Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Комп'ютеризована система керування FPV дроном для симулятора польоту

Виконав(ла): студент(ка) *IV* курсу, групи *CI-41*
спеціальності *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

Міндзіброцький Р.С.
(підпис) (прізвище та ініціали)

Керівник Лещишин Ю.З.
(підпис) (прізвище та ініціали)

Нормоконтроль Тиш Є.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Осухівська Г.М.
(підпис) (прізвище та ініціали)

Рецензент Млинко Б.Б.
(підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
« ____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *бакалавр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту Міндзіброцькому Роману Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система керування FPV дроном для симулятора польоту

Керівник роботи *Лещишин Юрій Зіновійович, к.т.н., доцент кафедри КС*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*24*» *квітня* *2024* року № *4/7-408*

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
1. Аналіз технічного завдання
2. Проектна частина
3. Практична частина
4. Безпека життєдіяльності, основи охорона праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	*Пилипець М.І., д.т.н., проф. каф. МТ*		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Розробка технічного завдання*	*24.04-27.04.2024*	*Виконано*
2	*Аналіз технічного завдання*	*28.04-01.05.2024*	*Виконано*
3	*Аналіз вимог до системи керування FPV дроном*	*02.05-10.05.2024*	*Виконано*
4	*Вибір елементної бази*	*11.05-13.05.2024*	*Виконано*
5	*Розробка структурної та функціональної схеми*	*14.05-16.05.2024*	*Виконано*
6	*Розробка програмного забезпечення для проектованої частини*	*17.05-27.05.2024*	*Виконано*
7	*Тестування системи керування FPV дроном*	*28.05-30.05.2024*	*Виконано*
8	*Оформлення кваліфікаційної роботи*		
9	*Попередній захист кваліфікаційної роботи*		
10	*Захист кваліфікаційної роботи*		

Студент _____ Міндзіброцький Р.С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Лещишин Ю.З.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризованої системи керування FPV дроном для симулятора польоту // Кваліфікаційна робота бакалавра // Міндзіброцький Роман Сергійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних систем та мереж, група СІ-41 // Тернопіль, 2024 // с. – 85, рис. – 45, табл. – 4, додат. – 3, бібліогр. – 20.

Ключові слова: комп'ютеризована система, керування, fpv дрон, симулятор польоту, технології дронів, програмне забезпечення, алгоритми керування, відтворення польоту.

Кваліфікаційна робота присвячена розробці комп'ютеризованої системи керування FPV дроном для симулятора польоту. Дослідження охоплює розробку спеціалізованого контролера, що може підключатися до існуючих програмних середовищ для симуляції польоту безпілотних літальних апаратів. В результаті аналізу сучасних засобів і технологій керування FPV дроном для симуляторів польоту було виявлено, що одним з найперспективніших напрямків є розробка контролера з використанням сучасних комп'ютерних технологій для забезпечення точного і реалістичного керування дроном. Було створено структурну схему комп'ютеризованої системи керування, яка дозволяє інтегрувати контролер з різними симуляторами польоту, а також забезпечує можливість налаштування під конкретні вимоги користувача. Розроблено електричну принципову схему контролера для FPV дронів, а також програмне забезпечення для його функціонування, що включає алгоритми обробки даних і взаємодії з симулятором польоту.

ANNOTATION

Computerized FPV drone flight simulator control system // Bachelor`s thesis // Mindzibrodskyi Roman // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Systems and Networks, group CI-41 // Ternopil, 2024 // p. – 85, fig. – 45, table. – 4, ref. – 20.

Key words: computerized system, control**,** fpv drone, flight simulator, drone technologies, software, control algorithms, flight reproduction.

The qualification work is dedicated to the development of a computerized control system for an FPV drone used in a flight simulator. The research focuses on designing a specialized controller that can connect to existing software environments for simulating unmanned aerial vehicle (UAV) flights. The analysis of current tools and technologies for controlling FPV drones in flight simulators revealed that one of the most promising directions is the development of a controller utilizing modern computing technologies to ensure precise and realistic drone control. A structural scheme of the computerized control system was created, allowing the integration of the controller with various flight simulators and providing the ability to customize settings according to user-specific requirements. An electrical schematic of the controller for FPV drones was developed, as well as the software for its operation, which includes data processing algorithms and interaction with the flight simulator.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ 10
1.1 Аналіз вимог до комп’ютеризованої системи 10
1.2 Мета створення системи 11
1.3 Огляд існуючих варіантів вирішення завдання 12
1.4 Основні завдання та функції системи 15
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА 17
2.1 Розробка структури комп’ютеризованої системи 17
2.2 Реалізація апаратної частини комп’ютеризованої системи керування FPV дроном для симулятора польоту 20
2.2.1 Опис плати STM32F411CCU6 20
2.2.1 Опис IPS-дисплея ST7789 23
2.2.3 Опис радіомодуля NRF24L01 25
2.2.3 Опис адаптера для NRF24L01 28
2.2.4 Опис модуля джойстика 30
2.2.5 Опис перемикача 31
2.3 Опис схеми електричної принципової комп’ютеризованої системи керування FPV дроном для симулятора польоту 32
2.4 Обґрунтування вибору середовища розробки ПЗ для реалізації системи керування FPV дроном для симулятора польоту 35
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА 37
3.1 Алгоритм роботи системи керування FPV дроном для симулятора польоту 37
3.2 Створення проєкту в STM32CubeIDE 40

3.3 Розробка ПЗ для приймача 41
3.4 Розробка ПЗ для передавача 47
3.5 Тестування системи керування FPV дроном для симулятора польоту 54
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ 57
4.1 Навчання працюючих і інструктажі з охорони праці 57
4.2 Заходи щодо захисту установки від короткого замикання 59
4.3 Можливість виникнення статичної електрики та заходи боротьби з нею 61
ВИСНОВКИ 63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 64
Додаток А Технічне завдання 66
Додаток Б Перелік елементів 74
Додаток В Лістинг програмного забезпечення 79

СПИСОК СКОРОЧЕНЬ

ADC – Analog Digital Convertor

DMA – Direct Memory Access

FPV – First Person View

GPIO – General Purpose Input/Output

HID – Human Interface Device

IDE – Integrated Development Environment

IPS – In-Plane Switching

LCD – Liquid Crystal Display

SPI – Serial Peripheral Interface

USB – Universal Serial Bus

АЦП – Аналого-цифровий перетворювач

ПЗ – Програмне забезпечення

ПК – Персональний комп’ютер

ВСТУП

У сучасному світі безпілотні дрони, стають невід'ємною частиною різних галузей, від розваг до промислових застосувань. Одним із найбільш популярних видів дронів є FPV дрони, які дозволяють оператору керувати польотом у реальному часі, використовуючи відеозображення з камери, встановленої на дроні.

Однією з головних проблем, з якими стикаються оператори безпілотників FPV, є висока складність керування, особливо для початківців. Сучасні тренажери та симулятори польотів можуть значно спростити процес навчання, дозволяючи оператору безпечно відпрацьовувати навички керування віртуальним дроном.

Саме тому, актуальною задачею є створення системи керування FPV дроном для симулятора польоту. Така система не лише сприятиме ефективному навчанню нових пілотів, але й дозволить професіоналам удосконалювати свої навички в безпечному середовищі. Симулятор також може бути використаний для тестування нових технологій та алгоритмів управління без ризику пошкодження реального обладнання.

Метою даної кваліфікаційної роботи є розробка та впровадження комп'ютеризованої системи керування FPV дроном для симулятора польоту. Робота включатиме в себе дослідження сучасних підходів до моделювання польотів, розробку алгоритмів керування та їх інтеграцію з віртуальним середовищем, розробку програмного забезпечення та тестування.

Для досягнення мети були поставлені такі задачі:

- провести аналіз наявних систем, щоб визначити їхні сильні та слабкі сторони;
- створити апаратну частину системи, включаючи синтез структурної схеми та розробку електричної принципової схеми;
- розробити алгоритм функціонування комп'ютеризованої системи управління FPV дроном для симулятора польотів;

- написати програмне забезпечення для реалізації цього алгоритму, яке працюватиме на мікроконтролерах;
- виконати тестування створеної системи для оцінки її ефективності та надійності.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп’ютеризованої системи

Розробка системи керування FPV дроном вимагає ретельного аналізу технічного завдання для забезпечення високої точності та реалістичності моделювання польоту. Для початку необхідно детально визначити функціональні можливості системи, які включають керування дроном у режимі реального часу, моделювання поведінки дрона, а також точну передачу сигналів від користувача до дрона.

Основне завдання – розробка алгоритмів, які зможуть забезпечити адекватну реакцію дрона на команди користувача з урахуванням різних умов польоту. Для цього необхідно провести детальний аналіз існуючих алгоритмів керування, а також оцінити їх придатність для інтеграції в симулятор. Важливим аспектом є забезпечення точності даних датчиків, які використовуються для симуляції польоту, що вимагає ретельного вибору відповідного обладнання.

Важливо враховувати вимоги до програмного забезпечення, яке повинно бути сумісним з різними типами симуляторів і підтримувати оновлення та модифікації. Щоб забезпечити реалістичність моделювання, необхідно проаналізувати затримки в передачі сигналу, тому що навіть невеликі затримки можуть вплинути на точність управління дроном.

Аналіз технічного завдання включає також розгляд питань безпеки та надійності системи. Це передбачає реалізацію механізмів захисту від збоїв, які можуть виникнути під час роботи системи.

Крім того, аналіз охоплює системну інтеграцію з різними типами дронів і симуляторів, що вимагає оцінки апаратної та програмної сумісності, а також

масштабованості системи для підтримки нових моделей дронів.

Таким чином, розробка комп'ютеризованої системи керування FPV дроном для симулятора польоту вимагає всебічного аналізу технічних вимог і функціональних можливостей, що дозволяють забезпечити ефективне, реалістичне та надійне управління дроном під час симуляції.

1.2 Мета створення системи

Метою створення комп'ютеризованої системи керування FPV дроном для симулятора польоту є розробка системи керування FPV дроном, призначеної для інтеграції з симулятором польоту. Основні завдання цієї системи включають забезпечення точного та реалістичного управління дроном, що дозволить користувачам ефективно розвивати та покращувати свої навички пілотування в умовах, максимально наближених до реальних. Основними завданнями, які має виконувати система, є:

- система повинна відтворювати реалістичні відчуття керування дроном, забезпечуючи високу точність та швидкість реакції на команди;
- система повинна забезпечити повну сумісність із наявними симуляторами польоту, дозволяючи використовувати її без необхідності значних модифікацій програмного забезпечення. Контролер повинен інтегруватися з симулятором, передаючи команди в реальному часі;
- завдяки можливості детально відпрацьовувати маневри та реагувати на різні ситуації в безпечному віртуальному середовищі, користувачі зможуть значно підвищити свої навички пілотування. Система дозволить уникнути ризику пошкодження обладнання та надасть можливість багаторазово відпрацьовувати складні ситуації;
- система повинна бути стійкою до збоїв та забезпечувати безперебійну роботу протягом усього часу тренувань. Це включає використання надійних

апаратних та програмних компонентів, що мінімізують ризики технічних несправностей;

Загалом, основна мета створення системи полягає в розробці високоякісного контролера для FPV дрона, який забезпечить користувачам можливість безпечно та ефективно тренуватися у віртуальному середовищі, підвищуючи свої навички пілотування та підготовки до реальних польотів.

1.3 Огляд існуючих варіантів вирішення завдання

Для вирішення задачі керування FPV дроном у симуляторі польоту можна використовувати різноманітні методи і пристрої.

1.3.1 Spektrum InterLink DX RC Simulator Controller

Одним з найпоширеніших і ефективних варіантів є використання контролера Spektrum InterLink DX RC Simulator Controller (рис. 1.1) [6].

Рисунок 1.1 – Контролер Spektrum InterLink

Spektrum InterLink є високоякісним контролером, призначеним для використання з різноманітними симуляторами польоту. Основні характеристики цього контролера включають:

- завдяки використанню передової технології Spektrum DSMX, контролер забезпечує надзвичайно точне управління моделлю;
- контролер оснащений декількома програмованими перемикачами та регуляторами, що дозволяє налаштовувати його під конкретні потреби користувача і симулятора;
- Spektrum InterLink DX підтримує роботу з більшістю популярних симуляторів польоту, що робить його універсальним інструментом для тренувань.

Переваги:

- використання DSMX технології гарантує стабільний і точний зв'язок між контролером та симулятором;
- зручний дизайн дозволяє довго використовувати контролер без дискомфорту;
- велика кількість програмованих елементів керування дозволяє адаптувати контролер під будь-які потреби.

Недоліки:

- висока якість і технологічні можливості контролера відображаються на його вартості, що може бути недоліком для деяких користувачів;
- велика кількість налаштувань може бути складною для новачків, що потребує додаткового часу для освоєння.

Spektrum InterLink DX RC Simulator Controller активно використовується як новачками, так і досвідченими пілотами для відпрацювання навичок польоту без ризику пошкодження реального дрона. Завдяки можливості симулювати різноманітні умови польоту та ситуації, контролер допомагає користувачам покращувати свої навички управління, відпрацьовувати складні маневри і готуватися до реальних польотів.

1.3.2 Jumper T-Lite V2 ELRS M2

Ще одним із популярних варіантів контролерів є Jumper T-Lite V2 ELRS M2 (рис 1.2) [7]. Цей контролер забезпечує високу точність і стабільність управління завдяки вбудованому модулю ExpressLRS, який забезпечує низьку затримку і високу дальність зв'язку.

Рисунок 1.2 – Контролер Jumper T-Lite V2 ELRS M2

Основні характеристики:

- підтримка частотного діапазону 2.4 ГГц з використанням протоколу ExpressLRS;
- 1.3-дюймовий LCD дисплей для зручного налаштування та моніторингу параметрів;
- живлення від одного акумулятора типу 18650, що забезпечує тривалий час роботи;
- підтримка до 16 каналів для більш гнучкого управління;
- компактний і легкий дизайн, зручний для тривалого використання;

- USB-C порт для швидкого підключення до симулятора польоту на ПК.

Переваги:

- висока стабільність і точність управління, завдяки протоколу ExpressLRS, затримка сигналу мінімізована, що забезпечує швидку і точну реакцію дрона на команди;
- широкі можливості налаштування параметрів під індивідуальні потреби користувача;
- легкий і зручний для транспортування, що важливо для користувачів, які часто беруть контролер з собою;
- використання акумулятора 18650 забезпечує тривалий час автономної роботи без необхідності частого заряджання.

Недоліки:

- 16 каналів є достатнім для більшості користувачів, деяким просунутим користувачам може не вистачати додаткових каналів для більш складних завдань;
- відсутність сенсорного управління може бути незручною для деяких користувачів, які звикли до сучасних сенсорних інтерфейсів;
- контролер має широкі можливості налаштування, але він може бути складним для новачків, які не мають досвіду роботи з такими пристроями.

Jumper T-Lite V2 ELRS M2 широко використовується як серед аматорів FPV польотів, так і серед професіоналів. Його активно використовують для тренувань у симуляторах польоту, що дозволяє пілотам удосконалювати свої навички без ризику пошкодження дрона. Крім того, цей контролер часто застосовують на змаганнях з FPV польотів завдяки його високій надійності та швидкості реакції.

1.4 Основні завдання та функції системи

Система керування FPV дроном призначена для забезпечення ефективного та інтуїтивно зрозумілого управління дроном в умовах віртуального середовища. Основні завдання та функції цієї системи включають:

- система повинна забезпечувати високоточне керування польотом дрона, дозволяючи користувачу контролювати висоту, швидкість, напрямок та виконання різноманітних маневрів. Це включає як базові маневри, такі як підйом, спуск, рух уперед та назад, повороти, так і більш складні маневри віражі та інші трюки;
- пульт керування має бути оснащений зручним та інтуїтивно зрозумілим інтерфейсом, що дозволяє користувачу швидко адаптуватися до управління дроном у симуляторі;
- система повинна бути спроектована таким чином, щоб забезпечити легку та безперебійну інтеграцію з різноманітними симуляторами польоту, надаючи можливість повної функціональності та синхронізації між фізичними командами, які надходять від користувача, та їх реалізацією у віртуальному середовищі симулятора. Це включає забезпечення швидкої та коректної обробки команд, мінімізацію затримок, а також підтримку сумісності з різними моделями симуляторів;
- система повинна бути надійною та безпечною, що включає мінімізацію ризику виникнення помилок, забезпечення стабільної роботи під час тривалих тренувальних сесій, а також впровадження механізмів захисту від можливих збоїв чи некоректних дій користувача.

Ці завдання та функції дозволяють створити ефективний інструмент для навчання та відпрацювання навичок керування FPV дроном у віртуальному середовищі, що сприяє підвищенню компетенції користувачів та підготовці до реальних польотів.

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка структури комп’ютеризованої системи

На рис. 2.1 зображено узагальнену структурну схему комп’ютеризованої системи керування FPV дроном для симулятора польоту.

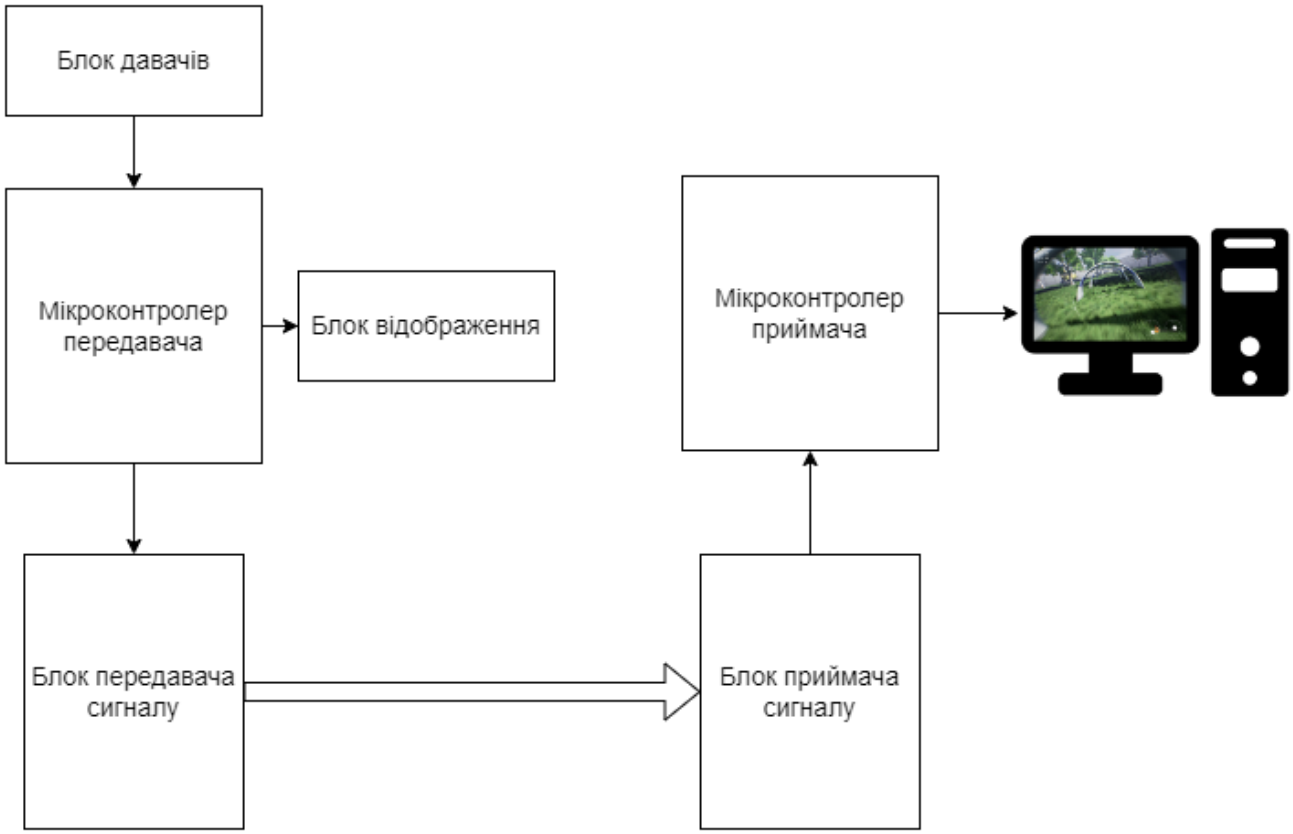

Рисунок 2.1 – Узагальнена функціональна схема комп’ютеризованої системи керування FPV дроном для симулятора польоту

Узагальнена функціональна схема комп’ютеризованої системи є інженерною системою, що включає в себе ряд важливих компонентів.

