Intersect, Vol 19, No 3 (2026)

Drones for Autonomous Medical Aid

Jiwoo Jung
University of Washington

Abstract

For underserved slum communities, access to important medical resources is a significant challenge, especially if they are in a densely populated environment where there is poor infrastructure, limited transportation, and environmental barriers that block access to healthcare services. These challenges delay the transportation of the medication and the supplies, which leads to a poorer health outcome. Unmanned aerial vehicles (UAVs), commonly known as drones, are a promising solution that could help overcome these challenges by allowing fast and direct delivery of critical medical supplies. This study investigates the deployment of low-cost autonomous UAVs to deliver temperature-sensitive medical supplies within underserved slum communities, focusing on communities in sub-Saharan Africa, such as Kibera, Kenya. A within-subjects experimental design was used, in which a low-cost drone prototype was developed and tested through a series of online flight simulations and tests designed to assess autonomous navigation, payload capacity, environmental resilience, and temperature-controlled medication transport. The performance was assessed through different payload conditions, stability, battery performance, motor temperature, and whether the insulated payload stayed at the safe medication storage temperatures. The findings showed that autonomous flight operations, including takeoff, navigation, hovering, altitude control, and landing, could be achieved using inexpensive drone hardware. They also indicated that payload weight significantly influenced flight duration and efficiency, and that thermal management was a limitation for long-range deployment. The insulated payload system, however, successfully maintained proper temperatures throughout testing. Overall, the study supports the viability of autonomous drone-assisted healthcare delivery in underserved communities and shows the potential of low-cost UAV systems to improve access to essential medicines. Future research should focus on large-scale field testing, better energy systems, and integration with existing healthcare infrastructure to evaluate real-world implementation and scalability.

Introduction

This study aims to test whether deploying low-cost, fully autonomous unmanned aerial vehicles (UAVs) to deliver temperature-sensitive medical supplies in underserved slum communities is a possibility in the future. Drone-assisted healthcare is a growing interest in the world, but there is a gap in research about whether inexpensive, consumer-grade drone hardware can deliver medicine and supplies while being fully autonomous in a resource-limited environment. This study addresses this gap by developing and testing a low-cost drone prototype through a series of flight simulations, assessing autonomous navigation, payload capacity under different weights, environmental resilience, and temperature-controlled transport. The findings provide information to the goal of improving access to medicine among the underserved communities.

Settlement Morphology and Health in Slums

Settlement patterns in slums greatly influence residents' health outcomes, highlighting the importance of infrastructure-sensitive health interventions. Overcrowding and substandard living conditions often worsen the spread of communicable diseases such as cholera, typhoid, and respiratory infections (Friesen et al., 2020a; Friesen et al., 2020b). Slums in South Africa, such as Khayelitsha in Cape Town and Soweto in Johannesburg, are characterized by informal housing constructed from materials like scrap metal and plastic (Mwai, 2022). These homes are tightly packed, with only narrow dirt paths connecting them. During the rainy season, these pathways become unusable due to mud and flooding, further isolating communities from essential medical services (Lilford et al., 2017; *Urban Adaptation*, 2022).

Moreover, poor sanitation systems, including open drains and the accumulation of uncollected garbage, contribute to the increase of vector-borne diseases like malaria and dengue (*Slum Health*, 2021; Mechan et al., 2023). Addressing these settlement-specific health risks requires innovative strategies, including the use of technology such as Geographic Information Systems (GIS) to create detailed maps of slum layouts. These maps could allow specific healthcare interventions and resource distribution, ensuring that medical services reach the most vulnerable populations to these diseases effectively(Desgroppes & Taupin, 2011; Lilford et al., 2017).

Barriers to Healthcare Access

One of the main challenges of healthcare in slums is the difficulty in accessing medical services due to geographic, financial, and systemic barriers. Park et al. (2022) describe how the financial burden of healthcare, combined with the high cost of transportation and a lack of proximate facilities, leaves many residents without treatment (Yenet et al., 2023). These obstacles make it challenging for slum residents to seek medical care.

Transportation issues, in particular, highlight the crucial role of proximity in healthcare accessibility. For slum residents, traveling to distant healthcare centers often means taking time off work, which many cannot afford. Initiatives such as mobile health clinics, telemedicine, and community health worker programs have shown promise in addressing these challenges by spreading out healthcare services and reducing associated costs. These solutions have the potential to not only improve access but also restore trust in healthcare systems, promoting long-term engagement between communities and healthcare providers (Quick et al., 2005).

Community-Centric Health Solutions

Lilford et al. (2017) highlight the importance of community-based interventions to improve health outcomes in slums, emphasizing the need for accessible health services. Autonomous drones can complement community health efforts by providing a quick and efficient way to deliver medications, vaccines, or other supplies during emergencies (Roberts et al., 2023; Mechan et al., 2023). By making drones deliver supplies even after regular clinic hours, slum populations could have more access to necessary medical products, addressing one of the major gaps in healthcare provision (Roberts et al., 2023). Moreover, integrating drones into community health initiatives can strengthen efforts to improve health services and provide a reliable solution for distributing health resources across large, densely populated areas (Mechan et al., 2023; Mohsan et al., 2023).

Challenges in Medicine Availability

The availability of essential medicines in slums is restricted by systemic inefficiencies, economic constraints, and logistical challenges. Yenet et al. (2023) highlight that disorganized supply chains and limited government funding often result in frequent stockouts of crucial medicines (World Health Organization, 2021). Quick et al. (2005) similarly note that the high cost of medications, in addition to inadequate distribution systems, affects low-income populations (Quick et al., 2005; Yenet, Nibret, & Tegegne, 2023).

Diseases such as malaria, pneumonia, and tuberculosis require consistent availability of medications like antimalarials, antibiotics, and vaccines (Mechan et al., 2023). Additionally, non-communicable diseases (NCDs) such as diabetes and hypertension are increasingly prevalent in slums, which need a steady supply of medications like insulin and antihypertensives (Yenet et al., 2023). Mechan et al. (2023) emphasize the need for stronger supply chain management, which includes better distribution of medicine demand and investments in cold-chain logistics to ensure that vaccines and temperature-sensitive drugs reach their destinations in usable condition (Mohsan et al., 2023).

Population Density and Route Planning

In Kibera, a densely populated slum area in Nairobi, Kenya, several medicines are important in addressing public health needs. One major need is deworming medications, particularly to combat soil-transmitted helminths (STH). The World Health Organization recommends regular deworming programs for children in areas like Kibera, where the majority of helminth infections are high due to poor sanitation and overcrowding (Harris et al., 2013). Medications such as albendazole and mebendazole are commonly used to treat these infections.

In addition to deworming, there is a significant need for vaccines and antibiotics to manage common diseases, such as respiratory infections and diarrhea, which are found due to the lack of clean water and sanitation (Desgroppes et al., 2011).

Given Kibera's high population density, which can reach up to 129,000 inhabitants per square kilometer, there is an urgent need for accurate mapping to optimize medical delivery (Desgroppes & Taupin, 2011; Friesen, Taubenböck, Wurm, & Pelz, 2020). Desgroppes et al. (2011) highlight the importance of using Geographic Information Systems (GIS) and remote sensing to plan delivery routes of medication effectively. With these tools, we can navigate these densely populated environments, ensuring they avoid overcrowded areas and deliver medical supplies directly to households or clinics that need them. This integration of population data and real-time navigation systems could enhance the efficiency of delivering essential medications such as deworming tablets, vaccines, and antibiotics, eventually improving health outcomes in slums like Kibera (Rodrigues et al., 2022).

Ensuring Stability and Effectiveness in HIV Medication Storage

Proper storage of HIV medications is important to maintain their effectiveness and safety. Most antiretroviral drugs should be stored at controlled room temperatures, typically between 15°C and 25°C (59°F to 77°F), in a cool, dry place away from direct sunlight (Heelon, 2023; Mohsan et al., 2023). This temperature range helps preserve the medication's stability and effectiveness (Jarrah et al., 2022).

Exposure to temperatures outside this recommended range can compromise medication stability. For instance, studies have shown that certain drugs may degrade when stored at temperatures exceeding 25°C, potentially reducing their effectiveness (Heelon, 2023). Contrarily, cooler temperatures can impact the properties of some medications, making precise temperature control essential.

In cases where temperature changes are unavoidable, solutions such as insulated packaging or phase-change materials can help maintain the required storage temperatures (Grepow, 2024; Prokopets, 2024). Additionally, temperature monitoring devices are often used to ensure medications remain within the safe temperature range throughout storage and transport (Rodrigues et al., 2021).

Proper dedication to these storage guidelines ensures that HIV medications remain effective, providing consistent and reliable treatment for individuals who depend on them (Mechan et al., 2023).

Methodology/Experimental Design

This study uses a within-subjects experimental design where a low-cost prototype of a drone was built and tested across two flight phases. Phase 1 was conducted in a controlled environment, focusing on baseline autonomous flight behaviors, while Phase 2 tested the drone's performance under various payload weights. All of the flight tests were conducted at Gapyeong, South Korea. This place was used because it had open terrain, variable weather conditions, and minimal air traffic restrictions. Data that were collected for both phases were flight duration, altitude stability, motor temperature, battery performance, and internal payload temperature. These data were used to see whether the drone was able to autonomously deliver medical supplies in underserved slum environments.

Research Analysis Method

The research question focused on developing a drone capable of carrying heavy payloads over long distances, particularly for delivering medical supplies to underserved areas such as South African slums. The project methodology utilized a within-subjects design, modifying and testing prototypes to address technical and environmental challenges. This ensured the final design was optimized for real-world scenarios.

The development of drone technology, as highlighted by Mohsan et al. (2023), has been driven by a combination of innovative ideas and practical needs, such as disaster monitoring, farming, and remote sensing. While current drones faced limitations like short flight times, limited autonomy, and constrained payload capacities, advancements in technologies like battery-swapping systems, mobile charging stations, and hybrid propulsion systems, as discussed by Jarrah et al. (2022), were beginning to address these issues. These innovations were essential for drones used in healthcare logistics, where reliability, flexibility, and long flight durations are critical.

