RMIT UNIVERSITY

# EEET2610 – ENGINEERING DESIGN 3

# Project Final Report

## Design And Control of a Quadcopter

## Lecturer: Dr. Dinh-Son Vu

Group G_ Tutorial 2: Thursday 14:30 – 16:00


| | |
|---|---|
| Dinh Ngoc Minh | s3925113 |
| Nguyen Nam Cuong | s3891758 |
| Phan Nhat Minh | s3904422 |
| Tran Thinh Thu | s3894296 |


January 19, 2024

School of Science, Engineering, and Technology

# Table of Contents

# 1. Abstract

This project aims to create an affordable, wireless-controlled quadcopter from scratch, offering a budget-friendly alternative to more expensive drones. The project emphasizes cost-effectiveness by targeting a broader audience, including students, amateur researchers, and hobbyists. The design features a quadcopter powered by a lithium polymer (LiPo) battery, chosen for its lightweight and high energy density, which enhances flight time and efficiency. Four brushless motors ensure reliable propulsion with an excellent power-to-weight ratio. The quadcopter's control system combines an ESP32 microcontroller and a Motion Processing Unit (MPU), utilizing a Proportional Integral Derivative (PID) control mechanism for stability and balance during flight. The ESP32 also handles wireless signal reception for drone maneuvering. To further reduce costs, recycled wood is used for the drone's airframe. The project's deliverables include a fully operational quadcopter and detailed assembly instructions, positioning this drone as a competitive option for various applications such as education, surveillance, and entertainment. Additionally, the project involves ongoing research, enhancements, and collaborations to advance drone technology and make it more accessible to a broader audience. A significant issue was encountered in balancing the drone in the project's outcomes. This challenge arose from the implementation of cascading PID control, where the PID gains for the gyroscopic sensors and angular positioning were identical. This uniformity in PID settings resulted in an inability to effectively maintain the drone's balance.

This comprehensive report on the drone project begins with an Introduction, setting the context and objectives. The main content is structured into detailed Project Task Descriptions, divided into three work packages. The first work package focuses on the drone's design, including CAD modeling, mechanical assembly of components like the frame, motor mount, landing gear, controller, system wiring, and a step-by-step assembly guide. The second work package covers unit testing, objectives, remote controller design, components research, and calibration processes. The third package addresses the control and validation of the drone. Time Management sections outline the project's progression from the proposal stage to the final report. Resources Management provides a meticulous breakdown of materials needed for each work package and the overall project. Risk Assessment is thorough, evaluating internal, external, and safety risks. The report concludes with a comprehensive summary of the findings and outcomes of the project.

# 2. Introduction

This report analyzes quadcopter maneuvering, focusing on essential movements such as hovering, pitch, roll, and yaw. The operational stability of a quadcopter is based on the effective implementation of cascading PID (Proportional, Integral, Derivative) control. A significant aspect of this project is identifying precise PID gains that ensure the drone's optimal balance, with input data derived from the MPU6050 module across the x, y, and z axes. The intricacies of both PID and cascading PID systems are thoroughly explored, offering insights into their functions and contributions to quadcopter control. The structural design, featuring motors on four beams, is critical. These motors, rotating in alternating clockwise and counterclockwise directions, are crucial for maintaining stability and enhancing maneuverability. This configuration aids in balance and significantly influences directional control. The project emphasizes precision in control system design, aiming to advance the understanding and application of PID controls in quadcopter dynamics for improved stability and maneuverability. Additionally, the project seeks to determine the optimal PID gain for Cascading PID to balance the drone and to implement the ESP-NOW protocol for reliable, fast communication between the remote controller and the drone.

## 2.1. Literature Review

Gaining an in-depth understanding of the Cascading PID implementation in quadcopters, along with their maneuvering capabilities, is pivotal for the success of this project. It is the cornerstone for steering the PID tuning process towards the right direction, ensuring the project aligns with its final objectives. This literature review is dedicated to examining the significance of Cascading PID in quadcopters and exploring various methodologies for tuning the PID settings. The goal is to optimize the quadcopter's performance by fine-tuning its control systems, essential for achieving precise and efficient maneuverability.

Jaehyun et al. presents a approach to enhance the stability and maneuvering of quadcopters. The research focuses on using Long Short-Term Memory (LSTM), a type of neural network that can optimize the Proportional-Integral-Derivative (PID) control system in quadcopters [1]. Traditional PID tuning methods are time consuming and require expert knowledge, but the LSTM-based approach streamlines this process, offering a more efficient solution. The study involves the development of a PID simulator algorithm and the use of back-propagation neural networks to approximate the relationship between PID gain values and drone stability. The researchers also employ the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) for further optimization. This method not only ensures stable hovering but also improves the overall maneuverability of the quadcopter, which is critical for various practical applications. The effectiveness of this method is validated through the construction of a drone and experimental comparisons between the LSTM-predicted motions and dynamic simulation results. The research marks a significant advancement in drone control technology, emphasizing efficiency and reducing reliance on expert input for PID tuning. This innovation holds promise for broader applications in the field of drone technology, particularly in scenarios where stable and precise flight is crucial.

IMU sensors in drones measure angular velocity and acceleration to keep the drone balanced. Azfar and colleagues have developed a way to use IMU sensor data with a PID controller for better quadrotor flight stability. They use accelerometer and gyroscope data from the IMU, processed by the Kalman Filter algorithm for precise flight control. This method greatly enhances quadcopter stability. Furthermore, Ho and colleagues suggest a different technique to stabilize the quadcopter by estimating its mass during flight. Their Instrumental Variable method is shown to be more effective than the Least Squares Estimation and the Extended Kalman Filter. Both

studies highlight the importance of statistical methods in managing drone stabilization using IMU data.

Kan et al focuses on the development of a drone with self-flying capabilities using GPS [2]. The drone uses a Raspberry Pi 2.0 microcomputer for controlling its flight. The research aims to improve the accuracy of the drone's GPS to reduce location errors. They conducted tests to compare the planned and actual flight paths of the drone to assess its performance. The results showed that integrating GPS with a microcomputer significantly improves the drone's ability to fly autonomously. This development is important for applications that need drones to navigate and control themselves accurately.


*Figure 1. ESP-Now communication*

A strong wireless communication protocol is vital for remotely controlling drones. Abdul et al. highlighted the ESP-NOW protocol, leveraging the ESP32 microcontroller for efficient peer-to-peer communication without needing a router. This protocol excels in indoor settings where signal interference is common [3]. Notably, ESP-NOW can transmit data effectively even with low power, and it boasts a fast transmission speed of 1 ms for 1 Byte of data. Such capabilities are ideal for drone operations requiring continuous and reliable communication. The research underscores ESP-NOW's effectiveness in connecting drones and controllers, particularly in challenging indoor environments. This advancement is crucial for the development of reliable communication methods in IoT devices, especially in drone technology where uninterrupted, speedy communication is essential for operational success and safety.

## 2.2. Problem Statement

Creating a quadcopter that is efficient, versatile, and reliable presents several challenges. Key areas of focus include:

**Design and Modeling:** Developing a CAD model for the quadcopter is crucial. It must balance weight, durability, and aerodynamics to ensure stable flight.

**Stability:** The project is based on the inverted pendulum concept, making balance a complex task, especially with four propellers. Adjusting each propeller's speed is vital for achieving stability.

**Efficiency:** The design must include effective system wiring and PCB layout to ensure efficient power distribution and communication among all components.

**Maneuverability:** Without a magnetometer, accurate yaw angle estimation is challenging. Fine-tuning the control system, particularly the PID settings for roll and pitch, is essential for responsive flight.

**Safety:** Safety is a major concern. Each quadcopter part must undergo thorough testing and validation before assembly. An emergency stop feature is crucial for crew safety and minimizing components' risks.

## 2.3. Contribution

Drone creation involves collaboration across mechanical engineering, electronics, software engineering, and aerodynamics. Key steps and expertise required include:

**Design and Mechanics:** Building the frame and body, focusing on weight and material choice for stability and durability.

**Propulsion System:** Selecting motors and propellers to ensure enough power and load-bearing capacity.

**Electronics and Power Supply:** Arranging and soldering components like flight controllers and GPS systems onto the PCB.

**Control System and Software Programming:** Developing control algorithms and software for stabilization, navigation, and user command responsiveness.

**Communication System:** Implementing wireless communication between remote and drone microcontrollers.

**Sensors:** Utilizing sensors like accelerometers for real-time flight data, aiding in stability and navigation.

**Testing and Calibration:** Ensuring all systems and components function together correctly.

**Compliance and Regulations:** Adhering to aviation regulations for safety and operational standards.

**Mechanical Assembly:** Assembling components as per CAD designs.

**Quality Assurance:** Conducting quality checks for safety and reliability.

**Continuous Improvement:** Regularly enhancing the drone's components and addressing any issues.

# 3. Project Task Description

The project task is divided into three work packages and shows what has been achieved from the project proposal.

## 3.1. Work packages 1: Design of the drone

Work Package 1 focuses on the drone's design, including three critical subsections to ensure an effective assembly procedure. The first subsection focuses on CAD modeling, where the drone's structural design is carefully crafted using SolidWorks design tool, laying the foundation for its physical attributes and aerodynamic properties. The second subsection delves into the drone's system wiring and PCB design using EasyEDA, detailing the electrical connections and circuit board layouts essential for efficient operation and control. The final subsection involves a step-by-step assembly guide for the drone, ensuring that each component is correctly and safely integrated.

### 3.1.1. CAD modeling of the drone

CAD modeling involves creating detailed 3D digital representations of drone components and structures using SolidWorks design tool. This process enables engineers and designers to visualize and refine the drone's size, ensuring accurate geometry and optimal weight distribution. These 3D models also play a crucial role in aiding team members and users to better understand the drone's design, facilitating a smoother transition from digital models to physical prototypes.

**Drone Frame:** The drone CAD began with designing the frame's appearance, tackling challenges such as ensuring the drone's maximum size and propeller diameters were compatible to avoid collisions. Various materials, including wooden beams (2 x 2 x 21cm) and plates (15 x 15 x 0.5cm), were evaluated and redesigned in SolidWorks for 3D integration. The True X layout was selected for suitability after exploring layouts like True X and H. This layout involved assembling beams and plates in SolidWorks, ensuring correct positioning and sufficient space at the corners to prevent propeller collisions. The structural integrity of the assembly was reinforced by using M4 and M5 wood screws of 20mm and 25mm lengths for securing the components, both digitally and in physical assembly. Due to time constraints and limited access to large-scale 3D printers, the team decided to omit propeller protectors from the final design.

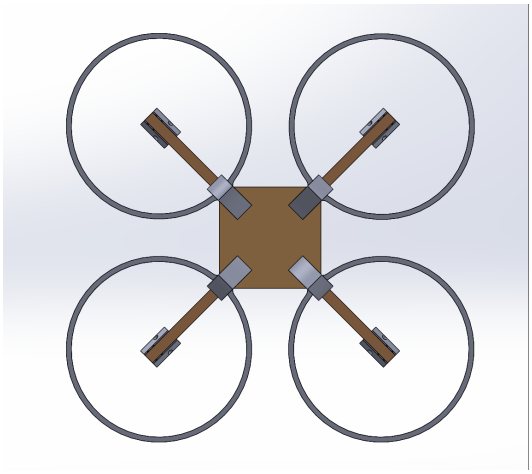


*Figure 2. True-X layout of the drone*

**Motor Mount:** The motor mount is a critical mechanical component designed to hold a BLDC motor in place securely. Its primary function is to provide a stable and secure platform for the motor, maintaining its position despite vibrations, movements, or external forces. Accurate measurement of bolts is vital when attaching the motor to the mount, as this precision ensures the motor remains firmly in place during operation, contributing to the stability and reliability of the drone's mechanical components. In this specific design, the wooden beams are securely fastened using M4 wood screws, applied from three distinct angles or surfaces, ensuring a robust and secure mounting for the motor.

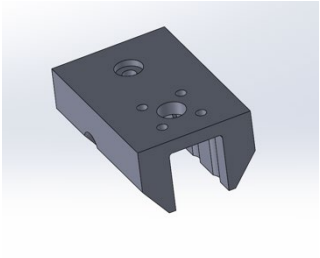


*Figure 3. Motor mount design*

**Landing Gear:** The landing gear serves a vital role in shock absorption during takeoff and landing. Its primary functions include ensuring stability on the ground to prevent tipping and protecting the drone's payload and sensitive components from damage during landings. The landing gear is designed with a bent rectangle shape, focusing on internal bonding to enhance structural strength and minimize impact during landings.

