N-3.5 ROCKET LAUNCH (2024)

RECOVERY TEAM: TECHNICAL REPORT

TEAM MEMBERS:

Chesang Naomi, Graig Kipkorir, Amos Korir,

Jerry Mumo, Esther Mamai, Brian Munene, Ian Kiptoo

INTRODUCTION

This report details the operations of the Nakuja Recovery Team in preparation for the N-3.5 rocket launch. It also details the post-launch analysis, including avionics performance and recovery. The recovery team successfully prepared for and executed the recovery operations following the N-3.5 rocket launch on 8th May, 2024. Key findings include the condition of the recovered avionics bay, flight computer performance, data retrieved, and any issues encountered during the process.

In any rocketry project, ensuring the safe recovery of the payload and the rocket itself is paramount. The recovery team plays a crucial role in achieving this objective, consisting of three specialised subteams: Flight Computer Team, Telemetry Team, and Ejection Team. Each subteam is tasked with unique responsibilities vital to the successful retrieval of the rocket.

I. **Flight Computer Team** - They're responsible for designing, programming, and maintaining the onboard flight computer. They ensure proper design and implementation of flight computer PCB and integration with flight software.

II. **Telemetry Team** - They establish communication with the rocket, transmitting real-time data for ground control. They design telemetry systems, select communication protocols, and analyse data for monitoring and recovery guidance.

III. **Ejection Team** - They design and test ejection mechanisms for deploying recovery systems at predetermined altitudes or conditions. They also integrate the ejection system with the flight computer, and conduct tests to ensure reliability and safety.

TABLE OF CONTENTS

PRE-LAUNCH PREPARATIONS

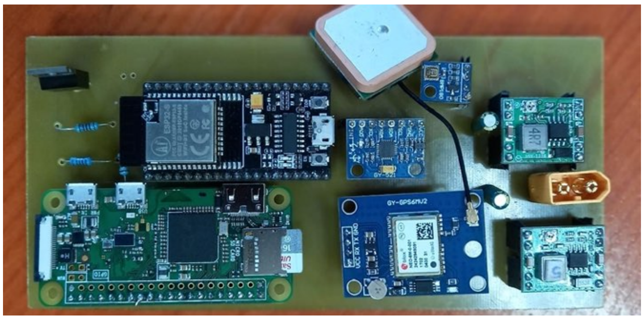
1. FLIGHT COMPUTER DESIGN AND FABRICATION

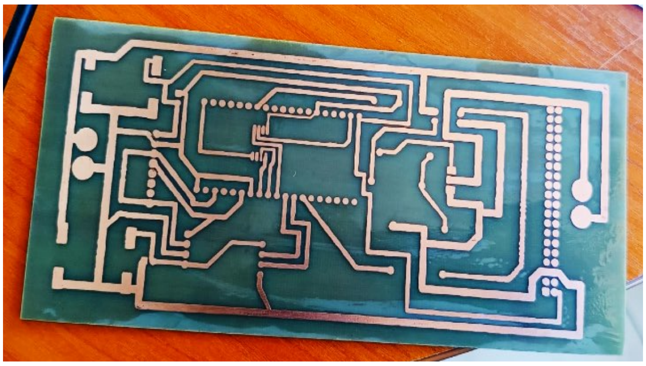
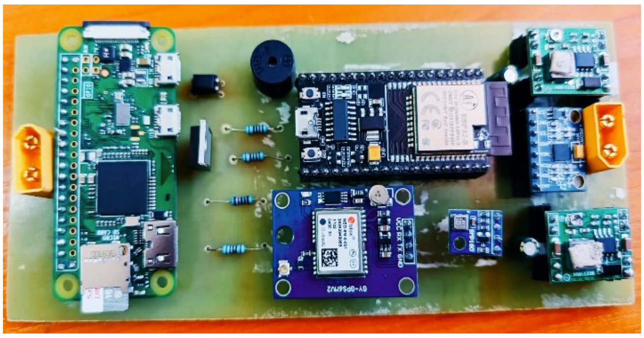
The Flight Computer Team was tasked with the design and fabrication of the rocket flight computer PCB. After consultations with both N2 and N3 teams, it was recommended that the new PCB design be an improvement of the N2 PCB as it functioned correctly. The following improvements were made:

1. The SD Card was replaced with an external flash memory
2. An optocoupler was added to act as an isolator between the main circuit and the ejection circuit (both circuits were on the same board)
3. Raspberry Pi Zero was added to the circuit to allow video streaming

1.1. PCB Iterations

The flight computer was re-designed severally with three iterations in total. The following details the specific changes made to obtain the final design.

Iteration	PCB	Changes Made
1st		• As an improvement of N2 flight computer, **raspberry pi was added** to the board • Both the ejection and main circuit was on the same board • An **optocoupler** was added to isolate the ejection and main circuit
2nd		• Added an **XT60 connector** to be used for connection to nichrome wire to allow heating of pyrotechnic charge (crimson powder) • Used an **input resistance of 100 Ohms** and **output resistance of 11k ohms.**

3rd & Final	 	● Changed the **position of the power and the ejection port (XT60 connectors)** for ease of integration with the avionics bay ● Changed the **width of the power supply line** board to handle large amount of current to the nichrome wire ● **Repositioned the mosfet** to prevent damage to the esp32. ● **Added a 100 ohms resistor** to reduce the amount of gate current that normally contributes to the heating up of the mosfet.