Review of Related UAV Applications

UAVs in Emergency Medical Services

In the healthcare world, drones have become a potential game-changer, especially for emergency medical services (EMS). The article by Roberts et al. (2023) highlights how drones can transform the delivery of essential medical supplies quickly and efficiently during emergencies. Their ability to improve response times for life-threatening conditions, like cardiac arrests and strokes, shows how important it is to get help fast in healthcare settings. For instance, in out-of-hospital cardiac arrest (OHCA) situations, drones have been able to deliver automated external defibrillators (AEDs) faster than ground-based EMS. This is huge for survival rates, as the study mentions that drones successfully delivered AEDs in 92% of cases and arrived faster than traditional EMS in 64% of instances.

Additionally, the study explains how drones can also deliver other important medical supplies, such as blood and anti-epileptic drugs, showcasing their adaptability in different emergencies. The researchers conducted surveys and interviews, finding that bystanders and healthcare providers generally support using drones in EMS. They see them as reliable and quick delivery systems for medical aid. The article highlights how efficient drones are and also discusses public perception, which is important for getting new healthcare technologies accepted.

However, UAVs in healthcare still face challenges, like limited payload capacities and regulatory issues that need to be worked out before they can become a standard part of EMS. Despite these challenges, Roberts et al. suggest that the benefits, especially regarding response time and patient outcomes, outweigh the downsides. Drones can operate in places where traditional EMS might struggle, like crowded urban areas or remote locations, making them a valuable asset for the future of emergency medical response.

UAVs in Environmental Health and Disease Control

Beyond just medical services, UAVs are becoming important in environmental health, especially for controlling diseases carried by insects, like malaria. Mechan et al. (2023) look into how drones are being used to manage these diseases. Drones have proven to be effective at delivering larvicides and adulticides to remote areas that are hard to reach using regular methods. For example, in a study from Zanzibar, Tanzania, using drones led to a 90% drop in the population of *Anopheles arabiensis* mosquitoes, which are a major carrier of malaria. This shows how effective UAVs can be in places where traditional ground methods struggle due to environments such as bad weather or tough terrain.

In South African slum areas, weather conditions like strong winds present additional challenges to health interventions. High winds during seasonal shifts and storms can damage infrastructure and delay medical deliveries. Urban Adaptation (2022) highlights that slum communities are particularly vulnerable to wind-related risks, emphasizing the importance of UAVs with wind-resistance capabilities. These drones could maintain stability and precision in adverse weather, ensuring critical health supplies are delivered safely and efficiently.

One of the biggest benefits of using drones in disease control is their precision and efficiency. Drones can fly below cloud cover, making them useful in various weather conditions, and they can deliver their payloads accurately, ensuring that treatments are targeted and effective. However, Mechan et al. point out that there are some challenges too. For instance, there are regulations that need to be followed, and some communities might not welcome UAVs, especially when it comes to health-related interventions. Cost is another issue. The use of drones needs to be justified by looking at both their operating costs and the health benefits they provide.

Even with these challenges, the use of drones for environmental purposes, especially in public health, is growing quickly. Their ability to monitor large and hard-to-reach areas makes them essential tools in global health efforts, like malaria control, and in adapting to extreme weather conditions.

Scaling Down the Drone

Scaling down the drone to fit the capabilities of a standard consumer drone, like the METAVIZ 8K V168, requires a lot of adjustments to the payload, flight range, and overall logistics. The METAVIZ 8K V168 has a payload capacity of approximately 0.2–0.5 kg, a battery life of around 25-30 minutes, and a flight range of about 6 km round trip, with a cruising speed of roughly 25-30 km/h. To accommodate these limitations, the payload must be reduced from the original 3–5 kg to fit within the drone's capacity. For instance, if the original payload serves 100 patients, scaling down would require limiting the delivery to 5–10 patients per trip.

Temperature-sensitive medications can still be carried using lightweight insulated packaging weighing between 0.2 and 0.5 kg. These packagings employ phase-change materials or vacuum-sealed designs to ensure the temperature stays within the 15°C–25°C range for up to 30 minutes.

The route also needs to be adjusted to ensure that the healthcare facility is within a 3 km radius of the Kibera clinic and that it stays within the drone's maximum range. This shorter route still requires identifying emergency landing sites along the way for safety. With a cruising speed of 25–30 km/h, the drone can complete a one-way trip in approximately 10 minutes, leaving enough battery capacity for the return journey and an additional buffer of around 5–10 minutes.

Most consumer drones, including the METAVIZ 8K V168, lack advanced delivery systems, and a simple gravity-based release or tether mechanism can be used to offload the payload at the drop-off point. This minimizes the need for extensive modifications while ensuring reliable delivery. In conclusion, scaling down for the METAVIZ 8K V168 involves reducing payload weight, limiting flight distances, optimizing lightweight temperature-controlled packaging, and utilizing simple delivery methods. These adjustments ensure the drone operates within its constraints while maintaining the efficiency and safety of medication delivery.

Experiment Design

1. Develop Initial Prototype and Baseline Configuration

The first step involved defining the drone's baseline configuration, which includes key parameters such as weight, material composition, motor power, and battery specifications. Based on the initial design

requirements, specifications for the battery type, motor output, frame composition, and payload capacity were determined. Lightweight, durable materials like carbon fiber or aluminum were selected for the frame to balance strength and weight reduction. For the motor system, brushless DC motors were chosen, typically configured in 4–8 motor setups to deliver the necessary thrust (3–6 kg per motor) for stable operation, even with heavy payloads. To achieve extended flight durations of at least 5 hours, high-capacity Lithium-ion (Li-ion) or Lithium-polymer (LiPo) batteries, typically in the 50,000 mAh range, were used to store the required energy. Once the drone is built, initial diagnostics were performed to ensure all components are functioning correctly before advancing to flight tests.
The solutions highlighted by Mohsan et al. (2023), such as battery-swapping systems and precision landing features, can be incorporated to enhance the prototype's operational capabilities. These advancements are key to improving flight times and autonomy, ensuring the drone can meet the demands of medical logistics. However, these improvements were outside the scope of this study and were not implemented in the prototype.

2. Controlled Environment Testing (Phase 1)
   The first phase of testing took place in a controlled indoor environment, such as a gymnasium, to eliminate external variables like wind or temperature fluctuations that could skew results. During these tests, metrics such as battery usage, motor temperature, and altitude stability were closely monitored. A standardized payload, designed to simulate medical supplies, was used to maintain consistency across trials. Key performance indicators such as flight times, battery levels, motor performance, and overall stability were recorded, with data averaged across the trials to confirm the drone's capabilities.
3. Varying Payload Testing (Phase 2)
   In this phase, the drone's performance was evaluated with varying payload weights to assess its payload flexibility and stability under different conditions. The payloads were categorized as light, medium, and maximum capacity to understand how varying loads impact flight performance. Flight duration, battery drainage rates, and altitude stability were recorded for each payload category. Additionally, video recordings of the flights were recorded to capture how the drone manages payload shifts and maintains stability during maneuvers. This footage provided insight into the drone's adaptability and the efficiency of its motor and propeller configurations under different load conditions.

Plan for a Typical Drone Delivery Trip in Kibera

Before each flight, medications should be packaged in medical-grade, vacuum-insulated containers with phase-change materials to keep the internal temperatures between 15°C and 25°C for up to 60 minutes. Each package should include a temperature logger to monitor internal temperatures during the flight. The medications should be sealed with shock-resistant blister packs or foil pouches to prevent damage. The total payload, including medications and insulation, should be kept between 3 and 5 kg.

Then, weather conditions should be verified to confirm clear skies, with wind speeds below 20 km/h, and no precipitation, because extreme humidity or temperature could affect drone sensors or payload stability. The flight route should be pre-programmed with three waypoints: the first being the launch site at a healthcare facility or warehouse, the second at an aerial entry point designed to avoid densely populated zones, and finally a drop-off location at an open safe zone. All no-fly zones set by the Kenya Civil Aviation Authority (KCAA) should be identified and excluded from the planned route.

Before takeoff, a full hardware check should be completed to confirm that the battery is fully charged with at least a 20% reserve buffer, and that the GPS, altimeter, compass, motors, propellers, and payload compartment are all calibrated correctly. The pre-planned flight route and emergency return protocol should then be uploaded to the drone's flight system. A live telemetry connection should be tested, and communication should be established with the receiving clinic team and local authorities.

During flight execution, the drone should be powered on, GPS lock confirmed, and the drone should ascend from the launch site to a cruising altitude of between 50 and 100 meters. It should then follow its preprogrammed flight path across open spaces, avoiding schools, markets, and residential zones. Battery level, GPS signal strength, payload temperature, and motor and internal temperatures should be monitored continuously throughout the flight. Upon reaching the drop-off area, the drone should descend, and successful delivery should be confirmed via messages from clinic staff.

On the return flight, the drone should retrace its original route or follow an alternative safe route while system checks continue to monitor for overheating or instability. After landing at the designated launch site, the drone should be powered down and retrieved. Flight logs and payload temperature data should then be downloaded and stored, and a review should be conducted to check for any irregularities such as temperature excursions or signal drops. A quality assurance review should be initiated if any issues are identified.

Test Site Selection

An ideal drone delivery testing site must meet several key criteria. It should feature rural or semi-urban characteristics to simulate remote or underserved areas, include diverse terrain for evaluating drone performance over natural obstacles, offer low air traffic and minimal regulatory restrictions, and present a range of environmental conditions to test flight stability under variable weather (Mohsan et al., 2023; Roberts et al., 2023; *Urban Adaptation*, 2022).

Gapyeong offers several advantages as a testing location for drone delivery operations. Its rural environment, with open fields and farmlands, closely resembles the less populated areas of Kibera, providing a realistic testing ground for navigation and delivery tasks. The region's mixed terrain, including hills, rivers, and valleys, allows us to evaluate drone performance over uneven and natural obstacles. Additionally, Gapyeong's controlled airspace, with fewer no-fly zones and reduced air traffic compared to urban areas like Seoul, guarantees a safer and less regulated testing environment. The close proximity to Seoul also makes it convenient to transport equipment and conduct experiments while benefiting from the resources of a major city. Finally, the weather conditions in Gapyeong, which include moderate seasonal changes, allow for testing under diverse environmental factors, such as wind and humidity, simulating the challenges faced in Kibera. These features make Gapyeong a practical and effective location for drone delivery trials.

