

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра інформаційних систем

Дипломна робота
на здобуття ступеня бакалавра

на тему “РОЗРОБКА НАВІГАЦІЙНОЇ СИСТЕМИ ДЛЯ ДРОНУ”

Виконала: студентка групи ПМІ-42с
спеціальності 122 – комп'ютерні науки
(шифр і назва спеціальності)

Матійчук С. В.
(підпис)(прізвище та ініціали)

Керівник доц. Козій І. Я.
(підпис) (прізвище та ініціали)

Рецензент доц. Огородник Н. П.
(підпис) (прізвище та ініціали)

2023

Факультет _____ прикладної математики та інформатики

Кафедра _____ інформаційних систем

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки _____

Спеціальність _____ 122 Комп'ютерні науки

«ЗАТВЕРДЖУЮ»

Зав. кафедрою
проф. Шинкаренко Г.А.

« 7 » вересня 2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ (КВАЛІФІКАЦІЙНУ) РОБОТУ СТУДЕНТА

Матійчук Соломії Василівни

(прізвище, ім'я, по батькові)

1. Тема роботи

Розробка навігаційної системи для дрону

керівник роботи доц. Козій І.Я.

затверджені Вченою радою факультету від « 13 » вересня 2022 р., № 15

2. Строк подання студентом роботи 12.06.2023

3. Вихідні дані до роботи _____

Література та інтернет-ресурси за тематикою роботи

1. PID контролер в дронах:

<https://oscarliang.com/pid/#Understanding-PID-in-FPV-Drones>

2. Навігаційні системи для дрона:

<https://www.immervision.com/drone-navigation-system/>

3. Обчислення відстані та азимута між двома географічними координатами:

<https://www.ancientportsantiques.com/wp-content/uploads/Documents/ETUDESarchivees/MedNavigationRoutes/MedNav/TrigoSpherique.pdf>

4. Визначення нахилу за допомогою акселерометра-гіроскопа:

<http://www.cas.mcmaster.ca/~rzheng/course/CAS765fa13/YueSun.pdf>

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. забезпечення трьох методів зв'язку з дроном

2. конфігурування сенсорів і модулів зв'язку для передачі даних через них

3. обчислення кутів нахилу дрону за початковими і кінцевими GPS координатами

4. вирівнювання дрона за допомогою PID контролера

5. розробка комп'ютерного застосунку для керування дроном

6. тестування та валідація розробленої навігаційної системи

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

Функціональна схема розміщення компонентів, блок-схеми до кожної з написаних програм, зображення інтерфейсу запущеного застосунку, зображення дрона та його компонентів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 14.09.22

КАЛЕНДАРНИЙ ПЛАН

№	Найменування етапів дипломної (кваліфікаційної) роботи	Строк виконання етапів роботи	Примітка
1.	<i>Постановка задачі</i>	<i>вересень 2022</i>	виконано
2.	<i>Вивчення існуючих систем та технологій для реалізації</i>	<i>жовтень 2022</i>	виконано
3.	<i>Розробка алгоритмів роботи програми</i>	<i>листопад 2022</i>	виконано
4.	<i>Створення програмної реалізації</i>	<i>грудень-березень 2023</i>	виконано
5.	<i>Оформлення роботи</i>	<i>травень 2023</i>	виконано

Студент _____ *Матійчук С. В.*
підписКерівник роботи _____ *доц Козій І.Я.*
підпис

Реферат

Дипломна робота: 64 ст., 47 рис., 36 джерел.

Предмет дослідження: система навігації та керування для дрона.

Метою даної роботи є розробка прототипу навігаційної системи для польоту дрона на невеликі дистанції; забезпечення трьох методів зв'язку: автопілот за допомогою GPS, зв'язок, використовуючи мобільну мережу, та радіозв'язок; розробка алгоритму обчислення кутів нахилу дрона; розробка комп'ютерного застосунку для передачі та відображення даних по GPS; забезпечення вирівнювання дрона при неочікуваному нахилі під дією зовнішніх сил.

Результатом роботи є комп'ютерний застосунок для керування дроном, використовуючи мобільний зв'язок та GPS; програма, завантажена в мікроконтролер, що знаходиться на дроні, і відповідає за обчислення кінцевих швидкостей моторів; програма, завантажена в мікроконтролер, що знаходиться на пульті управління, і відповідає за радіозв'язок; програма, що запущена на мінікомп'ютері Raspberry Pi, і відповідає за аналіз GPS координат та відповідні обчислення кутів нахилу дрона.

Ключові слова: дрон, GPS, географічна широта і довгота, мобільний зв'язок GPRS/GSM, кути нахилу roll (укр. крен) і pitch (укр. тантаж), азимут, bearing (укр. пеленг), course deviation angle, PID контролер, акселерометр-гіроскоп, IMU, мікроконтролер Arduino Nano, PWM (укр. ШІМ) сигнал, радіозв'язок, інтерфейси зв'язку I2C, SPI, UART, аналогово-цифровий перетворювач ADC (укр. АЦП), мінікомп'ютер Raspberry Pi, магнетометр, формула гаверсінуса, формула Ейлера для обчислення кутів.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП	5
РОЗДІЛ 1. КОНСТРУКТИВНО-ПАРАМЕТРИЧНИЙ АНАЛІЗ КОМПОНЕНТНОЇ БАЗИ.....	6
1.1 Загальний огляд	6
1.2 Raspberry Pi 3 Model A+.....	8
1.2.1 Raspberry Pi як елемент навігаційної системи для дрона.....	9
1.2.2 Операційна система Raspbian GNU/Linux 11 (bullseye).....	9
1.3 Arduino® Nano	10
1.3.1 Arduino Nano як контролер радіоприймача та для безпосереднього контролю польоту	11
1.3.2 Arduino Nano як контролер радіопередавача пульта управління.....	11
1.3.3 Програмне середовище Arduino IDE.....	12
1.4 Радіомодуль RF24L01	13
1.5 Акселерометр-гіроскоп MPU6050	13
1.6 GSM/GPRS/GPS модуль SIM868	15
1.6.1 GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi.....	16
1.6.2 Модуль SIM868 як GPS-приймач та пристрій для зв'язку по GSM/GPRS.....	17
1.7 Магнетометр QMC5883L.....	18
1.8 Двигуни та контролери двигунів	19
РОЗДІЛ 2. Розробка системи навігації для дрона.....	21
2.1 Аналіз та огляд предметної області.....	21
2.1.1 Опис предметної області	21
2.1.2 Огляд існуючих рішень навігаційних систем для дронів	22
2.1.3 Огляд використаних технологій та інтерфейсів	24
2.2 Постановка завдання	27
2.3 Функціональна схема	29
2.4 Розробка програмного забезпечення для застосування	32
2.4.1 Алгоритм роботи системи	33
2.4.2 Інтерфейс програми	33

2.5 Розробка програмного забезпечення для Raspberry Pi	37
2.5.1 Блок-схеми	38
2.5.2 Формула гаверсінуса.....	40
2.5.2 Алгоритм навігації за GPS.....	41
2.6 Розробка програмного забезпечення для Arduino	44
2.6.1 Блок-схема.....	46
2.6.2 Використання PID контролера для вирівнювання дрона і керуванням ним	47
2.6.3 Формула Ейлера для отримання кутів	50
2.7 Тестування та валідація навігаційної системи для дрона	51
2.7.1 Тестування навігації за GPS	52
2.7.2 Тестування мобільного зв'язку	56
2.7.3 Тестування радіозв'язку	57
2.7.4 Тестування вирівнювання дрона при неочікуваному нахилі	59
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БПЛА – Безпілотний Літальний Апарат.

Roll (укр. крен) - кутовий рух літального апарата відносно повздовжньої осі.

Pitch (укр. тангаж) - кутовий рух літального апарата відносно поперечної осі.

IDE - Integrated Development Environment.

Дебагер (англ. Debugger) – відлагоджувач, програма, що допомагає виявити помилки в програмному коді.

PWM (укр. ШІМ) – Pulse-Width Modulation (укр. широтно-імпульсна модуляція).

GSM – Global System for Mobile Communications.

GPRS – General Packet Radio Services.

GPS – Global Positioning System.

GNSS – Global Navigation Satellite System.

IMU – Inertial Measurement Unit.

I2C – Inter-Integrated Circuit (IIC). Послідовний синхронний інтерфейс.

UART – Universal Asynchronous Receiver/Transmitter (укр. універсальний асинхронний прийомо-передавач)

SPI – Serial Peripheral Interface. Шина для підключення периферійних пристроїв.

TCP – Transmission Control Protocol

IP – Internet Protocol

SIM – Subscriber Identity Module

ADC – Analog-to-digit converter

GUI – Graphical User Interface

PID – Proportional Integral Derivate

SSH – Secure Shell

ВСТУП

У сучасному світі дрони стали необхідною складовою розвитку технологій та військової техніки. Зокрема, в умовах війни в Україні попит на дрони значно збільшився, оскільки вони використовуються як засіб збору розвідувальної інформації, контролю за територією, а також як дрони-камікадзе. У зв'язку з цим, важливо забезпечити надійну систему комунікації та навігації для дронів. Тому, темою моєї дипломної роботи є "Розробка навігаційної системи для дрона".

Мета і завдання розробки. Основною метою дипломної роботи є розробка прототипу навігаційної системи для польоту дрона на невеликі дистанції. Основною перевагою даної роботи є можливість використання трьох методів зв'язку: автопілот за допомогою GPS, зв'язок по мобільній мережі та радіозв'язок. Такий підхід забезпечує надійність системи, оскільки у випадку втрати зв'язку по одному з цих методів, система автоматично переключається на резервний метод зв'язку.

Також, був розроблений алгоритм обчислення кутів обертання дрона, таких як крен та тантаж, на основі даних компаса та азимута між поточними GPS координатами і координатами пункту призначення.

Актуальність теми. На сьогоднішній день існує багато різних варіантів дронів та систем комунікації з ними. Проте, найвизначальнішим аспектом використання дронів є їх надійність, особливо у військових конфліктах. У зв'язку з цим, розробка навігаційної системи для дронів, яка забезпечує резервні методи зв'язку, має велике значення і залишається актуальною у наш час.

Бакалаврська кваліфікаційна робота складається з двох основних розділів, у кожному з яких висвітлена окрема тема цієї роботи. Дана послідовність подання матеріалу обґрунтована тим, що при ознайомленні з літературним оглядом читач досягає розуміння основних компонентів, використаних у розробці навігаційної системи для дрона, що далі логічно переходить в розгляд експериментальної частини даної бакалаврської кваліфікаційної роботи.

РОЗДІЛ 1. КОНСТРУКТИВНО-ПАРАМЕТРИЧНИЙ АНАЛІЗ КОМПОНЕНТНОЇ БАЗИ

1.1 Загальний огляд

Для розробки навігаційної системи для дрона було використано такі компоненти:

- Raspberry Pi 3 Model A+
- Arduino® Nano
- Радіомодуль RF24L01
- Акселерометр-гіроскоп MPU6050
- GSM/GPRS/GPS модуль SIM868
- Двигун
- Контролер двигуна
- Акумулятор
- Потенціометр
- Джойстик

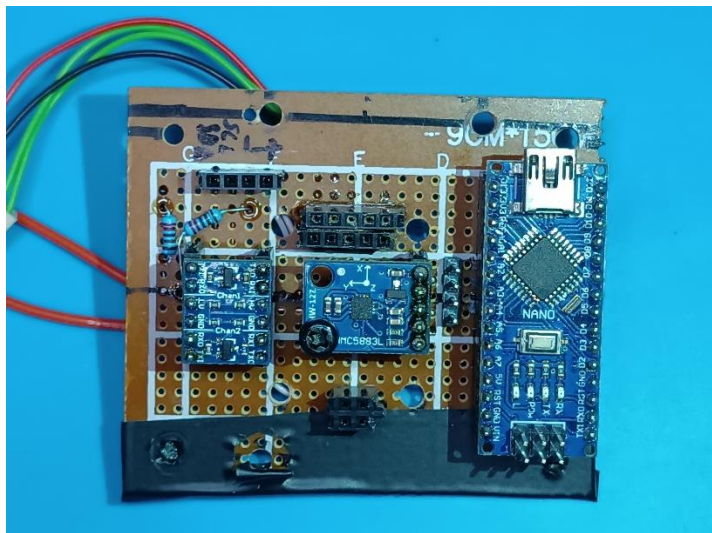


Рис. 1.1 – Додаткове Arduino Nano, магнітометр і перетворювач логічних рівнів 5V – 3.3V, розміщені на дроні

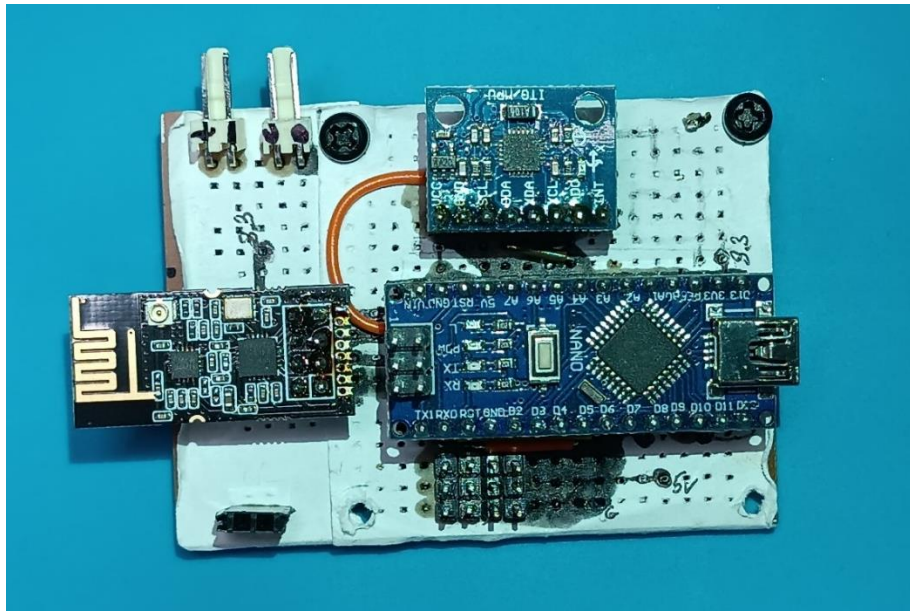


Рис. 1.2 – Arduino Nano для контролю польоту, IMU модуль, радіомодуль, розміщені на дроні

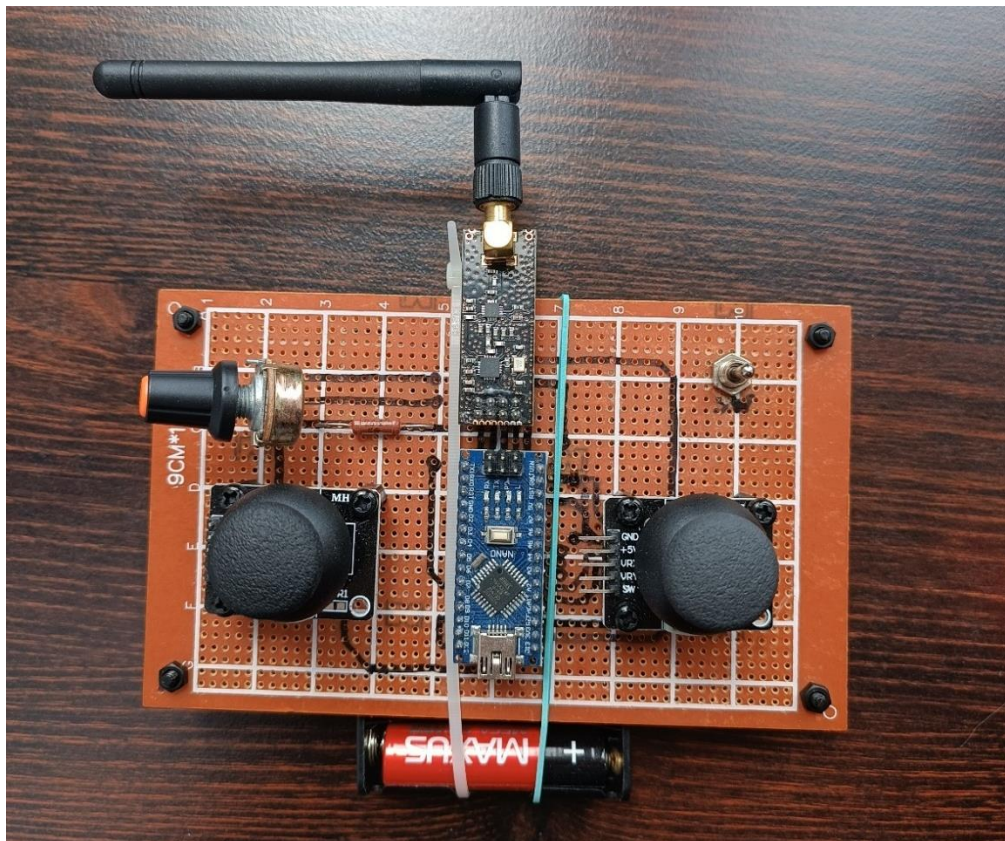


Рис. 1.3 – Пульти управління через радіозв'язок (Arduino Nano, 2 джойстика, потенціометр, радіомодуль)

