

# **DESIGN AND FABRICATION OF AUTONOMOUS DRONE DELIVERY SYSTEM**

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in  
Aeronautical Engineering

By

**Dinesh Sarangi (40260008)**

**Yenuganti Hanish Netha (40260025)**



**DEPARTMENT OF AERONAUTICAL ENGINEERING  
SCHOOL OF MECHANICAL ENGINEERING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND  
TECHNOLOGY(DEEMED TO BE  
UNIVERSITY)**

**Accredited with grade “A++” by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119**

**APRIL - 2024**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

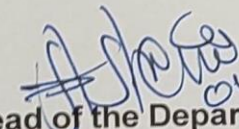
## DEPARTMENT OF AERONAUTICAL ENGINEERING

### BONAFIDE CERTIFICATE

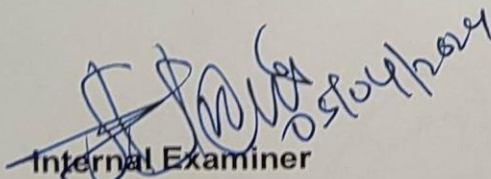
This is to certify that this Project Report is the bonafide work of **Dinesh Sarangi (40260008)** and **Yenuganti Hanish Netha (40260025)** who carried out the project entitled "**Design and Fabrication of Autonomous Drone Delivery System**" under my supervision from November 2023 to April 2024.

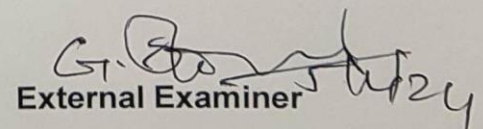
 04/04/2024  
Internal Guide

Dr. A. ANDERSON., M.E., Ph.D.,

 04/04/2024  
Head of the Department  
Dr. A. ANDERSON., M.E., Ph.D.,

Submitted for Viva Voce Examination held on 05-04-2024

 05/04/2024  
Internal Examiner

 05/04/2024  
External Examiner

## DECLARATION

We **DINESH SARANGI (40260008)** and **YENUGANTI HANISH NETHA (40260025)** hereby declare that the project report entitled "**DESIGN AND FABRICATION OF AUTONOMOUS DRONE DELIVERY SYSTEM**" Done by us under the guidance of **Dr. A. ANDERSON M.E., Ph.D.** is submitted in partial fulfillment of the requirement for the award of Bachelor of Engineering degree In Aeronautical Engineering.

Date: 05-04-2024

Place: Chennai

*Dinesh Sarangi*

1) **DINESH SARANGI (40260008)**

*Y. Hanish Netha*

2) **YENUGANTI HANISH NETHA (40260025)**

**SIGNATURE OF THE CANDIDATES**

## **ACKNOWLEDGEMENT**

We are pleased to acknowledge our sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our sincere thanks to **Dr. A. Anderson, M.E., Ph.D.**, Head of the Department, Department of Aeronautical Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to my Project Guide **Dr. A. Anderson M.E., Ph.D.**, for his valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of our project work.

We also express our thanks to all Teaching and Non-teaching staff members of the Department of Aeronautical Engineering who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Autonomous delivery drones are gaining prominence in transforming logistics, offering efficient, swift, and eco-friendly solutions for last-mile deliveries. However, there is a lack of literature focusing on the collaborative efforts of multiplerotor drones in establishing long-range connectivity for both air-to-air and air-toground operations. The practical implementation depends on overcoming challenges related to battery optimization, network connectivity, and navigation systems. The optimization of batteries is a critical concern, aiming to extend flight times and payload capacities while ensuring safety. Network connectivity is equally crucial for the reliable functioning of autonomous delivery drones, necessitating effective communication and data transmission between the drone, ground control stations, and the broader network infrastructure. The communication between the drone's flight controller's microprocessor and the ground station receiver engages network connectivity for a mission. Navigation System, Innovations in mesh networks, low Earth orbit (LEO) satellites communicate with the inbuilt drone's GPS that enhances the guidance, connectivity, and data transfer of the live location.

Keywords—Delivery Drones, Battery Optimization, Network Connectivity, Navigation System, Data Transfer

## TABLE OF CONTENT

| CHAPTER NO | TITLE                                             | PAGE NO   |
|------------|---------------------------------------------------|-----------|
|            | <b>ABSTRACT</b>                                   | i         |
|            | <b>LIST OF FIGURES</b>                            | iv        |
| <b>1</b>   | <b>INTRODUCTION</b>                               | <b>1</b>  |
|            | 1.1 INTRODUCTION                                  | 1         |
| <b>2</b>   | <b>LITERATURE SURVEY</b>                          | <b>4</b>  |
|            | 2.1 GENERAL                                       | 4         |
|            | 2.2 REVIEW OF THE LITERATURE                      | 4         |
|            | 2.3 NETWORK CONNECTIVITY                          | 7         |
|            | 2.4 POWER SOURCE OF AN UAV                        | 9         |
|            | 2.4.1 Hybrid Propulsion                           | 9         |
|            | 2.4.2 Battery Powered                             | 9         |
|            | 2.4.3 Swapping                                    | 10        |
|            | 2.4.4 Fuel Cell Power                             | 11        |
|            | 2.5 DESIGN CUSTOMIZATION IN DELIVERY              | 12        |
| <b>3</b>   | <b>AIM AND SCOPE OF THE PRESENT INVESTIGATION</b> | <b>14</b> |
|            | 3.1 AIM                                           | 14        |
|            | 3.2 SCOPE OF THE PRESENT INVESTIGATION            | 14        |
|            | 3.2.1 Last-Mile Delivery Revolution               | 14        |
|            | 3.2.2 E-commerce and Retail                       | 14        |
|            | 3.2.3 Medical Supply Transport                    | 15        |
|            | 3.2.4 Food and Grocery Delivery                   | 15        |
|            | 3.2.5 Disaster Relief and Humanitarian Aid        | 15        |
|            | 3.2.6 Agriculture and Precision Farming           | 15        |
|            | 3.2.7 Environmental Monitoring                    | 15        |
|            | 3.2.8 Infrastructure Inspection                   | 15        |
|            | 3.2.9 Security and Surveillance                   | 15        |
|            | 3.2.10 Customized Design                          | 15        |
|            | 3.2.11 Regulatory and Safety Compliance           | 16        |

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
|          | 3.2.12 Regulatory and Safety Compliance                    | 16        |
| <b>4</b> | <b>MATERIAL AND METHODOLOGY</b>                            | <b>17</b> |
|          | 4.1 MATERIAL                                               | 17        |
|          | 4.1.1 Frame of UAV                                         | 17        |
|          | 4.1.2 Motor of UAV                                         | 17        |
|          | 4.1.3 Electronic Speed Controller (ESC)                    | 18        |
|          | 4.1.4 Flight Control                                       | 18        |
|          | 4.1.5 Propeller                                            | 19        |
|          | 4.1.6 Radio Transmitter                                    | 19        |
|          | 4.1.7 Battery, El electronics power distribution<br>cables | 20        |
|          | 4.1.8 Camera                                               | 21        |
|          | 4.1.9 Landing Gear                                         | 22        |
|          | 4.2 METHODOLOGY                                            | 22        |
|          | 4.3 WORKING                                                | 23        |
| <b>5</b> | <b>ALGORITHMS</b>                                          | <b>25</b> |
| <b>6</b> | <b>RESULTS AND DISCUSSION</b>                              | <b>76</b> |
|          | 6.1 RESULT                                                 | 76        |
|          | 6.1.1 Optimizing battery life                              | 76        |
|          | 6.1.2 Handling Human Interaction                           | 76        |
|          | 6.1.3 Data Transmission                                    | 76        |
|          | 6.2 DISCUSSION ON FUTURISTIC SCOPE                         | 77        |
| <b>7</b> | <b>SUMMARY AND CONCLUSION</b>                              | <b>78</b> |
|          | 7.1 SUMMARY                                                | 78        |
|          | 7.2 CONCLUSION                                             | 78        |
|          | <b>REFERENCES</b>                                          | <b>79</b> |

## LIST OF FIGURES

| FIGURE NO | TITLE                                   | PAGE NO |
|-----------|-----------------------------------------|---------|
| 2.1       | Hybrid Propulsion Based UAV             | 9       |
| 2.2       | Battery-Powered UAV                     | 10      |
| 2.3       | Swapping Technique                      | 10      |
| 2.4       | Working Principle of Swapping Technique | 11      |
| 2.5       | Fuel Cell Powered UAV                   | 12      |
| 4.1       | Frame of a Quadcopter                   | 17      |
| 4.2       | Brushless Motor Used In Our Application | 18      |
| 4.3       | Electronic Speed Controller (ESC)       | 18      |
| 4.4       | Flight Control                          | 19      |
| 4.5       | Propeller                               | 19      |
| 4.6       | Radio Transmitter                       | 20      |
| 4.7       | Battery used in UAVs                    | 20      |
| 4.8       | Electronic components used in UAVs      | 21      |
| 4.9       | Camera                                  | 21      |
| 4.10      | Landing Gear                            | 22      |
| 4.11      | Working Principle of Drone              | 24      |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

Delivery systems are integral to the contemporary economy as they facilitate the transportation of goods and services from one location to another. These systems can range from being straightforward to intricate, but all of them play a crucial role in our daily lives. Drones, categorized as Unmanned Aerial Vehicles (UAVs), are autonomous flying machines that are operated remotely, without the need for a human pilot onboard. These vehicles come in various designs, and sizes, and serve different purposes. Key components of drones include motors, a central processing unit, propellers, a frame, and a power source, typically a battery. Users can control drones either using remote controllers or by pre-programming them with GPS coordinates and flight paths. Most contemporary drones are equipped with integrated GPS systems to ensure precise location tracking during their flights.

Lately, the last segment of supply and distribution chains, referred to as 'last-mile delivery,' has emerged as a notable challenge due to the proliferation of innovative delivery methods. Businesses are adopting increasingly inventive strategies for transporting goods, and with the growing prevalence of just-in-time management, delivery services are in high demand. Delivery drones find applications in various industries such as e-commerce, medical supply distribution, food delivery, disaster relief, agriculture, environmental monitoring, and others that are undergoing transformative changes. A notable advantage of delivery drones lies in their ability to transport goods swiftly and efficiently, reducing the reliance on human delivery personnel and potentially mitigating traffic congestion. These drones excel in the delivery of small, lightweight items like medications, food, and consumer products.

Prominent companies like Amazon, UPS, and Alphabet have been actively advancing the field of drone-based delivery services. According to reports, Flight-Forward, a subsidiary of UPS, has received certification as an air carrier for transporting small packages via drones. In a significant milestone, Alphabet's

division, Wing, launched the United States' first small-scale commercial delivery service in Christiansburg, Virginia, in late 2019. These developments highlight the global momentum within the drone delivery market. In late 2020, the National Aviation Agency (ANAC) of Brazil granted its first authorization to a private enterprise, Speed Bird, to conduct drone cargo testing in urban areas of Brazil.

However, drones present several advantages over alternative vehicles, such as cost-efficiency, the ability to operate in confined spaces without the need for expensive infrastructure, and the capability for autonomous operation. These advantages render drones highly valuable in a range of transportation and logistics tasks, including, but not limited to, traffic monitoring. Currently, delivery drones find application in international border scenarios, while their use for intra-city package delivery remains a subject of scrutiny.

Despite these advantages, three primary limitations are affecting the effectiveness of drone-based delivery systems. First and foremost, restrictions exist concerning the size and weight of payloads that drones can carry. Second, the limited capacity of current battery technology places constraints on the duration of drone flights, necessitating frequent recharging or battery replacement. Lastly, existing technology and regulations impose restrictions on the dynamic and coordinated control of multiple drone delivery models.

A drone can transport a load, which could be a camera, sensor, or package, to execute a range of functions. The particular use of a drone dictates the payload it carries, the sensors it uses, and its electronic components. The duration of flight and battery performance is influenced by the weight of the payload and prevailing weather conditions. Drones are classified based on criteria such as size, weight, autonomy, range, number of motors, types of sensors, battery capacity, and speed. These characteristics define their capabilities and potential applications, spanning from compact, self-navigating indoor drones to large, multi-motor, long-range drones suitable for various tasks.

The application deals with Data Transmission, the process of transferring digital information from one point to another. It involves encoding, sending, receiving, and decoding data for communication, information exchange, or system operation. Various methods, including wired and wireless technologies,

transmit data, enabling modern communication systems, the internet, and electronic devices. Reliable data transmission is essential for real-time interactions, data sharing, and the seamless operation of technology in our interconnected world.

Battery optimization aims to extend the lifespan and performance of rechargeable batteries. Techniques include efficient charging methods, reducing power consumption during operation, and managing battery temperature. These strategies improve battery efficiency, reduce environmental impact, and enhance the endurance of devices such as smartphones, laptops, electric vehicles, and drones, enabling them to operate longer on a single charge while minimizing the need for battery replacement.

Design customization of drones involves tailoring key aspects to meet specific mission requirements. This includes modifying payload mounts, propulsion systems, and control interfaces. Customization enhances drone versatility and performance for diverse applications like aerial photography, surveillance, agriculture, and delivery services. It allows operators to optimize drones for their unique needs, ensuring they can efficiently and effectively accomplish tasks ranging from data collection to cargo transport.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 GENERAL

The various research works on “Design and Fabrication of Autonomous Delivery Drones System” have been reviewed and summarized in this chapter.

#### 2.2 REVIEW OF THE LITERATURE

**De Fazio et al (2022)** Advancements in electronics, AI, navigation, and materials bolster unmanned platforms, expanding their military roles. They are poised to surpass manned platforms in future military operations. Fixed-wing drones, a type of UAV, draw design inspiration from traditional airplanes, featuring a similar aerodynamic structure and flight principles.

**Dr. Balaji B et al., 2022** Fixed-wing drones differ from quadcopters with wings and propellers for stability and longer flight. Ideal for tasks requiring extended coverage, like aerial surveys, mapping, and surveillance. Adaptable fixed-wing drones play crucial roles in agriculture, forestry, and environmental monitoring due to their efficient wide-area coverage capabilities, serving as invaluable tools in these sectors. Multi-rotor UAVs use three or more rotors for vertical take-off, hovering, and payload transport, making them versatile for various applications.

