Fault Management in Small Satellites

Lessons Learned from the Lunar Flashlight and ARMADILLO Missions

Dillan McDonald and E. Glenn Ligthtsey[1] [2]
Georgia Institute of Technology, Atlanta, GA, 30318

As spacecraft miniaturize, the implementation of an appropriate fault management system is increasingly important. Through experiences testing NASA's Jet Propulsion Laboratory's (JPL) Lunar Flashlight fault protection system and working closely with ARMADILLO's flight software implementation and other LEO cube-satellites, a set of recommendations has been formed. The recommendations provided specifically cover Under-Voltage Lock Out (UVLO) implementation, spacecraft error handling to help restore functionality, a simplified spacecraft mode state diagram, fault interaction rules, and a general architecture for fault protection implementation. Additionally, a fault hazard analysis was conducted to evaluate points of failure for the Georgia Tech Mission Operations Center (MOC), and finally a discussion on ground station development and the role the GT Ground Station Network (GSN) plays in mission operations.

Space Systems
Design Laboratory

1 2

[1] Graduate Research Assistant, Daniel Guggenheim School of Aerospace Engineering, dillanm@gatech.edu
[2] Professor, Daniel Guggenheim School of Aerospace Engineering, glenn.lightsey@gatech.edu

Table of Contents

1 Introduction

As small spacecraft get smaller, small satellite reliability issues become of greater importance. Small satellite programs historically cut corners that a typical larger scale space missions would implement. Redundancies and conservative design choices are removed in the interest of miniaturizing systems and reducing system cost. This extremely on-edge design approach promises significant returns on investment at a gamble. The result is the historically high failure rate of these systems, dominated by infant mortality.

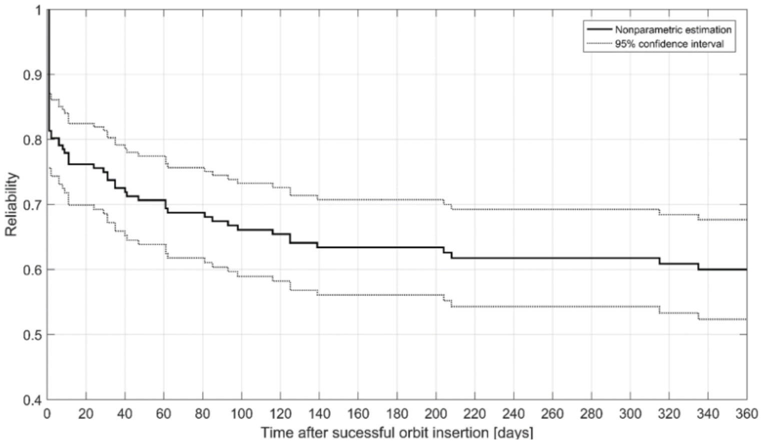


Figure 1: CubeSat Reliability with 95% Confidence Interval – First Year in Orbit [3]

This critical small satellite infant mortality rate can be attributed to a number of causes. Small satellites are typically secondary payloads ride-sharing on launch vehicles, and therefore suffer tight schedules and deadlines. These tight deadlines can be challenging when paired with inconsistencies in documentation standards, procedures, experience throughout the team, and rapid personnel turnover. With so much focus on delivering a satellite that may or may not work, there is typically little time to consider risk analysis/fault management and/or the ground station infrastructure that is necessary for these missions to be successful. Unfortunately, not assigning the appropriate resources to these typically overlooked components of the mission design can compound the issues faced at deployment.

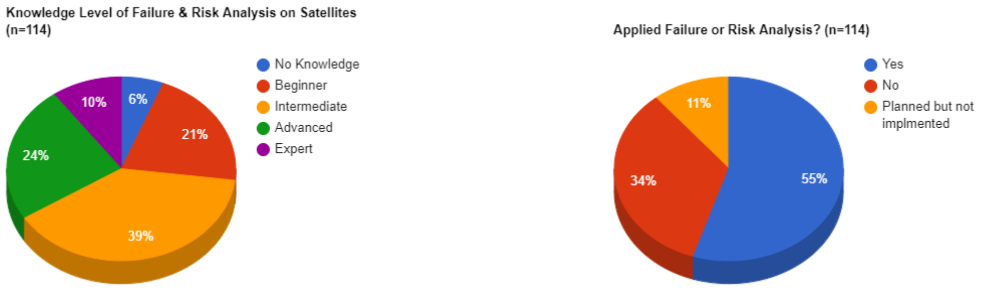


Figure 2: (A) Survey Results on Knowledge Level of Risk Analysis and Failure Analysis on Satellites [3], (B) Survey Results on the Implementation of Risk Analysis Within Their Cubesat Program [3]

To ensure that a given small satellite maintains health within the adverse environment that it operates, the implementation of fault management is imperative. With appropriate implementation, a spacecraft can manage a given condition via functional and hardware redundancies or fault protection techniques. The latter is the most logical approach for implementation in such a constrained form factor. A robust fault management infrastructure for small satellites, specifically fault protection technique implementation, is essential to improve the rate of mission success.

The goal of this paper is to provide an overview of the fault protection implementation used on the Lunar Flashlight and ARMADILLO spacecraft, notably two different missions with vastly different requirements, review the Lunar Flashlight fault protection testing campaign, and provide recommendations for future missions. The GT Ground Station Network and the Mission Operations Center play a key role in each mission's success, and therefore are also included in this discussion. This documents should serve as a reference for fault protection implementation in future GT missions.

2 Lunar Flashlight Overview

Lunar Flashlight (LF) is an interplanetary mission for the Small Spacecraft Technology program under NASA's Space Technology Mission Directorate (STMD). Lunar Flashlight is a 6U (12 x 24 x 36 cm) CubeSat developed and managed by the Jet Propulsion Laboratory (JPL) that intends to fly to the moon and insert itself into a Near-Rectilinear Halo Orbit (NRHO). While in orbit around the moon, Lunar Flashlight will use active laser spectroscopy to search for water ice and prove the capability of performing planetary science investigation in the CubeSat form factor.

Lunar Flashlight's Mission objectives include the demonstration of a novel green propellant small satellite propulsion system developed at Georgia Tech, as well as the investigation of permanently shadowed regions (PSRs) at the lunar south pole. These PSRs act as cold traps and can house volatiles for potentially billions of years. The Clementine, Lunar Prospector, and Lunar Reconnaissance Orbiter (LRO) missions made observations consistent with ice deposits centimeters-to-meters deep in PSRs that motivate the search for water ice [1].

Georgia Tech's role in Lunar Flashlight is the development of the propulsion system, integration and testing of the spacecraft, and operation of the spacecraft through its lifetime. To date, Lunar Flashlight has completed integration and testing, and Georgia Tech is to preparing for operations. Lunar Flashlight is scheduled to launch in November 2022 aboard a Falcon-9 with the ISpace Hakuro-R Lander.

2.1 CONOPS

Lunar Flashlight's mission consists of five different phases: Launch and Early Operations (LEOP), Cruise, Approach, Science, and End of Life (Deorbit). Figure 3 provides a high level overview of the mission concept of operations. Launch and Early Operations consists of deployment, initial checkout, fuel priming and conditioning, momentum wheel desaturation, and the first set of Trajectory Control Maneuvers (TCMs). Lunar Flashlight then spends two months in cruise, where it will use a low-thrust trajectory about the L2 Lagrange point to put the satellite on course for Lunar Orbit Insertion (LOI). During the approach phase, contacts become more frequent and more ranging is done to ensure that LF is prepared to perform LOI.

Once the satellite has performed LOI and performs its first orbit of the moon, the science phase begins. The Science phase consists of 10 orbits with a period of approximately 6 days. Within each orbit, there are three orbital trim maneuvers (OTMs) that ensure the spacecraft can remain in the NRHO. For each perilune pass, laser instrument is fired in a defined sequence from the science team, and throughout the rest of the orbit the laser battery is charged. Finally, after 60 days, the satellite will preform a deorbit maneuver and impact into the moon's surface.

