ElevateXY
Colby Beaman
Emmanuel Palma
Alyssa Rocco

CONCEPT OF OPERATIONS

REVISION – 4
8 December 2025

Concept of Operations
for
ElevateXY

Team 24

Approved by:

____
Colby Beaman Date

____
Prof. Nowka Date

____
Swarnabha Roy Date

Change Record

1	2/5/2025	Alyssa Rocco		Draft Release
2	2/11/2025	Emmanuel Palma		Post CONOPS
3	4/26/2025	Emmanuel Palma		Updated to Align with FSR/ICD
4	12/6/2025	Colby Beaman		Final Release

Concept of Operations Revision - 4
ElevateXY

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

1. Executive Summary

The objective of this project is to design a power-efficient attachment for a 6S drone, with a mobile application that provides real-time power consumption analytics and dual-control functionality. To achieve this, we will develop a DC-DC converter to be integrated into the drone's battery system, helping stabilize the output voltage and ensuring high power conversion efficiency. Specifically, we will use a buck converter to regulate a stable low-voltage output, while minimizing energy loss and maximizing efficiency. This converter will also help extend the battery's lifespan by reducing stress during discharge cycles. A microcontroller will be attached to monitor data such as voltage, current, and battery usage, and it will transmit this information to the mobile application. The attached microcontroller will extract data such as voltage, current, and battery usage, and transmit it to the mobile application.

2. Introduction

The following document introduces ElevateXY, an integrated power and control system for autonomous drone operation. ElevateXY will monitor the drone's performance during a live reception while providing users with feedback and suggestions via the app.

Autonomous controls will be applied when needed or when the user sets them. The overall goal of ElevateXY is to provide an efficient drone suitable for any application.

2.1. Background

Over the past decade, companies have gradually integrated AI into their systems and products, offering it as a service or tool for customers. These innovations range from self-driving cars to online tools that compile information from multiple articles into bullet points. There is increasing demand for autonomous vehicles across sectors, intended to transport people and goods by land or air. However, existing solutions, particularly drones used for delivery, surveillance, and agriculture, face challenges such as inefficient power management and limited autonomous navigation capabilities.

ElevateXY aims to optimize drone battery life by regulating output voltage and performance. This approach will ensure consistent, optimal power and extend the overall battery lifespan. While in flight, ElevateXY will also provide autonomous features, including object detection, obstacle avoidance, and tracking functions. This will ensure consistent, optimal power, thereby extending its overall battery life. While flying, ElevateXY will provide autonomous features such as object detection, obstacle avoidance, and following.

2.2. Overview

ElevateXY will integrate a DC-DC buck converter into the battery to optimize battery performance and ensure long-lasting operation. The buck converter used in this project is an LMR51635 synchronous buck converter from Texas Instruments. The LMR51635 provides a converter capable of driving up to 3.5 Amps of load current while providing a wide input voltage range. The converter will be powered by a 4S Li-Po battery, typically with a voltage range of 12-17V and a nominal 14.8V. The LMR51635 will operate at a 400kHz switching frequency to support relatively small inductors. This will help reduce the design size, allowing a lightweight converter.

Additionally, the device uses pulse frequency modulation, which provides high efficiency at light loads. Another feature the LMR51635 provides is built-in protection, including cycle-by-cycle current limit, thermal shutdown in the event of excessive power dissipation, and short-circuit protection. Attached to the converter will be a power management unit that will monitor the drone's temperature, voltage, and current. As it measures and monitors, it will send the data to a microcontroller attached to the system. Once the microcontroller collects the data from the power management unit, it will send the drone's data information to a user via the app. From the consumers' point of view, the app will display how the drone operates and its overall performance.

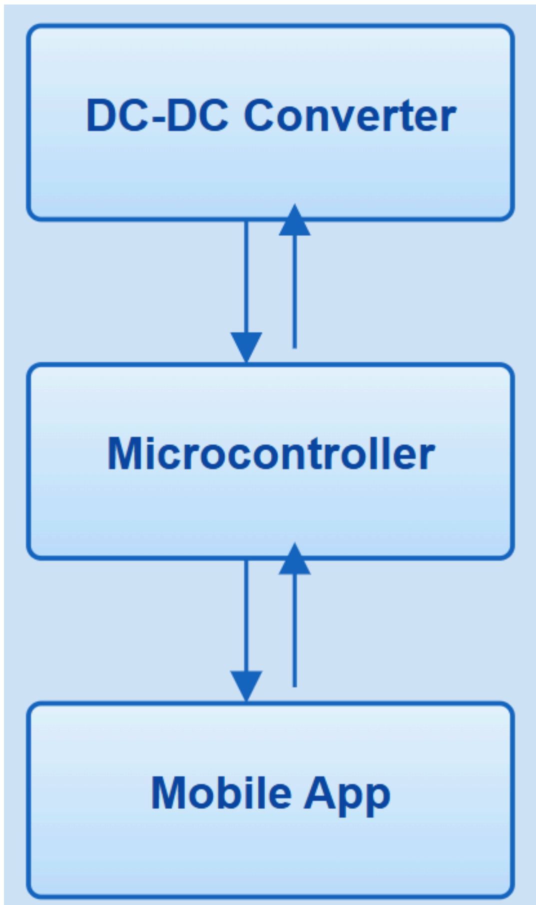


Figure 1: *Flow of information through ElevateXY subsystems*

2.3. Referenced Documents and Standards

- "Client Challenge." *MPS | Monolithic Power Systems*, www.monolithicpower.com/en/learning/mpscholar/power-electronics/dc-dc-converters/buck-converters#:~:text=The%20buck%20converter%2C%20also%20referred,to%20a%20lower%20output%20voltage.
- "Just a Moment..." *Just a Moment..*, thundersaidenergy.com/downloads/dc-dc-power-converters-efficiency-calculations/#:~:text=DC%2DDC%20power%20converters%20are,are%20higher%20at%20low%20loads
- "How to Design a Fly-Buck DC/DC Converter." *PCB Design & Analysis Resources | Cadence*, 4 Feb. 2025, resources.pcb.cadence.com/blog/how-to-design-a-fly-buck-dc-dc-converter.

3. Operating Concept

3.1. Scope

For the scope of this project, a proof of concept is being built that will be able to take advantage of already existing sensors in a drone's flight data recorder as well as an additional power management system. The deliverables of this project consist of the following:

- DC-DC Converter
- Manual / Autonomous Program with an MCU
- Mobile App

Proper documentation relative to design, programming, and building will be included later in the FSR document.

3.2. Operational Description and Constraints

Elevate XY is designed as an add-on for drones currently in the industry, which often suffer from limited flight time and high power consumption. This device aims to provide customers with enhanced data viewing capabilities for flight monitoring and battery health information. The intended operating environments for this device are as follows:

- Temperature Conditions (14° to 104°)
- Safely in Voltage and Charge Rate Parameters
- Unblocked communication between device and the telemetry module - Small enough noise to not affect RF in the telemetry module

Other factors that affect the device's performance include rain or water, areas with strong electromagnetic fields, areas with intense radiation, and tornadoes or hurricanes with strong winds.

3.3. System Description

Elevate XY will integrate multiple power sensors into a central telemetry unit, a database, and a mobile application to view the information. These can be broken down into three different subsystems described below:

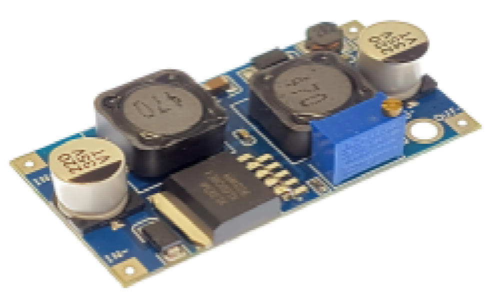


Figure 2: *Buck Converter*

DC-DC Converter Subsystem: A Buck converter converts a high input voltage to a lower output voltage. It can also be referred to as a step-down converter. Buck converters are operated primarily through switches, inductors, and capacitors.

Power Management Unit Subsystem: The Power Management Unit is dedicated to gathering information from the sensors and outputting critical information to the microcontroller. Specific information includes, but is not limited to, voltage, current, temperature, state of charge, and battery health. Displaying all of this information will help users better understand the performance of the drone's power.

Voltage Sensor: A Voltage Sensor is a device that measures the Voltage difference between two points on an electrical circuit. It is meant to ensure that systems are operating efficiently and not suffering from overvoltage or undervoltage, which can damage equipment if not correctly dealt with. Should this occur, the sensor can send a live voltage level reading to the Battery Management System.

Current Sensor: A Current Sensor is a device that Measures the Electrical current flowing through a circuit on a board. Standard current sensors include shunts and magnetic resistors. Both types use the voltage developed across a resistor or a magnetic sensing device element with a conductor to read the current value, which is sent to the Battery Management System to be combined with other elements.

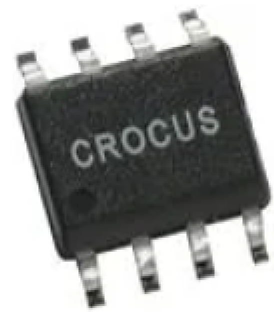


Figure 4: *Current Sensor*

Thermal Sensor: A Thermal Sensor is a device that monitors the board's temperature to prevent overheating. This sensor works by converting the change in heat into a change in resistance, subsequently changing a measurable amount of voltage, thus sending an electrical signal to the Battery Management system.

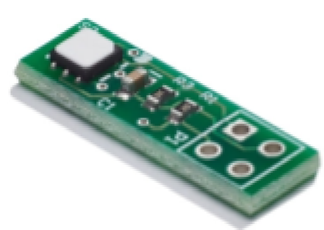
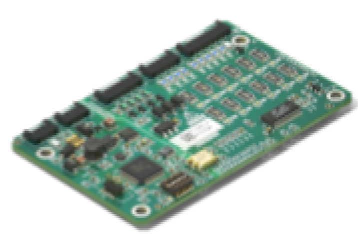


Figure 5: *Thermal Sensor*

Battery Management System: A Battery Management System is an electrical system that regulates and monitors the operation of a battery



during charge and discharge. It also works with all of the necessary sensors to output data that is used to determine power consumption and efficiency. Using this data, it can estimate the battery's state of charge and current state of health.

Microcontroller Subsystem: The Microcontroller will interact with each component to allow for seamless integration with sensors. The flight data collector is embedded with the drone and the wireless telemetry module. It will act as the brain of the device, using algorithms to execute different functions when needed in the system. The wireless Telemetry module can also convey data, sending necessary data to the Mobile application.

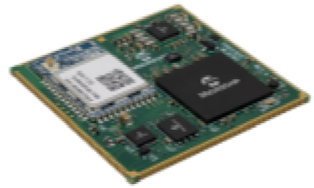


Figure 7: *Microcontroller*

Wireless Telemetry Module: The Wireless Telemetry

The module allows a connection between the Microcontroller and the Mobile Application. With this connection, the Microcontroller can send insights and information to the Mobile Application, and the Mobile Application can send functions to perform on the Microcontroller that will affect the system

.

Figure 8: *Wireless Telemetry Module*

Mobile Application Subsystem: The Mobile Application Subsystem will act as an online database, with proper formatting for any user to view information from the entire system. This will allow the user to analyze real-time data from the drone, which will come from the microcontroller's programming software. This will create an environment with a user control over the entire system, as well as automated operation.

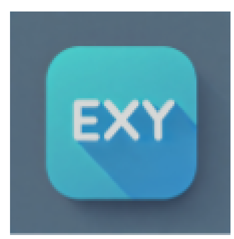


Figure 9: *Mobile Application*

3.4. Modes of Operations

Elevate XY will use a primary mode of operation where data is collected from sensors, the DC-DC converter, and the drone's flight data recorder, sent to the battery management system, then to the microcontroller, and finally sent through the telemetry module and able to be viewed on the mobile application.

The system will also have safety features in place, such as the microcontroller automatically turning off different parts of the system in case of abnormal thermal conditions, over or under voltage, and other scenarios where the battery is threatened. A subsequent notification will also be displayed on the mobile application as a secondary safety measure to ensure the user is aware.

3.5. Users

The ElevateXY is designed to be used by anyone with the ability to use a smartphone device and have an understanding of drone movements.

3.6. Support

A web application will contain information on the Elevate XY application for the drone. It will also provide real-time information about the battery, such as the output voltage, current, power delivery to each motor, predicted flight time remaining, etc.

4. Scenarios

4.1. Redundancy for Critical Operations

Drones used in critical operations, such as military reconnaissance or medical supply delivery, must not experience outages at crucial moments. Implementing an additional Battery Management System that provides drone operators with performance information could save countless lives in an emergency. Drones used in operations where the objective is vital, such as military reconnaissance or medical supply delivery, cannot experience an outage at a critical moment. Having an extra Battery Management System in place that can send Drone operators information on the drone's performance could save numerous lives in the event of a catastrophe.

4.2. Hybrid-Electric UAVs

UAVs using multiple different power sources need to transition seamlessly between them at a moment's notice. A system that monitors efficiency and selects the appropriate source based on load conditions could significantly extend a drone's lifespan. This becomes even more important when, in the event that one of the sources fails, a drone should be able to switch to a separate power source and stop flying.

4.3. High-Power Applications

Drones carrying heavy payloads, such as agricultural drones, consume significantly more power than smaller drones, such as quadcopters. Thus, they rely even more on efficiency than smaller drones. In addition to everyday use in industry, there is the concept of "hot-swappable batteries," which enable continuous operation. However, if a drone lacks a battery management system capable of allocating power from multiple sources, this concept cannot be achieved.

