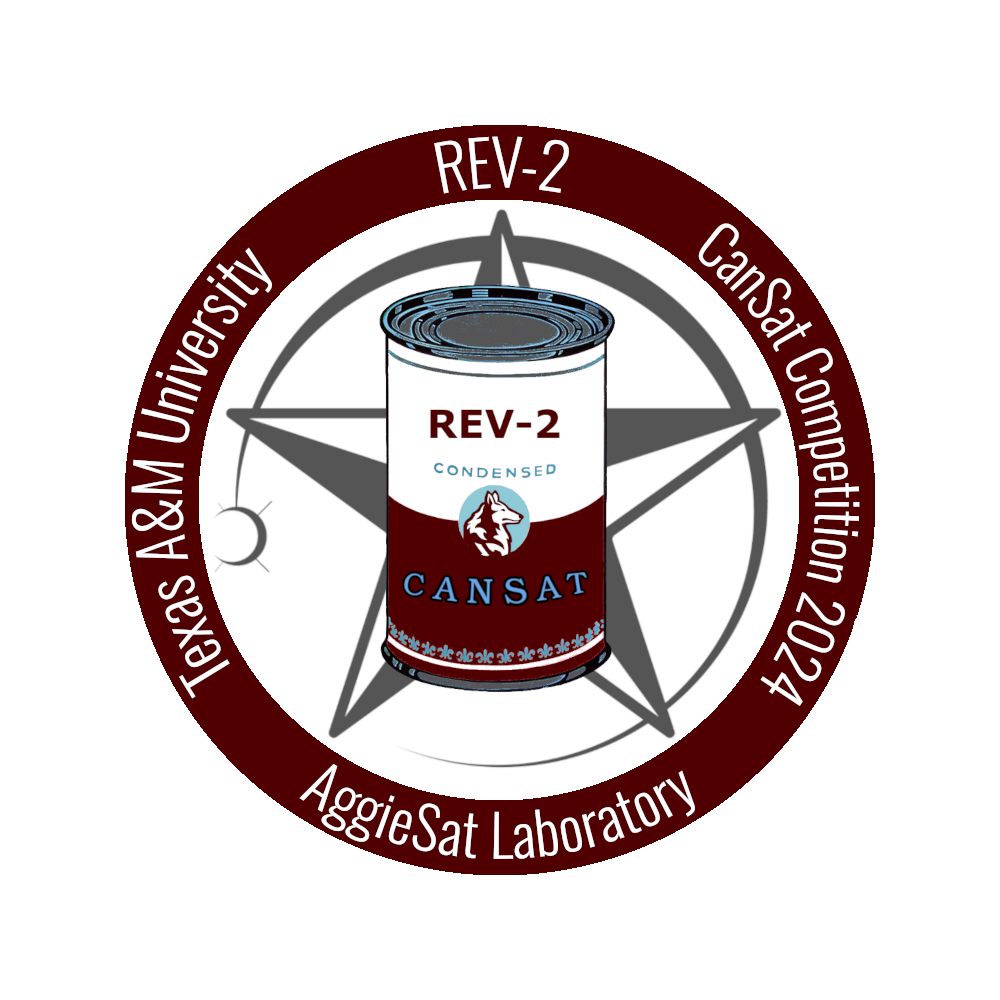
**Performance Analysis Report**

**Revision 1.0**

****

REV-2

AggieSat Laboratory

15 June 2024

Prepared by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Tate Crawford Date

Project Manager, REV-2

Reviewed by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Shirish Pandam Date

Program Manager, AGSL

| **REV** | **Date** | **Description of Change** | **Prepared** | **Reviewed** | **Approved** |
| --- | --- | --- | --- | --- | --- |
| - | 6/15/2024 | BASELINE RELEASE | TC |  |  |
| - | 7/8/2024 | VERSION 1 RELEASE | TC |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[**1 Executive Summary 4**](#_ch5v0w1y4nwx)

[**2 Introduction 4**](#_6wv3ruacp9ax)

[2.1 Background 4](#_gjdgxs)

[2.2 Timeline and Budget 5](#_xoi37x7hjl9x)

[**3 Systems Engineering Breakdown 5**](#_ucpi12vz41bd)

[3.1 REV-2 Team Organization 5](#_k3qwfm322ak)

[3.2 Competition Schedule and Go No-Go 6](#_25anpu2w9mtc)

[**4 The Envisioned vs Actual System 7**](#_1ci93xb)

[4.1 Brief Description of Concept of Operations 7](#_3whwml4)

[4.2 Comparison between envisioned subsystem design and subsystem at deployment 7](#_2p2csry)

[4.2.1 Mechanical (MEC) 7](#_3o7alnk)

[4.2.2 Software (SOFT) 9](#_32hioqz)

[4.2.3 Electrical Power Subsystem (EPS) 10](#_1hmsyys)

[**5 Causation for Failures 13**](#_28h4qwu)

[5.1 MEC 13](#_wko1ofkbbhd1)

[5.2 SOFT 13](#_nh1fllti6d1v)

[5.3 EPS 14](#_4f1mdlm)

[5.4 Systems Integration and Testing 15](#_1mrcu09)