**Dhulashia et al (2021)** Single-rotor drones excel in professional fields like aerial photography, agriculture, search and rescue, and surveillance. They offer extended flight times, increased range, and greater payload capacity. A hybrid platform combines VTOL fixed-wing aircraft and UAVs with thrust vectoring. It leverages rotorcraft and fixed-wing strengths for versatile missions, combining speed, endurance, agility, and VTOL capabilities.

**Farras et al., 2015 [11]** A flight controller, essentially a circuit board, directs the flying of a drone. In layman's words, its major duty is to manage the power or RPM (rotations per minute) of each drone motor based on the input from the drone controller.

**Glassock, (2009)** Compared to traditional rotorcraft, hybrid platforms offer substantial improvements in range, endurance, altitude, payload capacity, and

maximum forward velocity, making them more versatile for various missions. Compared to traditional rotorcraft, hybrid platforms offer substantial improvements in range, endurance, altitude, payload capacity, and maximum forward velocity, making them more versatile for various missions. Sensors on drones cooperate, not duplicate, as weight and budget limitations apply.

**Gonçalves et al., 2023** sensors capture complementary data that enhances the information gathered by the primary sensor. Revised drone categories emphasize smaller sizes. "Large drones" cover fixed-wing (20-150 kg) and multirotor (25-100 kg). "Small drones" include fixed-wing (up to 20 kg) and multirotor (up to 25 kg), with a "mini-drones" subcategory (grams to kilograms) for indoor and recreational use.

**Lawrence et al (2023)** Delivery systems, like drones (UAVs), play a pivotal role in the modern economy, moving goods and services efficiently. Drones are unmanned aerial vehicles with key components for remote operation and GPS-guided precision. Last-mile delivery hurdles grow with new delivery methods. Businesses innovate in goods transport, aligning with just-in-time management, leading to a surge in delivery services.

**Lee et al., 2023** Safe landings require a suitable zone and skilled piloting to protect the drone and its sensors from harm, underscoring the importance of precision in this aspect of flight. Multi-rotor drones, while cost-effective, suffer from drawbacks like short flight time and limited speed, making them unsuitable for extensive tasks like aerial mapping or surveillance.

**Maher Aljehani et al., 2020** Delivery drones serve e-commerce, medical supply, food, disaster relief, agriculture, and more. Their speed and efficiency reduce reliance on human delivery, potentially easing traffic, and excel in transporting small, light items.

**Nugroho et al (2015)** Multirotor UAVs have gained popularity in civilian and military sectors for their cost-effectiveness and agility. They serve diverse purposes, including aerial photography, crop protection, and package delivery.

**R. R. Glasscock, 2009** Energy consumption in stability maintenance is a key challenge. Single-rotor drones utilize a single, large rotor for lift and control,

providing superior stability and payload capacity compared to multi-rotor counterparts in UAV design.

**Vergouw et al (2016)** Drones vary by size, weight, autonomy, range, motors, sensors, batteries, and speed, influencing their capabilities and usage, spanning small, indoor drones to large, multi-motor, long-range models. Flight has fascinated humanity, from Daedalus to the Montgolfier brothers. Santos Dumont and Zeppelin advanced dirigibles. Otto Lilienthal pioneered powered flight. Military conflicts showcased the efficiency of unmanned aircraft. The Wright brothers' 1903 flight marked aviation progress. The first UAV was Elmer Sperry's autopilot-controlled aircraft, the Curtiss N-9, with test flights in 1917, initially with a pilot onboard.

**Verstraete (2012)** Turbine engines offer high power-to-weight ratios but are inefficient for small UAVs. Hybrid power systems aim to combine thermal and electric engines but can be complex and less eco-friendly. Battery-powered UAVs, like quadrotors, rely on batteries for simplicity and versatility. Ideal for hobbyists, they provide cost-effective, extended flight times in various applications. Small battery-operated UAVs face endurance limits due to battery weight. With LiPo batteries, their maximum flight time is restricted to 90 minutes. Hot swapping involves rapidly exchanging depleted UAV batteries with fully charged ones, ensuring continuous operation and task completion. Multiple UAVs can coordinate hotspot handovers, ensuring uninterrupted service coverage.

## 2.3 NETWORK CONNECTIVITY

Network connectivity in drones pertains to their capacity to establish wireless links, often through channels like Wi-Fi, cellular networks, or satellite connections. This connectivity serves as the foundation for real-time data exchange, remote control, and communication between drones and their operators [12]. It significantly amplifies their capabilities in various applications, including aerial photography, surveillance, and autonomous operations. Establishing a proficient communication network for Unmanned Aerial Vehicles (UAVs) heavily depends on communication elements and protocols [13]. Numerous strategies have been put forth by researchers, encompassing vital aspects such as antenna setups, communication modules for networks, various networking tactics, and the utilization of algorithms and methodologies within drone networks [5].

The transceiver, functioning as both the sender and receiver, holds a central position within the system. The remote transceiver is responsible for transmitting data that originates from image processing. It encodes the grayscale image using a serial communication protocol before dispatching it to the ground segment. Serial communication offers several advantages when compared to parallel communication. It is a commonly used method in embedded systems, simplifying the process of interfacing. Serial communication cables can transmit data over longer distances with fewer individual cable fibers [6].

The Inertial Measurement Unit (IMU) is a component within an electronic module responsible for collecting data concerning angular acceleration and linear acceleration. This data is later transmitted to the primary processing unit [6]. The IMU consists of both an accelerometer, which is utilized to measure object acceleration, and a gyroscope, which is employed to track object rotation, enabling the detection and monitoring of an object's presence and movement. An analysis of the IMU instrumentation system can be carried out by employing an accelerometer sensor and a gyroscope sensor [15]. Serial communication offers several advantages when compared to parallel communication. It is a widely utilized method in embedded systems, simplifying the process of interfacing. Additionally, serial communication cables can transmit data over longer distances while requiring fewer individual cable fibers [14].

Within electronic modules, the Inertial Measurement Unit (IMU) serves as a critical component responsible for collecting data related to angular acceleration and linear acceleration. This data is subsequently relayed to the primary processing unit. The IMU is composed of two key elements: an accelerometer, which is employed for measuring object acceleration, and a gyroscope, used for tracking object rotation. Together, these sensors allow for the detection and monitoring of an object's presence and motion. A thorough examination of the IMU instrumentation system can be carried out through the utilization of both an accelerometer sensor and a gyroscope sensor [4].

The accelerometer sensor serves as a tool for monitoring an object's motion over a period. Through the application of a dedicated algorithm, this data on movement can be transformed into information regarding the object's position and distance traveled [33]. The Inertial Measurement Unit (IMU) is a device that integrates the accelerometer sensor with other sensors, including gyroscopes and magnetometers, to offer a more comprehensive overview of the object's motion. Within the Guidance, Navigation, and Control (GNC) system, the IMU plays a critical role in assisting the object in its navigation towards its intended destination.

For the ground station, this entails configuring the computer with a serial USB port, specifically the 433 MHz 3DR radio module. It also includes configuring the test settings and the software for analysis [36]. The Global Positioning System (GPS) is a satellite-based navigation system that provides vital location and time information. It serves as a fundamental component in many drone systems, enabling them to navigate autonomously. GPS functions by receiving signals transmitted by a network of satellites orbiting the Earth. The drone's GPS receiver utilizes these signals to calculate its precise position and time, which, in turn, guides and controls the drone's flight path. GPS offers several advantages for drone navigation, notably its accuracy, reliability, and global availability. An additional benefit is that it doesn't rely on ground-based infrastructure, making it particularly suitable for deployment in remote areas. However, GPS does have certain limitations, as it can be susceptible to interference from obstacles like buildings and trees. In urban settings, the presence of tall structures can also affect its accuracy.

## 2.4 POWER SOURCE OF AN UAV

Turbine engines are utilized in aircraft propulsion systems due to their favorable power-to-weight ratio and extended operational endurance. Nevertheless, they are less appropriate for small-scale Unmanned Aerial Vehicle (UAV) applications due to their suboptimal fuel efficiency, limited overall effectiveness, and the high levels of noise they generate.

**2.4.1 Hybrid Propulsion:** A hybrid power propulsion prototype was developed for Unmanned Aerial Systems (UASs) to combine the benefits of both thermal and electric engines in a hybrid configuration. Although simulation results indicated a 13% improvement in endurance, it's important to highlight that the system is complex and does not exhibit eco-friendliness [11].



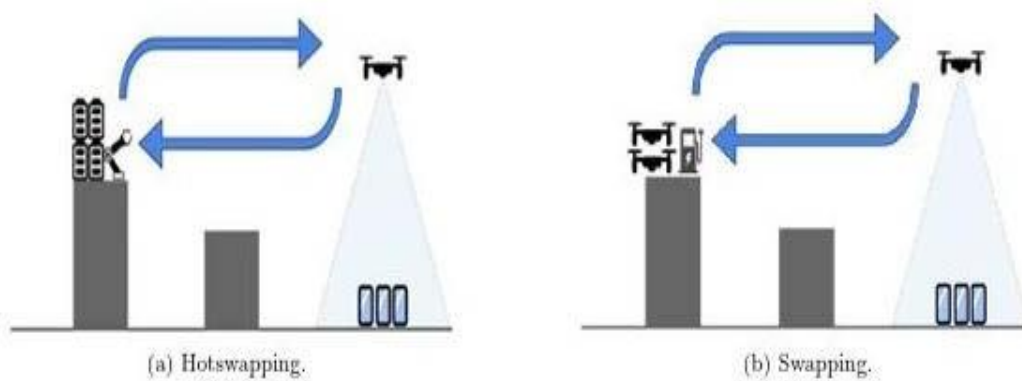
**Fig:2.1 : Hybrid Propulsion Based UAV**

**2.4.2 Battery Powered:** A battery-powered UAV, most small UAVs, particularly quadrotors, rely on battery power as their primary energy source. Batteries are regarded as the central element in these battery-powered UAVs, enhancing the simplicity and adaptability of the propulsion system [3]. Additionally, platforms based on batteries can effectively fulfill a range of hobbyist applications in terms of flight duration and cost-efficiency [6]. However, standard small battery-operated UAVs experience limited endurance because of restrictions imposed by the weight of the battery pack. With the use of LiPo batteries, their maximum flight time is capped at 90 minutes [3].

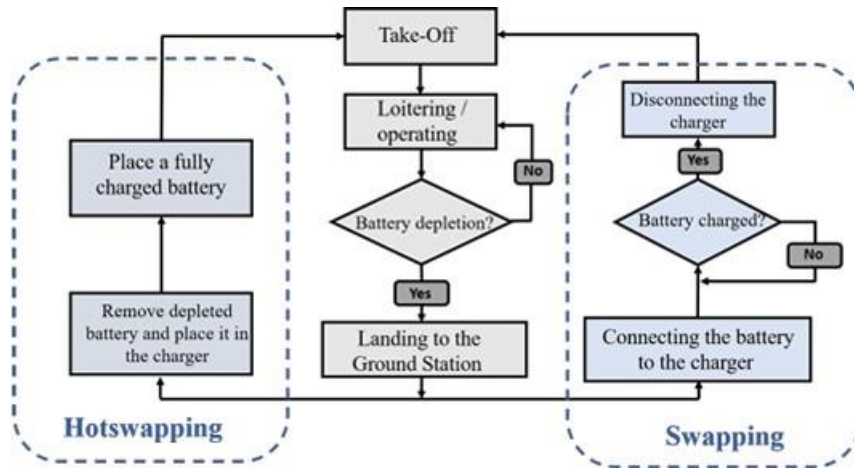


**Fig:2.2 : Battery-Powered UAV**

**2.4.3 Swapping:** Swapping is a method employed to recharge the drained batteries of UAVs while they are still on their mission. This process can be carried out autonomously or under human control [36]. Hot swapping is a specialized technique where a depleted battery is swiftly substituted with a fully charged one, ensuring that the UAV remains powered and operational. It can then rejoin its designated hotspot and continue its tasks. Utilizing multiple UAVs and effectively coordinating their handovers of hotspots to one another enables the multi-agent system to ensure uninterrupted service coverage in each area.



**Fig:2.3 : Swapping Technique**



**Fig:2.4 : Working Principle of Swapping Technique**

**2.4.4 Fuel Cell Power:** Fuel cell power sources offer a promising alternative to batteries for UAVs [14]. These fuel cells work by combining hydrogen and oxygen to generate electricity and water, presenting several advantages over batteries. Firstly, they boast a higher energy density, meaning that UAVs equipped with fuel cells can either achieve longer flight durations or carry heavier payloads [37]. Secondly, fuel cells have a longer lifespan since they can be refueled in flight, ensuring extended operational periods without the need for replacements. Lastly, their emissions are minimal, as they only produce water vapor and heat as waste products, making them a more environmentally friendly choice when compared to batteries. Challenges associated with using fuel cells in UAVs include weight and complexity, high cost, and hydrogen availability [18].

In the case of our drone, it operates using a 3-cell lithium-polymer (Li-Po) battery, featuring a combined voltage of 11.1 volts and a steady discharge rate of 40C. During flight, the drone's motors demand substantial power. To optimize battery longevity, you have the option to deactivate the entire system when the drone is not in flight, which will power down all components, including the motors [7].

Once the drone has landed and is prepared for parcel delivery, you can re-activate the system and provide power exclusively to the components essential for the delivery process, such as sensors and other electronic devices. This strategy minimizes power usage and extends the drone's battery life [27].

A microcontroller takes charge of managing the power supply to the various drone components. You can program the microcontroller to deactivate power to non-essential components during periods of inactivity. Furthermore, employing separate batteries for the motors, the sensors, and other electronic devices allows you to power down the motor battery when it's not in use.

Implementing a power management system can further optimize the drone's power consumption. This system can monitor the power consumption of different components and adjust the power supply as needed. By disarming the entire system and supplying power selectively, you can significantly prolong your drone's battery life. This practice holds particular significance for applications demanding long flight durations or the transportation of heavy payloads.



**Fig:2.5 : Fuel Cell Powered UAV**

## 2.5 DESIGN CUSTOMIZATION IN DELIVERY DRONE

The design of a drone can be customized to align with its intended purpose and use. While a quadcopter drone, equipped with four rotors, represents the most straightforward and prevalent drone type, it might not be the ideal choice for applications requiring substantial payload capacity or extended flight durations. In

such scenarios, opting for a hex-copter or octa-copter drone, featuring six or eight rotors, respectively, could be a more appropriate selection [19].