5. Analysis

5.1. Summary of Proposed Improvements

- ElevateXY will help optimize the drone's power system, allowing for longer flight times and potentially carrying heavier payloads in specific industries while providing real-time surveillance.

- Reducing energy consumption, maximizing battery usage, and decreasing the frequency of battery recharging will help conserve energy, limit electronic waste, and lower people's carbon footprint.
- Businesses that use drones will be able to lower operating costs by reducing energy and maintenance costs on these drones, allowing them to make more technological advancements and investments.

5.2. Disadvantages and Limitations

- Limitation in aerial distance
- Harsh weather conditions will limit our system's ability to provide longer flight times.
- Due to the attachment of the sensors and converter, our system may impact battery usage slightly due to its weight.
- Our system will be limited to repairs due to its assembly.

5.3. Alternatives

One alternative to improve drone performance is to attach a larger battery. However, this option presents several challenges. The increased battery size would add weight to the drone, potentially affecting its takeoff and landing capabilities. Additionally, a larger battery would require more frequent recharging, leading to increased battery waste and environmental concerns.

Another potential alternative for this project is to develop a fully autonomous drone. This design would be ideal for delivery services, where the drone can perform a specific task and then return to the user. To enhance its functionality, a scheduling system could be implemented for tasks that need to be repeated.

ElevateXY
Colby Beaman
Emmanuel Palma
Alyssa Rocco

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 1
20 February 2025

FUNCTIONAL SYSTEM REQUIREMENTS

FOR

Elevate XY

TEAM 24

APPROVED BY:

Colby Beaman Date

Prof. Nowka Date

Swarnabha Roy Date

Change Record

1	2/20/2025	Emmanuel Palma		Draft Release
2	12/6/2025	Emmanuel Palma		Final Release

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

ElevateXY aims to enable users to track their drone's flight activity and extend flight time by attaching a DC-DC converter to the drone. With the addition of an AI Microcontroller, users will be able to view the drone's voltage, output power, and battery usage in our app. Integrating the AI microcontroller will give us unique features, such as object detection and auto-follow while stopped, preventing users from potentially damaging their drones. Figure 1 shows a representation of the integration of our system.

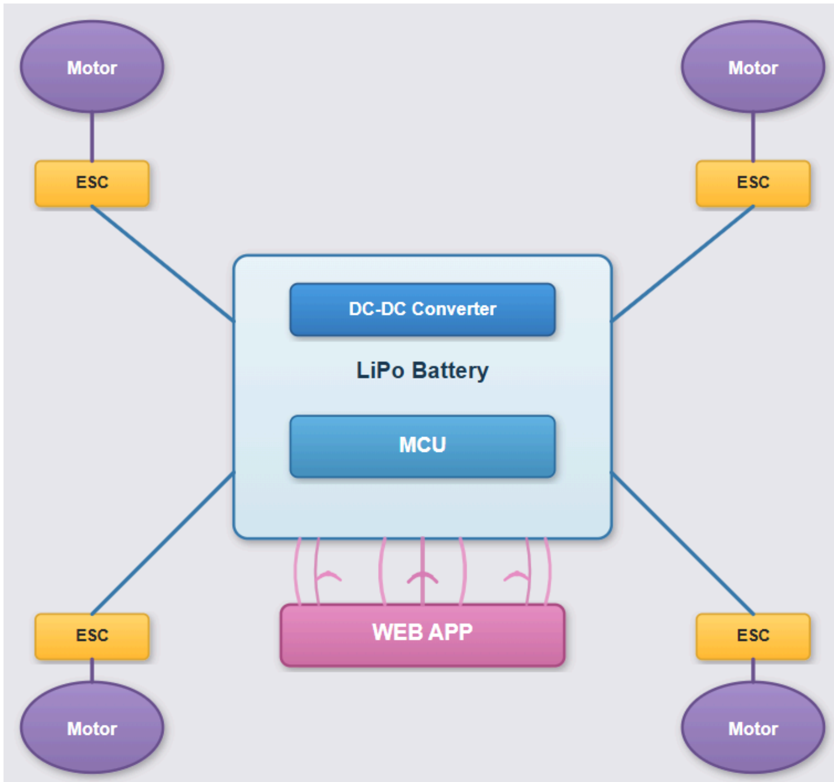


Figure 1. Project conceptual image

1.2. Responsibility and Change Authority

The team leader, Colby Beaman, is charged with ensuring that all system specifications are met. Any changes to the project's requirements or deliverables must be approved by the team leader, Colby Beaman, and the sponsor's official representative, Hadir Khan.

Subsystem	Responsibility
DC-DC Converter	Emmanuel Palma
AI/MCU	Colby Beaman
Web Application	Alyssa Rocco

Table 1: Subsystems Leads

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
1	5/10/2021	Diagnostic Trouble Codes (DTC) Alerts in DroneMoblie App
113823	Volume 255- 12/1/2019	Applied Energy 255- “Assessing the contribution of simultaneous heat and power generation from geothermal plants in off-grid municipalities.”
PEDS57185.2023.102 466 76	8/10/2023	Three-Phase AC/DC Converter fed Two parallel interleaved DC-DC Converters for Fast Charging Applications with Improved Power Quality
10.1109/IECON.2019.89 26 872	12/9/2019	DC-DC Converters for Medium and High Voltage Applications
ANSI C119.6	2011	American National Standard for Electric Connectors Non-Sealed, Multiport Connector Systems Rated 600V or Less for Aluminum and Copper Conductors
MIL-HDBK-5400	11/30/1995	Electronic Equipment, Airborne General Guidelines

Table 2: Applicable Documents

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
SLUSF64B	8/2024	LMR516x5 4.3 to 60V,2.5A/ 3.5A, Synchronous, Buck Converter with Low
2	Version 25	Altium Designer Documentation
DA_09402_003	12/17/2019	NVIDIA Jetson Nano Developer Kit
TB_10749-001	1.2 July 2022	NVIDIA Jetson AGX Orin Series

Table 3: Reference Documents

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

This section will focus on the comprehensive system being developed for "ElevateXY," which includes the buck converter, the artificial intelligence commands integrated into the microcontroller, and the mobile application that allows users to view real-time data being collected. The term "buck converter" refers explicitly to the converter connected to the motors to stabilize the input voltage. The term "artificial intelligence" refers solely to the commands that will be executed by the microcontroller to perform specific tasks. Additionally, the term "graphical user interface" refers explicitly to the mobile application, which provides users with information about the drone's battery and enables them to select specific controls for the drone, whether for manual or automated operation. Specific controls for the drone to perform either manually or automatically.

3.1. System Definition

ElevateXY will contain three subsystems, as shown in Figure 2. Each subsystem depends on the others to properly function as a whole. The main subsystems will include the buck converter, the mobile application, and the microcontroller/artificial intelligence. They all align with one another for the whole system to function correctly.

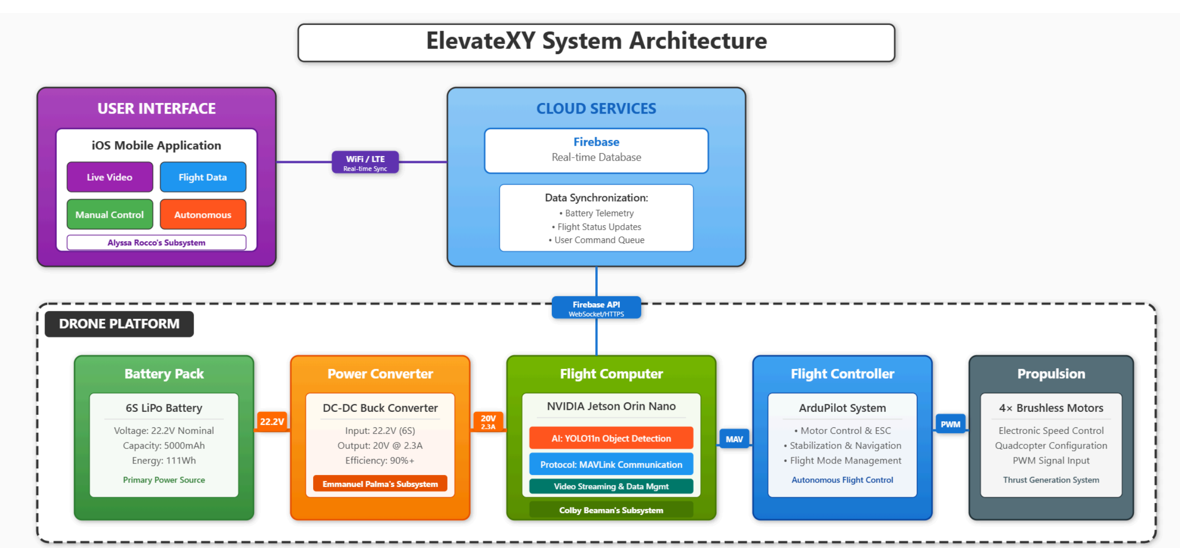


Figure 2: Block diagram of ElevateXY

The buck converter will take the input voltage from the drone's battery and regulate and stabilize the voltage sent to the motors. This will help optimize battery life and provide consistent power to the motors, maximizing their efficiency.

The microcontroller will act as the device's brain, transmitting and receiving data and executing commands. Specific commands can come from AI decision-making or from manual user input.

The mobile app displays the data collected from the microcontroller for the user to view in real time. There will also be a live video feed from the drone on the mobile app for the user to access. This is also where the user will be able to select the different commands that are from the AI subsystem for the drone to perform.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

The mobile application must display data from the microcontroller with 100% accuracy. Additionally, a live camera view will be shown to the user with a delay of at least 5 seconds.

3.2.1.1. Voltage Regulation

ElevateXY will stabilize the battery's output voltage to have a consistent output of 14.8 Volts within a 1% range.

Rationale: The motor for the drone requires this voltage in order to properly perform

3.2.1.2. Accuracy of Measurements

ElevateXY will provide real time data that is accurate from the battery management system that is within 0.25% to 1% of error.

Rationale: The values collected will be used to determine the State-of-Charge and the State of Health of the battery.

3.2.1.3. Communication Requirements

ElevateXY will require line of sight of the drone to be able to transmit real time information to the mobile application at least 20 feet away

Rationale: Clear sight of the drone allows for the data to be quickly collected and provided to the user

3.2.2. Physical Characteristics

3.2.2.1. Mass

ElevateXY will weigh less than 200 grams when not attached to the drone.

Rationale: This is a requirement in order to meet the MTOW limit of the drone

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

ElevateXY is specifically designed to draw power from a 4S LiPo battery connected to the drone. It is important to note that while the code is explicitly intended for drones equipped with 6S LiPo batteries, a 4S battery is required to power the MCU.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.1.1 Power Consumption

The system's peak power shall not exceed the maximum thrust test, which is roughly 424 Watts. Power consumption will be based on the 6s battery's current state.

3.2.3.1.2 Input Voltage Level

The input voltage level for each component shall be within +5 VDC to +25 VDC. *Rationale: The battery will be a 6s LiPo ranging from +18 VDC to +25 VDC to the DC-DC converter, but the microcontroller will need 20V but can be ranged from 9-20V.*

3.2.3.1.3 External Commands

ElevateXY shall document all external commands in the appropriate ICD. 10

3.2.3.2. Outputs

3.2.3.2.1 Data Output

The ElevateXY System will output data from the sensors to a server, which will store and display the data to the mobile application.

3.2.3.2.2 Diagnostic Output

ElevateXY shall include a diagnostic interface for control and data logging.

> *Rationale: Provides the ability to control things for debugging manually and a way to view/download the node map with associated potential targets.*

3.2.3.2.3 Raw Video Output

ElevateXY will include raw video interference, which will be seen by the user through the mobile application.

3.2.3.3. Connectors

ElevateXY shall use external connectors in accordance with the American National Standard for Electrical Connectors ANSI C119.6-2001

> *Rationale: Conform to connector standard*

3.2.3.4. Wiring

ElevateXY shall follow the guidelines outlined in MIL-HDBK-5400 paragraph 4.3.35 Wire and cable.

Rationale: Conform to aircraft standard.

3.2.4. Environmental Requirements

3.2.4.1. Altitude

ElevateXY will be designed to operate up to 400 ft.

Rationale: This is the legal limit from the FAA of the United States of how high a drone can fly in the sky.

3.2.4.2. Thermal

ElevateXY shall be designed to operate in temperatures between -13℉ to 177℉. This is mainly due to the fact that the microcontroller can be damaged outside of this range.
Rationale: This range will work for most regions and will be able to be implemented to many drones.

3.2.5. Failure Propagation

3.2.5.1. Failure Detection, Isolation, and Recovery (FDIR)

ElevateXY will use a checking algorithm to identify if the microcontroller is transmitting incorrect data and will notify the user if this error is occurring. They will also be able to troubleshoot the issue.

3.2.5.1.1 Battery Faulty Condition

ElevateXY will be able to monitor the battery life as well as inform the user of its state and perform an emergency landing if it will damage the battery in the state that it is in.

4. Support Requirements

4.1.1. iPhone with Internet Access: An iPhone with an Internet connection is needed to control the drone and view the necessary data provided in the GUI.

Rationale: Without a proper established connection data cannot be transmitted to and from the Drone. Though AI can be used, the Drone would be flying "blind" in the sense it couldn't receive user input.

4.1.2 Permitting Weather: Ensure that weather conditions are stable enough to not dramatically affect the Motor requirements.

Rationale: Should the Drone be used in a heavy-wind environment that could increase the load needed from the motors which would significantly decrease the usage time. As well should there be rain, the open electronics would cease to function optimally.

4.1.3. Battery Care: Proper charge and care of the battery is necessary to ensure that the drone has ample power to fly and execute commands.