Основними компонентами цієї системи є давачі, такі як перемикачі та джойстики. Також мікроконтролери, блок передавача, блок відображення, блок приймача та персональний комп’ютер. Кожен з цих компонентів виконує важливу роль у забезпеченні ефективності та надійності системи.

Блок давачів відповідає за введення користувацьких команд і сигналів в систему. Він опитується мікроконтролером передавача для отримання інформації про стани та поточне положення. Ці дані обробляються мікроконтролером передавача, після цього оброблені дані відправляються на блок відображення і на блок передавача.

Блок відображення відображає інформацію про положення джойстиків і стани перемикачів, щоб користувач міг бачити візуальний зворотний зв'язок.

Блок приймача приймає сигнали від блоку передавача і передає його мікроконтролеру приймача для подальшої обробки.

Мікроконтролер приймача відправляє ці дані через порт USB до персонального комп’ютера, де вони можуть бути використані у симуляторі польоту.

Таким чином, взаємодія всіх цих компонентів узагальненої структурної схеми системи керування FPV дроном дозволяє забезпечити ефективне та точне управління дроном у симуляторі польоту.

Структурна схема демонструє основні компоненти пристрою та взаємозв’язок між ними. Вона відображає, як дані з давачів обробляються мікроконтролером, який потім передає їх для відображення на екрані та на блок передачі сигналу. Мікроконтролер приймача отримує ці дані за допомогою блоку приймача, обробляє та відправляє на комп’ютер за допомогою інтерфейсу USB.

Структурна схема схема комп’ютеризованої системи керування FPV дроном для симулятора польоту включає компоненти:

а) мікроконтролер – це головний модуль пристрою, який здійснює отримання, обробку та передачу даних. Він повинен бути запрограмований на

виконання різних функцій, таких як отримання даних з давачів, опрацювання цих даних, надсилання оброблених даних на дисплей та передавач.

б) давачі передають дані про свої стани, включаючи точне положення джойстиків та активність перемикачів, до мікроконтролера. Ця інформація є основою для подальшої обробки.

в) екран відображає дані отримані з давачів які оброблені мікроконтролером, це дозволяє отримувати візуальну корисну інформацію.

г) приймач та передавач використовуються для безперервного обміну інформації між контролером та кінцевим пристроєм, що отримує дані, обмін виконується через антени.

Структурна схема системи керування FPV дроном зображена на рис. 2.2.

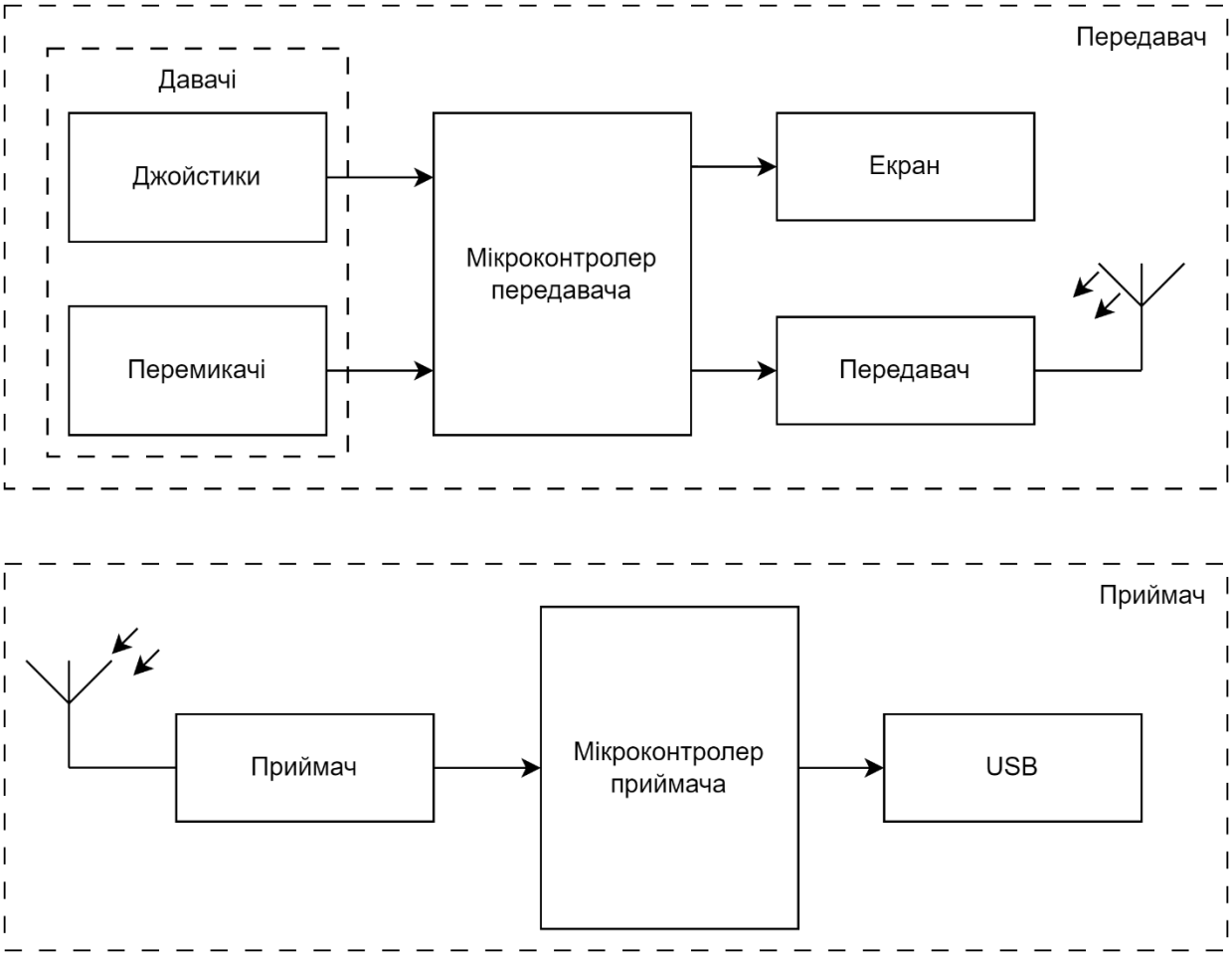

Рисунок 2.2 – Структурна схема комп’ютеризованої системи керування FPV дроном для симулятора польоту

2.2 Реалізація апаратної частини комп’ютеризованої системи керування FPV дроном для симулятора польоту

У цьому розділі розглянуто процес реалізації апаратної складової комп'ютеризованої системи керування FPV дроном для симулятора польоту.

2.2.1 Опис плати STM32F411CCU6

Плата STM32F411CCU6 – являє собою високопродуктивний мікроконтролер на основі ядра ARM Cortex-M4 від компанії STMicroelectronics (рис 2.3) [8].

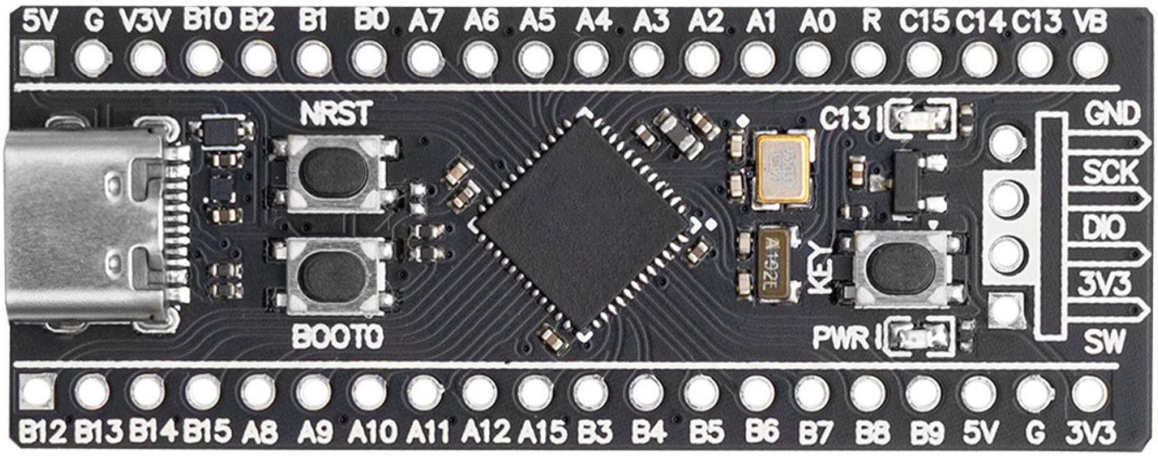

Рисунок 2.3 – Плата STM32F411CCU6

Цей мікроконтролер ідеально підходить для задач, що вимагають високої обчислювальної потужності та енергоефективності. Основою цієї плати є мікроконтролер STM32F411 серії STM32F4, який має 32-бітний ядро ARM Cortex-M4, що працює на частоті до 100 МГц. Завдяки своїй високій продуктивності і низькому енергоспоживанню, цей мікроконтролер забезпечує швидку і ефективну обробку даних, необхідну для складних обчислень.

В табл. 2.1 приведені технічні характеристики плати STM32F411CCU6.

Таблиця 2.1 – Основні характеристики STM32F411CCU6

Характеристика	Значення
Мікроконтролер	STM32F411CCU6
Тип ядра	Cortex-M4
Тактова частота	100 МГц
Пам'ять	256 КБ Flash, 128 КБ SRAM
Кількість DMA-каналів	16
Обчислення з плаваючою точкою	FPU
Інтерфейси	UART, SPI, I2C, USB OTG Full Speed, SDIO
АЦП	12-бітний, 16 каналів
ЦАП	2 x 12-бітних
ШІМ	16 каналів
Таймери	7 таймерів
Інші периферійні пристрої	RNG, WWDT, RTC, DMA, ADC, DAC, I2S, SPI
Напруга живлення	1.65 - 3.6 В
Діапазон робочих температур	-65°C - +150°C
Корпус	LQFP64

Плата STM32F411CCU6 забезпечує високу швидкість обробки даних і має достатню кількість периферійних пристроїв для реалізації повного комплексу задач, пов'язаних з управлінням дроном. Вона підтримує роботу з різними датчиками, модулями зв'язку та іншими компонентами системи, що робить її оптимальним вибором для розробки систем керування.

Плата підтримує широкий спектр інструментів і бібліотек для розробки програмного забезпечення, включаючи STM32CubeMX і STM32CubeIDE. Це

дозволяє легко розробляти, налагоджувати і оптимізувати програмне забезпечення для системи управління дроном.

Платформа STM32F411CCU6 має 44 піна, кожен з яких має різне функціональне призначення. Розташування пінів платформи STM32F411CCU6 зображено на рис. 2.4.

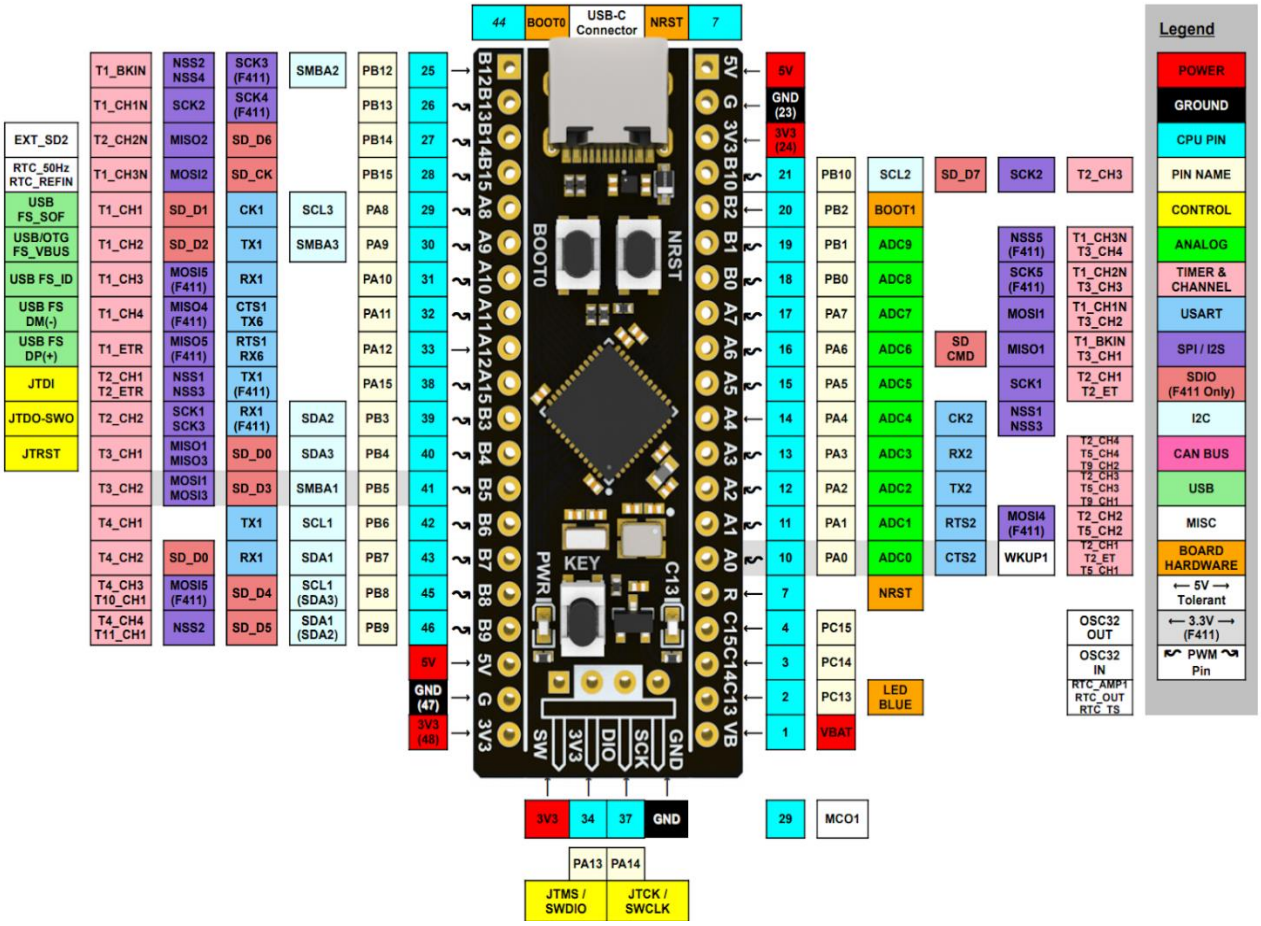

Рисунок 2.4 – Функціональне призначення пінів платформи STM32F411CCU6

STM32F411CCU6 має широкі можливості розширення, включаючи 37 GPIO пінів, які можна запрограмувати для різних потреб, USART x 3, I2C x 3, SPI x 5, USBFS x 1, та різноманітні інтерфейси зовнішніх пристроїв. Ці можливості роблять STM32F411CCU6 ідеальним вибором для розробки системи керування FPV дроном.

2.2.1 Опис IPS-дисплея ST7789

IPS-дисплей на драйвері ST7789 з інтерфейсом підключення SPI, розміром 240x240 пікселів знайшов широке застосування в системах керування та відображення інформації для різноманітних електронних пристроїв (рис. 2.5) [9].

Рисунок 2.5 – IPS-дисплей

Завдяки високоякісному кольоровому екрану своїм технічним характеристикам та перевагам, цей дисплей є ідеальним вибором для інтеграції у систему керування FPV дроном.

Роздільна здатність 240x240 пікселів, забезпечує достатньо високу деталізацію зображення для відображення як графічної, так і текстової інформації, що критично для ефективного контролю дрону в реальному часі. Це дозволяє користувачу швидко й чітко оцінювати показники системи.

Технологія IPS забезпечує широкі кути огляду до 178 градусів по горизонталі та вертикалі, що дозволяє зберігати якість зображення незалежно від положення користувача відносно екрану. Це важливо для оператора дрону, який може змінювати положення чи кут огляду під час керування, і при цьому йому не

потрібно турбуватися про спотворення кольорів або контрасту.

Висока контрастність і рівень яскравості сприяють чіткому та насиченому відображенню навіть за умов яскравого освітлення. Це дозволяє оператору легко розпізнавати інформацію на дисплеї під час роботи в різних умовах освітлення, що особливо важливо для зовнішнього використання.

IPS-дисплей ST7789 підтримує SPI інтерфейс для зв’язку, що дозволяє легко інтегрувати його з мікроконтролерами та іншими електронними компонентами.

Екран розроблений для енергоефективної роботи, що є важливим фактором для використання в автономних системах, таких як дрони.

В табл. 2.2 приведені технічні характеристики IPS дисплея ST7789.

Таблиця 2.2 – Основні характеристики IPS-дисплея

Характеристика	Значення
Розмір дисплея	1.3 дюйма
Роздільна здатність	240x240
Тип матриці	IPS
Кут огляду	>160°
Кількість кольорів	65000
Яскравість	250 кд/м²
Контрастність	1200:1
Час відгуку	10 мс
Інтерфейс	SPI
Напруга живлення	3.3 В
Споживання	0.06 Вт
Драйвер	ST7789
Розмір дисплея (мм)	26 x 26

Використання IPS дисплея ST7789 у системі керування FPV дроном значно покращує взаємодію оператора з системою, забезпечуючи високоякісне і надійне відображення інформації. Це сприяє підвищенню ефективності управління та безпеки польотів як у симуляторі, так і в реальних умовах.

