розміру квадрокоптера. Необхідно вибрати мотори з достатнім обертальним моментом. Були обрані двигуни щіткового (рис. 3.1.1) типу через деякі їх переваги, а саме: для керування обертанням щіткового двигуна не потрібен складний контролер, як у випадку з безщітковими двигунами, це робить їх використання простішим та менш вартісним; щіткові двигуни дешевші в виробництві та закупівлі; щіткові двигуни можуть надавати великий стартовий крутний момент і є в змозі стабільно працювати при низьких обертах.

Рис. 3.1.1. Зовнішній вигляд безщіткових двигунів [11]

Сенсори: основним і єдиним сенсором у нашому квадрокоптері є MPU6050 (рис. 3.1.2.), який включає 3-осний акселерометр і 3-осний

гіроскоп, що вимірюють прискорення та орієнтацію квадрокоптера, необхідні для його стабільності та керування польотом.

Рис. 3.1.2. Зовнішній вигляд сенсора MPU6050[12]

Модуль зв'язку: Модуль зв'язку, такий як NRF24L01 (рис. 3.1.3), може бути використаний для зв'язку з пультом управління. Для пульта управління також був обраний цей модуль ось деякі його переваги:

1. Низький рівень споживання енергії: NRF24L01 - це модуль з низьким споживанням енергії, що робить його відмінним вибором для батарейних пристроїв, таких як дрони.
2. Низька вартість: На ринку доступні недорогі модулі NRF24L01, що робить цей компонент економічно ефективним для багатьох проектів.

3. Малі розміри: NRF24L01 має компактні розміри, що робить його підходящим для використання в малих пристроях.
4. Швидкість передачі даних: Модуль підтримує швидкість передачі даних до 2 Мбіт/с, що робить його досить швидким для багатьох застосувань.
5. Багатоканальна передача: NRF24L01 підтримує до 125 каналів, що дозволяє створювати складні бездротові мережі.
6. Дальність зв'язку: З можливістю посилення сигналу, дальність зв'язку може досягти декількох сотень метрів в ідеальних умовах.
7. Вбудовані функції безпеки: Модуль має вбудовані функції кодування, які допомагають забезпечити безпеку передачі даних.

Рис. 3.1.3. Зовнішній вигляд модуля NRF23L01 [13]

Акумулятор: Літієво-полімерний акумулятор (LiPol) (рис. 3.1.4) вибирається відповідно до потреб моторів та інших компонентів. Важливо врахувати ємність (mAh) для забезпечення достатнього часу польоту та

вибрати відповідну напругу (V). Для нашого проєкту був обраний акумулятор LiPol із напругою 3.7 Вольт та ємністю 500 міліампергодин через

оптимальний баланс розмірів та ваги, вкрай необхідний для компактного квадрокоптера.

Рис. 3.1.4. Зовнішній вигляд акумулятора LiPol [14]

3.2 Розробка схеми підключення компонентів на базі Arduino

Для проектування системи керування квадрокоптером на основі Arduino використали KiCad - програмний комплекс проєктування електронних пристроїв (EDA) з відкритими сирцевим кодом, який призначений для розробки електричних схем і друкованих плат.

На Рис. 3.2.1. показана принципова схема керування квадрокоптером на основі Arduino.

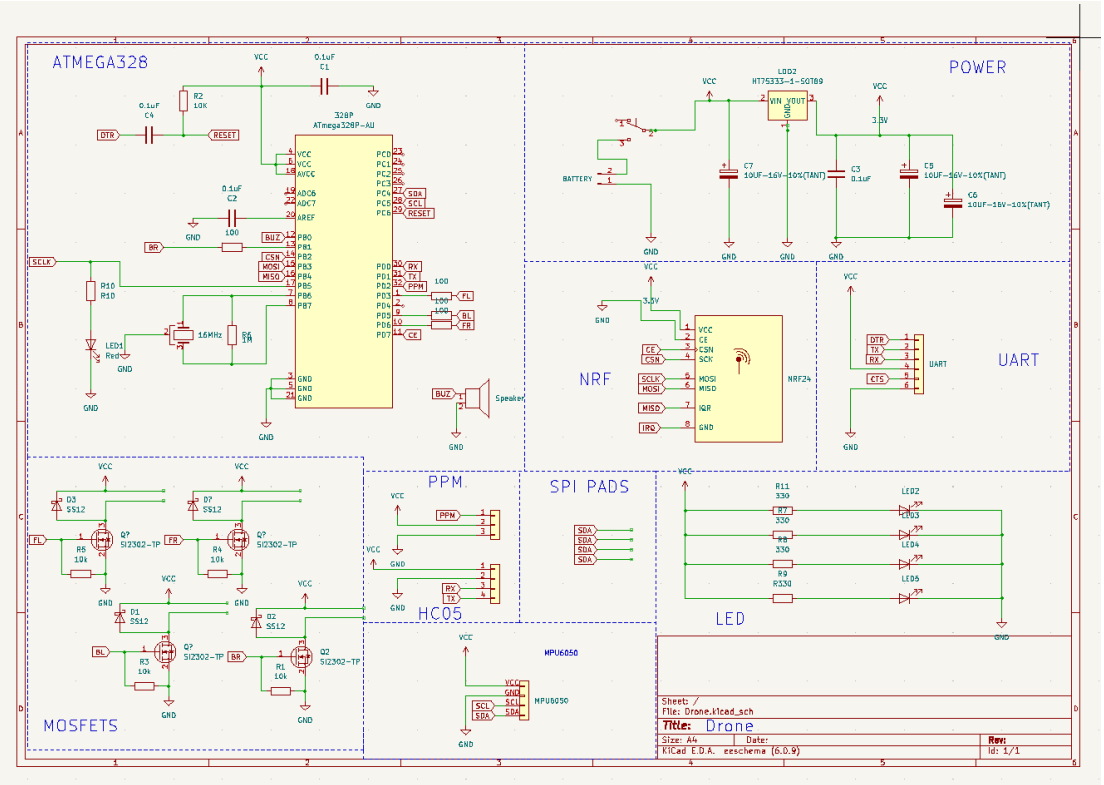

Рис. 3.2.1. Принципова електрична схема системи керування, розміщеної на квадрокоптері

Розглянемо детально елементи схеми та принцип їх роботи. На Рис. 3.2.2. показаний безпосередньо сам AVR контролер ATmega 328P з необхідною обв'язкою елементами, такими як кварцевий резонатор, блокувальною ємністю та джерелом опорної напруги, необхідними для коректного функціонування схеми. Окрім того, на пр инциповій схемі показані лінії підключення цифрових інтерфейсів SPI, I^2C UART, які використовуються для підключення датчиків і радіомодуля.