Rationale: The system is designed to have enough power to fly for a period of ten minutes along with being able to receive commands. Not keeping the battery in optimal condition will affect the intended function of the device.

Appendix A: Acronyms and Abbreviations

FAA Federal Aviation Administration
GPS Global Positioning System
ICD Interface Control Document
MTOW Maximum Takeoff Weight
mA Milliamp
kHz Kilohertz (1,000 Hz)
mW Milliwatt
PCB Printed Circuit B
TBD To Be Determined
USB Universal Serial Bus

Appendix B: Definition of Terms

Buck Converter	A Buck converter is a type of converter that changes an input voltage to a lower voltage level. Also referred to as a step-down converter. operated primarily through switches, inductors, and capacitors.
BMS	A BMS is a Battery Management system that ensures proper Voltage, Current, and Temperature are maintained during use.

ElevateXY

Colby Beaman
Emmanuel Palma
Alyssa Rocco

INTERFACE CONTROL DOCUMENT

Interface Control Document Revision -1 ElevateXY

REVISION – Draft
20 February 2025

INTERFACE CONTROL DOCUMENT

FOR

ElevateXY

TEAM 24

APPROVED BY:

Colby Beaman Date

Prof. Nowka Date

Swarnabha Roy Date

Change Record

1	2/20/25	Emmanuel Palma		Draft Release
2	12/07/25	Emmanuel Palma		Final Release

Table of Contents

List of Tables

List of Figures

Interface Control Document Revision -1
ElevateXY

1. Overview

The Interface Control Document (ICD) for the entire Elevate XY will provide a comprehensive overview of the subsystems listed in the Concept of Operations and Functional System Requirements will be produced. The ICD will include physical descriptions of the needed components including power, sensors, and converters. This document will also explain the integration between the subsystems necessary to accomplish the objectives and requirements written in the FSR and ConOps documents.

2. References and Definitions

2.1. References

Version 25
Altium Designer Documentation

DS_10712_001
NVIDIA Jetson Orin Development Kit
11/8/2022

10.1109/IMIS.2016.128
A Collaborative Safety Flight Control System for Multiple Drones
7/2016

SLUSF64B
LMR516x5 4.3 to 60V,2.5A/ 3.5A, Synchronous, Buck Converter with Low
8/2024

2.2. Definitions

A Amps
V Volts
uF microFarad(10^-6F)
W Watts
H Henry
kHz Hertz (1,000 Hz)
D Duty cycle
FSW Switching frequency
IC Integrated circuit
TBD To Be Determined
MTOW Maximum-Takeoff Weight
PCB Printable-circuit Board
MCU Microcontroller Unit
COG Center of Gravity
USB Universal Serial Bus
UART Universal Asynchronous Receiver-Transmitter GUI Graphical
User Interface

3. Physical Interface

3.1. Weight

3.1.1 Weight of the Buck Converter

The total weight of the Buck Converter needs to be under 200 grams, which includes parts like the PCB, Resistors, Capacitors, Transformers, and all of the other necessary circuitry. Ensuring that the Converter is within the MTOW is vital as it can drastically affect the drone's flight time if not managed properly.

3.1.2 Weight of the MCU

The NVIDIA Jetson Orin weight is essential as it is the brain of the project. Should it shift during mid flight, that could lead to a loss of connection with sensor data. Potentially even worse is the loss of connection with controlling the drone, meaning that the drone would be unlikely to survive landing.

Component	**Weight**
Tattu Plus 220Ah 6S LiPo Battery	2650 g
NVIDIA Jetson Orin Nano	176 g
DC-DC Buck Converter	36 g
AI Camera Module	90 g
Propellers (15-18”)	90 g
Landing Gear & Miscellaneous needed Hardware	225 g
Total Weight Additions	3,341 g or 3.341 kg

Table 1: Drone Part Weight Specifications

3.1.3 Weight of the Battery

The battery's weight is most important because it has the most implications of any component. At 2.65 kilograms, it is a large amount of weight that needs to be balanced, or else the whole system will be thrown off. This is especially true when considering the maximum current output to motors and maintaining balance in the circuit.

3.2. Dimensions

3.2.1. Dimensions of Components

Component	**Diameter**	**Length (mm)**	**Width (mm)**	**Height (mm)**
Tattu Plus Bat.	N/A	116	34	38.5
NVIDIA Jetson Orin Nano	79	N/A	100	21

Buck Converter		67	24	20
AI Camera	N/A	38	38	38
Propellers	380-457	N/A	N/A	N/A

Table 2: Dimensions of Components

3.3. Mounting Locations

3.3.1 Placement of the Battery

The battery must be mounted close to the center of gravity (COG) to maintain flight stability and achieve balanced weight distribution. It will be securely fastened using a dedicated battery tray to prevent any movement during flight. Additionally, the USB port should be easily accessible to ensure that the microcontroller unit (MCU) maintains a constant connection with the integrated battery management system (BMS).

3.3.2 Placement of the MCU

The NVIDIA Jetson Orin will be mounted on the Drone's Central Unit, with proper reinforcement to ensure the drone is not too top-heavy. The MCU will be positioned for easy access to the battery and the camera, enabling communication between necessary sensors for real-time updates. A subsequent heat sink will be attached to prevent thermal throttling under heavy load use.

3.3.3 Mounting of Camera

The Camera must be mounted towards the front of the drone, far enough forward on the top or bottom to allow for an unobstructed view. Proper stabilization is vital as well, with a damped gimbal to maintain stable imaging for AI communication.

3.3.4 Mounting of Buck Converter

The converter will be positioned between the Battery and the MCU to minimize cable length and reduce voltage drop. It will also maintain a location with ample airflow to dissipate heat while in use. A protective casing will be installed on the converter to prevent weather damage while airborne.

4. Thermal Interface

4.1 Thermal Management of MCU

The NVIDIA Jetson Orin can generate a substantial amount of heat during AI processing. To prevent thermal throttling, a heatsink and fan will be installed on the MCU.

4.2 Cooling of Battery

The Battery will be designed as an open-air system to promote passive cooling and will feature an additional dedicated fan. If the Battery starts to overheat, the integrated Battery Management System (BMS) will monitor the situation and send an alert through the graphical user interface (GUI). This alert will also propagate to the Jetson system, which will then transmit the information to the mobile application, notifying the user of the issue.

5. Electrical Interface

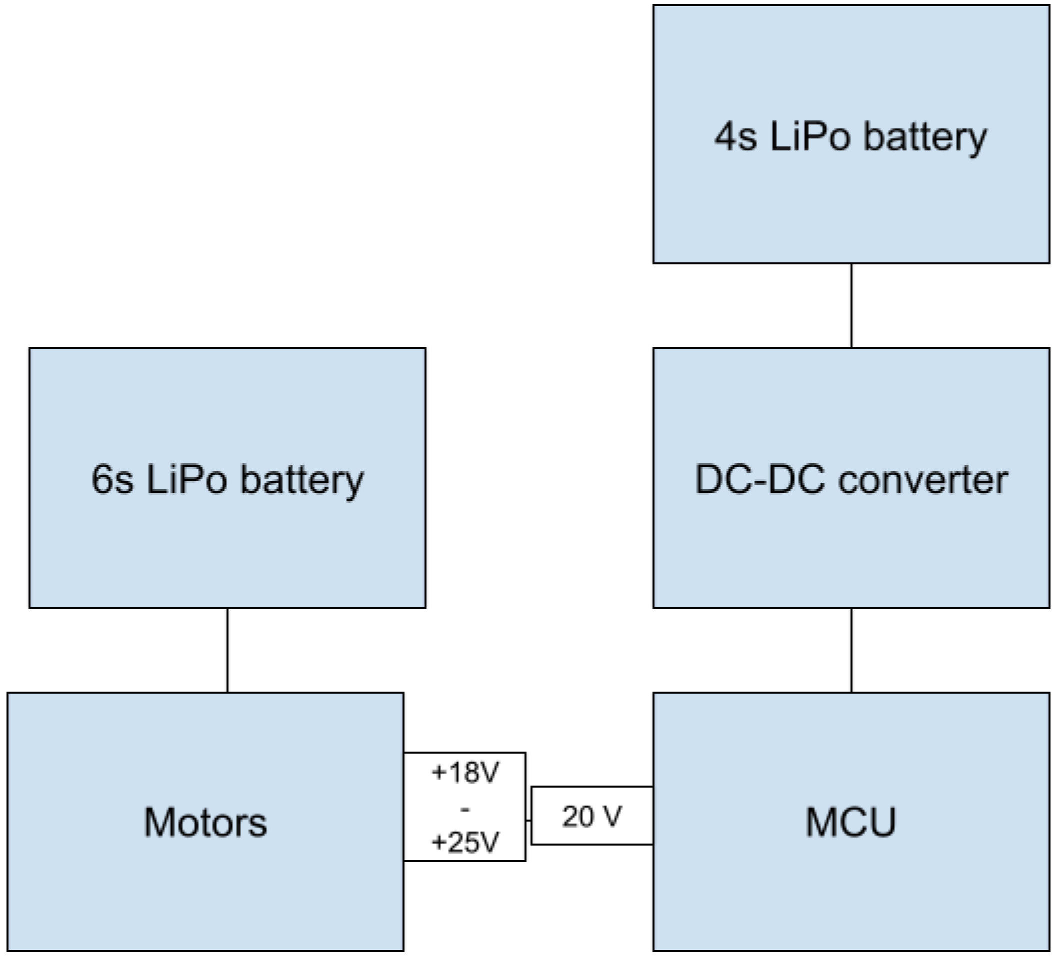


Figure 1: Electrical Interface Diagram

5.1. Primary Input Power

All power for the drone will come from the 6S LiPo battery, which will supply all components attached to the drone, such as the MCU, DC-DC converter, and motors.

5.2. Voltage and Current Levels

Floating Current	Voltage (V)	Current (A)
	10	0.4
	22.2	0.5
	25.2	0.6

Table 3: Voltage and Current values of the motors while Floating

The voltage levels shown in Table 3 will vary depending on the current state of the battery. When the battery is at its maximum charge state, 25.2 V, it will need a current of 0.6 A to float. The drone will mostly stay floating during the battery's nominal voltage, 22.2 V, producing an 11W output power for the motor. Anything lower than that will require the drone to land to charge.

5.3. Video Interfaces

The Elevate XY video system will use a USB 3.0 interface for camera connectivity, ensuring high-speed data transmission, creating an optimal AI processing environment on the microcontroller, and reducing user input delay.

5.4. User Control Interface

Users will be able to view data and control the drone through a GUI, allowing seamless transitions between all parts of the system. To accomplish a particular goal, the user has the option of manual or AI-defined tasks.

6. Communications / Device Interface Protocols

6.1. Battery Telemetry Communication

The Tattu Plus 220 Ah 6S LiPo Smart Battery pack includes an integrated BMS, providing real-time data. Then, the data will be moved to the MCU through the UART over USB. This process allows the Jetson to receive constant communication regarding voltage, current, temperature, and battery health. This protocol operates in the specification of IEEE standard 802.3

6.2. Jetson Orin Interface

Using multiple protocols, the NVIDIA Jetson Orin will process and transmit data from several drone components. The aforementioned wired connections and 802.11 provided cloud connectivity for real-time remote monitoring and control.

6.3. AI Camera Data Transmission

The Raspberry Pi AI-powered camera is connected to the Jetson Orin Nano via USB 3.0 to ensure high-speed image processing for object detection. This protocol also follows the IEEE 802.11 standards.

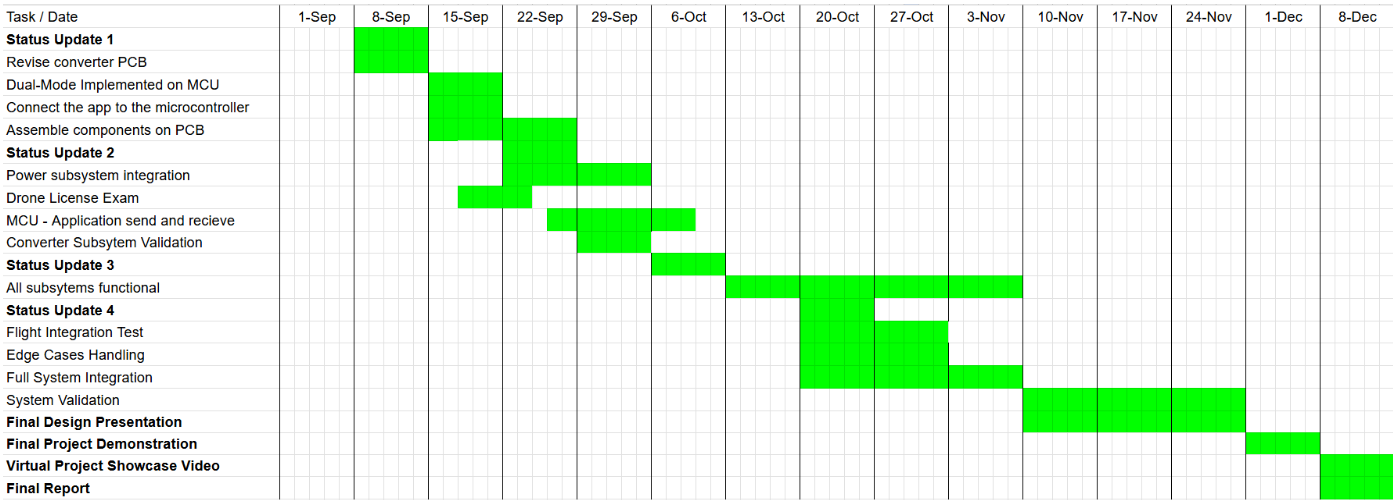
6.4. Wireless Telemetry and Communication