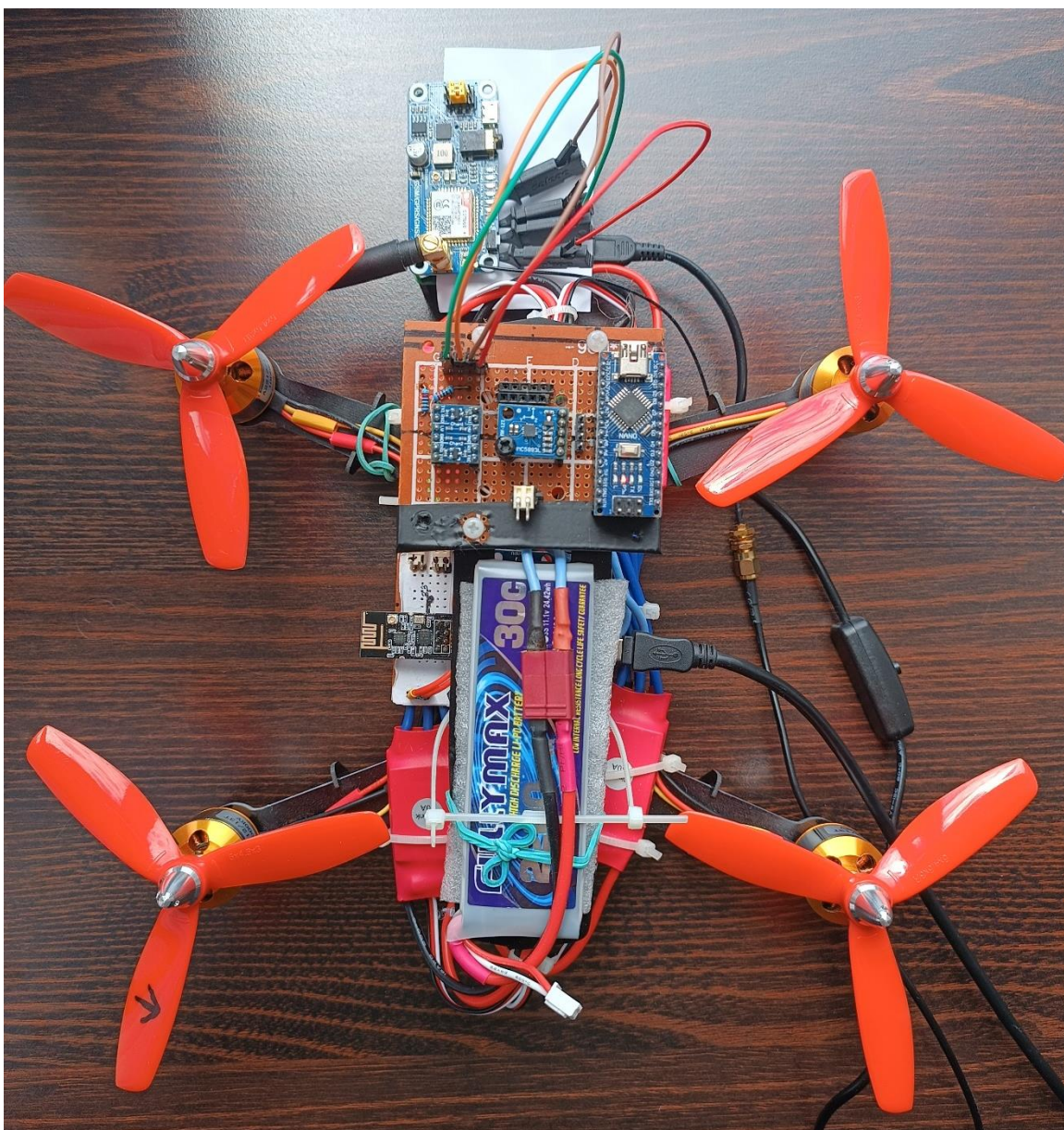


Рис. 1.4 – Прототип дрона

1.2 Raspberry Pi 3 Model A+

Raspberry Pi 3 Model A+ - це невеликий одноплатний комп'ютер, розроблений Фондом Raspberry Pi. В його основі лежить 64-бітний чотирьохядерний процесор, що працює на частоті 1,4 ГГц - Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz [1].

На рисунку 1.5 показаний зовнішній вигляд даного одноплатного комп'ютера.



Рис. 1.5 – Raspberry Pi 3 Model A+

1.2.1 Raspberry Pi як елемент навігаційної системи для дрона

У даній роботі Raspberry Pi Model A+ використовується для наступних цілей

- Передача поточних координат GPS через мобільний зв'язок (використовуючи GSM/GPRS/GPS модуль SIM868);
- Отримання координат пункту призначення (використовуючи GSM/GPRS/GPS модуль SIM868);
- Обчислення кутів roll та pitch необхідних для контролю польоту дрона на основі даних GPS та даних компаса QMC5883L;
- Обмін даними з Arduino Nano через інтерфейс I2C.

1.2.2 Операційна система Raspbian GNU/Linux 11 (bullseye)

На мікрокомп'ютер Raspberry Pi було встановлено операційну систему Raspbian GNU/Linux 11 (версії bullseye). У ній буде запускатися програма, написана на Python 3.9.2, що буде виконувати перелічені у пункті 1.2.1 цілі.

Raspbian GNU/Linux 11 (bullseye) є операційною системою, спеціально розробленою для Raspberry Pi, однієї з найпопулярніших одноплатних комп'ютерів [2]. Вона базується на відомій Linux-дистрибуції Debian, адаптованій до апаратної архітектури Raspberry Pi.

Операційна система Raspbian пропонує повноцінне середовище роботи на Raspberry Pi і включає в себе важливі компоненти, такі як ядро Linux, системні утиліти, графічне середовище, інструменти розробки та бібліотеки, необхідні для виконання різноманітних завдань на цій платформі.

1.3 Arduino® Nano

Arduino® Nano — це інтелектуальна плата розробки, призначена для створення швидших прототипів із найменшими розмірами. Arduino Nano, будучи найстарішим членом сімейства Nano, забезпечує достатньо інтерфейсів для програм. В основі плати лежить мікроконтролер ATmega328, що працює на частоті 16 МГц. Плата має 22 цифрових входу/виводу, 8 аналогових контактів і порт mini-USB. [3]

На рисунку 1.6 показаний зовнішній вигляд Arduino® Nano.

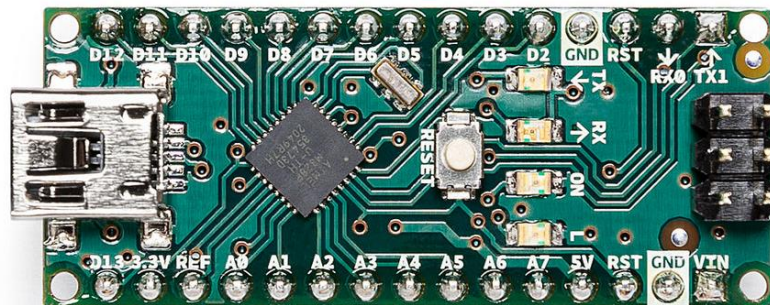


Рис. 1.6 – мікроконтролер Arduino® Nano

У даній дипломній роботі було використано два пристрої Arduino Nano — перший, як контролер радіоприймача та для безпосереднього контролю польоту дрона, а другий як контролер радіопередавача пульта управління.

1.3.1 Arduino Nano як контролер радіоприймача та для безпосереднього контролю польоту

У даній роботі Arduino Nano як радіоприймач та для безпосереднього керування дроном використаний у наступних цілях:

- Керування двигунами за допомогою ШІМ (англ. PWM) сигналу;
- Вчитка та обробка даних з гіроскопа та акселерометра для визначення положення дрона;
- Отримання азимуту з даних цифрового компасу;
- Отримання даних з радіомодуля через інтерфейс SPI;
- Обмін даними з Raspberry Pi через інтерфейс UART.

1.3.2 Arduino Nano як контролер радіопередавача пульта управління

У даній роботі Arduino Nano як контролер радіопередавача пульта управління використаний у наступних цілях:

- Комунікація з радіомодулем через інтерфейс SPI.
- Зчитування даних з джойстика (рисунок 1.7) (для керування рухом вперед-назад, вліво-вправо) та потенціометра (рисунок 1.8) (для керування загальної швидкості руху двигунів) через аналогові піни Arduino Nano.



Рис. 1.7 – Джойстик, використаний для керуванням напрямку дрона



Рис. 1.8 – Потенціометр, використаний для регулювання висоти дрона

1.3.3 Програмне середовище Arduino IDE

При розробці навігаційної системи на Arduino Nano було використано середовище розробки Arduino IDE

Arduino IDE є основним інструментом розробки програмного забезпечення для платформи Arduino. Він надає зручне інтерфейсне середовище для написання, компіляції та завантаження програм на плату Arduino.

Arduino IDE має широкий вибір вбудованих бібліотек, які дозволяють спростити взаємодію з різними пристроями та модулями. Ці бібліотеки містять готові функції та методи для роботи з сенсорами, комунікаціями, зберіганням даних та багатьма іншими компонентами.

Крім того, для тестування навігаційної системи в процесі її розробки особливо корисною стала можливість використання терміналу послідовного зв'язку (англ. Serial monitor). Це дозволяє відстежувати вивід програми та взаємодіяти з платою Arduino через послідовний порт (англ. Serial port)

1.4 Радіомодуль RF24L01

nRF24L01 – це однокристальний передавач радіосигналів, розроблений для бездротових додатків із наднизьким енергоспоживанням. nRF24L01 працює в діапазоні частот 2,400 - 2,4835 ГГц. Максимальний радіус передачі даних – 1100м.

Даний радіомодуль не потребує додаткових зовнішніх пасивних компонентів. Для створення радіосистеми на основі nRF24L01 достатньо лише буде з'єднання з мікроконтролером (у нашому випадку – Arduino Nano) через послідовний периферійний інтерфейс (SPI).[4]



Рис. 1.9 – радіомодуль nRF24L01 [5]

У даній дипломній роботі використано два nRF24L01-модулі:

1. На дроні радіомодуль сконфігурований у якості приймача сигналів з пульта управління.
2. На пульті управління радіомодуль використовується як передавач

Комунікація через радіозв'язок існує як резервний канал з'єднання та управління дроном за неможливості роботи GPS-автопілоту, чи GPRS-керування.

1.5 Акселерометр-гіроскоп MPU6050

На дроні також знаходиться акселерометр-гіроскоп MPU6050. Він зображений на рисунку 1.10.

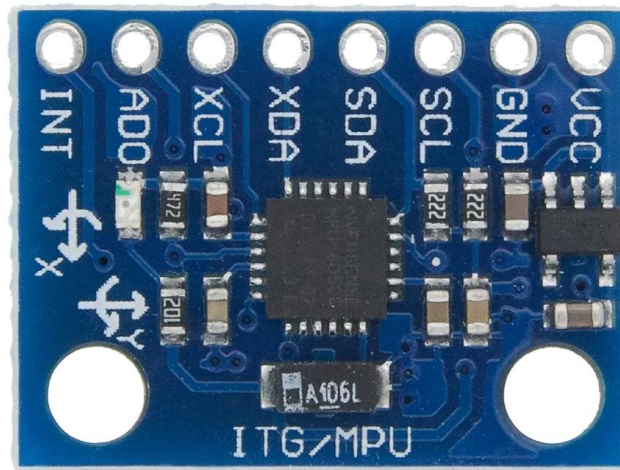


Рис. 1.10 – IMU модуль MPU6050

MPU6050 – інерціальний вимірювальний пристрій (Inertial Measurement Unit, IMU), який поєднує в собі акселерометр і гіроскоп. Цей пристрій розроблений фірмою InvenSense, і його широко використовують у багатьох проектах, пов'язаних зі збором даних про рух і орієнтацію. [6]

Функціональність MPU6050 дозволяє вимірювати прискорення по трьох осях (вперед-назад, вліво-вправо, вгору-вниз) за допомогою акселерометра і кутову швидкість обертання по трьох осях (крен, тангаж, риск) за допомогою гіроскопа. Окрім цього, пристрій має вбудований температурний датчик.

Характеристики MPU6050 включають:

- Точність: Акселерометр здатен вимірювати прискорення з точністю до ± 2 g, а гіроскоп – кутову швидкість до ± 250 °/с.
- Роздільна здатність: Акселерометр має 16-бітне розширення, що дозволяє вимірювати значення з високою точністю. Гіроскоп також має 16-бітне розширення.
- Інтерфейс: MPU6050 підтримує зв'язок з мікроконтролером за допомогою шини I2C (Inter-Integrated Circuit) з можливістю передачі даних з швидкістю до 400 кбіт/с.

У даній дипломній роботі він використаний у наступних цілях:

- Вимірювання кутів розвороту (roll) та нахилу (pitch) дрона.
- Відстеження відхилення від курсу (напрямку).

1.6 GSM/GPRS/GPS модуль SIM868

SIM868 є комбінованим модулем, розроблений компанією SIMCom, який поєднує в собі функціонал GSM/GPRS та GPS. SIM868 інтегрує протокол TCP/IP і розширені команди TCP/IP AT, які дуже корисні для програм передачі даних. [7]

- **GSM/GPRS:** SIM868 забезпечує підтримку мобільного зв'язку стандартів 2G (GSM) та 2.5G (GPRS). Це означає, що модуль може працювати зі стандартними мобільними операторами, що підтримують ці технології. Він може передавати і отримувати дані, включаючи голосові дзвінки, текстові повідомлення (SMS) та передачу даних через GPRS. Це дає можливість забезпечити зв'язок у віддалених місцях, де доступ до Інтернету обмежений або відсутній.
- **GPS:** SIM868 також включає GPS-приймач, що дозволяє отримувати інформацію про географічне положення. За допомогою супутникової навігації, модуль може визначати координати місця розташування з високою точністю. Це надає можливість використовувати SIM868 для геолокації, трекінгу, навігації та інших застосувань, пов'язаних з визначенням місцезнаходження. (точність 5-15 метрів)
- **Інтерфейси:** SIM868 має різноманітні інтерфейси для зв'язку зі зовнішніми пристроями. Він підтримує UART, USB і I2C для зв'язку з мікроконтролерами або іншими пристроями
- **Підтримка SIM-карт:** SIM868 має слот для вставки SIM-карти, що дозволяє забезпечити зв'язок з мережею оператора мобільного зв'язку. Це дає можливість модулю отримувати доступ до мобільної мережі для передачі даних, виконання дзвінків та обміну SMS.

- **Низьке споживання енергії:** SIM868 має енергозберігаючий режим, що дозволяє зменшити споживання енергії при неактивному використанні. Це особливо важливо для дрона, що працює від батарейного живлення.

Даний модуль зображений на рисунку 1.11.



Рис. 1.11 – GSM/GPRS/GPS модуль SIM868 [8]

1.6.1 GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi

GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi є спеціалізованою платою (HAT) розширення для Raspberry Pi. [9]

Тому ця плата була використана для можливості бездротового з'єднання з Raspberry Pi.

Цей модуль поставляється з адаптером живлення, кабелем мікро-USB, антеною GSM, GPS-антеною та антеною Bluetooth [10]. На рис. 1.12 зображено усі компоненти.