The number of propulsion units on a drone also varies based on its specific type. For instance, a fixed-wing drone, which relies on wings for lift, commonly incorporates one or two propellers. In contrast, a quadcopter drone boasts four propellers, while a hex copter employs six propellers, and an octa copter employs eight propellers [21]. The choice of drone type is influenced by factors like the drone's size and the distance it needs to cover. For instance, a compact drone with limited flight duration might be well-suited for indoor applications like inspecting warehouses or making doorstep deliveries. Conversely, outdoor tasks such as land surveying or delivering packages to remote regions may necessitate a larger drone with an extended flight time [34].

The drone's payload capacity is another vital aspect to consider. Some drones are designed for carrying light payloads, like cameras or sensors, while others are tailored for heavier loads, such as packages or even passengers [30]. Additionally, the design of the cargo attachment mechanism, like the trunk, can be adjusted to accommodate various payload types. For instance, a basic basket may suffice +for transporting a small package, while a more robust and aerodynamic trunk may be required for heavier or fragile payloads [23].

## CHAPTER 3

### AIM AND SCOPE OF THE PRESENT INVESTIGATION

#### 3.1 AIM

The primary objective of this project is to design and create an autonomous delivery drone system. This innovative system aims to revolutionize the delivery of goods and services, offering an efficient, cost-effective, and convenient solution for the transportation of various items. By developing a reliable and autonomous drone platform, we aim to address last-mile delivery challenges, enhance the accessibility and affordability of delivery services, and contribute to the advancement of the drone technology landscape. The project seeks to demonstrate the potential of autonomous delivery drones in various sectors, from e-commerce and medical supply delivery to disaster relief and agriculture. The goal is to develop a practical and scalable solution that can positively impact industries and society while mitigating some of the challenges associated with traditional delivery methods.

#### 3.2 SCOPE OF THE PRESENT INVESTIGATION

The scope of autonomous delivery drone systems is broad, encompassing a wide range of applications and industries. These systems are becoming increasingly integral to modern logistics and transportation, offering innovative solutions to address various challenges. Here is a detailed note on the scope of autonomous delivery drone systems. Autonomous delivery drone systems are designed to revolutionize the way goods and services are transported and delivered. Their scope includes:

**3.2.1 Last-Mile Delivery Revolution:** Autonomous delivery drones primarily focus on addressing the "last-mile" delivery challenge. They offer a solution to efficiently transport packages from distribution centers or hubs to the destination, often a customer's doorstep. This has the potential to reduce delivery times and costs significantly.

**3.2.2 E-commerce and Retail:** E-commerce platforms and retailers are the primary beneficiaries of autonomous delivery drones. These systems promise to enhance

the online shopping experience by providing rapid and convenient delivery options, improving customer satisfaction.

**3.2.3 Medical Supply Transport:** Autonomous drones have a crucial role in transporting medical supplies, including vaccines, medicines, and diagnostic equipment, especially in remote or inaccessible areas. They can help save lives during emergencies.

**3.2.4 Food and Grocery Delivery:** The food industry can benefit from autonomous delivery drone systems. They enable the efficient and contactless delivery of food, groceries, and other perishable items in urban and suburban settings.

**3.2.5 Disaster Relief and Humanitarian Aid:** Autonomous delivery drones can be deployed in disaster-stricken regions to deliver essential supplies, conduct search and rescue missions, and facilitate communication, providing critical support in times of crisis.

**3.2.6 Agriculture and Precision Farming:** Drones play a vital role in precision agriculture by delivering fertilizers, pesticides, and other agricultural inputs precisely to crops. This promotes sustainable and efficient farming practices.

**3.2.7 Environmental Monitoring:** Autonomous drones equipped with various sensors and cameras contribute to environmental monitoring, wildlife conservation, and research, providing valuable data for environmental protection efforts.

**3.2.8 Infrastructure Inspection:** These systems can inspect and monitor critical infrastructure, such as bridges, power lines, and pipelines, contributing to infrastructure maintenance and safety.

**3.2.9 Security and Surveillance:** Autonomous drones are employed for surveillance and security purposes, enhancing real-time monitoring, and safeguarding sensitive areas and borders.

**3.2.10 Customized Design:** The adaptability and flexibility of autonomous delivery drone systems allow for customization based on specific requirements. They can be designed in various sizes, payload capacities, and ranges to cater to the diverse needs of industries.

**3.2.11 Environmental Impact:** These systems have the potential to reduce the environmental impact of traditional delivery methods, such as reducing road traffic, lowering emissions, and conserving fuel.

**3.2.12 Regulatory and Safety Compliance:** The scope of autonomous delivery drone systems also encompasses addressing regulatory and safety concerns. Ensuring safe and compliant operations with aviation authorities is vital for the successful integration of drones into our daily lives.

Overall, the scope of autonomous delivery drone systems is vast and dynamic, promising to transform the delivery landscape across multiple sectors. As technology continues to advance and regulations evolve, the potential for these systems to improve efficiency, reduce costs, and enhance convenience for individuals and businesses is limitless. However, it is crucial to navigate these advancements responsibly and ensure safety and adherence to regulatory standards as these systems become an integral part of our daily lives.

## CHAPTER 4

### MATERIAL AND METHODOLOGY

#### 4.1 MATERIAL

A drone consists of several parts. The purpose of each part is to make a drone airborne, control, and stable flight, generate power, and quickly maneuver. The mentioned is the list of parts used in our application.

##### ***4.1.1 Frame of UAV***

It is a structure or a frame into which all the other body parts can be fit, it can hold 4 motors in it. A drone's frame is its structural core, typically made of lightweight materials like carbon fiber or aluminum. It provides support for all components, offers design flexibility, and ensures durability during flight and potential accidents.



***Fig:4.1 : Frame of a Quadcopter***

##### ***4.1.2 Motor of UAV***

Drones employ brushless electric motors, lightweight and efficient, to power their propellers. These motors vary in speed to control movement, providing stability, lift, and precise maneuvering, essential for their flight and functionality.



***Fig:4.2 : Brushless motor used in our application.***

#### **4.1.3 Electronic Speed Controller (ESC)**

EAC, or Environmental Air Circulation, in a drone, refers to the airflow system that helps cool its components and maintain optimal operating temperatures. It's crucial for preventing overheating during flight, ensuring reliable performance.



***Fig:4.3 : Electronic Speed Controller (ESC)***

#### **4.1.4 Flight Control**

Flight control creates a log of the take-off location that guides the drone and thereby the need to return to the take-off location. This has become known as the 'back home' feature.



***Fig:4.4 : Flight Control***

#### ***4.1.5 Propeller***

propellers in drones are vital components that generate thrust for flight. Typically, drones have two or more propellers attached to motors. By spinning at varying speeds, propellers control the drone's movement, enabling stability, lift, and direction changes during flight.



***Fig:4.5 : Propeller***

#### ***4.1.6 Radio Transmitter***

A radio transmitter in a drone is a handheld device or onboard system used to communicate commands wirelessly. It sends signals to the drone's receiver, enabling remote control, navigation, and functions like flying, capturing photos, or recording video.



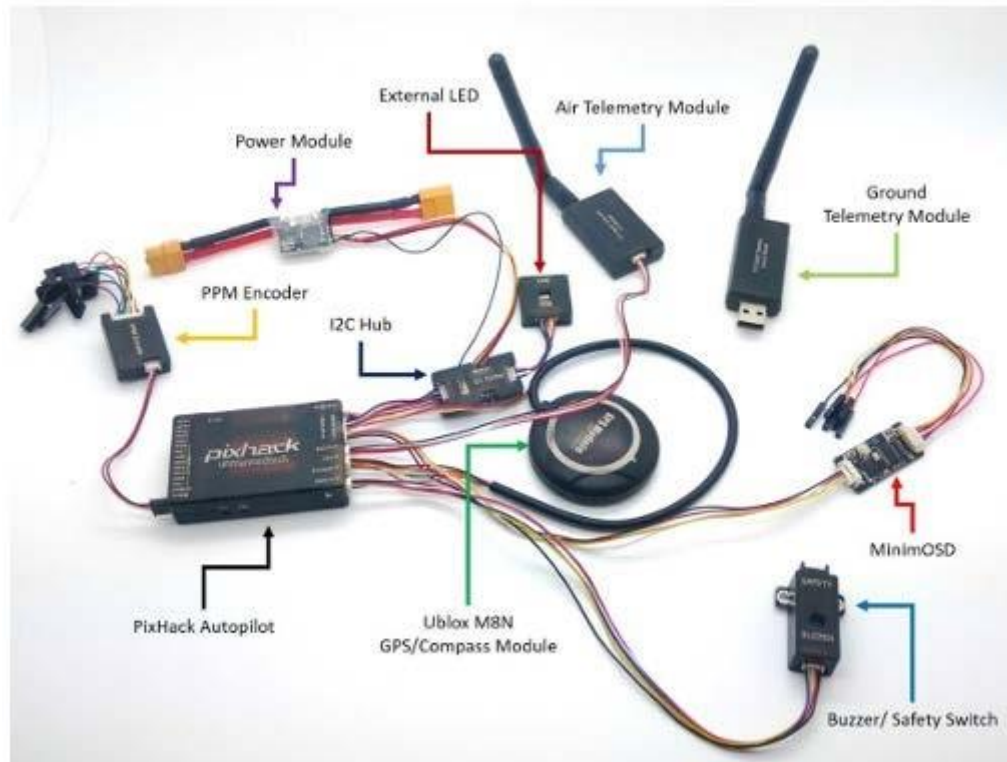
***Fig:4.6 : Radio Transmitter***

#### **4.1.7 Battery, Electronics, and Power Distribution Cables**

Batteries and electronics are integral to drones' functionality. Lithium-polymer or lithium-ion batteries provide power, determining flight time. Electronics, including flight controllers, GPS, gyroscopes, and cameras, enable stability, navigation, and various features like autonomous flight and aerial photography. The mentioned figures are the battery and other electronic components used in drones.



***Fig:4.7 : Battery used in UAVs***



***Fig:4.8 : Electronic components used in UAVs.***

#### **4.1.8 Camera**

Drones often feature cameras for capturing photos and videos from aerial perspectives. These cameras vary in quality and capabilities, providing essential tools for photography, videography, mapping, surveillance, and various professional and recreational applications.



***Fig:4.9 : Camera***

#### **4.1.9 Landing Gear**

The landing gear on drones provides support during take-off and landing. Typically retractable, they ensure stability during flight and protect the drone's payload and components. Different designs suit various drone types and applications.



***Fig:4.10 : Landing gear***

#### **4.2 METHODOLOGY**

We use a quadcopter in our project, A quadcopter drone has four motors, two of which rotate clockwise and the other two of which rotate counterclockwise. The electronic speed controller (ESC) controls the speed of the motors [6]. If the two motors on the back of the drone spin quickly, the drone moves forward. If the two motors on the front of the drone spin quickly, the drone moves backward. If the two motors on the left side of the drone spin quickly, the drone moves to the right [25]. If the two motors on the right side of the drone spin quickly, the drone moves to the left. The transmitter sends the desired signal to the receiver, and the flight controller sends the corresponding signals to the ESC [35]. Connecting a GPS to the flight controller allows for drone navigation. All the inputs for the delivery system can be uploaded into Mission Planner software, including the coordinates of the receiver's location, altitude, and speed. Mission Planner is a ground control station (GCS) software for drones [2]. It allows users to plan and execute complex flight missions. If the two motors on the back of the drone spin quickly, the drone moves forward. If the two motors on the front of the drone spin quickly, the drone moves backward. If the two motors on the left side of the drone spin quickly, the drone moves to the right. If the two motors on the right side of the drone spin quickly, the drone moves

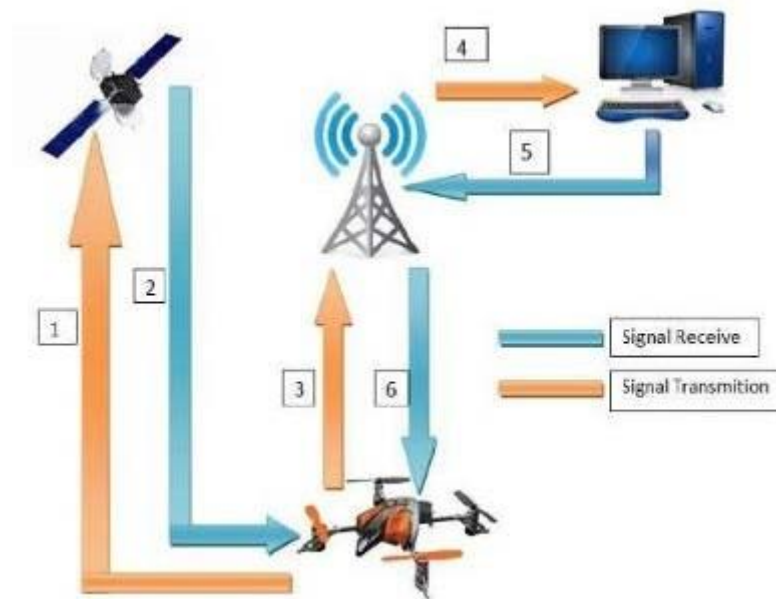
to the left. The transmitter sends the desired signal to the receiver, and the flight controller sends the corresponding signals to the ESC. Connecting a GPS to the flight controller allows for drone navigation. All the inputs for the delivery system can be uploaded into Mission Planner software, including the coordinates of the receiver's location, altitude, and speed. Mission Planner is a ground control station (GCS) software for drones. It allows users to plan and execute complex flight missions [25]. The drone can be designed and configured in such a way that not all electronic components and sensors are always activated. This can be done to save battery power, reduce weight, and improve performance.