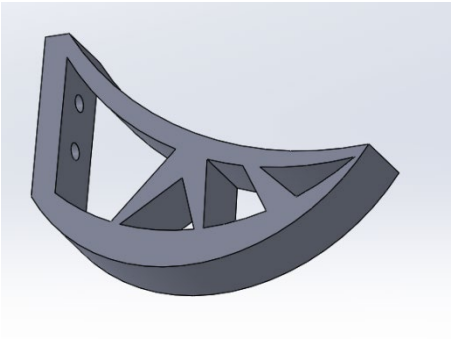


*Figure 4. Landing Gear design*

### 3.1.2. System Wiring and PCB design.

**Schematic design of the drone:** The wiring diagram outlines the power distribution and communication pathways for a drone system, featuring a PDB XT60 that distributes power from a 14.8V Lipo battery to four 30A Brushless ESCs, which in turn control individual motors via PWM signals from the ESP32 microcontroller pins GPIO32, GPIO33, GPIO25, and GPIO26. Additionally, the ESP32 is powered through its VIN pin by a regulated 5.0V output from the PDB. The MPU6050 sensor, requiring a 3.3V supply, is connected to the ESP32's 3V3 pin and communicates via I2C, with SCL and SDA lines attached to pins 22 and 21 of the ESP32, strategically placed to avoid interference from the PWM signals. Furthermore, the GPS NEO8M module, utilizing UART communication, connects its RX pin to the ESP32's TX2 on pin 17 and its TX to the ESP32's RX2 on pin 16, ensuring precise satellite signal reception and location tracking for the drone. UART2 is preferred because UART0 serves as the USB port, and UART1 is allocated for SPI flashing. The electrical circuit of the drone in milestone 1.2 is present in the diagram below.

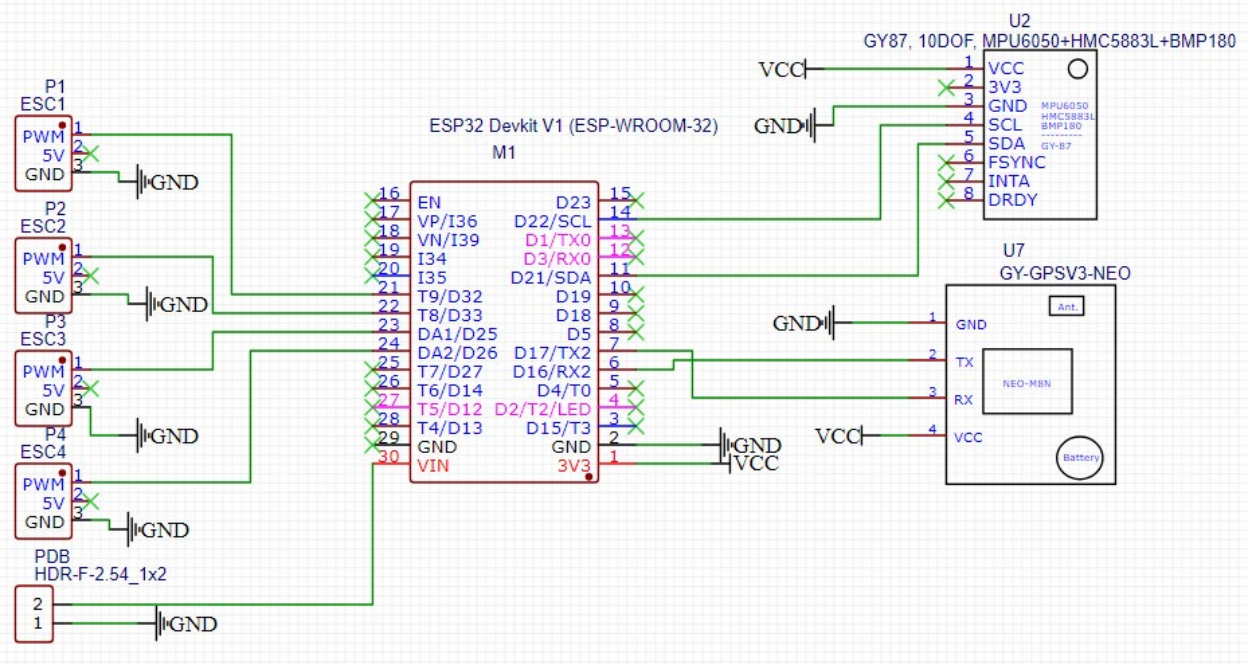
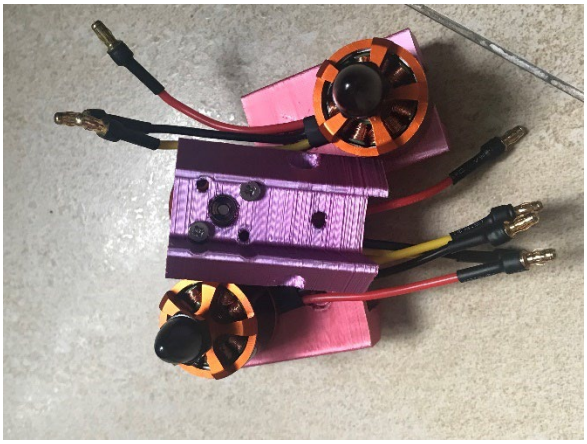



*Figure 5. The electric schematic design of the drone*

### 3.1.3. Assembly guidance

Assembling the drone requires precise steps, beginning with 3D printing and gathering materials, followed by manual tasks such as drilling and bonding. The assembly list includes 4 printed legs, wooden beams, motor mounts, a wooden bottom plate, a mica plastic top cover, a drone PCB with components, 4 ESCs, and 4 BLDC motors, joined using M4x20 wood screws. The initial step involves designing and preparing CAD files for 3D printing. Next, wooden beams and plates are collected as instructed. Then, holes are drilled into the beams and plates for component assembly. Each beam is then attached to its motor mount and landing gear using screws. Motors are mounted onto their respective motor mounts.



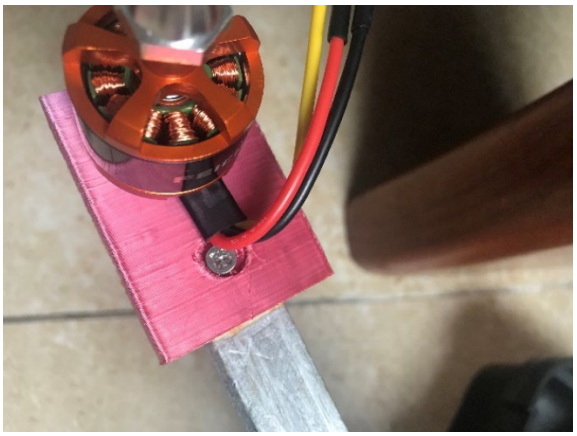


*Figure 6 & 7. Assembly of the motor's mount*

This stage also included attaching the drone's landing gear, a key component for safe landings and take-offs.

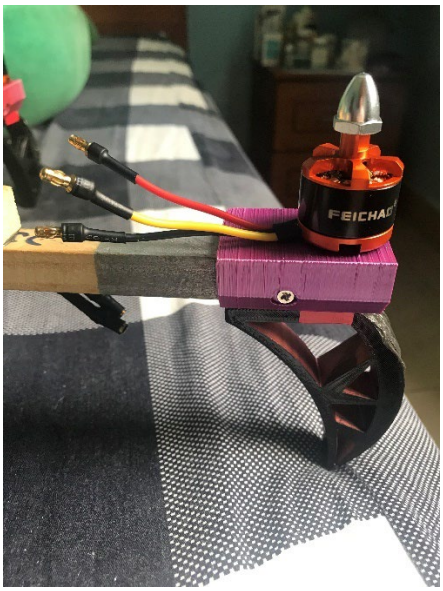


*Figure 8 & 9. Assembly of the drone's leg and final result*

ESCs are tied to wooden beams with zip-lock ties. Subsequently, ESCs are connected to the drone's PCB. Additionally, the IMU is stabilized on a small wooden piece at the center of the drone with a hot glue gun, enhancing the drone's navigational accuracy.

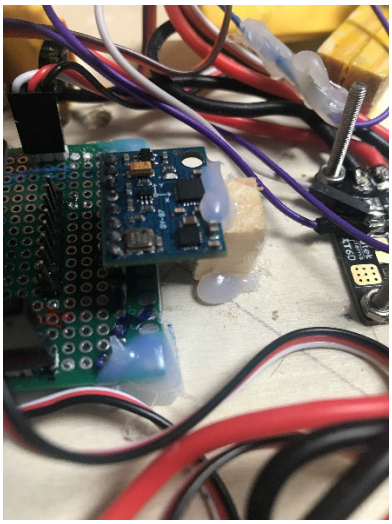
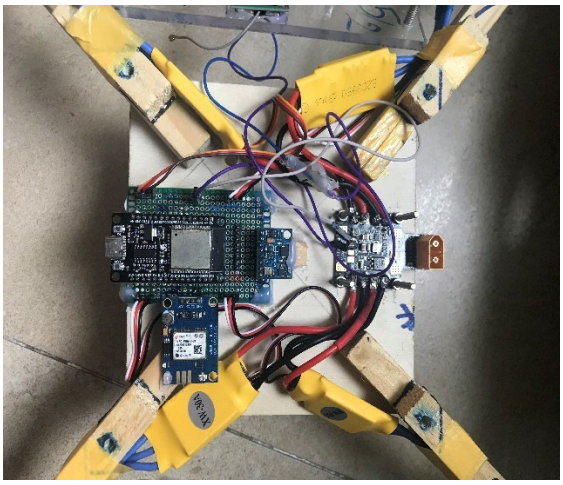


*Figure 10. Wooden piece stabilizes the IMU*

The mica top cover is then placed on the drone. The final assembly step is positioning the LiPo battery onto the top cover, completing the drone's assembly.



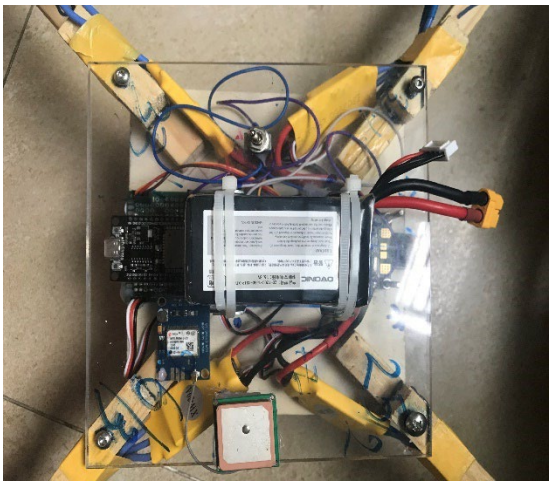


*Figure 11 & 12. Final assembly of the drone's body*

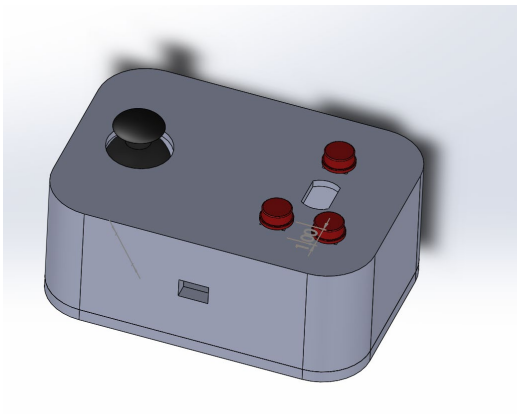
Each stage was critical to building a quadcopter, with great attention to detail and collaborative work required for the drone's successful assembly.

## 3.2. Work package 2: Unit testing

Work Package 2 (WP2) focuses on crucial aspects of the project's development, centering on three main objectives. The first objective involves the design of the remote control, ensuring its efficacy and user-friendliness. The second objective concentrates on the calibration and efficient functioning of the driving unit, a vital component for operational success. Lastly, the third objective is dedicated to thoroughly testing and validating sensors, confirming their accuracy and reliability. Each component will undergo individual testing to guarantee optimal performance before integrating into the system. This methodical approach ensures that each element meets the highest quality and functionality standards.

### 3.2.1. Design of the remote controller

**The CAD controller design:** The controller case primarily functions as a protective enclosure, safeguarding essential buttons and components such as the yaw button, a potentiometer, a stop button, a joystick, and the ESP32 module.



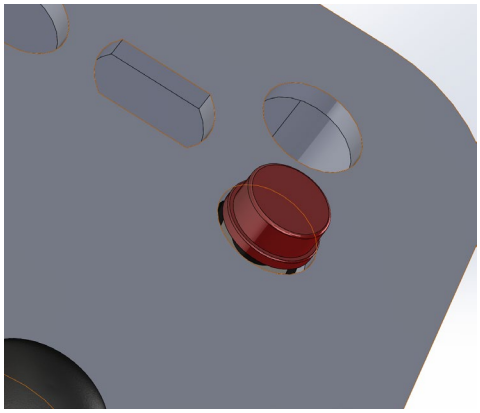


*Figure 13 & 14. CAD Controller & Final CAD design of the remote control*

As the design progressed, it became necessary to revise the CAD model, particularly the buttonhole on the remote control. The decision was made to change the shape of the buttonhole from square to round (14mm diameter). This adjustment was crucial to facilitate the easy attachment of the red module button, ensuring that the button could be pressed without any hindrance. The modification was driven by a need for practical usability and ease of assembly, ensuring that the remote control would be both functional and user-friendly.

The top and bottom sections of the remote control are fastened together using four M4-sized bolts. The bottom section of the controller includes a 4cm square extension, purposefully designed to support and position the joystick. Additionally, the buttons and potentiometer are securely attached to the controller using the hot glue gun, following the design strategy established at the beginning of the design process.