[**6 Human Factors 15**](#_37m2jsg)

[6.1 Breaks and End of School 15](#_nlxi6jhifqgm)

[**7 Lessons Learned 15**](#_97kt0vg0qaqq)

[7.1 Earlier Integration 15](#_yp6dgnkiaaw2)

[7.2 Inter-subteam Communication 16](#_4hqk7ecctg0t)

[7.3 Earlier Testing 16](#_e42c8kh6ysb6)

[7.4 Changed Go/No-Go Schedule 16](#_vpnpltwyno8h)

[**8 Conclusion 17**](#_9p6w2ymnx8fi)

[**9 References 17**](#_46r0co2)

# **Executive Summary**

The REV-2 project was the AggieSat Laboratory’s entry into the 2024 International CanSat Competition. The goal of the project was to deploy a nose-cone shaped probe with an egg as its payload to collect telemetry and safely return to the ground. The system was fairly well integrated and completed many mission objectives, however, mid-flight technical issues resulted in a failure of the parachute to deploy, leading to partial mission success. The REV-2 team has determined the root cause of the issue was in the testing and integration timeline in the latter stages of the project, in addition to availability and minor technical issues during the testing phases. The team has also identified key strategies for future teams to keep in mind for future iterations of this project.

# **Introduction**

## ***Background***

The REV-2 team was formed for participation in the 2024 International CanSat Competition hosted by the American Astronautical Society. The competition consisted of 40 teams attempting to complete an extensive set of mission objectives, primarily including surviving launch atop a rocket, collecting and displaying live telemetry, controlling descent speed, and safely landing the payload (a hen’s egg) to the ground. Success and rankings were determined by a set of requirements defined in the CanSat 2024 Mission Guide published by the AAS, however, the primary requirements listed above were weighted significantly higher than other objectives.

On June 9th, 2024, REV-2 was launched in Monterey, VA, atop a J-class rocket. The project successfully collected data from most of its sensors and transmitted the data to its ground station. However, the CanSat failed to deploy its parachute and protect its egg, resulting in the fuselage suffering severe damage and the egg cracking upon impact. This resulted in REV-2 scoring an 80.96/100 and being placed 20th out of 40.

Although the payload was ultimately destroyed, the successful collection and plotting of telemetry data throughout the course of the mission has allowed the REV-2 team to designate the mission as a partial success. This document assesses the factors embedded in REV-2’s life cycle that led to the final result of the mission and details preventative measures for future teams participating in similar missions.

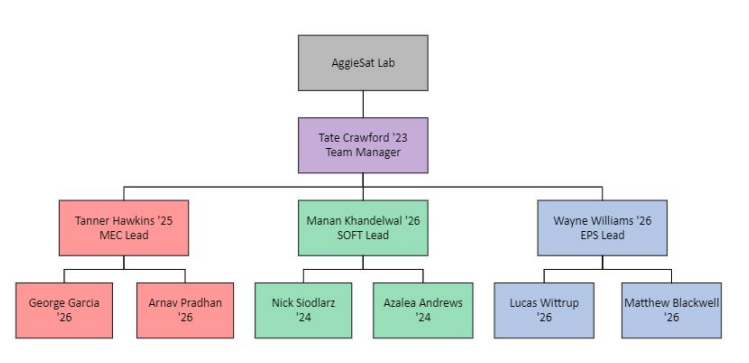
## ***Timeline and Budget***

The period for the development of the REV-2 system was approximately 8 months as specified per the CanSat 2024 Mission Guide. The REV-2 system had a total cost of $795.00 in accordance with the $1,000 limit levied by the CanSat competition mission constraints.

# **Systems Engineering Breakdown**

## ***REV-2 Team Organization***

Over the 8-month duration of the CanSat, the REV-2 team had the following members.

***Figure 1. REV-2 Org Chart***

The REV-2 team consisted of a project manager and three domain-specific subteams, each corresponding to one subsystem. Each subteam had one lead.

The project manager role was to orchestrate the budget, personnel, and technical aspects of the project in support of the project goals. The project manager also supported systems engineering efforts and facilitated general team meetings.

The subteam leads were responsible for the specific domains of their respective subsystems. Each subteam lead supervised several members. They performed technical work on the design and implementation in coordination with their subteam members, and facilitated subteam meetings.

The subteam members performed the rest of the concept, design, implementation, and support work required to create the system.

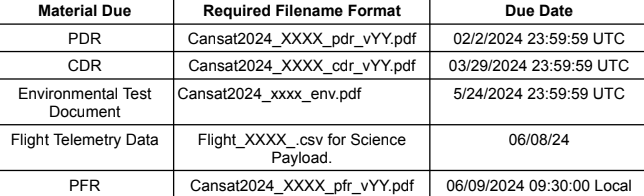
The MEC subteam designed and implemented the mechanical portion of the CanSat.

The SOFT subteam designed and implemented the command portion of the CanSat, including the flight software and ground station.

The EPS subteam designed and implemented the electrical portion of the CanSat, including power storage and distribution, sensors, RF communication, and circuitry.