Results

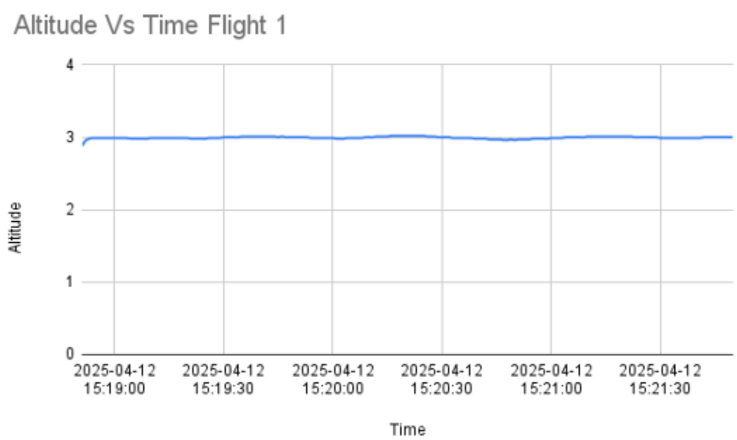

FIGURE 1 (Altitude vs. Time – Flight 1 (500g Payload)): This shows how the drone maintained a steady altitude of around 3 meters throughout the flight, proving stable hovering even with a 500g payload.

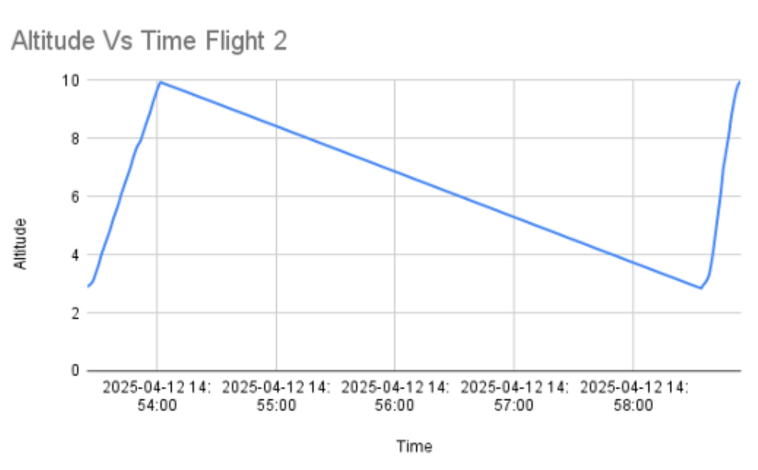

FIGURE 2 (Altitude vs. Time – Flight 2 (250g Payload)): This graph illustrates the drone's ability to autonomously adjust its altitude mid-flight, navigating a more complex path while carrying a lighter 250g payload. The results of the experiment show that the drone achieved the basic flight behaviors (vertical ascent, stable hovering, precise altitude control, and accurate landing) without human intervention, suggesting meaningful autonomous capability consistent with mid-level autonomy (Ergen et al., 2016). Even though full Level 5 autonomy requires complete independence from human input, these results show that low-cost UAVs can execute pre-defined autonomous missions. This is a viable foundation for further development towards Level 5 autonomy. These are the basic flight capabilities that can be applied to the issue of medical deliveries in the South African slum communities, where there are many factors that make it difficult to deliver medicine to, such as rough environments and narrow alleys, which are not suitable for piloted drones (Friesen et al., 2020; Lilford et al., 2017). The data demonstrates that the lower-level self-driving behaviors are executable even in situations that are modeled after the layout of slums through the use of people from the local areas (Desgroppes et al., 2011).

This finding directly responds to the central research question of whether a fully autonomous drone can navigate complex slum infrastructure and fulfill essential medical delivery tasks. The ability to operate without human guidance is impactful in underserved environments because a shortage of trained drone operators would make piloted drones impractical at scale. Since automation reduces the need for skilled labor, it would lower the operational cost per delivery, which allows the operation to continue after clinic hours. These are especially critical advantages in communities, like Kibera, where healthcare infrastructure is constrained. These results show that autonomous UAVs are a realistic and scalable option for medical deliveries in underserved environments.

The flight behavior analysis indicates that the drone performs differently in the two test cases, which are in line with the two real-world mission profiles, leading to various outcomes. In the first test case, a simple one-way delivery mission was simulated by the drone by staying at a stable altitude (see Figure 1). As an example, the delivery of medical supplies directly to the clinic was considered. On the other hand, the second flight mission represented two completely different flight phases; one, in which the drone flew up and then flew down in the middle, and finally finished the end part of the journey with a constant altitude, and the other with an alteration of the flight altitude beyond a mid-flight jump, etc. (see Figure 2). The change in altitude, up to around 10 meters, and the quick stabilization of the position are the factors suggesting that the drone can figure out the direction of the flight, in the absence of human control. This adaptability is not only important for slum area deliveries but also applies to cases where the local ground-level conditions or the environmental conditions might act as unpredictable hindrances (Roberts et al., 2023).

Roberts et al. (2023) was used because their study focused on real-world conditions encountered during drone-based emergency medical deliveries, such as variable environmental stress and multi-phase missions. Their findings on AED delivery times and mission success rates provided a real-world reference point for this study. This confirmed that testing the drone under variable conditions and environmental stress was a well-supported methodological choice backed by existing field research.

#	Test #	Date	Location	Payload Weight (kg)	Flight Duration (min)	Motor Temp Start (°C)	Motor Temp End (°C)	Max Altitude (m)	Distance Covered (km)	Avg Speed (km/h)	Weather Conditions	Notes
	1	4/12/2025	[illegible]	500g	3 min	25°C	88°C	8	120m	0.1	Windy(4m/s), rainy	
	2	4/12/2025	[illegible]	250g	7 min	25°C	47°C	10	200m	0.1	Windy(4m/s), rainy	

FIGURE 3 (Summary Table of Drone Flight Performance): This summary table highlights how payload weight affected flight time, motor temperature, and distance, confirming key performance trade-offs under real-world conditions.

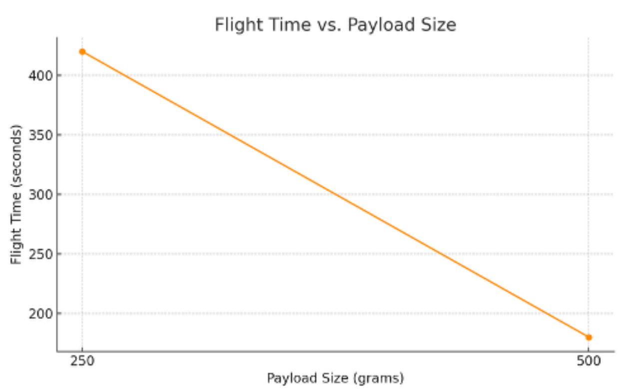

FIGURE 4 (Flight Time vs. Payload Size): This graph shows the drone's ability to carry varying payload weights, with the only concern being the decrease in flight time.

Further examination of the payload testing also pointed out important weaknesses or areas of concern. The drone with a 500g payload can fly for 3 minutes, or can fly for 7 minutes when it holds a 250 g payload, depending on the motion, and compared to 18 minutes when the drone is hovering in one position (see Figure 3/4). This result also confirms that the load weight affects the total time of the drone's flight, which is a trend that has been previously spotted in UAV performance studies (Mohsan et al., 2023; Grepow, 2024). In poor urban areas, where the health sector is far from the main population, the issue of the highest-priority, lightweight loads for maximum flight efficiency is important (Mechan et al., 2023). The difference between 3 minutes at 500g and 7 minutes at 250g suggests a roughly linear relationship between payload reduction and flight time gain, which shows that halving the payload approximately doubles the available flight time. This trade-off means that for a fixed delivery radius, drone operators would need to either reduce the quantity of medication per trip or accept a shorter operational range.

These findings support the conclusion that drone-aided delivery should prioritize temperature-sensitive and small-dose medication (like vaccines or antiretrovirals). This prioritization addresses the constraints about the trade-off between payload weight and flight time. Mohsan et al. (2023) was used for this decision because their explanation about the power-to-weight ratios in UAV design helped explain how even the smallest increase in payload weight could constrain flight duration and therefore limit real-world delivery range in underserved communities.

Moreover, it was found that there were issues regarding the functionality, which were mainly associated with the thermal management of the system and the way the drone managed different weather conditions. As the flights were ongoing, the motor temperatures were rising from 25°C initially to 47°C after the load was put on (see Appendix D). Such a drastic increase in temperature can negatively impact motor efficiency. This issue is important as Jarrah et al. (2022) state that efficient thermal control is necessary for the reliability of UAVs, especially those that are engaged in long-range missions. The rise from 25°C to 47°C represents an 88% increase in motor temperature over a relatively short flight duration, which is important because sustained operation above 50°C has risks of thermal throttling or permanent motor damage. For missions in South African slum environments where temperatures exceed 30°C, the baseline of the temperature of the motor would already be increased before takeoff, shortening the safe temperature margins. This would require either an active cooling system or upgrading to motors with higher thermal tolerance ratings.

Jarrah et al. (2022) show proof for this concern, as their study on hybrid drone systems established how high-temperature thresholds reduce UAV longevity and flight predictability. Their findings explain why thermal risks observed in this experiment pose a long-term barrier to sustainable use in health delivery missions. On the other hand, the drone lab passed moderate rain and violent winds, which helped simulate real-life slum weather conditions, as shown by Urban Adaptation (2022), and this was enough to suggest that with changes to help with motor temperatures, the missions would be more reliable and stable.

The decision to introduce wind and rain as testing variables was made after reading Urban Adaptation (2022) because their analysis of microclimate stress in informal settlements showed that there were frequent and severe amounts of wind-related disruptions in slum environments. By replicating these conditions during flight testing, the experiment was designed to show the stresses a drone would encounter during medical deliveries in South African slum communities.

From a healthcare perspective, the capability to transport even tiny payloads safely and without the requirement of constant human control is highly advantageous for the delivery of temperature-sensitive drugs such as HIV drugs. The compartment used had two parts. The first is the outside compartment where the dry ice is stored, and the second compartment holds the medication. The box stayed approximately 21°C the entire time it was in flight, which is the ideal temperature for the HIV drug (Heelon, 2023). Using dry ice as a thermal buffer in the outer layer, while isolating the medication in the inner chamber, as a dual-compartment design effectively isolated the payload from external temperature. This result provides evidence that passive cold-chain solutions are viable for short-duration UAV missions, though further testing across longer flight times and higher temperatures would be necessary before drawing broader conclusions about suitability for full deployment.