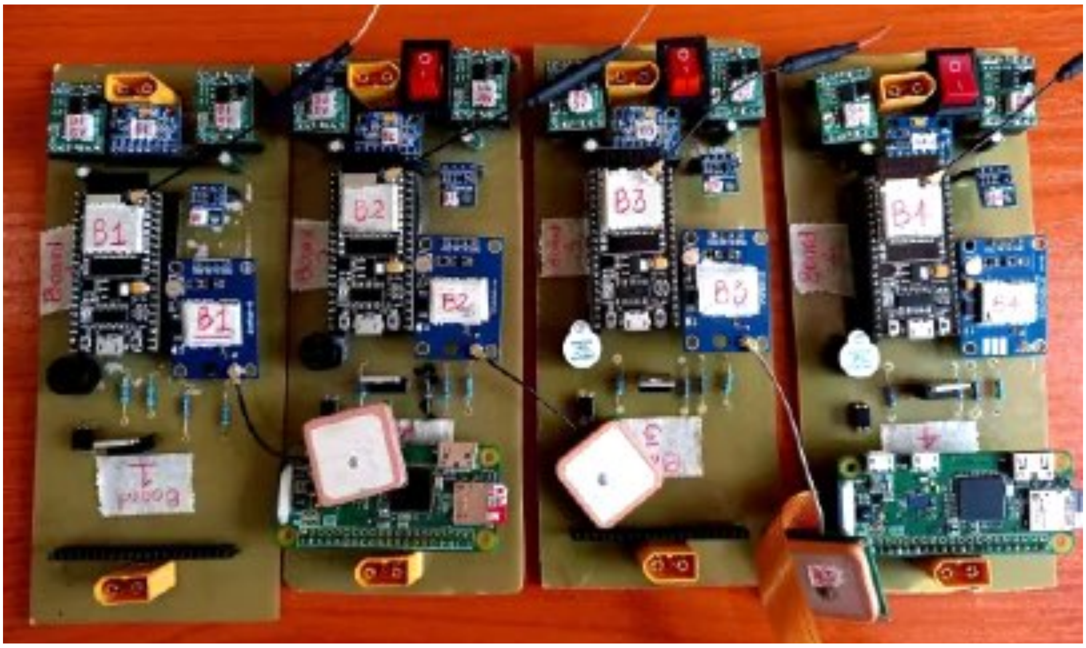
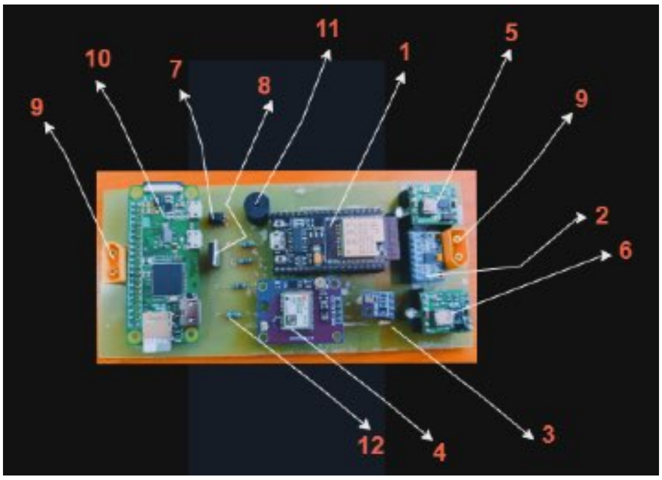


Fig: Final design with all 4 boards for each of the four rockets

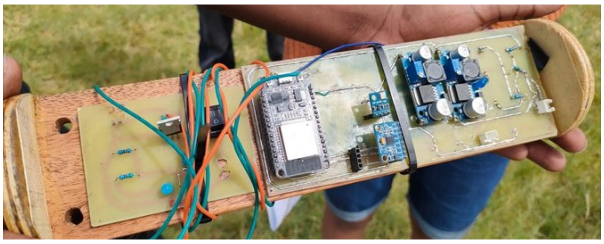
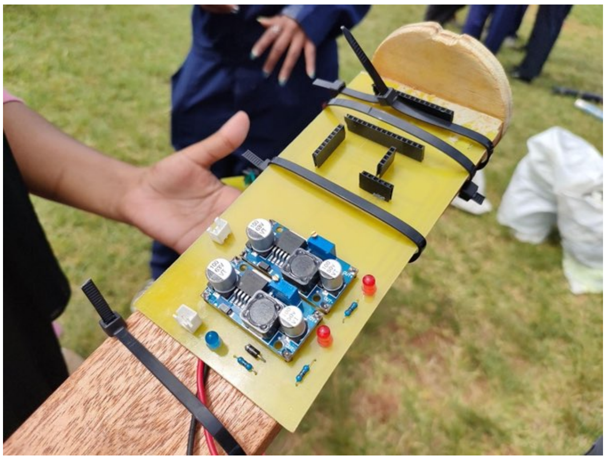
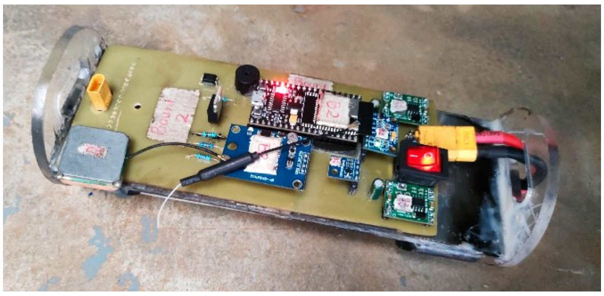
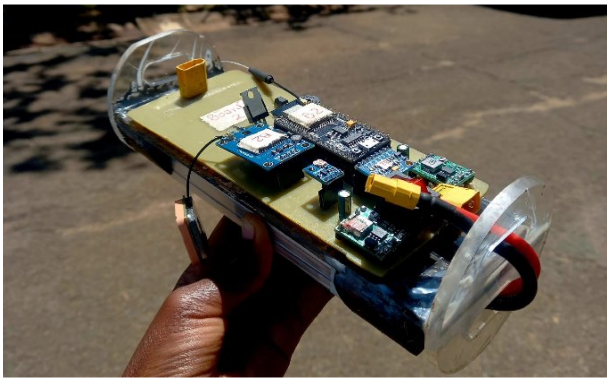
1.2. Flight Computer Parts Breakdown



1. **ESP32 NodeMCU-32S** - The central processing unit, interprets sensor data and executes flight software.
2. **MPU6050** Sensor - Has an accelerometer and gyroscope to measure the rocket's acceleration and orientation in real-time.
3. **BMP180** Sensor - Barometric Pressure and Temperature sensor that measures air pressure and estimates altitude.
4. **GPS NEO-6M (With an external antenna)** - Gives geolocation with map tiles displayed on the base station:
5. **5V Buck Converter** - Steps down 14v from the battery to 5V to be used by the ESP32, Raspberry Pi and the BMP Sensor
6. **3V Buck Converter** - Steps down 14V from the battery to 3V to be used for the rest of the components
7. **Optocoupler** - Isolates the main circuit from the ejection circuit
8. **MOSFET** - Functions as a high-power switch, controlling the current to the nichrome wire for ignition.
9. **XT60 Connector** - Connects to the battery and to the nichrome wire used to ignite the crimson powder. (NB: For the final design, this was replaced by **XT30 connectors** for compact design) Battery used: **14V, 5200A LiPo Battery**
10. **Raspberry Pi Zero** - For video streaming during flight (paired with a Pi Camera)
11. **Buzzer** - Beeps once to indicate the ESP32 has connected to Wi-Fi, then beeps a second time to indicate all sensors have been detected
12. **Resistor(s)** - Manage current flow and control voltage levels within various parts of the circuit.