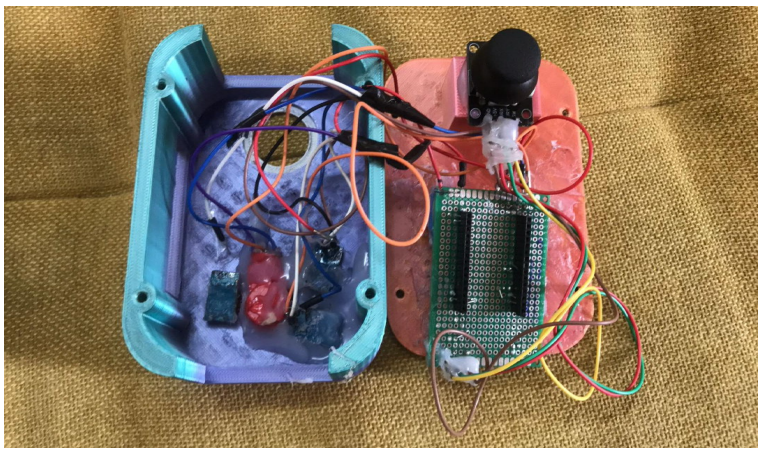


*Figure 15. Controller assembly*

**Schematic design of the remote control:** The wiring diagram displays the electrical connections for a remote control, utilizing an ESP32 as the central microcontroller. It shows the ESP32 connected to a potentiometer (U2), three buttons (U3, U4, U5), and a joystick (U6). The potentiometer's three pins are connected to ground (GND), an analog output (OUTPUT), and voltage (VCC), while each button has similar connections for GND, digital output (OUT), and $V_{CC}$. The joystick is connected to the ESP32 with five wires, corresponding to GND, a 3.3V supply, and three outputs for the x-axis (VRX), y-axis (VRY), and switch (SW). Additionally, the diagram includes grounding points and power supply lines, with the ESP32 receiving power through the $V_{IN}$ pin connected to +5V and a common ground shared among all components. The digital and analog signal connections from the potentiometer, buttons, and joystick to the ESP32's GPIO pins are designed to allow for input capture and processing for remote control operations. The electrical circuit of the remote control in milestone 2.1 is present in the diagram below.

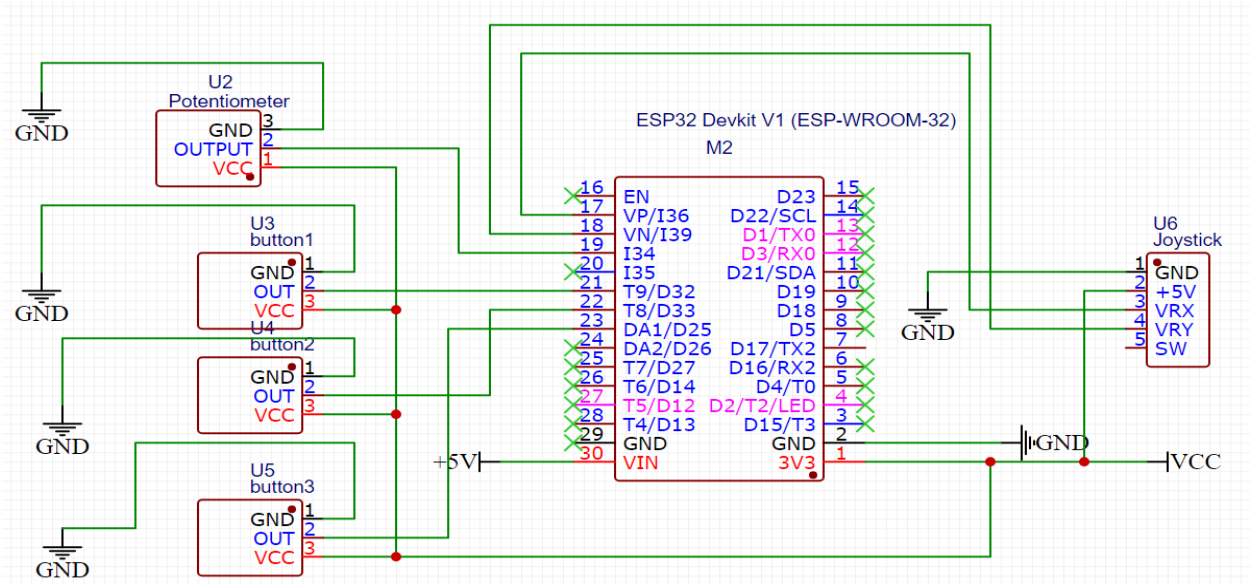



*Figure 16. The electric schematic design of the remote control*

One of the most significant challenges encountered was searching for precise components matching the project's requirements. This included sourcing an ESP32 module with 30 pins, a GY87 IMU, and a GY-GPSV3 GPS module. The time-consuming aspect revolved around locating components with identical pin layouts and sizes, ensuring compatibility with the design. In the process of research and PCB design, it was discovered that ESP NOW is incompatible with ADC2. Therefore, a careful examination of the pin allocation for the remote controller, as shown

in the figure below, was conducted to ensure that its ports are consistently allocated to ADC1. This ensures compatibility and effective functioning of the system.

### 3.2.2. Calibration and functioning of the driving unit.

Implementing the structure for wireless communication is a crucial phase. The coding format and structure utilized user-defined libraries, facilitating easier tracking, and debugging of the code. GitHub served as the version control platform for storing the project, with two separate repositories created for the drone and controller. The links provided below offer a comprehensive view of the entire structure.

- Drone's source code (new update in main branch).
- Remote control's source code (new update in main branch).

**ESPNow Remote configuration:** The initial step in setting up ESPNow for the remote involved creating header files for both the drone and remote, which incorporated the ESP-Now library and declared struct variables for two-way data communication. These struct variables had to have a unique size to differentiate the data types sent between two ESP32s. For efficient data transfer, ESPNow configuration was necessary for both sender and receiver, requiring the addition of peers based on the ESP32's MAC address and the implementation of callback functions onDataSent and onDataReceived. These functions managed the transmission status and data reception, respectively. Furthermore, to facilitate the thurst of the quadcopter, a function was introduced in the remote to send potentiometer data to the motors. This function accounted for rapid changes in potentiometer readings, ensuring smooth data transmission. Similarly, for joystick control, a function sendDataIfJoystickMoved was used to delay ADC data transmission, allowing the receiver ample time to process incoming data based on joystick movements along the x and y axes.

**ESPNow Drone configuration:** The ESP-NOW setup for the drone begins with configuring the ESP32 in Station mode, initializing ESP-NOW, and registering callback functions for sending and receiving data. A peer, identified by its MAC address, is configured without encryption on the default channel and added to the ESP-NOW network, enabling wireless communication with the controller. The drone's ESP-NOW setup involves initializing various structures to receive joystick movements, voltage levels, button states, calibration signals, and tuning parameters from the

controller. The onDataSent function, currently without implementation, is designed to handle the status of sent data, while onDataReceived processes incoming data based on its size, distinguishing different types of data for appropriate actions. This setup ensures effective communication and response to controller inputs, which is crucial for controlling the drone's orientation and movement. The controller's ESP-NOW configuration closely mirrors the drone's, focusing on aligning data structures in size for consistent and accurate data transmission.

**ESCs and Motor:** Setting up and calibrating the Electronic Speed Controllers (ESCs) for the drone involves ensuring the ESCs recognize the full throttle range, with the process beginning by setting the throttle's maximum signal to 2000 PWM and the minimum to 1000 PWM, as required for the 30A ESCs. The calibration, guided by ArduiPilot, consists of two stages: connecting and disconnecting the battery. During each stage, the ESCs must first receive the maximum signal, enter calibration mode, and then receive the minimum signal, with these steps repeated when the battery is reconnected. The calibration algorithm is detailed in a flow chart. Functions were created to pause for user input and display calibration steps, such as disconnecting the battery, connecting it, and concluding the calibration. Signals of 2000 PWM and 1000 PWM are sent in respective stages to the ESCs through ESP-Now communication from the remote to the drone. The drone's setup includes receiving and relaying these signals to the ESCs, ensuring proper calibration for optimal motor control.

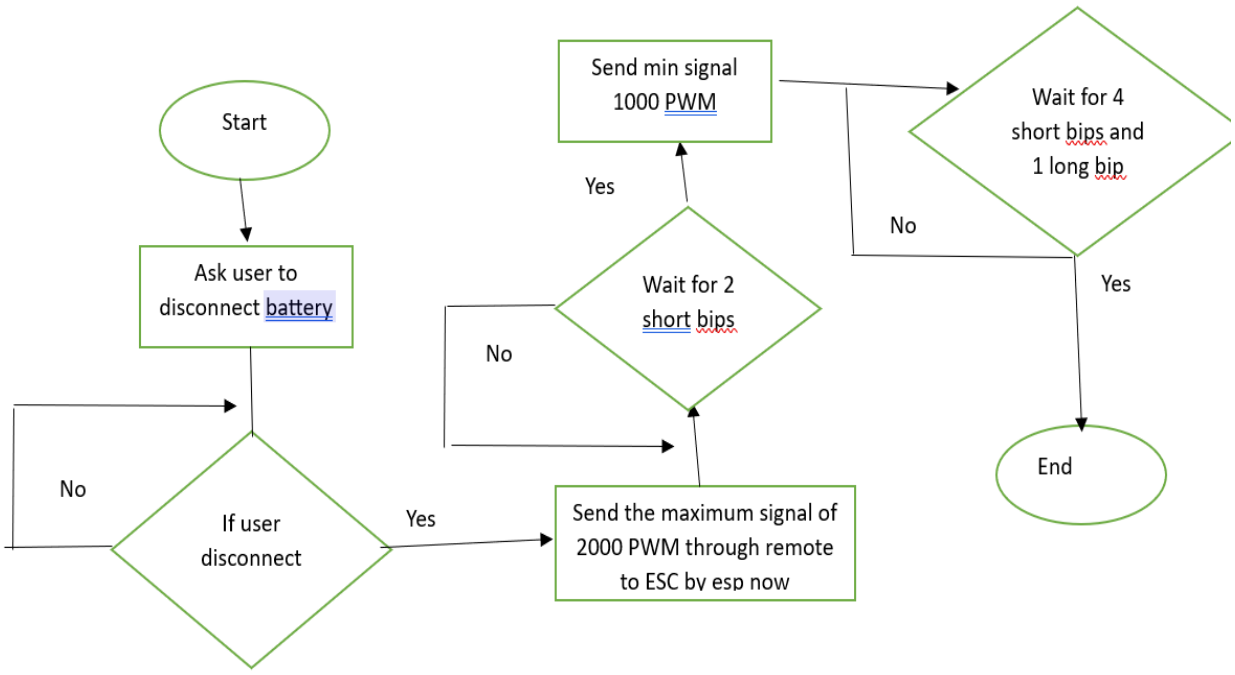



*Figure 17. Calibration flow chart*

**IMU MPU6050:** The code for the MPU6050 sensor, vital for motion measurement in a quadcopter, includes initializing and configuring the Inertial Measurement Unit (IMU). It starts with setting up I2C communication using specific pins and a 400KHz clock speed, then initializing the accelerometer and gyroscope components. The Digital Motion Processor (DMP) within the MPU6050 is prepared and enabled for angle calculations. Functions 'CalibrateAccel' and 'CalibrateGyro' ensure accurate sensor readings. The Get_MPUangle function clears the sensor's FIFO buffer, waits for sufficient data, and then retrieves and processes this data to calculate yaw, pitch, and roll angles in degrees. Similarly, the Get_accelgyro function reads acceleration and gyroscope data, converting them into degrees per second and g-forces, respectively, crucial for the quadcopter's stabilization and navigation. These steps are essential for accurate motion tracking in quadcopters.

**GPS:** Initializing the Serial2 communication interface for a GPS module, specifically the NEO-8M, starts by assigning PINs 16 (RXD2) and 17 (TXD2) for receiving and transmitting serial data. The baud rate, crucial for communication speed, is 9600, a NEO-8M GPS module standard. In the Init_Serial2() function, Serial2.begin is invoked with this baud rate and standard settings (SERIAL_8N1, indicating 8 data bits, no parity, and 1 stop bit). This function effectively prepares

Serial2 using the designated RX and TX pins and includes a line to ensure the code waits until Serial2 is operational. This setup is critical for facilitating data exchange between the microcontroller and the GPS module, enabling the transmission and reception of location data.

### 3.2.3. Testing and validation of the sensors

**IMU MPU6050 Python data visualization:** The source code script sets up real-time visualization of IMU and motor data from a controller, using Python and Matplotlib for plotting as an alternative to MATLAB, which lacks support on MacOS. It initializes a serial connection on a specified port. It uses deques to store angle data (X, Y, Z) and motor data (for four motors), each with a maximum length of 100 entries. The read_serial function reads incoming data from the serial port, splits it into angle and motor data, and appends these values to their respective deques and timestamps. Two separate functions, animate_angles and animate_motors, are defined for plotting angle and motor data over time, respectively. These functions clear their axes, plot data from the deques, and adjust axis labels for clarity. The script creates two figures for angles and motors, setting up animations for each with a refresh interval of 100 milliseconds. A separate thread is started to read serial data continuously, and the plots are displayed in real time, providing an effective solution for visualizing IMU data directly from the controller.