To enable remote monitoring and control, the drone will use the wireless connection capability through the MCU. This will be done through the use of Bluetooth, which falls under the IEEE Protocol of 802.15.1

ElevateXY

Colby Beaman
Emmanuel Palma
Alyssa Rocco

SCHEDULE



Not Started	
Completed	
In Progress	
Behind	

All pending tasks were completed. Any further development would be for further data analysis and or troubleshooting

VALIDATION PLAN

Paragraph #	Test name	Verification Goal	Methodology	Status	Assigned to
3.2.1.3	Object Detection Model Integration	Model loads with 3 seconds, inference <30ms per frame	Time startup protocol, optimize Model to ensure efficiency	VALIDATED	Colby
3.2.1.3	Motor startup Sequence	All motors arm and reach idle with 3 seconds, no failures (specifically ESC errors / disarms)	Send Arm command and monitor RPM, Check for delayed response	VALIDATED	Colby
3.4	Pass Part 107 Exam	Pass and Receive Subsequent Documentation for Licence, followed with blanket form on Flight Request	Take and pass Part 107 exam, Obtain Remote Pilot Certificate from FAA, Complete TAMUS Flight Authorization	VALIDATED	Colby
3.2.1.2	Person Detection Accuracy	Achieves >85% precision within 5-10 feet	Test single person scenarios, multiple person scenarios, and zero person scenarios	VALIDATED	Colby

3.2.1.2	Real-time Processing	Maintains 20+ FPS, displays bounding boxes, no frame drops over 5 minutes	Implement frame counter and monitor under different scenarios, Verify bounding boxes render correctly	VALIDATED	Colby
3.2.3.1.3	MCU-Mobile App Communication	Supports simultaneous telemetry and command transmission, no packet loss during flight	Implement timestamped messages from MCU to App that also Echo, Count received packets on a 10 minute test	VALIDATED	Colby
3.2.3.1.1	Power Consumption Analysis	Nano operates with 15W power during normal AI inference, <25W peak during inference and telemetry	Monitor Power with Model under various loads, Develop a normal threshold based on Testing to Monitor	VALIDATED	Colby
3.2.4.1	Edge Case Handling	No crash: Partial Occlusion, 10+ persons, low light condition (<50 lux)	Test at night (with FAA approved lights), Use video footage of 10+ people with Model, Test with	UNTESTED	Colby

			person behind object		
3.2.3.1.3	Flight Integration Test	Maintains detection during flight maneuvers, no inference with manual controls	Monitor Flight Log, Monitor GPU usage during basic maneuvers (Non-AI), Save video of camera feed	UNTESTED	Colby
3.2.3.1.3	Full System Integration	MCU sends data to mobile app, receives power/flight data, 0 system crashes, no memory leaks	Monitor Network throughput, Monitor CPU and GPU utilization, Monitor various Battery analytics	VALIDATED	Colby
3.2.2.1	Drone weight	Not exceeding the MTOW for the drone	Aiming to have the converter weight >200 grams as well as charging between 6s or 4s	VALIDATED	Emmanuel
3.2.3.1	Update Buck Converter	Redesign Buck Converter to be powered by a LiPo 4S battery	Changing the input of the PCB with the dimension of an XT60 bullet connector	VALIDATED	Emmanuel

3.2.3.1.2	Integrate Buck Converter	Successfully adapt a XT90 connector input to power the converter which will supply a 5V=4A output	Using a multimeter to record the output load of the integrated PCB	VALIDATED	Emmanuel
3.2.3.2.2	Test efficiency of new converter	Prove that the converter has remained highly efficient at 80% after integration	Measuring the calculated output power over the energy input of the PCB	VALIDATED	Emmanuel
3.2.3.2.2	Simulate flight duration	Converter functioning with high efficiency for 10 constant minutes	Using a stopwatch, record a 10 minute lap of the PCB powering the multicontrolle r	VALIDATED	Emmanuel
3.2.3.2.1	Real-time Processing	Drone Data is able to update within one second	Created a test file that updates and see if the data changes on the app	VALIDATED	Alyssa
3.2.1	Display Data	80% of the data is able to be sent and received to the microcontrolle r	Send signals from the app to see if the MCU will output the correct control selected	VALIDATED	Alyssa

3.2.3.2.3	Camera Connection/Quality	Camera quality has been improved and has less than five second delay	Optimize to find the best output	VALIDATED	Alyssa
3.2.3.2.2	Develop API Calls to send the MCU	All the commands run on the MCU	Monitor output sent from App versus Input received on MCU	VALIDATED	Alyssa
3.2.3.2.1	AI Components/ Commands	Drone is able to understand the commands and execute within five seconds	Set a time stamp from when the program started until it is connected	VALIDATED	Alyssa
4.1.1	Test all features combined on the app	Able to perform the functions created	Run successfully with no bugs/ errors throughout the flight time	VALIDATED	Alyssa

Performance on Execution Plan

The execution plan was overall completed, considering that our project was modified mid-semester. Outside factors such as the government shutdown restricted us from having to fly the drone thus having a halt on our subsystems integration analysis. Areas in all subsystems required additional implementation or rework.

Performance on Validation Plan

Validation plans for individual subsystems were overall completed and fully integrated.

ElevateXY can be powered by either a wall outlet cable or by the power converter which power derives from LiPo battery. The Jetson Orin can successfully run a program design to allow a “Dual Control” mode between Manual Autonomos. Offering 3 types of modes adjusted by speed; economy, standard, and performance. The program is able to communicate with the app giving a realtime live stream view of the camera that's connected to the Jetson Orin as well as any data information.

SUBSYSTEM REPORTS

REVISION – 1

Calibrated LWIR Raw Video Data Collection - Gold Standard – Test 1 Revision - 26 April

2025

SUBSYSTEMS REPORT

FOR

Elevate XY

TEAM 24

APPROVED BY:

Colby Beaman Date

Prof. Nowka Date

Swarnabha Roy Date

Change Record

1	4/26/2025	Emmanuel Palma		Original Release
2	12/6/2025	Emmanuel Palma		Final Release

Table of Contents

List of Tables

List of Figures

Subsystem Reports Revision -1
ElevateXY

1. Introduction

The system is broken down into power, microcontroller, and mobile application subsystems. Each subsystem was designed and rigorously tested with multiple variables in consideration. Since each subsystem was validated to work properly, passing all requirements, there will be a clear path to the integration of these subsystems into the full system specified in the ConOps, FSR, and ICD.

2. DC-DC Converter Subsystem Report

2.1. Subsystem Introduction

The DC-DC converter subsystem is designed to supply regulated power to the microcontroller attached to the drone. The converter is powered exclusively by the 4s Li-Po battery, which will be stored below the microcontroller by a surface attached to the drone.

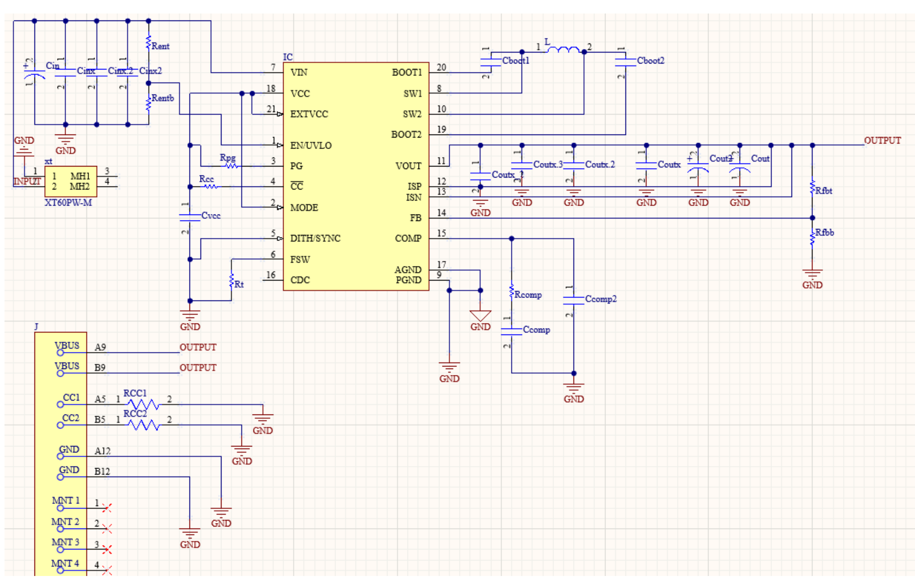
The converter subsystem was tested to confirm its stability, capacity as well and consistency. The test performed helped verify that the buck converter will support the components of the system that will be powered by it. In addition, the test will require a time run to demonstrate that the converter will hover during a 10-minute flight

2.2. Subsystem Details

The primary challenge of this power subsystem shifted throughout the course of the semester. Initially, the power subsystem had to output a power of 5V, with changing of the Jetson nano to a Jetson Orin the power output had to be adjusted. A new converter had to be designed in order to power the MCU at 20V. The overall goal for this subsystem was to develop a converter that would be able to provide efficient power while having a minimal weight influence on the drone. A challenge facing this subsystem was ensuring a stable output when the input voltage would be coming from a 4s Li-Po battery, changing voltage from 16 Volts to as low as 12 Volts with a nominal voltage of 14.8V. In addition, the current needed to power the microcontroller will require a current of 2.3 amps. In addition to the converter a protective casing was also needed to be designed when implementing integration, this was needed to help prevent any damage caused by weather or debris when the drone is airborne.

2.3. Design

One major change that was made when updating the buck converter was its input, rather than having an xt90 male bullet connector input, it was changed to an xt60 in order to reduce the size as well as availability in parts at the time. The new power converter was designed to power the Jetson Orin Nano with a required voltage output of 20 V and an output current of 2.3 amps. The output was designed to be a usb c output in order to be able to provide a reliable fast output current which will be attached to a 2.5 mm power jack. Typically non power outlets power jack are only available to provide a 0.7 current output due to a usb connector, when finding one that is able to provide high current with was only available with a ubc input.



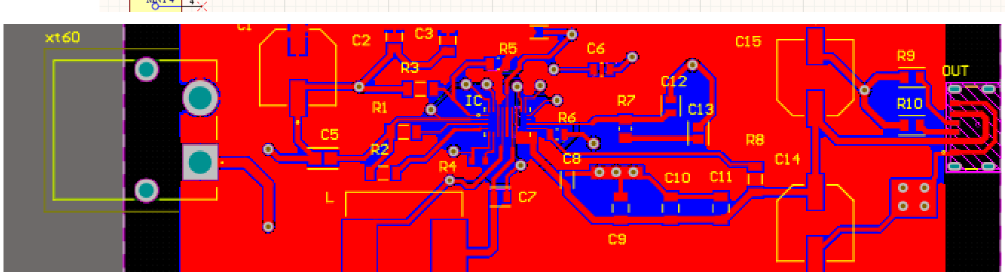


Figure 1. Schematic/PCB layout of updated Converter

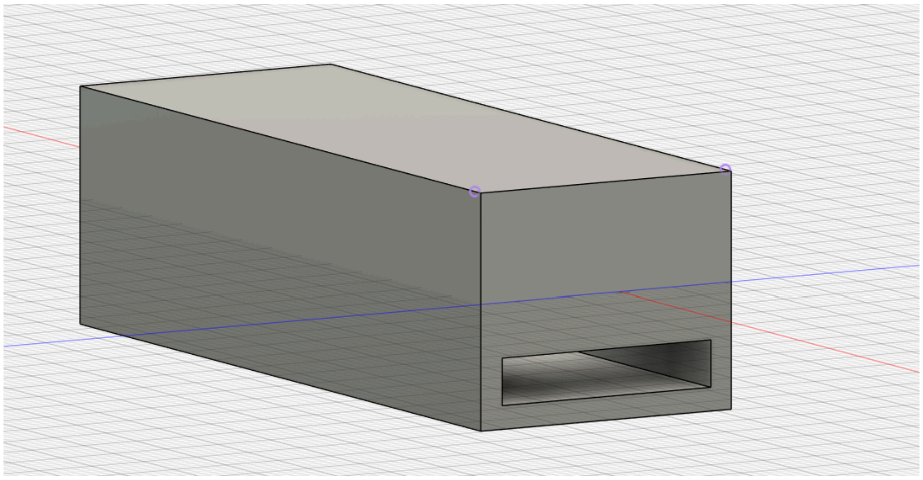


Figure 2. Protective casing for power converter

2.4. Subsystem Validation

Converter Type	Length(mm)	Width (mm)	Height (mm)
Buck 403	60	20	10
Buck 404	67	24	20

Table 1. Dimensions of the converters

Converter Type	Weight (g)
Buck 403	18
Buck 404	36

Table 2. Weight of the converters in grams

2.3.1. Buck Converter

The buck converter was first tested for stability, weighing in at around 18g. The converter was tested by supplying a constant voltage of 22.2 volts, simulating the 6s battery nominal voltage. With a fixed frequency of 400 kHz there was no need for an additional input. For the converter to work correctly, the output voltage needs to be outputting a constant 5 Volts with an efficiency of 80% or higher. Failure to do so will result in troubleshooting the converter to find the error that resulted in the contrast between simulated and measured results.