This observation demonstrates not just technical success, but public health relevance, meeting WHO medication storage standards in hostile conditions. The selection of Heelon (2023) as a source is purposeful: its temperature range specification is widely recognized in clinical pharmacy and serves as a benchmark to assess the reliability of the compartment system in drone-based cold-chain delivery.

Conclusion, Limitations, and Implications

This study was designed to see the feasibility of deploying fully autonomous drones to deliver medical supplies in South African slum areas. Using experiments that were designed to simulate real-world environmental and logistical challenges, the research demonstrated that low-cost drones can achieve the vertical mobility, stability, and payload management necessary for autonomous healthcare logistics. The findings show that, even with inexpensive hardware, drones can operate in unpredictable environments, maintain altitude and stability, withstand moderate environmental stress like wind and rain, and deliver lightweight medical payloads reliably.

In the context of South African slums, where poor road infrastructure, high population density, and environmental instability severely limit access to healthcare services (Friesen et al., 2020; Lilford et al., 2017), autonomous drones present an innovative and potentially transformative solution. By bypassing ground-based barriers, drones could reduce the time and cost required to deliver vaccines, antiretrovirals, antibiotics, and other essential medicines. Furthermore, this research provides a strong foundation for more initiatives to integrate drone-based logistics into slum healthcare systems. Overall, the project offers the vision of reliable, inexpensive healthcare delivery for some of the world’s slum communities.

Many methodological choices in this study were inspired by earlier UAV research. For example, the simulation of environmental challenges such as wind and rainfall was modeled after test frameworks in UAV emergency deployment studies, including those by Roberts et al. (2023) and Mechan et al. (2023), which emphasize the importance of environmental certainty in real-world health delivery missions. Similarly, the compartment design and temperature-stability assessment for HIV drugs drew upon guidelines found in Heelon (2023) and the cold-chain logistics challenges identified by Mohsan et al. (2023). These studies provided the theoretical and practical justifications for including thermal testing, payload adaptability, and autonomy benchmarks into the experiment.

Importantly, this drone was developed for approximately $250 USD, a cost that makes the model significantly more accessible for underfunded health systems and local governments. This affordability reinforces the potential for wide-scale replication in resource-limited contexts, where commercial drones often exceed budgetary limits. The combination of low cost, autonomous functionality, and adaptability to rugged terrain points to a promising frontier for low-barrier entry into technologically enhanced public health delivery systems.

However, even with these procedures, there are many limitations in this project. One major drawback would be the drone's battery life, which would hinder its ability to fly long distances while carrying large loads. However, even with peak motor efficiency, the energy requirements could easily overtake the ability of a battery to support it, particularly in regions where charging points are far apart. To overcome this constraint, research will be conducted on high-energy density batteries to give extra energy during flight.

This limitation is further increased by the absence of real-time energy management systems in the prototype, which could lead to power inefficiencies in multi-phase missions. A methodological refinement in future iterations could involve dynamic route planning to reduce power draw based on terrain and payload. The other constraint is the weather factors that could be the reason for the unstable flight, such as wind, temperature, and dust. Although these factors can be somewhat simulated in semi-controlled testing arenas, they may present unexpected obstacles in real-world situations. This reflects a limitation in external validity—the results from lab conditions may not fully generalize to slums with extreme climate profiles. Uncontrolled variables such as humidity, pollution, and sudden thermal shifts were not tested, and these could impact drone sensors or motor efficiency in ways not captured here.

There is also variability in the weight and size of the payload, and different medical supplies have different handling properties. This could affect the drone's stability and flight. However, to compensate for this, the payload compartments will be made to hold all sorts of medical supplies, with adjustable devices for variously shaped loads.

While promising, this solution assumes a consistent center of mass and minimal shift during flight, which may not hold true for liquid or fragile payloads. Bias may also be introduced by selecting only static or rigid loads during early trials. Future versions should incorporate vibration isolation systems or adaptive gimbal mechanisms to handle such variables.

Finally, scalability is still a challenge. While the prototype can be tested, expanding operations to a fleet of drones in underserved slums requires infrastructure, such as charging stations and maintenance support. Partnerships with local governments and NGOs will be crucial for developing the necessary infrastructure to support a scalable solution. Mobile charging stations and solar-powered charging options will be explored to enable operations in areas without reliable electricity. However, it is worth noting that implementation feasibility depends on political and socio-economic factors not addressed in this study. Local skepticism toward automated systems, regulatory limitations on UAV airspace usage, and uneven digital infrastructure could restrict usage in some regions. Additionally, constraints on interpretation include the limited number of test missions and the lack of diversity in environmental settings, which may overstate performance reliability. More vigorous field-testing across slum topographies and weather cycles is essential before drawing broader conclusions.

The implications of this research are large, both for the technical advancement of autonomous drones and for the future of healthcare delivery in underrepresented populations. Technologically, the findings suggest that low-cost autonomous drones for healthcare logistics show meaningful promise, though achieving full Level 5 autonomy remains a longer-term goal that will require advances in battery innovation, energy-efficient propulsion, lightweight materials, and real-time navigation algorithms capable of handling active urban environments (Mohsan et al., 2023; Jarrah et al., 2022).

For healthcare systems in slums, this research suggests that change is possible: from facility-based care that often remains inaccessible to decentralized, point-of-need delivery systems. Drones could serve as critical links in a distributed healthcare network, allowing immediate deployment of medicines, vaccines, and emergency supplies directly to communities that otherwise lack reliable transportation infrastructure (Quick et al., 2005). Moreover, by demonstrating the potential feasibility of small-scale, low-cost drone solutions, this study emphasizes that healthcare innovations do not necessarily require large amounts of financial investment. Partnerships between government agencies and private drone operators could create affordable, scalable drone logistics networks tailored to slum conditions. However, realizing these benefits will require diverse collaboration. Engineers must work with public health officials, policymakers, and local communities to design drone systems that are not only effective but also acceptable to the public. Regulatory frameworks must evolve to allow safe drone operations over populated areas without compromising privacy or safety standards.

In future research, pilot programs must be conducted in actual slum areas, integrating real-time environmental sensing and community feedback. Scaling strategies must also include investment in renewable energy-based charging systems, such as mobile solar charging stations, to overcome gaps in electricity supply.

References

Desgroppes, A., & Taupin, S. (2011). Kibera: The biggest slum in Africa? *Les Cahiers d'Afrique de l'Est / The East African Review, 44*, 23–33. https://doi.org/10.4000/eastafrica.521

Euchi, J. (2020). Do drones have a realistic place in a pandemic fight for delivering medical supplies in healthcare systems problems? *Chinese Journal of Aeronautics*. https://doi.org/10.1016/j.cja.2020.06.006

Friesen, J., Friesen, V., Dietrich, I., & Pelz, P. F. (2020a). Slums, space, and state of health—A link between settlement morphology and health data. *International Journal of Environmental Research and Public Health, 17*(6), 2022.https://doi.org/10.3390/ijerph17062022

Friesen, J., Taubenböck, H., Wurm, M., & Pelz, P. F. (2020b). The morphology of slums: The role of urban form in shaping the quality of life in informal settlements. *Habitat International, 98*, 102098. https://doi.org/10.1016/j.habitatint.2020.102098

Grepow. (2024). Understanding Drone Payload: A Comprehensive Guide. *Grepow*. https://www.grepow.com/blog/understanding-drone-payload-a-comprehensive-guide.html

Heelon, M. (2023). Temperature tips for safe and effective medication storage. *Baystate Health*.https://www.baystatehealth.org/articles/medication-storage-temperature-guidelines

Jarrah, K., Alali, Y., Lalko, A., & Rawashdeh, O. (2022). Flight time optimization and modeling of a hybrid gasoline–electric multirotor drone: An experimental study. *Aerospace, 9*(12). https://doi.org/10.3390/aerospace9120799

Lilford, R. J., Oyebode, O., Satterthwaite, D., Melendez-Torres, G. J., Chen, Y.-F., Mberu, B., Watson, S. I., Sartori, J., Ndugwa, R., Caiaffa, W., Haregu, T., Capon, A., Saith, R., & Ezeh, A. (2017). Improving the health and welfare of people who live in slums. *The*

Lancet, 389(10068), 559–570. https://doi.org/10.1016/S0140-6736(16)31848-7
Mechan, F., Bartonicek, Z., Malone, D., & Lees, R. S. (2023). Unmanned aerial vehicles for surveillance and control of vectors of malaria and other vector-borne diseases. *Malaria Journal, 22*(1), 23. https://doi.org/10.1186/s12936-022-04414-0
Mohsan, S. A. H., Othman, N. Q. H., Li, Y., Alsharif, M. H., & Khan, M. A. (2023). Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics, 16*(1), 109–137.https://doi.org/10.1007/s11370-022-00452-4
Mwai, P. (2022). *What is a slum?A look into Khayelitsha, one of the world's biggest slums. Habitat.org*.https://www.habitat.org/emea/stories/what-is-slum
Park, J.-E., Kibe, P., Yeboah, G., Oyebode, O., Harris, B., Ajisola, M. M., Griffiths, F., Aujla, N., Gill, P., Lilford, R. J., Chen, Y.-F., & Improving Health in Slums Collaborative. (2022). Factors associated with accessing and utilisation of healthcare and provision of health services for residents of slums in low and middle-income countries: A scoping review of recent literature. *BMJ*.https://doi.org/10.1136/bmjopen-2021-055415
Peksa, J., & Mamchur, D. (2024). A review on the state of the art in copter drones and flight control systems. *Sensors, 24*(11).https://doi.org/10.3390/s24113349
Harris, J. R., Worrell, C. M., Davis, S. M., Odero, K., Mogeni, O. D., Deming, M. S., … Addiss, D. G. (2015). *PLOS Neglected Tropical Diseases*, 9(3). https://doi.org/10.1371/journal.pntd.0003590
Prokopets, E. (2024). Guide to drone payloads: Types and use cases. *Voliro*. https://voliro.com/blog/drone-payloads/
Quick, J. D., Boohene, N.-A., Rankin, J., & Mbwasi, R. J. (2005). Medicines supply in Africa. *BMJ, 331*(7519), 709–710.https://doi.org/10.1136/bmj.331.7519.709
Roberts, N. B., Ager, E., Leith, T., Lott, I., Mason-Maready, M., Nix, T., Gottula, A., Hunt, N., & Brent, C. (2023). Current summary of the evidence in drone-based emergency medical services care. *Resuscitation Plus, 13*, 100347.https://doi.org/10.1016/j.resplu.2022.100347
Rodrigues, T. A., Patrikar, J., Choudhry, A., Feldgoise, J., Arcot, V., Gahlaut, A., Lau, S., Moon, B., Wagner, B., Matthews, H. S., Scherer, S., & Samaras, C. (2021). In-flight positional and energy use data set of a DJI Matrice 100 quadcopter for small package delivery. *Scientific Data, 8*(1), 155. https://doi.org/10.1038/s41597-021-00930-x
Rodrigues, T. A., Patrikar, J., Oliveira, N. L., Matthews, H. S., Scherer, S., & Samaras, C. (2022). Drone flight data reveal energy and

greenhouse gas emissions savings for very small package delivery. *Patterns, 3*(8), 100569.https://doi.org/10.1016/j.patter.2022.100569