1.3. Avionics Bay Design

The avionics bay, responsible for housing the flight computer and battery, was re-designed. The following details the improvements made on the bay design:

N-3 Bay	 	• Used **wood** material • Used **zip ties** for mounting of flight computer, ejection board and battery • Was **long** and **bulky**
N-3.5 Bay	 	• Used 4mm **perspex** sheet • **Compact** design • Perspex was **laser cut** and **epoxy** was used to glue the parts together • **Screws** were used to secure the flight computer • The battery had **3D printed supports** that were also screwed on the bay • Every component was **firmly mounted** on the bay with no hanging wires

2. TELEMETRY DESIGN

The telemetry team employed a communication system to ensure reliable data transmission and monitoring throughout the rocket's flight. This system integrated modern communication protocols and hardware components to facilitate real-time data exchange between the rocket and the ground control station. The breakdown is as follows:

2.1. Communication Protocol: MQTT with Mosquitto

MQTT (Message Queuing Telemetry Transport) was the lightweight, publish-subscribe network protocol used. Its design is well-suited for high-latency, low-bandwidth, or unreliable networks, making it ideal for rocketry applications.

MQTT Features:

- Lightweight - Minimal protocol overhead, crucial for bandwidth-limited environments.
- Reliable - Quality of Service (QoS) levels ensured message delivery (QoS 0, 1, and 2).
- Scalable - Capable of handling numerous devices and data points, supporting complex telemetry systems.

Mosquitto was the open-source MQTT broker that acted as the server in this setup. It facilitated communication between the rocket (clients) and the ground control systems (clients). The following gives a breakdown of the mosquitto server:

- Message Broker Role - Receives messages from publishing clients (sensors on the rocket) and distributes them to subscribing clients (ground control).
- Publish/Subscribe Model:
    - ➔ Publishers - Devices (like sensors) that send data to specific topics.
    - ➔ Subscribers - Devices or systems that receive data by subscribing to topics of interest.
    - ➔ Topics - Hierarchical channels through which data is transmitted, allowing for organised and efficient data flow.
- Quality of Service (QoS) - Ensures reliable message delivery with three levels: QoS 0: At most once delivery, QoS 1: At least once delivery, QoS 2: Exactly once delivery.

2.2. Hardware Components

1. **Ubiquiti NanoStation** - Acted as the primary access point for the telemetry data transmission. The following are its advantages:
    - High-Performance - Long-range wireless communication capability.
    - Stability - Reliable connection in various environmental conditions.
    - Flexibility - Could be configured for different network topologies and requirements.

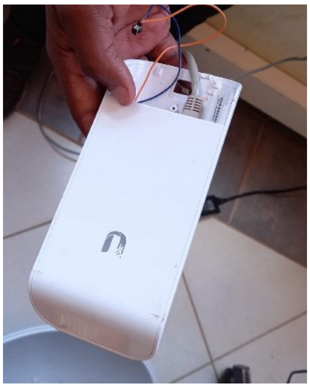


Fig: Ubiquiti Nanostation Antenna

2. **WiFi Router with PoE (Power over Ethernet) -** Provided a local network environment to facilitate communication between the rocket's telemetry system and the ground control station.
    - PoE - Supplied power to the Ubiquiti NanoStation and other network devices via Ethernet cables, simplifying installation and reducing the need for additional power sources.
    - Network Management - Managed the data traffic, ensuring efficient and prioritised telemetry data handling.

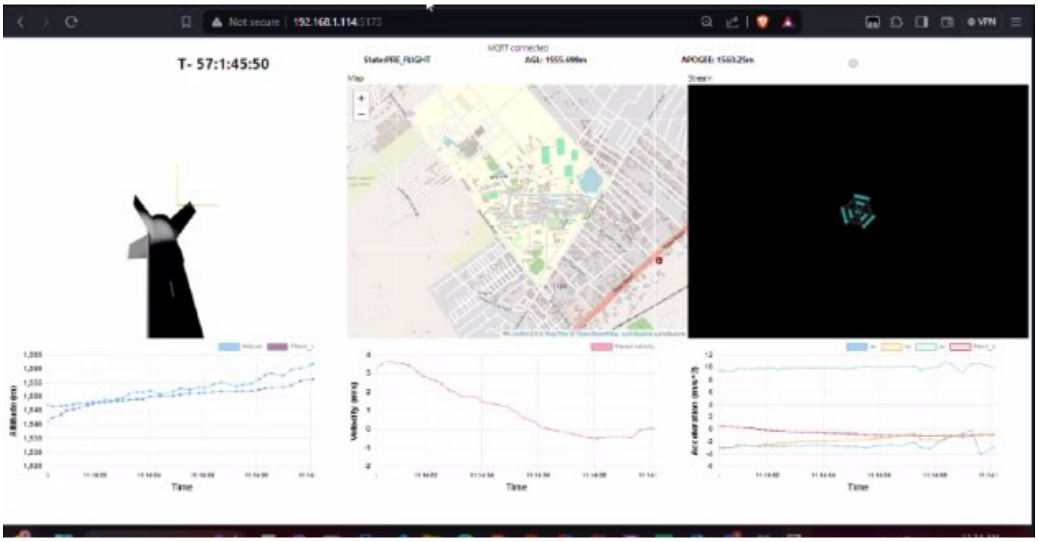
2.3. System Workflow

I. Sensor Data Collection - Various **sensors on the N-3.5 rocket collected critical flight data** (e.g., altitude, velocity, acceleration). Sensor data was processed by the onboard microcontroller, formatted, and prepared for transmission.