## ***Competition Schedule and Go No-Go***

During the lifecycle of REV-2 there were two provided schedules that needed to be followed. They were the schedule provided by CanSat in the mission guide [1] and the REV-2 Go No-Go [2] schedule provided internally by the lab.



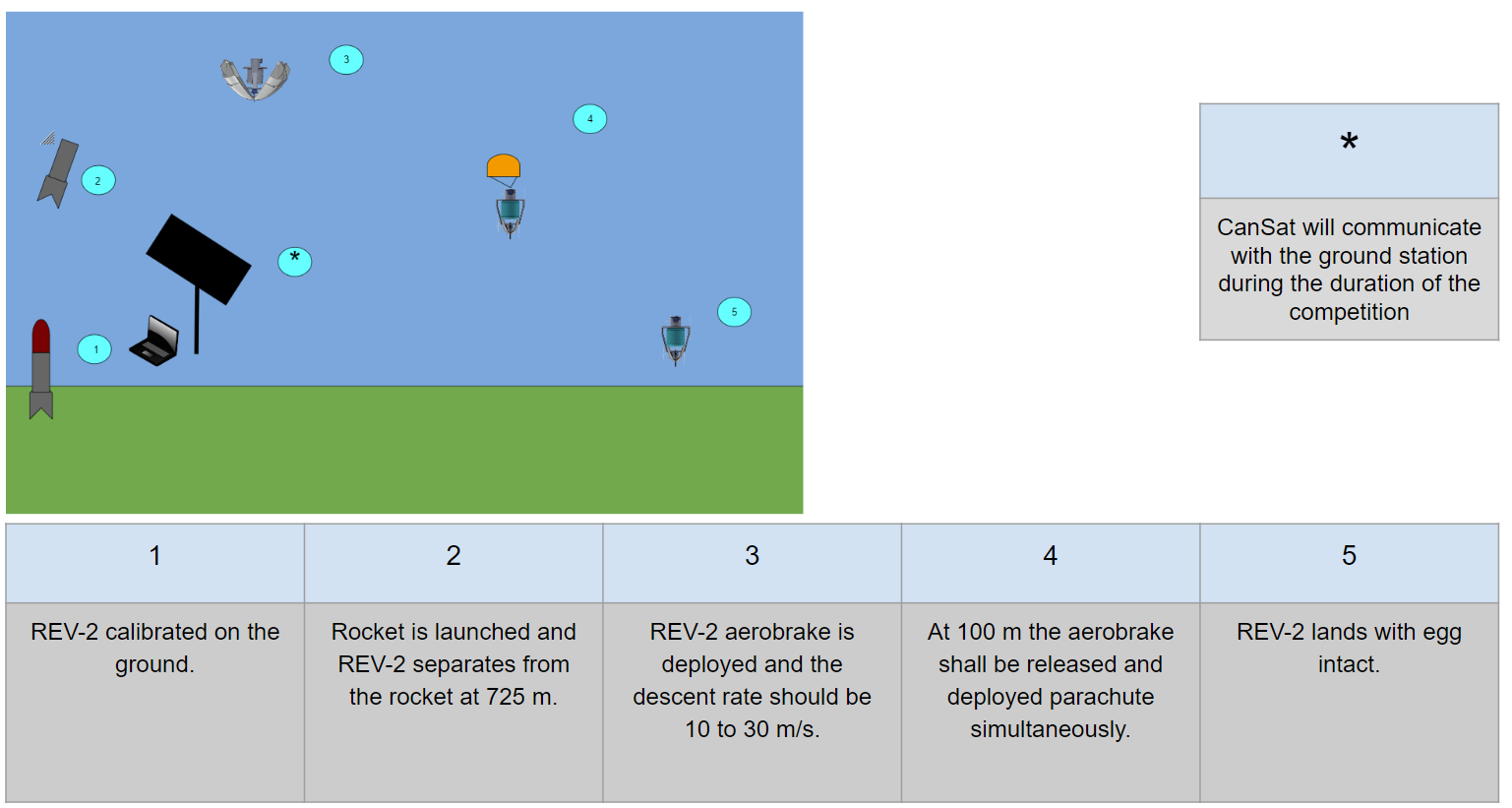
***Table 1. CanSat Schedule***

Following the schedule established by both tables gave the team a well-defined timeline to follow throughout the course of project development. This helped the team complete tasks and communicate subteam needs in a timely order. However, there were times when either the team fell behind the schedule or the order of some of the Go/No-Go tasks were out of logical sequence. For example, the deadlines for electronics integration and software verification were set to be due on the same day. The electronics were successfully integrated by the deadline but the software verification had to be rushed, requiring some team members to work in a time crunch to complete software testing. Overall, the inclusion of the Go/No-Go schedule helped the team immensely in completing the project before flight, but we propose that future teams also take note of the logical fallacies REV-2 discovered while following the timeline.

# **The Envisioned vs Actual System**

## ***Brief Description of Concept of Operations***

The REV-2 system concept of operations would be made up of a few operational modes, illustrated in ***Figure 3.1***.



***Figure 4.1 REV-2 Mission Phases***

A more detailed description of the REV-2 operational modes and specific subsystem and system level design can be found in the REV-2 CONOPS [3].