Urban adaptation to climate change in South African cities. (2022). *South African Journal of Science, 118*(9/10).https://doi.org/10.17159/sajs.2022/14492

World Health Organization. (2021). Mapping and demand analysis of essential medical products and technologies for maternal, new-born, and child health and hypertension in Africa. *UN* https://www.uneca.org/sites/default/files/Africa-Investment-Summit-on-Health/mapping-and-demand-analysis-of-essential-medical-products-and-technologies-for-mnch-and-hypertension-in-africa.pdf

Yenet, A., Nibret, G., & Tegegne, B. A. (2023). Challenges to the availability and affordability of essential medicines in African countries: A scoping review. *ClinicoEconomics and Outcomes Research, 15*, 443–458. https://doi.org/10.2147/CEOR.S413546

Code (Appendix A)

1. drone_sim
1. This file acts as the backend API that connects your frontend application to the MySQL database storing drone, delivery, and organization data. When a request is made to this file (usually via GET or POST), it checks the method parameter to determine what kind of data is being requested or updated. For example, if the method is drones, it retrieves information about a specific drone using its ID. If it's dronesFromOrganization, it returns a list of all drones registered under a certain organization. There's also logic to get delivery assignments (nextDelivery) or a list of deliveries for a drone. When the API receives a POST request, it updates a drone's location, status, and other relevant data in the database. All results are returned in JSON format, making them easy to use in JavaScript on the frontend.
2. api.php
1. This file generates a simple dashboard interface where users can see all drones and their current delivery status. It starts by retrieving the organization's ID from the URL, then queries the API to get all drones associated with that organization. For each drone, it then checks if the drone has any current deliveries using the nextDelivery API method. The file displays each drone's name, coordinates (latitude and longitude), and whether or not it has a package assigned. If it does, the payload item and destination are also shown. The layout is made using basic HTML and inline PHP, organizing each drone's information into separate cards or blocks.
3. map.php
1. This file provides a real-time interactive map to track the movement of a specific drone using Leaflet.js, a popular JavaScript library for interactive maps. It starts by retrieving the drone and organization IDs from the URL,

then uses the API to fetch the drone's current coordinates and metadata. It sets up the map centered on the drone's location and displays the drone's icon using a custom marker. Then, using JavaScript's setInterval, the app continuously calls the API every second to update the drone's location. It moves the marker, draws a path using a polyline to show the route taken, and calculates live data such as speed and direction (bearing). It also shows detailed information about the payload and the delivery destination on the sidebar.

Appendix A

```
import time
import json
import requests
from dronekit import connect, VehicleMode, LocationGlobalRelative
import argparse
import sys
from pymavlink import mavutil
def truncate_float(float_number, decimal_places):
   multiplier = 10 ** decimal_places
   return int(float_number * multiplier) / multiplier
class Drone():
   def __init__(self,id):
       self.id = id
       self.destLat = 0
       self.destLon = 0
       self.droneLat = 0
       self.droneLon = 0
       self.statusId = 0

       self.vehicle = None
       self.sitl = None
       # Create the connection
       self.master = mavutil.mavlink_connection('udp:127.0.0.1:14555')
       # Wait a heartbeat before sending commands
       self.master.wait_heartbeat()
   def get_json (self, url, sign=None, timestamp=None):
       response=requests.get(url, timeout=30).content.decode('utf-8')
       #print(response)
       data = json.loads(response)
       return data

   def post_json (self, url, data):
       response=requests.post(url, timeout=30,
data=data).content.decode('utf-8')
       #print(response)
       data = json.loads(response)
       return data

   def getDroneStatus(self):
       url = "http://localhost/droneapi/api.php?method=drones&id=" +
str(self.id)
       response=self.get_json(url)
       self.statusId = int(response['drone_status_id'])

   def nextDelivery (self):
```

```
        url =
"http://localhost/droneapi/api.php?method=nextDelivery&drone_id=" +
str(self.id)
        response=self.get_json(url)
        self.destLat = float(response['lat'])
        self.destLon = float(response['lon'])

    def home (self):
        url = "http://localhost/droneapi/api.php?method=home&id=" +
str(self.id)
        response=self.get_json(url)
        self.destLat = float(response['lat'])
        self.destLon = float(response['lon'])

    def updateCoords (self):
        url = "http://localhost/droneapi/api.php?method=drones"
        data={
            "id": self.id,
            "lat": self.droneLat,
            "lon": self.droneLon,
            "drone_status_id":6
        }
        response=self.post_json(url, json.dumps(data))

    def updateStatusAtDestination (self):
        url = "http://localhost/droneapi/api.php?method=drones"
        data={
            "id": self.id,
            "lat": self.droneLat,
            "lon": self.droneLon,
            "drone_status_id":4
        }
        response=self.post_json(url, json.dumps(data))

    def updateStatusAtHome (self):
        url = "http://localhost/droneapi/api.php?method=drones"
        data={
            "id": self.id,
            "lat": self.droneLat,
            "lon": self.droneLon,
            "drone_status_id":1
        }
        response=self.post_json(url, json.dumps(data))

    def connect(self):
        # Choose a mode
        mode = 'GUIDED'

        # Check if mode is available
        if mode not in self.master.mode_mapping():
            print('Unknown mode : {}'.format(mode))
            print('Try:', list(self.master.mode_mapping().keys()))
            sys.exit(1)

        # Get mode ID
        mode_id = self.master.mode_mapping()[mode]
        self.master.mav.set_mode_send(
            self.master.target_system,
            mavutil.mavlink.MAV_MODE_FLAG_CUSTOM_MODE_ENABLED,
            mode_id)

        connection_string = 'udp:127.0.0.1:14550'
```

```
        # Start SITL if no connection string specified
        if not connection_string:
            import dronekit_sitl
            self.sitl = dronekit_sitl.start_default()
            connection_string = sitl.connection_string()

        # Connect to the Vehicle
        print('Connecting to vehicle on: %s' % connection_string)
        self.vehicle = connect(connection_string, wait_ready=True)


    def arm_and_takeoff(self, aTargetAltitude):
        """
        Arms vehicle and fly to aTargetAltitude.
        """

        print("Basic pre-arm checks")
        # Don't try to arm until autopilot is ready
        while not self.vehicle.is_armable:
            print(" Waiting for vehicle to initialise...")
            time.sleep(1)

        print("Arming motors")
        # Copter should arm in GUIDED mode
        self.vehicle.mode = VehicleMode("GUIDED")
        self.vehicle.armed = True

        # Confirm vehicle armed before attempting to take off
        while not self.vehicle.armed:
            print(" Waiting for arming...")
            time.sleep(1)

        print("Taking off!")
        self.vehicle.simple_takeoff(aTargetAltitude)  # Take off to target
altitude

        # Wait until the vehicle reaches a safe height before processing
the goto
        #  (otherwise the command after Vehicle.simple_takeoff will execute
        #   immediately).
        while True:
            print(" Altitude: ",
self.vehicle.location.global_relative_frame.alt)
            # Break and return from function just below target altitude.
            if self.vehicle.location.global_relative_frame.alt >=
aTargetAltitude * 0.95:
                print("Reached target altitude")
                break
            time.sleep(1)
    def flyToDest(self):
        print("Set default/target airspeed to 10")
        self.vehicle.airspeed = 100

        print("Going towards destination")
        point1 = LocationGlobalRelative(self.destLat, self.destLon, 10)
        self.vehicle.simple_goto(point1)

        # sleep so we can see the change in map
        while True:
            # Printing Vehicle's Latitude
```

```
            print("Vehicle's Latitude =  ",
self.vehicle.location.global_relative_frame.lat)

            # Printing Vehicle's Longitude
            print("Vehicle's Longitude =  ",
self.vehicle.location.global_relative_frame.lon)

            self.droneLat = self.vehicle.location.global_relative_frame.lat
            self.droneLon = self.vehicle.location.global_relative_frame.lon
            self.updateCoords()
            time.sleep(1)

            # Absolute value to not overshoot landing zone

if(truncate_float(abs(self.vehicle.location.global_relative_frame.lat), 4)
== truncate_float(abs(self.destLat),4) and
truncate_float(abs(self.vehicle.location.global_relative_frame.lon), 4) ==
truncate_float(abs(self.destLon), 4)):
                break

   def land(self):

       print("Landing")

       # Choose a mode
       mode = 'LAND'
       # Check if mode is available
       if mode not in self.master.mode_mapping():
           print('Unknown mode : {}'.format(mode))
           print('Try:', list(self.master.mode_mapping().keys()))
           sys.exit(1)

       # Get mode ID
       mode_id = self.master.mode_mapping()[mode]
       self.master.mav.set_mode_send(
           self.master.target_system,
           mavutil.mavlink.MAV_MODE_FLAG_CUSTOM_MODE_ENABLED,
           mode_id)

       while True:
           print(" Altitude: ",
self.vehicle.location.global_relative_frame.alt)
           # Break and return from function just below target altitude.
           if self.vehicle.location.global_relative_frame.alt <= 0.05:
               print("Landed")
               break
           time.sleep(1)

drone1 = Drone(1)
drone1.connect()
drone1.getDroneStatus()
if drone1.statusId==1:
   print("Is available but needs to be loaded")

elif drone1.statusId==2:
   print("Is being loaded but not yet loaded loaded")
elif drone1.statusId==3:
   drone1.nextDelivery()
   drone1.arm_and_takeoff(10)
   drone1.flyToDest()
   drone1.land()
   drone1.updateStatusAtDestination()
```

```
elif drone1.statusId==4:
   print("Need to be offloaded first")
elif drone1.statusId==5:
   drone1.home()
   drone1.arm_and_takeoff(10)
   drone1.flyToDest()
   drone1.land()
   drone1.updateStatusAtHome()

elif drone1.statusId==6:
   print("I thought I was already delivering but check again")

elif drone1.statusId==7:
   print("I am on vacation please leave me be")


drone1.vehicle.close()
drone1.master=None
# Shut down simulator if it was started.
if drone1.sitl:
   drone1.sitl.stop()
```