2.2.3 Опис радіомодуля NRF24L01

Радіомодуль NRF24L01 є високочастотним бездротовим передавачем, що працює в діапазоні 2,4 ГГц (рис 2.6) [10].

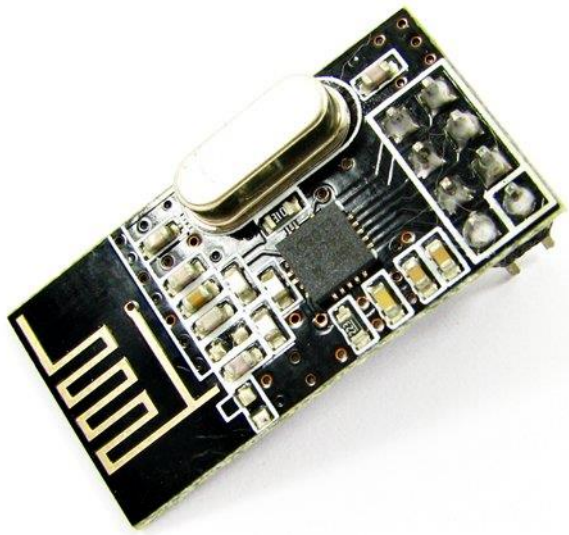

Рисунок 2.6 – Радіомодуль NRF24L01

Він призначений для забезпечення надійного зв'язку між передавачем (контролером) та приймачем, підключеним до персонального комп'ютера через USB-інтерфейс. Даний модуль відіграє ключову роль в реалізації системи керування FPV дроном, оскільки забезпечує передачу команд керування в реальному часі від користувача до дрону.

Частота роботи 2,4 ГГц, що забезпечує високу стійкість до перешкод та велику пропускну здатність.

Швидкість передачі даних 2 Мбіт/с, що дозволяє мінімізувати затримки в передачі команд керування.

Дальність зв'язку до 100 метрів на відкритій місцевості, що є достатнім для використання в симуляторі польоту.

Тип модуляції GFSK (Gaussian Frequency Shift Keying) забезпечує високий рівень захисту від перешкод.

Низьке споживання енергії, що забезпечує тривалу роботу на одному заряді акумулятора.

В табл. 2.3 приведені технічні характеристики радіомодуля NRF24L01.

Таблиця 2.3 – Основні характеристики радіомодуля NRF24L01

Характеристика	Значення
Частота	2,4 ГГц
Напруга живлення	1,9 В - 3,6 В
Інтерфейс	SPI
Максимальна швидкість передачі даних	2 Мбіт/с
Чутливість приймача	-90 дБм
Вихідна потужність	+20 дБм
Дальність зв'язку	До 100 м (на відкритому просторі), до 30 м (в приміщенні)
Кількість каналів	125
Розмір	25 мм x 15 мм
Робоча температура	-20°C - +85°C
Додаткові можливості	Автоматичне визначення швидкості передачі даних, шифрування даних, повторна передача пакетів

Розташування пінів платформи радіомодуля NRF24L01 зображено на рис. 2.7.

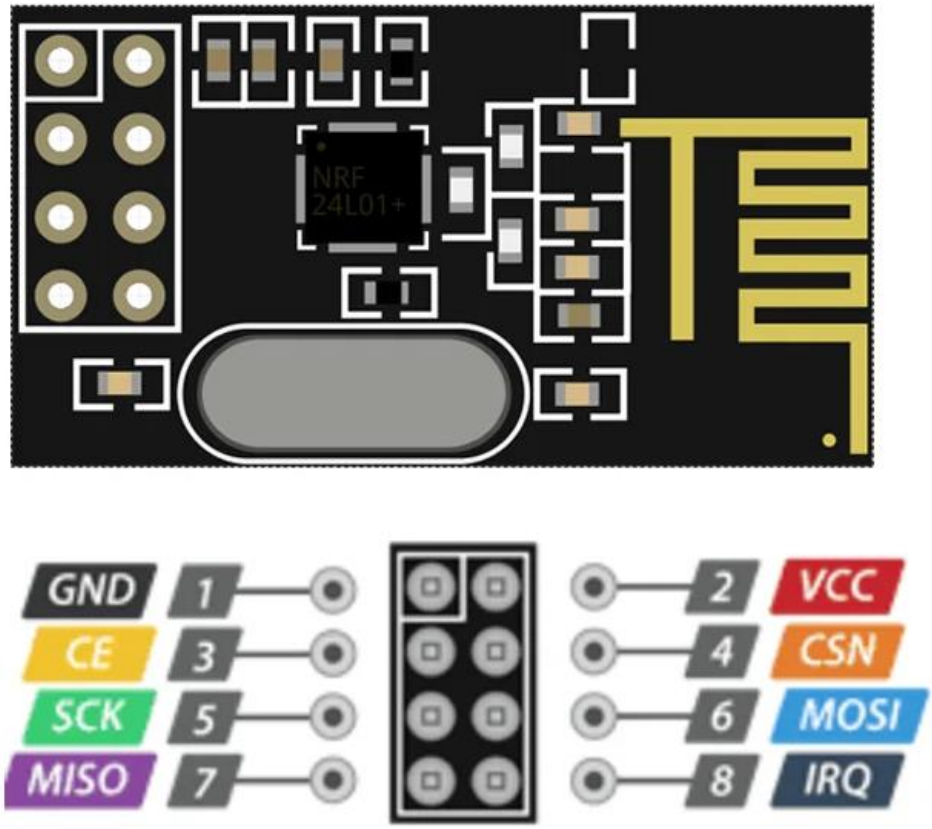

Рисунок 2.7 – Функціональне призначення пінів радіомодуля NRF24L01

Радіомодуль NRF24L01 використовується для створення надійного бездротового каналу зв'язку між передавачем, яким є ручний контролер, і приймачем, підключеним до ПК через USB-інтерфейс. Це дозволяє реалізувати реалістичний симулятор польоту, де команди від передавача швидко і надійно передаються до дрону, забезпечуючи плавне і точне керування.

Приймач, інтегрований з радіомодулем NRF24L01, приймає дані від контролера та передає їх на комп'ютер для обробки в реальному часі. Це дозволяє користувачу здійснювати керування дроном з мінімальною затримкою, що є критично важливим для забезпечення реалістичності симулятора польоту.

Висока швидкість передачі даних і надійність зв'язку, що забезпечуються NRF24L01, роблять цей модуль ідеальним вибором для використання в FPV дронах, які потребують точного і оперативного керування.

Таким чином, радіомодуль NRF24L01 є важливим компонентом в розробці системи керування FPV дроном, що забезпечує надійну і ефективну передачу даних між контролером і симулятором польоту, значно підвищуючи точність та ефективність керування дроном.

2.2.3 Опис адаптера для NRF24L01

Адаптер для радіомодулів NRF24L01 представляє собою спеціальну плату-перехідник, яка дозволяє зручно підключати ці модулі до мікроконтролерних плат (рис 2.8) [11]. Основна функція адаптера полягає в забезпеченні стабільного живлення і правильного з'єднання контактів радіомодуля з мікроконтролером.

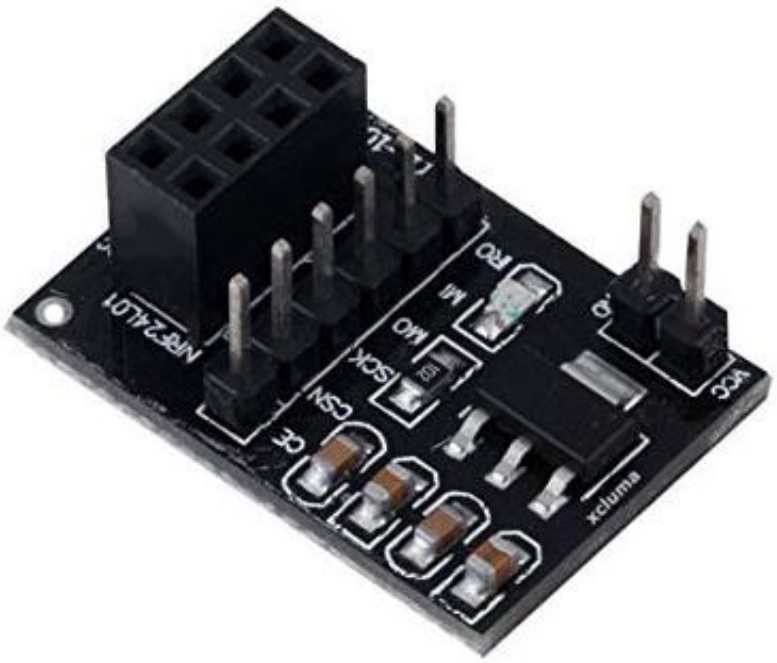

Рисунок 2.8 – Адаптер для NRF24L01

Вбудований стабілізатор на 3.3В забезпечує необхідну напругу для живлення радіомодулів NRF24L01, що дозволяє використовувати адаптер з

платами, які не мають власного стабілізатора або мають слабкий стабілізатор Адаптер підтримує живлення в діапазоні від 5 до 9 В.

Максимальний вихідний струм адаптера становить 600 мА, що забезпечує достатню потужність для стабільної роботи радіомодулів навіть при високих навантаженнях.

В табл. 2.4 приведені технічні характеристики адаптера для NRF24L01.

Таблиця 2.4 – Основні характеристики адаптера для NRF24L01

Характеристика	Значення
Напруга живлення	5…9В
Вихідна напруга	3.3В
Максимальний вихідний ток	600мА
Робоча температура	-40-125°С
Розміри	26х18х12мм

Електрична схема адаптера для NRF24L01 зображена на рис. 2.9.

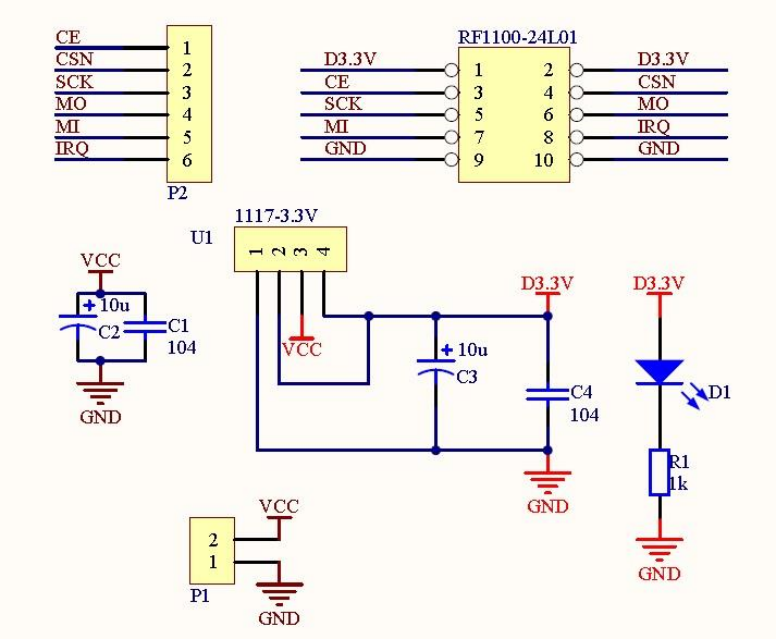

Рисунок 2.9 – Електрична схема адаптера для NRF24L01

Цей адаптер ідеально підходить для використання в проекті керування FPV дроном для симулятора польоту, забезпечуючи надійний зв'язок між мікроконтролером і радіомодулем NRF24L01, що є ключовим для ефективного керування дроном у режимі реального часу.

2.2.4 Опис модуля джойстика

Модуль джойстика, складається з двох аналогових датчиків з підключеними проводами для передачі сигналів. Кожен датчик розміщений в окремому корпусі, обладнаному монтажними отворами для фіксації на платі або іншій поверхні (рис 2.10).

Рисунок 2.10 – Модуль джойстика

Основною функцією модуля є вимірювання положення джойстика по двох осях – горизонтальній та вертикальній, що дозволяє точно визначати нахил

джойстика і передавати відповідні дані до комп'ютера або іншої системи управління.

Кожен модуль підключений до системи за допомогою проводів різного кольору, які передають сигнали позиції та живлення.

Цей модуль джойстика є одним з ключових компонентів для керування FPV дроном в симуляторі польоту, забезпечуючи точне і надійне передавання команд користувача на систему управління дроном. Він дозволяє користувачеві керувати дроном у реальному часі, відтворюючи реалістичні умови польоту та забезпечуючи інтерактивний досвід під час симуляції.

2.2.5 Опис перемикача

На рис. 2.11 зображено тумблерний перемикач, який є однією з ключових компонентів в системах управління, зокрема системах керування FPV дроном. Такий перемикач використовується для включення і виключення певних функцій або режимів роботи системи.

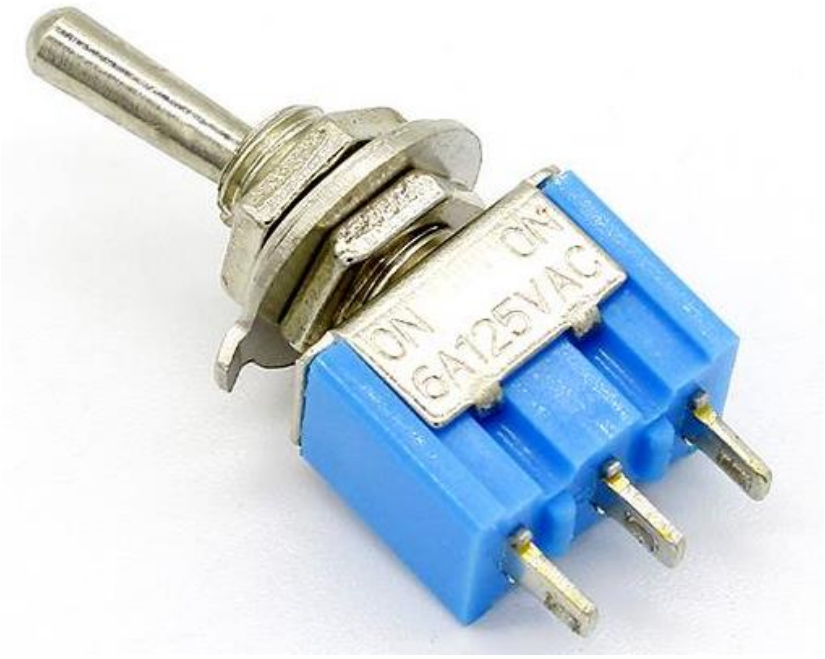

Рисунок 2.11 – Перемикач

Тумблерний перемикач є невід'ємною частиною системи керування FPV дроном для симулятора польоту. Його використання забезпечує зручність та надійність в експлуатації, дозволяючи швидко змінювати режими та функції дрона під час польоту.

2.3 Опис схеми електричної принципової комп’ютеризованої системи керування FPV дроном для симулятора польоту

На рис 2.12 представлена схема електрична принципова передавача комп’ютеризованої системи керування FPV дроном для симулятора польоту.

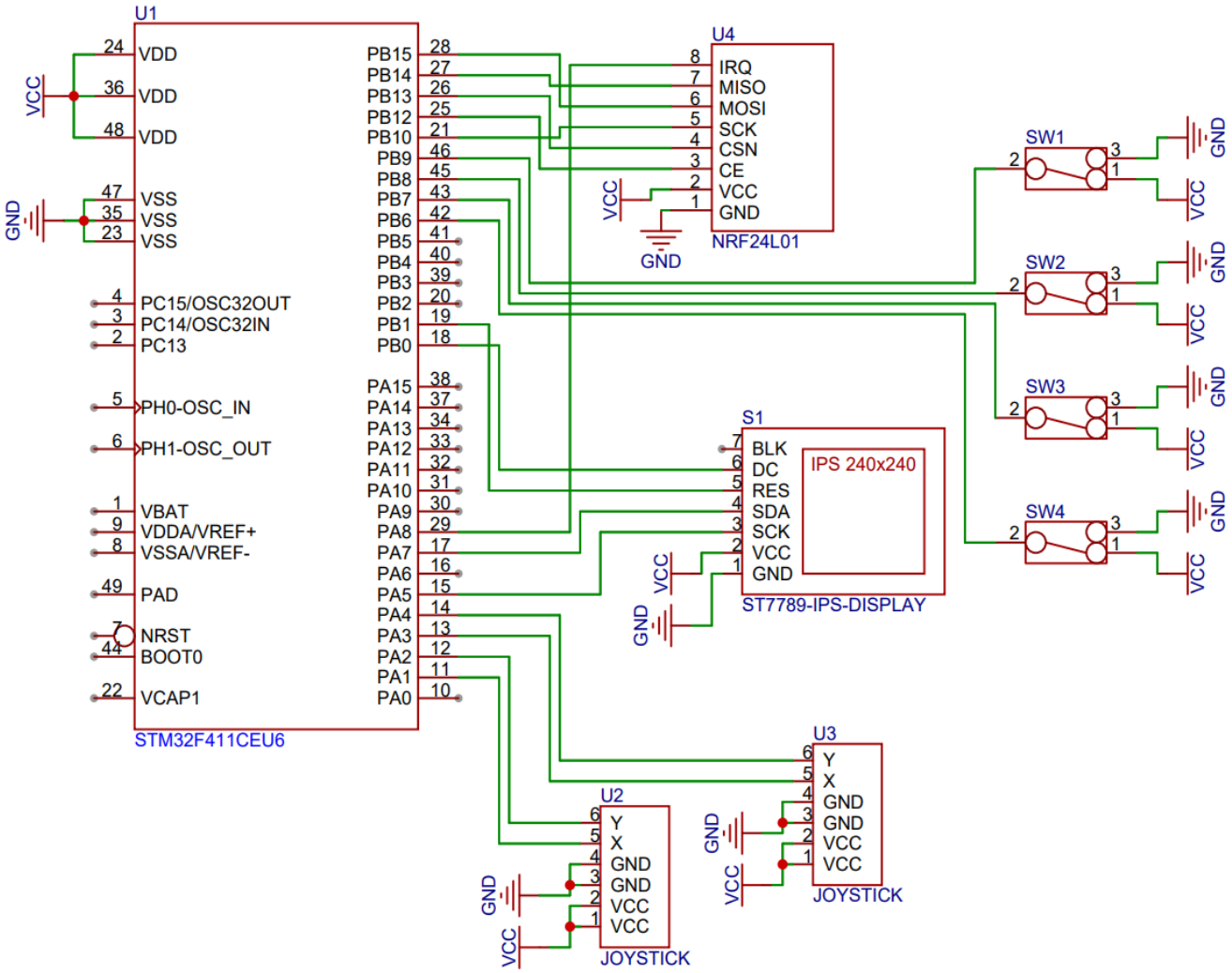

Рисунок 2.12 – Схема електрична принципова передавача

На схемі мікроконтролер STM32F411CCU6 U1 – центральний елемент керування, який забезпечує обробку сигналів від джойстиків і перемикачів, а також передає команди на передатчик. Він з'єднаний з іншими компонентами через інтерфейси SPI і GPIO.