Рис. 1.12 – GSM/GPRS/GNSS/Bluetooth HAT

1.6.2 Модуль SIM868 як GPS-приймач та пристрій для зв'язку по GSM/GPRS

У даній дипломній роботі модуль SIM868 використовується у наступних цілях:

- Отримання поточних GPS координат: SIM868 використовує вбудований GNSS-приймач для визначення географічного положення. Це дозволяє отримувати точні GPS координати, які будуть використані для навігаційних цілей.
- Зв'язок з застосунком на комп'ютері через протокол TCP/IP, використовуючи мобільний зв'язок стандартів GSM/GPRS. Це забезпечує передачу таких даних, як:
 - Поточні GPS координати з Raspberry Pi на застосунок;
 - GPS координати пункту призначення з застосунку на Raspberry Pi;
 - Напрямок польоту при відсутності поточних даних GPS;
 - Швидкість моторів з застосунку на Raspberry Pi.

1.7 Магнетометр QMC5883L

Модуль QMC5883L (GY-273) - це електронний магнітний компас з 3-ма осями. [11]. Магнетометр QMC5883L є компактним датчиком, призначеним для вимірювання магнітного поля. Він забезпечує можливість визначення напрямку і сили магнітного поля в тривимірному просторі. Магнетометри такого типу широко використовуються в різних додатках, включаючи навігацію, розпізнавання положення та орієнтацію об'єктів, а також для досліджень магнітного середовища.

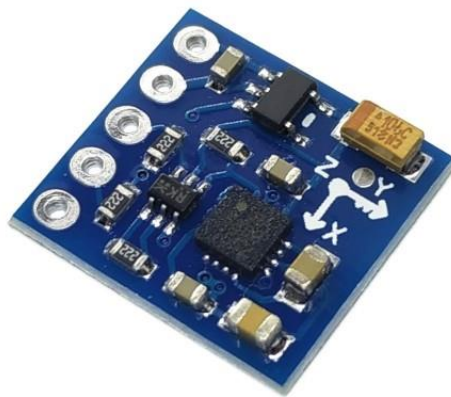


Рис. 1.13 – магнетометр QMC5883L

Основні функції та характеристики магнетометра QMC5883L:

- Вимірювання магнітного поля: QMC5883L може вимірювати магнітне поле у трьох основних напрямках (X, Y, Z). Він використовує технологію Hall-ефекту для точного визначення сили і напрямку магнітного поля.
- Висока точність: Цей магнетометр має високу точність вимірювання, що дозволяє отримувати достовірні дані про магнітне поле. Він здатний розрізняти навіть невеликі зміни магнітного поля і забезпечує стабільну роботу навіть в умовах шуму або впливу зовнішніх факторів, що дуже важливо при розробці дрона.

- Інтерфейс зв'язку: QMC5883L підтримує різні інтерфейси зв'язку, такі як I2C (Inter-Integrated Circuit) або SPI (Serial Peripheral Interface), що спрощує його підключення та взаємодію з мікроконтролером або іншими пристроями.
- Низьке споживання енергії: Цей магнетометр має низьке споживання енергії, що робить його ефективним для живлення від батарейки.

У даній дипломній роботі магнетометр QMC5883L використовується з метою отримання азимуту. Азимут - це горизонтальний кут відносно північного напрямку, який вказує на орієнтацію об'єкта в просторі. Цей кут застосовуватиметься для розробки алгоритму контролю польоту дрона на основі даних GPS.

1.8 Двигуни та контролери двигунів

Для того, аби дрон піднявся у повітря було використано чотири безколекторні двигуни A2212/13T-1000kv [12]. Безколекторний двигун – це вид двигуна без колектора і щіток, він використовує електроніку для передачі струму і використовується в дронах для точного і ефективного керування рухом.



Рис. 1.14 - безколекторний двигун A2212/13T-1000kv

Для керування цими двигунами був використаний програмований регулятор швидкості Simonk 30 A. Він є сумісним з безколекторними двигунами [13].



Рис. 1.15 - контролер швидкості Simonk 30 А

Керування ним відбувається за допомогою сигналу PWM. Регулятор швидкості отримує сигнал у вигляді ширино-імпульсної модуляції від радіокеровальної системи, який вказує необхідну швидкість обертання. Він інтерпретує цей сигнал і керує струмом, що подається на безколекторний двигун, для досягнення відповідної швидкості обертання.

РОЗДІЛ 2. Розробка системи навігації для дрона

2.1 Аналіз та огляд предметної області

2.1.1 Опис предметної області

Дрони (безпілотні літальні апарати) є необхідною складовою розвитку технологій та займають важливе місце у різних сферах, таких як військова, цивільна чи комерційна.

Система навігації дрона відіграє ключову роль у забезпеченні точності та стабільності його руху, визначенні його положення та орієнтації в просторі, забезпеченні надійного зв'язку з системою управління. Для досягнення даних цілей найчастіше використовують такі основні компоненти:

- Глобальна система позиціонування (GPS). GPS-приймач дозволяє визначати географічні координати дрона (широта, довгота, висота). Це важливо для навігації, відстеження маршруту та виконання завдань;
- Модулі зв'язку. Для забезпечення надійного зв'язку між БПЛА та зовнішнім середовищем необхідний якийсь модуль зв'язку. Використовуються різні модулі, такі як GSM/GPRS, Bluetooth, радіозв'язок та інші. Це дозволяє передавати дані, керувати дроном та забезпечувати зв'язок з оператором або системою управління. Надійний зв'язок є критично важливим для безпеки та успішності польоту дрона;
- IMU сенсори. Крім того потрібно забезпечити стабільність польоту дрона, для цього зазвичай використовують якийсь інерційний вимірювальний блок (IMU).

Крім вище переліченого сучасні тенденції розвитку дронів передбачають використання камер, щоб збирати додаткові дані про навколишнє середовище та отримувати зображення в режимі реального часу. Часто використовують штучний інтелект, для обробки зображень чи інших даних. Також використовуються різноманітні автоматизовані системи управління. Загалом дрони стають все кращими і кращими.

У даній ж дипломній роботі увага зосереджена на розробці прототипу навігаційної системи для дрона-камікадзе з забезпеченням кількох способів зв'язку.

2.1.2 Огляд існуючих рішень навігаційних систем для дронів

Перед тим, як розробляти навігаційну систему для дрона, необхідно проаналізувати уже існуючі рішення.

Декілька прикладів дронів:

- Phantom 4 Pro, Phantom 4 Advanced, Phantom 3 – це серія дронів від компанії DJI. Перевагами даних дронів є можливість знімати 4K відео та роботи якісні фото. Ці БПЛА мають вбудований модуль Wi-Fi, який дозволяє встановлювати зв'язок між дроном і пультом управління. Важливо мати на увазі, що зв'язок Wi-Fi має обмежену дальність дії, і дрони можуть летіти на відстань до 100 метрів від контролера, що є їх недоліком. [14]



Рис. 2.1 – дрон Phantom 4 Pro

- DJI Tello є компактним безпілотним літальним апаратом, розробленим спільно зі швейцарською компанією Ryze Tech. Він не має власної камери, проте він має низьку вагу, можливість польоту за траєкторією та повернення назад. Крім того у ньому присутня технологія аналізу зіткнень. Зв'язок з дроном відбувається на основі технології Wi-Fi. [15]



Рис. 2.2 – дрон DJI Tello

- ІТЕС SKIF – дрон вироблений Київською компанією ІТЕС. Це БПЛА з крилом типу «Промінь», працює від електричного двигуна та запускається за допомогою еластичної катапульти. Він здатний робити фотографії та має у собі радіомодуль для зв'язку з комп'ютером. Вагомим плюсом даної моделі є повна автономність роботи. Оператор налаштовує маршрут польоту, запускає дрон і далі лише здійснює контроль за польотом. [16]



Рис. 2.3 – дрон ІТЕС SKIF

Незважаючи на недоліки, такі як відсутність камери чи менша точність у порівнянні з вищезгаданими варіантами, розроблена у даній роботі навігаційна система має ряд переваг:

- Наявність резервних методів зв'язку;
- Використання мобільного зв'язку, що дозволяє комунікувати з дроном з будь-якої точки;
- Наявність радіозв'язку з радіусом комунікації до 1100м;
- Здешевлення виробництва.

2.1.3 Огляд використаних технологій та інтерфейсів

Загальний огляд використаних технологій та інтерфейсів наведений нижче:

- **Радіозв'язок**

Це різновид зв'язку, у якому носієм інформації є радіохвилі.

Частотна сітка, що використовується у радіозв'язку, поділяється на діапазони:

- Довгі хвилі (ДХ) — $f = 150 - 450$ кГц ($\lambda = 2000-670$ м)
- Середні хвилі (СХ) — $f = 500 - 1600$ кГц ($\lambda = 600-190$ м)
- Короткі хвилі (КХ) — $f = 3-30$ МГц ($\lambda = 100-10$ м)
- Ультракороткі хвилі (УКХ) — $f = 30$ МГц - 300 МГц ($\lambda = 10-1$ м) [17]

Основні переваги використання радіозв'язку – велика пропускну здатність, бездротова комунікація на великі відстані.

Основні недоліки використання радіозв'язку – можливість втрати сигналу через фізичні перешкоди, сигнал може бути перехоплений.

- **GSM/GPRS**

GSM (Global System for Mobile Communications) і GPRS (General Packet Radio Service) є двома технологіями бездротового зв'язку, які використовуються в пристроях з підтримкою мобільного зв'язку. [18]

GSM - призначений для голосових послуг і доставки даних за допомогою цифрової модуляції. Це стандартний носій технологій 2G.

Основне використання GSM – це голосовий зв'язок. Також він використовується для надсилання та отримання коротких текстових повідомлень (SMS), що є популярною формою спілкування.

GPRS — це модернізація функцій GSM порівняно з основними функціями для отримання значно вищої швидкості передачі даних і простого бездротового доступу до мереж пакетної передачі даних, ніж стандартний GSM. Це стандартний носій технологій 2.5G. GPRS в основному використовується для отримання доступу до Інтернету.

- **GNSS (GPS)**

GNSS (Global navigation satellite system) це загальний термін, що описує будь-яку супутникову групу, яка надає послуги позиціонування на глобальній чи регіональній основі. [19]

GPS є найпоширенішою GNSS. Система GPS використовує мережу супутників, які надсилають сигнали до приймачів на землі. Ці сигнали містять інформацію про положення супутників та точний час відліку, на основі якого приймачі визначають своє місцезнаходження.

- **Протокол TCP/IP**

TCP/IP (Transmission Control Protocol/Internet Protocol) – це набір протоколів, які використовуються для забезпечення комунікації у комп'ютерних мережах.

Принцип роботи TCP/IP базується на розподіленому засобі передачі даних у мережі. Він використовує два основні протоколи - TCP і IP. [20]

Протокол IP відповідає за передачу даних через мережу. Він розбиває дані на пакети і призначає кожному з них IP-адресу відправника та отримувача. IP-адреси – це унікальні ідентифікатори пристроїв у мережі. Пакети маршрутизуються від відправника до отримувача через різні вузли мережі за допомогою маршрутизаторів.

Протокол TCP забезпечує надійну доставку даних. Він розбиває дані на сегменти, нумерує їх та використовує механізми підтвердження та повторної передачі, щоб переконатися, що дані доставлені до отримувача.

- **UART**

UART – універсальний асинхронний прийомо-передавач. “Асинхронний” означає, що інтерфейс не використовує лінію для синхросигналу, а приймач і передавач “заздалегідь домовляються” про частоту передачі даних (тобто, налаштовуються на однакову частоту).

Для передачі та прийому даних UART використовує дві лінії: TX (TXD) і RX (RXD). TX – лінія, що передає, а RX – лінія, що приймає дані.

Дані у UART передаються по одному біту в рівні проміжки часу, які визначаються заданою швидкістю UART. Швидкість задається в бодах.[21]

- **SPI**

SPI (Serial Peripheral Interface – шина для підключення периферійних пристроїв) – послідовний синхронний повнодуплексний інтерфейс.

Побудований за принципом Master/Slave. “Ініціатором” обміну даними за SPI завжди є ведучий пристрій (Master), ведений пристрій (Slave) не може повідомляти ведучому нічого про себе, зокрема, про прийнятну швидкість передачі даних, тому ведучий пристрій повинен сам знати, як правильно “спілкуватися” з веденим пристроєм.

Типово для спілкування ведучого пристрою з веденим застосовуються 4 лінії: SCLK, MOSI, MISO та SS. [21]

- **I2C**

I2C (Inter-Integrated Circuit, IIC) – послідовний синхронний інтерфейс, розроблений компанією Philips. Побудований за принципом Master/Slave.

У I2C для передавання інформації, незалежно від кількості ведених пристроїв, застосовується усього дві лінії – DATA (SDA, Serial Data) і CLOCK (SCK, Serial Clock).

Схематично роботу можна зобразити так:

1. Ведучий пристрій формує СТРАТ-умову
 2. Ведучий пристрій вибирає адресата і напрям передачі даних
 3. Відбувається власне передача даних (пристрій, що приймає, має підтвердити успішний прийом)
 4. Ведучий пристрій формує СТОП-умову для завершення передачі. [21]
- **PWM сигнали**

Широтно-імпульсна модуляція (PWM) - модуляція за тривалістю імпульсів. Це процес керування шириною (тривалістю) високочастотних імпульсів за законом, який задає низькочастотний сигнал. [22]

Використовуючи PWM, можна ефективно керувати потужністю або яскравістю сигналу, змінюючи співвідношення тривалості "ввімкненого" (1) і "вимкненого" (0) станів сигналу.

В залежності від тривалості "ввімкненого" стану відносно загальної періоди, отримується аналоговий сигнал.

- **Аналогові сигнали**

Аналоговий сигнал - це сигнал, який змінюється впродовж часу і може приймати будь-яке значення у певному діапазоні.

Мікроконтролер сприймає цифрові коди, а навколишній світ – аналоговий (неперервний). Для забезпечення взаємодії МК з рядом пристроїв застосовується АЦП – аналого-цифровий перетворювач (ADC, Analog-to-digit converter). [21]

2.2 Постановка завдання

Метою даної дипломної роботи є розробити прототип навігаційної системи для дрона-камікадзе для польоту на недалекі відстані з трьома методами зв'язку.

Для реалізації цієї мети було сформовано такі завдання:

На контролері польоту Arduino Nano:

1. Забезпечити отримання даних IMU сенсора, даних з модуля радіоз'язку, компаса та Raspberry Pi;
2. Обрахувати поточний нахил дрона на основі даних з IMU сенсора;
3. Забезпечити точність і плавність польоту, використовуючи PID контролер;
На пульті управління:
4. Забезпечити отримання даних з джойстика і потенціометра;
5. Забезпечити відправку даних через модуль радіоз'язку;
На Raspberry Pi:
6. Налаштувати мобільний зв'язок за допомогою модуля SIM868;
7. Налаштувати отримання даних з GPS антени за допомогою модуля SIM868;
8. Розробити алгоритм обчислення кутів roll та pitch на основі поточних координат, координат пункту призначення і даних з компаса;
9. Налаштувати зв'язок по TCP/IP з застосунком;
10. Налаштувати зв'язок з Arduino Nano та придумати протокол комунікації між ними;
На застосунку:
11. Налаштувати застосунок, як сервер та налаштувати комунікацію по TCP/IP з дроном ;
12. Забезпечити відображення та постійне оновлення поточних GPS координат.
13. Дозволити вибрати координати пункту призначення та швидкість моторів.
14. Забезпечити автоматичне переключення на мобільний зв'язок (без GPS) та можливість керування за допомогою вказання напрямку у випадку втрати даних на GPS. Повідомити користувача, що слід перейти на радіозв'язок у випадку втрати мобільного зв'язку.
І останнім завданням є:
15. Тестування та валідація усієї розробленої навігаційної системи.

2.3 Функціональна схема

На рисунку 2.4 зображена функціональна схема, яка використовується для реалізації та тестування навігаційної системи. На схемі позначено різні компоненти, які використовуються як на дроні, так і на пульті управління. Елементи, розташовані безпосередньо на дроні, позначені зеленим квадратом, тоді як елементи на пульті управління позначені фіолетовим квадратом.