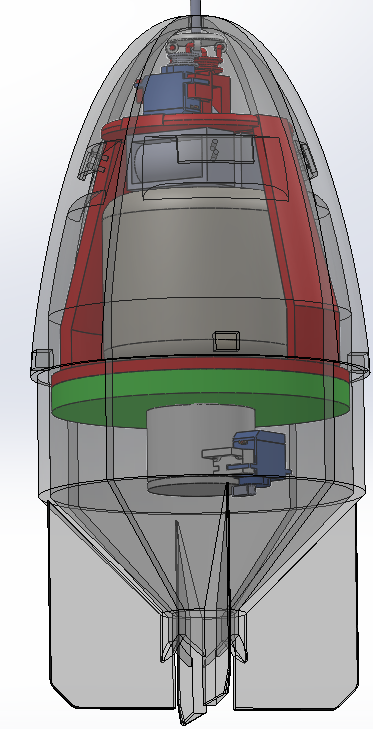
## **Comparison between envisioned subsystem design and subsystem at deployment**

REV-2 was composed of three subsystems. At the time of competition work on all subsystems had been completed and were believed to be in working condition. The section below will describe how the subsystems deviated from their planned form. To see all of the requirements for the subsystems reference the CanSat 2024 Mission Guide [1].

### Mechanical (MEC)

The MEC subsystem was responsible for creating the structure and mechanisms of the CanSat and integrating them into the final system. An overview of the most important requirements from the mission guide is listed below:

* C1: The Cansat shall function as a nose cone during the rocket ascent portion of the flight.
* C2: The Cansat shall be deployed from the rocket when the rocket motor ejection charge fires.
* C3: After deployment from the rocket, the Cansat shall deploy its heat shield/aerobraking mechanism.
* C10: During descent with the heat shield deployed, the descent rate shall be between 10 to 30 m/s.
* M7: At 100 meters, the probe shall release a parachute to reduce the descent rate to less than 5 m/s.
* M8: The Cansat shall protect a hen's egg from damage during all portions of the flight.

***Figure 4.2 MEC Finalized SolidWorks Model of CanSat (Planned)***

The image shown above in ***Figure 4.2*** shows the planned model based on the initial construction and integration of the CanSat. The final physical model contained many additional electronic components that are not shown, and several changes were made to solve problems encountered during testing.

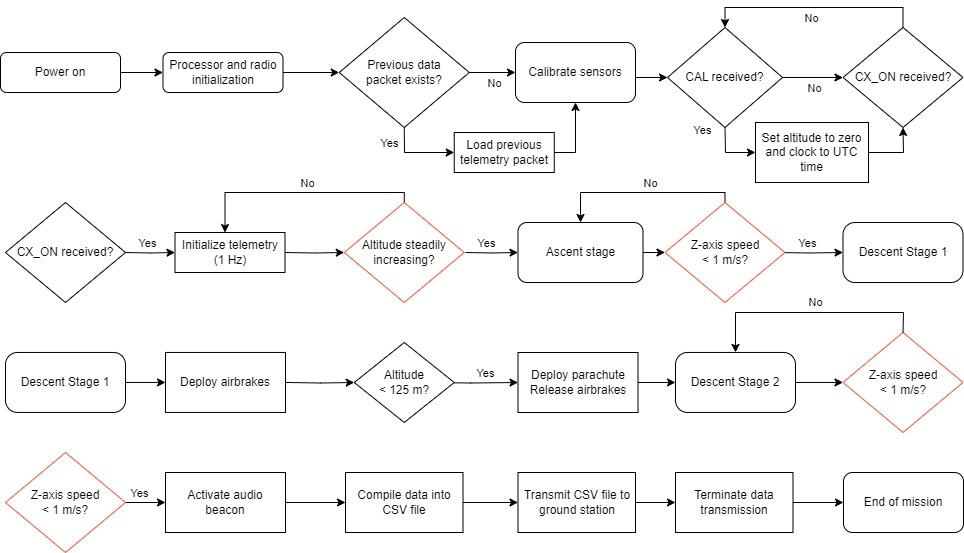
This final iteration of the CanSat successfully served as the nose cone during ascent and then deployed the heat shield completing requirements C1, C2, and C3. Under aerobraking, the CanSat fell at an average velocity of 27.65 m/s, meeting the requirements of C10. However, due to insufficient spring force pushing the parachute out of its container, the CanSat failed to deploy its parachute therefore failing requirement M7. The lack of a parachute also resulted in a hard landing causing the egg payload to break and fail M8.

### Software (SOFT)

The SOFT subteam was responsible for controlling the operation of the electrical and mechanical systems onboard as well as interpreting, communicating, and displaying telemetry data during flight. An overview of the key requirements attributed to the SOFT subteam are as follows:

* X5 - The probe telemetry shall include altitude, air pressure, temperature, battery voltage, probe tilt angles, air speed, command echo, and GPS coordinates that include latitude, longitude, altitude, and number of satellites tracked.
* G2: The ground station shall generate CSV files of all sensor data as specified in the Telemetry Requirements section.
* G6: All telemetry shall be displayed in real time during descent on the ground station.
* G8: Teams shall plot each telemetry data field in real time during flight.
* G11: The ground station shall be able to command the payload to operate in simulation mode by sending two commands, SIMULATION ENABLE and SIMULATION ACTIVATE.
* F1: The flight software shall maintain a count of packets transmitted which shall increment with each packet transmission throughout the mission. The value shall be maintained through processor resets.