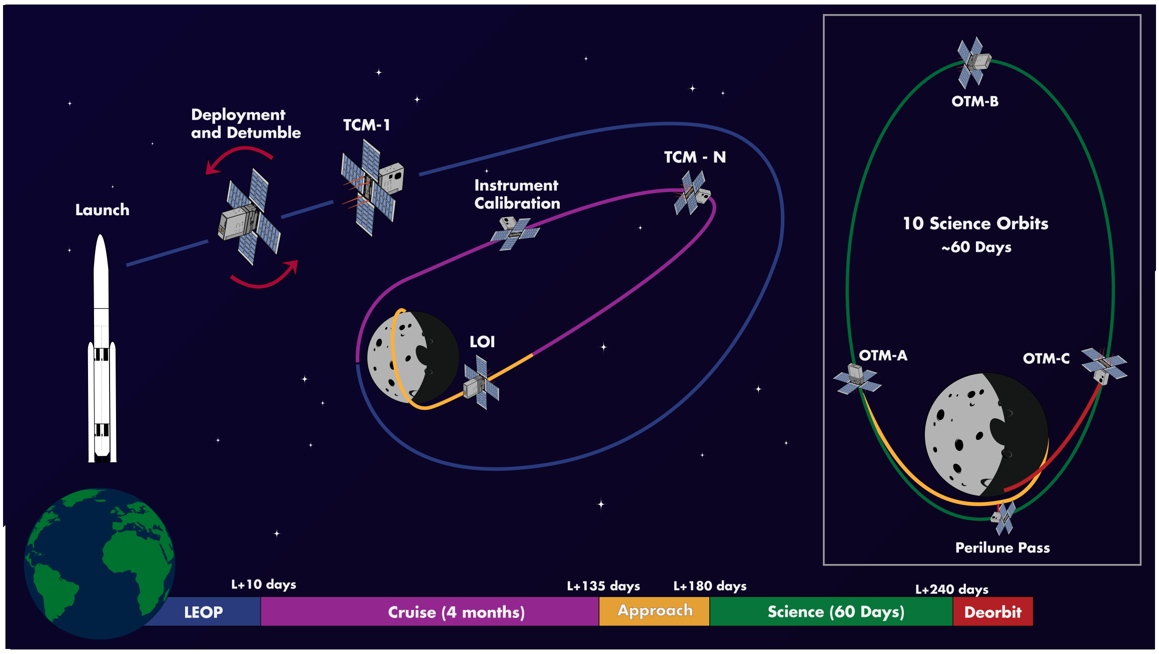


Figure 3: Lunar Flashlight Concept of Operations

2.2 Satellite

The Lunar Flashlight satellite was designed at JPL, and shares many common components with its sister mission NEAScout. The specific subsystem breakdown is shown in Figure 4. For the core bus of the spacecraft, Lunar Flashlight uses a radiation hardened in-house command and data handling board called the Sphinx. The radio in use is the Iris radio v2.1, a revision of the same radio flow on the two deep space MarCO CubeSats. Lunar Flashlight uses the XACT-50 attitude control system from Blue Canyon Technologies, also a revision of the same attitude control system series flown on MarCO.

Lunar Flashlight's propulsion system was developed by Marshall Space Flight Center and Georgia Tech and uses a novel green propellant called ASCENT. The thrusters were developed by Plasma Processes Inc. The payload was developed at JPL, and consists of a laser power card, laser projector assembly, and reflectometer. Lunar Flashlight houses two power systems: one for the spacecraft bus that includes the power card, lithium ion battery, and four solar panels, and one for the payload specific power card and battery that can be charged and operated independently. This implementation is due to the large power consumption of the laser projector assembly during the perilune passes.

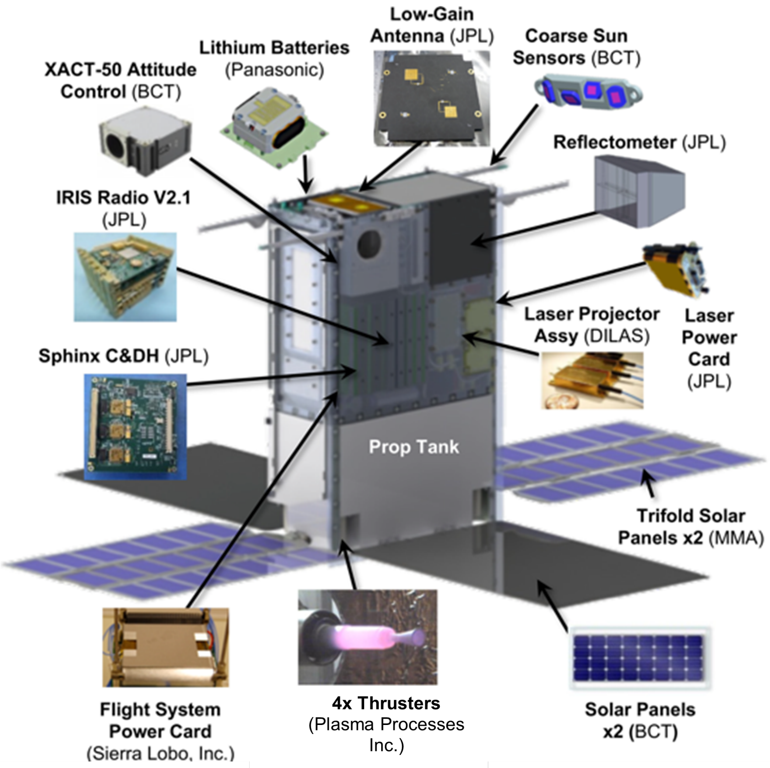

XACT-50 Attitude Control (BCT)
Lithium Batteries (Panasonic)
Low-Gain Antenna (JPL)
Coarse Sun Sensors (BCT)
Reflectometer (JPL)
IRIS Radio V2.1 (JPL)
Laser Projector Assy (DILAS)
Laser Power Card (JPL)
Sphinx C&DH (JPL)
Prop Tank
Trifold Solar Panels x2 (MMA)
Flight System Power Card (Sierra Lobo, Inc.)
4x Thrusters (Plasma Processes Inc.)
Solar Panels x2 (BCT)

Figure 4: Lunar Flashlight Spacecraft Overview

Georgia Tech was also responsible for the integration and testing of the spacecraft. This enabled the operations team to participate in the I&T testing campaigns, and provided a deeper understanding of the spacecraft that would have otherwise not been possible. The operations team specifically owned two of the tests in the I&T campaign: Day In The Life (DITL) testing and Fault Protection (FP) testing. These testing campaigns had a large impact on the development of the mission operations procedures and flight rules.

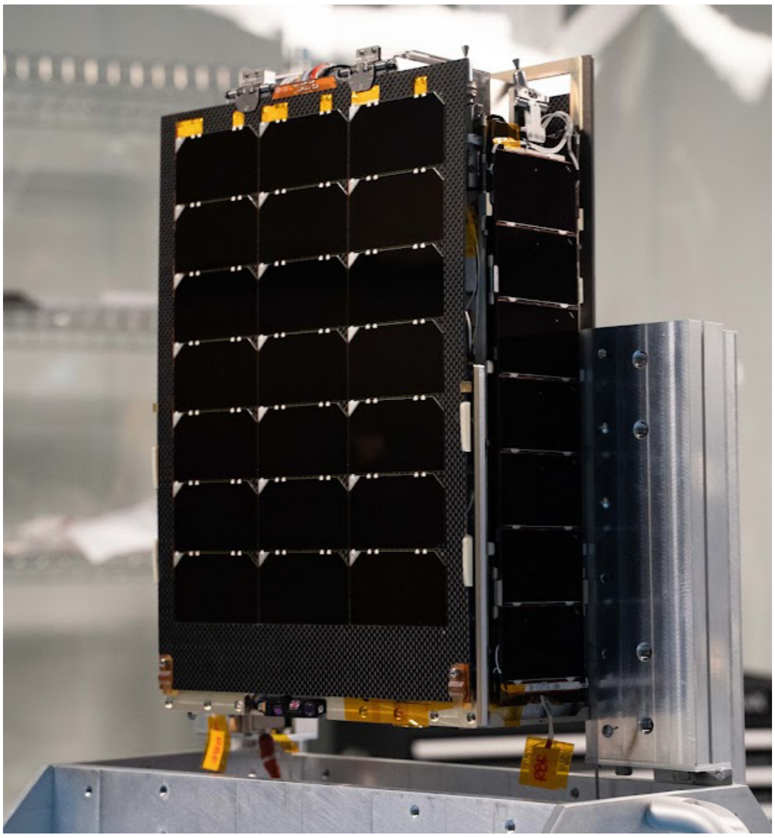


Figure 5: Lunar Flashlight Spacecraft in Vertical Fixture

2.3 GT MOC

The Mission Operations Center (MOC) at Georgia Tech was originally developed to be a multi-mission operations center to support CubeSat missions in Low Earth Orbit. The Lunar Flashlight mission Mission Operations System and Ground Data System (MOS-GDS) levied requirements to enhance the GT MOC capabilities and support Deep Space Network (DSN) access for commanding and telemetry reception in lunar orbit. On this mission, the MOC also acts as an interface for the Science Operations Center (SOC) located at UCLA, the propulsion team at Marshall Space Flight Center, and the Engineering support team and Mission Design and Navigation (MDNAV) group at JPL.

Figure 6: Georgia Tech Mission Operations Center

The GT MOC is a complex system in itself, and its data flow is shown in Figure 7 [5].