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
header("Access-Control-Allow-Origin: *");
header("Content-Type:application/json");
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
       if (isset($_GET['method']) && $_GET['method']=="deliveries") {
              if (isset($_GET['id']) && $_GET['id']!="") {
                     include('db.php');
                     $id = $_GET['id'];
                     $result = mysqli_query($con,"SELECT * FROM
`deliveries` WHERE id=$id");
                     if(mysqli_num_rows($result)>0){
                            $row = mysqli_fetch_assoc($result);
                            echo json_encode($row);
                     }
                     else{
                            $response["code"]=200;
                            $response["status"]="No Record Found";
                            $json_response = json_encode($response);
                            echo $json_response;
                     }
                     mysqli_close($con);
              }
              else{
                     $response["code"]=400;
                     $response["status"]="Invalid Request";
                     $json_response = json_encode($response);
                     echo $json_response;
              }
       }
       //pending delivery is id 1
       if (isset($_GET['method']) && $_GET['method']=="nextDelivery") {
              if (isset($_GET['drone_id']) && $_GET['drone_id']!="") {
                     include('db.php');
                     $droneId = $_GET['drone_id'];
```

```
                    $sql="SELECT organizations.lat, organizations.lon
FROM `deliveries` JOIN `organizations` ON
deliveries.organization_id=organizations.id WHERE
deliveries.drone_id=$droneId AND deliveries.delivery_status_id=1";
                    //echo $sql;
                    $result = mysqli_query($con, $sql);
                    if(mysqli_num_rows($result)>0){
                          $row = mysqli_fetch_assoc($result);
                          echo json_encode($row);
                    }
                    else{
                          $response["code"]=200;
                          $response["status"]="No Record Found";
                          $json_response = json_encode($response);
                          echo $json_response;
                    }
                    mysqli_close($con);
             }
             else{
                    $response["code"]=400;
                    $response["status"]="Invalid Request";
                    $json_response = json_encode($response);
                    echo $json_response;
             }
      }

      if (isset($_GET['method']) && $_GET['method']=="drones") {
             if (isset($_GET['id']) && $_GET['id']!="") {
                    include('db.php');
                    $id = $_GET['id'];
                    $result = mysqli_query($con,"SELECT * FROM `drones`
WHERE id=$id");
                    if(mysqli_num_rows($result)>0){
                          $row = mysqli_fetch_assoc($result);
                          echo json_encode($row);
                    }
                    else{
                          $response["code"]=200;
                          $response["status"]="No Record Found";
                          $json_response = json_encode($response);
                          echo $json_response;
                    }
                    mysqli_close($con);
             }
             else{
                    $response["code"]=400;
                    $response["status"]="Invalid Request";
                    $json_response = json_encode($response);
                    echo $json_response;
             }
      }

      if (isset($_GET['method']) && $_GET['method']=="home") {
             if (isset($_GET['id']) && $_GET['id']!="") {
                    include('db.php');
                    $id = $_GET['id'];
                    $result = mysqli_query($con,"SELECT
organizations.lat, organizations.lon FROM `drones` JOIN `organizations` ON
drones.organization_id=organizations.id WHERE drones.id=$id");
                    if(mysqli_num_rows($result)>0){
                          $row = mysqli_fetch_assoc($result);
                          echo json_encode($row);
```

```
                }
                else{
                        $response["code"]=200;
                        $response["status"]="No Record Found";
                        $json_response = json_encode($response);
                        echo $json_response;
                }
                mysqli_close($con);
        }
        else{
                $response["code"]=400;
                $response["status"]="Invalid Request";
                $json_response = json_encode($response);
                echo $json_response;
        }
    }
    //get list of drones for organization
    if (isset($_GET['method']) &&
$_GET['method']=="dronesFromOrganization") {
        if (isset($_GET['id']) && $_GET['id']!="") {
                include('db.php');
                $id = $_GET['id'];
                $result = mysqli_query($con,"select drones.*,
items.name as item_name, items.description, organizations.name as
organization_name, organizations.address from drones left join deliveries
on drones.id=deliveries.drone_id and deliveries.delivery_status_id=4 left
join items on deliveries.item_id=items.id left join organizations on
deliveries.organization_id=organizations.id where
drones.organization_id=$id");
                $rows = array();
                if(mysqli_num_rows($result)>0){
                        while($row = $result->fetch_assoc()){
                                $rows[] = $row;
                        }
                        echo json_encode($rows);
                }
                else{
                        $response["code"]=200;
                        $response["status"]="No Record Found";
                        $json_response = json_encode($response);
                        echo $json_response;
                }
                mysqli_close($con);
        }
        else{
                $response["code"]=400;
                $response["status"]="Invalid Request";
                $json_response = json_encode($response);
                echo $json_response;
        }
    }
}
else if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $json = file_get_contents('php://input');
    $values = json_decode($json, true);

    //var_dump($values);
    if (isset($_GET['method']) && $_GET['method']=="drones") {
        if (isset($values['id']) && $values['id']!="") {
                include('db.php');
                $id = $values['id'];
```

```
                        $lat = $values['lat'];
                        $lon = $values['lon'];
                        $droneStatusId = $values['drone_status_id'];
                        if (mysqli_query($con,"UPDATE `drones` SET lat=$lat,
lon=$lon, drone_status_id=$droneStatusId WHERE id=$id") === TRUE) {
                                $response["code"]=200;
                                $response["status"]="Update Successful";
                                $json_response = json_encode($response);
                                echo $json_response;
                        }
                        else{
                                $response["code"]=500;
                                $response["status"]="Error Updating Record";
                                $json_response = json_encode($response);
                                echo $json_response;
                        }
                }
                else{
                        $response["code"]=400;
                        $response["status"]="Invalid Request";
                        $json_response = json_encode($response);
                        echo $json_response;
                }
        }
}
else if ($_SERVER['REQUEST_METHOD'] === 'DELETE') {
}
function errorResponse($code, $error){
        $response['code'] = $code;
        $response['error'] = $error;
        $json_response = json_encode($response);
        echo $json_response;
}
?>

<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
$droneId=$_GET['drone_id'];
$organizationId=$_GET['organization_id'];
?>
<!DOCTYPE html>
<html lang="en">
<head>
        <base target="_top">
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">

        <title>Drone location</title>

        <link rel="shortcut icon" type="image/x-icon"
href="docs/images/favicon.ico" />
    <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" integrity="sha256-
p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY=" crossorigin=""/>
    <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo="
crossorigin=""></script>
        <script src="https://code.jquery.com/jquery-3.6.3.min.js"></script>

        <style>
```

```
                html, body {
                        height: 100%;
                        margin: 0;
                }
                .map {
                        position:absolute;
                        z-index:0;
                }
                .heading {
                       text-align: center;
                       position:absolute;
                        left: 50%;
                        font-size: 150%;
                        transform: translate(-50%, 0%);
                        top: 2.5%;
                        background-color: #F5F5F5;

                        border-radius: 10px;
                        background: #F5F5F5;
                        padding: 10px;
                        border: 1px solid #D6D5D5;
                        box-shadow: 0.15em 5px 5px #D6D5D5;
                        z-index:1;
                }

                .details{
                       text-align: left;
                       position:absolute;
                       top: 2.5%;
                        right: 2.5%;
                        background-color: #F5F5F5;

                        border-radius: 10px;
                        background: #F5F5F5;
                        padding: 10px;
                        border: 1px solid #D6D5D5;
                        box-shadow: 0.15em 5px 5px #D6D5D5;
                        z-index:1;
                }
                .dronemenu{
                        text-align: left;
                       position:absolute;
                       top: 11.0%;
                        left: .5%;
                        background-color: #F5F5F5;

                        border-radius: 10px;
                        background: #F5F5F5;
                        padding: 10px;
                        border: 1px solid #D6D5D5;
                        box-shadow: 0.15em 5px 5px #D6D5D5;
                        z-index:1;
                }

       </style>
</head>
<body>
<div class=dronemenu ><b>Drones</b><br /><br />
<?php
$response =
file_get_contents('http://localhost/droneapi/api.php?method=dronesFromOrga
nization&id='.$organizationId);
```

```
$response = json_decode($response);
$itemName=null;
$itemDescription=null;
$organizationName=null;
$organizationAddress=null;
foreach($response as $item){
        if($item->id==$droneId){
                $itemName=$item->item_name;
                $itemDescription=$item->description;
                $organizationName=$item->organization_name;
                $organizationAddress=$item->address;
        }
        echo('<a
href="http://localhost/map/map.php?organization_id='.$organizationId.'&dro
ne_id='.$item->id.'">'.$item->name.'</a><br />');
}
//can't display null so substituded this instead
if($itemName==null){
        $itemName='None';
        $itemDescription='None';
}
?>
</div>
<div class=details ><b>Details</b><br /><br />
<div id="details"></div></div>
<div id="map" class=map style="width: 100%; height: 100%;"></div>
<div class=heading>Drone <?php echo($droneId)?> map</div>
<script>
        function measure(lat1, lon1, lat2, lon2){  // generally used geo
measurement function
            var R = 6378.137; // Radius of earth in KM
            var dLat = lat2 * Math.PI / 180 - lat1 * Math.PI / 180;
            var dLon = lon2 * Math.PI / 180 - lon1 * Math.PI / 180;
            var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
            Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 * Math.PI / 180)
*
            Math.sin(dLon/2) * Math.sin(dLon/2);
            var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
            var d = R * c;
            return d * 1000; // meters
        }
        function bearing(lat1, lon1, lat2, lon2){
              var y = Math.sin(lon2-lon1) * Math.cos(lat2);
              var x = Math.cos(lat1)*Math.sin(lat2) -
                      Math.sin(lat1)*Math.cos(lat2)*Math.cos(lon2-lon1);
              return Math.atan2(y, x) * 180 / Math.PI;
        }
        lat=0;
        lng=0;
        prevLat=0;
        prevLng=0;
        droneID=<?php echo($droneId)?>;
        itemName="<?php echo($itemName)?>";
        itemDescription="<?php echo($itemDescription)?>";
        organizationName="<?php echo($organizationName)?>";
        organizationAddress="<?php echo($organizationAddress)?>";
        var req =
$.getJSON('http://localhost/droneapi/api.php?method=drones&id='+droneID,
function(json) {
              var data = JSON.parse(JSON.stringify(json));
              lat=parseFloat(data.lat);
              lng=parseFloat(data.lon);
```