Модулі джойстика U2 та U3 відповідають за зчитування положення стіків, які потім передається на мікроконтролер. Джойстики використовують аналогові сигнали, які надходять на аналогово-цифровий перетворювач мікроконтролера для подальшої цифрової обробки.

Бездротовий модуль NRF24L01 U4 для передачі даних, що працює на частоті 2.4 ГГц. Він забезпечує передачу команд керування від мікроконтролера до приймача.

Перемикачі Q1, Q2, Q3 та Q4, які використовуються для вибору режимів роботи або налаштування параметрів керування. Вони підключені до цифрових входів мікроконтролера і використовуються для зміни станів керування.

IPS дисплей ST7789 S1, призначений для виведення інформації про поточне положення стіків і позиції перемикачів. Він забезпечує високу роздільну здатність і яскравість зображення, що дозволяє оператору легко відстежувати поточні параметри керування.

Мікроконтролер отримує аналогові сигнали від модулів джойстиків U2 і U3, оцифровує їх і обробляє для визначення точного положення стіків.

Перемикачі SW1-SW4 підключені до цифрових входів мікроконтролера, дозволяють змінювати режими роботи системи і керувати функціями дрону.

Після обробки сигналів, мікроконтролер формує команди керування і передає їх на передатчик NRF24L01 (U4) через SPI інтерфейс для подальшої передачі.

Дисплей ST7789 (S1) підключений до мікроконтролера через інтерфейс SPI відображає поточні значення положення стіків і стан перемикачів, що забезпечує візуальний зворотній зв'язок для користувача.

На рис. 2.13 представлена схема електрична принципова приймача комп’ютеризованої системи керування FPV дроном для симулятора польоту.

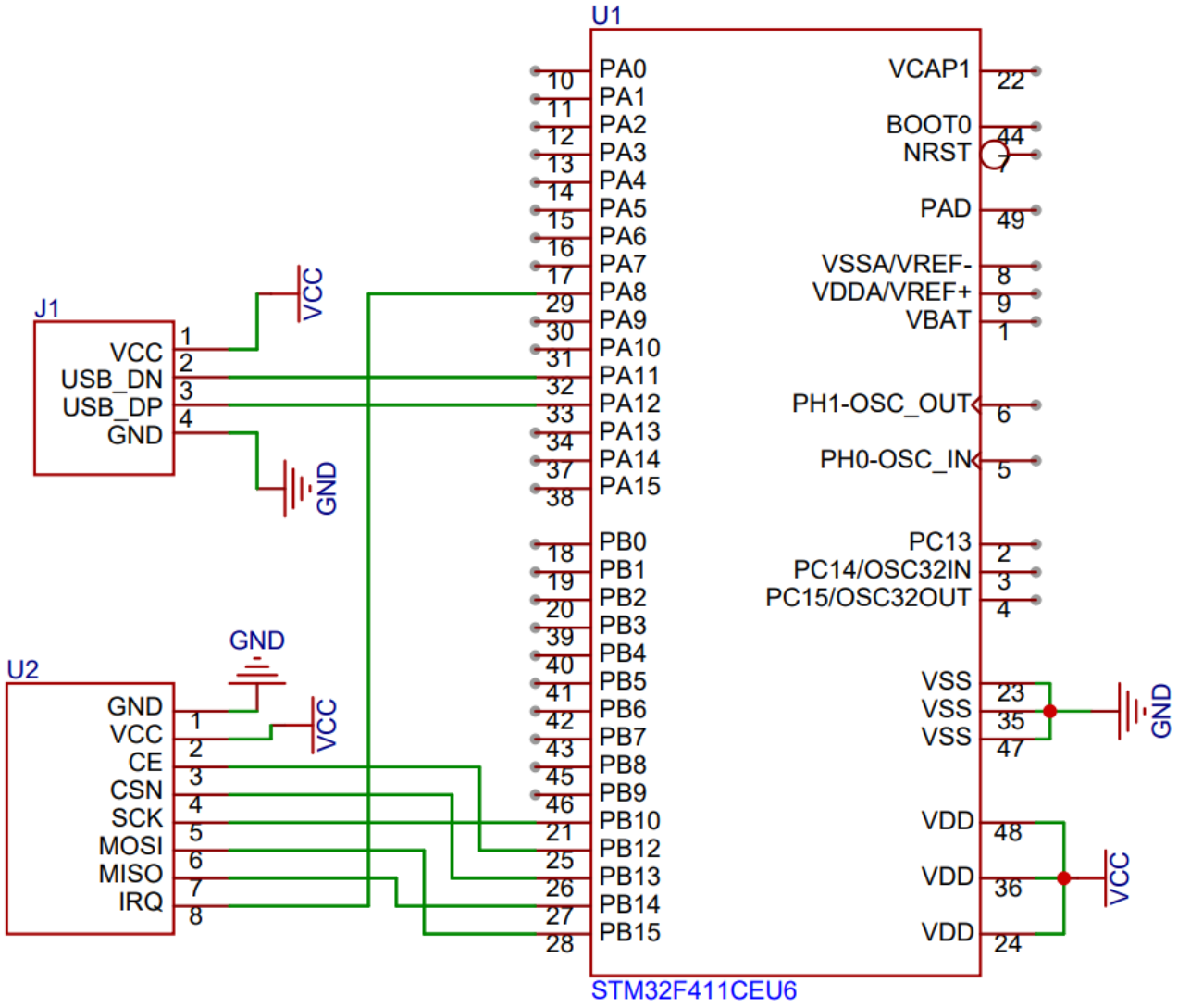

Рисунок 2.13 – Схема електрична принципова приймача

На схемі мікроконтролер STM32F411CCU6 U1 відповідає за обробку сигналів, що надходять з передатчика, і передачу даних на ПК через USB. До мікроконтролера підключені інтерфейси SPI для взаємодії з приймачем і USB для підключення до ПК.

До USB порту мікроконтролера підключено живлення і передача даних. Через цей інтерфейс дані передаються на ПК для подальшої обробки симулятором польоту. Мікроконтролер запрограмований для роботи в режимі HID, що дозволяє

ПК розпізнавати його як стандартний пристрій введення, такий як джойстик або геймпад.

Бездротовий модуль прийому даних NRF24L01 U2, який працює на частоті 2.4 ГГц і приймає команди з передатчика, що знаходиться на пульті керування. Він підтримує зв'язок і забезпечує стабільну передачу даних на великі відстані з низьким енергоспоживанням. Він підключений до мікроконтролера через інтерфейс SPI для швидкої передачі прийнятих сигналів.

Ця принципова електрична схема демонструє взаємозв'язок основних компонентів пристрою для керування FPV дроном у симуляторі польоту. Вона відображає шляхи передачі даних та команд між різними елементами системи.

2.4 Обґрунтування вибору середовища розробки ПЗ для реалізації системи керування FPV дроном для симулятора польоту

Для розробки системи керування FPV дроном було обрано середовище розробки STM32CubeIDE [12]. Це інтегроване середовище розробки, спеціально розроблене для створення проектів на основі мікроконтролерів платформи STM32. Воно забезпечує зручний та простий спосіб програмування мікроконтролерів за допомогою мови програмування C/C++.

Основні причини вибору STM32CubeIDE включають:

- STM32CubeIDE створене спеціально для роботи з мікроконтролерами STM32, що дозволяє максимально ефективно використовувати можливості апаратного забезпечення;
- інтеграція з STM32CubeMX дозволяє легко налаштовувати конфігурацію мікроконтролера, обирати потрібні периферійні пристрої і генерувати початковий код;
- STM32CubeIDE дозволяє інтегруватися з іншими інструментами та середовищами розробки, що розширює можливості розробників і дозволяє

використовувати додаткові інструменти для аналізу, тестування і оптимізації коду;

- STM32CubeIDE підтримує різноманітні інструменти для налагодження, включаючи JTAG і SWD, що дозволяє ефективно діагностувати та виправляти помилки в коді.

На рис. 2.14 представлено вікно програми STM32CubeIDE.

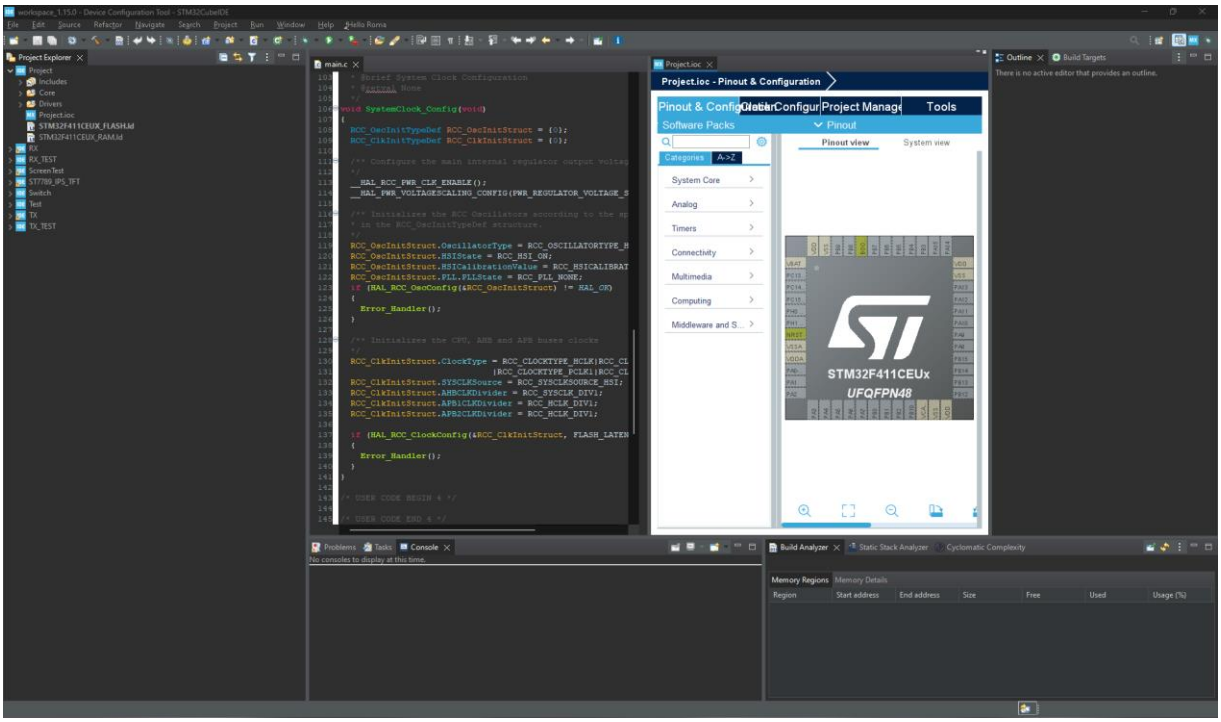

Рисунок 2.14 – Застосунок STM32CubeIDE

Таким чином, STM32CubeIDE забезпечує зручний та ефективний інструмент для розробки програмного забезпечення для керування FPV дроном, що дозволяє швидко реалізувати функціонал і забезпечити високу надійність і продуктивність системи.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Алгоритм роботи системи керування FPV дроном для симулятора польоту

Алгоритм функціонування системи керування FPV дроном для симулятора польоту складається з кількох послідовних етапів, кожен з яких виконує певні завдання для забезпечення безперебійної і точної роботи системи. Основні етапи включають:

а) ініціалізація системи:

- відбувається запуск системи;
- проводиться початкове налаштування компонентів, таких як давачі, екран та модуль передачі даних;
- здійснюється ініціалізація давачів для забезпечення коректного зчитування даних у подальших етапах;

б) встановлення зв’язку:

- система ініціює процес встановлення зв’язку з приймачем;
- проводиться синхронізація з приймачем для встановлення стабільного з'єднання;
- забезпечується підтримка надійного каналу зв'язку для передачі даних у реальному часі;
- проводиться перевірка стабільності з'єднання, і у разі виявлення проблем зв'язок автоматично перезапускається для відновлення з’єднання;

в) зчитування даних з давачів:

- система зчитує дані з різних давачів, що включають положення джойстиків, активність перемикачів та інші необхідні параметри;

- отримані дані використовуються для подальшої обробки і аналізу;

г) перетворення даних для передачі:

- отримані з давачів дані перетворюються у відповідний формат для подальшої передачі на приймач;
- забезпечується збереження точності даних для коректного керування дроном;

ґ) відправлення даних на приймач:

- перетворені дані передаються на приймач через встановлений канал зв'язку;
- відправка даних виконується у реальному часі для мінімізації затримок;

е) вивід даних на екран:

- дані про положення джойстиків та стани перемикачів виводяться на екран;
- це забезпечує візуальну інформацію для користувача, що дозволяє оцінити поточний стан системи;

є) отримання даних на приймачі:

- приймач отримує дані, що передані з контролера;
- здійснюється перевірка отриманих даних на цілісність і точність;

ж) керування дроном:

- отримані дані направляються у симулятор, що дозволяє керувати дроном у реальному часі;
- забезпечується відповідна реакція дрона на команди користувача, що моделює реальну поведінку дрона.

Цей алгоритм забезпечує узгоджену і ефективну роботу системи, дозволяючи точно керувати FPV дроном у середовищі симулятора польоту. Кожен етап є критично важливим для досягнення високої точності і надійності керування дроном, забезпечуючи реалістичність і плавність симуляції польоту.

Блок-схема алгоритму роботи пристрою для керування FPV дроном у симулятора польоту зображено на рис. 3.1.

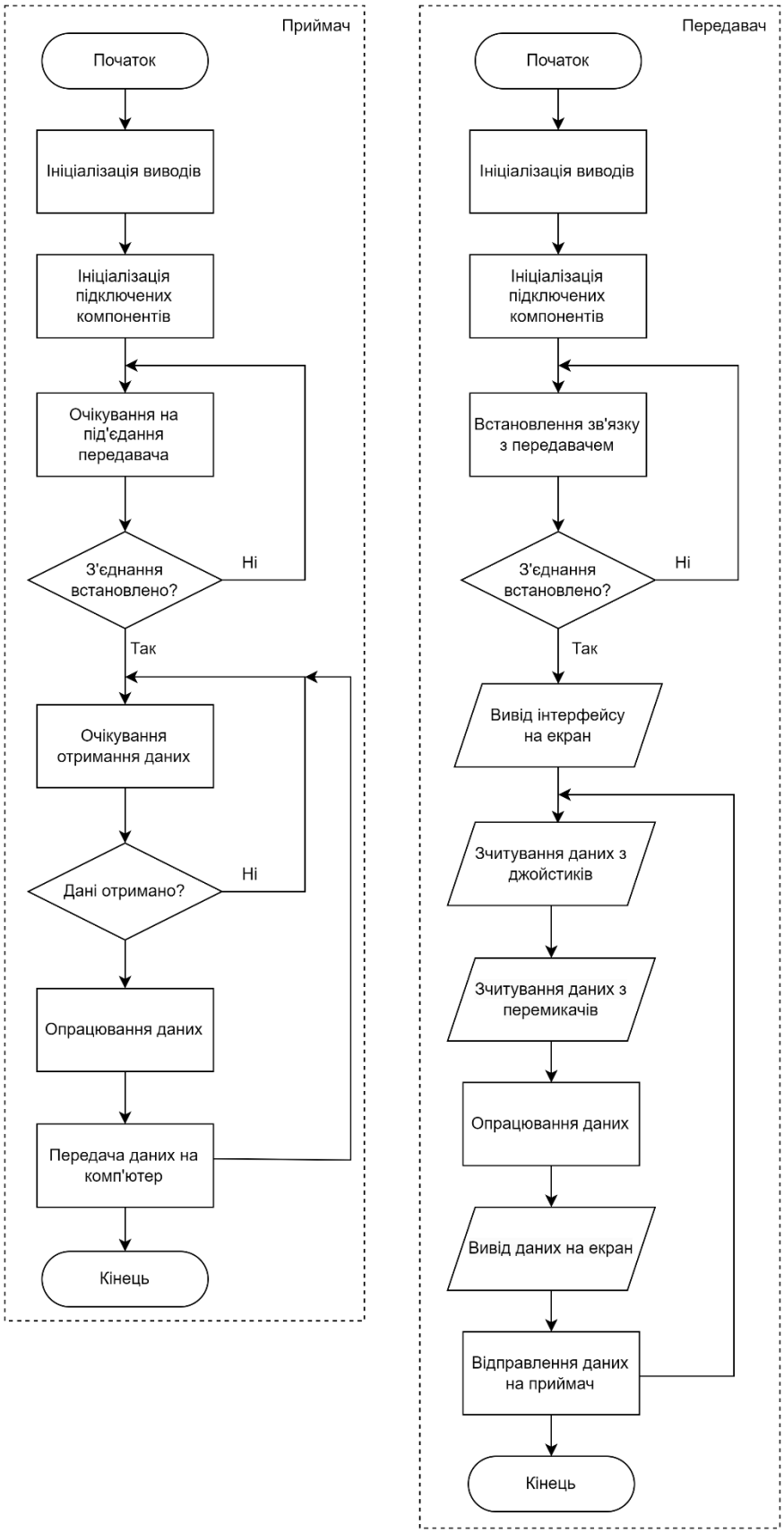

Рисунок 3.1 – Блок-схема алгоритму роботи пристрою для керування FPV дроном у симулятора польоту

3.2 Створення проєкту в STM32CubeIDE

Для початок роботи з середовищем розробки STM32CubeIDE необхідно створити новий проєкт, який використовується для написання та відлагодження коду для мікроконтролера.

Відкриваємо програму STM32CubeIDE, яка є інструментом для розробки програмного забезпечення для мікроконтролерів STM32.

У верхньому меню відкриваємо вкладку «File», далі «New», а потім «STM32 Project». Це відкриє вікно для створення нового проєкту.

У діалоговому вікні для вибору мікроконтролера знаходимо модель STM32F411CEU6 і обираємо її (рис 3.2).