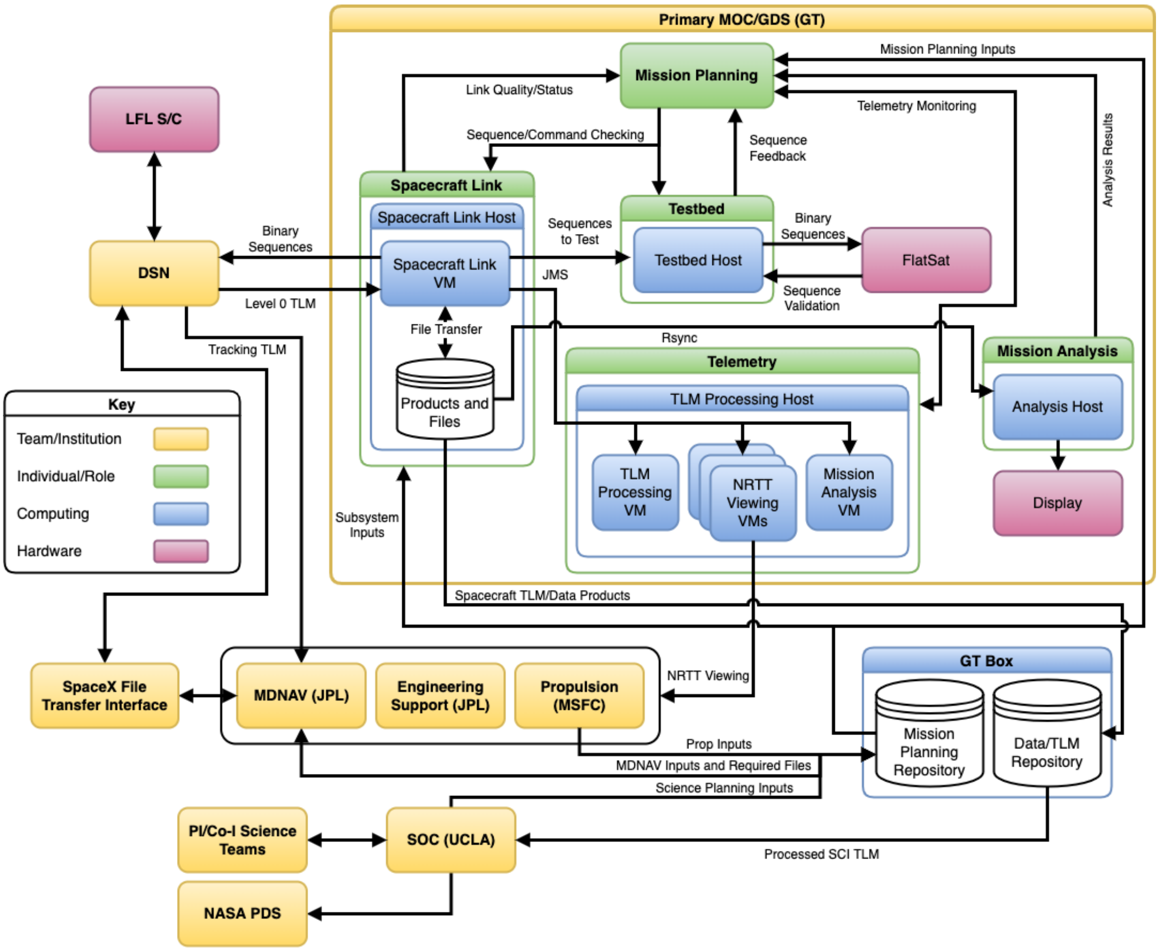


Figure 7: Georgia Tech MOC Lunar Flashlight Ops Data Flow

The MOC consists of four major hosts for operations. The spacecraft link host provides the connection between the DSN and the MOC. This host holds the commanding capability, and re-directs the telemetry and data products to their respective hosts or databases. The spacecraft link host is the single most critical host in the MOC for tactical mission operations. Following the spacecraft link host, the telemetry host houses the telemetry viewing virtual machines (VM). The telemetry host provides an access point for the different parties to view telemetry while in a live contact. The telemetry host also includes a telemetry processing VM that is used for the spacecraft health checkout in every contact.The telemetry host also has access to the spacecraft power profiles, attitude analysis tools for keep-out zones, and other spacecraft specific analysis tooling.

The testbed host is connected to the spacecraft testbed, and is used to validate sequences for execution. The testbed is also utilized in the Operational Readiness Testing campaign as a flight unit analog. This host does not typically need to be staffed in tactical spacecraft operations, and is primarily used in strategic periods for sequence verification and validation. Finally the mission analysis host contains a deployment of OpenMCT, a software tool which is used for historical telemetry visualization. The goal mission analysis host to provide a common access point for the historical data throughout the flight.

The operational roles in the GT MOC data flow are defined as follows:

2.3.1 Mission Planner

The Mission Planner (MP) acts as the operations chief for a given contact. They own the schedule of activities for that contact, and evaluate the information across the roles to determine the plan of action. They give the final go/no-go calls for the Spacecraft Link Engineer (SLE) for command execution, and act as the DSN interface within a given contact.

2.3.2 Spacecraft Link Engineer

The Spacecraft Link Engineer (SLE) is responsible for performing all spacecraft commanding within a given pass, and works directly with the mission planner to orchestrate the spacecraft operation. The SLE is also responsible for monitoring the link with the DSN and identifying immediate issues with the telemetry via the AMPCS alarm system.

2.3.3 Telemetry Engineer

The Telemetry Engineer (TLM) is responsible for interfacing with external parties and providing the appropriate telemetry displays. They are also responsible for performing a thorough spacecraft health checkout while the SLE is executing the procedure for a given contact. They monitor the spacecraft's pulse throughout the contact. They are also required to have a general understanding of the Power Equipment List (PEL) and TBall: the attitude analysis tool to check for Keep Out Zones (KOZ).

2.3.4 Testbed Engineer

The Testbed Engineer (TEST) is responsible for maintaining a testbed state that is reflective of the flight unit state. They are also responsible for testing the sequences generated for a contact on the testbed and troubleshooting any issues that arise.

2.3.5 Mission Analyst

Th Mission Analyst (MA) is responsible for responding to any issues that require deeper analysis. They monitor the Open-MCT display for additional situational awareness. Mission Analyst is not a role that will be expressly staffed in tactical operations.

3 ARMADILLO Overview

ARMADILLO, Attitude Related Maneuvers And Debris Instrument in Low (L) Orbit, is a 3-Unit CubeSat developed at the University of Texas at Austin (UT-Austin) that was designed to detect sub-millimeter space debris via a Piezo Dust Detector (PDD) developed by the Center for Astrophysics, Space Physics and Engineering Research (CASPER) at Baylor University. Additionally, ARMADILLO is equipped to perform radio occultation measurements via a software-defined FOTON GPS receiver developed by the Radionavigation Laboratory at UT-Austin. [4]

ARMADILLO is one of three missions from the Texas Spacecraft Laboratory (TSL) to use a modular and reusable CubeSat Bus, alongside BEVO-2 and RACE, which were built at approximately the same time in the TSL. ARMADILLO was selected for flight as part of the UNP-7 cohort in 2014, and was launched on June 25th, 2019 on the Falcon Heavy STP-2 launch.[4]

The specific mission objectives were to

1. Characterize in-situ sub-millimeter level dust and debris particles in LEO by sensing impacts at varying times, directions and locations.
2. Demonstrate ionospheric radio-occultation within a single CubeSat volume (10 cm x 10 cm x 10 cm) using a software-defined dual frequency GPS receiver.

3.1 CONOPS

The ARMADILLO mission followed a typical LEO CubeSat Concept of Operations shown in Figure 8. First the spacecraft was launched and deployed. Once deployed, ARMADILLO would perform detumble and first contact with the ground control. Once ground control was established, it would then go into initial checkout to ensure the spacecraft subsystems survived launch. After checkout, ARMADILLO could then perform science operations interspersed with data downlink to the ground for the rest of its lifetime.