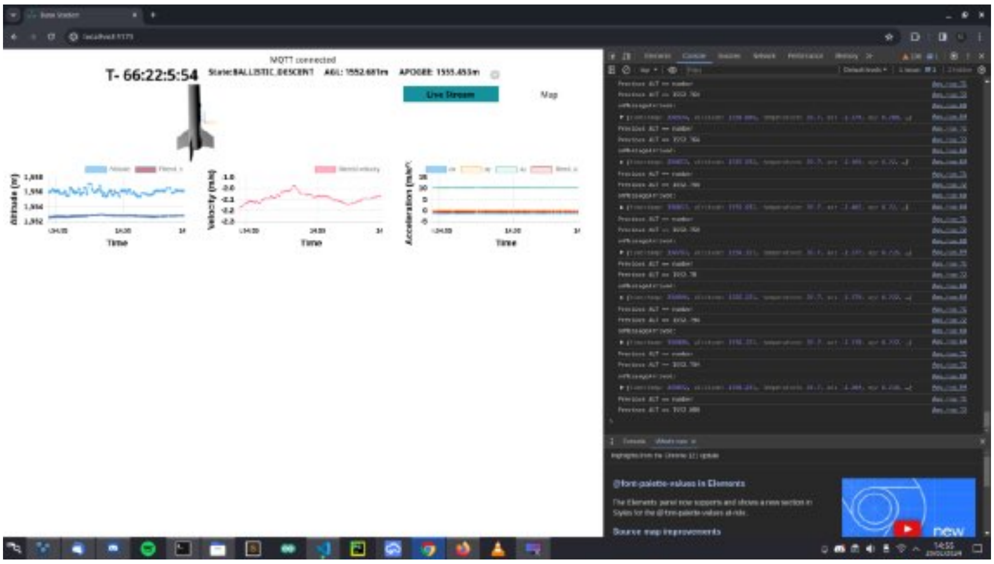
II. MQTT Protocol Implementation - The onboard system acted as an MQTT client, publishing data to topics managed by the Mosquitto broker. **MQTT messages were sent over WiFi to the ground control system via the Ubiquiti NanoStation.**

III. Data Transmission - The **Ubiquiti NanoStation was mounted on a stable antenna stand structure at the ground station**. This positioning ensured a clear line of sight to the rocket during its flight.. The WiFi router, connected via PoE, ensured stable power supply and efficient data routing.

IV. Ground Station Reception - The ground station's MQTT clients subscribed to relevant topics to receive real-time telemetry data. The following image shows the **base station setup showing real time data**:



V. Data Analysis and Storage **-** Real-time data was displayed on the base station web app. All data was logged for post-flight analysis, aiding in performance assessment and future mission planning.



3. PARACHUTE EJECTION MECHANISM DESIGN

The N-3.5 rocket utilised a simple and reliable parachute deployment mechanism designed for a **single main parachute**. The system used crimson powder as the pyrotechnic charge and a nichrome wire was used to ignite the crimson powder.

3.1. Ejection System components

- **Piston Mechanism** - Housed within a slim steel cylinder, this mechanism drove the deployment.

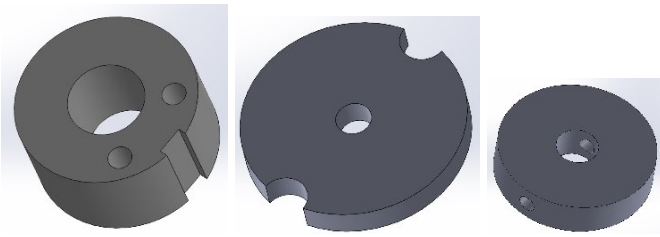


Fig: 3D design of piston parts, 76mm diameter

Fig: Assembled piston with 3D printed parts

- Pyrotechnic Charge - **Crimson powder** was used as the propellant.

Fig: Sieving of prepared crimson powder

- Ignition System - A **single nichrome wire** ignited the charge.

Fig: Cutting of nichrome wire

- Parachute Housing - A **<u>pvc tubing</u>** securely housed the folded parachute.

Fig: Piston mechanism with parachute housing

- Release Mechanism - The **piston's forward motion** triggers parachute release.
- **<u>Single Main Parachute</u>** - This parachute was carefully selected and sized to ensure safe deceleration.

Fig: Folding of parachute

3.2. Deployment Sequence

I. Trigger - When the rocket reaches apogee or if the ejection is triggered manually from the ejection test-UI, a high signal is sent and the nichrome wire heats up

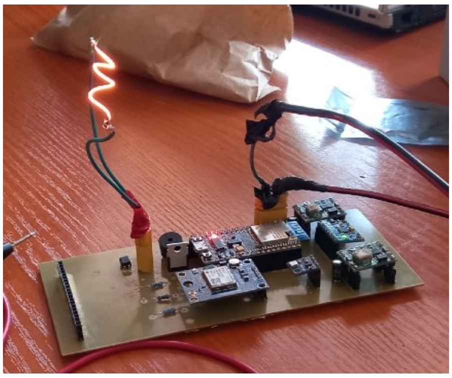


Fig: Heated up nichrome wire after a high signal has been sent by clicking a manual eject button on the ejection test-UI

II. Ignition - The primary nichrome wire heats up, igniting the pyrotechnic charge

III. Gas Generation - The burning charge rapidly creates high-pressure gas.

IV. Piston Actuation - The expanding gas pushes against the piston, driving it forward.
V. Parachute Release - The piston's forward motion releases the parachute from its housing.