```
        });

        req.done(function(response){
               var route=[lat, lng];

               const map = L.map('map').setView([lat, lng], 15);

               const tiles =
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
                       maxZoom: 19,
                       attribution: '&copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
               }).addTo(map);

               var droneIcon = L.icon({
                   iconUrl: 'drone2.png',
                   iconSize: [35, 35],
                   popupAnchor: [0.5, -11]
                   });

               const marker = L.marker([lat, lng],{icon:
droneIcon}).addTo(map)
                       .bindPopup('<b>Drone '+droneID+'</b><br
/>').openPopup();

               var polyline = new L.Polyline([[lat, lng]], {
                   color: 'red',
                   weight: 3,
                   opacity: 0.5,
                   smoothFactor: 1
               });
               polyline.addTo(map);

               setInterval(function () {

$.getJSON('http://localhost/droneapi/api.php?method=drones&id='+droneID,
function(json) {
                              var data = JSON.parse(JSON.stringify(json));
                              lat=parseFloat(data.lat);
                              lng=parseFloat(data.lon);
                              speed=measure(prevLat, prevLng, lat, lng);
                              degrees=bearing(prevLat, prevLng, lat, lng);

                              if(prevLat!=0 && prevLng!=0){
                                     payloadStr='';
                                     if(itemName!='None'){
                                            payloadStr="<br /><br
/>Payload: "+itemName+"<br />Description: "+
                                                   itemDescription+"<br
/><br />Destination: "+organizationName+",<br />"+organizationAddress;
                                     }
                                     $('#details').html("Latitude:
"+lat+"<br />Longitude: "+lng+"<br />Speed: "+speed.toFixed(2)+
                                            "m/s"+"<br />Bearing:
"+degrees+payloadStr);
                              }
                              prevLat=lat;
                              prevLng=lng;
                              //alert(lat);