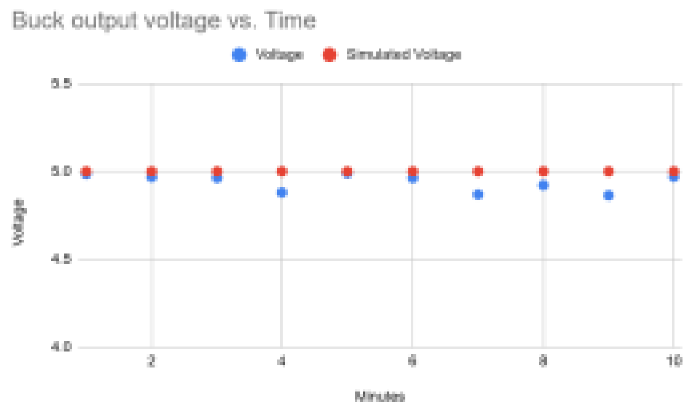


Figure 3. Buck converter output voltage based on 10-minute flight time

Figure 3 shows an ideal outcome, averaging at an output voltage of 4.96. With an efficiency of 90%, this met our requirements for an efficient converter. A second test was performed to simulate the battery drainage when being used. Measuring the output voltage when having an input voltage at a minimal voltage of a 6s battery at 18 Volts to a fully charged battery of 25 volts.

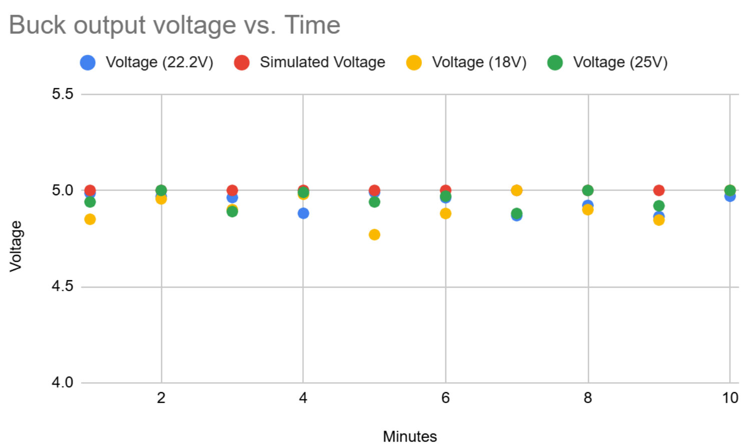


Figure 4. Output voltage based on 10-minute flight time for low, nominal,and high voltage of a 6s battery

When analyzing the overall output voltage of the 3 set conditions, the buck converter was performing at an 89% efficiency. A last validation to test having successfully

passed the other requirements is to test the output power. This will verify that the converter can produce enough power to have the microcontroller running without any interference or issue. The required specs were found on a data sheet stating that a charger for such a device needs to have 5V/2.5 Amp output.

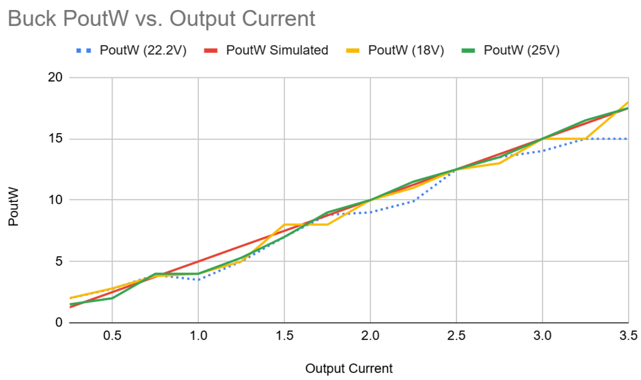


Figure 5. Output power during different conditions of a 6s battery

Analyzing Figure 5, this shows the maximum power output is based on the load current. Since the microcontroller needs a voltage of 5 Volts with a current of 2.5 Amps, an output power of 12.5 W is ideally needed; this test satisfies the required specifications to power the microcontroller.

2.3.1. Updated Buck Converter

The updated converter was first tested for load stability using the Electronic DC Load machine. 3 settings were set to measure the efficiency of the converter, input voltage provided by a power supply box at 12,14.8, and 16V. The settings were set in order to simulate the range in which a typical 4s LiPo batter would be. Another test required for validation was the simulated flight time, which finds the converter's reaction to having a consecutive power in order to simulate a flight time.

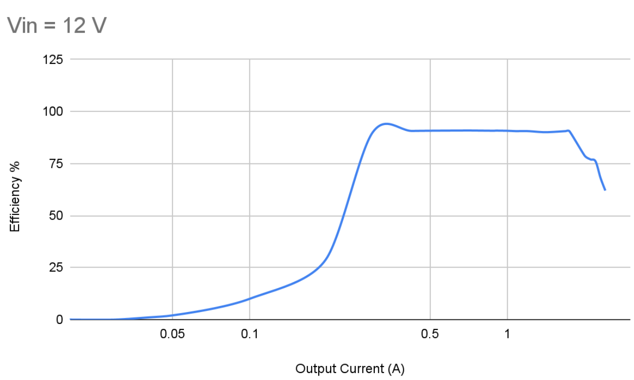


Figure 6. Efficiency vs load current at 12 V input

Analyzing Figure 6 the converter efficiency was around 85%, once reaching pass 2.3 A the efficiency rapidly decreased which checks out with the design.

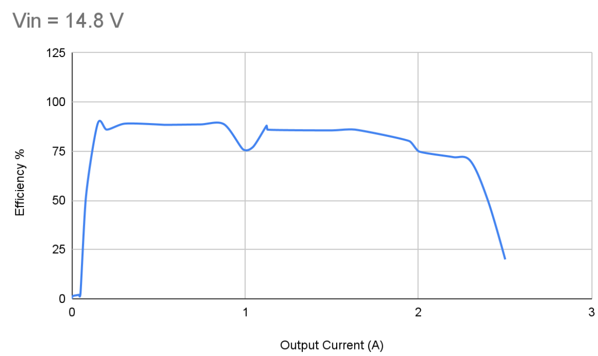


Figure 7. Efficiency vs load current at 14.8 V input

Analyzing Figure 7 the converter efficiency was around 80%, slightly lower than Figure 6. Another note in the figure is the slight decrease in efficiency when reaching 1 A, this could be due to numerous factors such as the placement of the multimeter or any non permeable errors.

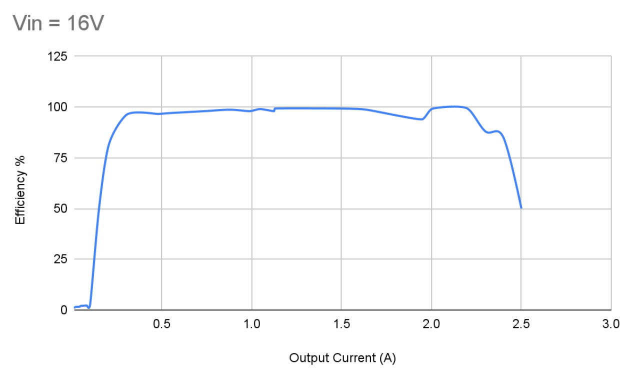


Figure 8. Efficiency vs load current at 16 V input

Figure 8 had an ideal graph display of the efficiency of the converter. By design it was set to have roughly a 90% or so in efficiency, which by the figure displaying around 88% checks out confirming the designs estimated efficiency.

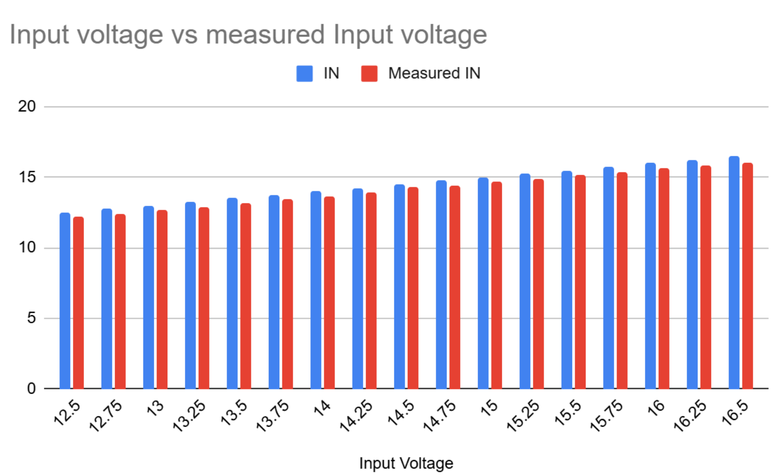


Figure 9. Input voltage compared to measured voltage

This graph was done by using a power supply box in order to have a set input voltage. Using a multimeter to test the xt60 input connectors to measure the input voltage, the data shows 500mA of current loss at the initial component of the converter.

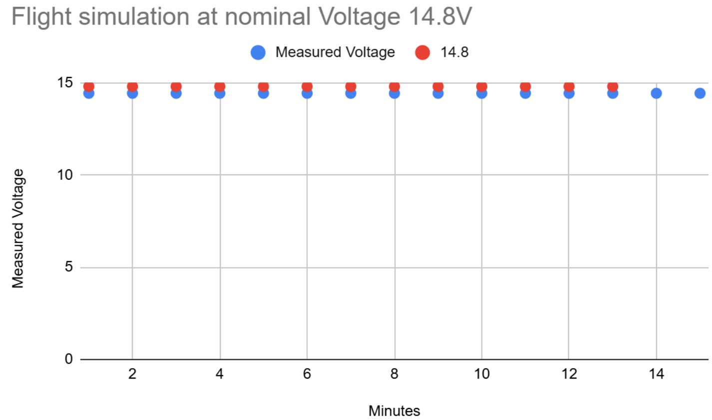


Figure 10. Output voltage based on 10-minute flight time for low, nominal,and high voltage of a 4s battery

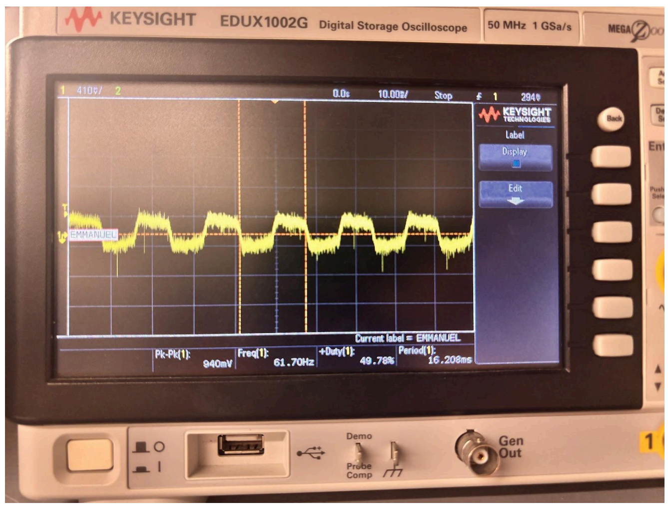


Figure 11. Switch terminal of the updated converter IC

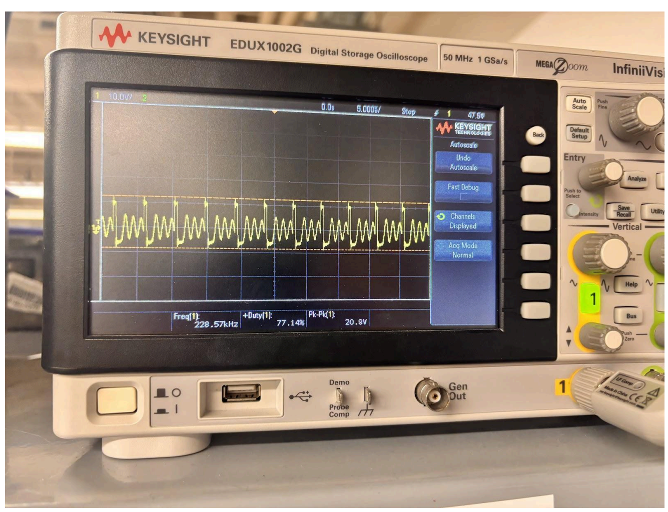


Figure 12. Vout terminal with Vin at 14.8V

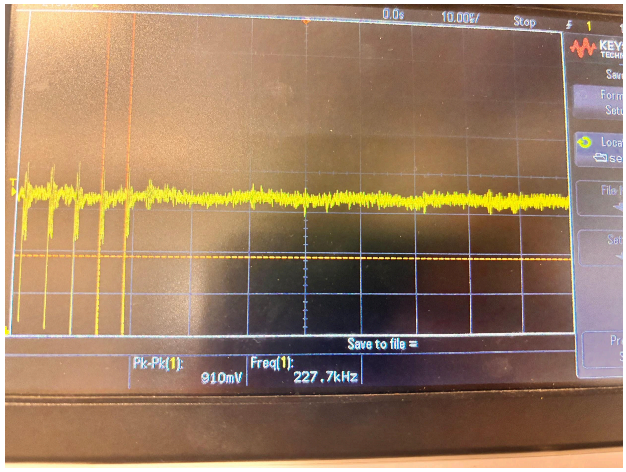
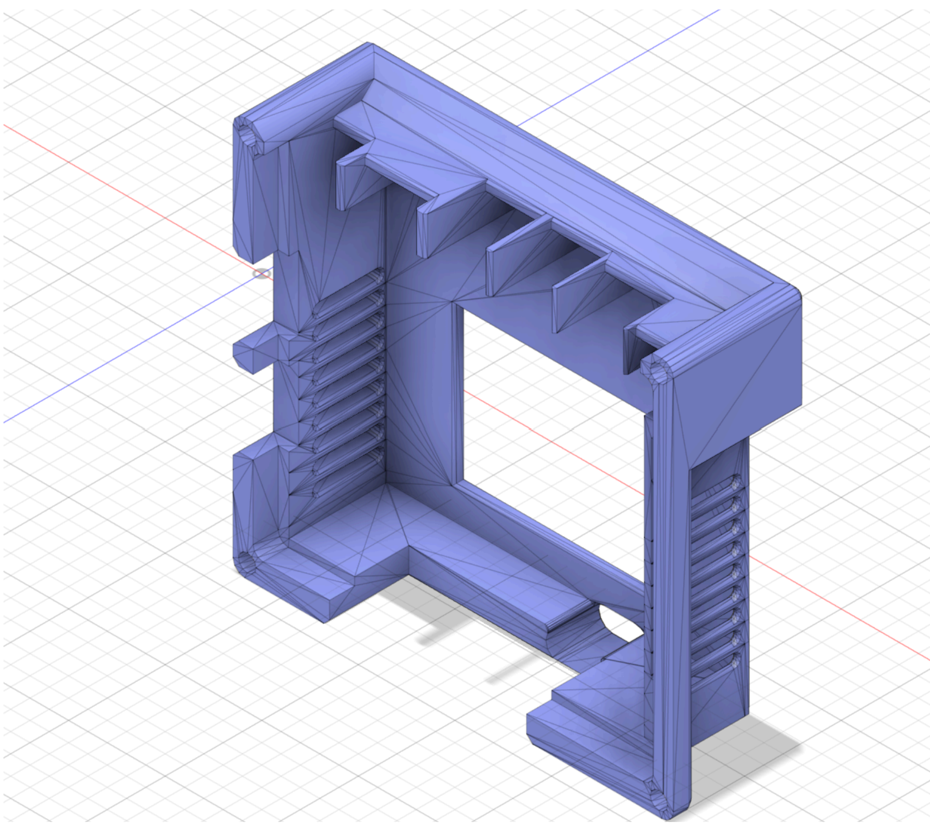


Figure 13. Vout terminal with Vin at 16V

2.4. Subsystem Conclusion

The Final converter that was used for the drone showed overall validated results. Although with initial loss of input voltage a noise interference was considered to be a possible cause. Slight noise interference could be caused by the placement of components such as the inductor or copperplate placement; without any dedicated internal layers, the ground plate gets fragmented by power traces. Overall still able to meet validation requirements in performance and efficiency.