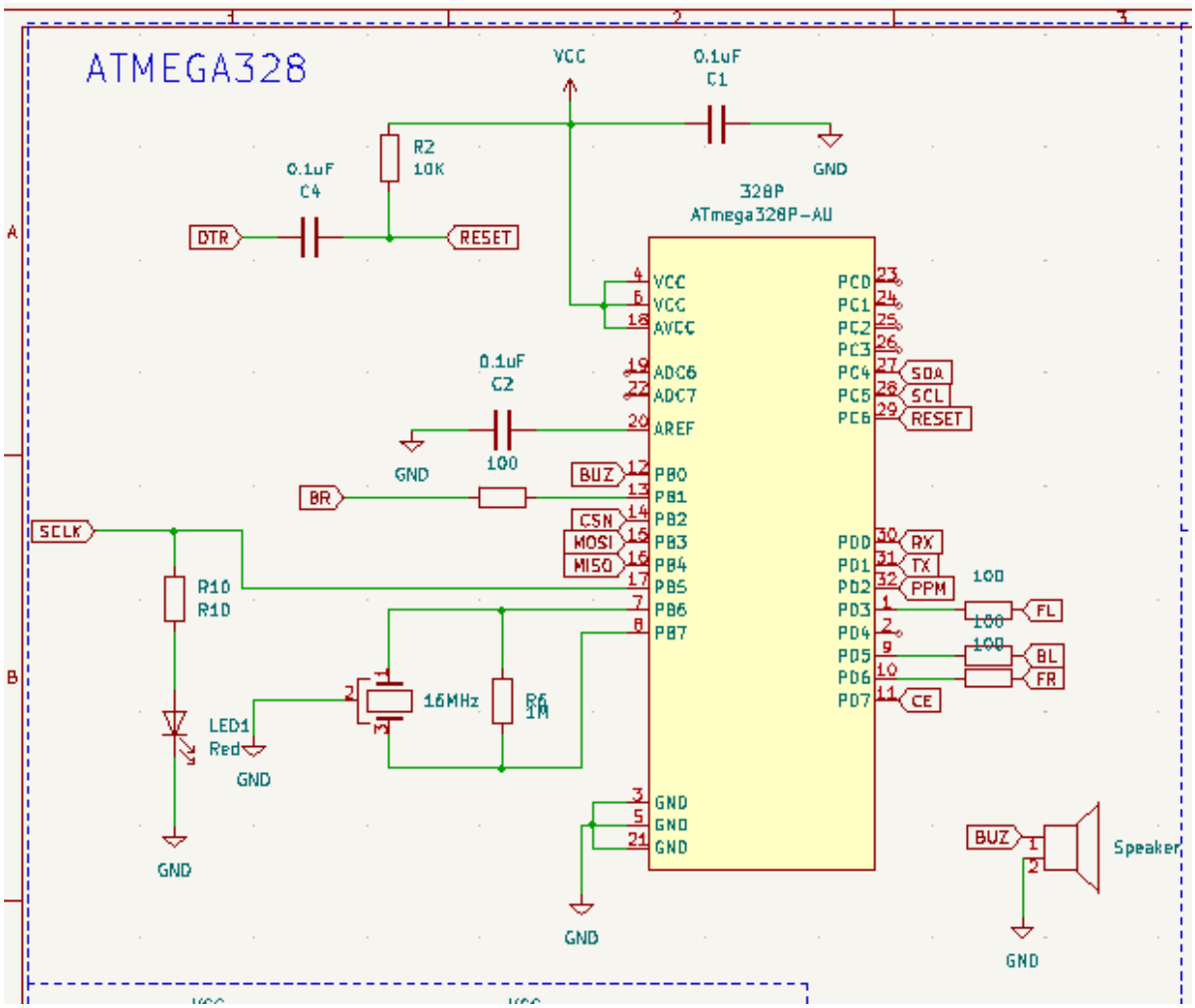

Рис. 3.2.2 Контролер AVR ATmega 328P

На Рис. 3.2.3 зображено блок регулювання живлення з входами для підключення акумулятора.

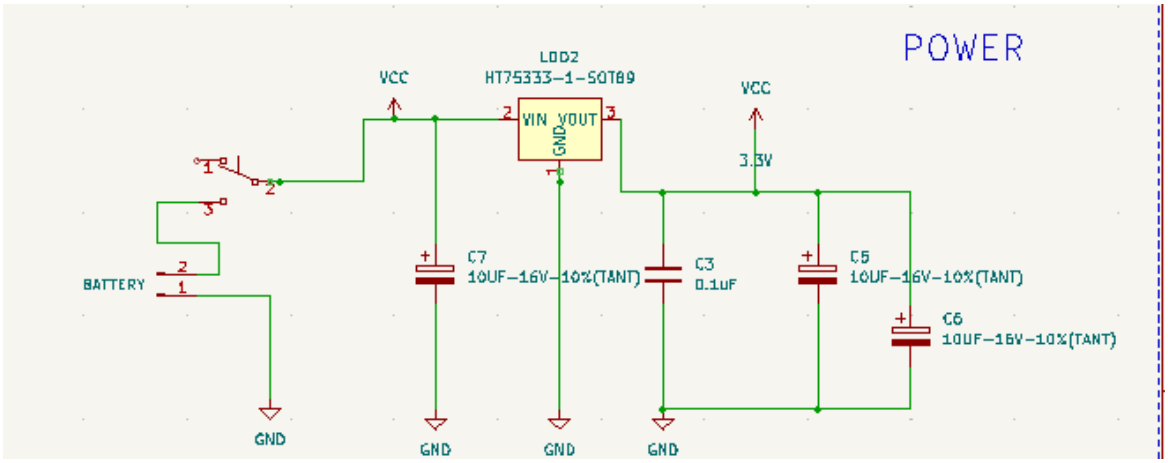

Рис. 3.2.3. Схема підключення живлення

У якості регулятора напруги використовувався лінійний стабілізатор HT7333-1, який видає напругу 3,3 В для живлення радіомодуля NRF24. Слід зауважити, що для живлення мікроконтролера використовується нестабілізована напруга акумуляторів. У нашому випадку це допустимо, оскільки АЦП мікроконтролера не використовується. Головне, щоб пульсації напруги живлення не виходили за допустимий діапазон, що забезпечується згладжувальною ємністю C7.

На рисунку 3.2.4. показана принципова схема керування електродвигунами.