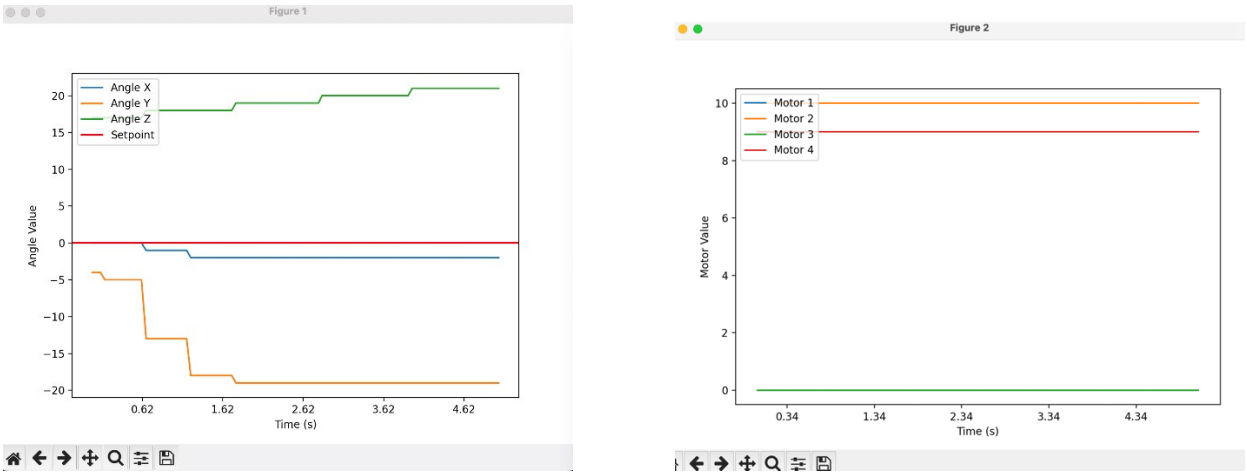



*Figure 18 & 19. Python IMU data visualization*

**GPS Google Sheet data visualization:** The setup involves configuring a microcontroller to connect to WiFi using specified network credentials (ssid and password) and preparing it to send GPS data to Google Sheets. A TinyGPSPlus object manages the GPS data, while script IDs for

Google Sheets are defined in GPS_GOOGLE_SCRIPT_ID. With a constant sendInterval of 0 milliseconds, the code establishes a WiFi connection in station mode, signaling connection progress on the Serial monitor. Once connected, the device's IP address is displayed. The recordGPStoGoogleSheet() function reads GPS data, checks its validity, and formats it into a string, gpsParam. This data, including latitude, longitude, speed, and other parameters, is sent to Google Sheets using the writeGoogleSheet function. This function constructs an HTTP request with the appropriate Google Script ID and sends the data through WiFi. The HTTP response confirms successful data transmission. This method, effective for GPS data, is not used for IMU data due to potential delays in WiFi signal and communication with Google Sheets, affecting real-time data accuracy.

The Google Apps Script, tailored for logging GPS data from an ESP32 device into a Google Sheets spreadsheet, engages with a specified document via its ID and uses 'Sheet1' for data recording. Upon activation, the script initially verifies if the incoming data from the ESP32 is undefined, responding with a message if it is. For valid data, it extracts key GPS parameters such as latitude, longitude, speed, number of satellites, altitude, GPS time, and date from the request. It then identifies the next row in the sheet to prevent data overwriting, logs the current date and time in the first column, and sequentially records the GPS data in the following columns. Post data entry, the script communicates back to the ESP32, confirming successful data recording with a "Status GPS Updated in Google Sheet" message. This setup enables streamlined and methodical tracking of GPS data.

```
var ss = SpreadsheetApp.openById('1EbX7DWZydwUP3WwBIiwQFX9eo6-H2ZzYoe4qLCoO4k0');
var sheet = ss.getSheetByName('Sheet1');

/**Get gps data from ESP32 and write */
function doGet(e){
  //get gps data from ESP32
  if (e.parameter == 'undefined') {
    return ContentService.createTextOutput("Received data is undefined");
  }
  //-----------------------------------------------------------------------------------
  var dateTime = new Date();
  latitude    = e.parameters.latitude;
  longitude   = e.parameters.longitude;
  speed       = e.parameters.speed;
  satellites  = e.parameters.satellites;
  altitude    = e.parameters.altitude;
  gps_time    = e.parameters.gps_time;
  gps_date    = e.parameters.gps_date;
  //Logger.log('latitude=' + latitude);
  //-----------------------------------------------------------------------------------
  var nextRow = sheet.getLastRow() + 1;
  sheet.getRange("A" + nextRow).setValue(dateTime);
  sheet.getRange("B" + nextRow).setValue(latitude);
  sheet.getRange("C" + nextRow).setValue(longitude);
  sheet.getRange("D" + nextRow).setValue(speed);
  sheet.getRange("E" + nextRow).setValue(satellites);
  sheet.getRange("F" + nextRow).setValue(altitude);
  sheet.getRange("G" + nextRow).setValue(gps_time);
  sheet.getRange("H" + nextRow).setValue(gps_date);
  //-----------------------------------------------------------------------------------

  //returns response back to ESP32
  return ContentService.createTextOutput("Status GPS Updated in Google Sheet");
  //-----------------------------------------------------------------------------------
}
```

*Figure 20. Google Apps Script for the GPS param*

The results from the GPS data are crucial for determining the device's geographical position and movement. They encompass details such as latitude, longitude, speed, the number of satellites, altitude, and GPS time and date. Latitude and longitude offer precise location coordinates, speed, and altitude, indicating the device's movement characteristics. The number of satellites reflects GPS signal reliability, and the GPS time and date add a time reference to each data set. This comprehensive information is vital for tracking the device's location and movements over time.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Date | Latitude | Longitude | Speed | Satellites | Altitude | GPS Time | GPS Date |
| 2 | 12/20/2023 | 10.784372 | 106.749218 | 1.39 | INVALID | INVALID | 13:10:09 | 12/20/2023 |
| 3 | 12/20/2023 | 10.784372 | 106.749218 | 1.39 | INVALID | INVALID | 13:10:09 | 12/20/2023 |
| 4 | 12/20/2023 | 10.784372 | 106.749218 | 1.39 | 4 | 7.5 | 13:10:09 | 12/20/2023 |
| 5 | 12/20/2023 | 10.784372 | 106.749218 | 1.39 | 4 | 7.5 | 13:10:09 | 12/20/2023 |
| 6 | 12/20/2023 | 10.784373 | 106.74921 | 0.5 | 4 | 7.5 | 13:10:18 | 12/20/2023 |
| 7 | 12/20/2023 | 10.784373 | 106.74921 | 0.5 | 4 | 7.5 | 13:10:18 | 12/20/2023 |
| 8 | 12/20/2023 | 10.784373 | 106.74921 | 0.5 | 4 | 7.5 | 13:11:00 | 12/20/2023 |
| 9 | 12/20/2023 | 10.784373 | 106.74921 | 0.5 | 4 | 7.5 | 13:11:00 | 12/20/2023 |
| 10 | 12/20/2023 | 10.784373 | 106.74921 | 0.5 | 4 | 7.5 | 13:11:00 | 12/20/2023 |
| 11 | 12/20/2023 | 10.784336 | 106.749249 | 0.37 | 4 | 7.5 | 13:12:07 | 12/20/2023 |
| 12 | 12/20/2023 | 10.784336 | 106.749249 | 0.37 | 4 | 7.5 | 13:12:07 | 12/20/2023 |
| 13 | 12/20/2023 | 10.784336 | 106.749249 | 0.37 | 5 | 11.6 | 13:12:07 | 12/20/2023 |
| 14 | 12/20/2023 | 10.784336 | 106.749249 | 0.37 | 5 | 11.6 | 13:12:07 | 12/20/2023 |
| 15 | 12/20/2023 | 10.784351 | 106.749203 | 0.44 | 5 | 11.6 | 13:12:47 | 12/20/2023 |
| 16 | 12/20/2023 | 10.784338 | 106.749208 | 0.87 | 5 | 11.6 | 13:13:00 | 12/20/2023 |
| 17 | | | | | | | | |

*Figure 21. Google Sheet displays results from the GPS param*

## 3.3. Work package 3: Control and validation of the drone

Work Package 3 has been a significant and productive phase, marked by dedicated efforts and strategic growth over the semester. The focus has been on enhancing control system model, ensuring safety compliance, and developing remote controller to control the drone. This period has seen the successful design and implementation of several key components, contributing to the overall project's progress and objectives.

### 3.3.1. PID model design

**Concept:** Our group followed the concept of Cascading PID to handle the external variable interaction which the single PID cannot carry on. For example, in the drone project, managing the complex dynamics of flight is a challenging task when trying to control the six degrees of freedom (DOF) – pitch, roll, yaw, and three-dimensional angle measurements. With a single PID controller, managing these six variables simultaneously is quite challenging.

A single PID controller, when applied to a drone, can struggle to balance the interactions between different axes of motion. For instance, adjusting the pitch might accidentally affect the roll or yaw, leading to unstable movements. This is because a single PID controller can find it hard to simultaneously to calculate between multiple axes with just one axis of variable kp,ki,kd.

Alternatively, a cascading PID model is considered more efficient way. In this setup, there are two layers of PID controllers. The first layer of PID controllers can be dedicated to managing the primary movements such as pitch, roll, and yaw based on the gyroscope readings. These controllers intermediately set to the setpoints for the second layer of controllers.

The second layer of PID controllers then focuses on calculate the singal sent to motors by precisely controlling the angles based on the setpoints provided by the first layer. This cascading approach allows for a more variables and responsive handling of the drone's motion due to the axis in X,Y and Z leading to more accurate and stable control.

The key advantage of this cascading PID system is the ability to handle multiple variables at the same time. By dividing the complex task of flight control into multiple controllers, each focusing on a specific aspect of the motion, the drone can achieve a more stable and responsive flight.

**Solution:**

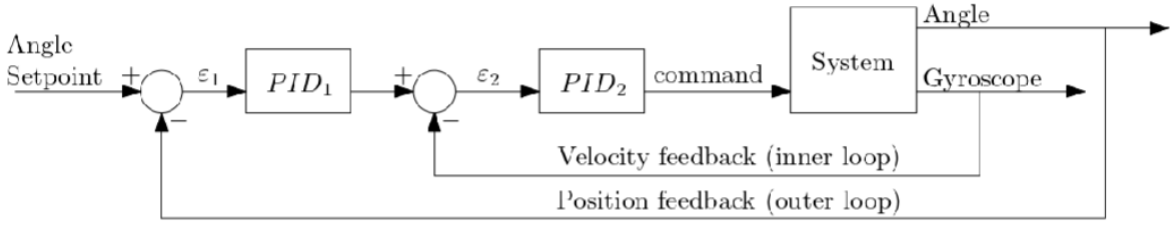



*Figure 22. Cascading PID model*

The figure above is cascading PID model that the quadcopter's system followed. Additionally, the process of the PID concept lead by 2 layer as represented in the figure.

In the first tier, the PID controllers are handling the drone's angular position. These controllers keep track the drone's current orientation and adjust it to match the desired input, in this case the input are the angle data read from IMU. They operate by comparing the actual angular position of the drone with the angle setpoint, and their outputs are not the final control signals but then set to setpoints for the drone's gyroscope. Eventually, this is the process for the primary loop of Cascading, which we call the outer loop.

The second tier of PID controllers then takes over, using the setpoints provided by the first tier. These controllers are responsible for controlling the drone's rotational velocity. They work by

adjusting the rate of change at which the drone rotates around each axis to compare with the gyroscope setpoints computed by the first tier. The outputs from these controllers directly being sent the drone's motors, to enable the speed of the quadcopter. The process being called is considered inner loop which is used to adjust the gyroscope variable from the setpoint exported from the outer- loop

Overall, this cascading PID model allows for a flexible control over the drone's movements. The first PID sets angular velocity output based on desired angle, and the second layer takes that velocity output as setpoint to achieve these targets. This structure is effective for stabilizing and maneuvering the drone with high precision.

### 3.3.2. PID Tunning method

**PID Tunning problem:** The decision to follow a cascading PID model for the drone's control system was a strategic one. This model, with its layered approach, provided a solid framework for controlling the drone's pitch, roll, and yaw movements effectively. However, the tuning process of this model failed due to two reasons.

Firstly, there were lack of distinct in tuning parameters for gyroscope and angle. To be more specific, the approach used only three terms - kp, ki, and kd - for both the gyroscope and angle measurements. This approach did not differentiate the term into six variables which are responsible for their corresponding function: three for the gyroscope readings and three for the angle control.

Furthermore, the tuning focused on the angle aspect of the drone, leading the gyroscope tuning altogether. As a result, the tunning of the drone just recorded the data from angle only while the gyroscope setpoint which was output from outer-loop was not static, it changed the setpoint overtime, the effectiveness of its implementation was inaccurate in defining the correct terms of the model, limiting the drone's ability to achieve optimal flight stability and responsiveness.