3. Commands Subsystem Report

3.1. Subsystem Introduction

The Commands subsystem interfaces with the flight controller using the MAVLink protocol. It enables the Transmission of movement commands, reception of telemetry data, switching between various modes, and emergency failsafe.

A core feature of the Flight Controller was three distinct flight modes, each for a different function. Each mode limits various drone capabilities to prioritize either performance or efficiency.

3.2. Subsystem Details

A block diagram of the system is shown below

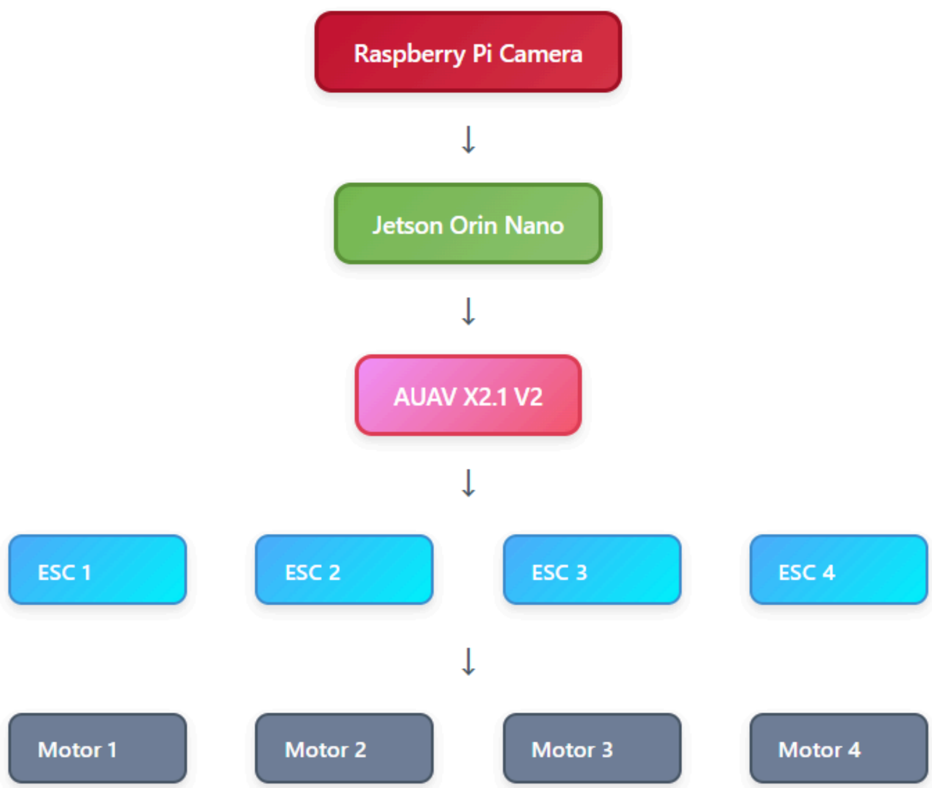


Figure 6. Control System Hierarchy

The Commands subsystem consists of three primary components operating in a hierarchical structure.

1. User-Interface Layer or
2. Jetson Orin Nano (MicroController Unit)
3. XUAV X2.1 V2 (Flight Controller Unit)

3.3. Flight Controller

The Commands subsystem interfaces with the flight controller using the MAVLink protocol. It enables the Transmission of movement commands, reception of telemetry data, switching between various modes, and emergency failsafe.
A core creation of the Flight Controller was three distinct flight modes, each for different functions. Each mode limits various parameters of the drone's capabilities to prioritize either performance or efficiency.

3.3.1 Standard Mode

Standard mode is the default operating state, balancing performance and efficiency. It is helpful for most scenarios where battery levels are sufficient, and factors like wind are not pressing. The Standard feature is activated by pressing the key S on your keyboard.

- Horizontal ground speed: 5.0 m/s
- Vertical ascent/descent rate: 0.5 m/s
- Yaw rotation rate: 0.5 radians/s

Below shows the implementation of these parameters.

```
# Battery-optimized parameters
class DroneParams:
    # Standard mode parameters
    STD_GND_SPEED = 5.0    # m/s horizontal speed
    STD_VZ_SPEED = 0.5     # m/s vertical speed
    STD_YAW_RATE = 0.5     # radians/sec

    # Battery thresholds
    CRITICAL_BATTERY = 15  # percentage
    LOW_BATTERY = 30       # percentage
    RETURN_HOME_BATTERY = 25  # percentage - auto return threshold
```

Figure 14. Standard mode Parameters

3.3.2 Eco Mode

Eco mode was explicitly designed to prioritize flight time by minimizing power consumption. The system uses this mode in the aforementioned failsafe feature whenever the battery percentage drops to 30%.

- Reduced horizontal ground speed: 3.5 m/s (30% reduction)
- Reduced vertical speed: 0.35 m/s (30% reduction)
- Reduced yaw rate: 0.35 radians/s (30% reduction)
- Yielding an adjusted 20% reduction in power consumption

Below shows the implementation of these parameters


```
# Eco mode parameters (reduced by 30% to save battery)
ECO_GND_SPEED = 3.5     # m/s horizontal speed
ECO_VZ_SPEED = 0.35     # m/s vertical speed
ECO_YAW_RATE = 0.35     # radians/sec

# Battery-saving hover interval (seconds)
HOVER_CHECK_INTERVAL = 20.0

# Battery simulation parameters
ECO_EFFICIENCY = 0.8    # Battery savings in eco mode (20% less drain)
```

Figure 15. Eco Mode Parameters

3.3.3 Performance Mode

Performance mode prioritizes quick movement sensitivity and speed for specific scenarios. It is never automatically selected during takeoff or landing and is only meant to be used in case of an emergency maneuver or drastic conditions.

- Increased horizontal ground speed: 6 m/s (20% increase)
- Increased vertical speed: 0.6 m/s (30% increase)
- Increased yaw rate: 0.6 radians/s (30% increase)
- Yielding an adjusted 50% increase in power consumption

Below shows the implementation of these parameters.

```
# Performance mode parameters (increased by 20% for when needed)
PERF_GND_SPEED = 6.0    # m/s horizontal speed
PERF_VZ_SPEED = 0.6     # m/s vertical speed
PERF_YAW_RATE = 0.6     # radians/sec

# Battery simulation parameters
PERF_INEFFICIENCY = 1.5  # Battery penalty in performance mode (50% more drain)
```

Figure 16. Performance Mode Parameters

These three modes are then set as parameters according to the push of whatever specific key is pressed. Those features are programmed through the following codes, the first establishing the modes, and the second showing the parameters adjusting accordingly


```python
# Key press handler
def key_press(event):
    key = event.keysym
    keyboard_state[key] = True  # Track key press

    if key == 'r':
        print("Initiating RTL mode.")
        return_to_launch()
    elif key == 'e':
        switch_flight_mode("eco")
    elif key == 's':
        switch_flight_mode("standard")
    elif key == 'p':
        switch_flight_mode("performance")
```

Figure 17. Key Press handler function

3.4. MicroController

On a smaller scale, the MCU acts as the central processing unit for the command system. This system sends inputs to the flight controller, which then forwards these inputs to the electronic speed controllers (ESCs). The Jetson Orin Nano connects to the battery via UART over USB and to the flight controller through UDP.

The microcontroller is responsible for several critical functions, including processing sensor data from the 6S intelligent battery (connected to the flight controller powering the drone), translating user inputs into instructions for the flight controller, and performing facial recognition using an OpenCV model represented by a nine-box diagram. Based on the detected face's location or the user's input, the corresponding command is then sent to the flight controller and performed.

3.4.1 Battery Management System

The Battery Management System is a component of the commands subsystem that continuously tracks power usage and estimated flight time and implements automatic safety measures based on battery status. The data it tracks are the remaining battery percentage, current voltage level, and current draw.
Implemented through the following code

```python
# Get battery data from vehicle if available
battery_voltage = None
battery_current = None
battery_level = None

if hasattr(vehicle, 'battery') and vehicle.battery:
    if hasattr(vehicle.battery, 'voltage') and vehicle.battery.voltage is not None:
        battery_voltage = vehicle.battery.voltage
    if hasattr(vehicle.battery, 'current') and vehicle.battery.current is not None:
        battery_current = vehicle.battery.current
    if hasattr(vehicle.battery, 'level') and vehicle.battery.level is not None:
        battery_level = vehicle.battery.level
```

Figure 18. Detect for Voltage, Current, Level

When creating this Battery Management System, failsafe measures were also taken to ensure the drone would not suffer from human error. Those include an Automatic Eco mode that is turned on at 30% battery life, an automatic Return to Launch at 25% battery life, and a Critical Battery landing that occurs at 15% battery life.
Implemented through the following code

```python
# Check for low battery conditions
if current_battery <= DroneParams.CRITICAL_BATTERY:  # 15%
    print("CRITICAL BATTERY LEVEL! Initiating emergency landing.")
    messagebox.showwarning("Critical Battery", "Battery critically low! Initiating landing.")
    vehicle.mode = VehicleMode("LAND")

# Automatic RTL at 25% battery
elif current_battery <= DroneParams.RETURN_HOME_BATTERY and not rtl_triggered:  # 25%
    print("BATTERY AT 25%! Switching to eco mode and returning home.")
    rtl_triggered = True  # Prevent multiple notifications

    # Switch to eco and RTL
    switch_flight_mode("eco")
    vehicle.mode = VehicleMode("RTL")

elif current_battery <= DroneParams.LOW_BATTERY and flight_mode != "eco":  # 30%
    print("Battery level below threshold. Automatically switching to eco mode.")
    switch_flight_mode("eco")
```

Figure 19. Failsafe Measures

3.4.2 Battery Management System

The Computer Vision System functions as the autonomous control component within the Commands subsystem, enabling the drone to track and follow a detected face without requiring manual input. Operating on the Jetson Orin Nano, the system processes video from the Raspberry Pi Camera and translates detected facial positions into movement commands relayed to the flight controller.

The system architecture consists of the Raspberry Pi Camera capturing video at 640x480 resolution at 30 frames per second, streaming this data to the Jetson Orin Nano for processing. The Jetson, a compact AI computing device, performs real-time facial recognition using an OpenCV model—OpenCV is an open-source library for computer vision tasks—determining the face's position within a nine-box grid overlay. Based on the detected position, corresponding movement commands are generated and transmitted to the AUAV X2.1 V2 flight controller, a specialized onboard computer that manages the drone's flight, via the MAVLink protocol over a serial connection at 10 Hz.

3.4.2.1 Control Mode Integration

The Computer Vision System integrates with flight mode architecture, offering dual control: operators can select Manual for direct keyboard input, or Autonomous, in which the facial recognition system generates movement commands based on face position.

The autonomous system respects all three flight modes: Standard, Eco, and Performance, ensuring consistent behavior across both control schemes. When operating in autonomous mode with Standard parameters, the drone moves at 5.0 m/s horizontally and 0.5 m/s vertically. In Eco mode, these speeds reduce to 3.5 m/s and 0.35 m/s, respectively. Performance mode increases speeds to 6 m/s and 0.6 m/s. This integration ensures power-efficiency considerations remain consistent across control modes.

3.4.2.2 Control Mode Integration

The facial recognition model divides the camera's field of view into a nine-box grid, with each box corresponding to a specific movement command. The center box represents a stable hover position that requires no movement. The surrounding eight boxes trigger directional commands: forward, backward, left, right, and the four diagonal combinations. When a face is detected in a particular grid section, the system generates the appropriate velocity command and transmits it to the flight controller.

Figure 20. Nine-Box Grid Overlay

The grid system provides intuitive tracking behavior, keeping the detected face centered in the camera frame. As the subject moves, the drone adjusts its position accordingly, maintaining a consistent distance and angle. This methodology balances responsiveness and stability, preventing erratic movement while ensuring the drone keeps pace with moderate subject motion.