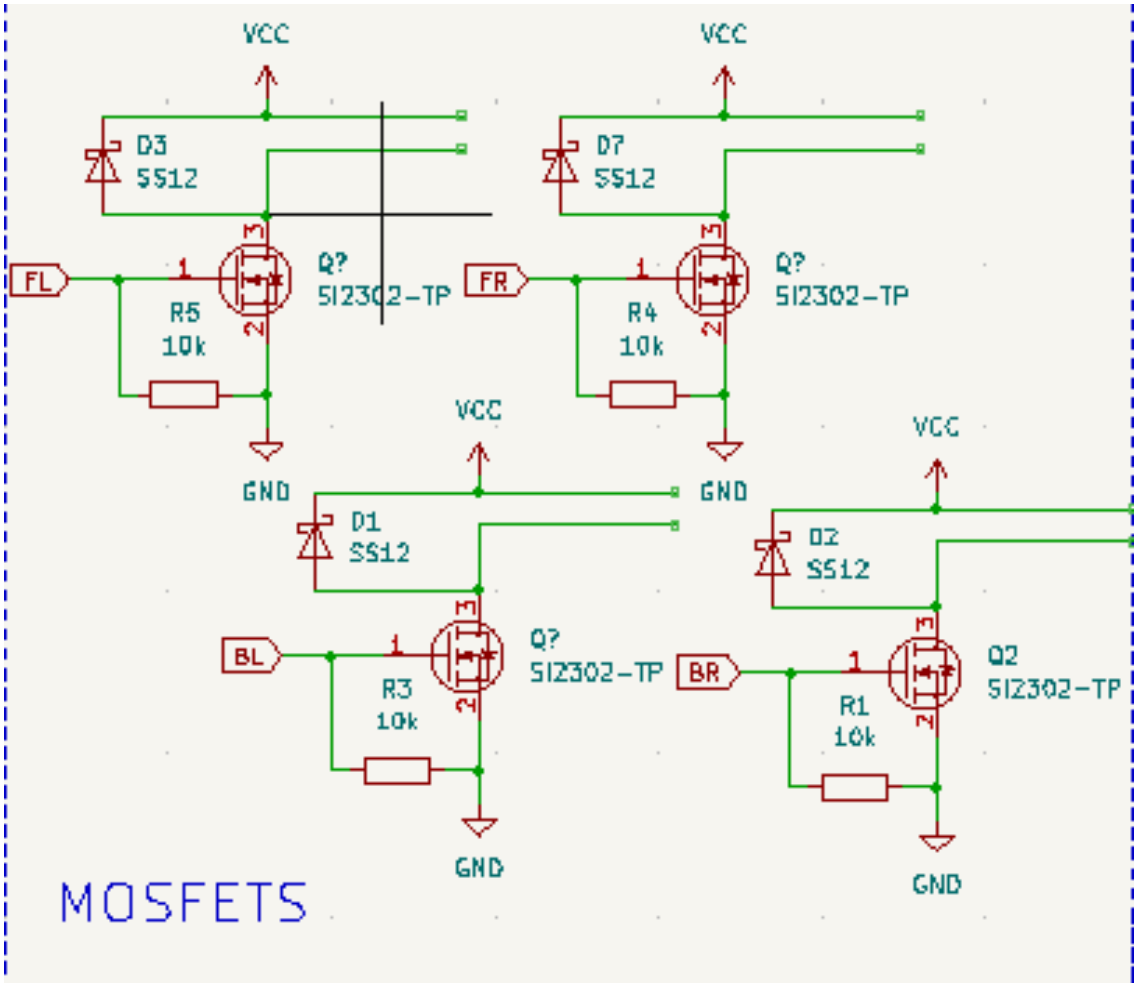

Рис. 3.2.4. Схема керування колекторними двигунами

Мікроконтролери AVR не можуть безпосередньо керувати двигунами, струм споживання яких значно перевищує максимальний струм виводу порта. Для підсилення струму використовуються транзистори на основі

метал-оксидного напівпровідника (MOSFET). Польові транзистори використані для зручності керування та збільшення ККД. Для уникнення пробою транзисторів напругою самоіндукції обмоток двигуна використовуються антипаралельні діоди.

На Рис. 3.2.5. показаний інтерфейс підключення гіроскопа/акселерометра MPU6050, для чого використовується апаратна шина I^2C мікроконтролера.

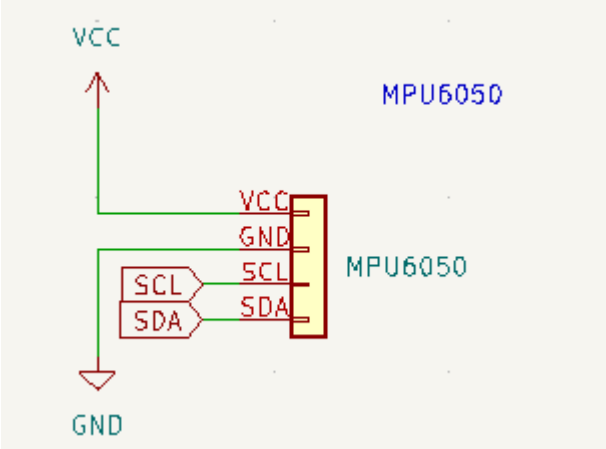

Рис. 3.2.5 Шина I^2C для підключення MPU6050

На Рис. 3.2.6. показаний інтерфейс SPI за допомогою якого відбувається обмін даними між мікроконтролером та радіомодулем (рис. 3.2.7). Окрім того, він використовується для програмування пам’яті мікроконтролера (для цього додатково використовується лінія RESET мікроконтролера).

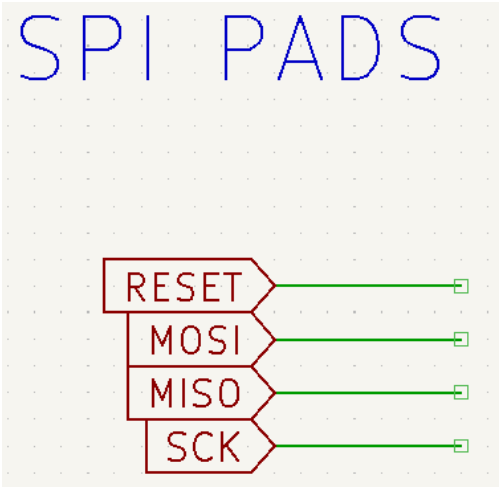

Рис. 3.2.6. Інтерфейс SPI для підключення радіомодуля та програмування мікроконтролера

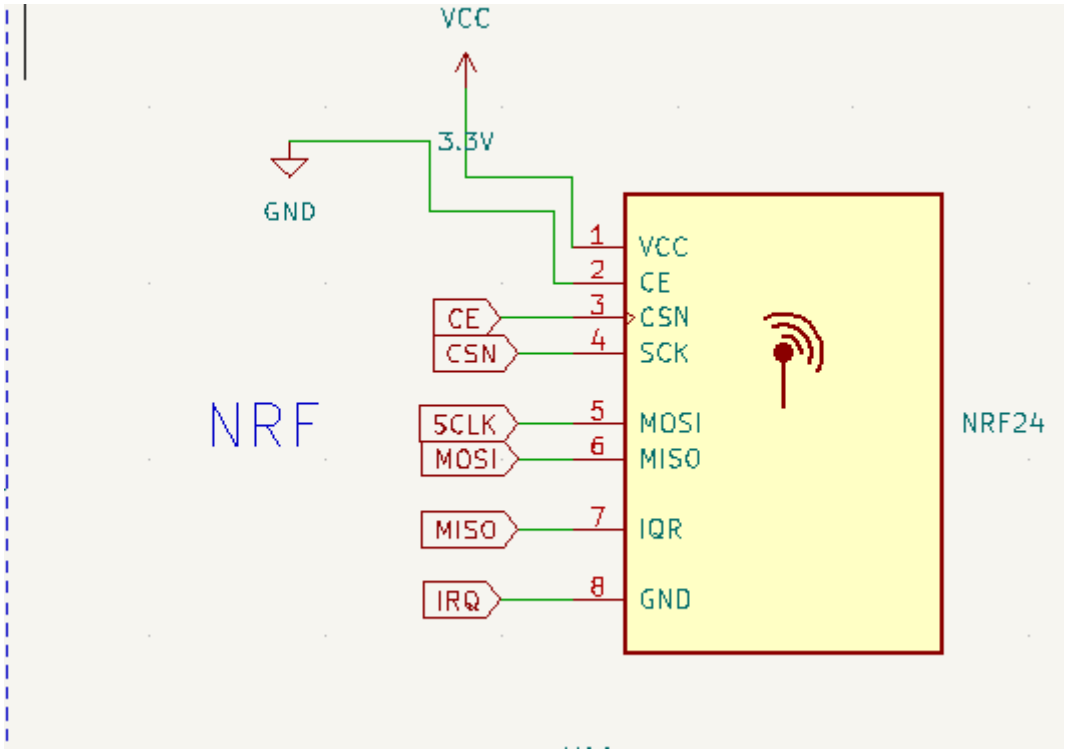

Рис. 3.2.7. Підключення радіомодуля NRF24

Окрім інтерфейсу SPI радіомодуль NRF24 використовує вхід зовнішнього переривання мікроконтролера IRQ.