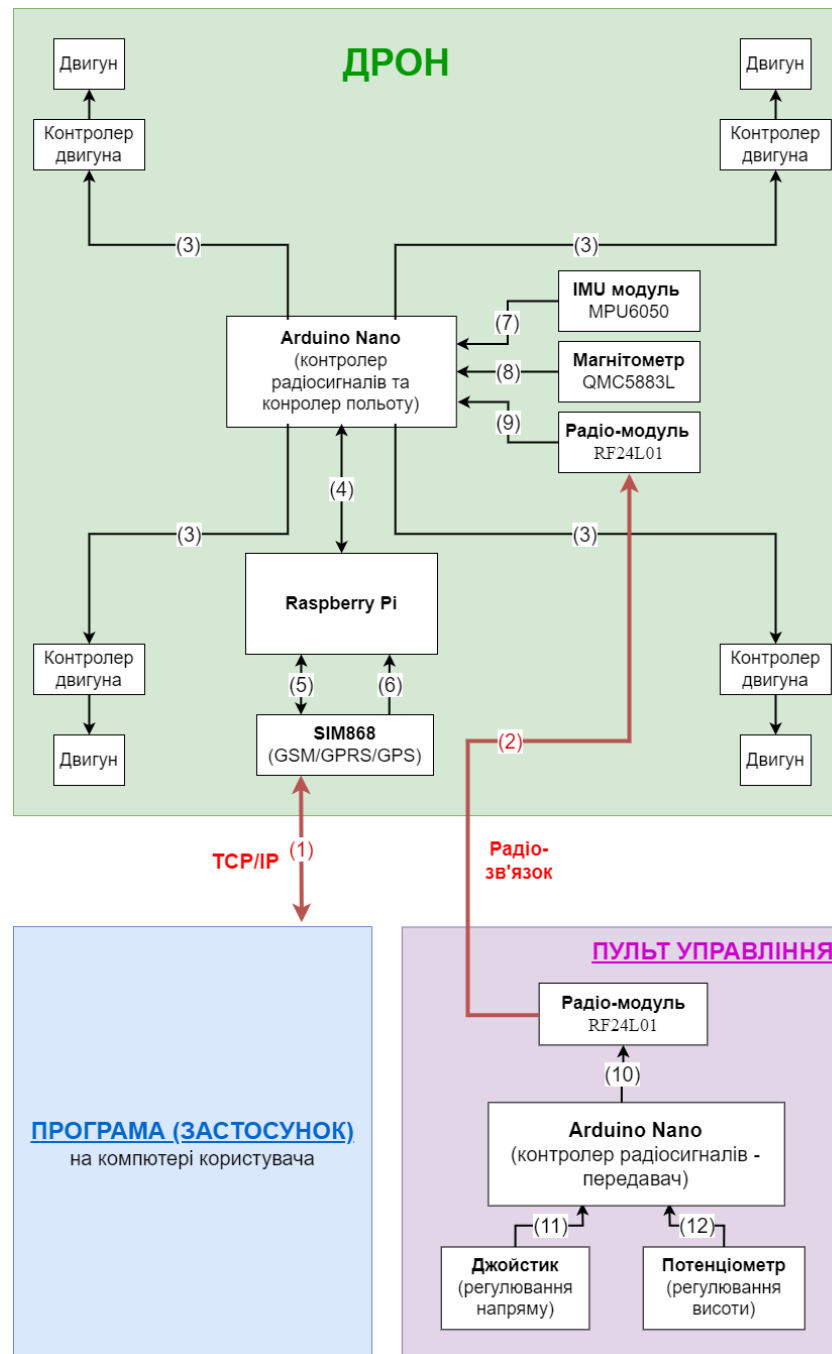


Рис. 2.4 – функціональна схема з розміщенням компонентів і їх зв'язками

Зв'язки між компонентами на рисунку 2.4 позначені стрілками з цифровою нумерацією. Пояснення до кожного методу зв'язку:

(1) – Зв'язок між дроном і програмою на комп'ютері користувача по TCP/IP.

Дані, які пересилаються між ними:

Модуль мобільного зв'язку на дроні → застосунок:

- поточні координати GPS.

Застосунок → модуль мобільного зв'язку на дроні:

- GPS координати пункту призначення;
- Швидкість моторів (висота);
- Напрямок руху дрона (у випадку недоступності GPS).

(2) – Зв'язок між дроном і пультом управління по радіозв'язку (альтернативний спосіб зв'язку)

Дані, які пересилаються між ними:

Радіомодуль пульта управління → радіомодуль на дроні:

- Швидкість моторів (дані з потенціометра);
- Напрямок руху дрона (дані з джойстика).

(3) - Передача розрахованої швидкості мотора з Arduino Nano на контролер двигуна, використовуючи PWM сигнали.

(4)– Передача даних між Raspberry Pi (мінікомп'ютер на якому здійснюється розрахунок напрямку польоту) та Arduino Nano (де здійснюється розрахунок швидкості моторів на основі даних з сенсорів та даних від Raspberry Pi)

Аби забезпечити між ними зв'язок було використано додаткове Arduino Nano, що приймає дані від Raspberry Pi по інтерфейсу комунікації I2C, передає їх на основний мікроконтролер Arduino Nano по UART, чекає на відповідь від нього та повертає дані по I2C на Raspberry Pi.

Дані, які пересилаються між ними:

Raspberry Pi → Arduino Nano:

- Запит на отримання даних компаса;

- Обраховані кути roll та pitch (або ж вказані, як напрямок у застосунку);
- Швидкість моторів (та, що була вказана у застосунку).

Arduino Nano → Raspberry Pi:

- Дані компаса;
- Відповідь-підтвердження отримання даних або помилка.

- (5)– Пересилання і отримання даних по мобільному зв'язку, використовуючи модуль SIM868 та інтерфейс зв'язку з ним - UART. Дані, що пересилаються по мобільному зв'язку, перелічені у пункті (1).
- (6)– Отримання поточних GPS координат на Raspberry Pi за допомогою інтерфейсу UART, використовуючи модуль SIM868 та GPS антену.
- (7)– Отримання даних про нахил дрону з гіроскопа-акселерометра за допомогою інтерфейсу комунікації I2C.
- (8)– Отримання даних про поточний азимут з магнітометра (компаса) за допомогою інтерфейсу комунікації I2C.
- (9) – Отримання радіосигналів з інформацією з пульта управління за допомогою інтерфейсу комунікації SPI. Дані, що пересилаються по радіозв'язку перелічені у пункті (2).

Зв'язки між компонентами на пульті управління:

- (10) – Передача даних з джойстика і потенціометра на радіомодуль за допомогою інтерфейсу комунікації SPI. Дані, що пересилаються по радіозв'язку, перелічені у пункті (2).
- (11) – Отримання даних на Arduino Nano з джойстика про регулювання напрямку польоту. Дані отримуються, як аналогові сигнали.
- (12) – Отримання даних на Arduino Nano з потенціометра про регулювання висоти польоту. Дані отримуються, як аналогові сигнали.

2.4 Розробка програмного забезпечення для застосунку

Основним методом зв'язку з дроном є мобільний зв'язок. Для реалізації зручного способу комунікації був розроблений комп'ютерний застосунок на мові програмування Python, використовуючи такі бібліотеки як:

- Tkinter - багатоплатформна графічна бібліотека інтерфейсів [23]. Tkinter є стандартною бібліотекою інтерфейсу користувача (UI) для мови програмування Python. Вона надає набір інструментів і об'єктів для створення графічних інтерфейсів, зокрема вікон, кнопок, полів введення, списків, меню та інших елементів.
- Socket - мережевий інтерфейс низького рівня [24]. У Python бібліотека socket забезпечує низькорівневий доступ до сокетів, дозволяючи програмістам створювати, встановлювати та керувати з'єднаннями між комп'ютерами. Вона підтримує як TCP, так і UDP протоколи передачі даних.
- Threading - паралелізм на основі потоків [25]. Бібліотека threading у Python надає засоби для роботи з потоками в програмі. Вона дозволяє створювати та керувати багатьма потоками в одній програмі, що дозволяє виконувати кілька операцій одночасно. Потоки (threads) - це окремі шляхи виконання у програмі, які можуть працювати паралельно.
- TkinterMapView – це бібліотека для відображення інтерактивної карти у графічному застосунку, що використовує tkinter. Вона не є стандартною бібліотекою – її код доступний за посиланням [26].

Даний застосунок містить у собі термінал, на якому відображаються дані, що приходять від дрона (дані про поточні GPS координати). Крім того, поточна геолокація постійно оновлюється на карті.

Дана програма також реалізує алгоритм переключення на різні засоби зв'язку. І у випадку втрати GPS координат дозволяє використовувати ручне регулювання напрямку польоту.

2.4.1 Алгоритм роботи системи

На рисунку 2.5 зображена блок-схема роботи даного застосунку.

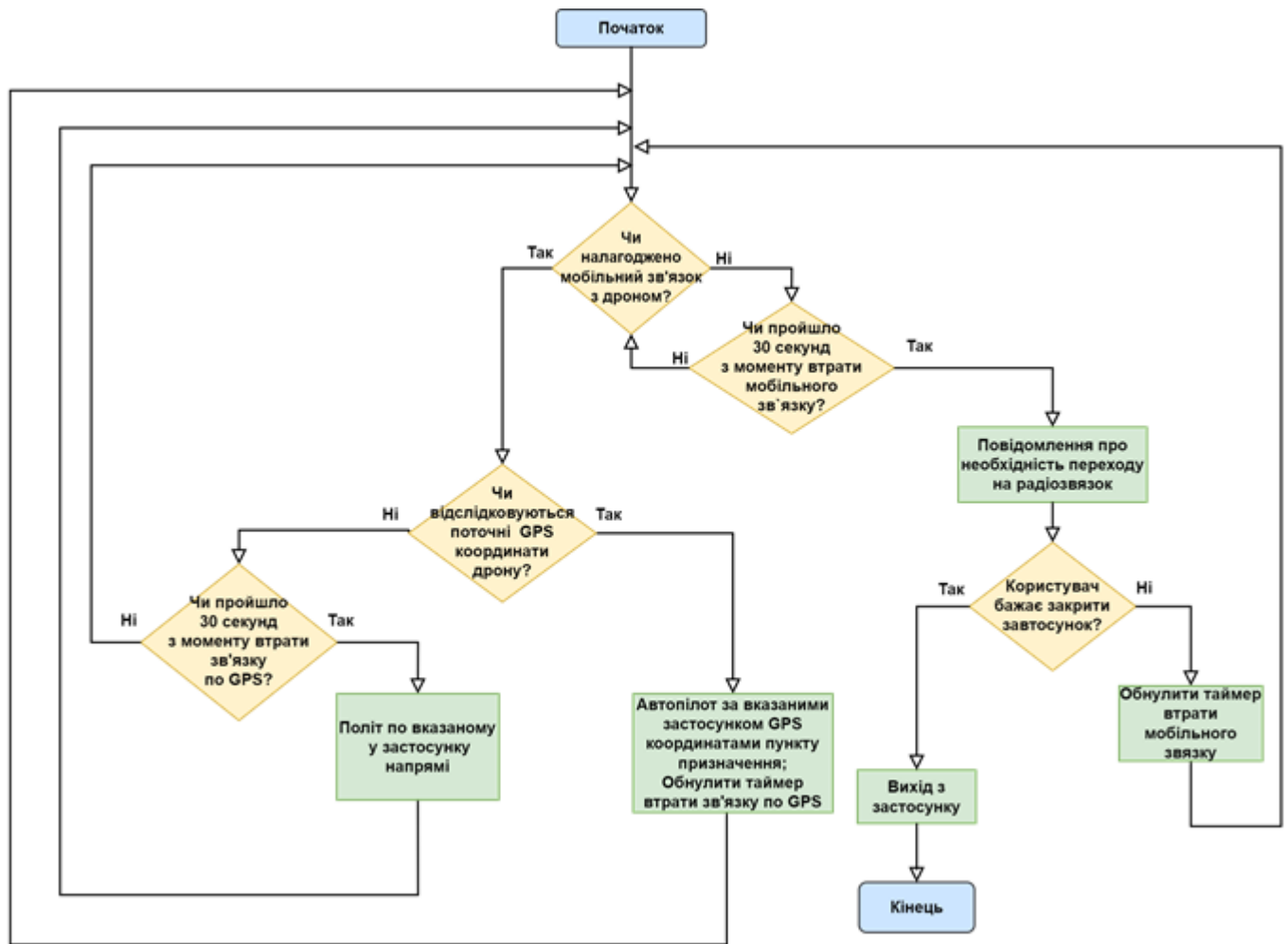


Рис. 2.5 – блок-схема роботи GUI застосунку

2.4.2 Інтерфейс програми

Відкривши даний застосунок, перше, що побачить користувач це буде термінал, карта, повзунки напряму руху (недоступні в режимі автоматичного польоту за GPS) та повзунок зміни швидкості моторів (рисунок 2.6). Тепер користувачу слід ввімкнути дрон та очікувати на з'єднання по мобільній мережі.

Натиснувши правою клавішею мишки на якесь місце на карті, можна позначити GPS координати пункту призначення (рисунок 2.8). Після цього вони автоматично відправляються на дрон. Ці координати можна змінювати за потреби таким самим чином.

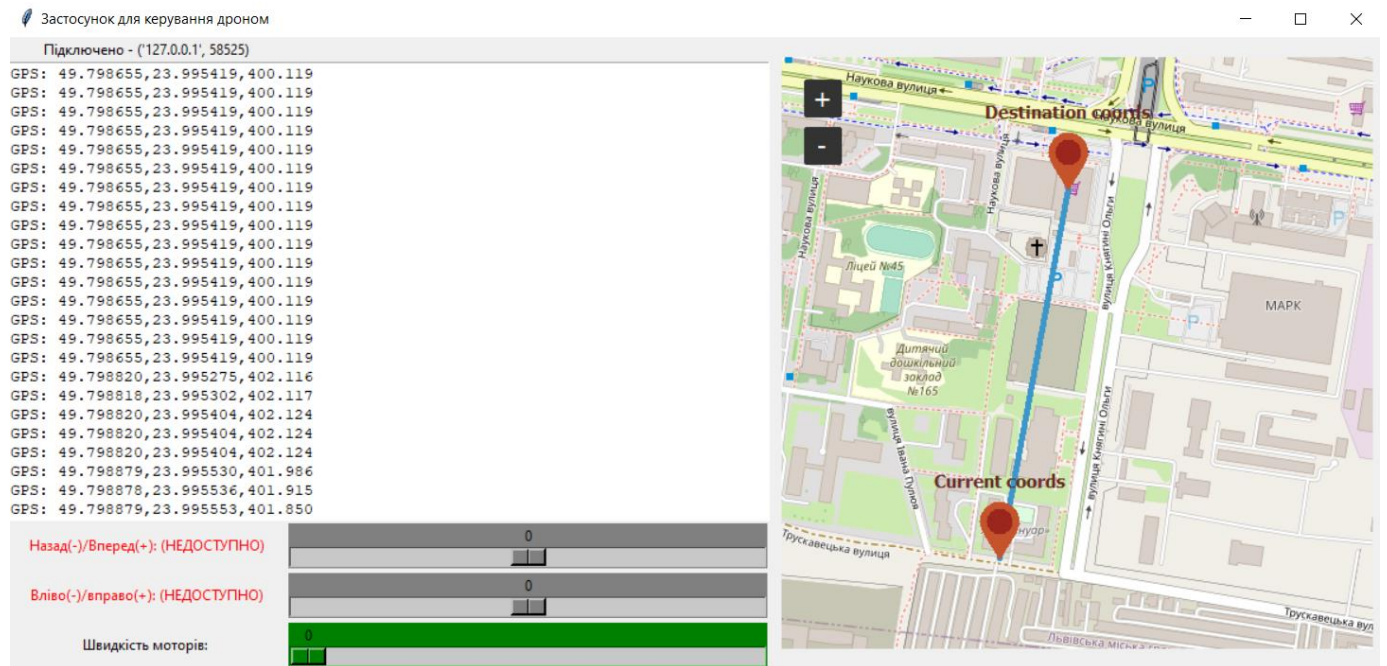


Рис. 2.8 – Можливість вибрати координати пункту призначення та динамічна побудова шляху від поточних координат

У випадку, якщо було втрачено можливість відслідковування GPS протягом 30 секунд, застосунок автоматично перейде на контроль польоту за допомогою лише мобільного зв'язку та регулювання напряму польоту дрону за допомогою двох повзунків (рисунок 2.9) у випадку, якщо користувач хоче змінити поточний напрям польоту.