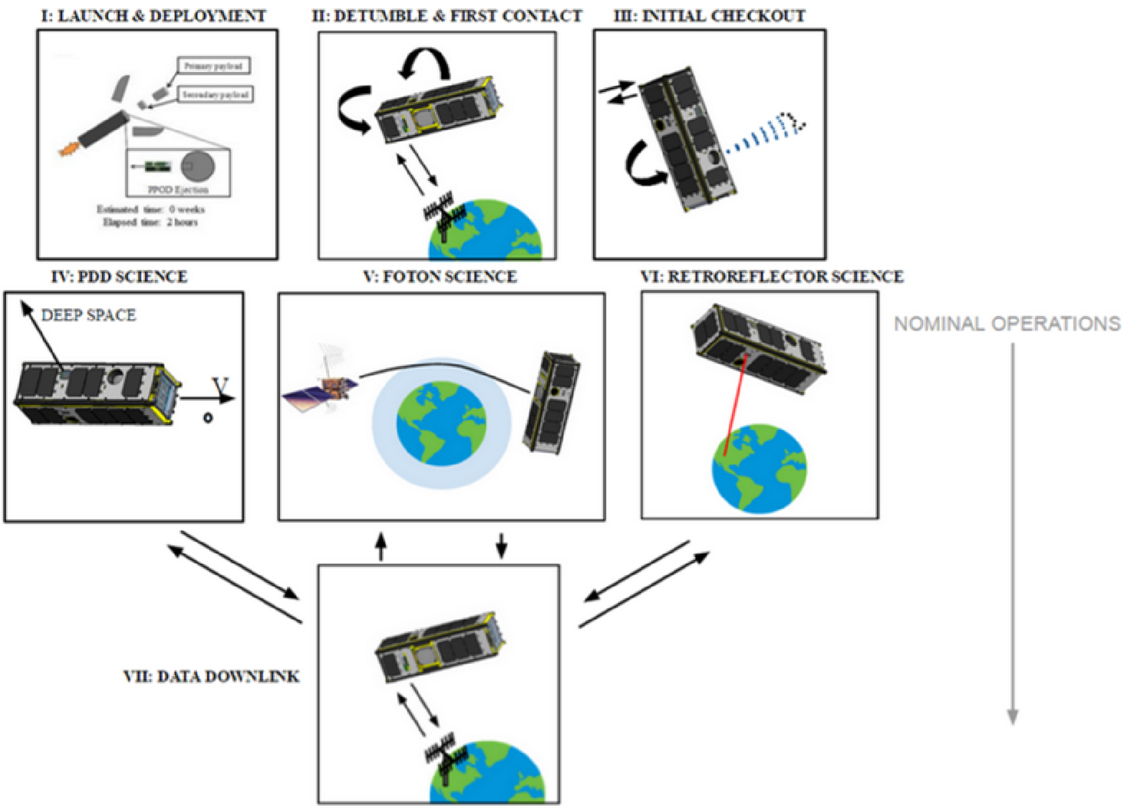


Figure 8: ARMADILLO Concept of Operations

3.2 Satellite

ARMADILLO benefited from a modular CubeSat bus design developed at the Texas Spacecraft Laboratory (TSL). This enabled a common bus to be evaluated and re-used among three different missions: Bevo-2, RACE, and ARMADILLO. The Bus module itself consisted of a 3U EPS stack, the UHF/VHF antenna, the CDH computer, and the Astrodev Helium radio. The ARMADILLO Attitude Determination and Control (ADC) module consisted of a MEMS gyro, Sun sensors, a magnetometer, three reaction wheels, three magnetorquers, and the L1/L2 GPS antenna. Finally the Payload Module consisted of the FOTON GPS, an in-house developed star tracker, and the Nine-element Piezoelectric Dust Detector (PDD).

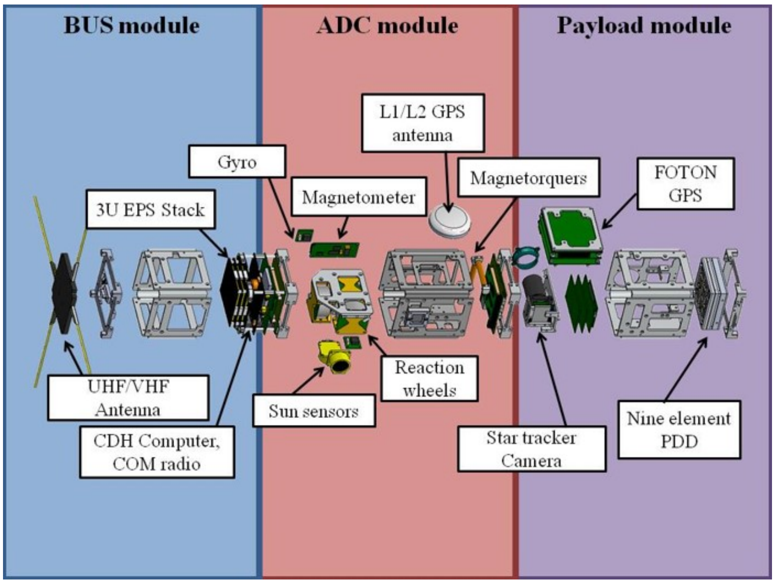


Figure 9: ARMADILLO Spacecraft Overview

The ARMADILLO CubeSat was integrated and eventually launched on the Falcon Heavy STP-2 mission on June 25th 2019. The mission was not initially successful, and did not establish contact until almost three years later on January 29th 2022. ARMADILLO completed a partial checkout before deorbiting on September 23rd 2022.

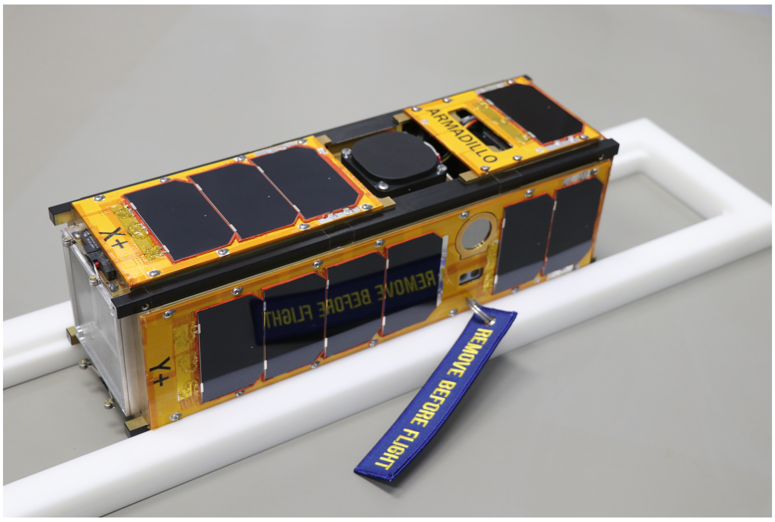


Figure 10: ARMADILLO Final Integration

3.3 GT GSN

The Georgia Tech Ground Station Network (GT GSN) is a network of 4 stations that are in use for spacecraft operations at GT. Each of the four stations have varying capabilities and architectures. The current ground station infrastructure is such that each station operates independently with individual configuration files that define which spacecraft to track. Each independent station follows the general tracking flow show in Figure 11.

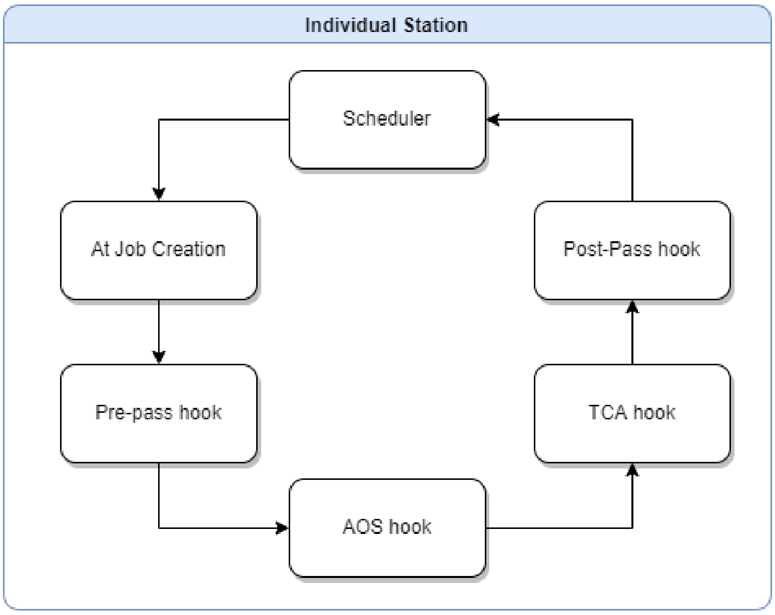
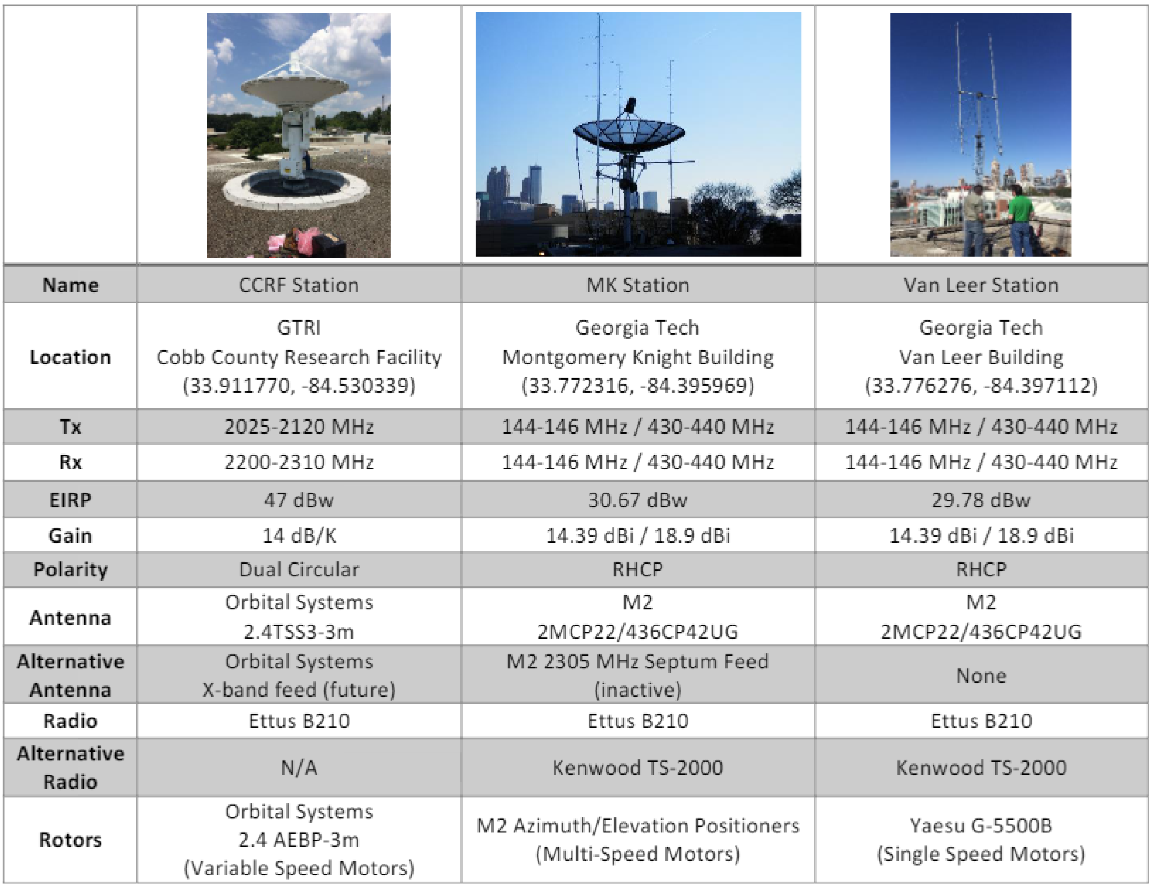