На Рис. 3.2.8. показано підключення до апаратного інтерфейсу UART мікроконтролера.

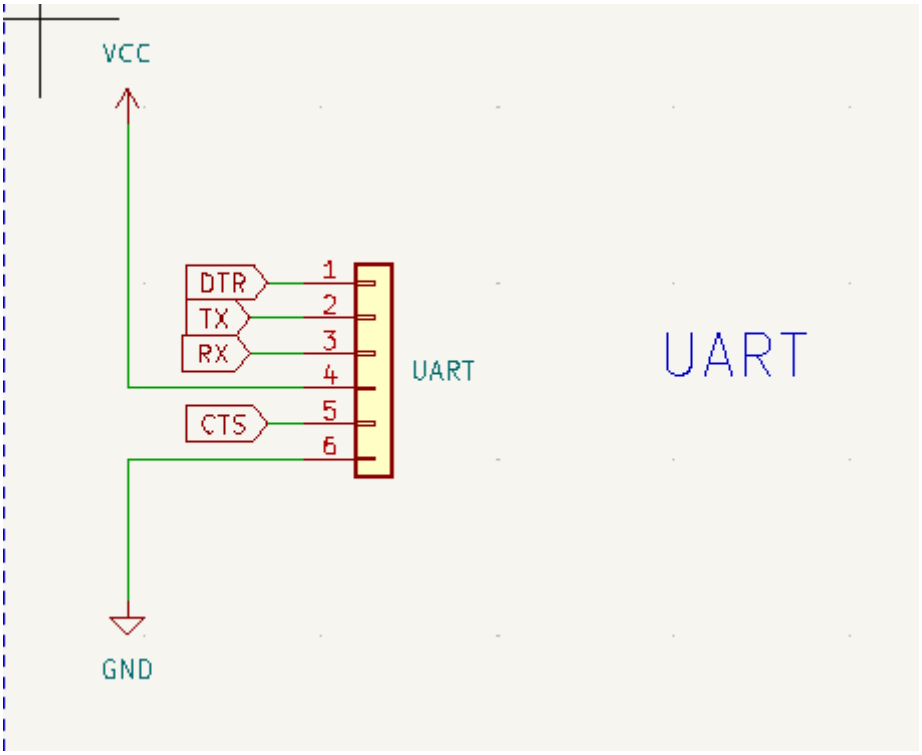

Рис. 3.2.8. Інтерфейс UART

Інтерфейс UART використовується для відлагодження коду, а також, для можливості підключення додаткових апаратних засобів, наприклад, Bluetooth модулів (рис. 3.2.9).

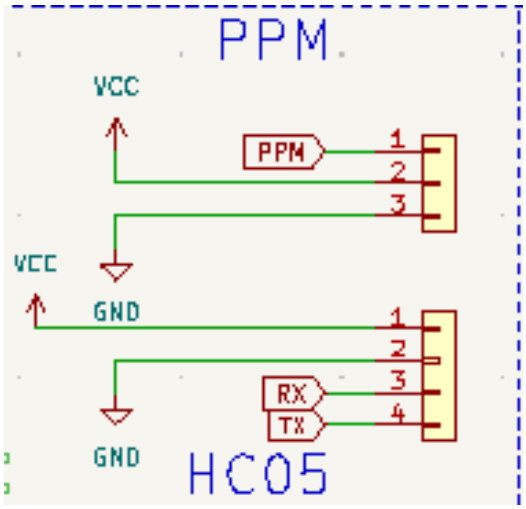

Рис. 3.2.9. Інтерфейс розширення апаратної можливості керування квадрокоптера

На Рис. 3.2.9 зображені дві можливі модифікації квадрокоптера PPM дешифратор та модуль Bluetooth. За допомогою першого з'являється можливість підключення модуля автопілота, а другий може забезпечити підключення до мобільного пристрою.

Для зручності налагодження коду та для індикації станів передбачено використання чотирьох світлодіодів (Рис. 3.2.10).

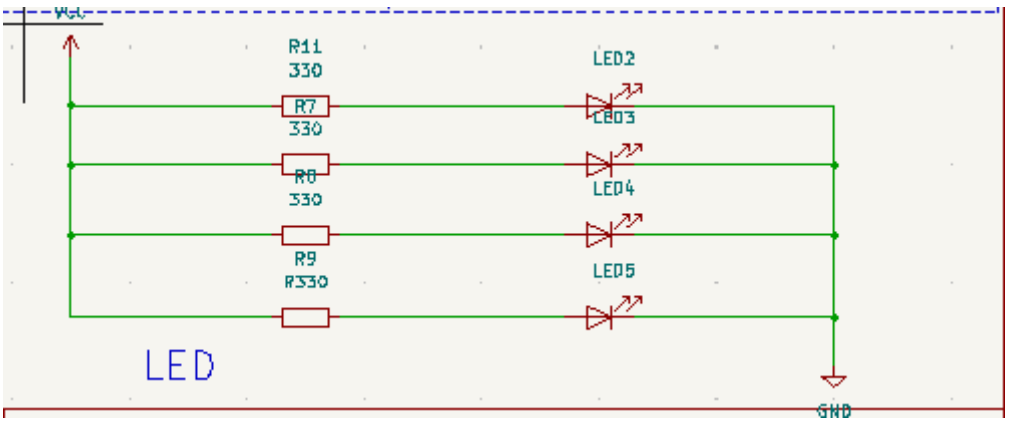

Рис. 3.2.10. Світлодіодна індикація станів

На Рис. 3.2.11. зображена принципова схема контролера керування квадрокоптером (схема пульта керування). Оскільки частина елементів була описана у попередніх пунктах, далі буде слідувати пояснення тільки деяких частин.