Якщо GPS зв'язок у якийсь момент відновиться – система знову перейде до контролю польоту за GPS.

2.5 Розробка програмного забезпечення для Raspberry Pi

На дроні для забезпечення мобільного зв'язку, обробки даних та знаходження напрямку польоту було використано одноплатний комп'ютер Raspberry Pi. На ньому було встановлено операційну систему Raspbian GNU/Linux 11, а код написаний на мові програмування Python з використанням таких бібліотек:

- Pyserial – бібліотека, що надає засоби для роботи з послідовним (серійним) портом [27]. Для даної програми ця бібліотека використана, щоб здійснювати зчитування та запис даних через послідовний порт UART (а саме для спілкування з SIM868);
- Smbus – бібліотека у мові програмування Python, що надає засоби для взаємодії з пристроями через шину I2C (Inter-Integrated Circuit). Для даної програми вона використовується для з'єднання через I2C з додатковим Arduino Nano;
- Math - вбудована бібліотека у Python, яка надає функції та константи для математичних операцій і обчислень. За допомогою цього модуля виконуються обчислення тригонометричних функцій;
- NumPy (Numeric Python) є потужним інструментом для роботи з числовими даними у мові програмування Python. З неї було використано функції `degrees()`, що перетворює радіани на градуси, і функцію `arctan2()`, що обчислює арктангенс двох аргументів (y, x) і повертає відповідне значення в радіанах;
- Threading – описаний у пункті 2.4 і використовується для паралельного обміну інформацією з застосунком і з Arduino Nano.

2.5.1 Блок-схеми

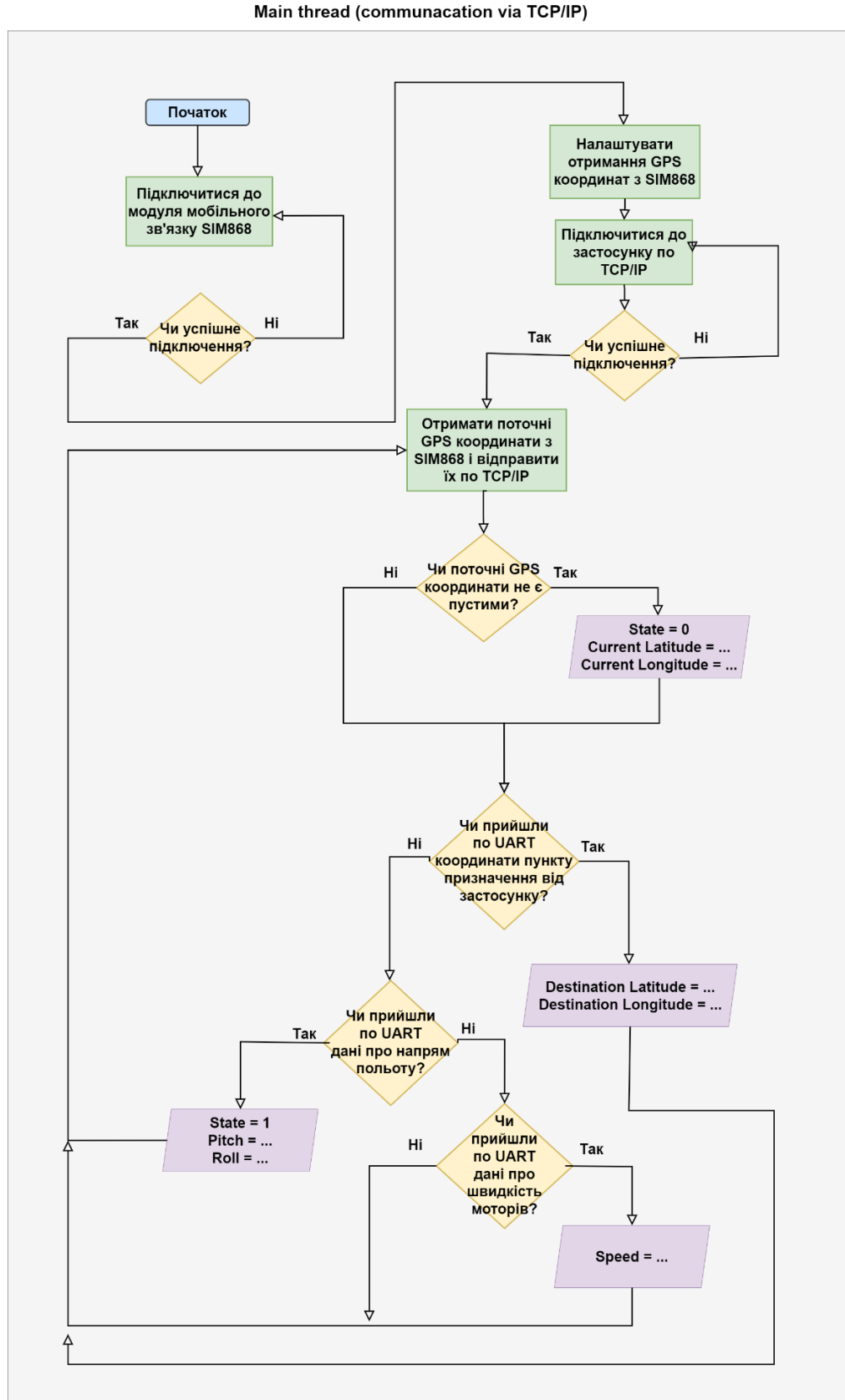


Рис. 2.11 – блок-схема головного потоку програми на Raspberry Pi

Second thread (communication with Arduino Nano via I2C)

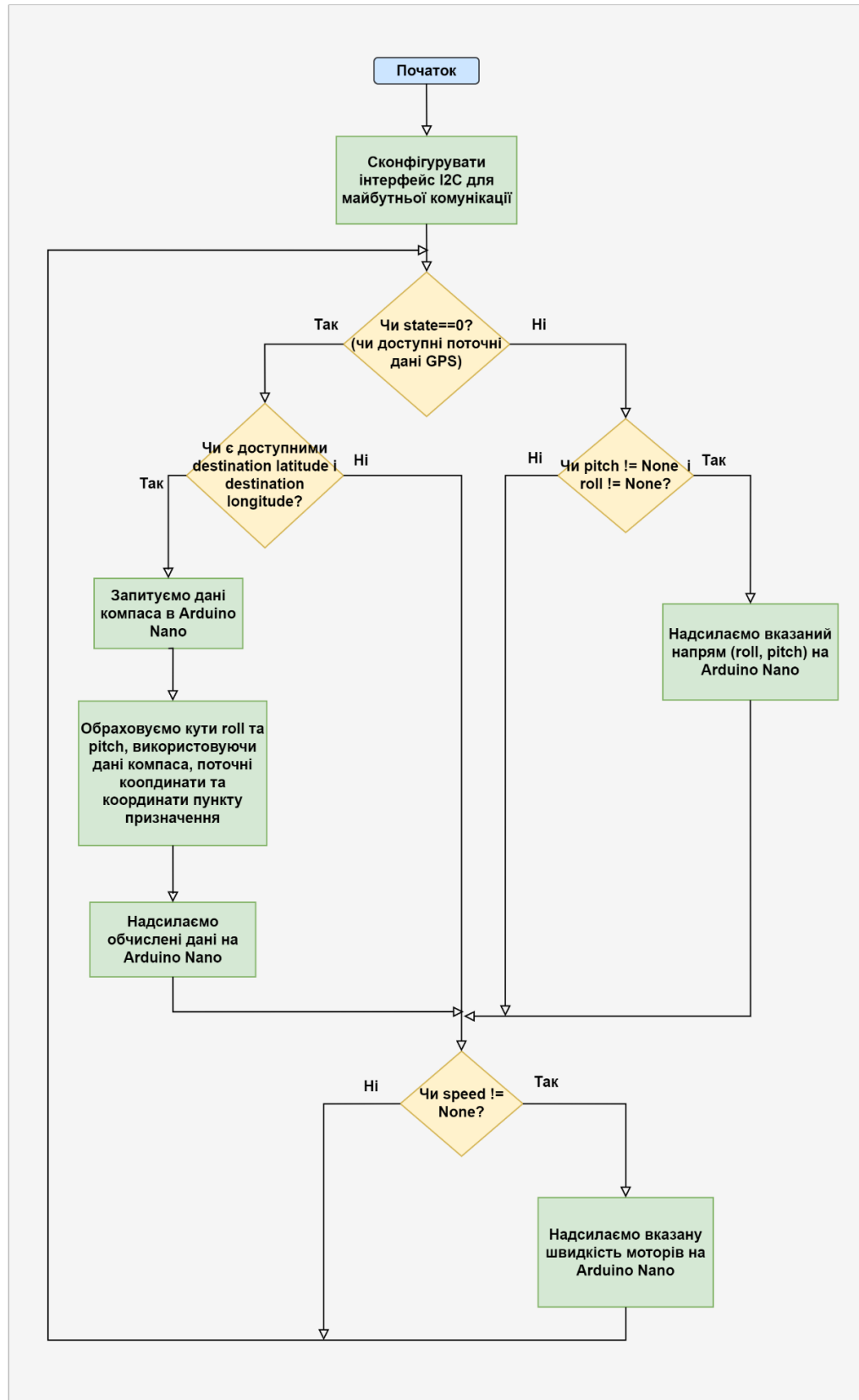


Рис. 2.12 – Блок схема потоку для комунікації з Arduino Nano та обчислення кутів roll та pitch на Raspberry Pi

2.5.2 Формула гаверсинуса

У даній дипломній роботі для знаходження бажаного азимуту між двома географічними точками була використана формула гаверсинуса для пошуку пеленгу (англ. bearing).

Азимут – це кутовий вимір у сферичній системі координат. Відлік значення азимута починається від північного напрямку й ведеться за годинниковою стрілкою. (від 0° до 360°). [28]

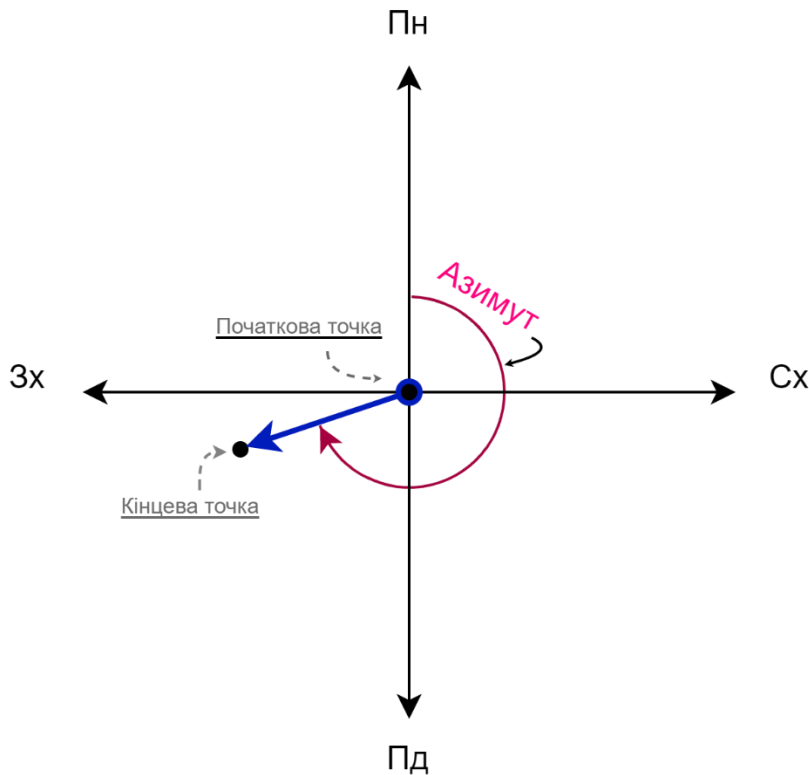


Рис. 2.13 – Кут азимуту рахується від півночі за годинниковою стрілкою

Формула гаверсинуса заснована на теоремі сферичних трикутників і дає можливість обчислити азимут (а також відстань) між двома географічними точками, використовуючи їх координати широти (latitude) і довготи (longitude). Формула для пеленгу (прямого азимуту) виглядає наступним чином: [29]

$$\Theta = \text{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda)$$

Де:

- θ : прямий азимут між двома точками;
- atan2 : функція арктангенс з двома аргументами, яка визначає значення кута між віссю x і точкою, яка має координати, вказані як її аргументи;
- $\Delta\lambda$: різниця між довготами двох точок;
- φ_1 : широта початкової точки;
- φ_2 : широта кінцевої точки.

2.5.2 Алгоритм навігації за GPS

Для навігації окрім бажаного азимуту, обчисленого за допомогою формули гаверсінуса, необхідно знати поточний азимут і змінити напрям польоту дрона, правильно порахувавши нові кути крену та тантажу (кути roll та pitch).

Поточний азимут запитується у Arduino Nano, який має під'єднаний до себе компас. Мікроконтролер зчитує ці дані і відсилає їх на Raspberry Pi.

Для того, аби дрон летів у певну сторону – необхідно обчислити кути крену та тантажу.

- Кут крену (англ. roll) використовується для визначення обертання об'єкта повздовжньої осі, що проходить через нього в горизонтальній площині (див. рисунок 2.14). Це означає, що рух по осі roll виконується, коли об'єкт повертається вправо або вліво.

У випадку руху вправо – кут крену буде додатнім значенням.

У випадку руху вліво – від'ємним.

- Кут тантажу (pitch) використовується для визначення нахилу об'єкта вздовж поперечної осі, що проходить через нього в горизонтальній площині (див. рисунок 2.14). Рух по осі pitch відбувається, коли об'єкт нахиляється вперед або назад.

У випадку руху вперед – кут тантажу буде додатнім значенням.

У випадку руху назад – від'ємним.

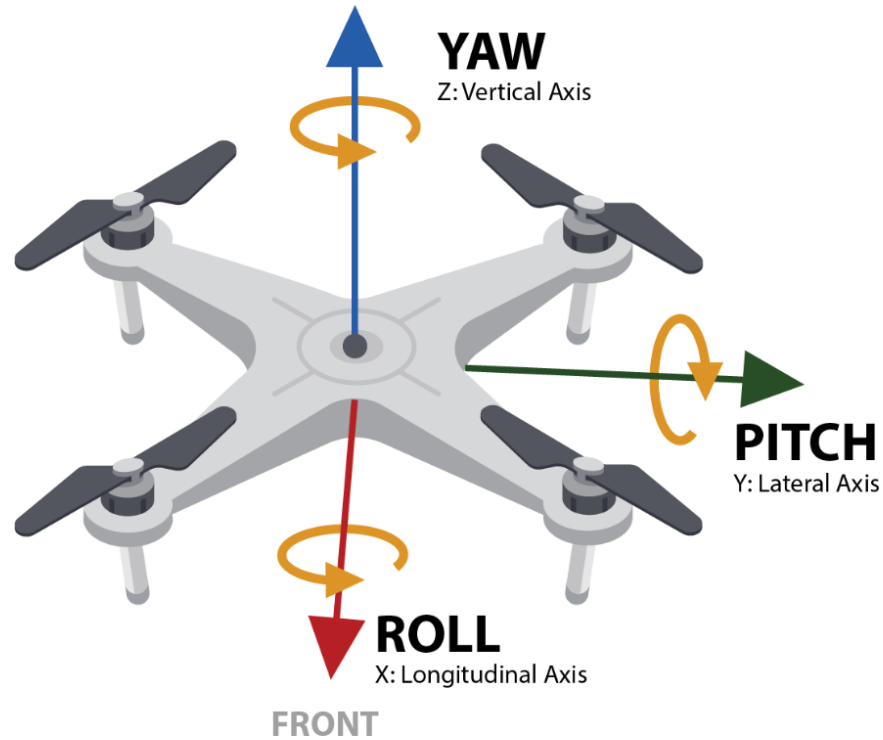


Рис. 2.14 – Кути roll, pitch і yaw у дроні [30]

Знаючи бажаний і поточний азимут можна порахувати швидкості roll і pitch за формулами:

$$\text{pitch} = \cos(\alpha_2 - \alpha_1) \cdot \text{max_tilt}$$

$$\text{roll} = \sin(\alpha_2 - \alpha_1) \cdot \text{max_tilt}$$

Де:

- α_1 – поточний азимут;
- α_2 – бажаний азимут;
- max_tilt – максимальне значення нахилу дрона;

Обґрунтування формул:

На рисунку 2.15 зображено дрон, синьою стрілкою позначено поточний азимут дрона (те, що повертає компас), червоною – бажаний напрямок руху. Кут між ними позначено як cda (course deviation angle).