#### 4.3 WORKING

Drones perform two core functions: flying and navigating. They use power sources, such as batteries or fuel, alongside rotors, propellers, and a lightweight composite frame to improve maneuverability. A remote controller is vital for takeoff, navigation, and landing, with communication achieved through radio waves like Wi-Fi. Drones can reach considerable altitudes, with key components comprising the operating system and the flight controller. Batteries supply power to the rotors, spinning the propellers for lift generation [1]. Flight controllers use various sensors like accelerometers, barometers, magnetometers, and gyroscopes to maintain flight. Additional features improving drone performance include GPS, obstacle detection, collision avoidance, cameras, and specialized software. Drones come with components such as an infrared camera, landing gear, rotor system, connectivity hardware, motors, and accelerometers. Drones can be thought of as flying robots, while traditional airplanes are typically controlled via a dedicated remote and rely on software programs and onboard sensors, like GPS, for flight management [15].

Drones function through a combination of remote control and autonomous systems. They usually feature four rotors for stability and maneuverability, which generate lift to keep the drone in the air and enable directional changes. Onboard sensors like GPS and accelerometers aid in navigation and stabilization [32]. Drone flight is controlled by a remote pilot or computer program, making it versatile for a wide range of applications, including aerial photography and surveillance. The flowchart below outlines the operational process of a drone system. The basic

quadcopter (QC) design consists of four rotor assemblies evenly spaced around a central hub. All rotors are positioned within a single plane, oriented to produce thrust perpendicular to the vehicle, as illustrated in the figure. If the rotor components have the same specifications and expected performance, each rotor will generate an equal amount of thrust for a given power input [17]. The angular momentum produced by any of the four rotors generates a torque around the drone's center of mass, which can be effectively countered by the opposing rotor's torque. In this configuration, opposing rotors must rotate in sync, while neighboring rotors rotate in opposite directions. One immediate advantage of the quad-rotor design is the elimination of the necessity for additional equipment, such as control moment gyroscopes. These are typically needed only to counteract external torques acting on the vehicle [10].



**Fig:4.11 : Working Principle of Drone**

## CHAPTER 5

### ALGORITHMS

ardupilotmega.xml

```
<?xml version="1.0"?>
```

```
<mavlink>
```

```
<include>common.xml</include>
```

```
<!-- Vendors -->
```

```
<include>uAvionix.xml</include>
```

```
<include>icarous.xml</include>
```

```
<include>cubepilot.xml</include>
```

```
<include>csAirLink.xml</include>
```

```
<dialect>2</dialect>
```

```
<!-- Note that ArduPilot-specific messages should use the command id range from  
150 to 250, to leave plenty of room for growth of
```

```
common.xml If you prototype a message here, then you should
```

```
consider if it is general enough to move into common.xml later -->
```

```
<enums>
```

```
<enum name="ACCELCAL_VEHICLE_POS">
```

```
<entry value="1" name="ACCELCAL_VEHICLE_POS_LEVEL"/>
```

```
<entry value="2" name="ACCELCAL_VEHICLE_POS_LEFT"/>
```

```
<entry value="3" name="ACCELCAL_VEHICLE_POS_RIGHT"/>
```

```
<entry value="4" name="ACCELCAL_VEHICLE_POS_NOSEDOWN"/>
```

```
<entry value="5" name="ACCELCAL_VEHICLE_POS_NOSEUP"/>
```

```
<entry value="6" name="ACCELCAL_VEHICLE_POS_BACK"/>
```

```

<entry value="16777215"
name="ACCELCAL_VEHICLE_POS_SUCCESS"/>
<entry value="16777216"
name="ACCELCAL_VEHICLE_POS_FAILED"/>
</enum>

```

```

<enum name="HEADING_TYPE">
<entry value="0"
name="HEADING_TYPE_COURSE_OVER_GROUND"/>
<entry value="1" name="HEADING_TYPE_HEADING"/>
</enum>

```

```

<!-- ardupilot specific MAV_CMD_* commands -->
<enum name="MAV_CMD">
<!-- 200 to 214 used by common.xml -->
<entry value="215"
name="MAV_CMD_DO_SET_RESUME_REPEAT_DIST"
hasLocation="false" isDestination="false">
<description>Set the distance to be repeated on mission resume</description>
<param index="1" label="Distance" units="m">Distance.</param>
<param index="2">Empty.</param>
<param index="3">Empty.</param>
<param index="4">Empty.</param>
<param index="5">Empty.</param>
<param index="6">Empty.</param>

```

<param index="7">Empty.</param>

</entry>

<entry value="216" name="MAV\_CMD\_DO\_SPRAYER" hasLocation="false" isDestination="false">

<description>Control attached liquid sprayer</description>

<param index="1" label="Sprayer Enable" minValue="0" maxValue="1" increment="1">0: disable sprayer. 1: enable sprayer.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="217"

name="MAV\_CMD\_DO\_SEND\_SCRIPT\_MESSAGE" hasLocation="false" isDestination="false">

<description>Pass instructions onto scripting, a script should be checking for a new command</description>

<param index="1" label="ID" minValue="0" maxValue="65535" increment="1">uint16 ID value to be passed to scripting</param>

<param index="2" label="param 1">float value to be passed to scripting</param>

<param index="3" label="param 2">float value to be passed to scripting</param>

<param index="4" label="param 3">float value to be passed to scripting</param>

<param index="5">Empty.</param>

```

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="218" name="MAV_CMD_DO_AUX_FUNCTION">

<description>Execute auxiliary function</description>

<param index="1" label="AuxiliaryFunction">Auxiliary Function.</param>


<param index="2" label="SwitchPosition"
enum="MAV_CMD_DO_AUX_FUNCTION_SWITCH_LEVEL">Switch
Level.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry          value="83"          name="MAV_CMD_NAV_ALTITUDE_WAIT"
hasLocation="false" isDestination="false">

<description>Mission command to wait for an altitude or downwards vertical speed.
This is meant for high altitude balloon launches, allowing the aircraft to be idle until
either an altitude is reached or a negative vertical speed is reached (indicating early
balloon burst). The wiggle time is how often to wiggle the control surfaces to prevent
them seizing up.</description>

<param index="1" label="Altitude" units="m">Altitude.</param>

<param index="2" label="Descent Speed" units="m/s">Descent speed.</param>

```

<param index="3" label="Wiggle Time" units="s">How long to wiggle the control surfaces to prevent them seizing up.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42000" name="MAV\_CMD\_POWER\_OFF\_INITIATED" hasLocation="false" isDestination="false">

<description>A system wide power-off event has been initiated.</description>

<param index="1">Empty.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<!-- MAV\_CMD\_SOLO\_BTN\_\* are here to provide vendor-specific

support for 3DR Solo until a better solution is found to atomically make multiple commands with control flow -->

<entry value="42001" name="MAV\_CMD\_SOLO\_BTN\_FLY\_CLICK" hasLocation="false" isDestination="false">

<description>FLY button has been clicked.</description>

<param index="1">Empty.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42002" name="MAV\_CMD\_SOLO\_BTN\_FLY\_HOLD"  
hasLocation="false" isDestination="false">

<description>FLY button has been held for 1.5 seconds.</description>

<param index="1" label="Takeoff Altitude" units="m">Takeoff altitude.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42003" name="MAV\_CMD\_SOLO\_BTN\_PAUSE\_CLICK"  
hasLocation="false" isDestination="false">

<description>PAUSE button has been clicked.</description>

<param index="1" label="Shot Mode" minValue="0" maxValue="1" increment="1">1  
if Solo is in a shot mode, 0 otherwise.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42004" name="MAV\_CMD\_FIXED\_MAG\_CAL" hasLocation="false" isDestination="false">

<description>Magnetometer calibration based on fixed position in earth field given by inclination, declination and intensity.</description>

<param index="1" label="Declination" units="deg">Magnetic declination.</param>

<param index="2" label="Inclination" units="deg">Magnetic inclination.</param>

<param index="3" label="Intensity" units="mgauss">Magnetic intensity.</param>

<param index="4" label="Yaw" units="deg">Yaw.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42005" name="MAV\_CMD\_FIXED\_MAG\_CAL\_FIELD" hasLocation="false" isDestination="false">

<description>Magnetometer calibration based on fixed expected field values.</description>

<param index="1" label="Field X" units="mgauss">Field strength X.</param>

<param index="2" label="Field Y" units="mgauss">Field strength Y.</param>

<param index="3" label="Field Z" units="mgauss">Field strength Z.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<!-- 42006 MAV\_CMD\_FIXED\_MAG\_CAL\_YAW moved to common.xml

-->

<entry value="42007" name="MAV\_CMD\_SET\_EKF\_SOURCE\_SET"

hasLocation="false" isDestination="false">

<description>Set EKF sensor source set.</description>

<param index="1" label="SourceSetId" minValue="1" maxValue="3"  
increment="1">Source Set Id.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42424" name="MAV\_CMD\_DO\_START\_MAG\_CAL"  
hasLocation="false" isDestination="false">

<description>Initiate a magnetometer calibration.</description>

<param index="1" label="Magnetometers Bitmask" minValue="0" maxValue="255" increment="1">Bitmask of magnetometers to calibrate. Use 0 to calibrate all sensors that can be started (sensors may not start if disabled, unhealthy, etc.). The command will NACK if calibration does not start for a sensor explicitly specified by the bitmask.</param>

<param index="2" label="Retry on Failure" minValue="0" maxValue="1" increment="1">Automatically retry on failure (0=no retry, 1=retry).</param>

<param index="3" label="Autosave" minValue="0" maxValue="1" increment="1">Save without user input (0=require input, 1=autosave).</param>

<param index="4" label="Delay" units="s">Delay.</param>

<param index="5" label="Autoreboot" minValue="0" maxValue="1" increment="1">Autoreboot (0=user reboot, 1=autoreboot).</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42425" name="MAV\_CMD\_DO\_ACCEPT\_MAG\_CAL" hasLocation="false" isDestination="false">

<description>Accept a magnetometer calibration.</description>

<param index="1" label="Magnetometers Bitmask" minValue="0" maxValue="255" increment="1">Bitmask of magnetometers that calibration is accepted (0 means all).</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

```

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry      value="42426"      name="MAV_CMD_DO_CANCEL_MAG_CAL"
hasLocation="false" isDestination="false">

<description>Cancel a running magnetometer calibration.</description>

<param index="1" label="Magnetometers Bitmask" minValue="0" maxValue="255"
increment="1">Bitmask of magnetometers to cancel a running calibration (0 means
all).</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry      value="42429"      name="MAV_CMD_ACCELCAL_VEHICLE_POS"
hasLocation="false" isDestination="false">

<description>Used when doing accelerometer calibration. When sent to the GCS
tells it what position to put the vehicle in. When sent to the vehicle says what position
the vehicle is in.</description>

<param      index="1"      label="Position"

enum="ACCELCAL_VEHICLE_POS">Position.</param>

<param index="2">Empty.</param>

```

```

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry      value="42428"      name="MAV_CMD_DO_SEND_BANNER"
hasLocation="false" isDestination="false">

<description>Reply with the version banner.</description>

<param index="1">Empty.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42427"

name="MAV_CMD_SET_FACTORY_TEST_MODE"      hasLocation="false"
isDestination="false">

<description>Command autopilot to get into factory test/diagnostic
mode.</description>

<param index="1" label="Test Mode" minValue="0" maxValue="1" increment="1">0:
activate test mode, 1: exit test mode.</param>

<param index="2">Empty.</param>

```

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42501" name="MAV\_CMD\_GIMBAL\_RESET" hasLocation="false" isDestination="false">

<description>Causes the gimbal to reset and boot as if it was just powered on.</description>

<param index="1">Empty.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42502"

name="MAV\_CMD\_GIMBAL\_AXIS\_CALIBRATION\_STATUS"

hasLocation="false" isDestination="false">

<description>Reports progress and success or failure of gimbal axis calibration procedure.</description>

<param index="1" label="Axis" enum="GIMBAL\_AXIS">Gimbal axis we're reporting calibration progress for.</param>

```

<param      index="2"      label="Progress"      units="%"      minValue="0"
maxValue="100">Current calibration progress for this axis.</param>

<param index="3" label="Status"
enum="GIMBAL_AXIS_CALIBRATION_STATUS">Status      of      the
calibration.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42503"
name="MAV_CMD_GIMBAL_REQUEST_AXIS_CALIBRATION"
hasLocation="false" isDestination="false">

<description>Starts commutation calibration on the gimbal.</description>

<param index="1">Empty.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry      value="42505"      name="MAV_CMD_GIMBAL_FULL_RESET"
hasLocation="false" isDestination="false">

<description>Erases gimbal application and parameters.</description>

```

```

<param index="1">Magic number.</param>

<param index="2">Magic number.</param>


<param index="3">Magic number.</param>
<param index="4">Magic number.</param>
<param index="5">Magic number.</param>
<param index="6">Magic number.</param>
<param index="7">Magic number.</param>

</entry>

<!-- 42600 used by common.xml -->

<entry      value="42650"      name="MAV_CMD_FLASH_BOOTLOADER"
hasLocation="false" isDestination="false">

<description>Update the bootloader</description>

<param index="1">Empty</param>
<param index="2">Empty</param>
<param index="3">Empty</param>
<param index="4">Empty</param>

<param index="5" label="Magic Number" increment="1">Magic number - set to
290876 to actually flash</param>
<param index="6">Empty</param>
<param index="7">Empty</param>

</entry>

<entry value="42651" name="MAV_CMD_BATTERY_RESET" hasLocation="false"
isDestination="false">

<description>Reset battery capacity for batteries that accumulate consumed battery
via integration.</description>

```

<param index="1" label="battery mask">Bitmask of batteries to reset. Least significant bit is for the first battery.</param>

<param index="2" label="percentage" minValue="0"

maxValue="100" increment="1">Battery percentage remaining to set.</param>

</entry>

<entry value="42700" name="MAV\_CMD\_DEBUG\_TRAP" hasLocation="false" isDestination="false">

<description>Issue a trap signal to the autopilot process, presumably to enter the debugger.</description>

<param index="1">Magic number - set to 32451 to actually trap.</param>

<param index="2">Empty.</param>

<param index="3">Empty.</param>

<param index="4">Empty.</param>

<param index="5">Empty.</param>

<param index="6">Empty.</param>

<param index="7">Empty.</param>

</entry>

<entry value="42701" name="MAV\_CMD\_SCRIPTING" hasLocation="false" isDestination="false">