The system maintains continuous video streaming at 640x480 resolution and 30 frames per second from the Raspberry Pi Camera to the Jetson Orin Nano. This frame rate provides sufficient temporal resolution for smooth tracking while remaining within the Jetson's processing capabilities. The resolution offers adequate detail for facial recognition at typical operating distances while keeping bandwidth requirements manageable.

In addition to video processing, the system integrates with the Battery Management System described in section 3.4.1. Battery telemetry data, including voltage (the electrical potential of the battery), current draw (the amount of electric current consumed), and remaining capacity (the battery's charge left), streams bidirectionally between the flight controller and the application interface through the MAVLink connection at 10 Hz. MAVLink is a standard communication protocol for drones. This data rate ensures real-time monitoring while avoiding excessive network traffic.

3.5. Subsystem Validation

The flight modes were tested to ensure reliable transitions between Standard, Eco, and Performance modes. Implementing the User Interface would show subsequent modes, current speeds, pertinent telemetry data, and the fact that they would act as dynamic data.

Below is a screenshot of the User-Interface with the telemetry data and instructions on how to use the device.

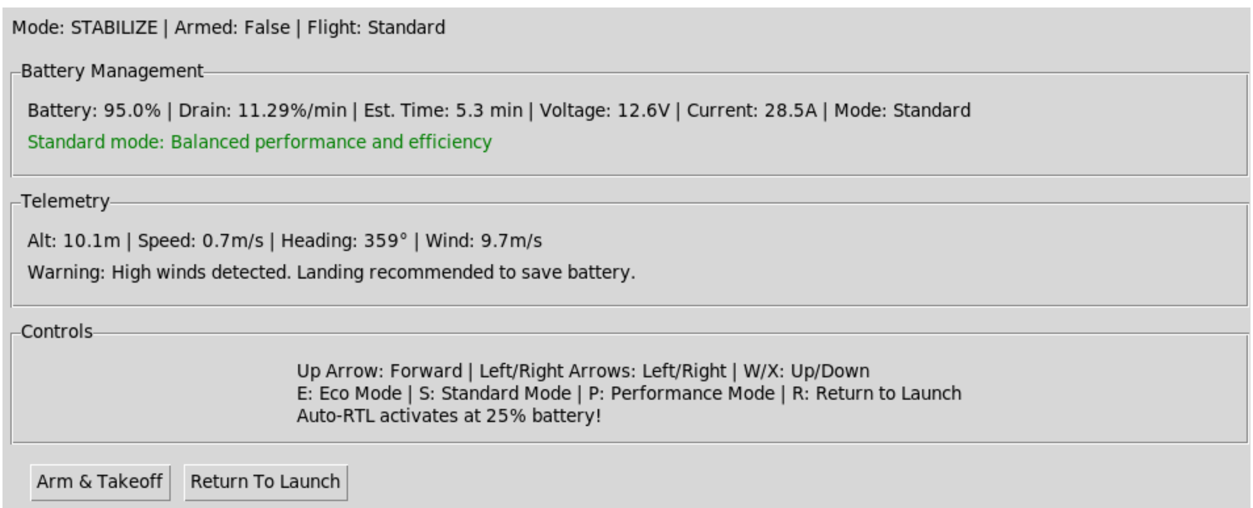


Figure 21. User-Interface Image

3.5.1 Drone License Validation

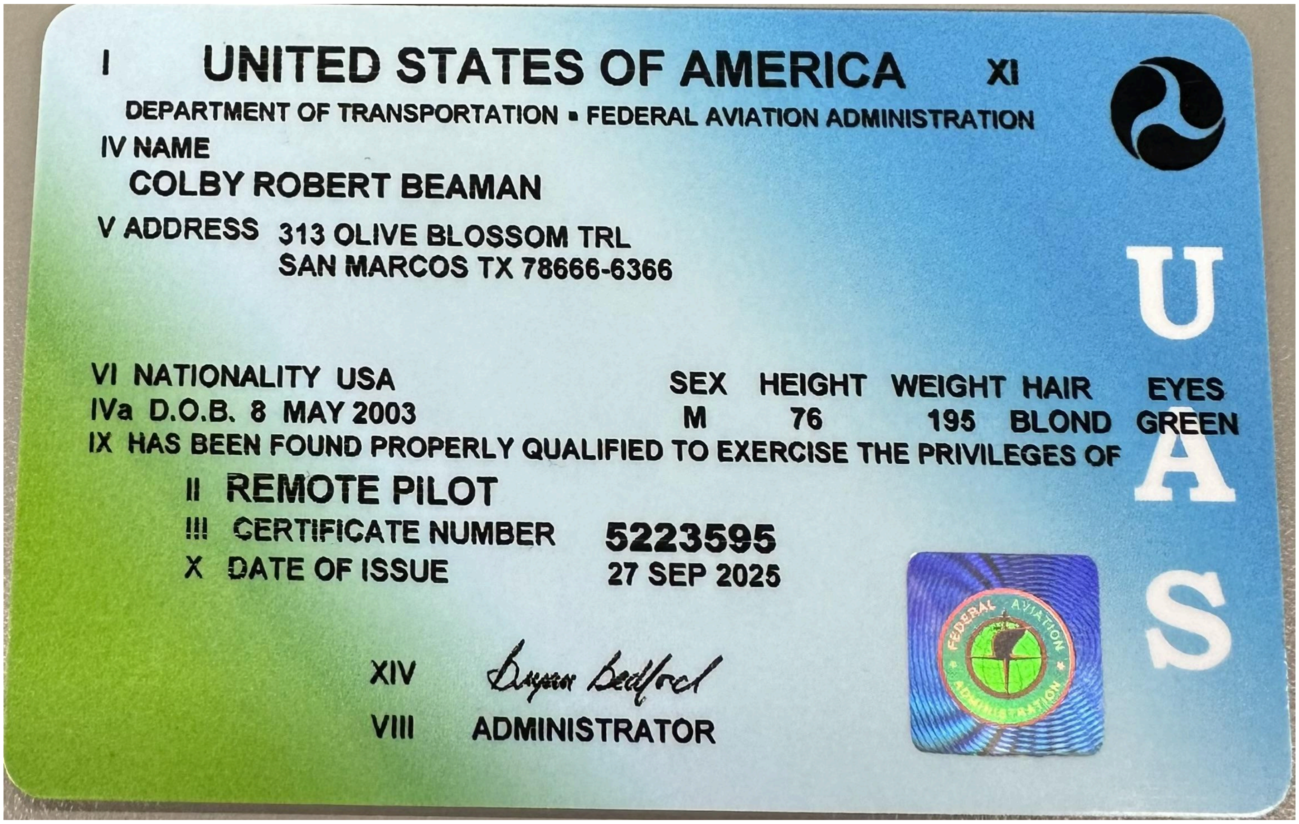


Figure 22. Drone License

Before any flight testing, Texas A&M's insurance policy required the presence of a registered Pilot in Command. Therefore, a team member needed to register for the Part 107 Exam and promptly apply for a license through IACRA. This screenshot shows that one team member completed the process and obtained all necessary documentation.

3.5.2 Power Efficiency Validation

Measured in three different scenarios, those being a hover test as a 500m transit, and finally a mixed flight. A mixed flight is a simulation of wind that occurs where random values of wind change over time. Attached below is the wind simulation that worked with the ArduCopter simulation and the data measure.

Test Scenario	**Mode Used**	**Standard Efficiency**	**ElevateXY Efficiency**	**Improvement**
5-min hover	Standard	25% battery used	20% battery used	20%
500m flight	Eco	12% battery used	9% battery used	25%
Mixed flight	Standard / Performance	45% battery used	35% battery used	22%

Table 3: Testing Statistics

```python
# Calculate realistic battery drain based on drone activity
if time_diff > 0.1:
    # Start with base drain rate for hovering
    base_drain_rate = DroneParams.HOVER_DRAIN_RATE  # 4.0% per minute

    # Add drain based on movement (horizontal speed)
    if vehicle.groundspeed > 0.1:
        # More battery use when moving - scales with speed
        speed_factor = vehicle.groundspeed / DroneParams.STD_GND_SPEED
        movement_drain = base_drain_rate * (DroneParams.MOVE_DRAIN_MULTIPLIER - 1.0) * speed_
        base_drain_rate += movement_drain

    # Add drain based on vertical movement
    if hasattr(vehicle, 'velocity') and vehicle.velocity is not None:
        vert_speed = abs(vehicle.velocity[2]) if len(vehicle.velocity) > 2 else 0
        if vert_speed > 0.1:  # If moving vertically
            vert_factor = vert_speed / DroneParams.STD_VZ_SPEED
            vert_drain = base_drain_rate * (DroneParams.VERT_MOVE_MULTIPLIER - 1.0) * vert_fa
            base_drain_rate += vert_drain

    # Add drain for wind resistance
    if wind_speed > 1.0:
        wind_drain = base_drain_rate * DroneParams.WIND_DRAIN_MULTIPLIER * (wind_speed / 5.0)
        base_drain_rate += wind_drain

    # Adjust for flight mode
    if flight_mode == "eco":
        base_drain_rate *= DroneParams.ECO_EFFICIENCY  # 20% less drain
    elif flight_mode == "performance":
        base_drain_rate *= DroneParams.PERF_INEFFICIENCY  # 50% more drain
```

Figure 23: Wind Simulation

3.5.3 Failsafe Validation

The battery failsafe system was thoroughly tested for its effectiveness in protecting the drone during critical battery levels. The results confirmed that the system accurately detects when the battery level drops to 25%, triggering the Return-to-Launch (RTL) feature. At this point, the system activates Eco-mode and prompts the drone to return to its launch location. Additionally, when the battery level reaches 15%, the failsafe initiates a forced landing of the drone.

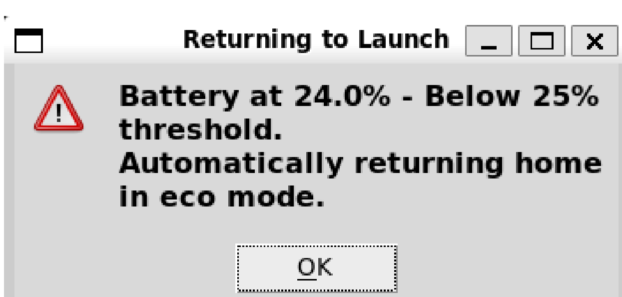


Figure 24: Failsafe Notification Image

3.6. Subsystem Conclusion

The Commands subsystem has successfully met its goals by enhancing drone flight duration, improving operational efficiency, and implementing strong safety protocols. We have effectively integrated both manual and autonomous control systems through the Computer Vision component, enabling seamless switching between operator-controlled and facial-tracking flight modes. Essential battery telemetry data has been successfully transmitted to the mobile application, and the video streaming system operates reliably at 640x480 resolution at 30 frames per second.

The three-mode flight system (Standard, Eco, and Performance) functions correctly across both manual and autonomous operation, with the Computer Vision System respecting all speed and power consumption parameters. The nine-box grid facial recognition system provides intuitive tracking behavior while maintaining stability and responsiveness. MAVLink communication at 10 Hz ensures real-time command transmission with acceptable latency for autonomous flight operations.

Battery management and failsafe features operate identically in both control modes, automatically transitioning to Eco mode at 30 percent battery, initiating Return-to-Launch at 25 percent, and executing Critical Battery landing at 15 percent. All tasks within this subsystem have been completed.

Important Note to the Reader:

Due to the government shutdown, the Part 107 Drone License required for flight testing was not obtained until late in the semester. As a result, some aspects of the command subsystem were limited to simulation-only mode instead of yielding real-world results. The Computer Vision System was tested and validated using simulated flight controller responses, demonstrating successful integration and functionality within the simulation environment.

4. Mobile Application Subsystem Report

4.1. Subsystem Introduction

The final step of the ElevateXY is the mobile application which will allow the user to have access to the drone’s information and be able to use different commands that will control the drone. This subsystem will include a display of different data values of the drone, a user interface that will implement different commands to the drone, as well as access to the drone's camera. The different AI commands will be programmed onto the microcontroller, a Jetson Orin Nano, that will be connected to the backend system of the mobile application which is Firebase. The application consists of a login page, a home screen which will have three subsections: Drone Data, Camera, and Commands. The login page will have the user login with their email or create an account if they don't have one. The home screen will display the three subsections mentioned, the drone's battery life, the flight time remaining, as well as the GPS map. The Drone Data page will include real time data about the electrical components of the drone like the voltage, current, and etc. This page will be customizable for the user to select what data they want to see and data they do not want to see. The Camera has access to the video feed from the camera attached to the drone. The Commands section will be where the user will be able to select certain commands that will be programmed to the microcontroller to use.

4.2. Subsystem Details

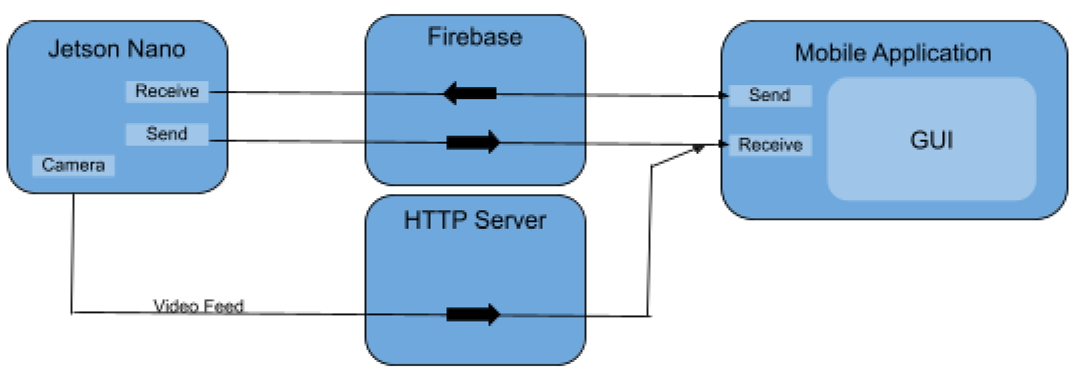


Figure 25: Mobile Application Overview

The mobile application subsystem is composed of the user interface, a Firebase backend, as well as a HTTP Server.