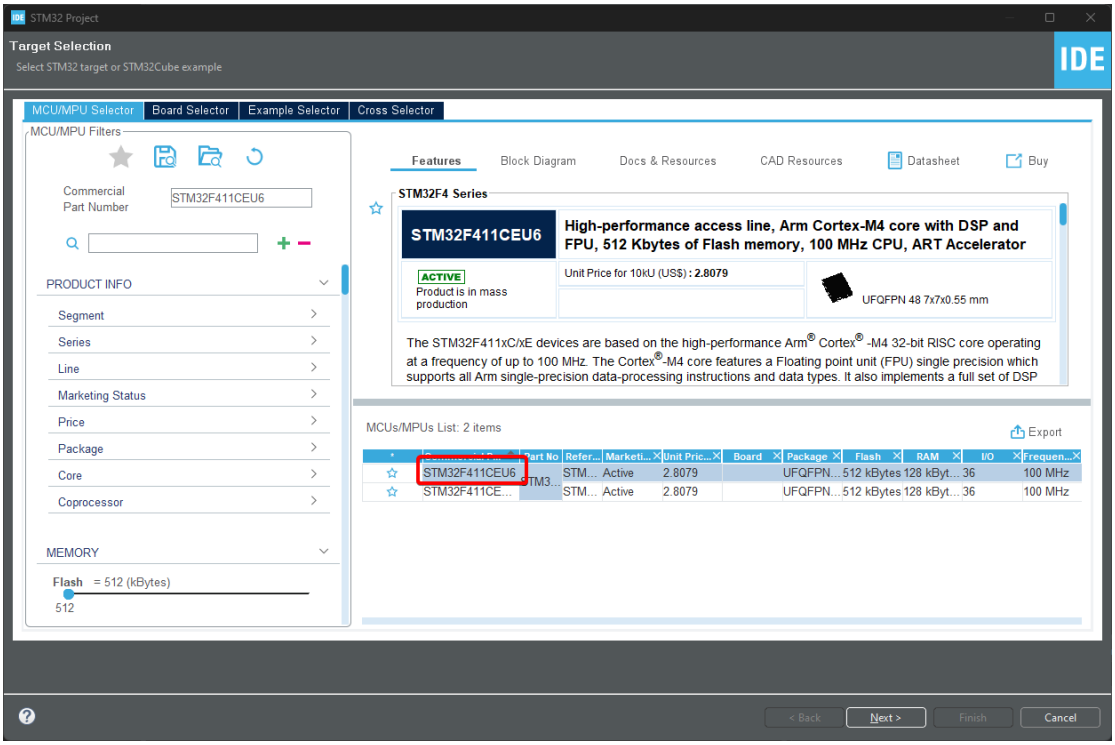

Рисунок 3.2 – Вибір мікроконтролера

Вводимо назву проєкту у відповідному полі. Під назвою проєкту знаходиться опція вибору мови програмування. Обираємо мову «C», яка є найбільш поширеною мовою для програмування мікроконтролерів (рис 3.3);

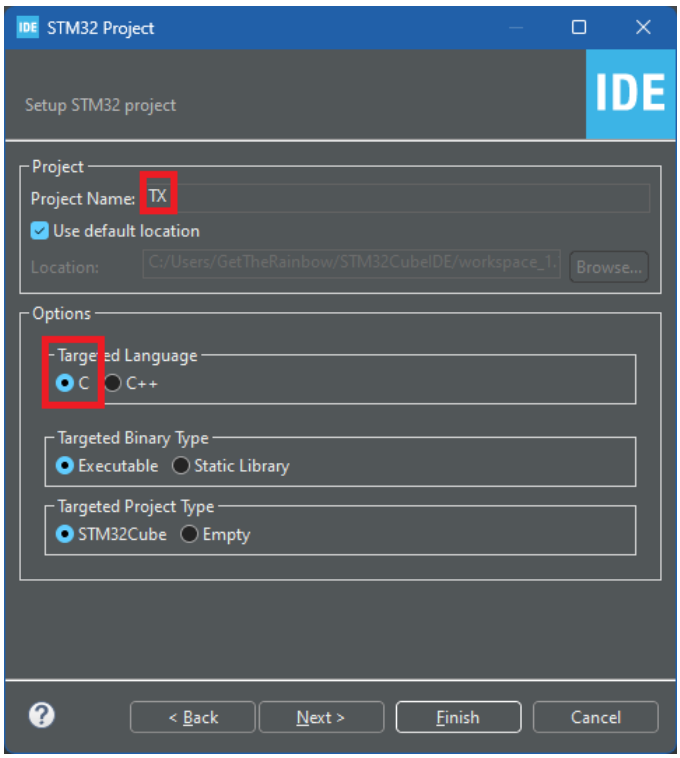

Рисунок 3.3 – Додаткові налаштування проєкта

Після введення всіх необхідних даних і вибору мікроконтролера натискаємо кнопку «Finish». Це завершить процес створення проєкту і відкриє нове робоче середовище для подальшого налаштування мікроконтролера.

3.3 Розробка ПЗ для приймача

Для налаштування плати мікроконтролера STM32F411 приймача, необхідно відкрити файл «RX.ioc», який використовується для конфігурації мікроконтролерів STM32. Спочатку потрібно налаштувати тактування мікроконтролера. Переходимо у вкладку «System Core», далі вибираємо опцію RCC. У випадаючому списку з назвою «High Speed Clock» обираємо «Crystal/Ceramic Resonator». Це дозволить використовувати зовнішній кварцовий резонатор для тактування мікроконтролера.

Після цього переходимо на вкладку «Clock Configuration». У цьому розділі встановлюємо джерело PLL у режим HSE, що означає використання зовнішнього

кварцового резонатора як джерела тактування. Далі необхідно задати значення HCLK на 84 MHz. Після введення цього значення натискаємо клавішу Enter, і чекаємо, доки середовище автоматично обрахує та налаштує значення тактування для всіх необхідних системних шин і переферійних пристроїв. Це забезпечить стабільну роботу мікроконтролера з необхідною частотою, що є критично важливим для виконання завдань комп'ютеризованої системи керування FPV дроном для симулятора польоту (рис. 3.4).

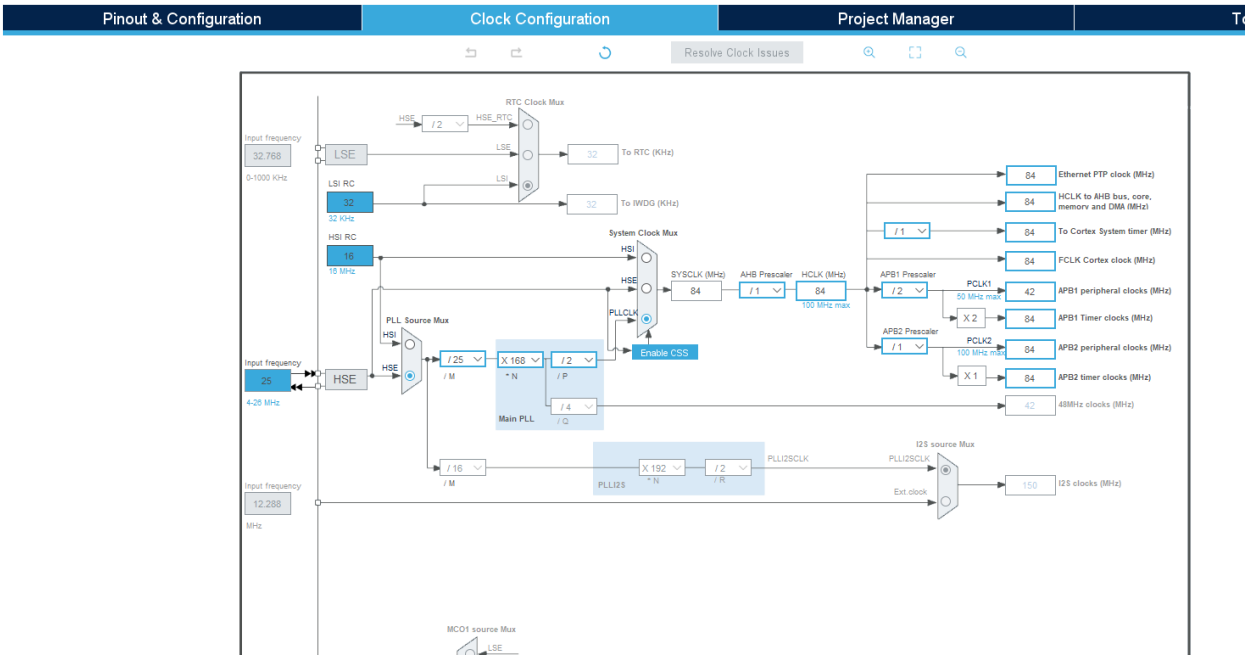

Рисунок 3.4 – Вкладка Clock Configuration

Для підключення NRF24L01 до плати мікроконтролера налаштуємо його піни. Для цього перейдемо у вкладку «Connectivity» далі SPI2. У випадаючому списку з назвою «Mode» обираємо «Full-Duplex Master». Додаємо додаткові піни для обміну даними з приймачем:

- PA8 -> GPIO_EXTI8;
- PB12 -> GPIO_Output;
- PB13 -> GPIO_Output.

Налаштовуємо піни як зображено на рис. 3.5.

Pin Name	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum output speed	User Label
PA8	n/a	External Interrupt Mode with Falling edge trigger detection	No pull-up and no pull-down	n/a	IRQ
PB12	Low	Output Push Pull	No pull-up and no pull-down	High	CE
PB13	High	Output Push Pull	No pull-up and no pull-down	High	SPI2_CSN

Рисунок 3.5 – Налаштування пінів

Для того, щоб приймач розпізнавався комп'ютером як USB HID пристрій переходимо на вкладку «Middleware and Software Packs». Шукаємо «USB_DEVICE», встановлюємо значення «Custom HID Class». На рис. 3.6 зображено повні налаштування USB.

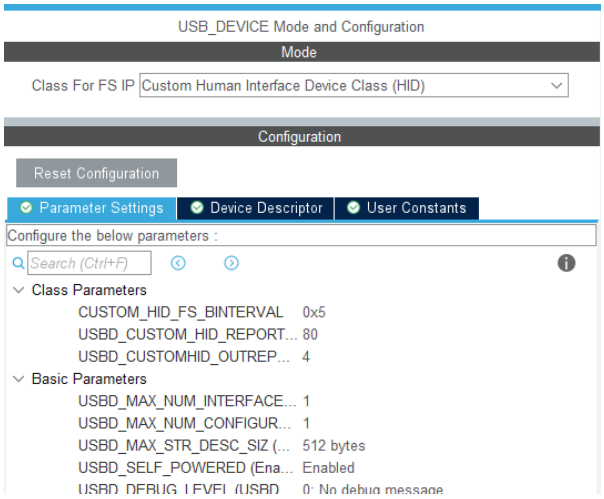

Рисунок 3.6 – Налаштування USB

Вигляд налаштованих пінів через CubeIDE зображено на рис. 3.7.

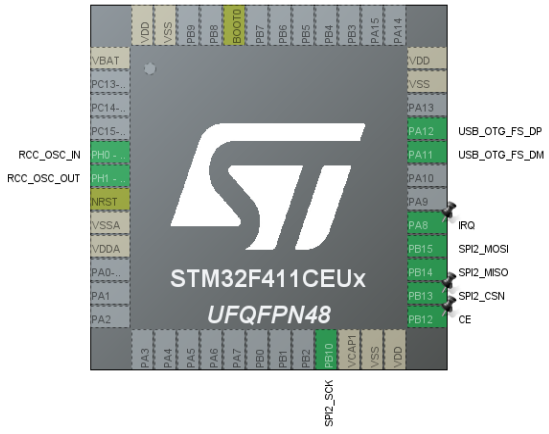

Рисунок 3.7 – Налаштовані піни

Після налаштування зберігаємо файл. Погоджуємося на генерацію коду від CubeIDE.

Для забезпечення взаємодії комп'ютеризованої системи керування FPV дроном із симулятором польоту був створений HID пристрій [14]. Цей пристрій дозволяє передавати дані з приймача на комп'ютер, забезпечуючи точний контроль над дроном у віртуальному середовищі.

Використовуючи інструмент HID Description Tool [15], було створено опис HID пристрою, який визначає, як дані з пристрою вводяться та обробляються. Опис містить визначення для осей руху (X, Y, Z) і повороту (Rz), а також для чотирьох кнопок (рис 3.8). Це дозволяє реалізувати підтримку основних функцій управління дроном, включаючи переміщення у просторі та виконання різних команд за допомогою перемикачів.

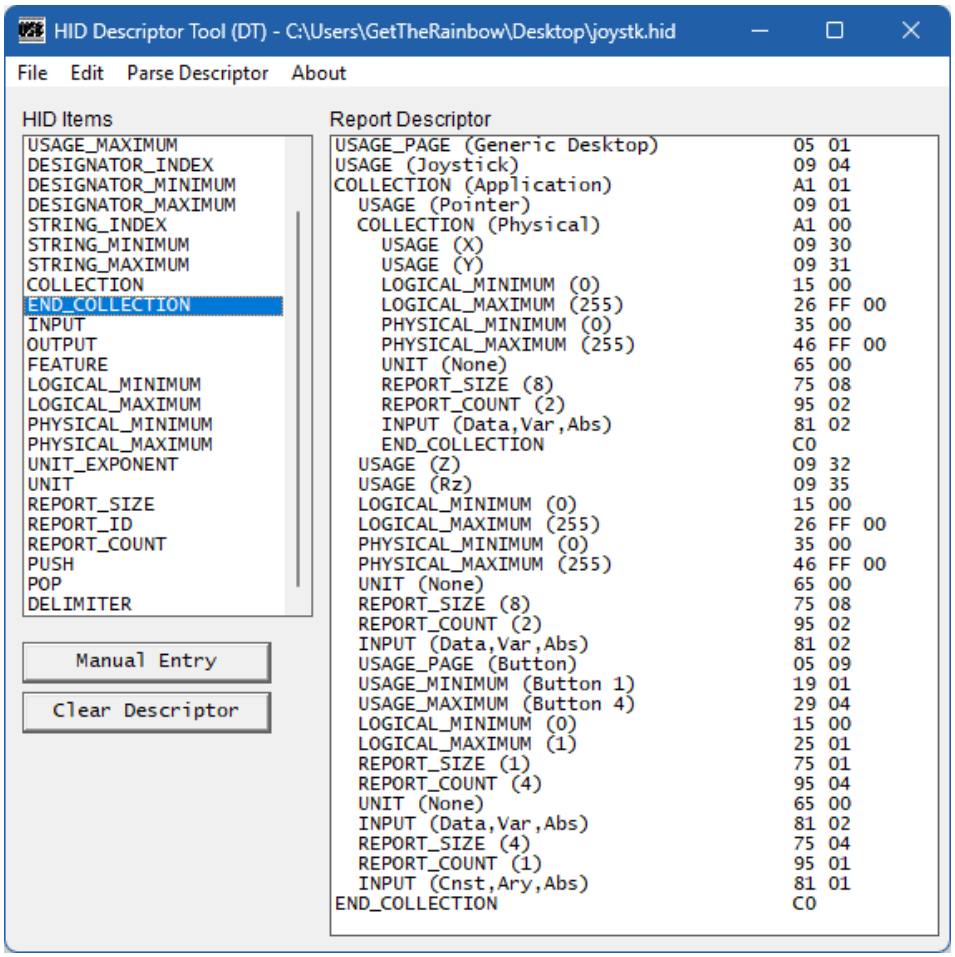

Рисунок 3.8 – HID опис

Зберігаємо отриманий опис, переносимо його у відповідний файл в IDE.

Щоб переконатися в правильності створеного опису HID пристрою, підключаємо плату мікроконтролера до комп'ютера. Прошиваємо його натисканням зеленої кнопки «Run Release». Відкриваємо програму для перевірки пристроїв, яка вбудована в операційну систему. Переконуємося, що контролер був успішно розпізнаний, і його налаштування відповідають очікуваним параметрам (рис. 3.9).

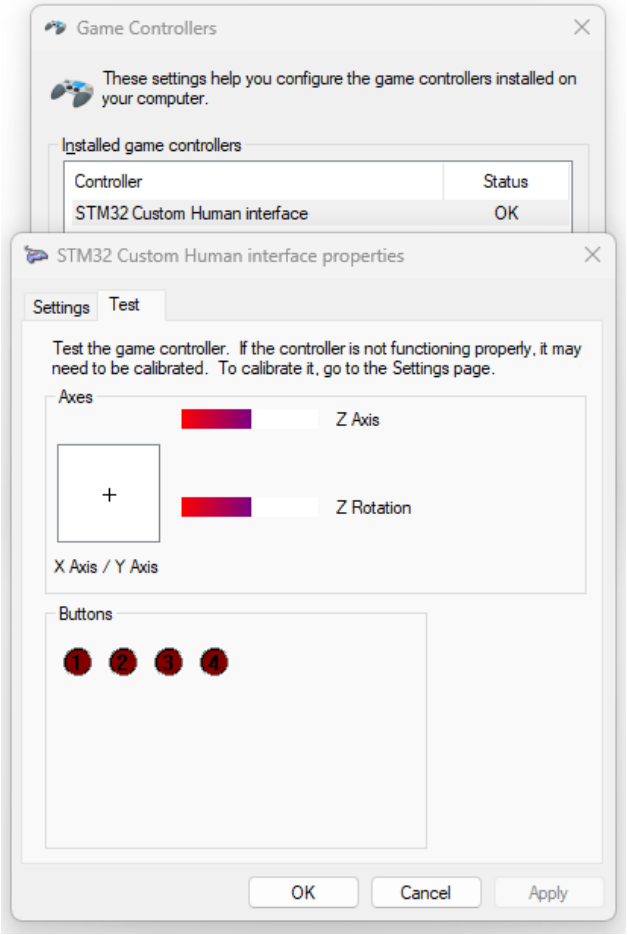

Рисунок 3.9 – Перевірка роботи приймача

Перейдемо до написання коду. Для цього відкриємо головний файл програми, який називається «main.c». У цьому файлі створюємо структуру для зручного збереження положення джойстиків та стани перемикачів (рис 3.10).

```
#pragma pack(push, 1)
struct myReportStruct {
    int8_t X;
    int8_t Y;
    int8_t Z;
    int8_t Rz;
    uint8_t Buttons;
};
#pragma pack(pop)

struct myReportStruct FPV_Controller;
```

Рисунок 3.10 – Лістинг структури для збереження даних

Ініціюємо модуль приймача, встановлюємо його частоту, швидкість та вихідну силу приймання (рис 3.11).

```
nrf24l01p_rx_init(2500, _1Mbps);
nrf24l01p_set_rf_tx_output_power(_18dBm);
```

Рисунок 3.11 – Лістинг ініціації модуля

Для отримання даних з NRF24L01 завантажимо відповідну бібліотеку [16], та перенесемо її у проєкт. Після чого підключимо головний (nrf24l01p.h) файл бібліотеки. Реєструємо переривання для зчитування даних які приходять на модуль. Отримані дані записуємо у створену структуру (рис. 3.12).