<description>Control onboard scripting.</description>

<param index="1" enum="SCRIPTING\_CMD">Scripting command to execute</param>

</entry>

<entry value="42702" name="MAV\_CMD\_NAV\_SCRIPT\_TIME" hasLocation="false" isDestination="false">

<description>Scripting command as NAV command with wait for completion.</description>

<param index="1" label="command">integer command number (0 to 255)</param>

<param index="2" label="timeout" units="s">timeout for operation in seconds. Zero means no timeout (0 to 255)</param>

<param index="3" label="arg1">argument1.</param>

<param index="4" label="arg2">argument2.</param>

<param index="5">Empty</param>

<param index="6">Empty</param>

<param index="7">Empty</param>

</entry>

<entry value="42703" name="MAV\_CMD\_NAV\_ATTITUDE\_TIME" hasLocation="false" isDestination="false">

<description>Maintain an attitude for a specified time.</description>

<param index="1" label="time" units="s">Time to maintain specified attitude and climb rate</param>

<param index="2" label="roll" units="deg">Roll angle in degrees (positive is lean right, negative is lean left)</param>

<param index="3" label="pitch" units="deg">Pitch angle in degrees (positive is lean back, negative is lean forward)</param>

<param index="4" label="yaw" units="deg">Yaw angle</param>

<param index="5" label="climb\_rate" units="m/s">Climb rate</param>

<param index="6">Empty</param>

<param index="7">Empty</param>

</entry>

<entry value="43000" name="MAV\_CMD\_GUIDED\_CHANGE\_SPEED" hasLocation="false" isDestination="false">

<description>Change flight speed at a given rate. This slews the vehicle at a controllable rate between it's previous speed and the new one. (affects GUIDED only. Outside GUIDED, aircraft ignores these

commands. Designed for onboard companion-computer command- and-control, not normally operator/GCS control.)</description>

<param index="1" label="speed" type="enum="SPEED\_TYPE">Airspeed or groundspeed.</param>

<param index="2" label="speed target" units="m/s">Target Speed</param>

<param index="3" label="speed rate-of-change"

units="m/s/s">Acceleration rate, 0 to take effect instantly</param>

<param index="4">Empty</param>

<param index="5">Empty</param>

<param index="6">Empty</param>

<param index="7">Empty</param>

</entry>

<entry value="43001"

name="MAV\_CMD\_GUIDED\_CHANGE\_ALTITUDE" hasLocation="false" isDestination="false">

<description>Change target altitude at a given rate. This slews the vehicle at a controllable rate between it's previous altitude and the new one. (affects GUIDED only. Outside GUIDED, aircraft ignores these

commands. Designed for onboard companion-computer command- and-control, not normally operator/GCS control.)</description>

<param index="1">Empty</param>

<param index="2">Empty</param>

<param index="3" label="alt rate-of-change" units="m/s/s"

minValue="0">Rate of change, toward new altitude. 0 for maximum rate change. Positive numbers only, as negative numbers will not converge on the new target alt.</param>

<param index="4">Empty</param>

<param index="5">Empty</param>

<param index="6">Empty</param>

<param index="7" label="target alt" units="m">Target Altitude</param>

</entry>

<entry value="43002"

name="MAV\_CMD\_GUIDED\_CHANGE\_HEADING" hasLocation="false"  
isDestination="false">

<description>Change to target heading at a given rate, overriding previous heading/s. This slews the vehicle at a controllable rate between it's previous heading and the new one. (affects GUIDED only. Exiting GUIDED returns aircraft to normal behaviour defined elsewhere. Designed for onboard companion-computer command-and-control, not normally operator/GCS control.)</description>

<param index="1" label="heading type"

enum="HEADING\_TYPE">course-over-ground or raw vehicle heading.</param>

<param index="2" label="heading target" units="deg" minValue="0" maxValue="359.99">Target heading.</param>

<param index="3" label="heading rate-of-change"

units="m/s/s">Maximum centripetal acceleration, ie rate of change, toward new heading.</param>

<param index="4">Empty</param>

```

<param index="5">Empty</param>

<param index="6">Empty</param>

<param index="7">Empty</param>

</entry>

<!-- 43003 MAV_CMD_EXTERNAL_POSITION_ESTIMATE moved to
common.xml -->

</enum>

<enum name="SCRIPTING_CMD">

<entry value="0" name="SCRIPTING_CMD_REPL_START">

<description>Start a REPL session.</description>

</entry>

<entry value="1" name="SCRIPTING_CMD_REPL_STOP">

<description>End a REPL session.</description>

</entry>

<entry value="2" name="SCRIPTING_CMD_STOP">

<description>Stop execution of scripts.</description>

</entry>

<entry value="3" name="SCRIPTING_CMD_STOP_AND_RESTART">

<description>Stop execution of scripts and restart.</description>

</entry>

</enum>

<!-- AP_Limits Enums -->

<enum name="LIMITS_STATE">

```

```

<entry value="0" name="LIMITS_INIT">
<description>Pre-initialization.</description>
</entry>

<entry value="1" name="LIMITS_DISABLED">
<description>Disabled.</description>
</entry>

<entry value="2" name="LIMITS_ENABLED">
<description>Checking limits.</description>
</entry>

<entry value="3" name="LIMITS_TRIGGERED">
<description>A limit has been breached.</description>
</entry>

<entry value="4" name="LIMITS_RECOVERING">
<description>Taking action e.g. Return/RTL.</description>
</entry>

<entry value="5" name="LIMITS_RECOVERED">
<description>We're no longer in breach of a limit.</description>
</entry>
</enum>

<!-- AP_Limits Modules - power of 2 (1,2,4,8,16,32 etc) so we can send as bitfield -
->

<enum name="LIMIT_MODULE" bitmask="true">

<entry value="1" name="LIMIT_GPSLOCK">
<description>Pre-initialization.</description>

```

```

</entry>

<entry value="2" name="LIMIT_GEOFENCE">

<description>Disabled.</description>

</entry>

<entry value="4" name="LIMIT_ALTITUDE">

<description>Checking limits.</description>

</entry>

</enum>

<!-- RALLY flags - power of 2 (1,2,4,8,16,32,64,128) so we can send as a bitfield --
>

<enum name="RALLY_FLAGS" bitmask="true">

<description>Flags in RALLY_POINT message.</description>

<entry value="1" name="FAVORABLE_WIND">

<description>Flag set when requiring favorable winds for landing.</description>

</entry>

<entry value="2" name="LAND_IMMEDIATELY">

<description>Flag set when plane is to immediately descend to break altitude and
land without GCS intervention. Flag not set when plane is to loiter at Rally point until
commanded to land.</description>

</entry>

</enum>

<!-- Camera event types -->

<enum name="CAMERA_STATUS_TYPES">

<entry value="0" name="CAMERA_STATUS_TYPE_HEARTBEAT">

```

<description>Camera heartbeat, announce camera component ID at 1Hz.</description>

</entry>

<entry value="1" name="CAMERA\_STATUS\_TYPE\_TRIGGER">

<description>Camera image triggered.</description>

</entry>

<entry value="2" name="CAMERA\_STATUS\_TYPE\_DISCONNECT">

<description>Camera connection lost.</description>

</entry>

<entry value="3" name="CAMERA\_STATUS\_TYPE\_ERROR">

<description>Camera unknown error.</description>

</entry>

<entry value="4" name="CAMERA\_STATUS\_TYPE\_LOWBATT">

<description>Camera battery low. Parameter p1 shows reported voltage.</description>

</entry>

<entry value="5" name="CAMERA\_STATUS\_TYPE\_LOWSTORE">

<description>Camera storage low. Parameter p1 shows reported shots remaining.</description>

</entry>

<entry value="6" name="CAMERA\_STATUS\_TYPE\_LOWSTOREV">

<description>Camera storage low. Parameter p1 shows reported video minutes remaining.</description>

</entry>

</enum>

```

<!-- camera feedback flags, a little bit of future-proofing -->

<enum name="CAMERA_FEEDBACK_FLAGS">

<entry value="0" name="CAMERA_FEEDBACK_PHOTO">

<description>Shooting photos, not video.</description>

</entry>

<entry value="1" name="CAMERA_FEEDBACK_VIDEO">

<description>Shooting video, not stills.</description>

</entry>

<entry value="2" name="CAMERA_FEEDBACK_BADEXPOSURE">

<description>Unable to achieve requested exposure (e.g. shutter speed too
low).</description>

</entry>

<entry value="3" name="CAMERA_FEEDBACK_CLOSEDLOOP">

<description>Closed loop feedback from camera, we know for sure it has
successfully taken a picture.</description>

</entry>

<entry value="4" name="CAMERA_FEEDBACK_OPENLOOP">

<description>Open loop camera, an image trigger has been requested but we can't
know for sure it has successfully taken a picture.</description>

</entry>

</enum>

<!-- Gimbal payload enumerations -->

<enum name="MAV_MODE_GIMBAL">

<entry value="0" name="MAV_MODE_GIMBAL_UNINITIALIZED">

```

<description>Gimbal is powered on but has not started initializing yet.</description>

</entry>

<entry value="1"

name="MAV\_MODE\_GIMBAL\_CALIBRATING\_PITCH">

<description>Gimbal is currently running calibration on the pitch axis.</description>

</entry>

<entry value="2" name="MAV\_MODE\_GIMBAL\_CALIBRATING\_ROLL">

<description>Gimbal is currently running calibration on the roll axis.</description>

</entry>

<entry value="3" name="MAV\_MODE\_GIMBAL\_CALIBRATING\_YAW">

<description>Gimbal is currently running calibration on the yaw axis.</description>

</entry>

<entry value="4" name="MAV\_MODE\_GIMBAL\_INITIALIZED">

<description>Gimbal has finished calibrating and initializing, but is relaxed pending reception of first rate command from

copter.</description>

</entry>

<entry value="5" name="MAV\_MODE\_GIMBAL\_ACTIVE">

<description>Gimbal is actively stabilizing.</description>

</entry>

<entry value="6"

name="MAV\_MODE\_GIMBAL\_RATE\_CMD\_TIMEOUT">

<description>Gimbal is relaxed because it missed more than 10 expected rate command messages in a row. Gimbal will move back to active mode when it receives a new rate command.</description>

</entry>

</enum>

<enum name="GIMBAL\_AXIS">

<entry value="0" name="GIMBAL\_AXIS\_YAW">

<description>Gimbal yaw axis.</description>

</entry>

<entry value="1" name="GIMBAL\_AXIS\_PITCH">

<description>Gimbal pitch axis.</description>

</entry>

<entry value="2" name="GIMBAL\_AXIS\_ROLL">

<description>Gimbal roll axis.</description>

</entry>

</enum>

<enum name="GIMBAL\_AXIS\_CALIBRATION\_STATUS">

<entry value="0"

name="GIMBAL\_AXIS\_CALIBRATION\_STATUS\_IN\_PROGRESS">

<description>Axis calibration is in progress.</description>

</entry>

<entry value="1"

name="GIMBAL\_AXIS\_CALIBRATION\_STATUS\_SUCCEEDED">

<description>Axis calibration succeeded.</description>

</entry>

```

<entry value="2"
name="GIMBAL_AXIS_CALIBRATION_STATUS_FAILED">
<description>Axis calibration failed.</description>

</entry>

</enum>

<enum name="GIMBAL_AXIS_CALIBRATION_REQUIRED">
<entry value="0"
name="GIMBAL_AXIS_CALIBRATION_REQUIRED_UNKNOWN">
<description>Whether or not this axis requires calibration is unknown at this
time.</description>

</entry>

<entry value="1"
name="GIMBAL_AXIS_CALIBRATION_REQUIRED_TRUE">
<description>This axis requires calibration.</description>

</entry>

<entry value="2"
name="GIMBAL_AXIS_CALIBRATION_REQUIRED_FALSE">
<description>This axis does not require calibration.</description>

</entry>

</enum>

<!-- GoPro System Enumerations -->

<enum name="GOPRO_HEARTBEAT_STATUS">
<entry value="0"
name="GOPRO_HEARTBEAT_STATUS_DISCONNECTED">

```

```

<description>No GoPro connected.</description>

</entry>

<entry value="1"
name="GOPRO_HEARTBEAT_STATUS_INCOMPATIBLE">
<description>The detected GoPro is not HeroBus compatible.</description>

</entry>

<entry value="2"
name="GOPRO_HEARTBEAT_STATUS_CONNECTED">
<description>A HeroBus compatible GoPro is connected.</description>

</entry>

<entry value="3" name="GOPRO_HEARTBEAT_STATUS_ERROR">
<description>An unrecoverable error was encountered with the connected GoPro,
it may require a power cycle.</description>

</entry>

</enum>

<enum name="GOPRO_HEARTBEAT_FLAGS" bitmask="true">

<!-- each entry is a mask to test a bit in GOPRO_HEARTBEAT_STATUS.flags -->

<entry value="1" name="GOPRO_FLAG_RECORDING">
<description>GoPro is currently recording.</description>

</entry>

</enum>

<enum name="GOPRO_REQUEST_STATUS">

<entry value="0" name="GOPRO_REQUEST_SUCCESS">
<description>The write message with ID indicated succeeded.</description>

```

```

</entry>

<entry value="1" name="GOPRO_REQUEST_FAILED">

<description>The write message with ID indicated failed.</description>

</entry>

</enum>

<enum name="GOPRO_COMMAND">

<entry value="0" name="GOPRO_COMMAND_POWER">

<description>(Get/Set).</description>

</entry>

<entry value="1" name="GOPRO_COMMAND_CAPTURE_MODE">

<description>(Get/Set).</description>

</entry>

<entry value="2" name="GOPRO_COMMAND_SHUTTER">

<description>(      /Set).</description>

</entry>

<entry value="3" name="GOPRO_COMMAND_BATTERY">

<description>(Get/ ).</description>

</entry>

<entry value="4" name="GOPRO_COMMAND_MODEL">

<description>(Get/ ).</description>

</entry>

<entry value="5" name="GOPRO_COMMAND_VIDEO_SETTINGS">

<description>(Get/Set).</description>

```

<!-- Packet structure for the four values is as follows byte 0: GOPRO\_RESOLUTION  
byte 1: GOPRO\_FRAME\_RATE byte 2:

GOPRO\_FIELD\_OF\_VIEW byte 3: GOPRO\_VIDEO\_SETTINGS\_FLAGS -->

</entry>

<entry value="6" name="GOPRO\_COMMAND\_LOW\_LIGHT">

<description>(Get/Set).</description>

</entry>

<entry value="7"

name="GOPRO\_COMMAND\_PHOTO\_RESOLUTION">

<description>(Get/Set).</description>

</entry>

<entry value="8"

name="GOPRO\_COMMAND\_PHOTO\_BURST\_RATE">

<description>(Get/Set).</description>

</entry>

<entry value="9" name="GOPRO\_COMMAND\_PROTUNE">

<description>(Get/Set).</description>

</entry>

<entry value="10"

name="GOPRO\_COMMAND\_PROTUNE\_WHITE\_BALANCE">

<description>(Get/Set) Hero 3+ Only.</description>

</entry>

<entry value="11"

name="GOPRO\_COMMAND\_PROTUNE\_COLOUR">

```

<description>(Get/Set) Hero 3+ Only.</description>

</entry>

<entry value="12" name="GOPRO_COMMAND_PROTUNE_GAIN">

<description>(Get/Set) Hero 3+ Only.</description>

</entry>

<entry value="13"

name="GOPRO_COMMAND_PROTUNE_SHARPNESS">

<description>(Get/Set) Hero 3+ Only.</description>

</entry>

<entry value="14"

name="GOPRO_COMMAND_PROTUNE_EXPOSURE">

<description>(Get/Set) Hero 3+ Only.</description>

</entry>

<entry value="15" name="GOPRO_COMMAND_TIME">

<description>(Get/Set).</description>

<!-- Packet structure for the four values is as follows byte 0...3: uint32_t unix
timestamp (byte 0 is MSB) -->

</entry>

<entry value="16" name="GOPRO_COMMAND_CHARGING">

<description>(Get/Set).</description>

</entry>

</enum>

<enum name="GOPRO_CAPTURE_MODE">

<entry value="0" name="GOPRO_CAPTURE_MODE_VIDEO">

```

<description>Video mode.</description>

</entry>

<entry value="1" name="GOPRO\_CAPTURE\_MODE\_PHOTO">

<description>Photo mode.</description>

</entry>

<entry value="2" name="GOPRO\_CAPTURE\_MODE\_BURST">

<description>Burst mode, Hero 3+ only.</description>

</entry>

<entry value="3" name="GOPRO\_CAPTURE\_MODE\_TIME\_LAPSE">

<description>Time lapse mode, Hero 3+ only.</description>

</entry>

<entry value="4" name="GOPRO\_CAPTURE\_MODE\_MULTI\_SHOT">

<description>Multi shot mode, Hero 4 only.</description>

</entry>

<entry value="5" name="GOPRO\_CAPTURE\_MODE\_PLAYBACK">

<description>Playback mode, Hero 4 only, silver only except when LCD or HDMI is connected to black.</description>

</entry>

<entry value="6" name="GOPRO\_CAPTURE\_MODE\_SETUP">

<description>Playback mode, Hero 4 only.</description>

</entry>

<entry value="255" name="GOPRO\_CAPTURE\_MODE\_UNKNOWN">

<description>Mode not yet known.</description>

</entry>

</enum>

<enum name="GOPRO\_RESOLUTION">

<entry value="0" name="GOPRO\_RESOLUTION\_480p">

<description>848 x 480 (480p).</description>

</entry>

<entry value="1" name="GOPRO\_RESOLUTION\_720p">

<description>1280 x 720 (720p).</description>

</entry>

<entry value="2" name="GOPRO\_RESOLUTION\_960p">

<description>1280 x 960 (960p).</description>

</entry>

<entry value="3" name="GOPRO\_RESOLUTION\_1080p">

<description>1920 x 1080 (1080p).</description>

</entry>

<entry value="4" name="GOPRO\_RESOLUTION\_1440p">

<description>1920 x 1440 (1440p).</description>

</entry>

<entry value="5" name="GOPRO\_RESOLUTION\_2\_7k\_17\_9">

<description>2704 x 1440 (2.7k-17:9).</description>

</entry>

<entry value="6" name="GOPRO\_RESOLUTION\_2\_7k\_16\_9">

<description>2704 x 1524 (2.7k-16:9).</description>

</entry>

<entry value="7" name="GOPRO\_RESOLUTION\_2\_7k\_4\_3">

<description>2704 x 2028 (2.7k-4:3).</description>

</entry>

<entry value="8" name="GOPRO\_RESOLUTION\_4k\_16\_9">

<description>3840 x 2160 (4k-16:9).</description>

</entry>

<entry value="9" name="GOPRO\_RESOLUTION\_4k\_17\_9">

<description>4096 x 2160 (4k-17:9).</description>

</entry>

<entry value="10"

name="GOPRO\_RESOLUTION\_720p\_SUPERVIEW">

<description>1280 x 720 (720p-SuperView).</description>

</entry>

<entry value="11"

name="GOPRO\_RESOLUTION\_1080p\_SUPERVIEW">

<description>1920 x 1080 (1080p-SuperView).</description>

</entry>

<entry value="12"

name="GOPRO\_RESOLUTION\_2\_7k\_SUPERVIEW">

<description>2704 x 1520 (2.7k-SuperView).</description>

</entry>

<entry value="13" name="GOPRO\_RESOLUTION\_4k\_SUPERVIEW">

<description>3840 x 2160 (4k-SuperView).</description>

```

</entry>

</enum>

<enum name="GOPRO_FRAME_RATE">

<entry value="0" name="GOPRO_FRAME_RATE_12">

<description>12 FPS.</description>

</entry>

<entry value="1" name="GOPRO_FRAME_RATE_15">

<description>15 FPS.</description>

</entry>

<entry value="2" name="GOPRO_FRAME_RATE_24">

<description>24 FPS.</description>

</entry>

<entry value="3" name="GOPRO_FRAME_RATE_25">

<description>25 FPS.</description>

</entry>

<entry value="4" name="GOPRO_FRAME_RATE_30">

<description>30 FPS.</description>

</entry>

<entry value="5" name="GOPRO_FRAME_RATE_48">

<description>48 FPS.</description>

</entry>

<entry value="6" name="GOPRO_FRAME_RATE_50">

<description>50 FPS.</description>

</entry>

```

```

<entry value="7" name="GOPRO_FRAME_RATE_60">
<description>60 FPS.</description>

</entry>

<entry value="8" name="GOPRO_FRAME_RATE_80">
<description>80 FPS.</description>

</entry>

<entry value="9" name="GOPRO_FRAME_RATE_90">
<description>90 FPS.</description>

</entry>

<entry value="10" name="GOPRO_FRAME_RATE_100">
<description>100 FPS.</description>

</entry>

<entry value="11" name="GOPRO_FRAME_RATE_120">
<description>120 FPS.</description>

</entry>

<entry value="12" name="GOPRO_FRAME_RATE_240">
<description>240 FPS.</description>

</entry>

<entry value="13" name="GOPRO_FRAME_RATE_12_5">
<description>12.5 FPS.</description>

</entry>

</enum>

<enum name="GOPRO_FIELD_OF_VIEW">

```

```

<entry value="0" name="GOPRO_FIELD_OF_VIEW_WIDE">
<description>0x00: Wide.</description>
</entry>

<entry value="1" name="GOPRO_FIELD_OF_VIEW_MEDIUM">
<description>0x01: Medium.</description>

</entry>

<entry value="2" name="GOPRO_FIELD_OF_VIEW_NARROW">
<description>0x02: Narrow.</description>
</entry>
</enum>

<enum name="GOPRO_VIDEO_SETTINGS_FLAGS" bitmask="true">
<entry value="1" name="GOPRO_VIDEO_SETTINGS_TV_MODE">
<description>0=NTSC, 1=PAL.</description>
</entry>
</enum>

<enum name="GOPRO_PHOTO_RESOLUTION">
<entry value="0"
name="GOPRO_PHOTO_RESOLUTION_5MP_MEDIUM">
<description>5MP Medium.</description>
</entry>

<entry value="1"
name="GOPRO_PHOTO_RESOLUTION_7MP_MEDIUM">
<description>7MP Medium.</description>

```

</entry>

<entry value="6" name="DEEPSTALL\_STAGE\_LAND">

<description>Stalling and steering towards the land point.</description>

</entry>

</enum>

<enum name="PLANE\_MODE">

<description>A mapping of plane flight modes for custom\_mode field of heartbeat.</description>

<entry value="0" name="PLANE\_MODE\_MANUAL"/>

<entry value="1" name="PLANE\_MODE\_CIRCLE"/>

<entry value="2" name="PLANE\_MODE\_STABILIZE"/>

<entry value="3" name="PLANE\_MODE\_TRAINING"/>

<entry value="4" name="PLANE\_MODE\_ACRO"/>

<entry value="5" name="PLANE\_MODE\_FLY\_BY\_WIRE\_A"/>

<entry value="6" name="PLANE\_MODE\_FLY\_BY\_WIRE\_B"/>

<entry value="7" name="PLANE\_MODE\_CRUISE"/>

<entry value="8" name="PLANE\_MODE\_AUTOTUNE"/>

<entry value="10" name="PLANE\_MODE\_AUTO"/>

<entry value="11" name="PLANE\_MODE\_RTL"/>

<entry value="12" name="PLANE\_MODE\_LOITER"/>

<entry value="13" name="PLANE\_MODE\_TAKEOFF"/>

<entry value="14" name="PLANE\_MODE\_AVOID\_ADSB"/>

<entry value="15" name="PLANE\_MODE\_GUIDED"/>

<entry value="16" name="PLANE\_MODE\_INITIALIZING"/>

```

<entry value="17" name="PLANE_MODE_QSTABILIZE"/>
<entry value="18" name="PLANE_MODE_QHOVER"/>
<entry value="19" name="PLANE_MODE_QLOITER"/>
<entry value="20" name="PLANE_MODE_QLAND"/>
<entry value="21" name="PLANE_MODE_QRTL"/>
<entry value="22" name="PLANE_MODE_QAUTOTUNE"/>
<entry value="23" name="PLANE_MODE_QACRO"/>
<entry value="24" name="PLANE_MODE_THERMAL"/>
</enum>

<enum name="COPTER_MODE">

<description>A mapping of copter flight modes for custom_mode field of
heartbeat.</description>

<entry value="0" name="COPTER_MODE_STABILIZE"/>
<entry value="1" name="COPTER_MODE_ACRO"/>
<entry value="2" name="COPTER_MODE_ALT_HOLD"/>
<entry value="3" name="COPTER_MODE_AUTO"/>
<entry value="4" name="COPTER_MODE_GUIDED"/>
<entry value="5" name="COPTER_MODE_LOITER"/>
<entry value="6" name="COPTER_MODE_RTL"/>
<entry value="7" name="COPTER_MODE_CIRCLE"/>
<entry value="9" name="COPTER_MODE_LAND"/>
<entry value="11" name="COPTER_MODE_DRIFT"/>
<entry value="13" name="COPTER_MODE_SPORT"/>
<entry value="14" name="COPTER_MODE_FLIP"/>

```

```

<entry value="15" name="COPTER_MODE_AUTOTUNE"/>
<entry value="16" name="COPTER_MODE_POSHOLD"/>
<entry value="17" name="COPTER_MODE_BRAKE"/>
<entry value="18" name="COPTER_MODE_THROW"/>
<entry value="19" name="COPTER_MODE_AVOID_ADSB"/>
<entry value="20" name="COPTER_MODE_GUIDED_NOGPS"/>
<entry value="21" name="COPTER_MODE_SMART_RTL"/>
<entry value="22" name="COPTER_MODE_FLOWHOLD"/>
<entry value="23" name="COPTER_MODE_FOLLOW"/>
<entry value="24" name="COPTER_MODE_ZIGZAG"/>
<entry value="25" name="COPTER_MODE_SYSTEMID"/>
<entry value="26" name="COPTER_MODE_AUTOROTATE"/>
<entry value="27" name="COPTER_MODE_AUTO_RTL"/>
</enum>

```

```

<enum name="SUB_MODE">

```

```

<description>A mapping of sub flight modes for custom_mode field of
heartbeat.</description>

```

```

<entry value="0" name="SUB_MODE_STABILIZE"/>
<entry value="1" name="SUB_MODE_ACRO"/>
<entry value="2" name="SUB_MODE_ALT_HOLD"/>
<entry value="3" name="SUB_MODE_AUTO"/>
<entry value="4" name="SUB_MODE_GUIDED"/>
<entry value="7" name="SUB_MODE_CIRCLE"/>
<entry value="9" name="SUB_MODE_SURFACE"/>
<entry value="16" name="SUB_MODE_POSHOLD"/>

```

```
<entry value="19" name="SUB_MODE_MANUAL"/>
```

```
</enum>
```

```
<enum name="ROVER_MODE">
```

```
<description>A mapping of rover flight modes for custom_mode field of heartbeat.</description>
```

```
<entry value="0" name="ROVER_MODE_MANUAL"/>
```

```
<entry value="1" name="ROVER_MODE_ACRO"/>
```

```
<entry value="3" name="ROVER_MODE_STEERING"/>
```

```
<entry value="4" name="ROVER_MODE_HOLD"/>
```

```
<entry value="5" name="ROVER_MODE_LOITER"/>
```

```
<entry value="6" name="ROVER_MODE_FOLLOW"/>
```

```
<entry value="7" name="ROVER_MODE_SIMPLE"/>
```

```
<entry value="10" name="ROVER_MODE_AUTO"/>
```

```
<entry value="11" name="ROVER_MODE_RTL"/>
```

```
<entry value="12" name="ROVER_MODE_SMART_RTL"/>
```

```
<entry value="15" name="ROVER_MODE_GUIDED"/>
```

```
<entry value="16" name="ROVER_MODE_INITIALIZING"/>
```

```
</enum>
```

```
<enum name="TRACKER_MODE">
```

```
<description>A mapping of antenna tracker flight modes for custom_mode field of heartbeat.</description>
```

```
<entry value="0" name="TRACKER_MODE_MANUAL"/>
```

```
<entry value="1" name="TRACKER_MODE_STOP"/>
```

```
<entry value="2" name="TRACKER_MODE_SCAN"/>
```

```

<entry value="3" name="TRACKER_MODE_SERVO_TEST"/>
<entry value="10" name="TRACKER_MODE_AUTO"/>
<entry value="16" name="TRACKER_MODE_INITIALIZING"/>
</enum>

<enum name="OSD_PARAM_CONFIG_TYPE">

<description>The type of parameter for the OSD parameter editor.</description>
<entry value="0" name="OSD_PARAM_NONE"/>
<entry value="1" name="OSD_PARAM_SERIAL_PROTOCOL"/>
<entry value="2" name="OSD_PARAM_SERVO_FUNCTION"/>
<entry value="3" name="OSD_PARAM_AUX_FUNCTION"/>
<entry value="4" name="OSD_PARAM_FLIGHT_MODE"/>
<entry value="5" name="OSD_PARAM_FAILSAFE_ACTION"/>
<entry value="6" name="OSD_PARAM_FAILSAFE_ACTION_1"/>
<entry value="7" name="OSD_PARAM_FAILSAFE_ACTION_2"/>
<entry value="8" name="OSD_PARAM_NUM_TYPES"/>
</enum>

<enum name="OSD_PARAM_CONFIG_ERROR">

<description>The error type for the OSD parameter editor.</description>
<entry value="0" name="OSD_PARAM_SUCCESS"/>
<entry value="1" name="OSD_PARAM_INVALID_SCREEN"/>
<entry value="2"
name="OSD_PARAM_INVALID_PARAMETER_INDEX"/>
<entry value="3" name="OSD_PARAM_INVALID_PARAMETER"/>