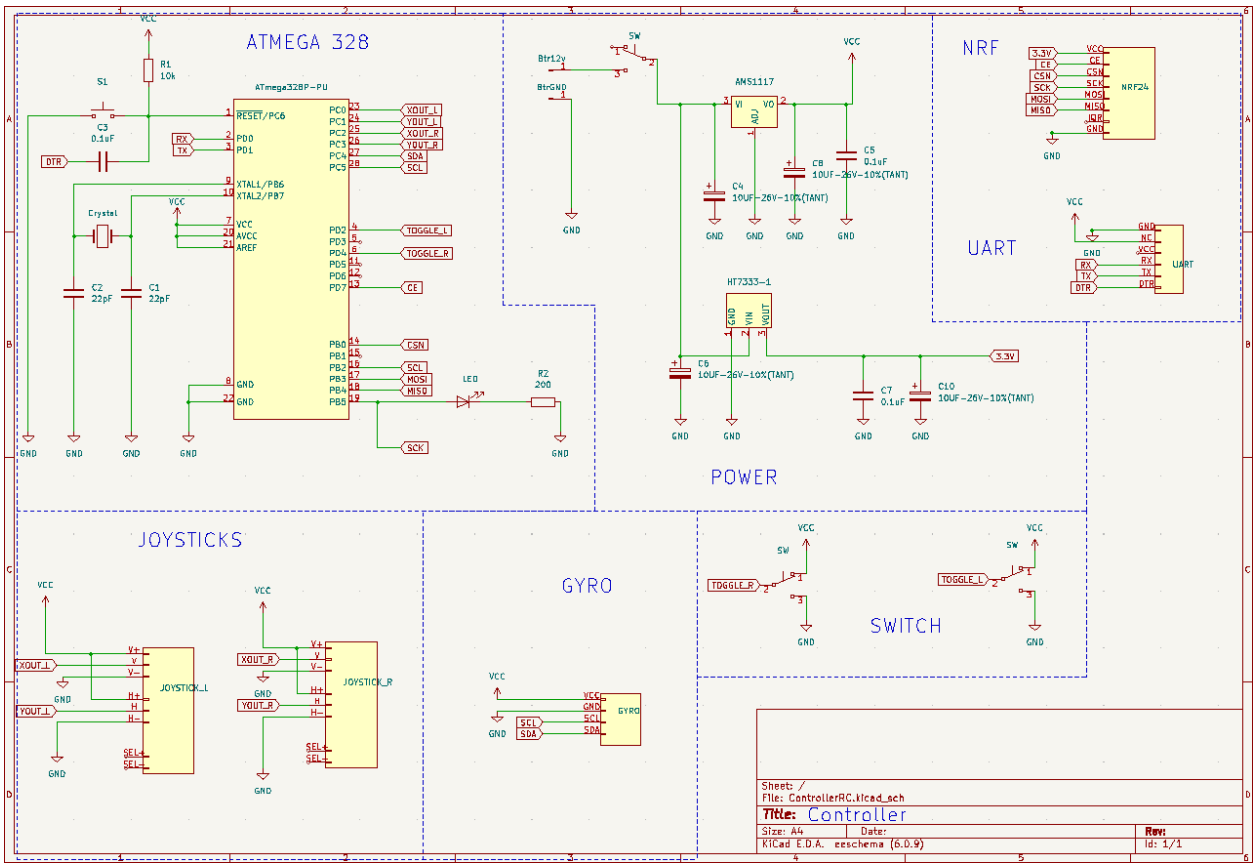

Рис. 3.2.11 Принципова схема контролера керування (пульта керування)

Основою схеми є той же мікроконтролер ATmega328, який обробляє сигнали джойстика, кнопок та гіроскопа (для керування квадрокоптером шляхом зміни нахилу пульта). Мікроконтролер обімнюється даними з платою квадрокоптера за допомгою того ж радіомодуля NRF24.

На рис. 3.2.12. зображена схема живлення нашого контролера. Оскільки було прийнято рішення для живлення контролера використати батарею з напругою 9В (Крона), то в з’явилась необхідність у другому стабілізаторі напруги (5В) для живлення мікроконтролера. Для збільшення дальності керування необхідно налаштувати радіомодуль на максимальну потужність дальності, що однак збільшить енергоспоживання.

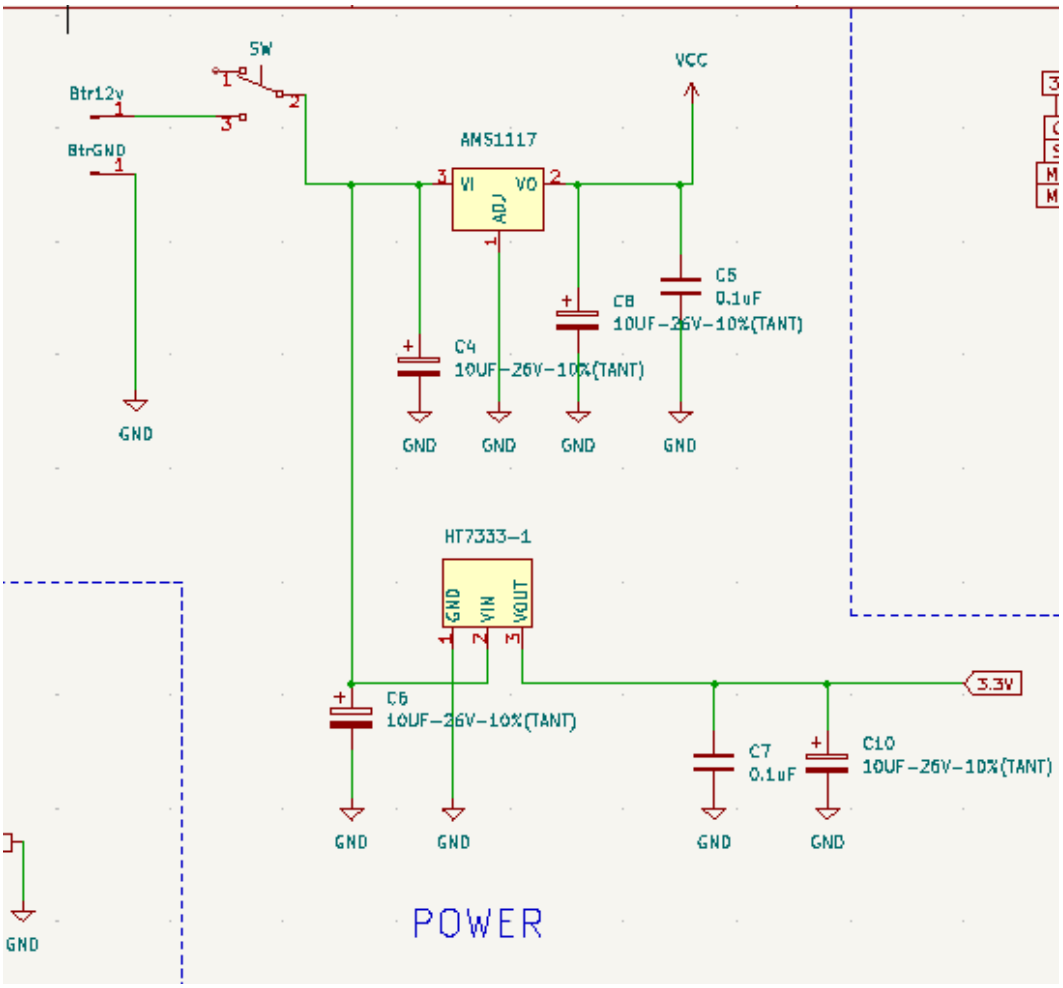

Рис. 3.2.12 Схема живлення контролера

Для керування напрямком польоту квадрокоптера використовується аналогові джойстики (Рис. 3.2.13) та два перемикачі режимів (Рис. 3.2.14). Один перемикач відповідає за включення моторів, інший- за режим польоту. Наявність стабілізатора напруги живлення дозволяє безперешкодно зчитувати аналоговий сигнал з джойстиків за допомогою внутрішнього АЦП мікроконтролера.