***Figure 4.3*** shows the CanSat state machine as designed. Upon receiving power, the CanSat initializes all its sensors and begins transmitting telemetry packets to the ground station. It checks for existing packets in its recovery cache as a failsafe for mid-flight power losses, before starting the telemetry uplink in the prelaunch (PLCH) state.



***Figure 4.3 State Machine Diagram***

The CanSat changed states based on changing pressure differentials. When the pressure was detected to change at a rate of below -100 Pa/s, the state of the CanSat changed from PLCH to launch (LNCH). This is also when the cameras were planned to have turned on, however, the mission did not have any functioning cameras at the time of launch. Once in LNCH and a pressure differential of > 100 Pa/s was detected, the CanSat recognized the start of descent and changed the state from LNCH to the Upper Descent State (UDSC). Subsequently, when an altitude of < 100 m was detected, the CanSat was programmed to change state to the Lower Descent State (LDSC) which triggered the movement of the parachute and heat shield servos in that order. Lastly, the CanSat was programmed to switch to the Landing (LAND) state when the pressure differentials were between +20 Pa/s and -20 Pa/s to trigger the beacon, which marked the conclusion of the mission.

Throughout the course of the mission, the ground station was programmed to receive and plot live telemetry information from the CanSat. It also had the capability to remotely send signals to the CanSat, such as the reset (RST) or altitude calibration (CAL) signals sent at the start of the mission. At the time of completion, the ground station successfully met all of its requirements, however, there were two mishaps on the flight software side that are explained in a later section (*Section 5.2)*.

### Electrical Power Subsystem (EPS)

The EPS subteam was responsible for the design and implementation of the electrical portion of the CanSat, including power storage and distribution, sensors, RF communication, and circuitry. An overview of the most important electrical requirements include but are not limited to:

* E1: Lithium Polymer batteries are not allowed
* E2: Battery source may be alkaline, Ni-Cad, Ni-MH or Lithium. Lithium polymer batteries are not allowed. Lithium cells must be manufactured with a metal package similar to 18650 cells. Coin cells are allowed.
* E5: The Cansat shall operate for a minimum of two hours when integrated into the rocket.

An overview of the most important operation requirements include but are not limited to:

* C6: Upon landing, the Cansat shall stop transmitting data.
* C7: Upon landing, the Cansat shall activate an audio beacon.

An overview of the most important communication requirements include but are not limited to:

* X1: XBEE radios shall be used for telemetry. 2.4 GHz Series radios are allowed. 900 MHz XBEE radios are also allowed.
* X5: The probe telemetry shall include altitude, air pressure, temperature, battery voltage, probe tilt angles, air speed, command echo, and GPS 10 coordinates that include latitude, longitude, altitude and number of satellites tracked.

An overview of the most important sensor requirements include but are not limited to:

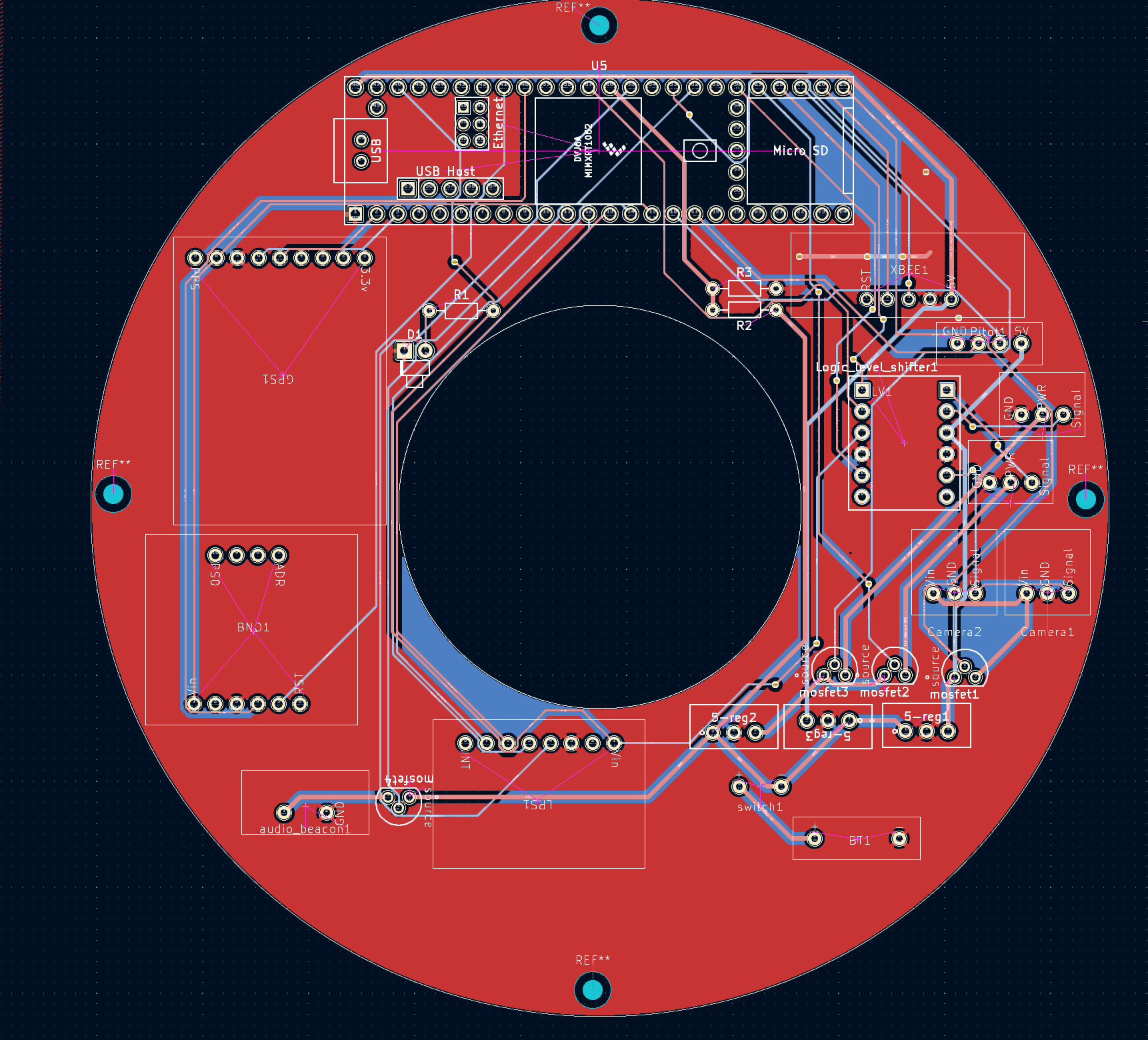
* SN1: Cansat shall measure its speed with a pitot tube during ascent and descent.
* SN2: Cansat shall measure its altitude using air pressure.
* SN3: Cansat shall measure its internal temperature.
* SN4: Cansat shall measure its angle stability with the aerobraking mechanism deployed.
* SN5: Cansat shall measure its rotation rate during descent.
* SN6: Cansat shall measure its battery voltage.
* SN7: The probe shall include a video camera pointing horizontally.
* SN8: The video camera shall record the flight of the probe from launch to landing

An overview of the most important ground station requirements include but are not limited to:

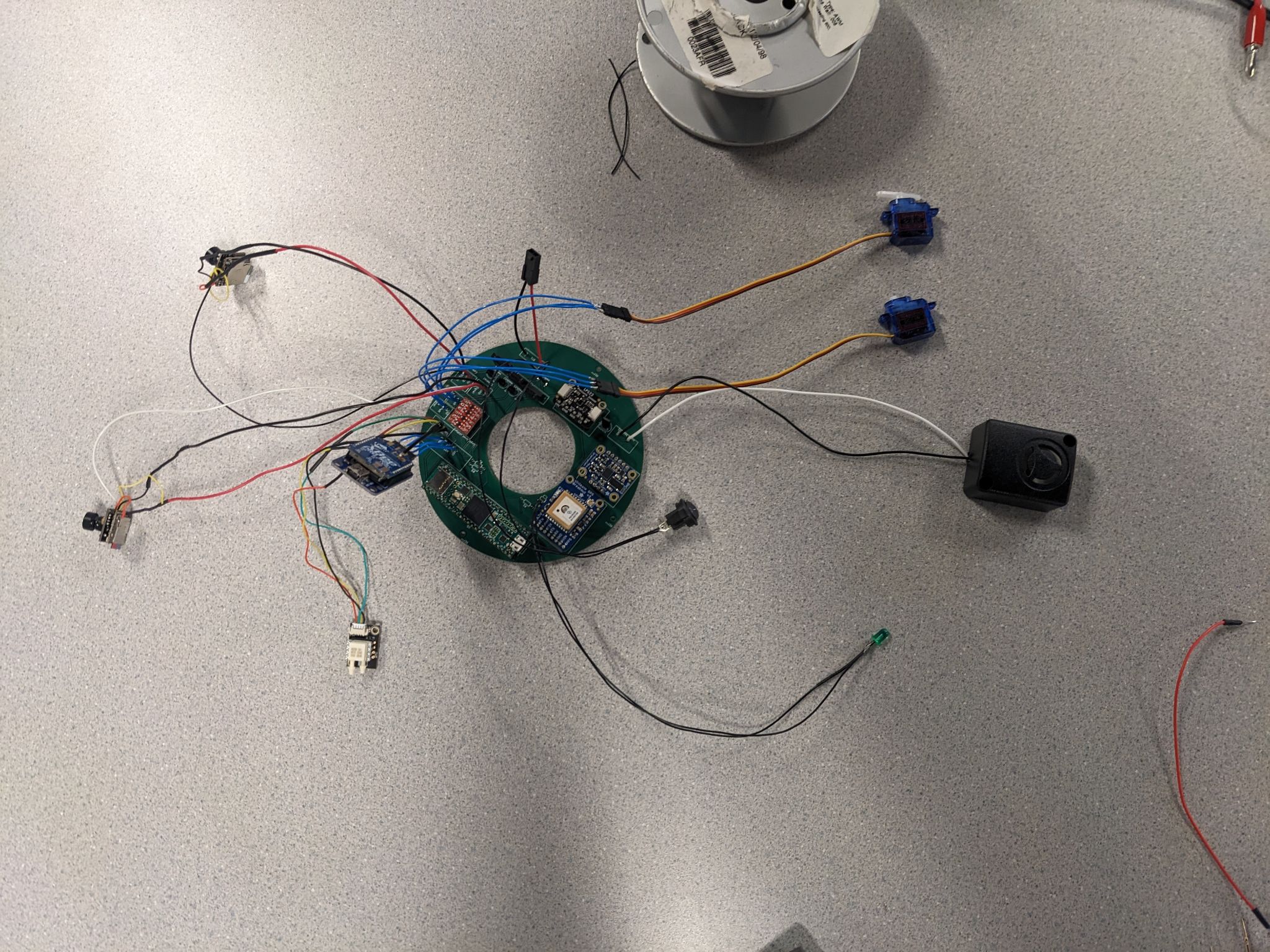
* G13: The ground station shall use a table top or handheld antenna.