**PID Tunning Automatic Menu:** During the tunning operation, it is important to keep track of the test results for each change made to the settings. This usually means changing the settings (kp, ki, kd), testing the drone, and then doing it all over again - which can take a lot of time if you must reload the code each time you make a change. To make this process easier and faster, a special PID tuning menu was created. This menu lets user change the kp, ki, kd values quickly and

automatically, without the need to reload the code every time. This is a big time-saver because it allows for fast adjustments and testing, helping to find the best settings for the drone more efficiently. This way, tuning the drone to fly just right becomes much simpler and quicker. The figure below is the system tunning menu in which programmed in terminal based with nice GUI to prompt users about 3 tunning options: pitch, roll, yaw.

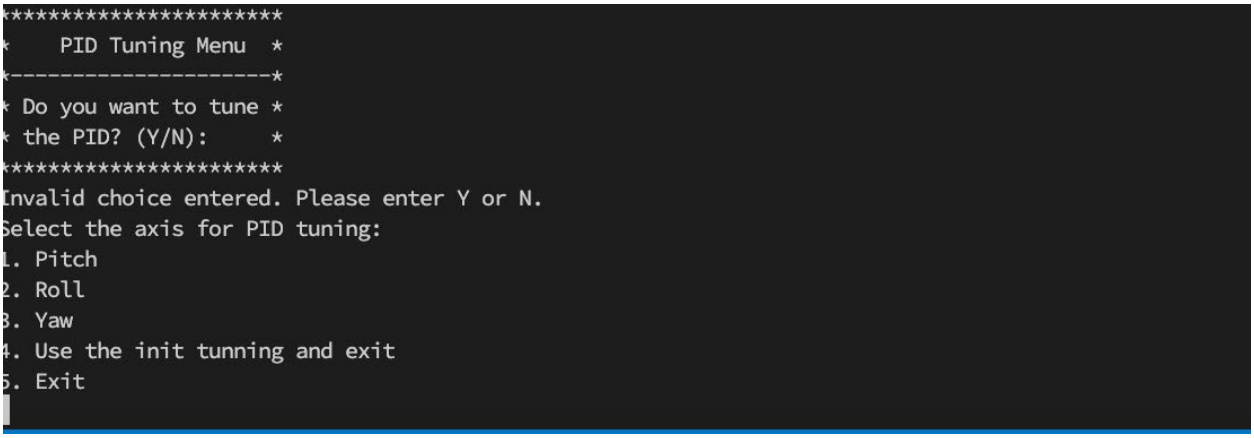



*Figure 23. PID tunning menu*

Specifically, when user needs to adjust the drone's settings for pitch, roll, or yaw using the PID menu, there is a helpful instruction showing up on the screen. This guide is easy to follow. It explains that if user want to change a setting, they just type a simple command. For example, if they want to adjust the 'Proportional' part of the setting, the user just type 'p' followed by the number they want to set it to.

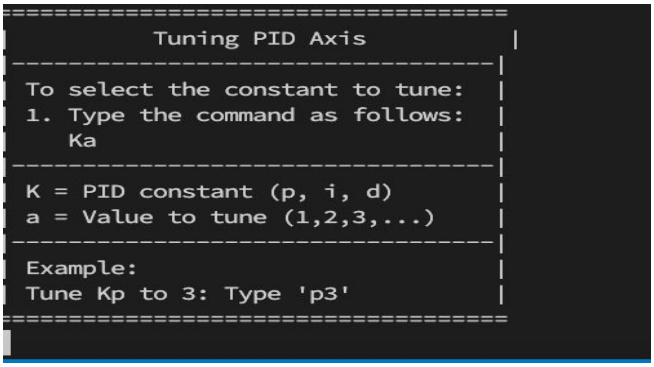



*Figure 24. PID tunning instructions*

The screen shows messages "To select the constant to tune: Type the command as follows: Ka", where 'K' stands for the PID part (‘P’, ‘I’ and ‘D’) and 'a' is the number they want to use. It also shows an example, if users want to change the Proportional constant to 3, they just type 'p3'.

This makes it straightforward for the users to change the drone's settings without getting confused. It is giving clear, step-by-step instructions, which makes adjusting the drone's flight controls much easier.

### 3.3.3. Safety implementation and stop button

**Problem:** In the process of tuning the quadcopter, safety implementation is the priority, especially considering the potential hazards caused by the BLDC motors and their propellers. These motors can spin at very high speeds, thus the attached propellers could cut to the hands or other body parts.

**Solution:** to minimalize these risks, several safety measures are practically applied. For instance, tuning process is placed in a controlled environment the space is large enough for the drone to freely move. Protective propeller guards might be used to shield the propellers when rotating to the body part. Unfortunately, as mentioned in the WP1, we could not access to the large scale 3D printer so the propeller guards were not able to be printed. Alternatively, the emergency stop button and power switch were created to allow for the immediate shutdown of the motors in case of an accident occurs.

**Test and validate emergency stop button:** to begin with the emergency stop button, the process of validating the emergency stop button for the quadcopter involves several key steps to ensure it works effectively. First, the feature of the button itself is tested using the “esp32_button.h” library. This is important to confirm that the button responds correctly when pressed. Next, a debounce function is added to the setup. The debounce function is important because it helps prevent extra triggers due to minor vibrations or accidental touches - it makes sure that the button only activates when it's genuinely pressed. After setting up the button and its debounce function, the next step is to program it to send a 'stop' signal from the remote control to the drone using ESP-NOW communication. This signal tells the drone to immediately set all of its motor speeds to zero. This is a critical safety feature, as it ensures that in any emergency, pressing the button will instantly stop the drone, preventing accidents or injuries. The performance of this button is validated safely when the team was tunning drone PID.

**Test and validate power switch button on drone:** when calibrating the esp32, it is possible that the power supply from computer can be conflict to the power supply from battery. As the result, it could cause the short circuit in the system. Our team had experience this kind of unexpected accident before. To solve this problem, the power switch had been designed to connect from the PDB to the ESP32. As consequence, when the circuit system seems to be short circuit, turn off the switch to avoid the components to be burnt.

### 3.3.4. Establish a reliable remote control of the drone.

**Remote controller functions:** to fully control the drone, the remote controller plays important role that can do the fundamental features of the quadcopter such as: roll, pitch, yaw. The remote controller designed for the drone is composed with components including potentiometer, buttons, joysticks to control over the drone's movements. First and foremost, is a potentiometer, which acts as a thrust control to hover the drone. By adjusting the potentiometer, the user can control the vertical lift of the drone by sending the signal read from ADC1 port of the microcontroller, and by the act of PID, the quadcopter can stably hover. For directional control, a joystick is integrated into the remote. This joystick is responsible for controlling the pitch and roll of the drone, allowing for forward, backward, and side-to-side movements. Additionally, the controller includes two buttons, each assigned to control the yaw movement. One button is designated for yawing left, and the other for yawing right. These buttons enable the drone to rotate left or right, providing complete 360-degree directional control. Together, these components – the potentiometer, joystick, and buttons – combine to offer a comprehensive and user-friendly interface for maneuvering the drone with precision and ease.

*Figure 25. Remote controller*

**Behind the scene of the controller functions:** The movement concept of a quadcopter is fundamentally based on the variation in speed of each of its four motors. Unlike vehicles with fixed structures for movement, a quadcopter has ability to move in multiple directions are driven by altering the rotational speeds of these motors independently. However, to launch more complex movements like tilting forward (pitching) or sideways (rolling), or rotating on its axis (yawing), the speeds of the motors are adjusted based on its position and direction of rotation.

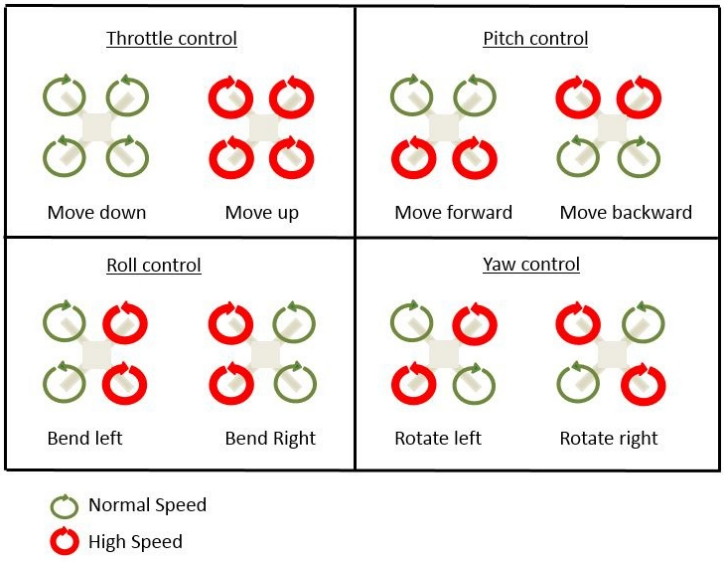


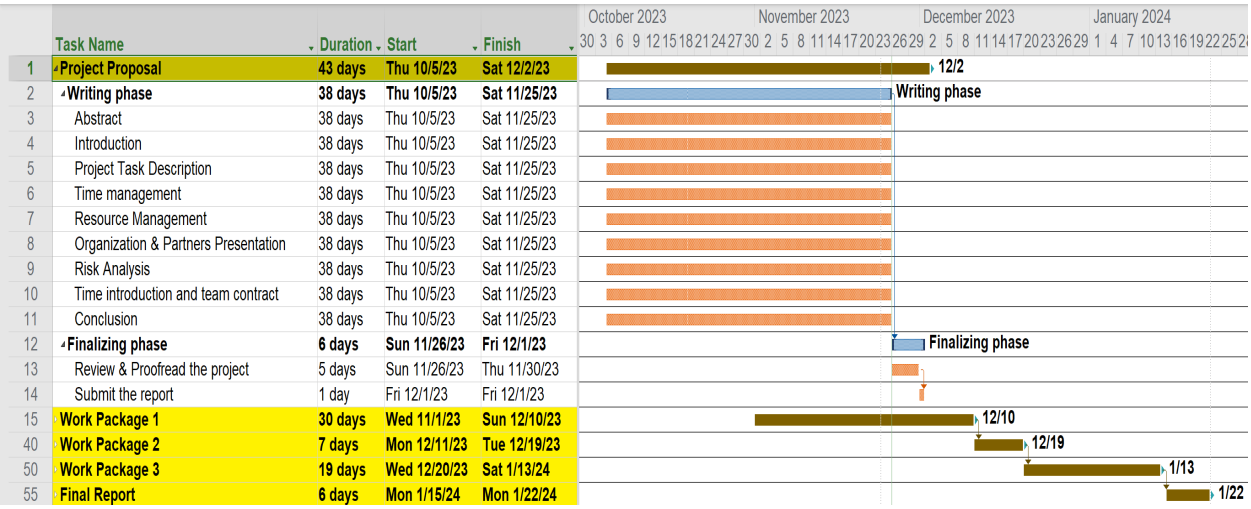

*Figure 26. Motors map to control the drone*

When the quadcopter needs to pitch forward, the speed of the rear motors increases, while the front motors slow down, tilting the quadcopter forward. For rolling, one side's motors speed up, and the opposite side's motors slow down, causing the quadcopter to lean and move sideways. Yawing is achieved by creating a differential in rotational direction or speed between the motors on opposite sides.

**PID role in pitch, roll, yaw the drone:** As mentioned above, the potentiometer used to send signal read from ADC and then converted to motor speed then sent to the drone to adjust the speed of 4 motors at the same time. Otherwise, the drone's ability to pitch and roll is controlled through the action of the PID system. For both pitching and rolling movement, the process begins by sending the signal from joystick module in the controller to the drone, then this received signal will correspondingly set an angle setpoint to achieve either in its pitch axis or its side-to-side roll axis. Once this setpoint is established, the PID system comes into play. It compares the drone's current angle with the setpoint and calculates the necessary adjustments. These adjustments are then transmitted into changes in motor speeds. Each motor receives specific speed instructions from the PID controller, ensuring that the drone moves to the desired pitch or roll angle. This ongoing computation and adjustment by the PID system enable the drone to maintain balance, stability, and precise control during flight, whether it's tilting forward, leaning backward, or rolling to either side. Eventually, with yaw button, there are two states for each left or right is sent from the remote, and for every time the drone receive these signals, the setpoint will be deducted or added correspondingly the signal it took. For example, if the users want the drone to yaw left, the setpoint of PID will be minus 10 degree each time to let the drone to yaw left and vice versa.