                       });
                       //alert(lat);
```

```
                      if (route!=[lat, lng]){
                          marker.setLatLng([lat, lng]);
                          polyline.addLatLng([lat, lng]);
                      }
              }, 1000);
       });
</script>
</body>
</html>
```

Appendix B

DateTime	Latitude	Longitude	Altitude
2025-04-12 15:18:51	37.9057117	127.5616893	2.88
2025-04-12 15:18:52	37.9057116	127.5616871	2.95
2025-04-12 15:18:53	37.9057124	127.56168	2.98

2025-04-12 15:18:54	37.9057127	127.5616723	2.99
2025-04-12 15:18:55	37.9057128	127.5616641	2.99
2025-04-12 15:18:56	37.9057131	127.5616532	2.99
2025-04-12 15:18:57	37.9057136	127.5616436	2.99
2025-04-12 15:18:58	37.9057141	127.5616335	2.99
2025-04-12 15:18:59	37.9057146	127.5616228	2.99
2025-04-12 15:19:00	37.9057147	127.561612	2.99
2025-04-12 15:19:01	37.9057148	127.5616013	2.99
2025-04-12 15:19:02	37.9057149	127.5615879	2.99
2025-04-12 15:19:03	37.905715	127.5615777	2.99
2025-04-12 15:19:05	37.9057149	127.5615678	2.98
2025-04-12 15:19:06	37.9057149	127.5615558	2.98
2025-04-12 15:19:07	37.9057149	127.5615465	2.98

2025-04-12 15:19:08	37.905715	127.5615372	2.98
2025-04-12 15:19:09	37.905715	127.5615281	2.98
2025-04-12 15:19:10	37.9057153	127.5615192	2.99
2025-04-12 15:19:11	37.9057156	127.5615102	2.99
2025-04-12 15:19:12	37.905716	127.5614987	2.99
2025-04-12 15:19:13	37.9057163	127.5614897	2.99
2025-04-12 15:19:14	37.9057165	127.5614807	2.99
2025-04-12 15:19:15	37.9057168	127.5614717	2.99
2025-04-12 15:19:16	37.9057171	127.5614602	2.99
2025-04-12 15:19:17	37.9057174	127.5614507	2.99
2025-04-12 15:19:18	37.9057178	127.5614412	2.99
2025-04-12 15:19:19	37.9057181	127.5614312	2.99
2025-04-12 15:19:20	37.9057183	127.5614213	2.99

2025-04-12 15:19:21	37.9057187	127.5614114	2.98
2025-04-12 15:19:22	37.9057189	127.5613988	2.98
2025-04-12 15:19:23	37.9057191	127.561389	2.98
2025-04-12 15:19:24	37.9057194	127.5613792	2.98
2025-04-12 15:19:25	37.9057197	127.5613667	2.98
2025-04-12 15:19:26	37.9057199	127.5613568	2.99
2025-04-12 15:19:27	37.9057203	127.5613466	2.99
2025-04-12 15:19:28	37.9057206	127.5613362	2.99
2025-04-12 15:19:29	37.9057209	127.5613257	2.99
2025-04-12 15:19:30	37.9057212	127.5613154	3
2025-04-12 15:19:31	37.9057214	127.5613022	3
2025-04-12 15:19:32	37.9057217	127.5612917	3
2025-04-12 15:19:33	37.9057218	127.5612813	3

2025-04-12 15:19:34	37.9057217	127.5612709	3
2025-04-12 15:19:35	37.9057217	127.5612584	3.01
2025-04-12 15:19:36	37.9057219	127.5612486	3.01
2025-04-12 15:19:37	37.9057221	127.5612387	3.01
2025-04-12 15:19:38	37.9057224	127.5612287	3.01
2025-04-12 15:19:39	37.9057228	127.5612188	3.01
2025-04-12 15:19:40	37.9057232	127.5612062	3.01
2025-04-12 15:19:41	37.9057236	127.561196	3.01
2025-04-12 15:19:42	37.905724	127.5611857	3.01
2025-04-12 15:19:43	37.9057243	127.5611751	3.01
2025-04-12 15:19:44	37.9057245	127.5611617	3.01
2025-04-12 15:19:45	37.9057248	127.5611511	3
2025-04-12 15:19:46	37.9057251	127.5611403	3.01

2025-04-12 15:19:47	37.9057254	127.5611294	3
2025-04-12 15:19:48	37.9057256	127.5611186	3
2025-04-12 15:19:49	37.9057258	127.5611052	3
2025-04-12 15:19:50	37.9057259	127.5610948	3
2025-04-12 15:19:51	37.9057261	127.5610847	3
2025-04-12 15:19:52	37.9057264	127.5610746	3
2025-04-12 15:19:53	37.9057266	127.5610645	3
2025-04-12 15:19:54	37.905727	127.5610519	2.99
2025-04-12 15:19:55	37.9057273	127.5610415	2.99
2025-04-12 15:19:56	37.9057276	127.5610311	2.99
2025-04-12 15:19:57	37.9057279	127.561018	2.99
2025-04-12 15:19:58	37.9057281	127.5610076	2.99
2025-04-12 15:19:59	37.9057284	127.5609973	2.99

2025-04-12 15:20:00	37.9057286	127.5609869	2.99
2025-04-12 15:20:01	37.905729	127.5609763	2.98
2025-04-12 15:20:02	37.9057292	127.5609656	2.98
2025-04-12 15:20:03	37.9057296	127.5609522	2.98
2025-04-12 15:20:04	37.9057298	127.5609417	2.99
2025-04-12 15:20:05	37.90573	127.5609313	2.99
2025-04-12 15:20:06	37.9057302	127.5609209	2.99
2025-04-12 15:20:07	37.9057305	127.560908	2.99
2025-04-12 15:20:08	37.9057309	127.5608976	2.99
2025-04-12 15:20:09	37.905731	127.5608872	3
2025-04-12 15:20:10	37.9057311	127.5608769	3
2025-04-12 15:20:11	37.9057313	127.560867	3
2025-04-12 15:20:12	37.9057316	127.560857	3.01

2025-04-12 15:20:13	37.9057319	127.5608445	3.01
2025-04-12 15:20:14	37.9057322	127.5608345	3.01
2025-04-12 15:20:15	37.9057326	127.5608244	3.01
2025-04-12 15:20:16	37.9057329	127.5608118	3.02
2025-04-12 15:20:17	37.9057332	127.5608017	3.02
2025-04-12 15:20:18	37.9057335	127.5607915	3.02
2025-04-12 15:20:19	37.9057336	127.5607813	3.02
2025-04-12 15:20:20	37.9057339	127.5607712	3.02
2025-04-12 15:20:21	37.905734	127.5607612	3.02
2025-04-12 15:20:22	37.9057343	127.560749	3.02
2025-04-12 15:20:23	37.9057345	127.5607393	3.02
2025-04-12 15:20:24	37.9057347	127.5607299	3.02
2025-04-12 15:20:25	37.9057349	127.5607203	3.02

2025-04-12 15:20:26	37.9057354	127.560708	3.01
2025-04-12 15:20:27	37.9057357	127.5606979	3.01
2025-04-12 15:20:28	37.9057359	127.5606879	3.01
2025-04-12 15:20:29	37.905736	127.5606779	3
2025-04-12 15:20:30	37.9057363	127.5606682	3
2025-04-12 15:20:31	37.9057365	127.5606585	3
2025-04-12 15:20:32	37.9057369	127.5606466	3
2025-04-12 15:20:33	37.9057373	127.5606371	2.99
2025-04-12 15:20:35	37.9057375	127.5606274	2.99
2025-04-12 15:20:36	37.9057377	127.5606152	2.99
2025-04-12 15:20:37	37.905738	127.5606056	2.99
2025-04-12 15:20:38	37.9057381	127.560596	2.99
2025-04-12 15:20:39	37.9057382	127.5605866	2.98

2025-04-12 15:20:40	37.9057385	127.5605774	2.98
2025-04-12 15:20:41	37.9057388	127.5605681	2.98
2025-04-12 15:20:42	37.905739	127.5605564	2.98
2025-04-12 15:20:43	37.9057393	127.5605472	2.97
2025-04-12 15:20:44	37.9057395	127.5605379	2.97
2025-04-12 15:20:45	37.9057397	127.5605285	2.97
2025-04-12 15:20:46	37.9057401	127.5605168	2.97
2025-04-12 15:20:47	37.9057403	127.5605073	2.96
2025-04-12 15:20:48	37.9057404	127.5604978	2.96
2025-04-12 15:20:49	37.9057404	127.5604883	2.97
2025-04-12 15:20:50	37.9057406	127.5604791	2.96
2025-04-12 15:20:51	37.9057409	127.5604699	2.97
2025-04-12 15:20:52	37.9057414	127.5604583	2.97

2025-04-12 15:20:53	37.9057418	127.560449	2.97
2025-04-12 15:20:54	37.9057421	127.5604397	2.97
2025-04-12 15:20:55	37.9057424	127.5604283	2.98
2025-04-12 15:20:56	37.9057426	127.5604192	2.98
2025-04-12 15:20:57	37.9057428	127.56041	2.98
2025-04-12 15:20:58	37.9057429	127.5604007	2.98
2025-04-12 15:20:59	37.905743	127.5603916	2.98
2025-04-12 15:21:00	37.9057431	127.5603826	2.99
2025-04-12 15:21:01	37.9057433	127.5603717	2.99
2025-04-12 15:21:02	37.9057434	127.5603629	2.99
2025-04-12 15:21:03	37.9057437	127.5603542	2.99
2025-04-12 15:21:04	37.9057441	127.5603452	3
2025-04-12 15:21:05	37.9057444	127.5603337	3

2025-04-12 15:21:06	37.9057446	127.5603246	3
2025-04-12 15:21:07	37.9057449	127.5603155	3
2025-04-12 15:21:08	37.9057452	127.5603063	3
2025-04-12 15:21:09	37.9057454	127.5602972	3
2025-04-12 15:21:10	37.9057457	127.5602861	3.01
2025-04-12 15:21:11	37.905746	127.5602771	3.01
2025-04-12 15:21:12	37.9057461	127.5602682	3.01
2025-04-12 15:21:13	37.9057463	127.5602594	3.01
2025-04-12 15:21:14	37.9057466	127.5602483	3.01
2025-04-12 15:21:15	37.9057468	127.5602391	3.01
2025-04-12 15:21:16	37.9057469	127.5602301	3.01
2025-04-12 15:21:17	37.9057471	127.5602213	3.01
2025-04-12 15:21:18	37.9057474	127.5602099	3.01

2025-04-12 15:21:19	37.9057476	127.5602006	3.01
2025-04-12 15:21:20	37.9057477	127.5601916	3.01
2025-04-12 15:21:21	37.9057479	127.5601828	3.01
2025-04-12 15:21:22	37.9057482	127.560174	3.01
2025-04-12 15:21:23	37.9057484	127.5601652	3
2025-04-12 15:21:24	37.9057487	127.5601539	3
2025-04-12 15:21:25	37.905749	127.5601448	3
2025-04-12 15:21:26	37.9057494	127.5601353	3
2025-04-12 15:21:27	37.9057498	127.5601233	3
2025-04-12 15:21:28	37.9057499	127.5601139	3
2025-04-12 15:21:29	37.9057502	127.5601045	3
2025-04-12 15:21:30	37.9057503	127.5600952	2.99
2025-04-12 15:21:31	37.9057504	127.560086	2.99

2025-04-12 15:21:32	37.9057505	127.5600766	2.99
2025-04-12 15:21:33	37.9057508	127.5600652	2.99
2025-04-12 15:21:34	37.9057509	127.560056	2.99
2025-04-12 15:21:35	37.905751	127.5600469	2.99
2025-04-12 15:21:36	37.9057511	127.5600378	2.99
2025-04-12 15:21:37	37.9057514	127.5600268	2.99
2025-04-12 15:21:38	37.9057518	127.560018	2.99
2025-04-12 15:21:39	37.9057521	127.560009	2.99
2025-04-12 15:21:40	37.9057525	127.5599998	2.99
2025-04-12 15:21:41	37.9057529	127.5599904	2.99
2025-04-12 15:21:42	37.9057532	127.5599809	3
2025-04-12 15:21:43	37.9057535	127.5599691	3
2025-04-12 15:21:44	37.9057537	127.5599597	3

2025-04-12 15:21:45	**37.9057538**	**127.55995**	**3**
2025-04-12 15:21:46	**37.9057539**	**127.5599382**	**3**
2025-04-12 15:21:47	**37.9057541**	**127.5599288**	**3**
2025-04-12 15:21:48	**37.9057542**	**127.5599195**	**3**
2025-04-12 15:21:49	**37.9057544**	**127.5599103**	**3**
2025-04-12 15:21:50	**37.9057545**	**127.5599011**	**3**

Appendix C

DateTime	Latitude	Longitude	Altitude
2025-04-12 14:53:25	37.9057121	127.5616895	2.89
2025-04-12 14:53:26	37.905712	127.561688	2.95
2025-04-12 14:53:27	37.9057128	127.5616778	3.01
2025-04-12 14:53:28	37.9057137	127.561659	3.11
2025-04-12 14:53:29	37.9057132	127.5616237	3.3
2025-04-12 14:53:30	37.9057133	127.5615889	3.5
2025-04-12 14:53:31	37.905714	127.56155	3.7

2025-04-12 14:53:32	37.905715	127.5614958	3.96
2025-04-12 14:53:33	37.9057163	127.5614491	4.17
2025-04-12 14:53:34	37.9057176	127.5614006	4.37
2025-04-12 14:53:35	37.9057188	127.5613512	4.56
2025-04-12 14:53:36	37.9057197	127.5613008	4.75
2025-04-12 14:53:37	37.9057206	127.5612497	4.95
2025-04-12 14:53:38	37.9057216	127.5611856	5.2
2025-04-12 14:53:39	37.9057223	127.5611343	5.4
2025-04-12 14:53:40	37.9057232	127.561083	5.6
2025-04-12 14:53:41	37.9057243	127.5610316	5.8
2025-04-12 14:53:42	37.905726	127.5609669	6.05
2025-04-12 14:53:43	37.9057271	127.5609152	6.24
2025-04-12 14:53:44	37.9057284	127.5608633	6.45
2025-04-12 14:53:45	37.9057299	127.5608113	6.65
2025-04-12 14:53:46	37.9057313	127.5607594	6.85
2025-04-12 14:53:47	37.9057327	127.5607075	7.05
2025-04-12 14:53:48	37.9057344	127.5606424	7.3
2025-04-12 14:53:49	37.9057358	127.5605902	7.5
2025-04-12 14:53:50	37.9057371	127.5605378	7.7
2025-04-12 14:53:52	37.9057387	127.5604718	7.95
2025-04-12 14:53:53	37.9057401	127.5604191	8.16
2025-04-12 14:53:54	37.9057416	127.5603663	8.36
2025-04-12 14:53:55	37.905743	127.5603133	8.57
2025-04-12 14:53:56	37.9057442	127.5602605	8.77
2025-04-12 14:53:57	37.9057455	127.5602079	8.98
2025-04-12 14:53:58	37.9057472	127.5601421	9.23
2025-04-12 14:53:59	37.9057484	127.5600894	9.43
2025-04-12 14:54:00	37.9057497	127.5600366	9.63
2025-04-12 14:54:01	37.905751	127.5599856	9.81
2025-04-12 14:54:02	37.9057525	127.55993	9.94

2025-04-12 14:58:34	37.9057512	127.5598696	2.85
2025-04-12 14:58:35	37.9057518	127.5598702	2.96
2025-04-12 14:58:36	37.9057556	127.5598799	3.04
2025-04-12 14:58:37	37.9057615	127.5598952	3.15
2025-04-12 14:58:38	37.9057705	127.5599179	3.33
2025-04-12 14:58:39	37.9057878	127.5599569	3.69
2025-04-12 14:58:40	37.9058065	127.5599973	4.1
2025-04-12 14:58:41	37.9058276	127.5600467	4.61
2025-04-12 14:58:42	37.9058513	127.5601026	5.14
2025-04-12 14:58:43	37.9058771	127.5601642	5.69
2025-04-12 14:58:44	37.9059048	127.560231	6.24
2025-04-12 14:58:45	37.9059419	127.5603197	6.92
2025-04-12 14:58:46	37.9059715	127.5603904	7.33
2025-04-12 14:58:47	37.9060018	127.5604631	7.74
2025-04-12 14:58:48	37.9060325	127.5605373	8.16
2025-04-12 14:58:49	37.9060715	127.5606306	8.7
2025-04-12 14:58:50	37.9061025	127.5607056	9.11
2025-04-12 14:58:51	37.9061328	127.5607788	9.47
2025-04-12 14:58:52	37.9061615	127.5608482	9.74
2025-04-12 14:58:53	37.9061884	127.560912	9.9

2025-04-12 14:58:54	37.9062122	127.5609686	9.97

Appendix D

Test #	Date	Location	Payload Weight (kg)	Flight Duration (min)	Motor Temp Start (°C)	Motor Temp End (°C)	Max Altitude (m)	Distance Covered (km)	Avg Speed (km/h)	Weather Conditions
1	4/12/2025	가평뷰 캠핑장	500g	3 min	25°C	38°C	3	125m	0.1	Windy(4m/s), rainy
2	4/12/2025	가평뷰 캠핑장	500g	7 min	25°C	47°C	10	200m	0.1	Windy(4m/s), rainy