```

```

</enum>

</enums>

<messages>

<message id="150" name="SENSOR_OFFSETS">

<description>Offsets and calibrations values for hardware sensors.

This makes it easier to debug the calibration process.</description>

<field type="int16_t" name="mag_ofs_x">Magnetometer X offset.</field>

<field type="int16_t" name="mag_ofs_y">Magnetometer Y offset.</field>

<field type="int16_t" name="mag_ofs_z">Magnetometer Z offset.</field>

<field      type="float"      name="mag_declination"      units="rad">Magnetic
declination.</field>

<field type="int32_t" name="raw_press">Raw pressure from barometer.</field>

<field type="int32_t" name="raw_temp">Raw temperature from barometer.</field>

<field type="float" name="gyro_cal_x">Gyro X calibration.</field>

<field type="float" name="gyro_cal_y">Gyro Y calibration.</field>

<field type="float" name="gyro_cal_z">Gyro Z calibration.</field>

<field type="float" name="accel_cal_x">Accel X calibration.</field>

<field type="float" name="accel_cal_y">Accel Y calibration.</field>

<field type="float" name="accel_cal_z">Accel Z calibration.</field>

</message>

<message id="151" name="SET_MAG_OFFSETS">

<deprecated since="2014-07"

replaced_by="MAV_CMD_PREFLIGHT_SET_SENSOR_OFFSETS"/>

<description>Set the magnetometer offsets</description>

```

```

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="int16_t" name="mag_ofs_x">Magnetometer X offset.</field>

<field type="int16_t" name="mag_ofs_y">Magnetometer Y offset.</field>


<field type="int16_t" name="mag_ofs_z">Magnetometer Z offset.</field>

</message>

<message id="152" name="MEMINFO">

<description>State of autopilot RAM.</description>

<field type="uint16_t" name="brkval">Heap top.</field>

<field type="uint16_t" name="freemem" units="bytes">Free memory.</field>

<extensions/>

<field type="uint32_t" name="freemem32" units="bytes">Free memory (32
bit).</field>

</message>

<message id="153" name="AP_ADC">

<description>Raw ADC output.</description>

<field type="uint16_t" name="adc1">ADC output 1.</field>

<field type="uint16_t" name="adc2">ADC output 2.</field>

<field type="uint16_t" name="adc3">ADC output 3.</field>

<field type="uint16_t" name="adc4">ADC output 4.</field>

<field type="uint16_t" name="adc5">ADC output 5.</field>

<field type="uint16_t" name="adc6">ADC output 6.</field>

</message>

<!-- Camera Controller Messages -->

```

```

<message id="154" name="DIGICAM_CONFIGURE">

<description>Configure on-board Camera Control System.</description>

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>


<field type="uint8_t" name="mode">Mode enumeration from 1 to N
//P, TV, AV, M, etc. (0 means ignore).</field>

<field type="uint16_t" name="shutter_speed">Divisor number //e.g.
1000 means 1/1000 (0 means ignore).</field>

<field type="uint8_t" name="aperture">F stop number x 10 //e.g. 28 means 2.8 (0
means ignore).</field>

<field type="uint8_t" name="iso">ISO enumeration from 1 to N //e.g.
80, 100, 200, Etc (0 means ignore).</field>

<field type="uint8_t" name="exposure_type">Exposure type enumeration from 1 to
N (0 means ignore).</field>

<field type="uint8_t" name="command_id">Command Identity
(incremental loop: 0 to 255). //A command sent multiple times will be executed or
pooled just once.</field>

<field type="uint8_t" name="engine_cut_off" units="ds">Main engine cut-off time
before camera trigger (0 means no cut-off).</field>

<field type="uint8_t" name="extra_param">Extra parameters enumeration (0
means ignore).</field>

<field type="float" name="extra_value">Correspondent value to given
extra_param.</field>

</message>

<message id="155" name="DIGICAM_CONTROL">

```

```

<description>Control on-board Camera Control System to take shots.</description>

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="uint8_t" name="session">0: stop, 1: start or keep it up
//Session control e.g. show/hide lens.</field>

<field type="uint8_t" name="zoom_pos">1 to N //Zoom's absolute position (0 means
ignore).</field>

<field type="int8_t" name="zoom_step">-100 to 100 //Zooming step value to offset
zoom from the current position.</field>

<field type="uint8_t" name="focus_lock">0: unlock focus or keep unlocked, 1: lock
focus or keep locked, 3: re-lock focus.</field>

<field type="uint8_t" name="shot">0: ignore, 1: shot or start filming.</field>

<field type="uint8_t" name="command_id">Command Identity
(incremental loop: 0 to 255)//A command sent multiple times will be executed or
pooled just once.</field>

<field type="uint8_t" name="extra_param">Extra parameters enumeration (0
means ignore).</field>

<field type="float" name="extra_value">Correspondent value to given
extra_param.</field>

</message>

<!-- Camera Mount Messages -->

<message id="156" name="MOUNT_CONFIGURE">

<description>Message to configure a camera mount, directional antenna,
etc.</description>

<field type="uint8_t" name="target_system">System ID.</field>

```

```

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="uint8_t" name="mount_mode"
enum="MAV_MOUNT_MODE">Mount operating mode.</field>

<field type="uint8_t" name="stab_roll">(1 = yes, 0 = no).</field>

<field type="uint8_t" name="stab_pitch">(1 = yes, 0 = no).</field>

<field type="uint8_t" name="stab_yaw">(1 = yes, 0 = no).</field>

</message>

<message id="157" name="MOUNT_CONTROL">

<description>Message to control a camera mount, directional antenna,
etc.</description>

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="int32_t" name="input_a">Pitch (centi-degrees) or lat (degE7),
depending on mount mode.</field>

<field type="int32_t" name="input_b">Roll (centi-degrees) or lon (degE7)
depending on mount mode.</field>

<field type="int32_t" name="input_c">Yaw (centi-degrees) or alt (cm) depending on
mount mode.</field>

<field type="uint8_t" name="save_position">If "1" it will save current trimmed
position on EEPROM (just valid for NEUTRAL and
LANDING).</field>

</message>

<message id="158" name="MOUNT_STATUS">

<description>Message with some status from autopilot to GCS about camera or
antenna mount.</description>

```

```

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="int32_t" name="pointing_a" units="cdeg">Pitch.</field>

<field type="int32_t" name="pointing_b" units="cdeg">Roll.</field>

<field type="int32_t" name="pointing_c" units="cdeg">Yaw.</field>

<extensions/>

<field type="uint8_t" name="mount_mode"
enum="MAV_MOUNT_MODE">Mount operating mode.</field>

</message>

<!-- geo-fence messages -->

<message id="160" name="FENCE_POINT">

<description>A fence point. Used to set a point when from GCS -&gt; MAV. Also
used to return a point from MAV -&gt; GCS.</description>

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="uint8_t" name="idx">Point index (first point is 1, 0 is for return
point).</field>

<field type="uint8_t" name="count">Total number of points (for sanity
checking).</field>

<field type="float" name="lat" units="deg">Latitude of point.</field>

<field type="float" name="lng" units="deg">Longitude of point.</field>

</message>

<message id="161" name="FENCE_FETCH_POINT">

<description>Request a current fence point from MAV.</description>

```

```

<field type="uint8_t" name="target_system">System ID.</field>

<field type="uint8_t" name="target_component">Component ID.</field>

<field type="uint8_t" name="idx">Point index (first point is 1, 0 is for return
point).</field>

</message>

<message id="163" name="AHRS">

<description>Status of DCM attitude estimator.</description>

<field type="float" name="omegalx" units="rad/s">X gyro drift estimate.</field>

<field type="float" name="omegaly" units="rad/s">Y gyro drift estimate.</field>

<field type="float" name="omegalz" units="rad/s">Z gyro drift estimate.</field>


<field type="float" name="accel_weight">Average accel_weight.</field>

<field type="float" name="renorm_val">Average renormalisation value.</field>

<field type="float" name="error_rp">Average error_roll_pitch value.</field>

<field type="float" name="error_yaw">Average error_yaw value.</field>

</message>

<message id="164" name="SIMSTATE">

<description>Status of simulation environment, if used.</description>

<field type="float" name="roll" units="rad">Roll angle.</field>

<field type="float" name="pitch" units="rad">Pitch angle.</field>

<field type="float" name="yaw" units="rad">Yaw angle.</field>

<field type="float" name="xacc" units="m/s/s">X acceleration.</field>

<field type="float" name="yacc" units="m/s/s">Y acceleration.</field>

<field type="float" name="zacc" units="m/s/s">Z acceleration.</field>

```

```

<field type="float" name="xgyro" units="rad/s">Angular speed around X
axis.</field>

<field type="float" name="ygyro" units="rad/s">Angular speed around Y
axis.</field>

<field type="float" name="zgyro" units="rad/s">Angular speed around Z
axis.</field>

<field type="int32_t" name="lat" units="degE7">Latitude.</field>

<field type="int32_t" name="lng" units="degE7">Longitude.</field>

</message>

```

```

<message id="165" name="HWSTATUS">

<description>Status of key hardware.</description>

<field type="uint16_t" name="Vcc" units="mV">Board voltage.</field>

<field type="uint8_t" name="I2Cerr">I2C error count.</field>

</message>

```

```

<message id="166" name="RADIO">

<description>Status generated by radio.</description>

<field type="uint8_t" name="rssi">Local signal strength.</field>

<field type="uint8_t" name="remrssi">Remote signal strength.</field>

<field type="uint8_t" name="txbuf" units="%">How full the tx buffer is.</field>

<field type="uint8_t" name="noise">Background noise level.</field>

<field type="uint8_t" name="remnoise">Remote background noise level.</field>

<field type="uint16_t" name="rxerrors">Receive errors.</field>

<field type="uint16_t" name="fixed">Count of error corrected packets.</field>

</message>

```

<!-- AP\_Limits status -->

<message id="167" name="LIMITS\_STATUS">

<description>Status of AP\_Limits. Sent in extended status stream when AP\_Limits is enabled.</description>

<field type="uint8\_t" name="limits\_state" enum="LIMITS\_STATE">State of AP\_Limits.</field>

<field type="uint32\_t" name="last\_trigger" units="ms">Time (since boot) of last breach.</field>

<field type="uint32\_t" name="last\_action" units="ms">Time (since boot) of last recovery action.</field>

<field type="uint32\_t" name="last\_recovery" units="ms">Time (since boot) of last successful recovery.</field>

<field type="uint32\_t" name="last\_clear" units="ms">Time (since boot) of last all-clear.</field>

<field type="uint16\_t" name="breach\_count">Number of fence breaches.</field>

<field type="uint8\_t" name="mods\_enabled"

enum="LIMIT\_MODULE" display="bitmask">AP\_Limit\_Module bitfield of enabled modules.</field>

<field type="uint8\_t" name="mods\_required"

enum="LIMIT\_MODULE" display="bitmask">AP\_Limit\_Module bitfield of required modules.</field>

<field type="uint8\_t" name="mods\_triggered"

enum="LIMIT\_MODULE" display="bitmask">AP\_Limit\_Module bitfield of triggered modules.</field>

</message>

<message id="168" name="WIND">

<description>Wind estimation.</description>

<field type="float" name="direction" units="deg">Wind direction (that wind is coming from).</field>

<field type="float" name="speed" units="m/s">Wind speed in ground plane.</field>

<field type="float" name="speed\_z" units="m/s">Vertical wind speed.</field>

</message>

<message id="169" name="DATA16">

<description>Data packet, size 16.</description>

<field type="uint8\_t" name="type">Data type.</field>

## CHAPTER 6

### RESULT AND DISCUSSION

#### 6.1 RESULT

In this study of this project here are some challenges that are been focused on,

##### ***6.1.1 Optimizing battery life:***

Background sensors can be a significant drain on a drone's battery as they often run continuously. Identifying which sensors are essential to remaining active in the background and which can be deactivated is a key challenge [28].

Additionally, human interaction with drones, such as taking photos or videos, can quickly deplete the battery. Balancing user-friendliness and minimizing battery consumption presents a design challenge [26].

##### ***6.1.2 Handling human interaction:***

Obstacle avoidance is a crucial aspect of drone flight, especially in urban settings. Detecting and responding to obstacles swiftly poses a challenge. Developing obstacle avoidance systems that are dependable and efficient is essential [31]. The drones frequently require long-distance data transmission, like video footage or sensor data. This is complicated by the fact that drones often operate in areas with limited cellular coverage. A challenge lies in creating data transmission systems that are both robust and capable of high bandwidth [8].

##### ***6.1.3 Data Transmission:***

Power consumption is a concern when it comes to data transmission, as it can deplete a drone's battery life. A challenge is to design energy-efficient data transmission systems. Bandwidth is another important aspect of data transmission systems. They must be capable of transmitting data at a high enough bandwidth to meet the application's requirements [24]. For example, when a drone is transmitting video footage, the data transmission system must handle high-resolution and high frame-rate video. Developing data transmission systems capable of efficiently transmitting high-bandwidth data over long distances is a challenge.