To fulfill the requirements above, the board-to-board circuitry for the probe-side electronics was made including a Teensy 4.1 (governing the the sensors and radio), voltage regulators and transistors (controlling the flow of power to 5V components and Teensy), sensors (including a GPS, pressure/temp, accelerometer/gyro), camera(s), audio beacon, XBee radio (with built in whip antenna), battery pack, power indicator LED, and a master power switch.

For testing and prototyping purposes, this circuit was first made using breadboards, forming a “flat-sat”. For the final flight model, a custom board-to-board PCB was designed from scratch using KiCad, and custom manufactured to fit the probe. The PCB comprises two layers, controlling the flow of power of the entirety of the probe-side electrical power subsystem and sensor subsystem.



***Figure 4.3 PCB Diagram***



***Figure 4.4 Assembled PCB***

Due to a lack of proper current to the camera it was not able to function leading to requirement SN7 being failed.

# **Causation for Failures**

## **MEC**

Subsystem failure was caused by a failure to test the parachute deployment system. During descent, the CanSat was performing nominally with the aerobrake deploying, slowing down the CanSat to an acceptable rate (10m/s to 30m/s) as stated in the competition guidelines. At an altitude of 100 meters, the CanSat failed to deploy its parachute and slow its descent rate and hit the ground at a velocity at around 27m/s, shattering the CanSate and destroying the egg in the process. It is believed that the parachute did not deploy because of lack of adequate spring force pushing the parachute up, a small amount of air traveling through the CanSat to eject the parachute, and the fact that such a large parachute was crammed into a tiny parachute cup.

The team attributed this failure to a lack of overall testing, especially with regards to the parachute deployment. Mechanical testing only began a week before leaving for Virginia and the lack of testing led to quick redesigns of the aerobrake release mechanism and the parachute deployment mechanism. The parachute deployment mechanism itself was redesigned the night before competition to make the parachute cup longer to incorporate longer springs to push the parachute out of the cup. The subteam expected the mechanism to work but never tested the mechanism to ensure that it would function nominally during launch. As a result, due to the lack of testing, the subteam was not able to predict that the parachute would not deploy from the CanSat to slow it down and protect the egg.

The subteam also attributed this to a lack of prototyping. When designing the CanSat, the MEC subteam designed the parts in CAD in as working concepts that had their theory down to the wire. However, because a lot of testing was done in the week leading up to the competition, some key parts such as the aerobrake deployment mechanism and the parachute deployment mechanism needed to be tweaked or completely redesigned. The subteam believes that there needs to be a designated prototyping phase for REV-3 and other projects in general to establish the viability of the concept they are working on.

## **SOFT**

The SOFT subteam mostly observed success, as the team was able to achieve all of our key requirements during testing, integration, and flight. However, there were two unexpected events during flight that we attributed to mistakes in the flight software.

First, the CanSat unexpectedly switched directly from the UDSC (Upper Descent) stage to the LAND (Landing) state, completely skipping over the LDSC (Lower Descent) state. The flight software used linear pressure differentials to denote spikes in acceleration and the direction the CanSat was traveling in, and the state change conditions for each relevant state in this scenario were as follows:

1. UDSC > LDSC: altitude must be below 100 m.
2. LDSC > LAND: pressure differential must be < 20 Pa/s and > -20 Pa/s.

Upon closer inspection of the CSV file, the change in pressure between packets 105/106 was observed at ~0.29 hPa, but there is a strong possibility that a combination of precision errors through the sensors and error propagations through unit conversions may have prematurely triggered the landing state. This anomaly was mostly inconsequential, however, since the servos are presumed to have turned. The parachute servo was discovered in the correct post-turn location but a failsafe was not implemented for this functionality, which presumably concludes that all functions of the LDSC state were deployed, but this change was not reflected in the output sheet since the state changed to LAND almost immediately.

| Packet # | State | Pressure (hPa) | Altitude (m) |
| --- | --- | --- | --- |
| 104 | UDSC | 90.98 | 115.73 |
| 105 | UDSC | 91.19 | 96.5 |
| 106 | LAND | 91.48 | 68.27 |

Second, we observed the beacon was activated prematurely on the launchpad. We attributed this event to the time-based failsafes present in our code, since the beacon was primed to turn on 120 seconds after receiving power. This problem was resolved when we remotely reset our system while on the launchpad, resetting the timer to 0 s immediately before launch.