Figure 11: Individual Station Tracking Flow

The scheduler is called every 8 hours. It pulls from the configuration file with the Spacecraft NORAD IDs in priority order, and creates a scheduled job for each pass. At 15 minutes before the start of each pass, the pre-pass hook is called. This hook builds the Keplerian elements for the Doppler adjustment script, gets the rotator's values, performs disk management, and spools up the radio flow-graphs for that particular satellite. Once the satellite reaches the horizon, the Acquisition of Signal (AOS) hook is called. The AOS hook begins running any kind of commanding software necessary for that particular mission, and begins transmitting. At the Time of Closest Approach (TCA), the TCA hook is called. This is typically used in the case where the Doppler implementation is not adequate, or the link is not particularly strong. Finally at the end of the pass, the Loss of Signal (LOS) hook is called and spawns the mission specific post-pass processes and ends the cycle. ARMADILLO primarily used the Montgomery Knight and Van Leer ground stations at GT. The Specifications for three of the four stations are given in Figure 12.



Name	CCRF Station	MK Station	Van Leer Station
Location	GTRI Cobb County Research Facility (33.911770, -84.530339)	Georgia Tech Montgomery Knight Building (33.772316, -84.395969)	Georgia Tech Van Leer Building (33.776276, -84.397112)
Tx	2025-2120 MHz	144-146 MHz / 430-440 MHz	144-146 MHz / 430-440 MHz
Rx	2200-2310 MHz	144-146 MHz / 430-440 MHz	144-146 MHz / 430-440 MHz
EIRP	47 dBw	30.67 dBw	29.78 dBw
Gain	14 dB/K	14.39 dBi / 18.9 dBi	14.39 dBi / 18.9 dBi
Polarity	Dual Circular	RHCP	RHCP
Antenna	Orbital Systems 2.4TSS3-3m	M2 2MCP22/436CP42UG	M2 2MCP22/436CP42UG
Alternative Antenna	Orbital Systems X-band feed (future)	M2 2305 MHz Septum Feed (inactive)	None
Radio	Ettus B210	Ettus B210	Ettus B210
Alternative Radio	N/A	Kenwood TS-2000	Kenwood TS-2000
Rotors	Orbital Systems 2.4 AEBP-3m (Variable Speed Motors)	M2 Azimuth/Elevation Positioners (Multi-Speed Motors)	Yaesu G-5500B (Single Speed Motors)

Figure 12: GT GSN Specifications

4 Fault Management

Fault Management is defined as the functional capabilities distributed throughout the observatory and ground elements that enable detection, isolation, and recovery from events that upset nominal operations. The goal of the fault management system is to achieve mission reliability objectives within program resources. Fault management must achieve this goal by balancing project risk and the cost of developing, testing, and operating the fault management system [2].

To ensure that a spacecraft operates nominally, all internal and external influences must be monitored and characterized. This is a monumental task, and can consume significant resources. However, if ignored, it can result in stunted spacecraft performance or spacecraft mortality. External influences can be anything from solar related thermal overruns, to radiation based damage to the spacecraft. Internal influences can be poor management of subsystems resulting in spacecraft under-voltage, to inducing electrical cross-talk in a poorly designed subsystem by enabling multiple functions.

Fault management can also capture the environmental considerations in the development of a spacecraft, like Electro-Static Discharge (ESD) events, or other mistreatment of hardware that can result in component failure during flight. These types of failures can have catastrophic implications and render components either partially or completely unusable. Failures may also arise based on operator error, and improper commanding of the spacecraft.

4.1 Classification of Faults

The Lunar Flashlight mission classified faults into different categories for fault protection testing. These categories are illustrated in Figure 13. Category 1 faults cover faults that have particular flight software responses in addition to an Event Verification Record (EVR). These can be split into category 1A faults, faults that can have failure conditions that can be created, and category 1B faults, faults that have scenarios that cannot be created. Similarly, Category 2 faults are faults that have flight software indications but no associated responses. These faults get separated into category 2A, faults that expressly have EVRs associated with them, and category 2B, faults that have to be discerned from telemetry. Finally there are category 3 faults, faults that have no FSW indication.

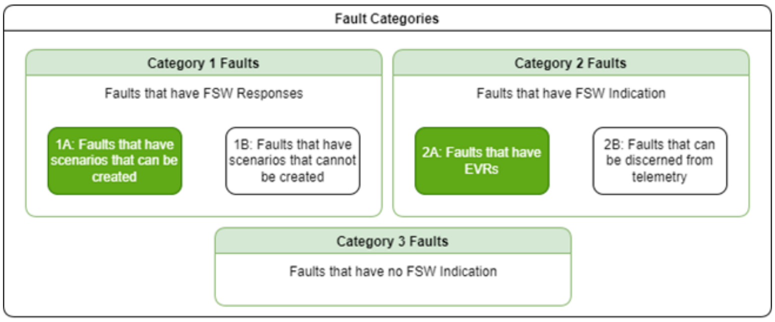

Fault Categories
Category 1 Faults
Faults that have FSW Responses
1A: Faults that have scenarios that can be created
1B: Faults that have scenarios that cannot be created
Category 2 Faults
Faults that have FSW Indication
2A: Faults that have EVRs
2B: Faults that can be discerned from telemetry
Category 3 Faults
Faults that have no FSW Indication

Figure 13: Fault Category Definitions

These categories were used to identify the most critical faults to validate in the Lunar Flashlight Fault Protection testing campaign. Category 1 faults are the most critical because they have autonomous responses, and therefore need thorough testing to ensure correct operation. Following that, Category 2A faults were important to verify as they provide critical insight into multi-point spacecraft failures. These categories are expressly focused on the Flight Software (FSW), and many hardware faults that are of high criticality fall under category 3 and category 2B faults.

4.2 LF Fault Protection Implementation

The Lunar Flashlight flight software was developed concurrently with the NEA Scout Mission.

> Flight software for both CubeSat missions is based on the open-source F Prime Flight Software Product Line developed by JPL. F Prime utilizes a reusable component-based architecture with typed ports that can be interconnected to form a topology. Also, F Prime includes a set of auto-coding tools used to generate components and topologies that can be deployed for various mission specific applications [7].

The Lunar Flashlight F Prime Architecture can be seen in Figure 14.

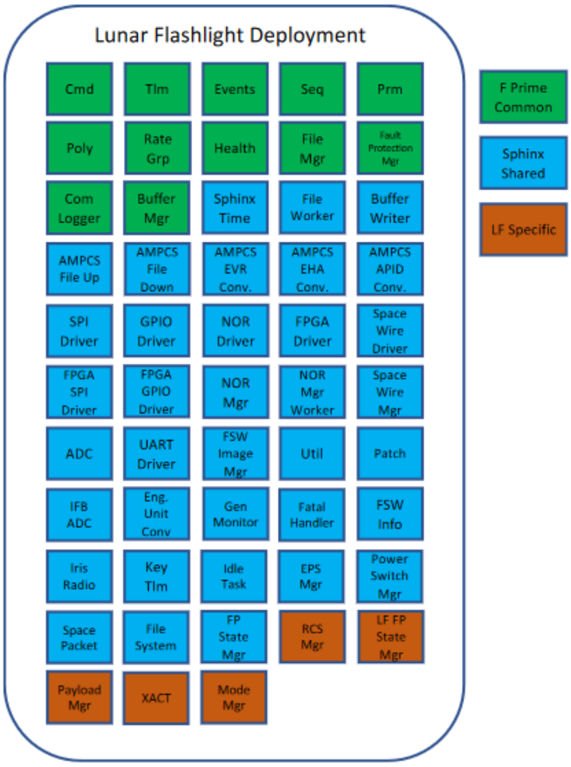


Figure 14: Lunar Flashlight F Prime Depoyment Components [7]

The Lunar Flashlight spacecraft has a fault management system that automatically identifies and responds to onboard faults. The implementation is illustrated in Figure 15.