Fig: Released parachute tied to nose cone and nose cone coupler

VI. Parachute Deployment - The packed parachute unfolds and is inflated due to airflow, slowing the rocket's descent.

3.3. Double Deployment Designs

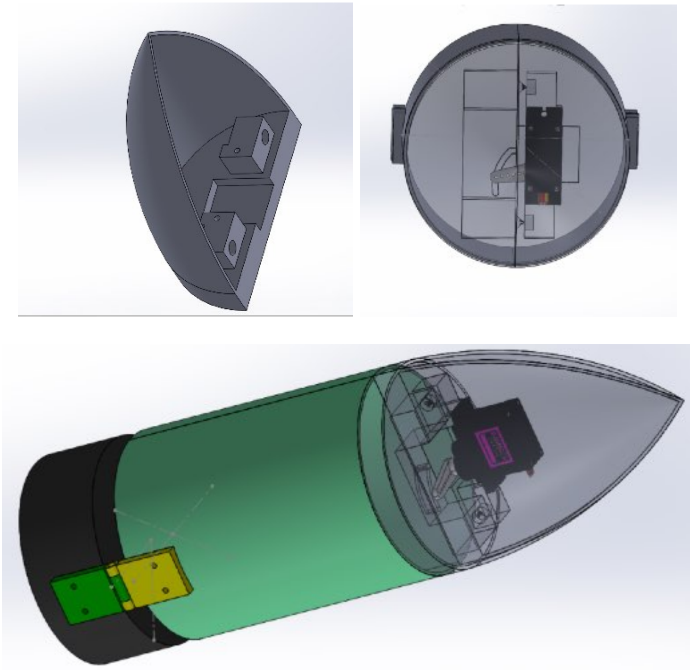
The team concurrently worked on designing a double ejection system, although it was eventually not implemented for the N3.5 launch. This design is compliant with Spaceport America Cup regulations and will be used in future launches. The team devised two designs for the ejection system: one utilising a spring mechanism and another employing a rack and pinion mechanism.

I. <u>Spring Ejection Mechanism</u>

This mechanism uses mechanical springs and a servo motor to manage the sequential deployment of two parachutes (typically a drogue and a main chute) to ensure a controlled descent. For the design the team came up with, the setup involves compressed springs housed in the nose cone. The springs store the potential energy required to push the parachutes out of the rocket. A servo motor, MG90S Servo motor, is integrated into the system to control the release mechanisms of the springs, enabling precise timing for each deployment.

The deployment sequence begins with the initial ejection of the drogue chute. At apogee, a signal triggers the servo motor to rotate, releasing the latch that holds the first spring in place. The spring then expands the nose cone, opening it up and pushing the drogue parachute out of its compartment.

The drogue chute stabilises the rocket's descent and reduces its speed. At a predetermined lower altitude, a second signal is sent to the servo motor to release the latch holding the second spring. The second spring expands, ejecting the main parachute, which further slows the descent for a safe landing.



Figs: 3D Designs of spring ejection mechanism

Fig: Spring ejection mechanism bench test implementation with 3D printed nose cone and MG90S Servo Motor

This design was implemented and a 40 degree rotation was obtained

NB: This design was dropped as the **potential energy stored in the springs was not enough** to open up the nose cone fully. In addition, **availability of springs locally was limited.**

II. Rack and Pinion Mechanism

Replaced spring mechanism with rack and pinion mechanism to **eliminate dependency on springs**. The setup involves a pinion gear, which is rotated by a servo motor, and a rack, which translates the rotational motion into linear motion to push the parachutes out of the rocket compartments.The deployment sequence begins with the initial ejection of the drogue chute. At apogee, a signal triggers the servo motor to rotate the pinion gear. The pinion gear engages with the teeth of the rack, moving it linearly. Rack and pinion teeth allow force distribution over a wider area at their points of contact.

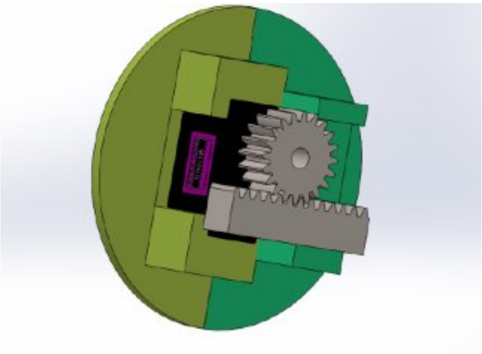


Fig: 3D design of rack and pinion mechanism

NB:

- This mechanism was neither physically tested nor implemented due to **challenges with 3D printing** and **time constraints.**
- The mechanism employed for all four rockets for N3.5 launch was a **single event piston ejection mechanism**. The mechanism is shown below:

Fig: Piston ejection mechanism

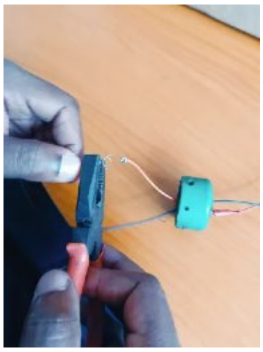
4. GROUND TESTING

Ground tests were done as part of pre-launch preparation to ensure reliability of each system. Three main tests were done - Pop Tests, Range Tests and Drone Tests.

4.1. Pop Tests

The objective of the pop tests conducted was to evaluate the functionality of the parachute recovery system. This included testing the performance of the ejection circuit, assessing the reliability of the ejection mechanism, evaluating the ignition of the crimson powder, and verifying the successful deployment of the parachute.

Several pop tests were conducted over a period of three months and the following details the **11 step sequence** followed in preparation for pop tests:

			
1. Sieving of crimson powder	2. Folding paper with crimson powder	3. Cutting of nichrome wire	4. Testing the circuit

			
5. Folding of parachute	6. Tying of parachute	7. Fitting parachute housing in the airframe	8. Screwing of piston into the airframe

 9. Fitting coupler and nose cone	 10. Testing of circuit on site	 11. Final setup

The trigger for heating the nichrome wire was manually controlled through an **<u>ejection button</u>** on a web app, which served as the **ejection UI**. When the ejection button is pressed on the UI, a high signal of 14V is sent, heating the nichrome wire and igniting the crimson powder.