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == NRF24L01P_IRQ_PIN_NUMBER) {

        static uint8_t rx_data[NRF24L01P_PAYLOAD_LENGTH] = { 0, };

        nrf24l01p_rx_receive(rx_data);

        FPV_Controller.Buttons = rx_data[1];
        FPV_Controller.X = MIN(MAX(rx_data[2], 0), 255);
        FPV_Controller.Y = MIN(MAX(rx_data[3], 0), 255);
        FPV_Controller.Z = MIN(MAX(rx_data[4], 0), 255);
        FPV_Controller.Rz = MIN(MAX(rx_data[5], 0), 255);

    }
}
```

Рисунок 3.12 – Лістинг отримання та запису даних

Щоб передати отримані дані на комп’ютер через USB використаємо вбудовану в бібліотеку для роботи з USB функцію, передаємо їй посилання на структуру (рис 3.13).

```
while (1) {
  USBD_CUSTOM_HID_SendReport(&hUsbDeviceFS, (uint8_t *) &FPV_Controller, sizeof(FPV_Controller));
  HAL_Delay(10);
}
```

Рисунок 3.13 – Лістинг передачі даних на комп’ютер

Після успішного написання коду, наступним кроком є прошивання мікроконтролера.

3.4 Розробка ПЗ для передавача

Налаштування тактування та підключення модулю NRF24L01 аналогічне до налаштування приймача.

Для підключення IPS-дисплею переходимо на вкладку «Connectivity», далі «SPI1». У випадаючому списку з назвою «Mode» обираємо «Transmit Master Only». Налаштовуємо SPI1 як зображено на рис. 3.14.

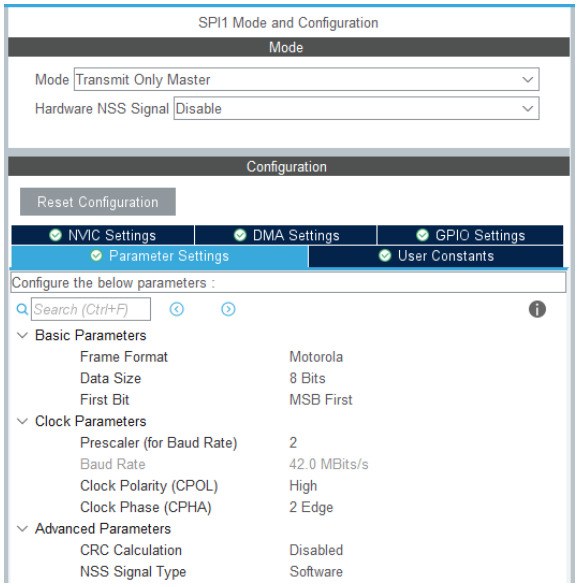

Рисунок 3.14 – Налаштування SPI1

Налаштуємо DMA для прямого доступу екрану до пам’яті мікроконтролера. Це необхідно для швидкого виводу даних на екран. Для цього переходимо на вкладку «DMA Settings». Натискаємо «Add», обираємо SPI1_TX. Залишаємо налаштування за замовчуванням (рис. 3.15).

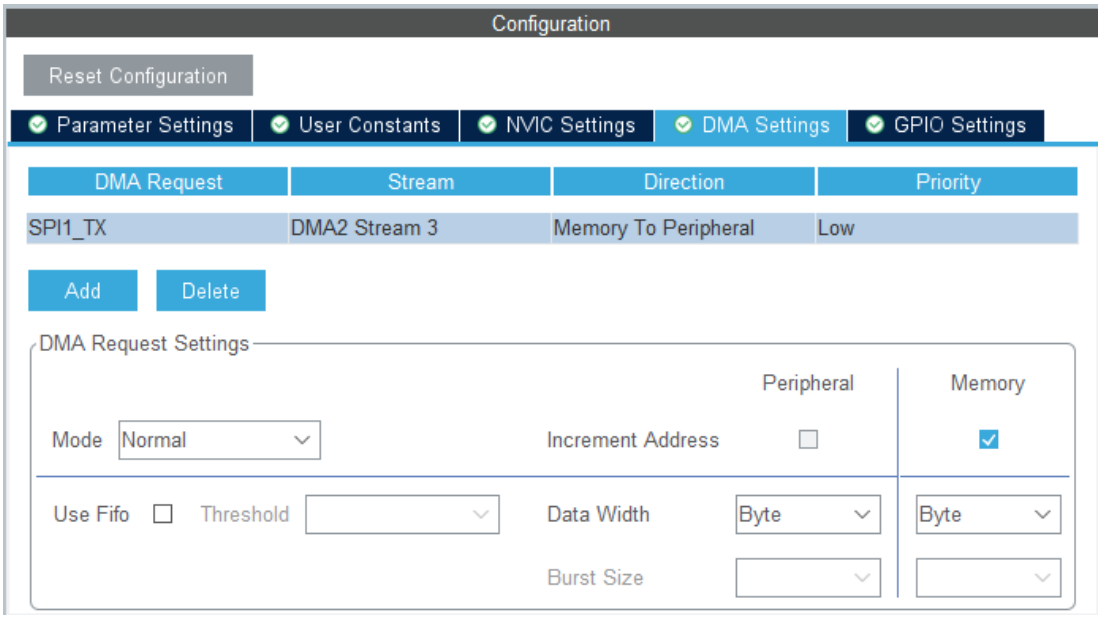

Рисунок 3.15 – Налаштування DMA

Налаштуємо піни для передачі даних:

- PA7 -> SPI1_MOSI;
- PB0 -> GPIO_Output;
- PB1 -> GPIO_Output;

Тепер налаштуємо піни для джойстиків, для цього переходимо на вкладку «Aanalog», далі «ADC1». Встановлюємо прапорці на проти «IN1», «IN2», «IIN3», «IN4». Переходимо на вкладку «DMA Settings» натискаємо Add, обираємо ADC1. Це необхідно для можливості зчитування більше 1 виходу аналогово-цифровим перетворювачем.

Повертаємося на вкладку «Parameter Settings». На рис. 3.16 зображено налаштування цієї вкладки.

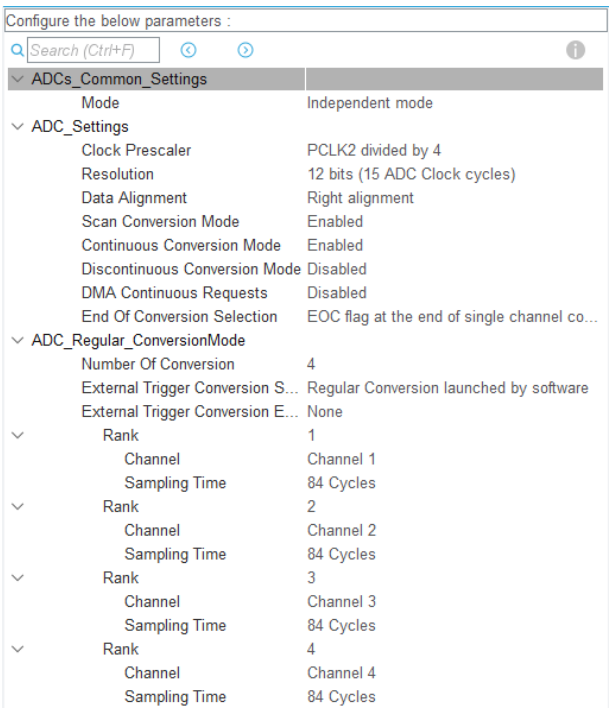

Рисунок 3.16 – Налаштування DMA

Налаштуємо піни для отримання активності перемикачів. Для цього переведемо піни PB6, BP7, PB8, PB9 у режим GPIO_Input.

На рис. 3.17 зображено налаштовані піни на мікроконтролері.

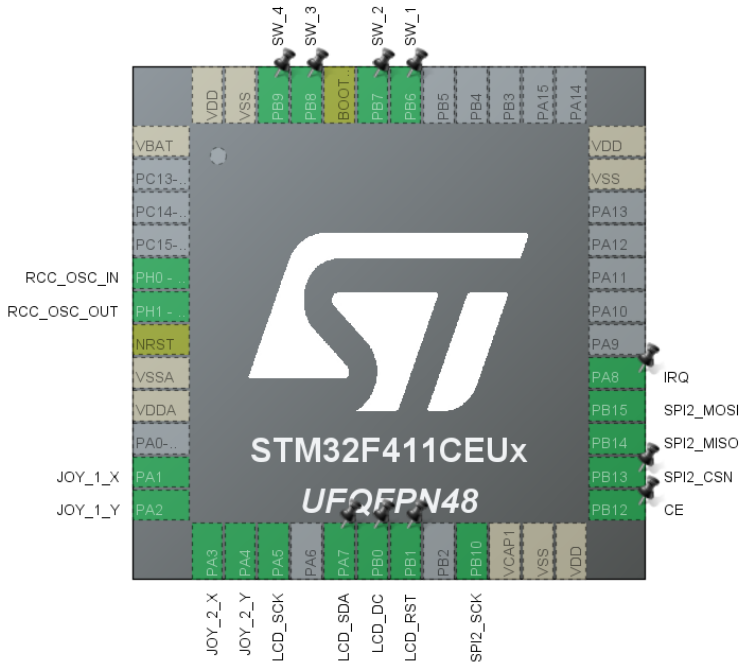

Рисунок 3.17 – Налаштовані піни

Увімкнемо таймер для періодичного надсилання отриманих даних з джойстиків та перемикачів. Це необхідно для того, щоб не забивати канал передачі і дані встигали оброблятися. Для цього перейдемо у вкладку «Timers», далі «TIM2». У випадаючому списку з назвою «Clock Source» обираємо «Internal Clock». Обраховуємо «Clock Period» за формулою (3.1), де t – необхідний час періодичності спрацювання таймеру в секундах, f – частота тактування таймеру в Герцах. Частота тактування таймера знаходиться на вкладці «Clock Configuration», далі «APB1 Timer clocks».

$$Clock\ period = (t * f) - 1 \quad (3.1)$$

Налаштування таймера зображені на рис. 3.18.

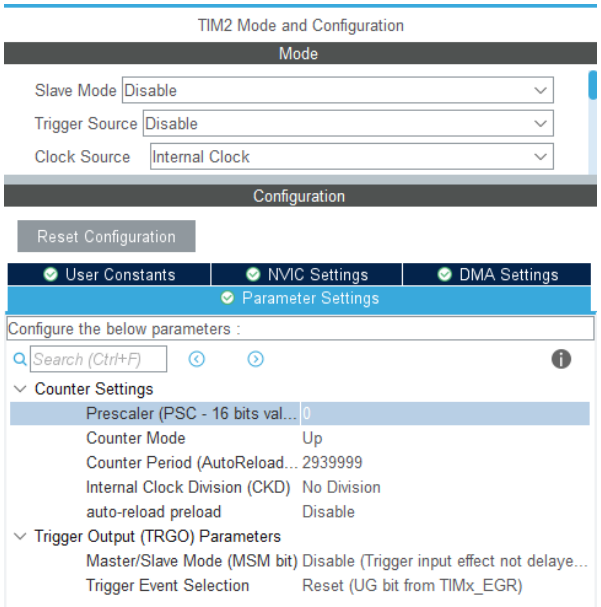

Рисунок 3.18 – Налаштування таймера

Активуємо глобальне переривання для таймера у розділі «NVIC Settings», щоб мати можливість викликати код, який буде передавати дані.

Зберігаємо файл, погоджуємося з генерацією коду в CubeIDE.

Переходимо до написання коду. Відкриваємо файл «main.c».

Підключаємо бібліотеки для роботи з передавачем (nrf24l01p.h) [16] та для роботи з IPS-екраном (st7789.h) [17].

Створюємо масив для збереження даних, які будемо відправляти на приймач (рис. 3.19).

```
uint8_t tx_data[NRF24L01P_PAYLOAD_LENGTH] = { 0, 0, 0, 0, 0, 0, 0, 0 };
```

Рисунок 3.19 – Лістинг створення масиву

Ініціюємо екран і передавач, також вмикаємо відлік таймера (рис. 3.20).

```
ST7789_Init();
nrf24l01p_tx_init(2500, _1Mbps);
nrf24l01p_set_rf_tx_output_power(_18dBm);
HAL_TIM_Base_Start_IT(&htim2);
```

Рисунок 3.20 – Лістинг для ініціація модулів і таймера

Виводимо на екран статичну інформацію, яка не потребує оновлення, щоб оптимізувати роботу мікроконтролера (рис 3.21).

```
ST7789_Fill_Color(BLACK);

ST7789_DrawFilledRectangle(0, 0, 240, 18, RED);
ST7789_WriteString(10, 0, "FPV Drone Controller", Font_11x18, WHITE, RED);

ST7789_WriteString(30, 37, "B1", Font_7x10, WHITE, BLACK);
ST7789_WriteString(85, 37, "B2", Font_7x10, WHITE, BLACK);
ST7789_WriteString(142, 37, "B3", Font_7x10, WHITE, BLACK);
ST7789_WriteString(197, 37, "B4", Font_7x10, WHITE, BLACK);

ST7789_WriteString(4, 165, "X", Font_7x10, WHITE, BLACK);
ST7789_WriteString(60, 107, "Y", Font_7x10, WHITE, BLACK);
ST7789_WriteString(57, 223, "J1", Font_7x10, WHITE, BLACK);

ST7789_WriteString(229, 165, "X", Font_7x10, WHITE, BLACK);
ST7789_WriteString(173, 107, "Y", Font_7x10, WHITE, BLACK);
ST7789_WriteString(169, 223, "J2", Font_7x10, WHITE, BLACK);

ST7789_DrawRectangleWithSize(14, 120, 100, 100, WHITE);
ST7789_DrawRectangleWithSize(126, 120, 100, 100, WHITE);
```

Рисунок 3.21 – Лістинг виводу інформації

Зчитуємо стани кнопок та записуємо результати зчитування у зміну (рис. 3.22).

```
uint8_t buttons = 0b00000000;

if (HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin) == GPIO_PIN_SET) {
    buttons = buttons | 0b00000001;
}
if (HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin) == GPIO_PIN_SET) {
    buttons = buttons | 0b00000010;
}
if (HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin) == GPIO_PIN_SET) {
    buttons = buttons | 0b00000100;
}
if (HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin) == GPIO_PIN_SET) {
    buttons = buttons | 0b00001000;
}

tx_data[1] = buttons;
```

Рисунок 3.22 – Лістинг зчитування кнопок

Виводимо на екран стани кнопок в тому випадку, якщо її стан було змінено (рис. 3.23).

```
if (HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin) != B1o) {
    uint16_t color;

    if (HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin) == GPIO_PIN_SET) color = GREEN;
    else color = RED;

    ST7789_DrawFilledRectangle(14, 50, 45, 45, color);

    B1o = HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin);
}

if (HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin) != B2o) {
    uint16_t color;

    if (HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin) == GPIO_PIN_SET) color = GREEN;
    else color = RED;

    ST7789_DrawFilledRectangle(69, 50, 45, 45, color);

    B2o = HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin);
}

if (HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin) != B3o) {
    uint16_t color;

    if (HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin) == GPIO_PIN_SET) color = GREEN;
    else color = RED;

    ST7789_DrawFilledRectangle(126, 50, 45, 45, color);

    B3o = HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin);
}

if (HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin) != B4o) {
    uint16_t color;

    if (HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin) == GPIO_PIN_SET) color = GREEN;
    else color = RED;

    ST7789_DrawFilledRectangle(181, 50, 45, 45, color);

    B4o = HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin);
}
```

Рисунок 3.23 – Лістинг виводу станів кнопок

Зчитуємо позиції джойстиків за допомогою АЦП, отримані значення з діапазону 0..4096, переводимо в 0..255 (рис. 3.24).

```
HAL_ADC_Start_DMA(&hadc1, ADC_VAL, ADC_Channels_Count);

tx_data[2] = (uint8_t) map(ADC_VAL[0], 0, 4096, 255, 0);
tx_data[3] = (uint8_t) map(ADC_VAL[1], 0, 4096, 0, 255);
tx_data[4] = (uint8_t) map(ADC_VAL[2], 0, 4096, 255, 0);
tx_data[5] = (uint8_t) map(ADC_VAL[3], 0, 4096, 0, 255);
```

Рисунок 3.24 – Лістинг зчитування позицій джойстиків

Розраховуємо положення джойстиків та виводимо ці дані на екран (рис. 3.25)

```
ST7789_DrawFilledRectangle(14 + j1X, 120 + j1Y, 2, 2, BLACK);

ST7789_DrawFilledRectangle(126 + j2X, 120 + j2Y, 2, 2, BLACK);

j1X = map(ADC_VAL[0], 0, 4096, 97, 3);
j1Y = map(ADC_VAL[1], 0, 4096, 3, 97);

j2X = map(ADC_VAL[2], 0, 4096, 97, 3);
j2Y = map(ADC_VAL[3], 0, 4096, 3, 97);

ST7789_DrawFilledRectangle(14 + j1X, 120 + j1Y, 2, 2, RED);

ST7789_DrawFilledRectangle(126 + j2X, 120 + j2Y, 2, 2, RED);
```

Рисунок 3.25 – Лістинг зчитування позицій джойстиків

Реєструємо переривання таймера, для викликання коду надсилання зчитаних даних (рис. 3.26).

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if (htim->Instance == TIM2) {
        nrf24l01p_tx_transmit(tx_data);
    }
}
```

Рисунок 3.26 – Лістинг відправлення даних

Завершивши написання коду прошиваємо іншу плату мікроконтролера.

3.5 Тестування системи керування FPV дроном для симулятора польоту

Прототип був розроблений з метою перевірки працездатності комп’ютеризованої системи керування FPV дроном для симулятора польоту (рис. 3.27).

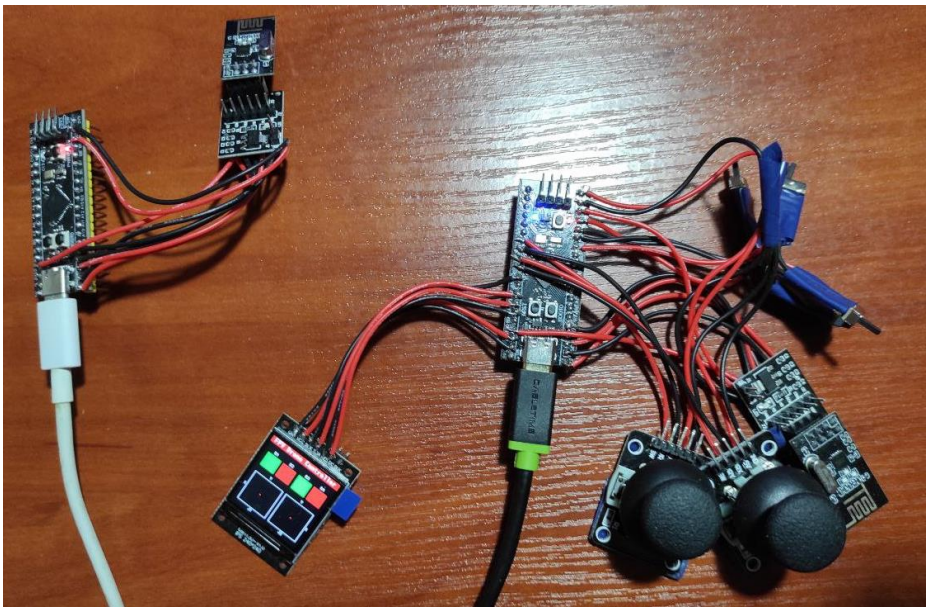

Рисунок 3.27 – Прототип системи керування FPV дроном для симулятора польоту

У ході тестування було використано апаратну складову, що включає дві плати мікроконтролерів STM32F411, одна з яких відповідає за роботу передавача, а інша – за роботу приймача. Для забезпечення бездротової передачі даних між передавачем і приймачем було використано два модулі NRF24L01.