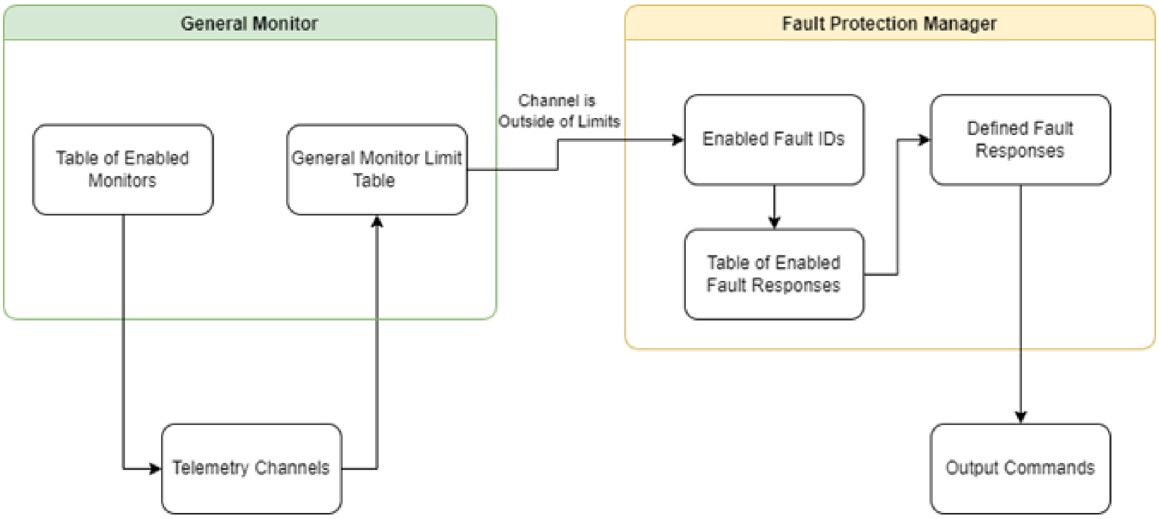

General Monitor
Table of Enabled Monitors
General Monitor Limit Table
Channel is Outside of Limits
Fault Protection Manager
Enabled Fault IDs
Defined Fault Responses
Table of Enabled Fault Responses
Telemetry Channels
Output Commands

Figure 15: Lunar Flashlight Fault Protection Implementation

First the general monitor (GenMon) process checks the table of enabled monitors. It then reads from the appropriate telemetry channels and checks the current telemetry values against the limits assigned in the general monitor limit table. If any of those channels are outside of the limits and have met a fault condition, then GenMon outputs an EVR and sends the channel in violation to the Fault Protection Manager (FPM).

An EVR will be output regardless of the fault response status, but if the fault ID is enabled, the response is requested. On Lunar Flashlight there is only one response, 0x0. 0x0 either enables all fault responses or disables all fault responses. If the fault response is enabled, the fault response function for the particular fault ID will be called and will output the appropriate commands. If the fault responses are disabled this allows the spacecraft to save the overall configuration of the enabled/disabled fault IDs. This implementation allows the safe mode to be a generic sequence, and the faults that require a transition to safe mode to execute a simple safe mode sequence execution command. The downside to this approach is the lack of branching logic and conditionals in the safe mode.

4.3 ARMADILLO Fault Protection Implementation

ARMADILLO, in comparison, had a very different fault protection implementation. ARMADILLO's FSW was developed from scratch, and had individual subsystem fault management. If a subsystem was enabled, the associated subsystem would have an aliveness check with either a subsystem watchdog or telemetry monitoring. If the subsystem did not respond, a subsystem power cycle was commanded. For thermal control, if ARMADILLO exceeded the expected thermal constraints, the spacecraft would be placed into safe mode until it recovered. The EPS on ARMADILLO had its own thermal control that would maintain the battery temperature within the margin for charging. If the temperature exceeded that margin, the EPS would not allow charging.

The main fault protection implementation on ARMADILLO was battery monitoring. The flow of this control is shown in Figure 16.

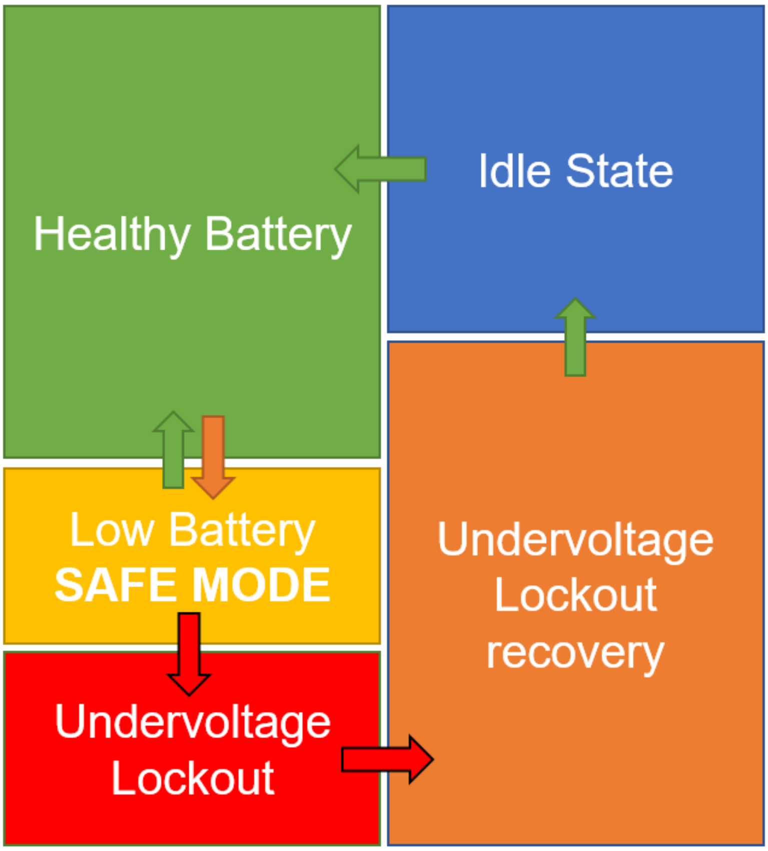


Figure 16: ARMADILLO Power Management

If the battery got below the healthy margin, the spacecraft entered safe mode. ARMADILLO's safe mode was defined to only have the CDH computer and radio beacon enabled. If the battery dropped lower than that into the Under-voltage Lockout limit defined by the EPS, the spacecraft would go into Under-voltage Lockout Recovery, which notably does not turn on the spacecraft until the battery voltage is within a healthy range. Once the spacecraft battery voltage has recovered to well within a healthy range, the spacecraft would transition into and idle state with a healthy battery. The idle state on ARMADILLO was defined the same as the safe mode, and therefore would only have the CDH computer and radio enabled.

5 LF Fault Protection Testing Campaign

The Lunar Flashlight spacecraft underwent fault protection testing from January 27th 2022 through February 25th 2022. The bulk of testing was performed on January 27th and 28th but due to some unexpected behaviors from the spacecraft, fault protection testing was re-run to close out some idiosyncrasies. In addition, the scope of the fault protection testing grew to include the under-voltage lock out (UVLO) test.

The procedure, immediate commands, and testing scripts were written and evaluated on the testbed. Once they were deemed appropriate, a procedure review was held to ensure the project had the appropriate visibility into the test, and to ensure there were no additionally incurred risks from running the test. Finally, once the procedure was approved the test was executed and a report was written to cover the results of the test. This document covers many of the results from that testing campaign.

The primary fault protection testing campaign followed the following testing. The particular detail about the flow can be found in Appendix A.

5.1 Flight Unit Testing

This subsection of testing was done on the flight unit directly. The faults that fell under category 1A were identified to be testable on the flight unit without inducing damage. The faults identified to be potentially harmful were tested via triggering the fault id directly instead of inducing the fault condition. If the fault was capable of being tested on the testbed, then it was.

5.1.1 Spacecraft Setup

This section was run anytime the testing spanned several days and if there was a required spacecraft power cycle. The procedure largely consisted of running the flight unit functional sections for spacecraft initialization and establishing contact with the spacecraft. Then, a safe mode test was conducted by triggering the safe mode without a sequence loaded. A safe mode sequence was then loaded, the spacecraft returned to normal mode, fault responses enabled, and then a fault ID with the safe mode sequence was triggered. This approach was used to verify the proper execution of the safe mode sequence.

5.1.2 Category 1 Fault Response Validation

The Category 1A faults were identified and reviewed by the subsystem experts to ensure minimal impact to the flight unit. The remaining faults were covered via commanding the fault ID and noting the spacecraft fault response. To test these faults first the safe mode sequence was removed to ensure that the spacecraft would not execute it. Then the appropriate sequences for the test were loaded, specifically the test interrupt sequence for this test. The test then went through each Category 1A fault and created the scenario for the fault trigger if possible. The XACT is the notable exception as it had unique tests per fault ID, and Gen Mon based faults were simple to test, as the range would be set out of bounds. Each test would enable Gen Mon, the fault ID, monitor for the fault trigger command execution, and then verify the fault response. Category 1B faults were tested by enabling the fault ID, triggering that fault ID, monitoring for the command execution, and verifying the fault response.