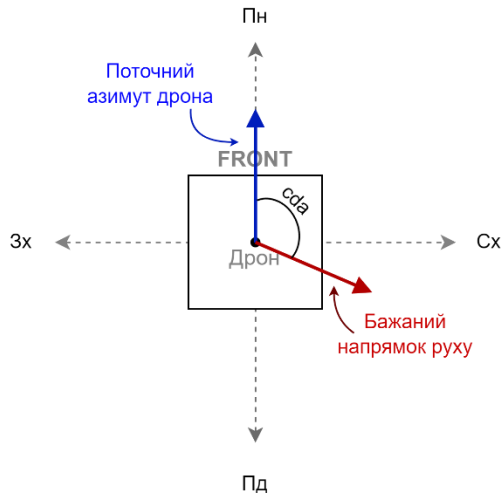


Рис. 2.15 – Поточний азимут дрона і бажаний напрямок руху

Результуючий напрямок руху дрона складається зі значення відхилення по поперечній осі та по повздовжній осі (pitch та roll). Звідси одержуємо прямокутний трикутник (рисунок 2.16), з якого, знаючи різницю між поточним і напрямом польоту (кут позначений як cda), можна порахувати roll та pitch за формулами для прямокутного трикутника.

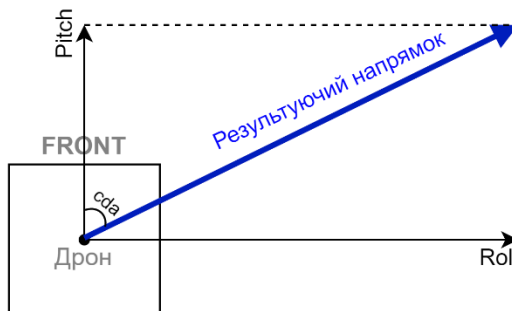


Рис. 2.16 – Результуючий напрямок дрона, що складається зі значень відхилень roll та pitch

Код для обчислення поточного азимуту і швидкостей roll і pitch:

```
def angles_calculate(current_lat, current_lon, destination_lat, destination_lon,
compass_data):
    # різниця між довготами двох точок
    dLon = (destination_lon - current_lon)
    # перший параметр для arctan2
    x = math.cos(math.radians(destination_lat)) * math.sin(math.radians(dLon))
    # другий параметр для arctan2
    y = math.cos(math.radians(current_lat)) * math.sin(math.radians(destination_lat)) -
    math.sin(math.radians(current_lat)) * math.cos(math.radians(destination_lat)) *
    math.cos(math.radians(dLon))
    # азимут (bearing)
    brng = numpy.arctan2(x,y)
    brng = numpy.degrees(brng) # переводимо в градуси
    brng = (brng + 360) % 360 # переводимо в проміжок 0-360 градусів

    # рахуємо відхилення від курсу (course deviation angle)
    cda = brng - compass_data
    cda_rad = math.radians(cda) # переводимо в радіани
    pitch = int(math.cos(cda_rad) * 1000) # pitch = cos(cda) * max_pitch
    roll = int(math.sin(cda_rad) * 1000) # roll = sin(cda) * max_roll

    return roll, pitch
```

2.6 Розробка програмного забезпечення для Arduino

Для безпосереднього контролю польоту, регулювання та контролювання швидкості моторів на дроні, вирівнювання дрона, зчитування даних з таких датчиків як IMU модуль та компас, отримання даних з радіомодуля використовується мікроконтролер Arduino Nano. Код програми був написаний на мові програмування C та прошитий на контролер за допомогою Arduino IDE. Крім того, були використані такі бібліотеки, як:

- Wire – бібліотека, що дозволяє спілкуватися з пристроями, використовуючи I2C/TWI [31].

У даній програмі ця бібліотека використовується для отримання даних з IMU сенсора по інтерфейсу I2C.

- RF24h – бібліотека для Arduino, що дозволяє взаємодіяти з бездротовим модулем передачі даних nRF24L01. Ця бібліотека надає зручний інтерфейс для налаштування і керування радіомодулем, що дозволяє передавати та отримувати дані бездротовим способом з інших пристроїв, які також використовують модуль nRF24L01. [32]

У даній програмі ця бібліотека використовується, як драйвер для радіомодуля, щоб налагодити комунікацію між пультом управління та дроном по радіозв'язку.

- Servo – бібліотека для Arduino, що використовується для контролю безколекторними двигунами. Servo двигуни мають три дроти: живлення, земля та сигнал. Бібліотека надає API для зручного контролю швидкості моторів. [33]
- QMC5883LCompass – це бібліотека Arduino для використання мікросхем серії QMC5583L як компаса [34].

Ця бібліотека підтримує обчислення азимуту, який потім у програмі відсилатиметься на Raspberry Pi.

- SoftwareSerial – бібліотека, що забезпечує послідовний інтерфейс на різних цифрових контактах мікроконтролера Arduino, використовуючи програмне забезпечення моделювання функціональності.

У даній програмі ця бібліотека використовується для передачі даних через UART на додаткове Arduino Nano, яке потім пересилатиме дані на Raspberry Pi.

2.6.1 Блок-схема

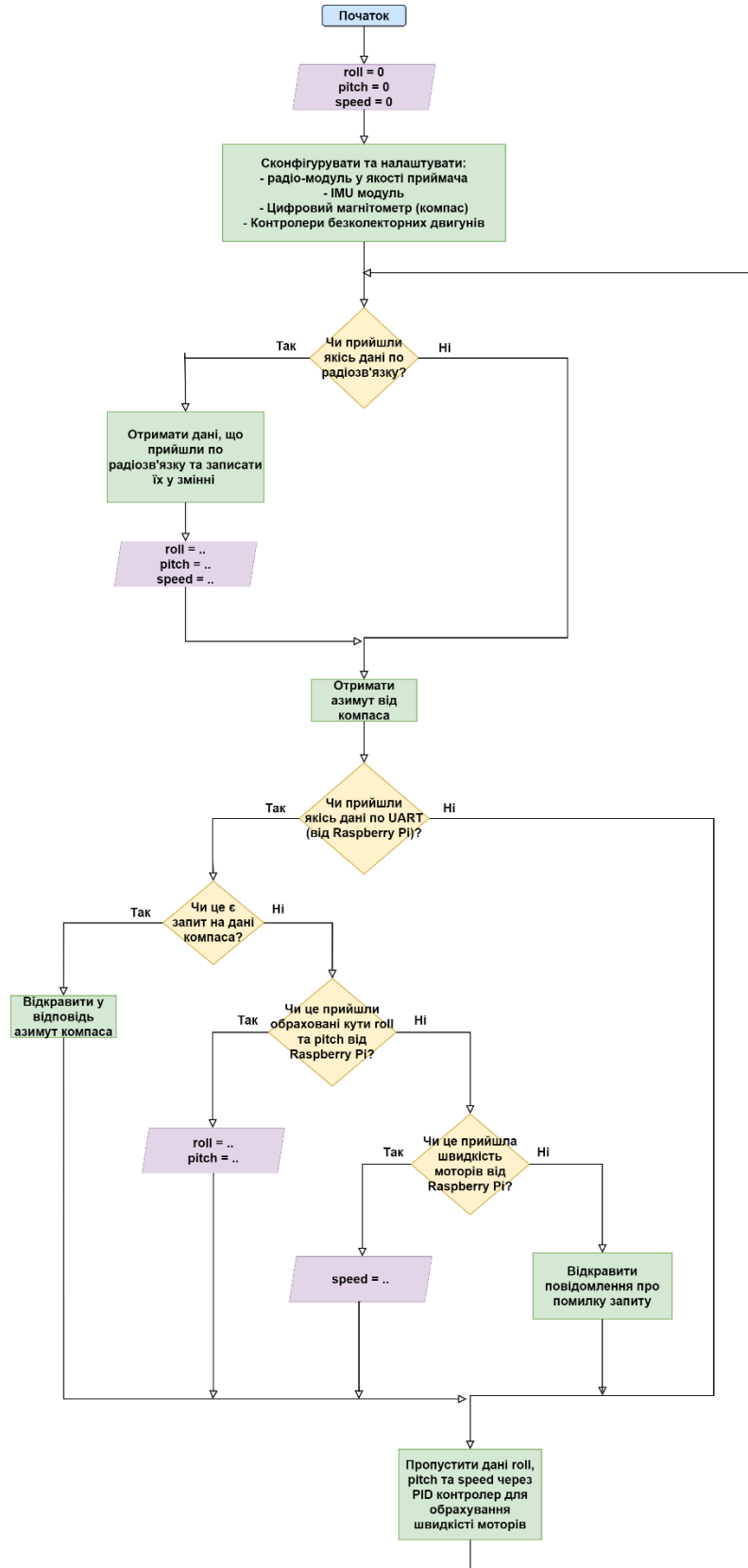


Рис. 2.17 - Блок-схема програми на Arduino Nano

2.6.2 Використання PID контролера для вирівнювання дрона і керуванням ним

PID контролер (пропорційно-інтегрально-диференціальний контролер) є одним з найпоширеніших методів автоматичного керування. Це алгоритм у програмному забезпеченні контролера польоту, який зчитує дані з датчиків і обчислює необхідну швидкість двигуна для досягнення бажаної частоти обертання. [35]

Він базується на комбінації трьох основних компонентів: пропорційного, інтегрального та диференціального. Розглянемо детальніше принципи роботи кожного з цих компонентів:

- **Пропорційна дія (P).** Пропорційна дія вимірює поточну помилку керування, яка виникає між вимірним значенням і заданим значенням (відхилення). За допомогою коефіцієнта пропорційності (K_p) помилка множиться на цей коефіцієнт і використовується для генерації вихідного сигналу керування. Чим більше відхилення, тим сильніше буде реакція системи.
- **Інтегральна дія (I).** (Минулі помилки). Інтегральна дія призначена для корекції системи в разі наявності постійної помилки відхилення (наприклад, вітер). Вона накопичує інтеграл від помилки в часі та множить його на коефіцієнт інтегрування (K_i).
- **Диференціальна дія (D).** (Майбутні помилки). Диференціальна дія орієнтована на прогнозування та реагування на зміну відхилення в часі. Вона вимірює швидкість зміни помилки відхилення та множить її на коефіцієнт диференціювання (K_d).

Комбінація цих трьох компонентів (P, I, D) дозволяє досягти стабільного та точного керування системою. Коефіцієнти (K_p , K_i , K_d) налаштовуються експериментально.

PID контролер в системі керування дроном дозволяє забезпечити точність позиціонування та стабільність польоту, реагуючи на зміни вітру, ваги дрона, зміну

обтікання повітря тощо. Він є ефективним і широко використовується методом керування дронами.

Основна мета PID контролера в дроні— виправити «помилку», регулюючи швидкість двигуна. Контур керування безперервно зчитує дані датчиків і розраховує швидкість двигуна, щоб мінімізувати помилку. «Помилка» — це різниця між заданим значенням (як швидко ми хочемо, щоб дрон обертався) і вимірюванням гіроскопічного датчика (як швидко дрон насправді обертається).

Схематично робота кожного з компонентів PID контролера показана на рисунку 2.18.

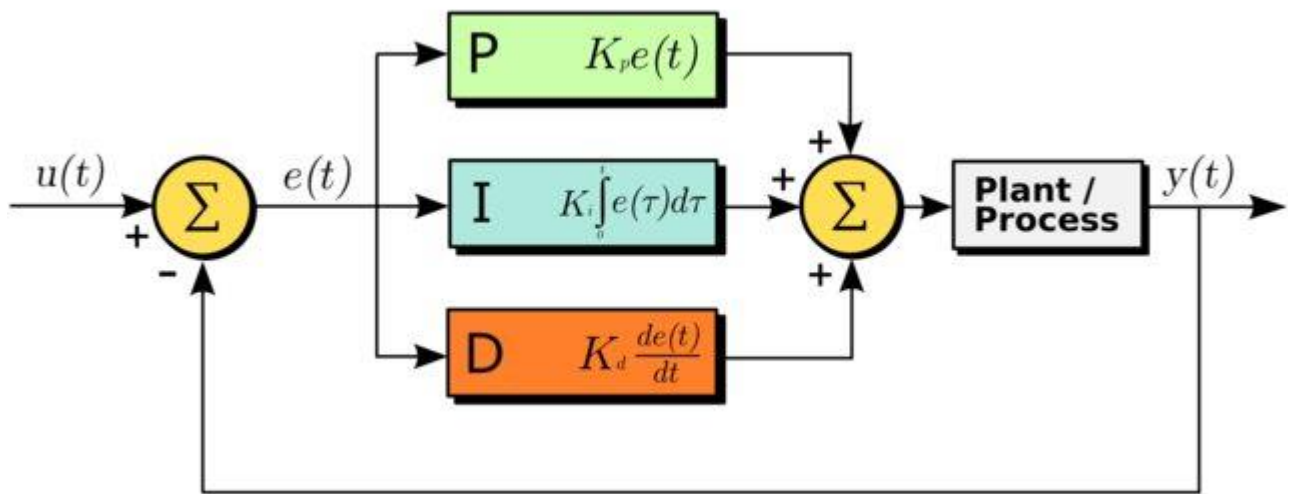


Рис. 2.18 – Робота PID контролера

У кодї функція PID контролера виглядає так:

```
void Pid_Control(int roll_pos, int pitch_pos, int throttle_pos)
{
    // перевести значення кутів з [-1000; 1000] до [-10; 10]
    roll_desired_angle = map(roll_pos, -1000, 1000, -10, 10);
    pitch_desired_angle = map(pitch_pos, -1000, 1000, -10, 10);

    // отримати кути з акселерометра-гіроскопа і порахувати total_angle_x total_angle_y
    Imu_Get_Angle(&total_angle_x, &total_angle_y);

    // обчислити помилку між бажаним значення кута і реальним значенням
    roll_error = total_angle_x - roll_desired_angle;
    pitch_error = total_angle_y - pitch_desired_angle;
}
```

```

// обчислити пропорційну (P) складову PID контролера  $K_p * e(t)$ 
roll_pid_p = roll_kp*roll_error;
pitch_pid_p = pitch_kp*pitch_error;

// обчислити інтегральну (I) складову PID контролера.
// вона повинна брати участь тоді коли ми близько до бажаної позиції, але потрібно
точно налаштувати помилку
// Для того, щоб інтегрувати потрібно просумувати попередній інтеграл з помилкою
помноженою на  $K_i$ 
// Це буде збільшувати значення помилки з кожним циклом доки система не вирівняється
if(-3 < roll_error <3)
{
    roll_pid_i = roll_pid_i+(roll_ki*roll_error);
}
if(-3 < pitch_error <3)
{
    pitch_pid_i = pitch_pid_i+(pitch_ki*pitch_error);
}

// обчислити диференціальну (D) складову PID контролера.
// вона обчислює швидкість помилки і дозволяє прогнозувати майбутні помилки
// швидкість - це зміна помилки в часі. Результат множимо на коефіцієнт
диференціювання  $K_d$ 
roll_pid_d = roll_kd*((roll_error - roll_previous_error)/elapsedTime);
pitch_pid_d = pitch_kd*((pitch_error - pitch_previous_error)/elapsedTime);

// Сумуємо усю компоненти PID контролера
roll_PID = roll_pid_p + roll_pid_i + roll_pid_d;
pitch_PID = pitch_pid_p + pitch_pid_i + pitch_pid_d;

// Мінімальне значення PWM сигналу становить 1000 us, а максимальне - 2000. Таким
чином,
// значення PID може коливатися більше, ніж -1000 і 1000, оскільки коли ми
// мають значення 2000, максимальне значення, яке ми можемо відняти, становить 1000, а
коли
// ми маємо значення 1000 для PWM сигналу, максимальне значення, яке ми можемо додати,
це 1000
// щоб досягти максимуму 2000us.
// +-400 достатньо, щоб виправити помилку
if(roll_PID < -400){roll_PID=-400;}
if(roll_PID > 400) {roll_PID=400;}
if(pitch_PID < -400){pitch_PID=-400;}
if(pitch_PID > 400) {pitch_PID=400;}

// Обчислимо ширину PWM сигналу. Сумуємо значення швидкості і значення коефіцієнтів
PID контролера