## **EPS**

The main Camera did not record any video. It lacked the required current to properly function. The camera and EPS’s functionality were tested and confirmed prior to travel, however, it is believed that damage to either the camera’s voltage regulator or transistor caused a decrease in power. This could have possibly been prevented through more thorough testing to accurately determine the margin of safety for component powers. Other than the issue with the camera the EPS subsystem performed as intended.

## **Systems Integration and Testing**

System testing is an important part of any engineering project. REV-2 had planned on performing system level testing before leaving for Virginia. Due to the issues stated in section 6.1 and the fact some of the mechanical components had to be redesigned within only a few days of leaving this testing was not able to be performed until only minutes before launch. If more time had been taken for system level testing the flaw with the parachute ejection system would have been caught and could have been redesigned.

# **Human Factors**

## **Breaks and End of School**

As with any school project there are breaks dispersed throughout the year. For larger projects this tends to not be too large of an issue however, for a smaller shorter timeline project like REV-2 this can be much larger issue. It was expected that Christmas break, spring break, and the end of school/finals would be times where many of the team's members would not be available and work would not be progressing as fast. There is no helping that finals will take up time and most people leave to go on trips after school ends. With REV-2 this led to a lot of crunch time and work for a week leading up to the competition. It also hampered REV-2’s environmental testing score because only a few team members were available and had to leave for other important events soon after which rushed the testing process.

# **Lessons Learned**

While the REV-2 system ultimately did not achieve full mission success, the team identified several areas for improvement. These lessons learned are valuable both to the REV-2 team and to future teams trying to achieve similar goals.

## **Earlier Integration**

Integration didn’t start until the end of March and finished a partially integrated system in April. Because of finals and general school work, integration slowed down, not given a lot of time for testing and any bug fixes that needed to happen for successful integration. It is believed that with a stricter go-no go schedule which had integration earlier in the schedule, there would have been more time for testing and ensuring a proper integration.

## **Inter-subteam Communication**

There was not sufficient communication between the subteams. This led to many challenges when working on the REV-2 system. Oftentimes, teams would have to design components around one another and adapt to the decisions made by one subteam. It also led to some problems with integrating certain components into the system. It is important for future teams to have better inter-subteam communication, especially during specific sections of the project that will require input from multiple different subteams. This could be implemented with design review meetings where all the subteams get together to discuss integration and make sure there is proper communication between all members of the team. Furthermore, there should be a less defining line between EPS and SOFT because of how interdependent their tasks are. They ought to perform trade studies together and work together during testing and integration.

## **Earlier Testing**

A lot of the testing for the REV-2 system happened in May during the summer break, where many of the team members were out of town. As a result, the testing was only conducted by a couple of members and only revealed certain flaws in the design and parts a week before the competition. During this week, some parts had to be redesigned and it was determined that some parts wouldn’t be used on the final system. Additionally, because of this delayed testing, not every component was subjected to thorough testing. Compounding this was the excessive assembly time which made physical changes a very slow process. This led to the eventual failure of the parachute deployment mechanism and the partial success of the CanSat. In addition, a battery drain test ought to be performed to ensure that the batteries will provide sufficient voltage and current throughout the mission.

## **Changed Go/No-Go Schedule**

Following the competition, the team discussed the idea of updating the Go No-Go schedule for future CanSat projects. After the competition, the team discussed the idea of having a stricter go/no-go schedule. Some initial ideas talked about specific meetings for the whole team to discuss integration during trade studies at the beginning of the project. Another idea the team discussed is a specific prototyping phase or a proof of concept phase that would confirm whatever ideas discussed in the trade studies would be viable concepts that would lead to a full mission success of the system. Some other ideas the team discussed are earlier integration and earlier testing to account for any problems that often occur during these phases and a proper phase dedicated to field testing. Furthermore, EPS deadlines should be ahead of SOFT deadlines because hardware is required to properly develop and test software.

# **Conclusion**

REV-2 achieving partial mission success is a large step up from REV-1 in the previous year. The Go No-Go schedule contributed a significant amount to this success. The team agrees that a Go No-Go schedule should be used in the future but with revisions. The failure that caused REV-2 to not achieve full mission success could have been fixed with better and sooner testing and integration. It can not be stressed how much that if testing had been performed earlier the system would have been much more successful. It is believed that if future projects learn from the shortcomings of REV-2 that they will continue to improve on placement and score in the CanSat Competition.

# 

# **References**

[1] CanSat 2024 Mission Guide Revision E

[2] REV-2 Go No-Go Schedule