5.1.3 Category 2A Fault Response Validation

For Category 2A, the goal was to trigger the faults "naturally" as opposed to using the trigger command, as an exercise to the entire fault identification system. To do this, the general monitor ranges were shifted out of range so that GenMon would throw the fault ID. This approach was very similar to the testing done for Category 1A faults where the GenMon range was set out of bounds. GenMon was then enabled, the fault ID enabled, the EVRs monitored for the fault trigger, and finally the fault response verified.

5.1.4 Fault Persistence Test

Certain faults also have specific persistence responses. For example, if the payload sends one fault 10 times, then a particular response is used after the 10th reception of that fault. The Iris radio in particular will re-attempt radio reset every hour if it does not receive a heartbeat from it. This behavior required a unique test per persistent fault.

First, the safe mode sequence was removed to ensure it did not execute. Then the fault responses were enabled and the specific fault ID being tested were enabled. For each of these faults, a persistence script was written to send a reoccurring fault to the spacecraft. This script was then executed, and then the particular response was verified. This was repeated for the XACT, the payload, and the Iris.

5.1.5 Fault Clear Test for the XACT, Iris, and Payload

Some faults aboard Lunar Flashlight require a "Clear" fault to be sent to restore condition. For example, if the time and references aboard the XACT are invalid, then to resolve it the operator would set the time and references. This would in turn send a "Clear" fault that would resolve the fault flag, and allow the XACT to respond to other faults.

These faults were tested by enabling the fault responses. Enabling the particular fault IDs and triggering the fault ID. Once the trigger was commanded, the initial response was verified. Then to ensure that the fault had no persistence, the same fault ID was triggered once more, and was verified that there was no response from the spacecraft. Finally the clear fault would be sent, and the test was run again to verify that the fault response executed once the clear fault cleared the previous condition.

5.1.6 Iris Fire Code Test

The Iris fire code is essential, as it enables the mission operations center to command a spacecraft reboot without utilizing the FSW. In the critical case where the FSW is hung, and the spacecraft needs to be emergency rebooted, then the Iris fire code is the only way to recover the spacecraft.

To test this logic, the typical functional procedure was run for spacecraft initialization. This test required the use of CASSY, the DSN "Simulator" that enabled RF commanding and reception. The procedure then established lock with the spacecraft, and then would execute the fire code. Then the test verified that the spacecraft restarted as expected.

5.1.7 Flight Unit UVLO Test

One of the major action items that came out of the initial fault protection testing campaign was the evaluation of the spacecraft UVLO. There was serious concern that if the spacecraft had reached its UVLO limits that it would be stuck in a boot loop due to the large transients associated with the Iris radio and XACT systems powering on. This test consisted of discharging the spacecraft to near its UVLO limit and executing the safe mode sequence. This was an attempt to test the transients without putting the flight unit into the UVLO. Unfortunately, during testing, the flight unit immediately went into UVLO and the test was re-written to be run on the testbed, and additional characterization of the flight unit transients were done on start-up.

The testing for UVLO was done by first powering on the spacecraft. This test in particular required a power supply set to charge the spacecraft battery at the minimum solar panel charge rate. It was also executed over RF, as the intention was to characterize the Iris start-up transients, and therefore required CASSY to be set up. Once lock was established the testing safe mode was uploaded, and unwanted fault IDs were disabled, as Lunar Flashlight throttles the fault ID EVRs to 5 per second. The APIDs on the spacecraft were set to high to provide the maximum number of data points to evaluate what was happening on boot. Then the dangerous part began with the discharge of the battery to 9 volts. Once at 9 volts, the safe mode was executed, and then the execution was monitored. After the test the battery was returned to storage voltage, and the other testing hardware was closed out.

5.2 Testbed Testing

Testbed testing has its limitations, just as the flight unit does. Where the testing on the flight unit is limited by the increase in potential risk associated with the tests, the testbed testing is limited by the actual functionality of the testbed, and the testbed's representation of the flight unit. The Lunar Flashlight testbed in particular has many limitations, documented in our internal Testbed/Flight unit difference log, but the most crucial difference for testing include the lack of an Iris radio, an incomplete payload subsystem, and floating thermal sensors.

5.2.1 Testbed UVLO Test

The testbed UVLO test was an effort to characterize the UVLO conditions of the spacecraft, the power-on transients, and to provide recommendations to prevent the spacecraft from entering a boot-loop. To do this, the testbed test had to have significantly more monitoring equipment in place, as this was for the fine details. This was also an ambitious test, as the original testbed did not include an interface to run off of a flight spare or EM battery.

A modified battery-EPS harness was created and mated to the testbed that enabled a battery monitor and voltage and current probes to be installed in-line. Once that was done, the battery voltage had to be checked and am EM battery was used that had come out of storage. A power supply was then voltage matched to the battery, the harness attached, and the battery charge ports mated to the power supply. The battery was then leveled out at just above 9 volts for the test. Once that was complete, the current and voltage probes could be enabled.

With the probes enabled, the testbed was turned on, a safe mode uploaded, and standard power on checks for the testbed were executed. This included the cleanup that disabled the Iris faults as the testbed does not have an Iris, but a FPGA simulating the Iris's responses to commanding. The APIDs were set to maximum for data collection, and then the safe mode sequence was executed. The testbed was then monitored until it triggered UVLO, and then the UART had to be unplugged as there are issues with spacecraft power on if the UART is still connected. The power supply was then set to charge the battery back up, and once the testbed came back online everything was shut down and exported, and the battery was set to storage voltage.

5.2.2 XACT Time and Refs Characterization Test

This test was established to evaluate if the XACT time and references would invalidate themselves after the XACT responded to a fault condition. This condition was theorized, as the XACT's fault responses managed by the Lunar Flashlight FSW would reboot the system. This test was done to ensure that the appropriate time and references were loaded during as a part of the XACT fault response. The test for this case was relatively simple, as it consisted of the typical power on tests, setting the references and time, enabling and triggering the XACT faults, and then waiting a minute to verify that the invalid time and refs fault was not radiated.

5.3 Thermal Vacuum Fault Testing

Category 1A faults that are thermal related, were able to be tested in full during the Thermal Vacuum (TVAC) testing campaign. These tests were run at the hot and cold dwells, and followed a similar procedure for each test. First the temperature of the spacecraft and the subsystem was verified. Then an immediate command was sent to update the Gen Mon limits, and the fault ID was enabled. For TVAC testing there was no safe mode onboard, and therefore no concern about the faults with safe mode triggers.

5.4 Fault Testing Results

Overall, the testing campaign examined 135 different flight software fault IDs, and a number of other fault conditions that the Lunar Flashlight spacecraft could potentially encounter. For the fault protection testing campaign, a total of 414 individual immediate command files were written, and numerous testing scripts. The total coverage of the FSW ids can be seen in Figure 17. From the testing there were 20 deviations from the procedure that identified numerous idiosyncrasies, and one issue. The large majority of the responses triggered with the appropriate means, and had the assigned fault responses.

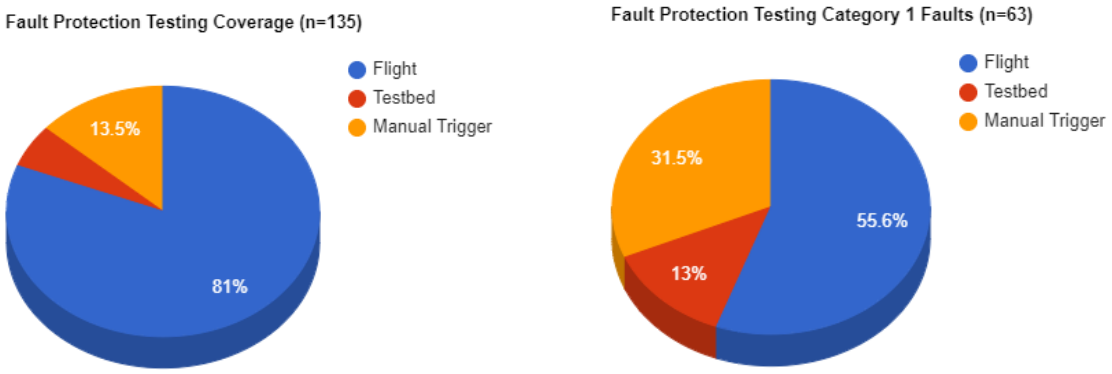


Figure 17: Fault Protection Testing Coverage

One issue uncovered was the UVLO triggering much faster than expected during testing. From the flight unit UVLO test that was executed on a flight spare battery, there was no intention to trigger the UVLO, but it was triggered after only 8 seconds. This issue was caused by the Iris and the XACT having high power draw and large startup transients. From this issue a flight software update was requested to stagger the XACT and Iris power on by two minutes to allow the XACT to get to sun-pointing before the Iris is enabled. In addition a flight rule was added.