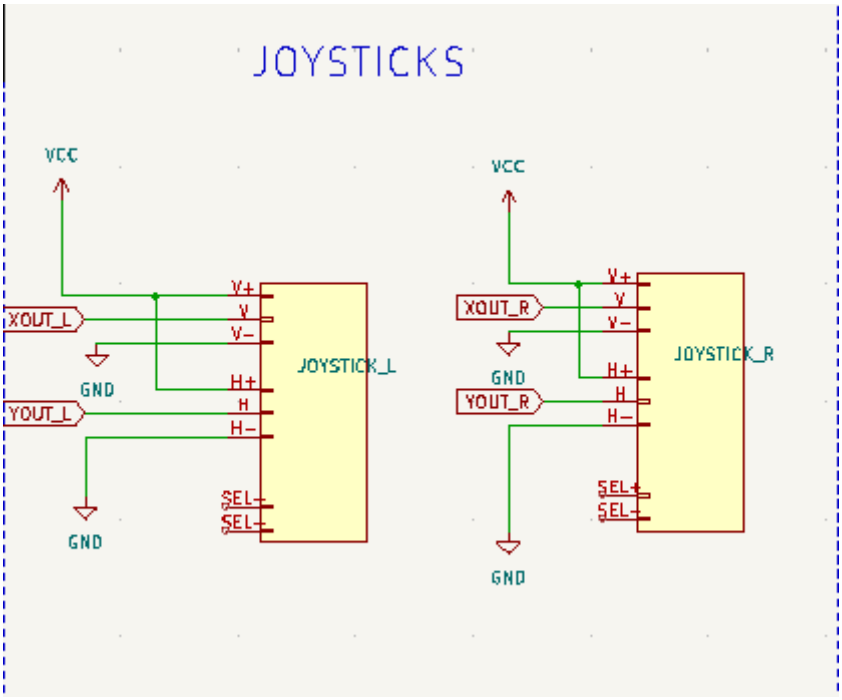

Рис. 3.2.13. Схема підключення джойстиків

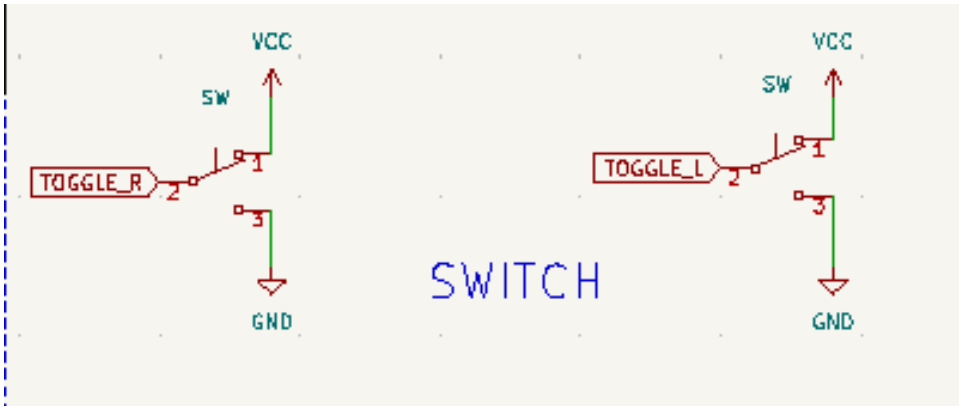

Рис. 3.2.14. Схема підключення перемикачів режимів польоту

На Рис. 3.2.15. показана схема підключення гіроскопу, за допомогою якого можна керувати польотом дрона шляхом зміни нахилу пульта керування, та змінювати висоту польоту надаючи пульту прискорення у вертикальному напрямку.

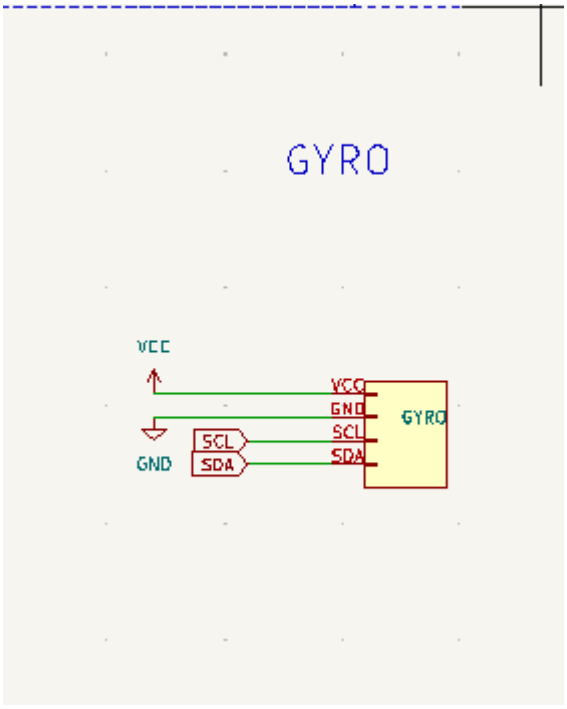

Рис. 3.2.15. Схема підключення гіроскопа / акселерометра

3.3 Програмування логіки керування квадрокоптером

У якості програмного забезпечення керування квадрокоптером було обрано MultiWii. MultiWii (Рис. 3.3.1) є проектом відкритого коду, який забезпечує програмне забезпечення для створення і управління багатоосевими літальними апаратами, такими як квадрокоптери, гексакоптери та інші.

Програмне забезпечення MultiWii розроблено для використання з мікроконтролерами, такими як Arduino, і може бути налаштовано для використання з різними датчиками руху та радіоуправлінням. Воно включає функції, такі як автоматичне вирівнювання, стабілізація положення, управління поворотом і багато інших.

Вибір MultiWii як програмного забезпечення для керування квадрокоптером може бути зумовлений декількома факторами:

1. Відкритий код: MultiWii є вільнодоступним програмним забезпеченням, що надає можливість модифікації і адаптації його функціональності до специфічних потреб користувача.
2. Сумісність з Arduino: MultiWii було розроблено спеціально для використання з платами Arduino, які є широко доступними та популярними серед DIY-спільноти.
3. Підтримка різноманітних датчиків: MultiWii сумісне з великою кількістю датчиків та аксесуарів, що відкриває широкі можливості для налаштування квадрокоптера.
4. Розширені можливості керування: MultiWii включає в себе функції автоматичного вирівнювання, стабілізації положення, управління поворотом і інші.

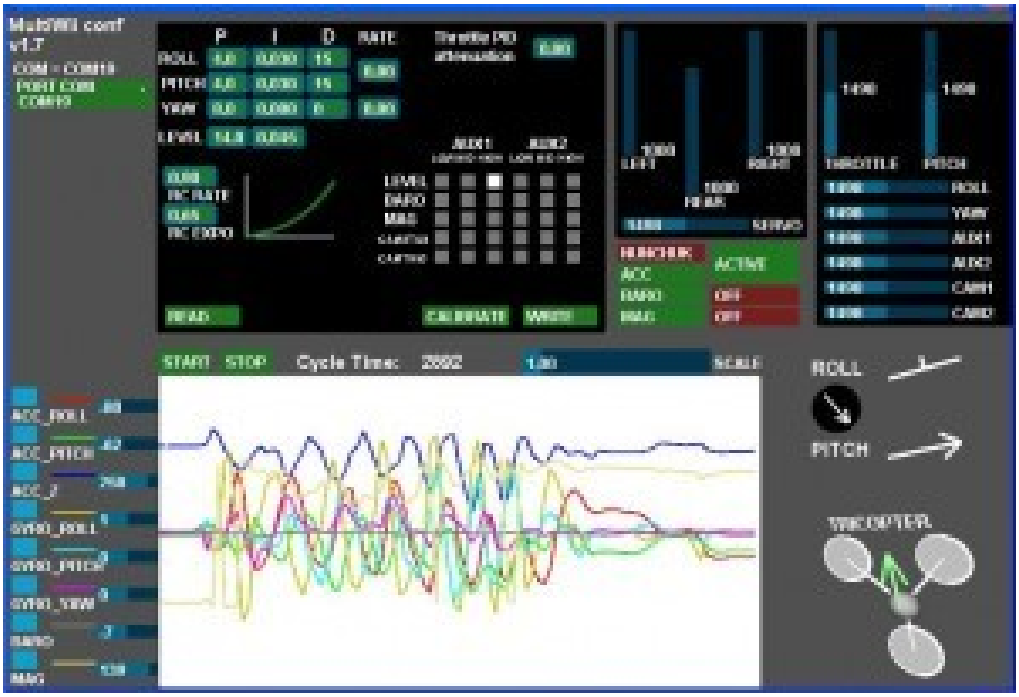

Рис. 3.3.1. Інтерфейс MultiWii [15]

Нижче показані фрагменти прокоментованого програмного коду мікроконтролера. Цей код є прикладом програми для контролера квадрокоптера, який використовує модуль бездротового зв'язку nRF24L01 та

два аналогові джойстика для управління газом, рулем, тангажем і креном. Це код для передавача

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
```

Рис. 3.2.2. Бібліотеки

На Рис. 3.2.2. показані використані бібліотеки. SPI.h: Стандартна бібліотека Arduino для комунікації з SPI (Serial Peripheral Interface) пристроями. nRF24L01.h та RF24.h: Бібліотеки для роботи з модулем бездротового зв'язку nRF24L01.