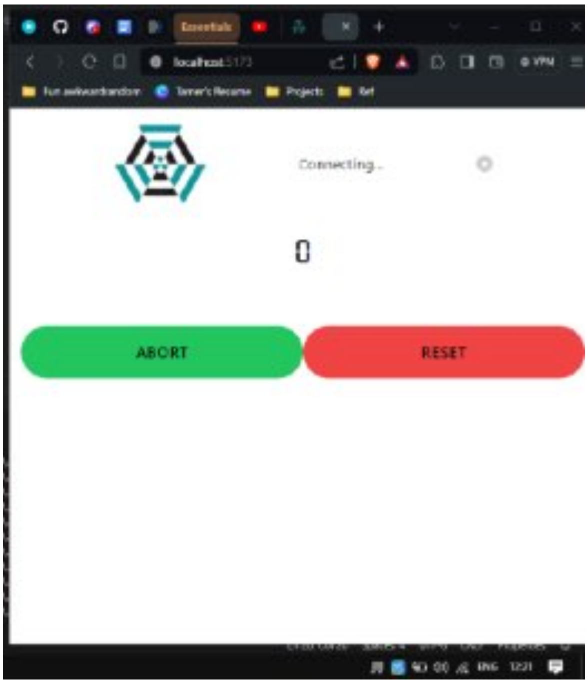


Fig: Ejection-UI after the ejection button has been pressed. The abort button changes the voltage from 14V to 0V. The reset button clears any ongoing processes or errors.

Challenges encountered during pop tests:

1. There was a 10 second delay before the nichrome wire burned.

   Mitigation: **Increased PCB track width** on the ejection circuit to allow carrying of more current.

2. The esp32 could pick a signal within a range of only 3 metres.

   Mitigation: Added a **Wifi antenna** to the ESP and the range was increased.

3. The explosive force was not enough to eject the coupler, thus preventing parachute ejection. **10 grams** of crimson powder was initially used.

   Mitigation: Used **21 grams** of crimson powder

4. The explosive force from 21 grams of crimson was too much and kept destroying the parachute housing and the piston parts.

   Mitigation: Reduced crimson powder from 21 grams to **17 grams** and secured the piston parts using **screws** and **epoxy** as shown below:

Figure: Screwed the stopper to prevent it from moving during ejection

Figure: Added epoxy to fasten the parachute holder to the piston

5. The shock cord snapped during one of the tests, separating the nose cone from the main rocket body.

   Mitigation: Used a **stronger nylon rope** as the shock cord

6. The piston kept going off axis, getting stuck and causing force to be transmitted to the lower cup causing damage

Fig: Damage done to the lower cup on the piston

Mitigation: Machined a **piston guide** to keep the piston straight and in-axis

Fig: Machined guide and the guide implemented on the piston mechanism

7. The piston parts were sheared during one of the tests.

   Mitigation: The **number of screws were increased** from three to six uniformly distributed around the diameter and **more epoxy** was used.

After all these changes were made, the final pop test was successful, with the parachute ejected. **17 grams of crimson powder** was used to generate enough pyrotechnic energy to push the piston.

Fig: Pop Test showing ***successful ejection*** *of parachute*

4.2. Range Tests

Range Tests were done for the following reasons:

1. To test communication reliability - Range tests help determine the reliability of communication links between the rocket and the base station over various distances. This is crucial for ensuring continuous communication throughout the rocket's flight trajectory.
2. To validate system performance - By measuring signal strength and link quality at different distances, range tests validate the performance of the avionics and communication systems onboard the rocket. This includes ensuring that telemetry data, commands, and other critical information can be reliably transmitted and received throughout the flight.

Two range tests were conducted, one to test **900MHZ WiFi antenna** and another to test a **2.4GHz antenna**.

The following shows the range test setup checklist:

TEST SETUP CHECKLIST	
Inverter and Battery Setup	
1.	Connect ethernet from inverter to battery
2.	Turn on the battery then the inverter
Antenna Connection	
1.	Connect POE to powersource
2.	Power wifi router
3.	Connect POE to ethernet
4.	Connect POE LAN to LAN on router

Fig: Complete setup of the range test ground station with battery and inverter

Range Test 1: 900MHz antenna

Performed the first range test with the configured Ubiquiti N2 Nanostation and 900MHz antenna. The antenna is shown in the image below:

Fig: ***900MHz Wifi antenna***

Specifications:

Connector type: uFL (I-PEX)

Reinforcement: > 3dBi

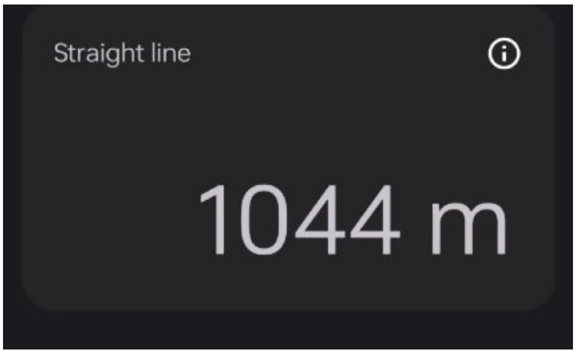
Impedance: 50 Ohm

Frequency range: 900MHz

Maximum input power: 50W

Length of the antenna: 10.8 cm and from the kink 8.7 cm

From the test, a **straight line distance of 1024 metres** was obtained.



Img: Recovery Team on site conducting range test

Observations:

1. Concluded that the use 2.4 Ghz ESP antenna, instead of the 900MHZ would be better suited
2. The ubiquiti nanostation needed to be in **direct line of sight** with the flight computer, otherwise the signal got lost and data stopped transmitting.

Range Test 2: 2.4 GHz Antenna

Conducted a second range test with the 2.4GHz antenna

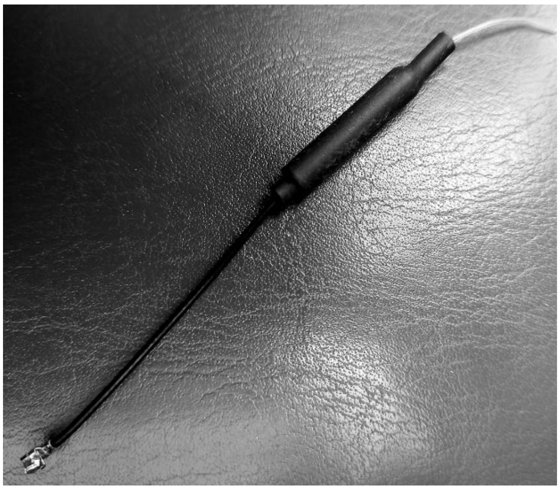


*Fig: **2.4GHz antenna***

Specifications:

Frequency: 2.4Ghz

Gains: 3dbi

Cable type: 1.13 cable

Cable Length: 5.5cm

Total Length: 10cm

Body Material: Brass

Connector: Ufl/Ipx connector

Obtained a straight line distance of **1024 metres**. This antenna was chosen for the final design due to its availability in local stores and its decent range.