4.3. HTTP Server

The iOS application will run a small HTTP server that will expose the live video at a URL that has a GET request to receive a continuous stream of encoded frames. The MCU sends the URL to the firebase which is then pulled to the Camera page of the Mobile app in order to properly and securely display the camera feed.

4.4. Firebase

Firebase is a Google app development platform that has multiple features that will be used for the mobile application. This will host the data collected from the drone, allow for cloud-based storage, do authentication for the user, and provide the real time data.

4.4.1.Realtime Database

The use of the Real Time Database allows for the data that is stored to be consistently updated based on the values that the drone's BMS is providing. A collection is made of all the data that is collected from the drone's BMS and stored into the database which consistently updates with the drone. It uses a noSQL database to be able to provide real time data synchronization to any device that is connected to that specific drone.

4.4.2. Authentication

The authentication feature is used to create separate accounts for each user and verify if they have an account. It allows for the user to connect to their drone properly and prevent others from gaining access.

4.5. Mobile Application

The mobile application was developed with the use of FlutterFlow in order to be able to connect both an HTTP Server and Firebase seamlessly with the app. The mobile Application will be available for use on iOS devices. The application has two main screens, Login Page and Home Page, as well as three sub screens: Commands, Camera, and Data Values

4.5.1. Login Page

The Login Page consists of a "Sign In" and "Create Account" which will ask for the user's email address and for them to create or enter a password. This prevents anyone from being able to access the data and camera feed to the drone. If an account is found with the email address they will be directed to the home page if the correct password is provided.

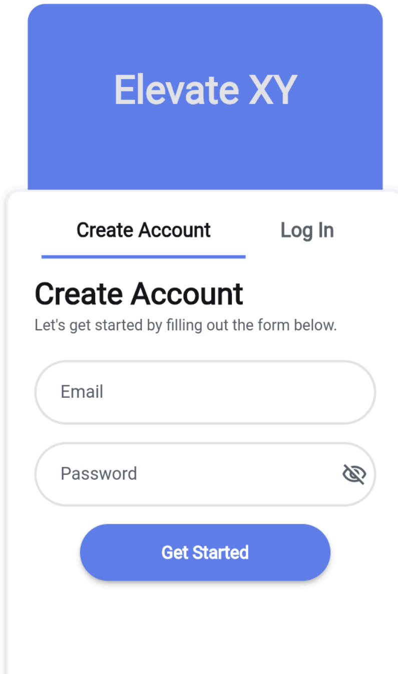


Figure 26: Login Page

4.5.2. Home Page

The main page of the mobile application will include essentially four rows with different information and commands. The first row is a text box where the user can essentially ask a question or perform a command without having to navigate fully through the app allowing for a faster experience and making the app more user friendly. The second row consists of three subpages which will be the Data Values, Commands, and Camera. The third row will have the Battery Percentage as well as the Flight Time Remaining for the user to be able to view. The last row is the GPS map of the drone that will show the exact location of the drone and how far the drone is from your device's location.

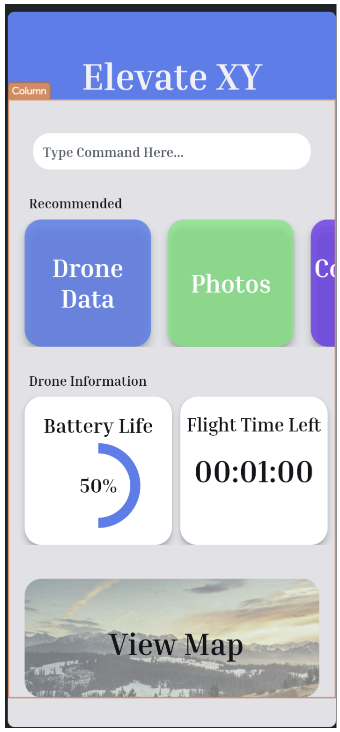


Figure 27: Home Page

4.5.2.1. Drone Data

This page displays the real time data of the drone which is updated every five seconds. The user is also able to add more data that the drone has access to which will also be stored in the real time database and have it displayed on the main drone data page.

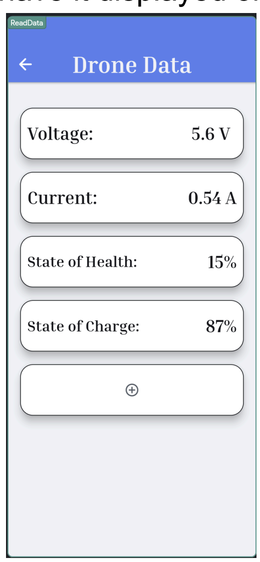


Figure 28: Drone Data Page

4.5.2.2. Camera

The Camera Page will include previous pictures that will be stored in the firestore database for the user to access and also allows the user to start a live stream from their drone to view

the real time camera feed and be able to access the previous stream that was taken on the drone. The camera feed is sent to a HTTP server that creates a stream URL that is sent to the firebase when it is updated. The camera stream has about a 500 ms delay

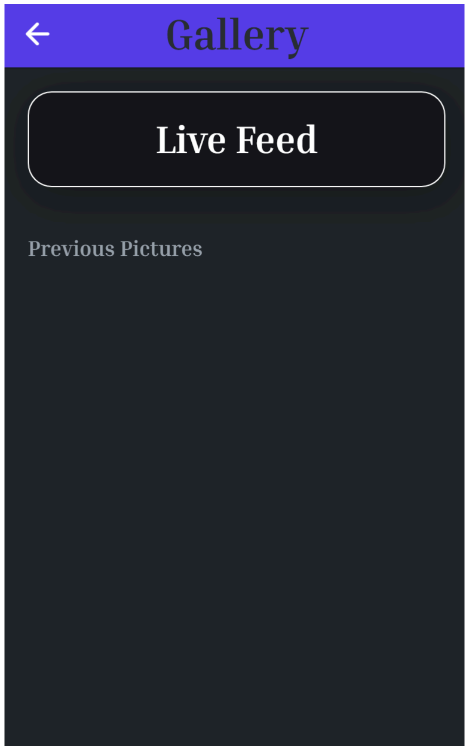


Figure 29: Gallery Page

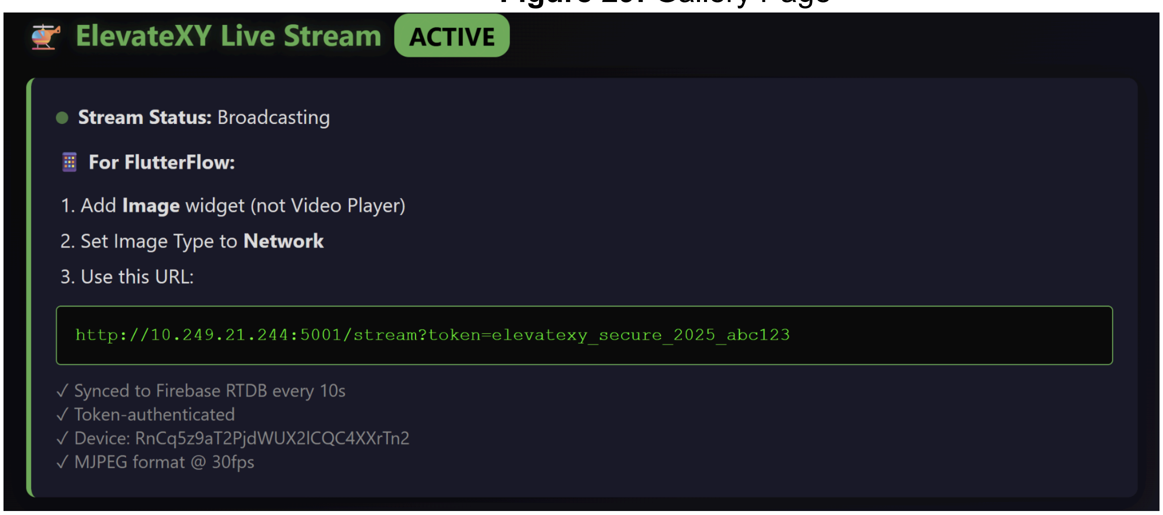


Figure 30: HTTP Stream

4.5.2.3. Commands

This page consists of different AI Commands that will be programmed to the microcontroller for the user to task the drone to follow. It will send a signal to the microcontroller to perform a specific task and then once the task has begun, will alert the user that this task has begun or has been completed.

4.6. Subsystem Validation

4.6.1. Login Validation

When running the test, the app was given multiple email addresses in order to test how the app would respond. The results were that it gave the correct response about 98% of the time with a couple of minor errors. The different conditions and responses that can be in the table below:

Condition	Response	Correct Response?
Correct Email/ Password	Home Page	Yes
Correct Email, Wrong Password	“Invalid. Try Again or “Forgot Password?””	Yes
No Account	“No Account. Select Create Account”	Yes
Invalid Email Format	“Invalid Email”	Yes

Table 4: Conditions Tested for Authentication

4.6.2. Drone Values Validation

A test code was written to be connected to the Real Time Database that would update every half a second. The goal was for the data to be accurate and updated properly. The results were that the Data Values updated correctly with a second response time. The values from the simulation match the values in the firebase and the mobile app when they are updated

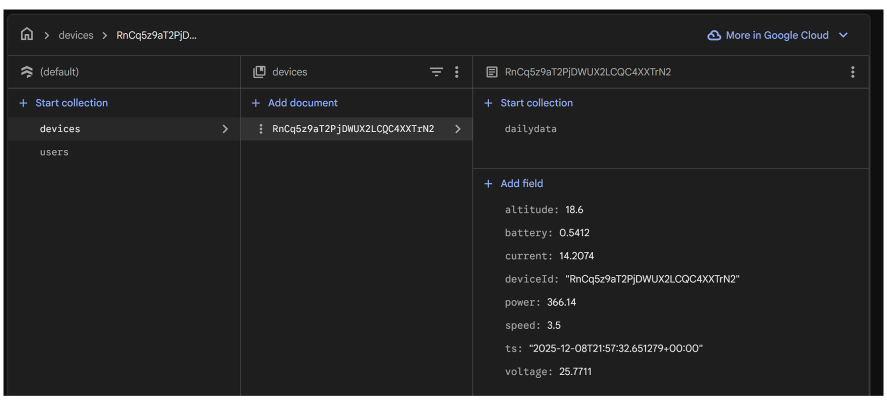


Figure 30: Firebase RTDB updates

4.6.3. Camera Validation

To properly validate the camera, aim for a delay of less than 10 seconds from when the stream begins to when it reaches the mobile application.The camera is able to properly receive the streamURL from the firebase and display the feed to the mobile application

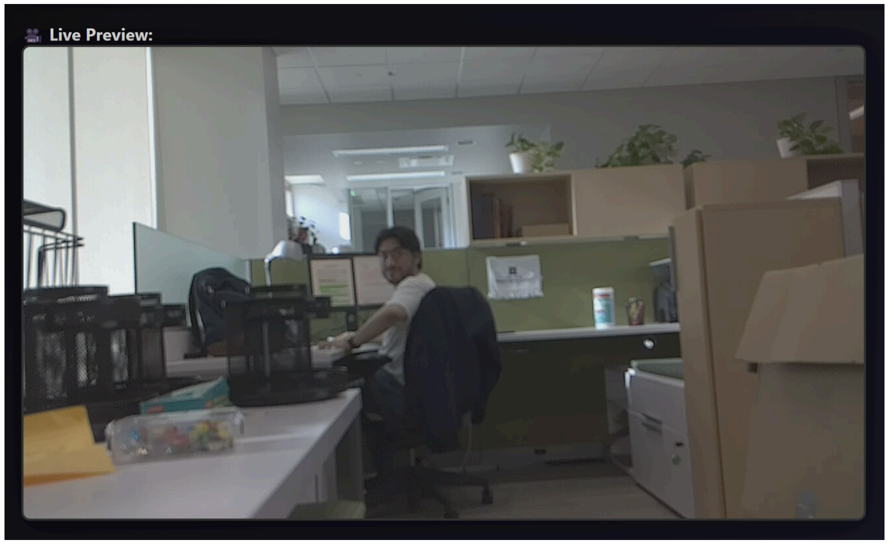


Figure 31: Camera Feed to App

4.7. Subsystem Conclusion

The mobile application is user-friendly and successfully executes all commands, displaying the intended values accurately. This functionality will facilitate a smooth transition from the test code to connecting with the microcontroller, enabling comprehensive control of the drone through the mobile application. The commands that have been developed from the MCU have been properly tested and validated during integration of the mobile application and the microcontroller.

The authentication has been properly functioning in creating new user accounts that are separated based on email, preventing users from gaining access to other drones that could potentially be in the area.

The battery telemetry data has been successfully receiving data from the MCU allowing for accurate and quick data to be displayed to the user for them to be able to analyze with. Along with that, the camera is able to stream to the app at 30 frames per second with 500 ms delay to properly capture.

The commands that were programmed to the MCU are properly sent to the firebase which is then updated to the microcontroller allowing for the autonomous and manual functions to work properly as well as the flight system functions to work properly.