```
const uint64_t pipeOut = 0xE8E8F0F0E1LL;
```

Рис.3.2.3. Індивідуальний код-ключ

Індивідуальний код-ключ (Рис.3.2.3.), який присвоюється контролеру та дрону для ідентифікації один одного

```
struct MyData {
  byte throttle;
  byte yaw;
  byte pitch;
  byte roll;
  byte AUX1;
  byte AUX2;
};
```

Рис. 3.2.4. Структура даних

У даному випадку, MyData (Рис. 3.2.4.) використовується для зберігання даних контролю квадрокоптера. Ця структура містить наступні поля:

throttle (газ): це значення, яке контролює швидкість обертання двигунів квадрокоптера.

yaw (курс): визначає поворот квадрокоптера навколо вертикальної осі (зміна орієнтації ліворуч/праворуч).

pitch (тангаж): контролює поворот квадрокоптера навколо бокової осі (хіл квадрокоптера вперед/назад).

roll (крен): контролює поворот квадрокоптера навколо поперечної осі (хіл вліво/вправо).

AUX1 і AUX2: це додаткові канали, які можуть використовуватися для різноманітних цілей, в залежності від потреб користувача (наприклад, переключення режимів польоту, активація певних особливостей тощо).

Кожне поле цієї структури представляє канал керування квадрокоптера. Значення кожного каналу відправляється через радіопередавач до квадрокоптера.

```
void resetData()
{
  data.throttle = 0;
  data.yaw = 127;
  data.pitch = 127;
  data.roll = 127;
  data.AUX1 = 0;
  data.AUX2 = 0;
}
```

Рис. 3.2.5. Початкові значення

Функція resetData() (Рис. 3.2.5) використовується для встановлення початкових значень керування для квадрокоптера. В цьому випадку:

data.throttle = 0; — встановлює потужність двигунів (газ) на 0, тобто двигуни повинні бути вимкнені.

data.yaw = 127; — встановлює початкове значення курсу (yaw) в 127, що відповідає нейтральній позиції (без обертання вліво або вправо).

data.pitch = 127; і data.roll = 127; — встановлюють початкові значення тангажу (pitch) і крену (roll) в 127, що також відповідає нейтральній позиції (без нахилу вперед/назад або вліво/вправо).

data.AUX1 = 0; і data.AUX2 = 0; — встановлюють значення додаткових каналів у 0, що може використовуватися для вимкнення певних додаткових функцій квадрокоптера.

Ця функція використовується в початку програми для ініціалізації керування квадрокоптером в нейтральному стані.

```
void setup()
{
  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(pipeOut);
  resetData();
}
```

Рис. 3.2.6. Ініціалізація параметрів передавача

Цей фрагмент коду використовується для ініціалізації параметрів передавача NRF24L01 та встановлення початкових значень керування. Ця функція setup() (Рис. 3.2.6) виконується один раз при запуску програми на Arduino.

Ось що робить кожна з вказаних команд:

radio.begin(); — ця команда ініціалізує радіомодуль NRF24L01.

radio.setAutoAck(false); — ця команда вимикає автоматичну відповідь ACK. ACK - це підтвердження отримання даних, яке відправляється від одного пристрою до іншого. Цей режим може бути вимкнений, якщо ви не хочете отримувати підтвердження відповіді.

radio.setDataRate(RF24_250KBPS); — встановлює швидкість передачі даних радіомодулю NRF24L01 на 250 кілобіт за секунду.

radio.openWritingPipe(pipeOut); — відкриває канал для відправки даних. Значення pipeOut визначає адресу каналу, на якому будуть передаватися дані.

resetData(); — викликається функція resetData(), яка встановлює початкові значення керування для квадрокоптера, як було описано раніше.

Ця функція викликається при першому запуску програми і встановлює всі потрібні параметри для радіомодулю та значення керування.

```
int mapJoystickValues(int val, int lower, int middle, int upper, bool reverse)
{
  val = constrain(val, lower, upper);
  if ( val < middle )
    val = map(val, lower, middle, 0, 128);
  else
    val = map(val, middle, upper, 128, 255);
  return ( reverse ? 255 - val : val );
}
```

Рис. 3.2.7. Перетворення значень

Ця функція mapJoystickValues() (Рис. 3.2.7) використовується для перетворення значень, отриманих від джойстика, до відповідного діапазону значень для керування квадрокоптером.

Аргументи цієї функції:

val - вхідне значення від джойстика, яке потрібно перетворити.

lower, middle, upper - це значення, які представляють нижню, середню та верхню границі вхідного діапазону значень від джойстика.

reverse - булевий аргумент, який вказує, чи повинно бути зворотнє відображення значень.

В цій функції:

val = constrain(val, lower, upper); - обмежує вхідне значення між нижньою і верхньою границями.

if (val < middle) val = map(val, lower, middle, 0, 128); - якщо вхідне значення менше середнього, то воно перетворюється в діапазон від 0 до 128.

else val = map(val, middle, upper, 128, 255); - якщо вхідне значення більше або дорівнює середньому, то воно перетворюється в діапазон від 128 до 255.

return (reverse ? 255 - val : val); - якщо флаг reverse встановлено в true, то значення інвертується (віднімається від 255), в іншому випадку повертається без змін.

Ця функція використовується для забезпечення правильного відображення значень джойстика на значення керування квадрокоптером.

```
void loop()
{
  data.throttle = mapJoystickValues( analogRead(A1), X1, Y1, Z1, true );
  data.yaw      = mapJoystickValues( analogRead(A0), X2, Y2, Z2, true );
  data.pitch    = mapJoystickValues( analogRead(A3), X3, Y3, Z3, true );
  data.roll     = mapJoystickValues( analogRead(A2), X4, Y4, Z4, true );
  data.AUX1     = digitalRead(4);
  data.AUX2     = digitalRead(2);