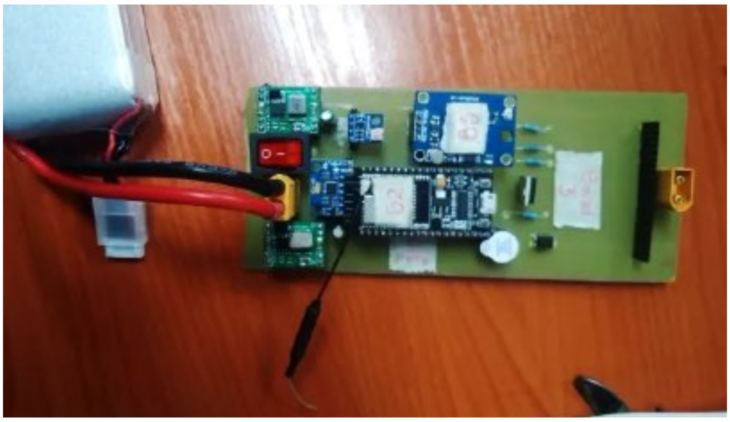


Fig: Flight computer with the 2.4GHz antenna mounted on the ESP32

4.3. Drone Tests

Drone tests were conducted to simulate flight for the purpose of testing the state machine. The state machine is a crucial segment of the flight software that indicates transitions between various flight states, including: **Pre-flight, powered flight, apogee, parachute descent, and post-flight**. These flight transitions are vital for detecting apogee to trigger parachute ejection.

In this test, the flight computer was mounted on a drone, which was then flown to simulate a rocket flight. Changes in flight states and the transmission of data were visualised on the base station, a web based application that allows visualisation of data.

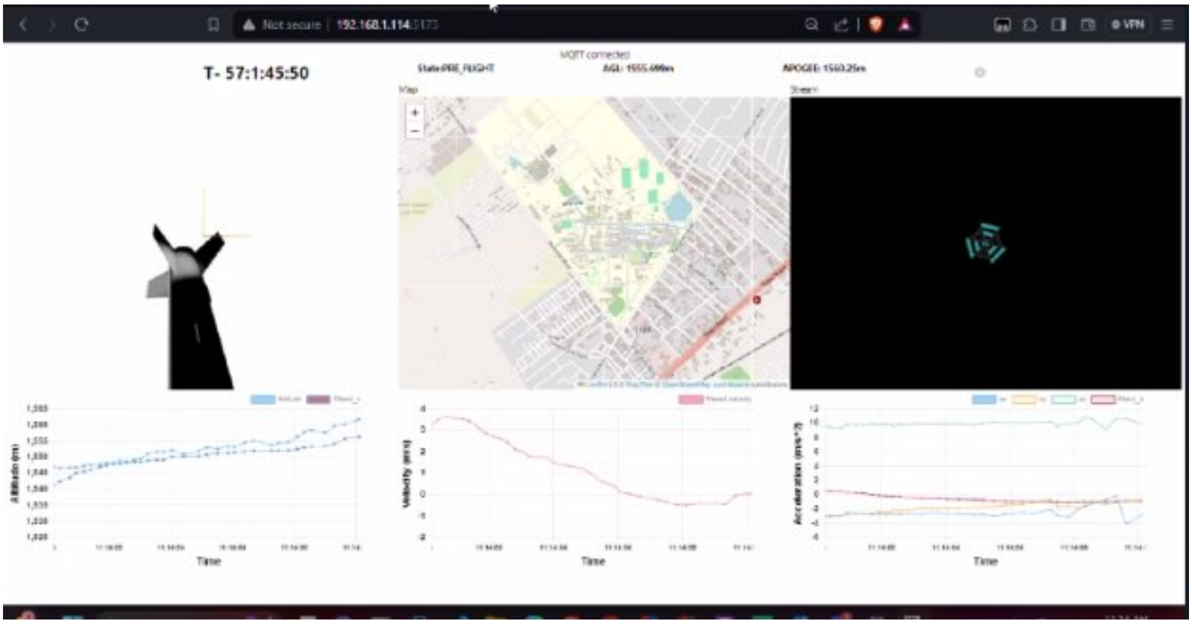


Fig: Visualisation of sensor data, rocket orientation, geolocation, video stream and flight state on web application on the base station

.

Two drone tests were conducted using a DJI Matrice M300/M350 drone.

Fig: Drone Test setup

Challenges encountered

1. Data logging - Team encountered issues with storage of data transmitted during the drone test.
2. Video transmission - The Pi Camera could not transmit live video stream.

LAUNCH SUMMARY

The launch was conducted on 8th May, 2024 with three out of four rockets launched.

1. Beta Rocket

- The first launch of the Beta rocket demonstrated **continuous data transmission** before, during, and after the flight.
- The recovered data indicated that the rocket reached an **apogee of 167 metres**.
- Although the **parachute deployed, it did so at a very low altitude** and the nylon rope used as the shock cord snapped, leading to the separation of the nose cone and coupler from the rocket body.

2. Delta Rocket

- The Delta rocket achieved a significant improvement in altitude, reaching **over 1,000 metres**.
- The parachute successfully deployed; however, this was not due to the intended piston mechanism. The parachute also got separated from the rocket body not so long after ejection.
- Unfortunately, the **signal was lost** after the rocket was mounted on the launch pad, leading to lack of data transmission during and after flight

3. Charlie Rocket

- The third launch with the Charlie rocket encountered significant issues.
- The **signal was lost early in the flight**, leading to a lack of data transmission.
- Additionally, the **parachute failed to deploy**, which resulted in a compromised recovery of the rocket.