```

```

pwm_R_F = PID_MOTOR_START_COEF + throttle_pos + roll_PID + pitch_PID;
pwm_R_B = PID_MOTOR_START_COEF + throttle_pos + roll_PID - pitch_PID;
pwm_L_B = PID_MOTOR_START_COEF + throttle_pos - roll_PID - pitch_PID;
pwm_L_F = PID_MOTOR_START_COEF + throttle_pos - roll_PID + pitch_PID;

// Ще раз перевіряємо, щоб значення PWM сигналу не виходили за межі [1000, 2000]
//Right front
if(pwm_R_F < 1000) {pwm_R_F= 1000;}
if(pwm_R_F > 2000) {pwm_R_F=2000; }

//Left front
if(pwm_L_F < 1000) {pwm_L_F= 1000; }
if(pwm_L_F > 2000) {pwm_L_F=2000; }

//Right back
if(pwm_R_B < 1000) {pwm_R_B= 1000;}
if(pwm_R_B > 2000) {pwm_R_B=2000; }

//Left back
if(pwm_L_B < 1000) {pwm_L_B= 1000; }
if(pwm_L_B > 2000) {pwm_L_B=2000; }

// Зберігаємо попереднє значення помилки
roll_previous_error = roll_error;
pitch_previous_error = pitch_error;

// Встановлюємо нові швидкості моторів
esc_front_left.writeMicroseconds(pwm_L_F);
esc_back_left.writeMicroseconds(pwm_L_B);
esc_front_right.writeMicroseconds(pwm_R_F);
esc_back_right.writeMicroseconds(pwm_R_B);
}

```

2.6.3 Формула Ейлера для отримання кутів

Справжні кути обертання roll та pitch рахуються, використовуючи дані з IMU сенсора. Формули за якими рахуються ці значення наведені нижче. Їх виведення доступне за джерелом [36]

Для розрахунку обертання навколо осі X (roll) використана наступна формула:

$$\tan \phi_{yxz} = \frac{G_{py}}{\sqrt{G_{px}^2 + G_{pz}^2}}$$

Де:

- G_{px} , G_{py} , G_{pz} – необроблені значення прискорення по осях x, y, z, вчитані з реєстрів IMU модуля;
- ϕ_{yxz} – значення кута roll.

Для розрахунку обертання навколо осі Y (pitch) використана наступна формула:

$$\tan \theta_{xyz} = \frac{-G_{px}}{\sqrt{G_{py}^2 + G_{pz}^2}}$$

Де:

- G_{px} , G_{py} , G_{pz} – необроблені значення прискорення по осях x, y, z, вчитані з реєстрів IMU модуля;
- θ_{xyz} – значення кута pitch.

2.7 Тестування та валідація навігаційної системи для дрона

Останнім і одним з важливих завдань даної дипломної роботи є протестувати розроблену систему навігації. У даному підрозділі будуть протестовані усі 3 способи зв'язку: розрахунок roll та pitch на основі GPS координат та передача тих даних на Arduino Nano, передача даних через мобільний і через радіоз'язок, тестування вирівнювання дрона за допомогою PID контролера. В процесі тестування виведемо усі проміжні значення та кінцеві значення швидкостей кожного з чотирьох моторів, перевіримо, чи співпадають вони з очікуваними.

2.7.1 Тестування навігації за GPS

Test setup:

1. Розмістимо дрон з напрямком приблизно на південь (рисунок 2.19);

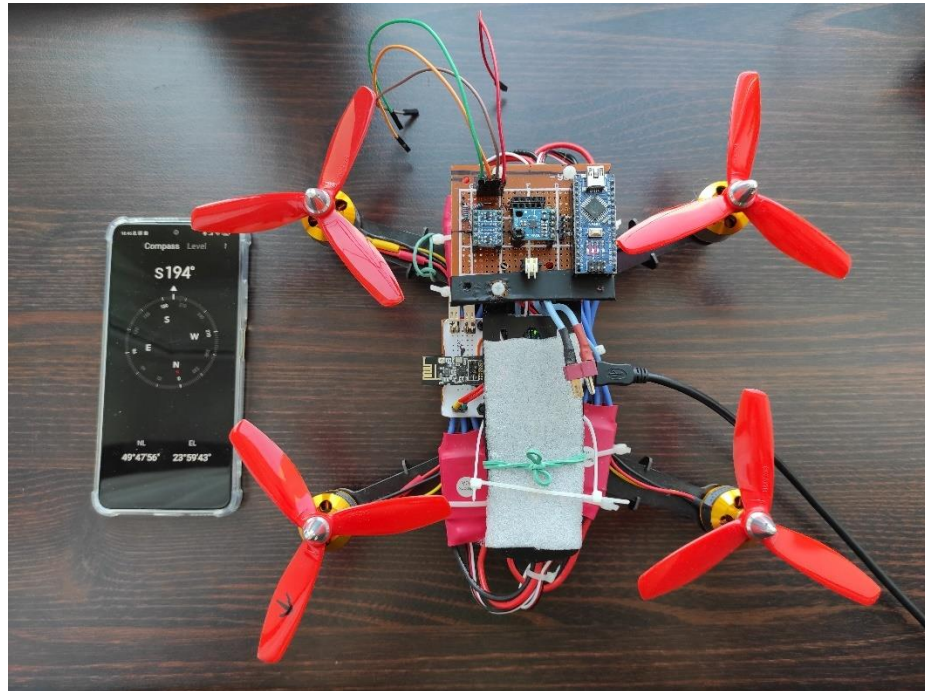


Рис. 2.19 – Дрон напрямком на південь

2. Підключимося до Raspberry Pi за допомогою ssh у Visual Studio, щоб бачити проміжні результати (рисунок 2.20);

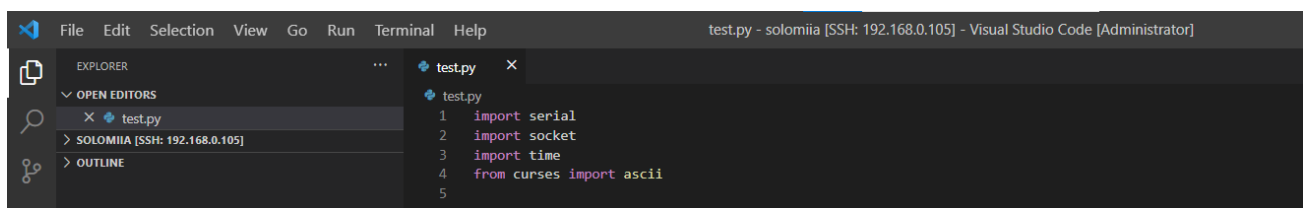


Рис. 2.20 – Підключення до Raspberry Pi за допомогою ssh у Visual Studio

3. Підключимося до Arduino Nano за допомогою Serial Monitor у Arduino Ide, щоб бачити кінцеві результати (рисунок 2.21);

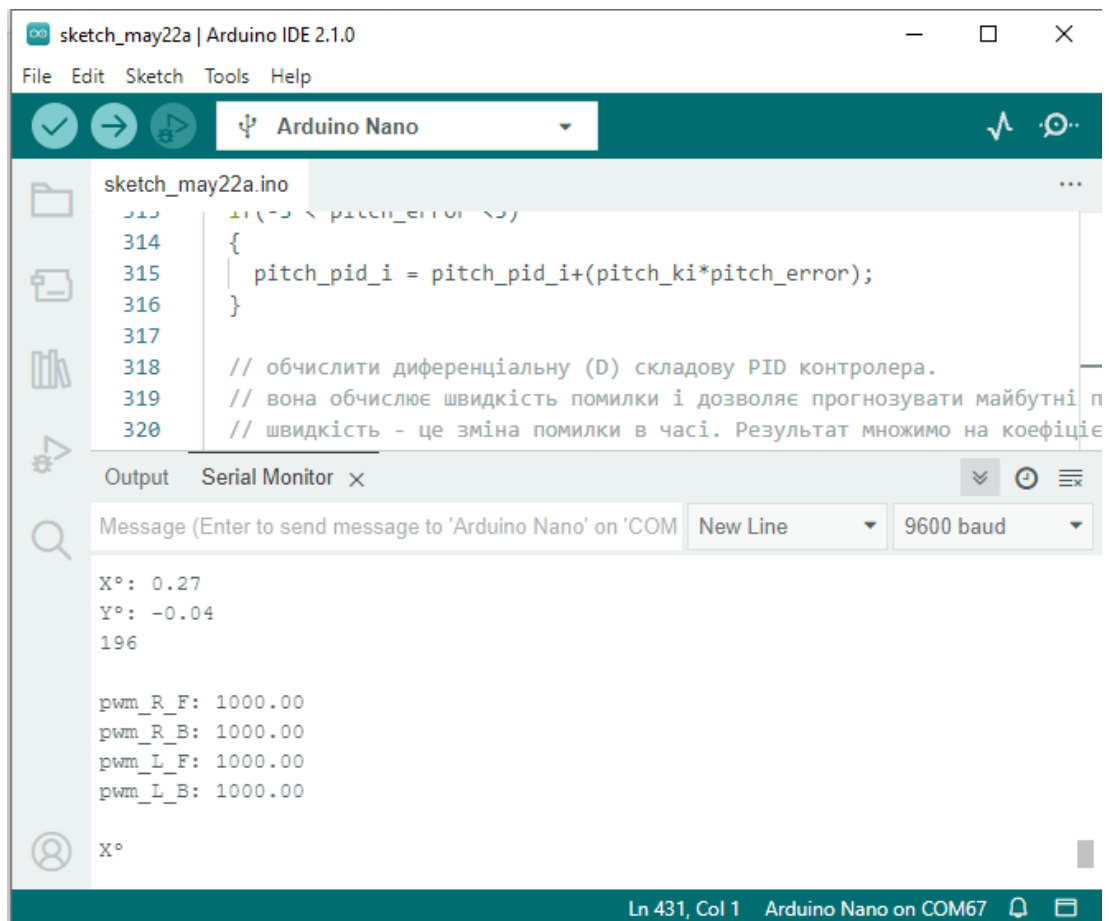


Рис. 2.21– Serial Monitor у Arduino IDE

Test Execution:

1. Запустимо застосунок для передачі даних по мобільному зв'язку та дочекаємося з'єднання з Raspberry Pi (рисунок 2.22 і рисунок 2.23);

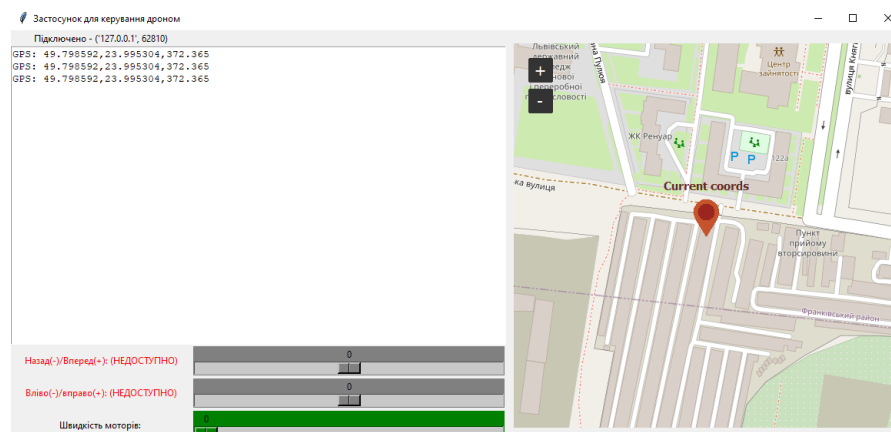


Рис. 2.22 – Застосунок підключений до Raspberry Pi


```

Connection to SIM868 successful
Connection through TCP is successful
Connection through TCP is successful
Latitude: 49.798592
Longitude: 23.995304
Altitude: 372.365
COG: +CGNSINF: 1,1,20230604082724.000,49.798592,23.995304,372.365,0.00,0.0,2,,0.9,1.2,0.8,,12,12,,44,,

Latitude: 49.798592
Longitude: 23.995304
Altitude: 372.365
COG: +CGNSINF: 1,1,20230604082726.000,49.798592,23.995304,372.365,0.00,0.0,2,,0.9,1.2,0.9,,12,11,,44,,

Latitude: 49.798592
Longitude: 23.995304
Altitude: 372.365
COG: +CGNSINF: 1,1,20230604082729.000,49.798592,23.995304,372.365,0.00,0.0,2,,0.9,1.3,0.9,,12,11,,44,,

```

Рис. 2.23 – Логи Raspberry Pi з успішним підключенням і даними GPS

2. Надішлемо GPS координати пункту призначення та швидкість (рисунок 2.24)

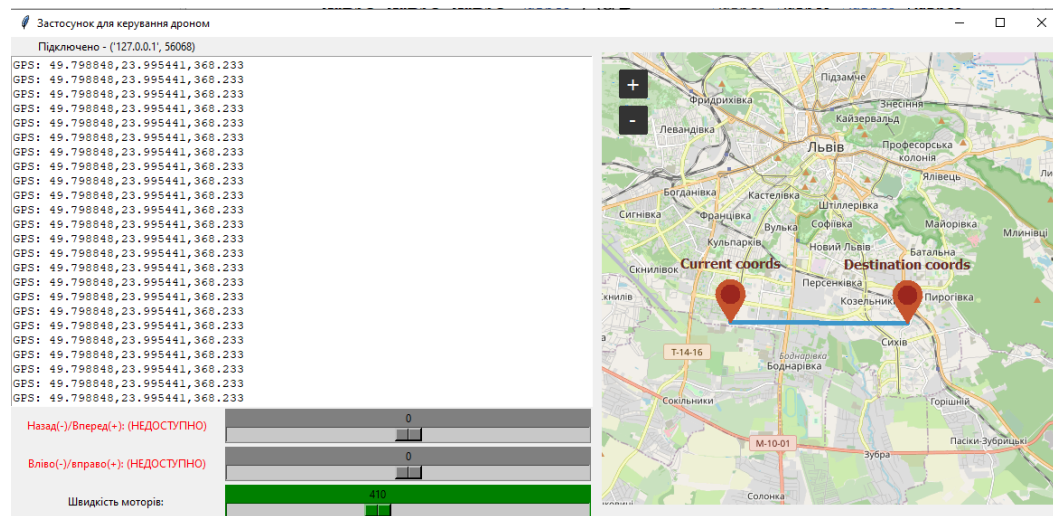


Рис 2.24 – Виставлені GPS координати пункту призначення та швидкість моторів

Test result:

1. Лог на Raspberry Pi (рисунок 2.25):

- Запит на дані компаса – успішно – 196 – **очікувано** (рисунок 2.19 – напрям на південь – 194+-5);
- Порахований азимут до пункту призначення – приблизно 90 градусів - **очікувано** (рис. 2.24 – напрям на схід)

- Пораховані roll та pitch – мінусові значення (вліво і назад) – практично максимальний рух вліво (значення 962) і рух назад (значення 272) – **очікувано** (оскільки дрон напрямлений на Пд. та повинен летіти на Сх.)

```

COMPASS DATA:
196

CALCULATED BEARING
90.20965357403031

CALCULATED ROLL AND PITCH
-962
-272

```

Рис. 2.25 – Лог Raspberry Pi

2. Дані швидкостей моторів (рисунок 2.26)