У процесі тестування перевірялася коректність роботи IPS-дисплея, який використовується для виведення даних з джойстиків та індикації режимів роботи перемикачів. Також було перевірено функціональність двох модулів джойстиків і чотирьох перемикачів, які забезпечують управління дроном і зміну режимів роботи системи.

Результати коректного зчитування та відображення даних зображено на рис. 3.28.

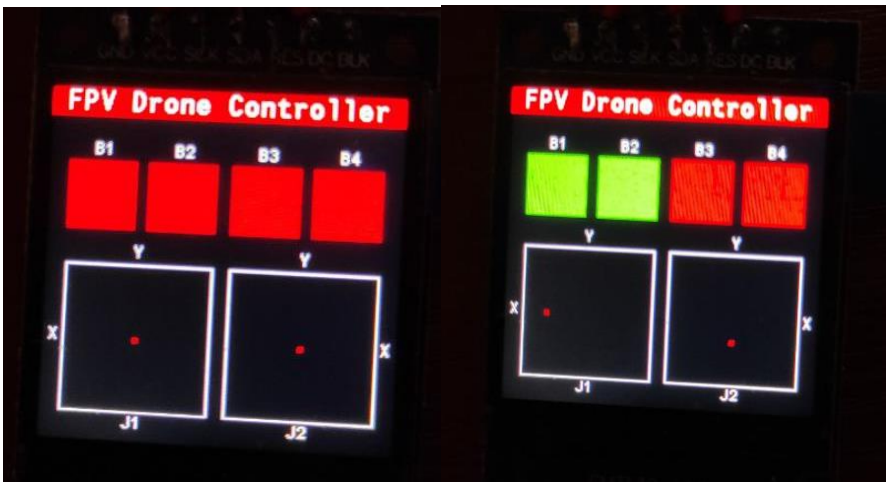

Рисунок 3.28 – Перевірка виводу даних на екран

Перевірка коректного отримання даних на комп'ютер через приймач зображено на рис. 3.29.

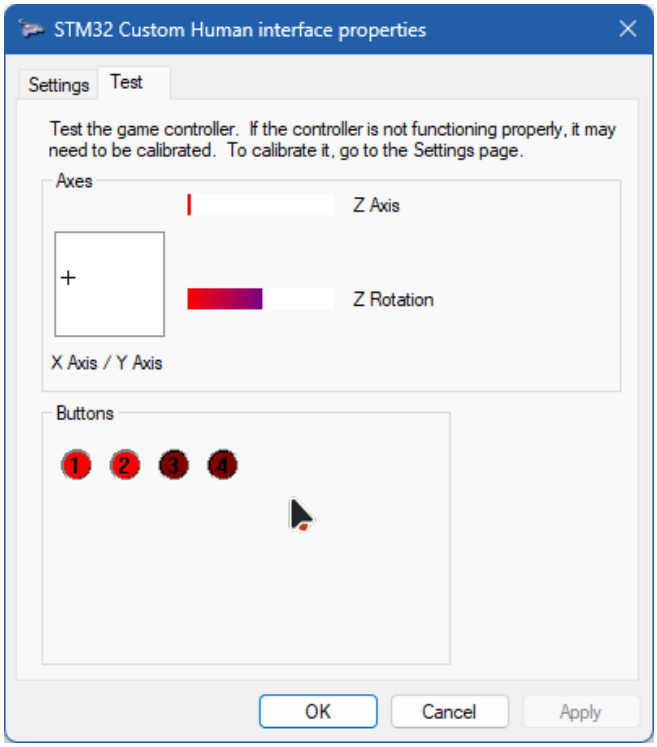

Рисунок 3.29 – Перевірка отриманих даних

Параметри X Axis і Y Axis відповідають першому джойстику, а параметри Z Axis і Z Rotation – другому джойстику. Кнопки 1-4 відповідають за перемикачі режимів.

За результатами тестування було підтверджено, що розроблена комп'ютеризована система керування FPV дроном для симулятора польоту є працездатною та забезпечує точне і надійне керування дроном в умовах симуляції. Система продемонструвала стабільність у передачі команд і даних, а також ефективність інтеграції всіх апаратних компонентів для досягнення заданих функціональних можливостей.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Розробка комп'ютеризованої системи є складним процесом, що вимагає високого рівня професіоналізму та знань у різних галузях, таких як програмування, електроніка та охорона праці. Безпека життєдіяльності і охорона праці є важливими аспектами, що повинні бути враховані на всіх етапах розробки і тестування системи.

Основні види навчання працюючих.

Навчання працюючих з охорони праці включає декілька видів, кожен з яких має свої особливості та спрямованість. Типове положення про порядок проведення навчання працюючих з охорони праці передбачає [18]:

- вступний інструктаж проводиться для нових працівників, а також для працівників, які переводяться на нові робочі місця або нові посади. Метою вступного інструктажу є ознайомлення працівників з основними правилами безпеки, характерними для даного підприємства, а також з потенційними небезпеками, що можуть виникнути під час роботи;
- первинний інструктаж проводиться безпосередньо на робочому місці перед початком самостійної роботи. Включає детальне ознайомлення з робочим обладнанням, інструментами та методами їх використання, а також з потенційними ризиками, що можуть виникнути під час виконання завдань;
- повторний інструктаж проводиться регулярно (раз на півроку) для всіх працівників з метою оновлення знань з охорони праці. Особлива увага приділяється новим нормативним актам і змінам в технологічному процесі;

- позаплановий інструктаж проводиться у випадках зміни умов праці, впровадження нових технологій, порушення правил охорони праці або після нещасних випадків на виробництві;

- цільовий інструктаж проводиться перед виконанням разових небезпечних робіт, таких як технічне обслуговування обладнання або ремонтні роботи.

Ефективне навчання з охорони праці передбачає використання різноманітних методів і засобів, таких як лекції, семінари, тренінги, практичні заняття та демонстрації. Важливо забезпечити інтерактивний характер навчання, щоб працівники могли задавати питання і обговорювати реальні ситуації.

Лекції та семінари включають теоретичні аспекти охорони праці, такі як нормативні вимоги, принципи безпеки, методи оцінки ризиків. Лектори повинні бути кваліфікованими спеціалістами з відповідними сертифікатами. Практичні заняття включають демонстрацію безпечного використання обладнання, вправи на виявлення потенційних небезпек та надання першої допомоги. Інтерактивні тренінги використовують методи моделювання реальних ситуацій, що дозволяє працівникам розвивати навички прийняття рішень у критичних умовах. Після завершення навчання проводиться тестування або перевірка знань з метою оцінки засвоєного матеріалу. Це може бути письмовий тест, усне опитування або практичне завдання.

Впровадження системи охорони праці в процес розробки систем:

- враховувати потенційні ризики при плануванні проекту, розробляти заходи з їх мінімізації;

- використовувати безпечні матеріали та методи, забезпечувати відповідність нормативним вимогам;

- проводити тестування системи в контрольованих умовах, з дотриманням всіх вимог безпеки;

- забезпечити працівників необхідними інструкціями та засобами захисту для безпечної експлуатації системи.

Навчання працюючих і інструктажі з охорони праці є критично важливими елементами в процесі розробки і впровадження комп'ютеризованої системи керування FPV дроном для симулятора польоту. Відповідальний підхід до цих питань дозволяє мінімізувати ризики для працівників, забезпечити безпечні умови праці та підвищити ефективність виробничого процесу.

4.2 Заходи щодо захисту установки від короткого замикання

Будь-які електронні пристрої піддаються ризику короткого замикання, яке може призвести до серйозних пошкоджень обладнання, втрати даних та навіть до пожежі. Тому заходи щодо захисту від короткого замикання є критично важливими для забезпечення безпеки життєдіяльності та охорони праці.

Коротке замикання відбувається, коли електричний струм обходить нормальний шлях через провідник з низьким опором, що призводить до надмірного виділення тепла. Основні причини включають [19]:

- механічні пошкодження проводів можуть виникнути через неакуратну установку, зношування або випадкові удари;
- використання неправильних або погано закріплених з'єднань;
- волога може спричинити корозію та створити шляхи для короткого замикання;
- використання компонентів з недостатньою потужністю або неправильне підключення додаткового обладнання призводить до перевантаження системи.

Заходи щодо захисту установки від короткого замикання:

а) використання високоякісних компонентів:

- забезпечення використання тільки сертифікованих та перевірених компонентів, які мають відповідні показники надійності та безпеки;
- регулярна перевірка стану проводів та конекторів на наявність зношування або пошкоджень;

б) ізоляція проводів:

- використання проводів з відповідною ізоляцією, що може витримувати необхідні електричні навантаження;

- розміщення проводів у спеціальних каналах або захисних оболонках для запобігання механічним пошкодженням та занесенню вологи;

в) застосування запобіжників і автоматичних вимикачів:

- встановлення запобіжників у ключових точках електричної системи для відключення живлення у разі перевантаження або короткого замикання;

- використання автоматичних вимикачів, які можуть миттєво реагувати на зміни в струмі та запобігти пошкодженню компонентів;

г) контроль та моніторинг електричних параметрів:

- використання систем моніторингу для постійного контролю за станом електричної мережі, що дозволяє виявляти та реагувати на відхилення в параметрах струму;

- регулярна діагностика системи з використанням спеціалізованого обладнання для виявлення потенційних проблем;

г) захист від вологи та пилу:

- розміщення компонентів у водонепроникних і пилонепроникних корпусах, особливо якщо система використовується у вологих або запилених умовах;

- забезпечення адекватної вентиляції для запобігання накопиченню вологи всередині корпусу;

е) правильне проектування системи:

- ретельне планування електричної схеми з урахуванням усіх можливих ризиків короткого замикання;

- використання схем із додатковими рівнями захисту, такими як захисні діоди та фільтри, які можуть запобігти поширенню короткого замикання.

Заходи щодо захисту комп'ютеризованих систем керування FPV дроном для симулятора польоту від короткого замикання є необхідною складовою забезпечення безпеки життєдіяльності та охорони праці. Вони включають

використання якісних компонентів, належну ізоляцію, встановлення запобіжників та автоматичних вимикачів, контроль електричних параметрів, захист від вологи та пилу, а також правильне проектування системи. Всі ці заходи спрямовані на мінімізацію ризиків та забезпечення надійної роботи обладнання.

4.3 Можливість виникнення статичної електрики та заходи боротьби з нею

Статична електрика є поширеною проблемою в електронних системах. Ця проблема може призвести до пошкодження електронних компонентів, порушення роботи системи та створення небезпечних умов для користувачів. Розуміння природи статичної електрики та вживання відповідних заходів для її запобігання є ключовими аспектами забезпечення безпеки життєдіяльності та основ охорони праці при роботі з такими системами.

Статична електрика виникає в результаті нерівномірного розподілу електричних зарядів на поверхнях матеріалів. Основні причини її виникнення включають [20]:

- тертя між різними матеріалами, наприклад, при руху частин системи;
- коли два матеріали контактують і потім розділяються, електрони можуть переноситися з одного матеріалу на інший, створюючи різницю потенціалів;
- присутність зарядженого об'єкту поблизу не зарядженого може викликати перерозподіл зарядів у матеріалі.

Негативні наслідки статичної електрики включають:

- електростатичні розряди (ESD) можуть спричинити пробої напівпровідникових приладів, знищуючи їх або знижуючи їхню ефективність;
- статична електрика може викликати збій у передачі даних або неправильну роботу контролера, що призводить до помилок у системі;
- в деяких випадках, особливо при роботі з високовольтними компонентами, електростатичні розряди можуть створити ризик для здоров'я користувача.

Для запобігання виникненню та негативному впливу статичної електрики в системах слід вживати наступні заходи:

- використання антистатичних килимків та робочих поверхонь, які підключені до заземлення для нейтралізації накопичених зарядів;
- використання спеціального антистатичного одягу та взуття для персоналу, що працює з електронними компонентами, щоб зменшити накопичі заряди на тілі;
- підтримка оптимальної вологості в приміщенні (близько 40-60%) для зниження накопичених статичних зарядів на поверхнях;
- використання іонізаторів повітря допомагає нейтралізувати заряди в повітрі та на поверхнях.
- зберігання та транспортування електронних компонентів у спеціальних антистатичних пакетах або контейнерах для запобігання накопиченню зарядів.
- використання антистатичних рідин або спреїв для обробки поверхонь, щоб зменшити їх здатність накопичувати заряди;
- проведення тренінгів та інструктажів для працівників щодо запобігання виникненню статичної електрики та методів захисту від неї.

Захист від статичної електрики є критично важливим аспектом безпеки життєдіяльності та охорони праці при роботі з комп'ютеризованими системами керування FPV дронами для симуляторів польоту. Розуміння причин виникнення статичних зарядів та впровадження ефективних заходів захисту допомагає запобігти пошкодженню обладнання та забезпечити безпечні умови праці. Враховуючи ці заходи, можна мінімізувати ризики, пов'язані зі статичною електрикою, та забезпечити стабільну та безпечну роботу системи.

ВИСНОВКИ

У кваліфікаційній роботі розроблено комп’ютеризовану систему керування FPV дроном, призначену для підключення до існуючих симуляторів польоту. Це розширює можливості тренувань та навчання операторів безпілотних літальних апаратів. У процесі виконання роботи досягнуто таких результатів:

В роботі проаналізовано існуючі рішення в області симуляції польотів FPV дронів, що дозволило виявити ключові особливості та недоліки наявних систем. На основі отриманих результатів визначено напрямки вдосконалення, які стали основою для розробки системи керування, що забезпечує більш точне та стабільне управління дроном у віртуальному середовищі.

Створено апаратну частину системи, яка включає розробку структурної та електричної принципової схем, а також вибір компонентів, що забезпечують необхідний рівень точності та надійності роботи.

Окрім цього, розроблено програмне забезпечення, яке реалізує алгоритм на мікроконтролерах і забезпечує інтеграцію з віртуальним середовищем симулятора. Це програмне забезпечення характеризується високою швидкістю обробки даних і стабільністю роботи, що дозволяє досягти високої точності моделювання польотів.

Проведене тестування підтвердило ефективність та надійність створеної системи, яка демонструє високі показник.

Результати роботи показують, що розроблена система керування FPV дроном є ефективним інструментом для навчання та тренування операторів, забезпечуючи безпечне середовище для відпрацювання складних маневрів та тестування нових алгоритмів управління.

Таким чином, створена система не лише знижує ризики, пов’язані з реальними польотами, але й відкриває нові можливості для досліджень у цій сфері.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

2. Лещишин Ю. З., Міська І. В., Назаревич Т. О.. Структурне моделювання цифрових каналів зв'язку. Матеріали VIII науково-технічної конференції „Інформаційні моделі, системи та технології". ТНТУ, 2020. С. 126.

3. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. Актуальні задачі сучасних технологій: збірник тез доповідей X міжнародної науково-технічної конференції молодих учених та студентів. Тернопіль: ТНТУ. 2021. С. 112.

4. Паламар А. М. Комп'ютерна система для моніторингу параметрів джерел безперебійного живлення на основі технології Internet of Things. Матеріали IV Міжнародної науково-технічної конференції «Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій», Тернопіль. 2019. С. 208–209.

5. Волоський В. П., Лещишин Ю. З., Романишин Н. Р.. Комп'ютерна система контролю та балансування літій-іонних акумуляторних батарей, Тернопіль 2021. Том I. С. 87–88.

6. Контролер Spektrum RC DX6e 6 Channel Full Range DSMX Transmitter. URL: https://www.ebay.com/itm/123932683505 (дата звернення: 02.04.2024).

7. Контролер FPV-комплекс Jumper T-Lite V2 ELRS M2. URL: https://prom.ua/ua/p2141730066-fpv-pult-jumper.html (дата звернення: 02.04.2024).

8. STM32F411xC datasheet. URL: https://www.st.com/resource/en/datasheet/stm32f411cc.pdf (дата звернення: 10.04.2024).

9. IPS-дисплей 1.3" SPI 240x240. URL: https://arduino.ua/ru/prod4590-ips-

displei-1-3-spi-240x240-rgb (дата звернення: 15.04.2024).

10. Ai-Thinker радіомодуль NRF24L01. URL: https://arduino.ua/prod231-radiomodyl-nrf24l01-2-4-ggc (дата звернення: 19.04.2024).

11. Адаптер модулей NRF24L01. URL: https://arduino.ua/ru/prod2743-adapter-modylei-nrf24l01-dlya-arduino (дата звернення: 21.04.2024).

12. Introduction to STM32CubeIDE URL: https://wiki.stmicroelectronics.cn/stm32mcu/wiki/STM32CubeIDE:Introduction_to_STM32CubeIDE (дата звернення: 02.05.2024).

13. Device Class Definition for Human Interface Devices (HID). URL: https://www.usb.org/sites/default/files/hid1_11.pdf (дата звернення: 05.05.2024).

14. Understanding HID report descriptors. URL: https://who-t.blogspot.com/2018/12/understanding-hid-report-descriptors.html (дата звернення: 05.05.2024).

15. HID Descriptor Tool. URL: https://www.usb.org/document-library/hid-descriptor-tool (дата звернення: 18.05.2024).

16. Бібліотека модуля NRF24L01 stm32_hal_nrf24l01p. URL: https://github.com/mokhwasomssi/stm32_hal_nrf24l01p (дата звернення: 20.05.2024).

17. Бібліотека модуля IPS-дисплею ST7789 ST7789-STM32. URL: https://github.com/mokhwasomssi/stm32_hal_nrf24l01p (дата звернення: 02.06.2024).

18. Типове положення про порядок проведення навчання та перевірку знань з питань охорони праці. (від 26.01.2005 р., № 15, із змінами від 30.01.2017 р., № 140 – НПАОП 0.00-4.36-05).

19. Коротке замикання: причини, класифікація, захист. URL: https://presa.com.ua/aktualne/korotke-zamikannya-prichini-klasifikatsiya-zakhist.html (дата звернення: 04.06.2024).

20. ДСТУ 7302:2013 Статична електрика. Терміни та визначення основних понять.

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп’ютерно-інформаційних систем і програмної інженерії

Кафедра комп’ютерних систем та мереж

«ЗАТВЕРДЖУЮ»

Завідувач кафедри КС

Осухівська Г.М.

“ 09 ” 02 2024р.