Ejection misfires

Accidental firing occurred three times during assembling. In the beginning, the short circuit of the flight computer PCB was suspected. However, the main culprit was concluded to be the interference between the first flight computer, which had already launched and landed, and the second flight computer, which was being prepared on the ground. The first flight computer was continuously sending its own state to the base station, which mistakenly updated the state of the second flight computer. To address the issue, it was proposed that each flight computer PCB must have a unique identifier, and the base station must recognize each board as a different board.

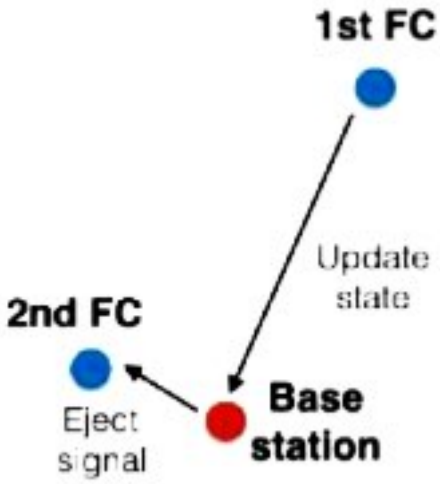


Fig: Accidental firing of crimson powder

POST-LAUNCH ANALYSIS

Fig: Recovered avionics bays after launch (The 3rd undamaged avionics bay is for the fourth rocket that was not launched)

Observations and Analysis

1. **Loss of Communication with the Base Station**
    - Communication issues were attributed to the ESP and GPS antennas being enclosed inside the metallic airframe, which attenuated most of the signal.
    - To resolve this, it is recommended that both antennas extend outside the airframe.
    - Additionally, WiFi proved too weak for reliable long-range communication. Future launches should use RF modules with appropriate frequencies connected to the ESP32 MCU via UART for downlink. Adequate antenna testing should also be conducted at the ground station to ensure reliability during launches.
2. **Flight Data Recovery and Storage**

- Data recovery issues were encountered due to damages to some ESP modules and signal loss.
- Data from one ESP was incomplete, potentially due to the 4MB limit of its flash memory.
- Incorporating external memory, such as flash memory, could address this issue and ensure complete data recovery.

3. **State Machine Failure: Ejection Misfires**

- The state-machine malfunctioned leading to several misfires of the chute ejection charge. Proposed solution is working on the software and adequate testing of its operation. Few tests were done to test the state-machine, more tests need to be arranged throughout the development process
- To prevent signal interference, it is suggested that each flight computer subscribe to distinct MQTT topics. Using separate servers for each board will facilitate easier debugging and prevent issues encountered when all four boards were on a single IP address.
- The pre- and post-flight states had identical conditions, causing the flight computer to repeatedly send a high signal to ignite the crimson powder and eject the parachute while on-site. Better-defined conditions for these states are needed to prevent this issue.
- Modular unit testing of software components, including sensors, data filtering, data logging, geolocation, data downlink, and the ejection system, should be conducted frequently. Hardware and software-in-the-loop testing, along with drone testing, are recommended.
- Implementing safety-critical software with a safe mode to turn off potentially harmful systems before flight is also proposed.
- Lastly, comprehensive ground station state reporting needs to be done. Most flight conclusions were made visually rather than through ground station software. Upgrading the ground station software to provide more comprehensive state reporting is necessary.

4. **Delay in On-Site Assembly**
    - Modules that require minimal on-site assembly, such as screwing components together, should be used. The recovery team spent significant time placing crimson on-site.
    - A successful technique from the solid team involved preparing crimson before travelling and wrapping it in silica to prevent dampening. This method should be applied consistently, requiring only the removal of the silica wrap on-site.
5. **Weak Shock Cords**
    - The parachute separated from the airframe after deployment, possibly due to loose connections. The connection between the parachute and the airframe should be made more rigid.
    - The first rocket's parachute deployment was compromised by a snapped nylon rope, resulting in the separation of the nose cone and coupler from the rocket body. Stronger materials should be used for these connections in future launches.
6. **Difficulty in Placing Avionics Bay in the Body Tube**
    - Issues with dimensions and component placement made it difficult to insert and remove the avionics bay from the body tubes.
    - Design improvements are needed to facilitate easier placement and removal.
7. **Difficulty in Locating Launched Rockets**
    - Recovering rockets from the thickets around the launch site was challenging.
    - This issue can be mitigated by ensuring the base station map and onboard camera function correctly, along with the GPS module.
    - Additionally, adding a siren to each rocket will aid in easier recovery after landing.

RECOMMENDATIONS

From the observations and post-launch analysis, the following recommendations for future launches were made:

1. Place **ESP and GPS antennas outside** the rocket's metal frame for better signal strength.
2. Use RF modules with suitable frequencies connected to the ESP32 for reliable communication. E.g. **XBEE module** with a range of up to 15 miles
3. Upgrade to external memory like **flash memory** to handle large data volumes and prevent data loss. A data dump can also be implemented for easier retrieval of data post flight.
4. Improve software and conduct more tests to prevent misfires.
5. Use **separate topics** for each flight computer to avoid signal interference.
6. Test software components (sensors, data handling, etc.) separately and frequently.
7. Implement a **safe mode** in software to disable risky systems before flight.
8. Update ground station software for better real-time reporting and analysis.
9. Simplify assembly with modular designs that require less time on-site.
10. Strengthen parachute connections to prevent failures during deployment.
11. Improve avionics bay design for easier insertion and removal.
12. Improve base station mapping and camera functionality for better rocket recovery.
13. Consider adding **sirens** to the rocket for easier location after landing.

CONCLUSION

The N-3.5 Recovery Team carried out extensive preparations and recovery operations following the N-3.5 rocket launch on 8th May, 2024. Despite encountering challenges such as communication issues, ejection misfires, and recovery difficulties, valuable insights were gained into flight computer performance, telemetry systems, and parachute deployment mechanisms. The launch itself, while demonstrating substantial progress in altitude achievements, encountered setbacks such as communication losses and parachute deployment failures. These difficulties highlighted the significance of strong hardware and software integration, particularly in maintaining state machine reliability and data recovery.

Moving forward, by implementing the recommended solutions, building on lessons learned, and improving the overall recovery system, the recovery team is set up to achieve higher reliability, safety, and performance for future launches.