- Дані компаса – 196 - **очікувано** (рисунок 2.19 – напрям на південь – 194+-5);
- Значення кутів roll, pitch – виставлені такі, що були надіслані з Raspberry Pi - **очікувано**;
- Значення швидкості дрона– $410 + 1000 = 1410$ (швидкість моторів, тобто значення PWM є в межах [1000, 2000], тому +1000) – **очікувано** (див. рисунок 2.24);
- Значення швидкостей моторів – найбільші значення швидкостей у моторів з правої сторони – **очікувано**, оскільки дрон має летіти наліво.

```

compass azimuth°: 196
-962
-272
1410
pwm_R_F: 2000.00
pwm_R_B: 1718.96
pwm_L_F: 1331.04
pwm_L_B: 1014.89

```

Рис. 2.26 – Лог Arduino Nano

2.7.2 Тестування мобільного зв'язку

Test setup:

Такий ж як і в пункті 2.7.1

Test execution:

1. Запустимо застосунок, змодельуємо втрату GPS сигналу (переставимо GPS антену в приміщення) і виставимо значення roll (вліво/вправо) та pitch (вперед/назад) як 0 та 1000, що означає рух вперед. Виставимо швидкість. (рисунок)

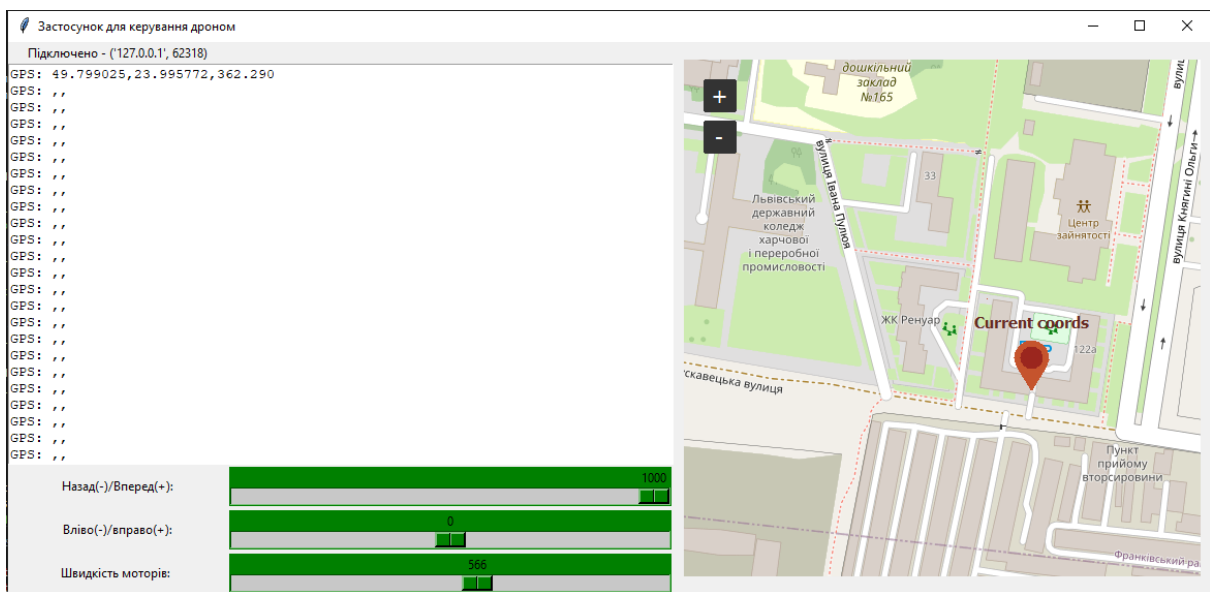


Рис. 2.27 – Застосунок при відсутності даних GPS та вибрані швидкість і напрям

Test results:

1. Лог Raspberry Pi:
 - Виставлені у застосунку швидкість та кути прийшли коректно – **очікувано** (рисунок 2.28);

```

solomia@raspberrypi:~ $ sudo python test.py
Connection to SIM868 successful
Pitch: 1000

Roll: 0

Speed: 566

```

Рис.2.28 – Лог на Raspberry Pi

2. Лог на Arduino Nano (рисунок 2.29)

- Значення кутів roll, pitch – виставлені такі, що були надіслані з Raspberry Pi - **очікувано**;
- Значення швидкості дрона– $566 + 1000 = 1566$ (швидкість моторів, тобто значення PWM є в межах [1000, 2000], тому +1000) – **очікувано** (див. рисунок 2.27);
- Значення швидкостей задніх моторів RB (задній правий мотор) та LB (задній лівий мотор) більші ніж значення швидкостей передніх моторів RF та LF – **очікувано**, оскільки бажаний напрям руху – вперед;

```

compass azimuth°: 200
0
1000
1566
pwm_R_F: 1508.04
pwm_R_B: 1867.56
pwm_L_F: 1494.44
pwm_L_B: 1853.96

```

Рис.2.29 – Лог на Arduino Nano

2.7.3 Тестування радіозв'язку

Test setup:

Такий ж як і в пункті 2.7.1

Крім того запустимо Serial Monitor в Arduino IDE для Arduino Nano, що знаходиться на пульті управління.

Test execution:

1. Виставимо значення швидкості дрона за допомогою потенціометра та значення roll та pitch за допомогою джойстика на пульті управління. Задамо рух по діагоналі – назад та вправо (рисунок 2.30).

```
Roll: 1000
Pitch: -1000
Speed: 1806
```

Рис. 2.30 – Лог на Arduino Nano, що знаходиться на пульті управління

Test result:

1. Лог на Arduino Nano (рисунок 2.31)
 - Значення кутів roll, pitch – виставлені такі, що були надіслані по радіозв'язку - **очікувано**;
 - Значення швидкості дрона– 1806 – **очікувано** (див. рисунок 2.30);
 - Значення швидкостей мотора LF (передній лівий мотор) найбільше, значення RB (задній правий мотор) найменше – **очікувано**, оскільки бажаний напрям руху – по діагоналі назад + вправо);

```
1000
-1000
1806
pwm_R_F: 1923.65
pwm_R_B: 1862.94
pwm_L_F: 1979.06
pwm_L_B: 1918.35
```

Рис. 2.31

2.7.4 Тестування вирівнювання дрона при неочікуваному нахилі

Test setup:

1. Підключимося до Arduino Nano за допомогою Serial Monitor у Arduino Ide, щоб бачити кінцеві результати
2. Нахилимо дрон вправо – рисунок 2.32.

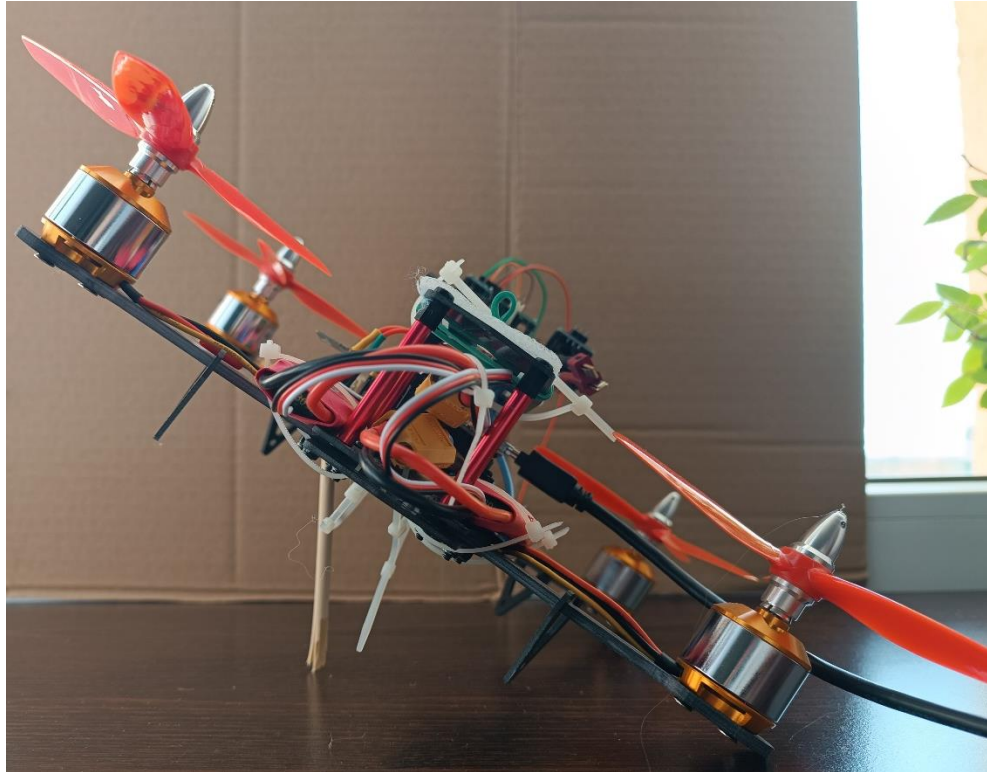


Рис. 2.32 – Дрон нахилений на ~ 30 градусів вправо

Test Execution:

1. Пультom управління виставимо швидкість, roll та pitch залишимо 0 (тобто ніякого бажаного нахилу не очікується);

Test Result:

1. Лог на Arduino Nano
 - Значення кутів з IMU модуля $x=32^\circ$, $y = 3^\circ$ – **очікувано** (див. нахил дрона на рисунку 2.32);

- Значення roll та pitch = 0, значення швидкості 1601 – виставляється на пульті управління – **очікувано**;
- Значення швидкостей мотора моторів справа (RF і RB) найбільше – **очікувано**, оскільки дрон намагається вирівнятися;

```
IMU angle X°: 32.36  
IMU angle Y°: 2.95  
compass azimuth°: 235  
0  
0  
1601  
pwm_R_F: 1804.51  
pwm_R_B: 1785.15  
pwm_L_F: 1646.85  
pwm_L_B: 1627.49
```

Рис.2.33 – Лог з Arduino Nano на дроні

ВИСНОВКИ

На основі проведеного аналізу існуючих програмних рішень був розроблений прототип навігаційної системи для дрона. У рамках розробки було реалізовано три способи зв'язку: політ за GPS координатами, політ за вказаним напрямком з використанням мобільного зв'язку та політ за вказаним напрямком на пульті управління з використанням радіозв'язку.

Було розроблене програмне забезпечення для пристроїв, таких як Raspberry Pi та Arduino Nano, і був створений комп'ютерний застосунок для мобільного зв'язку.

У розділі 1 розглянуто конструктивно-параметричний аналіз компонентної бази. У цьому розділі були представлені основні компоненти, що використовувалися для розробки програмного забезпечення навігаційної системи для дрона.

У розділі 2 було проаналізовано предметну область та існуючі рішення, детально розглянуто розробку навігаційної системи і висвітлено використані програмні рішення. Крім того, там представлена функціональна схема та блок-схеми для кожної з написаних програм. Алгоритми роботи важливих компонентів для навігаційної системи були детально описані, а кожен з методів зв'язку був протестований.

Результати розробки можуть бути використані як основа для подальших покращень і розвитку подібних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Raspberry Pi 3 Model A+ [Електронний ресурс] – Режим доступу: https://www.raspberrypi.org/app/uploads/2018/11/Raspberry_Pi_3A_product_brief.pdf
2. Bullseye – the new version of Raspberry Pi OS [Електронний ресурс] – Режим доступу: <https://www.raspberrypi.com/news/raspberry-pi-os-debian-bullseye/>
3. Arduino® Nano [Електронний ресурс] – Режим доступу: <https://docs.arduino.cc/static/12f5bdbb504fa7bf4901483241e92936/A000005-datasheet.pdf>
4. nRF24L01 Single Chip 2.4GHz Transceiver [Електронний ресурс] – Режим доступу: https://www.mouser.com/datasheet/2/297/nRF24L01_Product_Specification_v2_0-9199.pdf
5. Ai-Thinker радіомодуль NRF24L01 плюс 2.4 ГГц [Електронний ресурс] – Режим доступу: <https://arduino.ua/prod231-radiomodul-nrf24l01-2-4-ggc>
6. MPU-6000 and MPU-6050 Product Specification [Електронний ресурс] – Режим доступу: <https://www.robostore.com.ua/content/files/datasheet-mpu-6050-4746471.pdf>
7. SIM868_Hardware_Design_V1.00 [Електронний ресурс] – Режим доступу: <https://docs.rs-online.com/2190/0900766b815ca94b.pdf>
8. SIM868 Cellular GSM/GPRS/GNSS module [Електронний ресурс] – Режим доступу: <https://www.texim-europe.com/product/SIM868-SMM>
9. GSM/GPRS/GNSS/Bluetooth HAT for Raspberry Pi [Електронний ресурс] – Режим доступу: <https://www.waveshare.com/gsm-gprs-gnss-hat.htm>
10. GSM/GPRS/GNSS HAT User Manual [Електронний ресурс] – Режим доступу: https://copperhilltech.com/content/GSM_GPRS_GNSS_HAT_User_Manual_EN.pdf
11. QMC5883L Sensor Datasheet PDF [Електронний ресурс] – Режим доступу: <https://datasheetpdf.com/datasheet/QMC5883L.html>

12. Безколекторний двигун BLDC A2212 1000KV [Електронний ресурс] – Режим доступу: <https://www.mini-tech.com.ua/bldc-motor-a2212>
13. Регулятор швидкості Simonk 30 A [Електронний ресурс] – Режим доступу: <https://greenchip.com.ua/29-0-791-1.html>
14. Phantom 4 Pro V2.0 [Електронний ресурс] – Режим доступу: https://www.dji.com/phantom-4-pro-v2?site=brandsite&from=eol_phantom-4-pro
15. DJI Ryze Tello review [Електронний ресурс] – Режим доступу: <https://www.space.com/DJI-ryze-tello-review>
16. ОГЛЯД АГРОДРОНІВ УКРАЇНСЬКОГО ВИРОБНИЦТВА [Електронний ресурс] – Режим доступу: <https://www.smartfarming.ua/ohlyad-ahrodroniv-ukrayins-koho-vyrobnytstva-chastyna-1/>
17. Радіозв'язок [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Радіозв'язок>
18. Difference between GSM and GPRS [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/difference-between-gsm-and-gprs/>
19. Satellite navigation [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Satellite_navigation
20. TCP/IP Model [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/tcp-ip-model/>
21. Марусенкова Т.А. Програмування мікроконтролерів: конспект лекцій / Сам.видав № 6883, 2016
22. Широтно-імпульсна модуляція [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Широтно-імпульсна_модуляція
23. Tkinter [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Tkinter>
24. socket — Мережевий інтерфейс низького рівня [Електронний ресурс] – Режим доступу: <https://docs.python.org/uk/3.9/library/socket.html>

25. threading — Паралелізм на основі потоків [Електронний ресурс] – Режим доступу: <https://docs.python.org/uk/3/library/threading.html>
26. TkinterMapView - simple Tkinter map component [Електронний ресурс] – Режим доступу: <https://github.com/TomSchimansky/TkinterMapView>
27. pySerial [Електронний ресурс] – Режим доступу: <https://pyserial.readthedocs.io/en/latest/pyserial.html>
28. Азимут [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Азимут>
29. Calculate distance and bearing between two Latitude/Longitude points [Електронний ресурс] – Режим доступу: <https://www.ancientportsantiques.com/wp-content/uploads/Documents/ETUDESarchivees/MedNavigationRoutes/MedNav/Tri-goSpherique.pdf>
30. Drone navigation system [Електронний ресурс] – Режим доступу: <https://www.immervision.com/drone-navigation-system/>
31. Wire [Електронний ресурс] – Режим доступу: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>
32. Optimized high speed nRF24L01+ driver class documentation [Електронний ресурс] – Режим доступу: <https://nrf24.github.io/RF24/>
33. Servo [Електронний ресурс] – Режим доступу: <https://reference.arduino.cc/reference/en/libraries/servo/>
34. QMC5883L Compass Arduino Library [Електронний ресурс] – Режим доступу: <https://github.com/mprograms/QMC5883LCompass>
35. Understanding PID in Drones [Електронний ресурс] – Режим доступу: <https://oscarliang.com/pid/#Understanding-PID-in-FPV-Drones>
36. Tilt Sensing Using a Three-Axis Accelerometer [Електронний ресурс] – Режим доступу: <http://www.cas.mcmaster.ca/~rzheng/course/CAS765fa13/YueSun.pdf>