КОМП’ЮТЕРИЗОВАНА СИСТЕМА КЕРУВАННЯ FPV ДРОНОМ ДЛЯ СИМУЛЯТОРА ПОЛЬОТУ

ТЕХНІЧНЕ ЗАВДАННЯ

на 7 листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп’ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

к.т.н. Лещишин Ю. З.

“ 08 ” 02 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

Міндзіброцький Р. С.

“ 08 ” 02 2024 р.

Тернопіль 2024

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи бакалавра: «Комп'ютеризованої системи керування FPV дроном для симулятора польоту»

Умовне позначення дипломного проекту: КС КРБ 123 121.00.00.

1.2 Виконавець

Студент групи СІ-41, факультету комп’ютерно-інформаційних систем і програмної інженерії, кафедри комп’ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Міндзіброцький Роман Сергійович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи бакалавра є наказ по університету № 4/7-408 від «24» квітня 2024 року

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи бакалавра – 24.04.2023 р.

Плановий термін завершення виконання кваліфікаційної роботи бакалавра – 28.06.2024 р.

1.5 Порядок оформлення та пред’явлення результатів роботи

Оформлення технічної документації до кваліфікаційної роботи бакалавра здійснюється згідно діючих вимог вітчизняних та міжнародних стандартів. Технічна документація до кваліфікаційної роботи бакалавра включає в себе текст пояснювальної записки та креслення, які максимально інформативно та стисло відображають основні результати розробки комп’ютеризованої системи керування FPV дроном для симулятора польоту. Основними регламентними документами при оформленні та пред’явлені результатів проектування є групи діючих стандартів ДСТУ, ГОСТ, ISO та ЄСКД, ЕСПД. Пред’явлення результатів кваліфікаційної роботи бакалавра відбувається шляхом захисту роботи на відповідному засіданні ДЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Система призначена для реалізації системи керування FPV дроном для симулятора польоту.

2.2 Мета створення системи

Метою створення комп'ютеризованої системи дистанційного керування FPV дроном для симулятора польоту є розробка інноваційного рішення, яке дозволить користувачам ефективно та зручно керувати дроном у віртуальному середовищі навіть на відстані. Це дозволить покращити якість управління дроном шляхом впровадження автоматизованого, дистанційного та ефективного підходу, що забезпечує зручність, економію ресурсів та підвищений комфорт для користувачів.

2.3 Характеристика об'єкту

Система проектується для дистанційного керування FPV дроном для симулятора польоту, що включає в себе:

- розробку функціональної та структурної схеми;
- розробку схеми електричної принципової;
- розробку алгоритму роботи та програмного забезпечення для мікроконтролера.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп’ютеризована система керування FPV дроном для симулятора польоту повинна забезпечити:

1. Мінімальну затримку при передачі;
2. Точність передачі даних;
3. Інтеграцію з готовими симуляторами польоту;
4. Оптимізацію ресурсів.

3.1.1 Вимоги до структури та функціонування системи

Структура системи керування FPV дроном для симулятора польоту включає в себе:

- мікроконтролери, які забезпечують загальне керування функціонуванням системи;
- джойстики та перемикачі;
- екран;
- передавач ти приймач.

В загальному випадку, структура системи повинна реалізовувати функції керування FPV дроном для симулятора польоту. Основні функціональні вимоги

характеризуються наступними критеріями:

- функціональність;
- точність;
- ефективність;
- надійність;
- сумісність.

3.1.2 Вимоги до способів та засобів зв’язку між компонентами системи

Обмін даними між контролером та приймачем повинен здійснюватися з використанням бездротових технологій передачі інформації.

3.1.3 Вимоги до надійності системи

Система повинна бути захищена від фізичних чи механічних пошкоджень на рівні апаратного та програмного забезпечення. Надійність системи повинна забезпечувати відновлюваність функціонування у випадку збою апаратного чи програмного забезпечення.

3.1.7 Вимоги до апаратного забезпечення

Вимоги до елементної бази розробки:

- режими роботи і умови експлуатації вибраних елементів повинні відповідати вказаним в ТЗ;
- елементна база по можливості має бути широковживаною, доступною і дешевою. Необхідно також враховувати можливість заміни вибраних елементів на аналогічні (вітчизняні чи імпортного виробництва).

Вимоги до мікроконтролера:

- мікроконтролер має підтримувати RISC архітектуру команд;
- мікроконтролер повинен містити необхідний набір вбудованих периферійних пристроїв (таймери, АЦП і т.п.) та потрібну кількість керованих

портів введення /виведення.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ.

Комплект конструкторської документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:

1. схема з’єднань (приймача, передавача);
2. структурна схема;
3. схема електрична принципова (приймача, передавача);
4. блок-схема алгоритму роботи (приймача, передавача).

*Примітка: В комплект конструкторської документації можуть вноситися зміни та доповнення в процесі розробки.

5 Техніко-економічні показники

Собівартість розробки системи повинна становити не більше 3000 грн.

Термін експлуатації системи повинен бути не менший 10 років.

*Примітка: собівартість системи може змінюватись під час розрахунку в процесі розробки.

6 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання КРБ

№ етапу	Назва етапу виконання КРБ	Термін виконання
1	Розробка і затвердження технічного завдання	01.02 – 09.02
2	Аналіз технічного завдання	05.02 – 11.02
3	Аналіз вимог до системи керування FPV дроном	25.04 – 03.05
4	Вибір елементної бази	04.05 – 11.05
5	Розробка структурної та функціональної схеми	12.05 – 17.05
6	Розробка програмного забезпечення для проектованої частини	18.05 – 29.05
7	Тестування системи керування FPV дроном	29.05 – 02.06
8	Безпека життєдіяльності, основи охорони праці	03.06 – 07.06
9	Оформлення кваліфікаційної роботи	08.06 – 14.06
10	Попередній захист кваліфікаційної роботи	14.06
11	Захист кваліфікаційної роботи	24.06 – 28.06

7 Додаткові умови виконання кваліфікаційної роботи бакалавра

Під час виконання кваліфікаційної роботи в дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Перелік елементів

Позн.	*Найменування*	*К-сть*	*Примітка*
	Модулі		
U1	Плата STM32F411CEU6	1	
U2	Модуль отримання даних NRF24L01	1	
	Роз’єми		
X1	USB-C CX90BW-16P	1	

Зм.	*Арк.*	*№ документа*	*Підпис*	*Дата*	КС КРБ 123.121.00.01 ПЕЗ			
Розробив		Міндзіброцький Р.С.			*Приймач системи керування FPV дроном для симулятора польоту* *Перелік елементів*	*Літ*	*Аркуш*	*Аркушів*
Перевірив		Лещишин Ю.З.						
Консульт.						*ТНТУ, каф. КС, гр. СІ-41*		
Н. контр.		Тиш Є.В.						
Зав. каф.		Осухівська Г.М.						

Позн.	Найменування	К-сть	Примітка
	Давачі		
Q1…Q4	Перемикачі SW-TO-MTS-102-A2	4	
	Модулі		
S1	Дисплей ST7789-IPS	1	
U1	Плата STM32F411CEU6	1	
U2…U3	Модуль джойстика RKJXV1224002	2	
U4	Модуль передачі даних NRF24L01	1	

Зм.	Арк.	№ документа	Підпис	Дата	КС КРБ 123.121.00.02 ПЕЗ			
Розробив		Міндзіброцький Р.С.			Передавач системи керування FPV дроном для симулятора польоту Перелік елементів	Літ	Аркуш	Аркушів
Перевірив		Лещишин Ю.З.						
Консульт.						ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Зав. каф.		Осухівська Г.М.						

Позн.	Найменування	К-сть	Примітка
	Модулі		
U1	Плата STM32F411CEU6	1	
U2	Модуль отримання даних NRF24L01	1	

Зм.	Арк.	№ документа	Підпис	Дата	КС КРБ 123.121.00.0 ПЕ4			
Розробив		Міндзіброцький Р.С.			Приймач системи керування FPV дроном для симулятора польоту Перелік елементів	Літ	Аркуш	Аркушів
Перевірив		Лещишин Ю.З.						
Консульт.						ТНТУ, каф. КС, гр. СІ-41		
Н. контр.		Тиш Є.В.						
Зав. каф.		Осухівська Г.М.						

Позн.	*Найменування*	*К-сть*	*Примітка*
	Давачі		
Q1…Q4	Перемикачі SW-TO-MTS-102-A2	4	
	Модулі		
S1	Дисплей ST7789-IPS	1	
U1	Плата STM32F411CEU6	1	
U2…U3	Модуль джойстика KY-023	2	
U4	Модуль передачі даних NRF24L01	1	

Зм.	*Арк.*	*№ документа*	*Підпис*	*Дата*	КС КРБ 123.121.00.0 ПЕ4			
Розробив		Міндзіброцький Р.С.			*Передавач системи керування FPV дроном для симулятора польоту* *Перелік елементів*	*Літ*	*Аркуш*	*Аркушів*
Перевірив		Лещишин Ю.З.						
Консульт.						*ТНТУ, каф. КС, гр. СІ-41*		
Н. контр.		Тиш Є.В.						
Зав. каф.		Осухівська Г.М.						

Додаток В

Лістинг програмного забезпечення

Лістинг В.1 – Код програми мікроконтролера для приймача.

```
#include "main.h"
#include "spi.h"
#include "usb_device.h"
#include "gpio.h"

#include "nrf24l01p.h"

#pragma pack(push, 1)
struct myReportStruct {
     int8_t X;
     int8_t Y;
     int8_t Z;
     int8_t Rz;
     uint8_t Buttons;
};
#pragma pack(pop)

struct myReportStruct FPV_Controller;

extern USBD_HandleTypeDef hUsbDeviceFS;

#define MAX(x, y) (((x) > (y)) ? (x) : (y))
#define MIN(x, y) (((x) < (y)) ? (x) : (y))

void SystemClock_Config(void);

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin);

int main(void) {
  HAL_Init();

  SystemClock_Config();

  MX_GPIO_Init();
  MX_SPI2_Init();
  MX_USB_DEVICE_Init();

  nrf24l01p_rx_init(2500, _1Mbps);
  nrf24l01p_set_rf_tx_output_power(_18dBm);

  FPV_Controller.X = 128;
  FPV_Controller.Y = 128;
  FPV_Controller.Z = 128;
  FPV_Controller.Rz = 128;'
```

```
  while (1) {
       USBD_CUSTOM_HID_SendReport(&hUsbDeviceFS,    (uint8_t    *)
&FPV_Controller, sizeof(FPV_Controller));
       HAL_Delay(10);
  }
}

void SystemClock_Config(void) {
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 25;
  RCC_OscInitStruct.PLL.PLLN = 336;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
  RCC_OscInitStruct.PLL.PLLQ = 7;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK) {
    Error_Handler();
  }

  RCC_ClkInitStruct.ClockType                                    =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK

|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if  (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,  FLASH_LATENCY_2)  !=
HAL_OK) {
    Error_Handler();
  }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
     if(GPIO_Pin == NRF24L01P_IRQ_PIN_NUMBER) {
          static uint8_t rx_data[NRF24L01P_PAYLOAD_LENGTH] = { 0, };

          nrf24l01p_rx_receive(rx_data);

          FPV_Controller.Buttons = rx_data[1];
          FPV_Controller.X = MIN(MAX(rx_data[2], 0), 255);
          FPV_Controller.Y = MIN(MAX(rx_data[3], 0), 255);
          FPV_Controller.Z = MIN(MAX(rx_data[4], 0), 255);
          FPV_Controller.Rz = MIN(MAX(rx_data[5], 0), 255);
```

```
    }
}

void Error_Handler(void){
  __disable_irq();
  while (1) {}
}
```

Лістинг В.2 – Код програми мікроконтролера для передавача.

```
#include "main.h"
#include "adc.h"
#include "dma.h"
#include "spi.h"
#include "tim.h"
#include "gpio.h"

#include "st7789.h"
#include "nrf24l01p.h"
#include <stdio.h>

#define MAX(x, y) (((x) > (y)) ? (x) : (y))
#define MIN(x, y) (((x) < (y)) ? (x) : (y))

uint8_t tx_data[NRF24L01P_PAYLOAD_LENGTH] = { 0, 0, 0, 0, 0, 0, 0,
0 };

void SystemClock_Config(void);

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin);
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim);

uint32_t ADC_VAL[4];
const int ADC_Channels_Count = sizeof(ADC_VAL) / sizeof(ADC_VAL[0]);

uint32_t map(uint32_t x, uint32_t in_min, uint32_t in_max, uint32_t
out_min, uint32_t out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min)
+ out_min;
}

int main(void) {
    uint8_t j1X = 62;
    uint8_t j1Y = 187;
    uint8_t j2X = 177;
    uint8_t j2Y = 177;

    GPIO_PinState B1o;
    GPIO_PinState B2o;
    GPIO_PinState B3o;
```

```
    GPIO_PinState B4o;

    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();
    MX_DMA_Init();
    MX_SPI2_Init();
    MX_ADC1_Init();
    MX_SPI1_Init();
    MX_TIM2_Init();

    HAL_Delay(200);

    ST7789_Init();
    nrf24l01p_tx_init(2500, _1Mbps);
    nrf24l01p_set_rf_tx_output_power(_18dBm);
    HAL_TIM_Base_Start_IT(&htim2);

    ST7789_Fill_Color(BLACK);

    ST7789_DrawFilledRectangle(0, 0, 240, 18, RED);
    ST7789_WriteString(10, 0, "FPV Drone Controller", Font_11x18,
WHITE, RED);

    ST7789_WriteString(30, 37, "B1", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(85, 37, "B2", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(142, 37, "B3", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(197, 37, "B4", Font_7x10, WHITE, BLACK);

    ST7789_WriteString(4, 165, "X", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(60, 107, "Y", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(57, 223, "J1", Font_7x10, WHITE, BLACK);

    ST7789_WriteString(229, 165, "X", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(173, 107, "Y", Font_7x10, WHITE, BLACK);
    ST7789_WriteString(169, 223, "J2", Font_7x10, WHITE, BLACK);

    ST7789_DrawRectangleWithSize(14, 120, 100, 100, WHITE);
    ST7789_DrawRectangleWithSize(126, 120, 100, 100, WHITE);

    B1o = !HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin);
    B2o = !HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin);
    B3o = !HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin);
    B4o = !HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin);

    while (1) {
        uint8_t buttons = 0b00000000;

        if    (HAL_GPIO_ReadPin(SW_1_GPIO_Port,    SW_1_Pin)    ==
GPIO_PIN_SET) {
            buttons = buttons | 0b00000001;
```

```
          }
          if    (HAL_GPIO_ReadPin(SW_2_GPIO_Port,    SW_2_Pin)    ==
GPIO_PIN_SET) {
               buttons = buttons | 0b00000010;
          }
          if    (HAL_GPIO_ReadPin(SW_3_GPIO_Port,    SW_3_Pin)    ==
GPIO_PIN_SET) {
               buttons = buttons | 0b00000100;
          }
          if    (HAL_GPIO_ReadPin(SW_4_GPIO_Port,    SW_4_Pin)    ==
GPIO_PIN_SET) {
               buttons = buttons | 0b00001000;
          }

          tx_data[1] = buttons;

          if (HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin) != B1o) {
               uint16_t color;

               if  (HAL_GPIO_ReadPin(SW_1_GPIO_Port,  SW_1_Pin)  ==
GPIO_PIN_SET) color = GREEN;
               else color = RED;

               ST7789_DrawFilledRectangle(14, 50, 45, 45, color);

               B1o = HAL_GPIO_ReadPin(SW_1_GPIO_Port, SW_1_Pin);
          }

          if (HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin) != B2o) {
               uint16_t color;

               if  (HAL_GPIO_ReadPin(SW_2_GPIO_Port,  SW_2_Pin)  ==
GPIO_PIN_SET) color = GREEN;
               else color = RED;

               ST7789_DrawFilledRectangle(69, 50, 45, 45, color);

               B2o = HAL_GPIO_ReadPin(SW_2_GPIO_Port, SW_2_Pin);
          }

          if (HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin) != B3o) {
               uint16_t color;

               if  (HAL_GPIO_ReadPin(SW_3_GPIO_Port,  SW_3_Pin)  ==
GPIO_PIN_SET) color = GREEN;
               else color = RED;

               ST7789_DrawFilledRectangle(126, 50, 45, 45, color);

               B3o = HAL_GPIO_ReadPin(SW_3_GPIO_Port, SW_3_Pin);
          }

          if (HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin) != B4o) {
```

```
                uint16_t color;

                if  (HAL_GPIO_ReadPin(SW_4_GPIO_Port,  SW_4_Pin)  ==
GPIO_PIN_SET) color = GREEN;
                else color = RED;

                ST7789_DrawFilledRectangle(181, 50, 45, 45, color);

                B4o = HAL_GPIO_ReadPin(SW_4_GPIO_Port, SW_4_Pin);
            }

            HAL_ADC_Start_DMA(&hadc1, ADC_VAL, ADC_Channels_Count);

            tx_data[2] = (uint8_t) map(ADC_VAL[0], 0, 4096, 255, 0);
            tx_data[3] = (uint8_t) map(ADC_VAL[1], 0, 4096, 0, 255);
            tx_data[4] = (uint8_t) map(ADC_VAL[2], 0, 4096, 255, 0);
            tx_data[5] = (uint8_t) map(ADC_VAL[3], 0, 4096, 0, 255);

            ST7789_DrawFilledRectangle(14  +  j1X,  120  +  j1Y,  2,  2,
BLACK);

            ST7789_DrawFilledRectangle(126  +  j2X,  120  +  j2Y,  2,  2,
BLACK);

            j1X = map(ADC_VAL[0], 0, 4096, 97, 3);
            j1Y = map(ADC_VAL[1], 0, 4096, 3, 97);

            j2X = map(ADC_VAL[2], 0, 4096, 97, 3);
            j2Y = map(ADC_VAL[3], 0, 4096, 3, 97);

            ST7789_DrawFilledRectangle(14  +  j1X,  120  +  j1Y,  2,  2,
RED);

            ST7789_DrawFilledRectangle(126  +  j2X,  120  +  j2Y,  2,  2,
RED);

            HAL_Delay(10);
        }
}

void SystemClock_Config(void) {
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 25;
  RCC_OscInitStruct.PLL.PLLN = 168;
```

```
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK) {
    Error_Handler();
  }

  RCC_ClkInitStruct.ClockType                                    =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK

|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
HAL_OK) {
    Error_Handler();
  }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
     if (GPIO_Pin == NRF24L01P_IRQ_PIN_NUMBER) nrf24l01p_tx_irq();
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
     if (hadc->Instance == ADC1) {
          HAL_ADC_Stop_DMA(&hadc1);
     }
}

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
     if (htim->Instance == TIM2) {
          nrf24l01p_tx_transmit(tx_data);
     }
}

void Error_Handler(void) {
     __disable_irq();
     while (1) {}
}
```