# 4. Time Management

## 4.1. The Project Proposal Stage



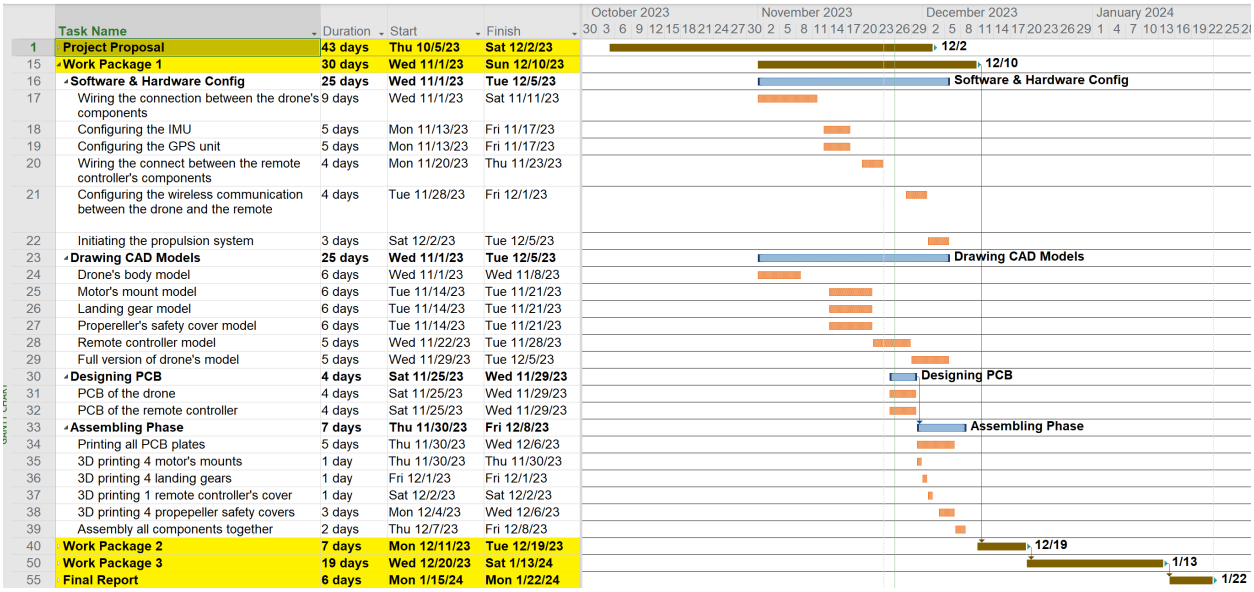
| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | Project Proposal | 43 days | Thu 10/5/23 | Sat 12/2/23 |
| 2 | Writing phase | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 3 | Abstract | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 4 | Introduction | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 5 | Project Task Description | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 6 | Time management | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 7 | Resource Management | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 8 | Organization & Partners Presentation | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 9 | Risk Analysis | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 10 | Time introduction and team contract | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 11 | Conclusion | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 12 | Finalizing phase | 6 days | Sun 11/26/23 | Fri 12/1/23 |
| 13 | Review & Proofread the project | 5 days | Sun 11/26/23 | Thu 11/30/23 |
| 14 | Submit the report | 1 day | Fri 12/1/23 | Fri 12/1/23 |
| 15 | Work Package 1 | 30 days | Wed 11/1/23 | Sun 12/10/23 |
| 40 | Work Package 2 | 7 days | Mon 12/11/23 | Tue 12/19/23 |
| 50 | Work Package 3 | 19 days | Wed 12/20/23 | Sat 1/13/24 |
| 55 | Final Report | 6 days | Mon 1/15/24 | Mon 1/22/24 |



*Figure 27. Gantt chart of the project proposal stage.*

The Project Proposal process started on October 5th and is planned to end by December 2nd, but the team intends to finish writing by November 25th. After the project description was released, tasks were allocated among all team members, with the Proposal comprising 9 specific tasks spread evenly across the 6 members. Key aspects include Project Task Descriptions and Time Management. The project is divided into 3 Work Packages, each handled by a pair of team members. The Task Descriptions guide research and clarify requirements, vital for effective work on the Work Packages post-Proposal. Time Management is crucial, outlining a task timeline for better workflow understanding and deadline adherence. The Finalizing phase, lasting 6 days from November 26th to December 1st, includes an online meeting on November 30th for thorough Proposal review before submission.

## 4.2. The WP1 Stage



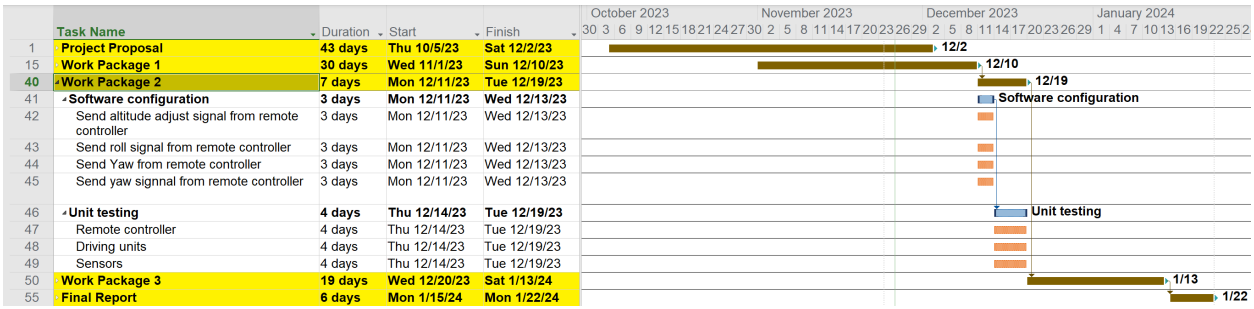
| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Project Proposal** | **43 days** | **Thu 10/5/23** | **Sat 12/2/23** |
| 15 | **Work Package 1** | **30 days** | **Wed 11/1/23** | **Sun 12/10/23** |
| 16 | **Software & Hardware Config** | **25 days** | **Wed 11/1/23** | **Tue 12/5/23** |
| 17 | Wiring the connection between the drone's components | 9 days | Wed 11/1/23 | Sat 11/11/23 |
| 18 | Configuring the IMU | 5 days | Mon 11/13/23 | Fri 11/17/23 |
| 19 | Configuring the GPS unit | 5 days | Mon 11/13/23 | Fri 11/17/23 |
| 20 | Wiring the connect between the remote controller's components | 4 days | Mon 11/20/23 | Thu 11/23/23 |
| 21 | Configuring the wireless communication between the drone and the remote | 4 days | Tue 11/28/23 | Fri 12/1/23 |
| 22 | Initiating the propulsion system | 3 days | Sat 12/2/23 | Tue 12/5/23 |
| 23 | **Drawing CAD Models** | **25 days** | **Wed 11/1/23** | **Tue 12/5/23** |
| 24 | Drone's body model | 6 days | Wed 11/1/23 | Wed 11/8/23 |
| 25 | Motor's mount model | 6 days | Tue 11/14/23 | Tue 11/21/23 |
| 26 | Landing gear model | 6 days | Tue 11/14/23 | Tue 11/21/23 |
| 27 | Propereller's safety cover model | 6 days | Tue 11/14/23 | Tue 11/21/23 |
| 28 | Remote controller model | 5 days | Wed 11/22/23 | Tue 11/28/23 |
| 29 | Full version of drone's model | 5 days | Wed 11/29/23 | Tue 12/5/23 |
| 30 | **Designing PCB** | **4 days** | **Sat 11/25/23** | **Wed 11/29/23** |
| 31 | PCB of the drone | 4 days | Sat 11/25/23 | Wed 11/29/23 |
| 32 | PCB of the remote controller | 4 days | Sat 11/25/23 | Wed 11/29/23 |
| 33 | **Assembling Phase** | **7 days** | **Thu 11/30/23** | **Fri 12/8/23** |
| 34 | Printing all PCB plates | 5 days | Thu 11/30/23 | Wed 12/6/23 |
| 35 | 3D printing 4 motor's mounts | 1 day | Thu 11/30/23 | Thu 11/30/23 |
| 36 | 3D printing 4 landing gears | 1 day | Fri 12/1/23 | Fri 12/1/23 |
| 37 | 3D printing 1 remote controller's cover | 1 day | Sat 12/2/23 | Sat 12/2/23 |
| 38 | 3D printing 4 propepeller safety covers | 3 days | Mon 12/4/23 | Wed 12/6/23 |
| 39 | Assembly all components together | 2 days | Thu 12/7/23 | Fri 12/8/23 |
| 40 | **Work Package 2** | **7 days** | **Mon 12/11/23** | **Tue 12/19/23** |
| 50 | **Work Package 3** | **19 days** | **Wed 12/20/23** | **Sat 1/13/24** |
| 55 | **Final Report** | **6 days** | **Mon 1/15/24** | **Mon 1/22/24** |



*Figure 28. Gantt chart of the work package 1.*

Work Package One of the project, initiated during the proposal stage, encompasses four main subtasks: Software and Hardware Configuration, Drawing CAD Model, Designing PCB, and Assembly. The aim is to complete these by December 10th, 2023, with a primary focus on software and hardware, specifically microcontroller integration, and sensor and module configuration. Software tasks include configuring the Inertial Measurement Unit (IMU) and GPS device, and establishing wireless connectivity. Hardware tasks involve interconnecting components, creating detailed CAD models for various drone parts, designing printed circuit boards, incorporating 3D printing in fabrication, and assembling the complete drone system. Team members assigned to these tasks have relevant experience or the ability to conduct necessary research. Flexibility and teamwork are emphasized, especially if tasks expand beyond their original scope, requiring collective effort to maintain project progression. This approach highlights the importance of adaptability, teamwork, and individual responsibility within the project.

## 4.3. The WP2 Stage



| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Project Proposal** | **43 days** | **Thu 10/5/23** | **Sat 12/2/23** |
| 15 | **Work Package 1** | **30 days** | **Wed 11/1/23** | **Sun 12/10/23** |
| 40 | **Work Package 2** | **7 days** | **Mon 12/11/23** | **Tue 12/19/23** |
| 41 | **Software configuration** | **3 days** | **Mon 12/11/23** | **Wed 12/13/23** |
| 42 | Send altitude adjust signal from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 43 | Send roll signal from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 44 | Send Yaw from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 45 | Send yaw signnal from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 46 | **Unit testing** | **4 days** | **Thu 12/14/23** | **Tue 12/19/23** |
| 47 | Remote controller | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 48 | Driving units | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 49 | Sensors | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 50 | **Work Package 3** | **19 days** | **Wed 12/20/23** | **Sat 1/13/24** |
| 55 | **Final Report** | **6 days** | **Mon 1/15/24** | **Mon 1/22/24** |

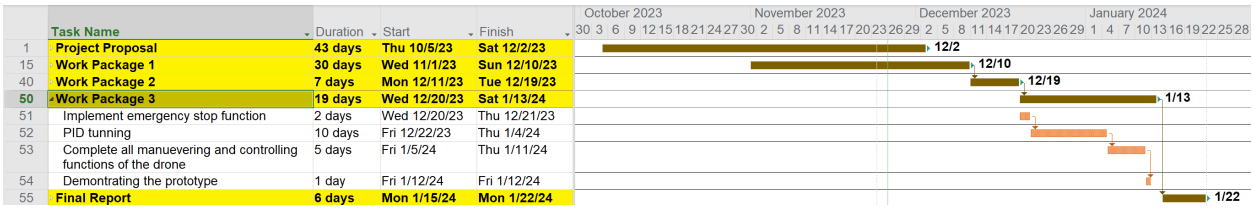


*Figure 29. Gantt chart of the work package 2.*

Work Package 2 (WP2) is set to begin immediately after the completion of Work Package 1 (WP1), running from December 11 to December 19, 2023. WP2 involves several critical milestones, including designing the remote control, calibrating and operating the driving unit, and testing and validating sensors, with each milestone needing completion before moving to the next. The focus of WP2 is primarily on software, which involves tasks like implementing code for transmitting various angle signals (attitude, roll, yaw, pitch) through the remote control.

Before assembly, it's mandatory to perform unit testing of each software component. This includes confirming the remote control's functionality, verifying the driving unit's operation, and validating the sensors for accurate data collection. To maximize efficiency, team members are assigned specific tasks based on their expertise and the demands of each task, as detailed in the Task Explanation section. This strategic approach ensures that qualified individuals handle every aspect of WP2, promoting the project's successful progression.

## 4.4. The WP3 Stage



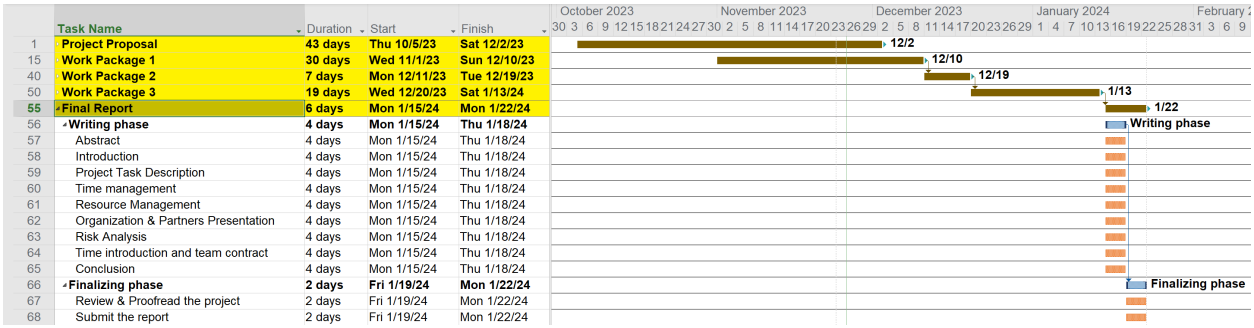
| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Project Proposal** | **43 days** | **Thu 10/5/23** | **Sat 12/2/23** |
| 15 | **Work Package 1** | **30 days** | **Wed 11/1/23** | **Sun 12/10/23** |
| 40 | **Work Package 2** | **7 days** | **Mon 12/11/23** | **Tue 12/19/23** |
| 50 | **Work Package 3** | **19 days** | **Wed 12/20/23** | **Sat 1/13/24** |
| 51 | Implement emergency stop function | 2 days | Wed 12/20/23 | Thu 12/21/23 |
| 52 | PID tunning | 10 days | Fri 12/22/23 | Thu 1/4/24 |
| 53 | Complete all manuevering and controlling functions of the drone | 5 days | Fri 1/5/24 | Thu 1/11/24 |
| 54 | Demontrating the prototype | 1 day | Fri 1/12/24 | Fri 1/12/24 |
| 55 | **Final Report** | **6 days** | **Mon 1/15/24** | **Mon 1/22/24** |



*Figure 30. Gantt chart of the work package 3.*

Work Package 3 (WP3) of our project is set to commence on December 20th, immediately following the completion of Work Package 2 (WP2), and is planned to last until January 13th, giving us a 19-day window for completion. This phase is organized into four critical stages. The first stage involves implementing an emergency stop function in the drone's control system, a vital safety feature for halting operations swiftly in unexpected situations. Next, we focus on the meticulous tuning of the Proportional-Integral-Derivative (PID) controller, which is crucial for achieving precise and stable control of the drone. The third stage is dedicated to finalizing all maneuvering and controlling functions of the drone, addressing any potential issues such as wiring errors, assembly adaptations, or challenges in component integration. The final stage of WP3 is a comprehensive demonstration of the prototype, showcasing the successful integration of individual components and the drone's ability to perform controlled maneuvers. This demonstration will also be presented in front of an audience. WP3 plays a pivotal role in our project, focusing primarily on the control and validation of the drone. The overarching objective is to seamlessly integrate various components into a functional and reliable drone system, encompassing a finely tuned PID controller, robust safety measures like the emergency stop function, and ensuring precise control for testing and validation. The success of WP3 is crucial for the overall success of the project.