## 6.2 DISCUSSION ON FUTURISTIC SCOPE

To optimize battery life, hybrid power systems, which combine batteries with other power sources like solar panels or fuel cells, can be used. These systems allow drones to recharge their batteries during flight, extending their operational time. Reducing human interaction by implementing artificial intelligence (AI) can save battery life [38]. AI can automate tasks such as capturing images and videos or controlling the drone's flight path, minimizing the need for human input [29]. Obstacle avoidance can be achieved using LiDAR (light detection and ranging) sensors. LiDAR sensors create 3D maps of the drone's surroundings, aiding obstacle detection. As these sensors become more affordable and compact, they become a practical option for drones.

For improved data transmission, mesh networks and satellite-based data links are viable solutions. Mesh networks enable direct communication between devices without the need for a central base station, enhancing reliability in areas with poor cellular coverage [22]. Satellite-based data links can transmit data over long distances, although they may be costlier and bulkier. The integration of AI into drone technology has the potential to be transformative. AI-equipped drones can operate with greater autonomy, intelligence, and efficiency, expanding their utility across various applications and making them accessible to a broader user base.

The scope of autonomous delivery drone systems extends across various industries and applications, including e-commerce, medical supply transport, food and grocery delivery, disaster relief, agriculture, environmental monitoring, and more [20]. These systems are also adaptable, with different sizes, payload capacities, and ranges, catering to diverse needs. The potential for autonomous delivery drones to transform the delivery landscape is substantial, enhancing efficiency, reducing costs, and improving convenience [9].

## **CHAPTER 7**

### **SUMMARY AND CONCLUSION**

#### **7.1 SUMMARY**

Autonomous delivery drones are revolutionizing logistics by efficiently transporting goods and services. Their scope spans e-commerce, medical supplies, food delivery, disaster relief, and more. Despite challenges like payload limitations, battery life, and regulations, these drones promise to transform last-mile delivery, reduce costs, and enhance convenience in various industries.

#### **7.2 CONCLUSION**

Autonomous UAVs offer vast potential for future applications across various industries, promising increased efficiency, and convenience. This paper explores solutions for battery optimization, network connectivity, and design customization. Leveraging AI, new battery management systems can extend flight times, while improved network protocols enhance reliability. AI-driven drone designs are more aerodynamic and efficient, emphasizing the transformative role of technology in drone development.

## REFERENCES

- [1] Abdul Aabid, "Reviews on Design and Development of Unmanned Aerial Vehicle (Drone) for Different Applications," Research Gate, vol. 25, 2022, pp. 101-103.
- [2] Aurello Patrik, "GNSS-based navigation systems of autonomous drone for delivering items," Springer, vol. 6, 2019, pp.77-83.
- [3] Baek. D, Chen. Y, Bocca. A, Macii. A, Macii. E, and Poncino. M , "Battery-Aware Energy Model of Drone Delivery Tasks.", ACS, vol. 05, 2021, pp. 56-58.
- [4] Cevik. P, Kocaman. I, Akgul. A.S, and Akca. B, "The small and silent force multiplier: A swarm UAV-electronic attack," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 70, 2013, pp. 595–608.
- [5] Choi Y. S and Yeo. G. T, "An analysis of research trends of inland waterway utilizing social network analysis," Asian Journal of Shipping and Logistics, vol. 39, 2023, pp. 52–59.
- [6] De Fazio. R, Dinoi. L .M,sDe Vittorio. M , and Visconti. P , "A sensor-based drone for pollutants detection in eco-friendly cities: Hardware design and data analysis application," Electronics, vol. 11, no. , Jan. 2022, pp. 23-31.
- [7] Deepika. M, Pan. T, An. L , Wen. C, "Recent advances in fuel cells based propulsion systems for unmanned aerial vehicles", IEEE, vol. 8, 2019, pp.13.
- [8] Dhulashia. D, Peters. A, Horne. C, Beasley. P, and Ritchie. M, "Multi-Frequency Radar Micro-Doppler Based Classification of Micro-Drone Payload Weight," Frontiers in Signal Processing, vol. 1, 2021, pp. 78-82.
- [9] Diaz. A. L, "The Bathy-Drone: An Autonomous Unmanned Drone-Tethered Sonar System," Drones, MDPI, vol. 6, 2022, pp. 220.
- [10] Dukowitz. Z, "A Look at How They Work and Important Terminology uavcoach.com/drone-controller Drone Controllers: A Look at How They Work, Important Terminology, and Why They're Unique in the RC Aircraft World," AAAI, vol.02, 2019. pp. 18-26

- [11] Goncalves. R. S, "Drone–Robot to Clean Power Line Insulators," *Sensors*, vol. 23, 2023,pp.12-15.
- [12] Hadiwardoyo. S. A, Hernandez. S. A, Calafate C. T, Cano J. C., and Manzoni. P , "Experimental characterization of UAV-to-car communications," *Computer Networks*, vol. 136, 2018, pp. 105-118.
- [13] Javidsharifi. M, Arabani. H. P, Kerekes. T, Sera.D , Spataru. S.V , and Guerrero. J.M , "Optimum sizing of photovoltaic-battery power supply for drone-based cellular networks," *IEEE*, vol. 5, 2021,pp. 18-16.
- [14] Karpathakis. S. F. E, "Ground-to-Drone Optical Pulse Position Modulation Demonstration as a Testbed for Lunar Communications," *Drones*, vol. 7,2023, pp.20-23
- [15] Lee. T, McKeever. S, and Courtney. J , "Flying free: A research overview of deep learning in drone navigation autonomy," *Drones*, vol. 5, 2021,pp.50-63.
- [16] Madani. B and Ndiaye. M , "Hybrid Truck-Drone Delivery Systems: A systematic literature review," *IEEE Access*, vol. 12, 2022, pp. 88-91.
- [17] Maher Aljehani, Masahiro Inoue, Akira Watanabe, Taketoshi Yokemura, Fumiya Ogyu, and Hidemasa Iida, "UAV communication system integrated into network traversal with mobility," *Springer*, vol. 2, 2020, pp. 213-218.
- [18] Mukhopadhyay. S, Fernandes. S, Shihab. M, and Waleed. D, "Using small capacity fuel cells onboard drones for battery cooling: An experimental study," *Sciences*, vol. 8, 2018, pp. 03-11.
- [19] Musa. S, "Techniques for Quadcopter Modelling & Design: A Review," *Journal of Unmanned System Technology*, MDPI, vol. 07, 2018, pp. 25-27.
- [20] N. T. K. Chi, L. T. Phong, and N. T. Hanh, "The drone delivery services: An innovative application in an emerging economy," *Asian Journal of Shipping and Logistics*, vol. 39, 2023, pp. 39-45.
- [21] Panigrahi. S, Krishna. Y.S.S, and Thondiyath. A, "Design, analysis, and testing of a hybrid VTOL tilt-rotor UAV for increased endurance," *Sensors*, vol. 21, 2021, pp. 308-321.

- [22] Ramya. A, Omidkar. R, Roudi. N P, Abbas. R, and Kim. S, "Machine-learning-enabled intrusion detection system for cellular-connected UAV networks," *Electronics* , vol. 10, 2021, pp. 56-61.
- [23] Rajalingam. S , S. Kanagamalliga, Priya. K. S, and Karuppiah. N, "Optimizing Drone delivery: An efficient design for shipper applications," *Journal of Autonomous Intelligence*, vol. 7, 2023, pp. 37-41.
- [24] Ramadani. R. A and Aini. Y.K, "Analysis of Performance of 433 MHz Telemetry Devices for Data Transmission in Unmanned Aerial Vehicle (UAV)", *MDPI*, vol. 2, 2018, pp. 47-49.
- [25] Rania Elgohary, "Software Development for Autonomous Unmanned Aerial Vehicle," *Springer*, vol. 11757, 2021, pp. 57–64.
- [26] Rohan. A, Rabah. M, Asghar. F, Talha. M, and Kim. S. K, "Advanced Drone Battery Charging System," *Journal of Electrical Engineering and Technology*, vol. 14, 2019, pp. 1395–1405.
- [27] Sarayu. A, "Advances in Aerospace Technology, "Recent advances in fuel cells based propulsion systems for unmanned aerial vehicles, applied energy," *MDPI*, vol. 3, 2019, pp.56-58.
- [28] Schonfeld. P, "Optimization of Multi-package Drone Deliveries Considering Battery Capacity." *Research Gate*, vol.02, 2019. pp. 18-26.
- [29] Shahzad Hameed, "Connectivity of Drones in FANETs Using Biologically Inspired Dragonfly Algorithm (DA) through Machine Learning", *Research Gate*, vol. 2022, 2022, pp-235-246.
- [30] Shankar. N. V. S, "Design and Fabrication of Quadcopter Drone for Delivery.", *CVF*, vol. 8, pp. 73-78.
- [31] Stephan. F, Reinsperger. N, Grünthal. M, Paulicke. D, and Jahn. P, "Human drone interaction in delivery of medical supplies: A scoping review of experimental studies," *PLoS One*, vol. 17, 2022, pp. 06-09
- [32] Sudhakar. T and Ramadass. O, "A study on drone stability on lifting and hovering with the CTD sensor and instrumentation payload for ocean observation applications," *MDPI*, vol.13, 2021, pp. 22-27.

- [33] Szczupak. P and Kossowski. T, "Response of drone electronic systems to a standardized lightning pulse," *Energies* (Basel), vol. 14, 2021, pp. 48-51.
- [34] Ullah. M, "Design and Fabrication of a Quadcopter Delivery Drone", *ACS*, vol. 4, 2021, pp. 56-58.
- [35] Umut Can Cabuk, Mustafa Tosun, and Vahid Khalilpour Akram, "Connectivity Management in Drone Networks: Models, Algorithms and Methods," *Research Gate*, vol. 07, 2021. pp. 96-102.
- [36] Varma. S. N , Vangari. P , Kumar. S, Joshi. K, and Penta. S, "Integrating Aerial Electric Vehicles for Sustainable Agriculture and to Optimize the Overhead Cost of Farming," *E3S Web of Conferences*, vol. 430,2023, pp-1003.
- [37] Verstraete K. L. K. C. W. D., "Design of a Fuel Cell Powered Blended Wing Body UAV," *Advances in Aerospace Technology*, ASME, vol. 09, 2012, pp. 09-14.
- [38] Youssef Amr M. and Riham Altawy, "Security, Privacy, and Safety Aspects of Civilian Drones: A Survey," *Research Gate*, vol. 4, 2016, pp. 13-20.

# DESIGN AND FABRICATION OF AUTONOMOUS DRONE DELIVERY SYSTEM

## ORIGINALITY REPORT

7 %

SIMILARITY INDEX

5 %

INTERNET SOURCES

2 %

PUBLICATIONS

3 %

STUDENT PAPERS

## PRIMARY SOURCES

1

[sist.sathyabama.ac.in](http://sist.sathyabama.ac.in)

Internet Source

2 %

2

[ieomsociety.org](http://ieomsociety.org)

Internet Source

1 %

3

Submitted to Engineers Australia

Student Paper

<1 %

4

Md R Haque, M Muhammad, D Swarnaker, M Arifuzzaman. "Autonomous Quadcopter for product home delivery", 2014 International Conference on Electrical Engineering and Information & Communication Technology, 2014

Publication

<1 %

5

[files.osf.io](http://files.osf.io)

Internet Source

<1 %

6

Submitted to Woodcroft College

Student Paper

<1 %

7

Mohamed Nadir Boukoberine, Zhibin Zhou, Mohamed Benbouzid. "A critical review on

<1 %

# unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects", Applied Energy, 2019

Publication

|    |                                                                                                                                                           |      |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 8  | encyclopedia.pub<br>Internet Source                                                                                                                       | <1 % |
| 9  | "The Future of Drone Use", Springer Science and Business Media LLC, 2016<br>Publication                                                                   | <1 % |
| 10 | satprnews.com<br>Internet Source                                                                                                                          | <1 % |
| 11 | Submitted to University of Queensland<br>Student Paper                                                                                                    | <1 % |
| 12 | Salwa Elouarouar, Hicham Medromi. "Multi-Rotors Unmanned Aerial Vehicles Power Supply and Energy Management", E3S Web of Conferences, 2022<br>Publication | <1 % |
| 13 | kb.psu.ac.th:8080<br>Internet Source                                                                                                                      | <1 % |
| 14 | etd.lib.metu.edu.tr<br>Internet Source                                                                                                                    | <1 % |
| 15 | financedocbox.com<br>Internet Source                                                                                                                      | <1 % |
| 16 | Batool Madani, Malick Ndiaye. "Hybrid Truck-Drone Delivery Systems: A systematic                                                                          | <1 % |

|    |                                                                                                                                                                                                                   |      |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 17 | <a href="https://journals.cihanuniversity.edu.iq">journals.cihanuniversity.edu.iq</a><br>Internet Source                                                                                                          | <1 % |
| 18 | <a href="https://uavcoach.com">uavcoach.com</a><br>Internet Source                                                                                                                                                | <1 % |
| 19 | Submitted to Engineering Institute of Technology<br>Student Paper                                                                                                                                                 | <1 % |
| 20 | <a href="https://www.bioenergyconsult.com">www.bioenergyconsult.com</a><br>Internet Source                                                                                                                        | <1 % |
| 21 | Submitted to University of Petroleum and Energy Studies<br>Student Paper                                                                                                                                          | <1 % |
| 22 | <a href="https://ijirset.com">ijirset.com</a><br>Internet Source                                                                                                                                                  | <1 % |
| 23 | <a href="https://www.frontiersin.org">www.frontiersin.org</a><br>Internet Source                                                                                                                                  | <1 % |
| 24 | Nishat I. Mowla, Nguyen H. Tran, Inshil Doh, Kijoon Chae. "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET", Journal of Communications and Networks, 2020<br>Publication | <1 % |
| 25 | <a href="https://web.archive.org">web.archive.org</a><br>Internet Source                                                                                                                                          | <1 % |

26

[worldwidescience.org](http://worldwidescience.org)

Internet Source

<1 %

27

[www.american tv.com](http://www.american tv.com)

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On