  radio.write(&data, sizeof(MyData));
}
```

Рис. 3.2.8. Читання значень з джойстика

Кожний рядок коду, який починається з data., читає значення з джойстика, перетворює це значення за допомогою функції mapJoystickValues() та зберігає результат у відповідному полі структури MyData.

data.throttle, data.yaw, data.pitch, та data.roll отримують значення з аналогових входів A1, A0, A3, та A2 відповідно. Ці значення потім перетворюються за допомогою функції mapJoystickValues(), де X1, Y1, Z1 (та інші відповідні триади) - це специфічні для вашої системи калібрувальні значення.

data.AUX1 та data.AUX2 - це додаткові канали керування, які читають стани від цифрових входів 4 та 2 відповідно. Значення з цих входів можуть бути використані для додаткового функціоналу, наприклад, переключення режимів польоту.

radio.write(&data, sizeof(MyData)); - цей рядок передає дані, зібрані та оброблені вище, через радіомодуль NRF24L01. &data - це адреса в пам'яті, де зберігаються дані, а sizeof(MyData) - це розмір цих даних у байтах.

ВИСНОВКИ

Провівши аналіз платформи Arduino, було виявлено, що вона є досить гнучка, а мікроконтролер AVR, який є її основою, досить потужний для побудови багатьох технічних рішень. Саме через це було обрано написання дипломної роботи на основі Arduino.

У цій дипломній роботі було проведено дослідження та розроблено систему керування квадрокоптером на основі плати Arduino. Було вивчено основні принципи керування квадрокоптерами, а також переваги та характеристики різних компонентів, які використовуються в цих системах.

Виконано обґрунтований вибір компонентів для квадрокоптера, включаючи Arduino Uno, MPU6050, NRF24L01, щіткові двигуни, а також батарею LiPo. Також було розроблено детальну схему підключення цих компонентів за допомогою програми KiCad 6.

Окрім того, була обрана програма керування MultiWii, заснована на відкритому вихідному коді, яка надає гнучкі можливості для налаштування та керування квадрокоптером. Використовуючи це програмне забезпечення, вдалось налаштувати квадрокоптер на роботу з нашими специфічними компонентами та параметрами.

Під час роботи було створено та протестовано програмне забезпечення для контролера, зокрема на стабільність зв'язку.

Враховуючи поточний стан технології квадрокоптерів, даний дипломний проект має велику актуальність. Існує велика кількість потенційних застосувань для систем, подібних до тієї, яку було розроблено в цій роботі, включаючи, але не обмежуючись: зйомку відео та фотографій, моніторинг об'єктів, доставку вантажів, рятувальні операції та пошуку, а також військові застосування.

Загалом, проект був успішним, а отримані результати є обнадійливими для подальшого розвитку та вдосконалення системи керування квадрокоптером на основі Arduino. Він показує, що з допомогою доступних

та відносно дешевих компонентів можна створити ефективну та гнучку систему керування квадрокоптером.

Однак, на основі зроблених випробувань були зафіксовані певні недоліки системи, що потребує подальшого удосконалення, а саме:

1. Необхідність вдосконалення системи стабілізації: Хоча в цій роботі було розроблено базову систему стабілізації, можливе додаткове удосконалення за допомогою більш складних алгоритмів контролю, як-то PID контролери, або застосування методів машинного навчання.
2. Розширення функціональності: Додаткові функції, такі як GPS-навігація, автоматичне повернення до дому, відстеження об'єктів або автономне планування маршрутів, можуть бути впроваджені в систему.
3. Енергоефективність: Оптимізація використання енергії може значно збільшити час роботи квадрокоптера. Це може бути досягнуто шляхом розробки більш ефективних алгоритмів керування або використанням більш ефективних компонентів.
4. Безпека та надійність: Важливим аспектом є забезпечення безпеки польоту квадрокоптера. Однією з можливих вдосконалень може бути розробка системи аварійного відключення або заходів безпеки для захисту від зовнішніх впливів.
5. Розробка кастомного апаратного забезпечення: Впровадження спеціалізованої апаратної платформи, замість універсальної плати Arduino, може дати можливість кращого адаптування системи до конкретних вимог та умов.
6. Інтеграція відеокамери: інтеграція камери в квадрокоптер здатна значно розширити його функціональні можливості. Камера може використовуватися для обробки зображень в реальному часі і використовуватися для визначення положення квадрокоптера відносно навколишнього середовища. Інтегрована камера може транслювати відео в реальному часі на пульт управління, дозволяючи оператору керувати квадрокоптером з першої особи.

Ці напрямки дослідження можуть допомогти підвищити ефективність, надійність та безпеку системи керування квадрокоптером, а також розширити її можливі застосування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kyaw Myat Thua , A.I. Gavrilov. Designing and modeling of quadcopter control system using L1 adaptive control. Moscow, 2016. p. 8
2. Omkar Tatale, Nitinkumar Anekar, Supriya Phatak, Suraj Sarkale, Quadcopter: Design, Construction and Testing, Maharashtra, India. 2018. p. 7
3. Martí POMÉS ARNAU, Model Based Control of Quadcopters, Lausanne. 2016. p. 59
4. ATMEGA328P-PU. Datasheet, 2010. p. 32
5. John Boxall,AVR Workshop,2022. p. 376
6. cfdflowengineering[Електронний ресурс]]. – Режим доступу до ресурсу: https://cfdflowengineering.com/working-principle-and-components-of-drone/
7. Сучасний квадрокоптер DJI Air 2S[Електронний ресурс]]. – Режим доступу до ресурсу: https://store.dji.com/product/dji-air-2s?vid=104122
8. Український дрон-камікадзе eBOSH [Електронний ресурс]]. – Режим доступу до ресурсу: https://dev.ua/news/dron-ebosh-ta-motopes-myslyvets
9. Ретранслятор DJI D-RTK 2 [Електронний ресурс]]. – Режим доступу до ресурсу: https://store.dji.com/product/d-rtk-2-high-precision-gnss-mobile-station
10. Arduino Uno [Електронний ресурс]]. – Режим доступу до ресурсу: https://learn.sparkfun.com/tutorials/what-is-an-arduino/all
11. Безщіткові мотори [Електронний ресурс]]. – Режим доступу до ресурсу: https://hobbyking.com/en_us/brushed-motor-4-pcs-per-set.html?___store=en_us
12. Сенсор MPU6050 [Електронний ресурс]]. – Режим доступу до ресурсу: https://arduino.ua/prod512-akselerometr-i-giroskop-mpu-6050-modyl-6dof
13. NRF23L01 [Електронний ресурс]. – Режим доступу до ресурсу: https://arduino.ua/prod231-radiomodyl-nrf24l01-2-4-ggc

14. Акумулятор LiPol [Електронний ресурс]. – Режим доступу до ресурсу: https://www.deltakit.net/product/3-7v-1100mah-lipo-battery/
15. Інтерфейс MultiWii [Електронний ресурс]. – Режим доступу до ресурсу: http://www.multiwii.com/

ДОДАТКИ

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

const uint64_t pipeOut = 0xE8E8F0F0E1LL;

RF24 radio(7, 8);


struct MyData {
  byte throttle;
  byte yaw;
  byte pitch;
  byte roll;
  byte AUX1;
  byte AUX2;
};

MyData data;

void resetData()
{
  data.throttle = 0;
  data.yaw = 127;
  data.pitch = 127;
  data.roll = 127;
  data.AUX1 = 0;
  data.AUX2 = 0;
}

void setup()
{
  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(pipeOut);
  resetData();
}

/**************************************************/

int mapJoystickValues(int val, int lower, int middle, int upper, bool reverse)
{
  val = constrain(val, lower, upper);
  if ( val < middle )
    val = map(val, lower, middle, 0, 128);
  else
    val = map(val, middle, upper, 128, 255);
  return ( reverse ? 255 - val : val );
}

void loop()
{
  data.throttle = mapJoystickValues( analogRead(A1), 13, 524, 1015, true );
  data.yaw      = mapJoystickValues( analogRead(A0),  1, 505, 1020, true );
  data.pitch    = mapJoystickValues( analogRead(A3), 12, 544, 1021, true );
  data.roll     = mapJoystickValues( analogRead(A2), 34, 522, 1020, true );
  data.AUX1     = digitalRead(4);
  data.AUX2     = digitalRead(2);

  radio.write(&data, sizeof(MyData));
}
```