## 4.5. The Final Report



| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | **Project Proposal** | **43 days** | **Thu 10/5/23** | **Sat 12/2/23** |
| 15 | **Work Package 1** | **30 days** | **Wed 11/1/23** | **Sun 12/10/23** |
| 40 | **Work Package 2** | **7 days** | **Mon 12/11/23** | **Tue 12/19/23** |
| 50 | **Work Package 3** | **19 days** | **Wed 12/20/23** | **Sat 1/13/24** |
| 55 | **Final Report** | **6 days** | **Mon 1/15/24** | **Mon 1/22/24** |
| 56 | **Writing phase** | **4 days** | **Mon 1/15/24** | **Thu 1/18/24** |
| 57 | Abstract | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 58 | Introduction | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 59 | Project Task Description | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 60 | Time management | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 61 | Resource Management | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 62 | Organization & Partners Presentation | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 63 | Risk Analysis | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 64 | Time introduction and team contract | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 65 | Conclusion | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 66 | **Finalizing phase** | **2 days** | **Fri 1/19/24** | **Mon 1/22/24** |
| 67 | Review & Proofread the project | 2 days | Fri 1/19/24 | Mon 1/22/24 |
| 68 | Submit the report | 2 days | Fri 1/19/24 | Mon 1/22/24 |



*Figure 31. Gantt chart of the final report stage.*

The final phase of the project, while similar to the initial proposal phase, requires more detailed information and system specifications. It has a deadline set for January 21st, starting from January 15th, spanning six days. This phase is divided into two milestones: the Writing phase, taking four days, and the Finalizing phase, taking two days. The Writing phase involves creating various sections such as the Abstract, Introduction, Project Task Description, Time Management, Resource Management, Organization & Partners Presentation, Risk Analysis, and Conclusion, paralleling the sections in the Proposal. The main goal of this report is to compile and summarize all the project data. It highlights the ability to collaboratively design and build a functional product prototype, emphasizing efficient time management both individually and as a team. Participants are encouraged to reflect on their technical and non-technical learning experiences, drawing on their foundational engineering knowledge. The completed report is then presented to academics, industry experts, and peers, showcasing the wide range of skills developed throughout the project.

# 5. Resources Management:

Resource management in drone construction requires a comprehensive strategy that addresses material and component selection, design, and manufacture. Integrating these elements into the manufacturing process require thorough planning for each part listed in the Bill of Materials (BOM) within the designated work packages. It is crucial to ensure the drone's safe and efficient

operation. Careful coordination and management of each aspect are imperative to achieve a final product aligned with the set specifications and performance goals.

## 5.1. The WP1 Bill of Materials

| Item Name | Description | Quantity | Total Price (VND) |
|---|---|---|---|
| ESP32-wroom | A microcontroller module for this project, which supports Wi-Fi protocol and able to connect and communicate over Wi-Fi networks. It is one of the low-cost modules and widely used in various Internet-Of-Things applications. | 5 | 395,000 |
| Breadboard | Small breadboard to form the circuit. | 1 | 20,000 |
| Wood beams and plates | Wood beams and plates. | 8 | 40,000 |
| Lipo_Battery for Drone | Power supply for a drone. | 1 | 473,000 |
| Lipo_Battery for controller | Power supply for remote control. | 1 | 23,000 |
| ESC | An electronic Speed Controller (ESC) is an important component empowered electrical RC model such as drone, car and boats. It serves as an intersection between power source and motor to control speed and direction of the motor based on remote control system from user. | 8 | 579,280 |
| PDB | A Power Distribution Board is an essential component in many electronic systems. In this drone making project, it plays a role to distribute electrical power from high voltage battery to supply into 4 different ESC components. | 2 | 136,000 |

| | | | |
|---|---|---|---|
| Wood screws | A type of fastener designed specifically for use in wooden materials. It has a threaded shaft with a pointed tip at one end and a head at the other. | 100/set | 30,000 |
| Mica plastic plate | Designed to serve as a cover for the top section of the drone, providing a protective enclosure where the LiPo (Lithium Polymer) battery of the drone can be securely placed. | 1 | 40,000 |
| Lipo Battery charger | Charger for drone and controller battery. | 1 | 359,000 |

*1Table 1. Bill of materials of the work package 1.*

During the experimentation phase of work package 1, four ESCs and one ESP32 were burned, increasing the project's budget to accommodate the unforeseen component failures.

## 5.2. The WP2 Bill of Materials

| Item Name | Description | Quantity | Total Price (VND) |
|---|---|---|---|
| PCB printing | PCB production refers to a process of making Printed Circuit Boards, which are used mechanically support and electrically connect components after finished the schematic design from EasyEDA online open platform. | 2 | 500,000 |
| Red button | Two buttons for controlling yaw rate angle. | 4 | 32,000 |
| Joystick | The joystick controls the roll and pitch rate of the drone. | 2 | 34,000 |
| Switch button | Power the remote control and the drone. | 2 | 8,000 |
| PLA material for 3d printing | It is known as Polylactic Acid, widely used for 3d printing. It is an environmentally friendly material and offers high quality 3d printing in the market. | 1 | 310,000 |
| USB_C cable | Transmission wire between the ESP and laptop. | 2 | 134,000 |

| | | | |
|---|---|---|---|
| Brushless Motors | A brushless motor to utilize onto the frame of drone. | 6 | 720,000 |
| Potentiometer | Control the thrust level of the drone. | 3 | 7,500 |
| Prototype PCB | sized at 6x8 centimeters, facilitates the connection of electronic components through wires and pins. It serves as a valuable tool during the early stages of electronics development, enabling quick assembly and testing of circuits. | 2 | 100,000 |
| Jumper wires and solid wires | Jumper wires and solid wires are two distinct types of wires used for connecting components in electronic circuits. | 2 | 150,000 |
| Propellers | The drone's propeller | 8 | 76,000 |

*Table 2. Bill of materials of the work package 2.*

During Work Package 2, a setback emerged when the Printed Circuit Board (PCB) malfunctioned due to a wiring schematic error. To resolve this, a prototype PCB was used for system wiring, enabling necessary electrical connections, and allowing progress despite the PCB issue. A crash occurred in the PID testing phase, damaging two motors and all four propellers. This incident necessitated a reevaluation and adjustment of the code to avoid future issues, highlighting the critical role of thorough testing and validation in ensuring project success.

## 5.3. The WP3 Bill of Materials

| Item Name | Description | Quantity | Total Price (VND) |
|---|---|---|---|
| MPU6050 | Low-cost inertial measurement unit (IMU) used in various applications in electronic and robotic. It provides 3-axis from both accelerometer and gyroscope that help to track 3d environments and motion process on object. | 2 | 55,000 |
| Red button | A red button to stop the drone. | 1 | 8,000 |

| | | | |
|---|---|---|---|
| GPS NEO8M with ceramic antenna | Sensor sends the latitude, longitude and altitude of the drone. | 1 | 242,000 |
| Tripod | Tripod for PID tunning-test | 1 | 800,000 |

*Table 3. Bill of materials of the work package 3.*

In Work Package 3, an MPU6050 was damaged, but no extra components are needed for this project phase.

### 5.4. The Total Bill of Materials

| No. Of Work Package | Items | Total cost |
|---|---|---|
| **Work Package 1** | 29 | 2,095,280 |
| **Work Package 2** | 34 | 2,071,500 |
| **Work Package 3** | 5 | 1,105,000 |
| **All of Work Package** | **68** | **5,271,780** |

*Table 4. Total cost of the project.*

The final project budget has risen from the initial proposal, influenced by damages from burns and crashes in the testing phase. The final budget has doubled compared to the initial proposal, presenting a challenge to develop a professional-grade quadcopter compared to market-available drones. This highlights the necessity for efficient resource use and cost-effective approaches to achieve project goals within budget limits.

## 6. Risk Assessment

The detailed risk assessment in a project aims to methodically identify, analyze, and manage potential challenges that might affect the project's successful completion. This process involves assessing the likelihood of risks and their possible effects on the project's goals. It is a preemptive measure to foresee and reduce issues that may hinder the project's flow or results. In this specific project, the risk assessment process is divided into three primary risk categories: internal, external, and safety risks. Each category encompasses various potential obstacles the project might face.

| | Likelihood |
|---|---|

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **Risk probability** | Rare | Unlikely | Possible | Likely | Certain |
| **Risk impact** | Insignificant | Minor | Moderate | Major | Catastrophic |

*Table 5. Risk assessment likelihood.*

| **Overall Risk Scale** | | |
|---|---|---|
| Below 5 | From 5 – 7 | Over 8 |
| Minor | Moderate | Require attention and risk mitigation |

*Table 6. Risk assessment scale.*

## 6.1. Internal Risks

| Risk Description | Risk Probability | Risk Impact | Overall Score | Mitigation |
|---|---|---|---|---|
| Budget overruns | 5 | 4 | 15 | Develop a detailed project cost estimation and budget preparation |
| Damaged components | 4 | 5 | 20 | Frequently conduct system check and have a backup components. |
| Poor testing plan | 4 | 5 | 20 | Develop a detailed testing plan and regular review testing outcomes. |
| Poor communication | 3 | 4 | 12 | Frequently communicate and check on the progress of each member. |
| Teammates are unable to work due to external factors. | 5 | 5 | 20 | Frequently communicate and check on the progress of each member to plan beforehand the resolution. |
| Poor planning led to tasks delay. | 4 | 4 | 16 | Use project management software to streamline project tasks (Trello, Notion). |
| Computer or equipment damaged. | 4 | 5 | 20 | Transport the computer or equipment in a cushioned bag. Avoid liquid near the electrical components. |
| Software glitches or accidentally deleted files. | 3 | 5 | 15 | Make sure to save the project regularly in the secure folder or create backups. |

| | | | | |
|---|---|---|---|---|
| Component delivery delay. | 2 | 4 | 8 | Keep track with the component status and prepare backup plant. |
| Low individual performance. | 3 | 4 | 12 | Document performance history and provide feedback. Encourage members to ask for help. |
| Disagreement and conflict. | 4 | 3 | 12 | Address disagreement and come to a mutual understanding. |

*Table 7. Internal risks.*

## 6.2. External Risk

| Risk Description | Risk probability | Risk impact | Overall score | Mitigation |
|---|---|---|---|---|
| Legal Concerns and Regulatory Compliance. | 3 | 3 | 9 | Flight Restrictions, Data Protection, and Insurance Requirements must be included. |
| Environment factors. | 3 | 4 | 12 | Wind conditions, Altitude, Obstacles, and Terrain may significantly impact the drone testing process. |
| Public Perception and Acceptance. | 3 | 3 | 9 | Negative public opinion or hostility to drone testing might affect public policy, limit usage, or damage outside people or buildings. |

*Table 8. External risks.*

## 6.3. Safety Risk

| Risk description | Risk probability | Risk impact | Overall score | Mitigation |
|---|---|---|---|---|
| Injuries during drone testing | 4 | 5 | 12 | Implement emergency stop button, wear protective gear, ensure |

| | | | | |
|---|---|---|---|---|
| | | | | supervised testing in controlled environments. |
| Electrical shock | 3 | 2 | 6 | Frequently check the wire connection. Always wear insulating gloves to avoid being electrocuted. |
| Injuries during assembly | 3 | 4 | 6 | Wear protective gear and check for tool malfunction. |

*Table 9. Safety risks*

# 7. Conclusion

**Work done overall:** In summary, the drone project was a big success in both making the parts and writing the software. For the hardware parts, a lot of work went into designing the drone. The motor mounts, landing gear, and a special case for the remote controller were all made using a 3D printer. Also, the design of the PCB, which is considered the brain of the drone, was a big step. On the software side, there was a lot of important work that needed to be handled. Setting up the IMU, GPS module, and ESP-Now was key for the drone to work right. Also, making a button that sends data and setting up a special kind of control system called cascading PID helped make the drone easy to control and stable in the air. All together, these efforts in building and programming made a drone that works well and is easy to use. It shows how well the team did in combining different tech skills to make it effective.

**Future development:** However, despite its perfect system, the drone had problems flying steadily because we didn't tune the Cascading PID control system the right way. The correct way to do this was explained in WP3, a guide we should have followed. Tuning the PID system is the last step needed to make the drone work properly. By following the correct tuning procedures as mentioned in WP3, we anticipate successfully completing the PID tuning process. This adjustment is expected to resolve the stability challenges, enabling the quadcopter to fly efficiently and effectively, thereby fulfilling its intended design objectives.

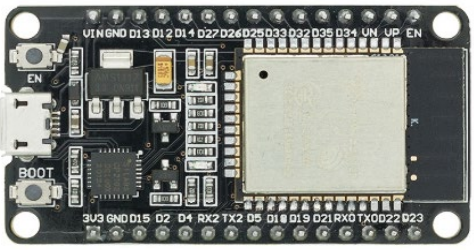
# 8. References

[1] J. Yoon and J. Doh, "Optimal PID control for hovering stabilization of quadcopter using long short term memory," *Advanced Engineering Informatics,* vol. 53, p. 101679, 2022/08/01/ 2022, doi: https://doi.org/10.1016/j.aei.2022.101679.

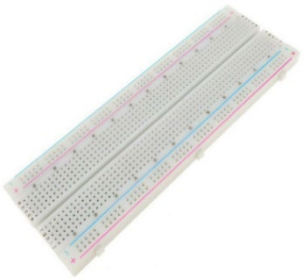
[2] M. Kan, S. Okamoto, and J. H. Lee, "Development of drone capable of autonomous flight using GPS," in *Proceedings of the international multi conference of engineers and computer scientists*, 2018, vol. 2.

[3] D. Eridani, A. F. Rochim, and F. N. Cesara, "Comparative performance study of ESP-NOW, Wi-Fi, bluetooth protocols based on range, transmission speed, latency, energy usage and barrier resistance," in 2021 international seminar on application for technology of information and communication (iSemantic), 2021: IEEE, pp. 322-328.

# 9. Appendix




*Appendix 1. ESP32. Link*

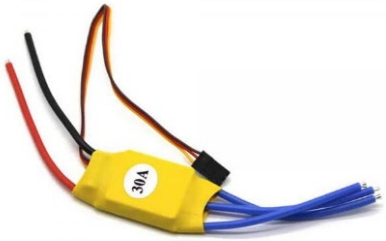


*Appendix 2. Breadboard. Link*

*Appendix 3. Wood beams and plates. Link*


*Appendix 4. Lipo battery for controller. Link*

*Appendix 5. Lipo battery for drone. Link*




*Appendix 6. ESC. Link*

*Appendix 7. PDB. Link*


*Appendix 8. Lipo Battery Charger. Link*


*Appendix 9. Red button. Link*


*Appendix 10. Joystick. Link*

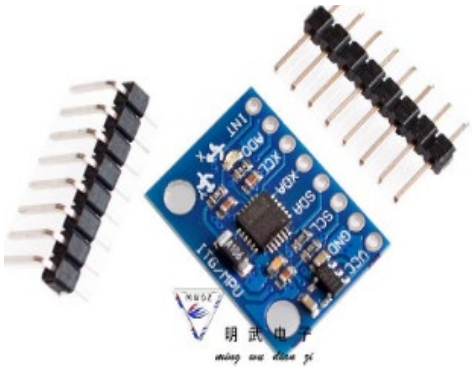

*Appendix 11. Power switch button. Link*


*Appendix 12. USB_C cable. Link*


*Appendix 13. Motors. Link*

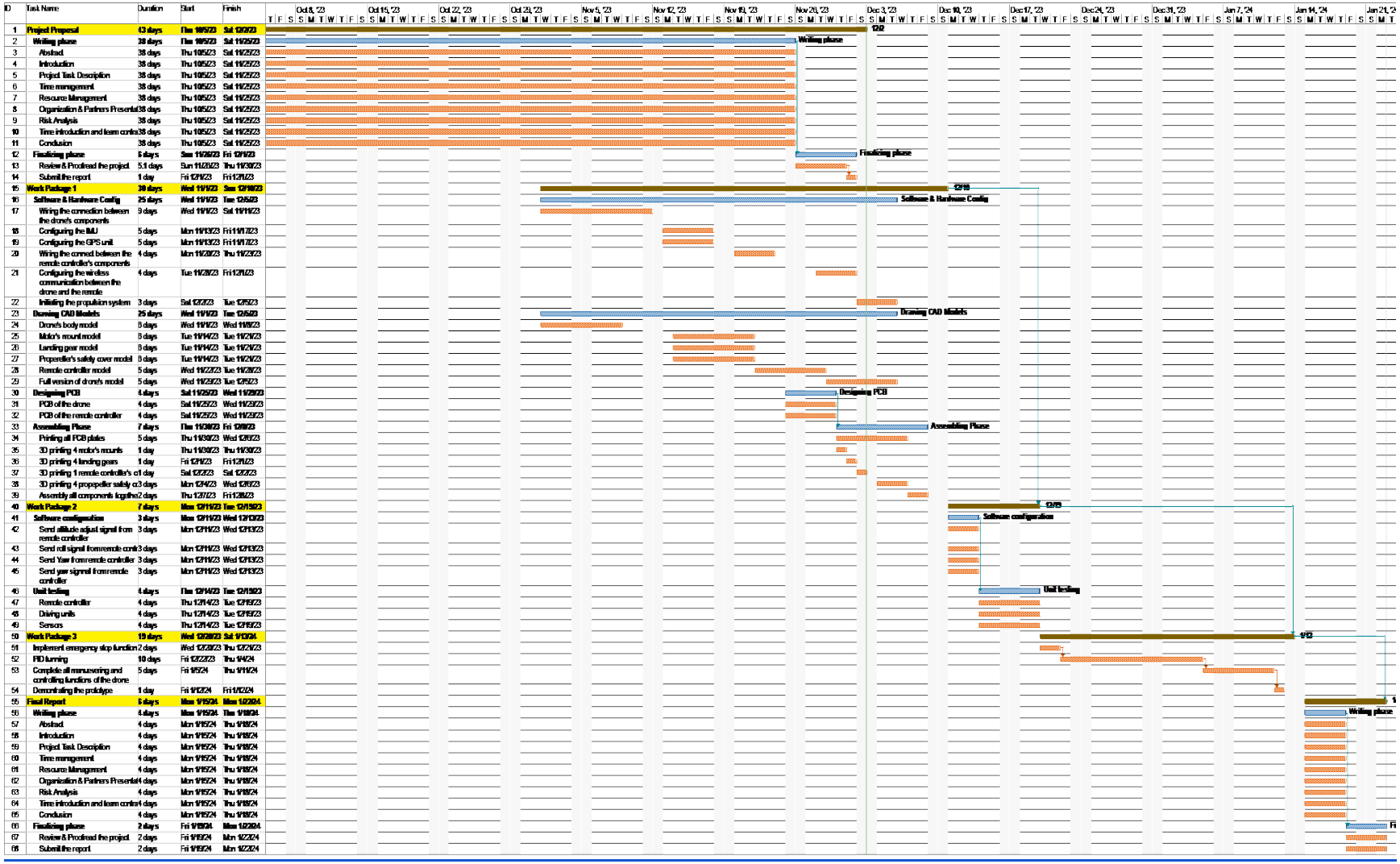

*Appendix 14. Potentiometer. Link*

*Appendix 15. Propellers. Link*




*Appendix 16. MPU6050. Link*

*Appendix 17. GPS NEO8M. Link*




| ID | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | Project Proposal | 42 days | Thu 10/5/23 | Sat 12/2/23 |
| 2 | Writing phase | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 3 | Abstract | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 4 | Introduction | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 5 | Project Task Description | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 6 | Time management | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 7 | Resource Management | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 8 | Organization & Partners Presenta | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 9 | Risk Analysis | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 10 | Time introduction and team contra | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 11 | Conclusion | 38 days | Thu 10/5/23 | Sat 11/25/23 |
| 12 | Finalizing phase | 6 days | Sun 11/26/23 | Fri 12/1/23 |
| 13 | Review & Proofread the project | 5.1 days | Sun 11/26/23 | Thu 11/30/23 |
| 14 | Submit the report | 1 day | Fri 12/1/23 | Fri 12/1/23 |
| 15 | Work Package 1 | 38 days | Wed 11/1/23 | Sun 12/10/23 |
| 16 | Software & Hardware Config | 25 days | Wed 11/1/23 | Tue 12/5/23 |
| 17 | Wiring the connection between the drone's components | 9 days | Wed 11/1/23 | Sat 11/11/23 |
| 18 | Configuring the IMU | 5 days | Mon 11/13/23 | Fri 11/17/23 |
| 19 | Configuring the GPS unit | 5 days | Mon 11/13/23 | Fri 11/17/23 |
| 20 | Wiring the connect between the remote controller's components | 4 days | Mon 11/20/23 | Thu 11/23/23 |
| 21 | Configuring the wireless communication between the drone and the remote | 4 days | Tue 11/28/23 | Fri 12/1/23 |
| 22 | Initiating the propulsion system | 3 days | Sat 12/2/23 | Tue 12/5/23 |
| 23 | Drawing CAD Models | 25 days | Wed 11/1/23 | Tue 12/5/23 |
| 24 | Drone's body model | 8 days | Wed 11/1/23 | Wed 11/8/23 |
| 25 | Motor's mount model | 8 days | Tue 11/14/23 | Tue 11/21/23 |
| 26 | Landing gear model | 8 days | Tue 11/14/23 | Tue 11/21/23 |
| 27 | Propeller's safety cover model | 8 days | Tue 11/14/23 | Tue 11/21/23 |
| 28 | Remote controller model | 5 days | Wed 11/22/23 | Tue 11/28/23 |
| 29 | Full version of drone's model | 5 days | Wed 11/29/23 | Tue 12/5/23 |
| 30 | Designing PCB | 4 days | Sat 11/25/23 | Wed 11/29/23 |
| 31 | PCB of the drone | 4 days | Sat 11/25/23 | Wed 11/29/23 |
| 32 | PCB of the remote controller | 4 days | Sat 11/25/23 | Wed 11/29/23 |
| 33 | Assembling Phase | 7 days | Thu 11/30/23 | Fri 12/8/23 |
| 34 | Printing all PCB plates | 5 days | Thu 11/30/23 | Wed 12/6/23 |
| 35 | 3D printing 4 motor's mounts | 1 day | Thu 11/30/23 | Thu 11/30/23 |
| 36 | 3D printing 4 landing gears | 1 day | Fri 12/1/23 | Fri 12/1/23 |
| 37 | 3D printing 1 remote controller's co | 1 day | Sat 12/2/23 | Sat 12/2/23 |
| 38 | 3D printing 4 propepeller safety co | 3 days | Mon 12/4/23 | Wed 12/6/23 |
| 39 | Assembly all components together | 2 days | Thu 12/7/23 | Fri 12/8/23 |
| 40 | Work Package 2 | 7 days | Mon 12/11/23 | Tue 12/19/23 |
| 41 | Software configuration | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 42 | Send altitude adjust signal from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 43 | Send roll signal from remote contr | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 44 | Send Yaw from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 45 | Send yaw signmal from remote controller | 3 days | Mon 12/11/23 | Wed 12/13/23 |
| 46 | Unit testing | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 47 | Remote controller | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 48 | Driving units | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 49 | Sensors | 4 days | Thu 12/14/23 | Tue 12/19/23 |
| 50 | Work Package 3 | 19 days | Wed 12/20/23 | Sat 1/13/24 |
| 51 | Implement emergency stop function | 2 days | Wed 12/20/23 | Thu 12/21/23 |
| 52 | PID turning | 10 days | Fri 12/22/23 | Thu 1/4/24 |
| 53 | Complete all maneuvering and controlling functions of the drone | 5 days | Fri 1/5/24 | Thu 1/11/24 |
| 54 | Demonstrating the prototype | 1 day | Fri 1/12/24 | Fri 1/12/24 |
| 55 | Final Report | 5 days | Mon 1/15/24 | Mon 1/22/24 |
| 56 | Writing phase | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 57 | Abstract | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 58 | Introduction | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 59 | Project Task Description | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 60 | Time management | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 61 | Resource Management | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 62 | Organization & Partners Presenta | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 63 | Risk Analysis | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 64 | Time introduction and team contra | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 65 | Conclusion | 4 days | Mon 1/15/24 | Thu 1/18/24 |
| 66 | Finalizing phase | 2 days | Fri 1/19/24 | Mon 1/22/24 |
| 67 | Review & Proofread the project | 2 days | Fri 1/19/24 | Mon 1/22/24 |
| 68 | Submit the report | 2 days | Fri 1/19/24 | Mon 1/22/24 |

*Appendix 18. Full version of the project's Gantt chart*