6 Operations Assurance and Contingency Plan

The purpose of an Operations Assurance and Contingency Plan (OACP) is to provide an overview of the Mission Operations Center, its points of failure, and the actions taken by the Georgia Tech Operations team to assure any given mission success. In addition, it encapsulates contingencies of the Operations Team to critical mission failures, and the spacecraft fault management configuration throughout the mission phases. Some of this detail is out of scope of this paper, but can be found within the Lunar Flashlight project documentation. The scope of an OACP is to :

- Provide an overview of the MOC and its points of failure.
- Cover the staffing plan for the MOC, the approach taken to assign personnel to contacts, and the assignment of on-call status.
- Define the fault management plan for the spacecraft to provide which faults are enabled at each stage of the mission.
- Define the methodology for keeping the Georgia Tech Mission Operations Team accountable in decision making, reporting anomalies, and keeping the project up to date on the actions taken each contact.
- Provide contingency plans for key mission events such as TCM-1 and LOI, and for individual subsystem failures.

Crucially, this is different from the Anomaly Response Plan (ARP), whose purpose is to provide a framework for the GT Mission Operations Team to respond to anomalies in a rapid and appropriate manner. The Anomaly Response Plan, paired with response trees and reports, provides a flow to respond to both spacecraft and operations center-based anomalies. The OACP provides the planning and ground work for the spacecraft operations and fault management, where the ARP puts that ground work into action to help resolve any anomalies in real-time.

6.1 Reportable Incidents

For Lunar Flashlight Operations, all incidents are logged internally. Regardless of criticality, any incident, surprise, spacecraft hardware/software anomalies, or operational process and procedure should be recorded via the GT anomaly tracking system. For any incident that affects the spacecraft, the incident should be recorded in a Spacecraft Operations Anomaly Record as well as Spacecraft Operations Anomaly Tracking Log. For any incident that affects the Mission Operations Center, the incident should be recorded in a MOC Operations Anomaly Record as well as MOC Operations Anomaly Tracking Log.

All incidents should be recorded as soon as practical. Based on incident criticality, response to each incident will vary. All anomalies/incidents should be reported within 24 hours, external Problem Reporting System (PRS) involvement is indicated at each level of criticality. For missions events that are less high-profile, PRS involvement can be skipped.

6.2 Criticality

Each incident may vary in impact to the mission and are to be evaluated based on the following tiers of criticality.

6.2.1 Criticality 1

Represents major impact or threat to achieving mission success as illustrated but not limited to the following examples:

- Loss of required functional capability of spacecraft, instrument, or ground data system element
- Reduction in the project primary mission timeline
- Delay in mission activities resulting in failure to achieve a critical mission milestone
- Loss of capability to control spacecraft or instrument
- Permanent loss of essential engineering, science telemetry, or navigation radiometric data
- Spacecraft safing during critical event operations
- Error in the command uplink process indicating a vulnerability to sending a command with major mission degradation effects

6.2.2 Criticality 2

Represents significant impact or threat to achieving mission success as illustrated but not limited to the following examples:

- Degradation of required functional capability of spacecraft, instrument, or ground data system element
- Reduction in spacecraft subsystem or instrument lifetime in primary mission
- Delay in mission activities resulting in a slip of a critical mission milestone
- Degradation of capability to control spacecraft or instrument
- Partial loss of essential engineering or science telemetry or navigation radiometric data
- Loss of minor spacecraft or payload function
- Spacecraft safing during routine cruise operations
- Error in the command uplink process indicating a vulnerability to sending a command with significant mission degradation effects

6.2.3 Criticality 3

Represents negligible impact or threat to achieving mission success as illustrated but not limited to the following examples:

- Degradation of non-essential functional capability of spacecraft, instrument, or ground data system element
- Reduction in spacecraft subsystem or instrument lifetime in extended mission
- Delay in mission activities resulting in loss of schedule margin
- Degradation of engineering or science data or navigation radiometric data
- Degradation of minor spacecraft or payload function
- Increase in difficulty to perform operations functions
- Spacecraft idiosyncrasy requiring no corrective action
- Error in the command uplink process indicating a vulnerability to sending a command with negligible mission degradation effects

6.3 Anomaly Tracking System

For Lunar Flashlight, Georgia Tech uses the anomaly tracking and response shown in Figure 18. Starting at the point of the incident, first the anomaly is characterized as either a Mission Operations Center incident or Spacecraft incident. Based on the incident it is then evaluated on the previous levels of criticality to establish the formal response. Based on that criticality the appropriate personnel are called to be brought into the loop.

If the incident is spacecraft related, then a review of the flight rules is completed to check for flight rule infringement. Concurrently, the incident is responded to according to the section in the ground fault response tree. If the incident is MOC related, the impact to the ongoing contact is to be evaluated and if a immediate back-up is in place, implemented. Once the immediate response is completed, an Anomaly record is recorded. Spacecraft and MOC Anomaly record templates exist, and have an associated naming scheme.

Following the contact termination, the criticality response to the incident occurs. If the incident was of criticality 3, then the anomaly closure memo can be written and no further action is required. If the incident was of criticality levels 1 or 2, then the anomaly will need to be filed with JPL Problem Reporting System (PRS). Once that is complete, the anomaly needs to be reconstructed to confirm the suspicions as to why the incident occurred. Once the anomaly reconstruction is complete and course of action identified, an anomaly closure memo can be written and the resolution action executed.

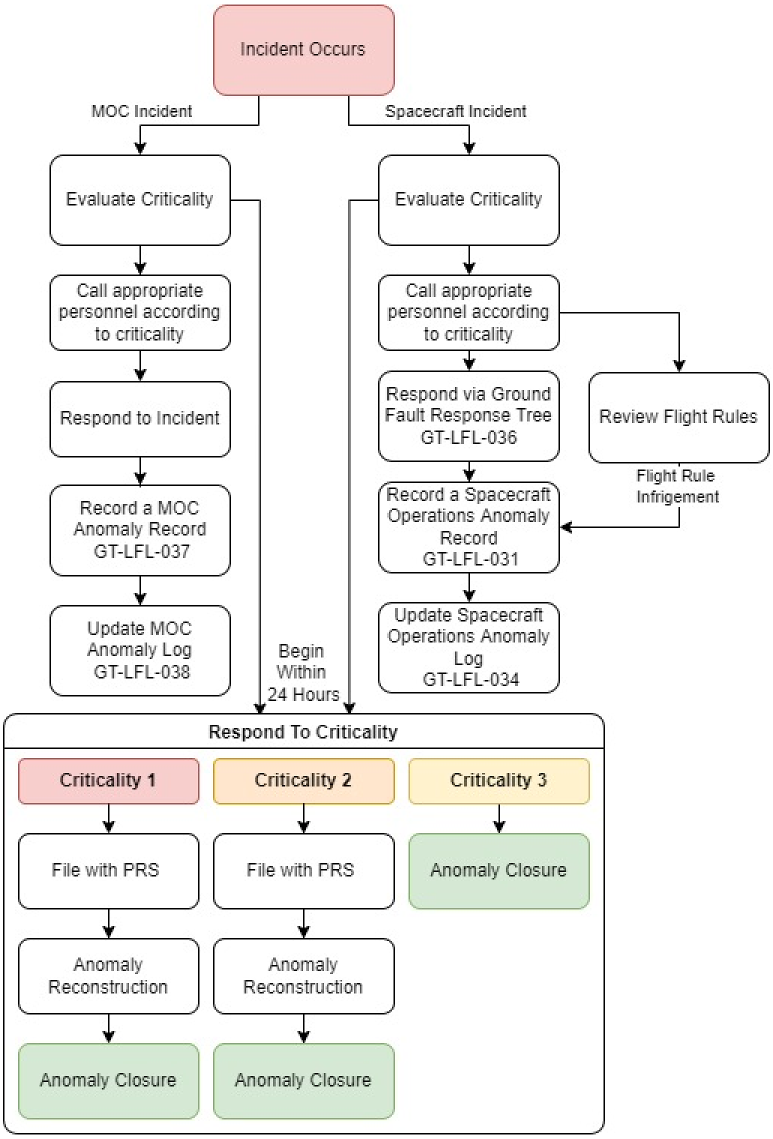


Figure 18: Georiga Tech LF Anomaly Tracking Flow

6.4 Decision-Making

To enable operators to make quick decisions and respond to anomalies in a short time-frame, a ground response tree was create for Lunar Flashlight FSW fault ids. Figure 19 shows the Category 1 fault ids and the color schema for the different expected fault responses. The fault ground response tree was split into each subsystem, as the range of fault id's typically indicate which subsystem to investigate. In the ground fault response tree, clear/harmless faults are indicated by a green outline. Faults that were likely caused by a FSW bug or radiation damage are indicated with a red outline. Low temperature faults are outlined in light blue, and high temperature faults are outlined in light red. Low voltage/current faults are outlined in grey, and high voltage/current faults are outlined in yellow. Finally there are two fault ids that have no response, indicated by the purple box.

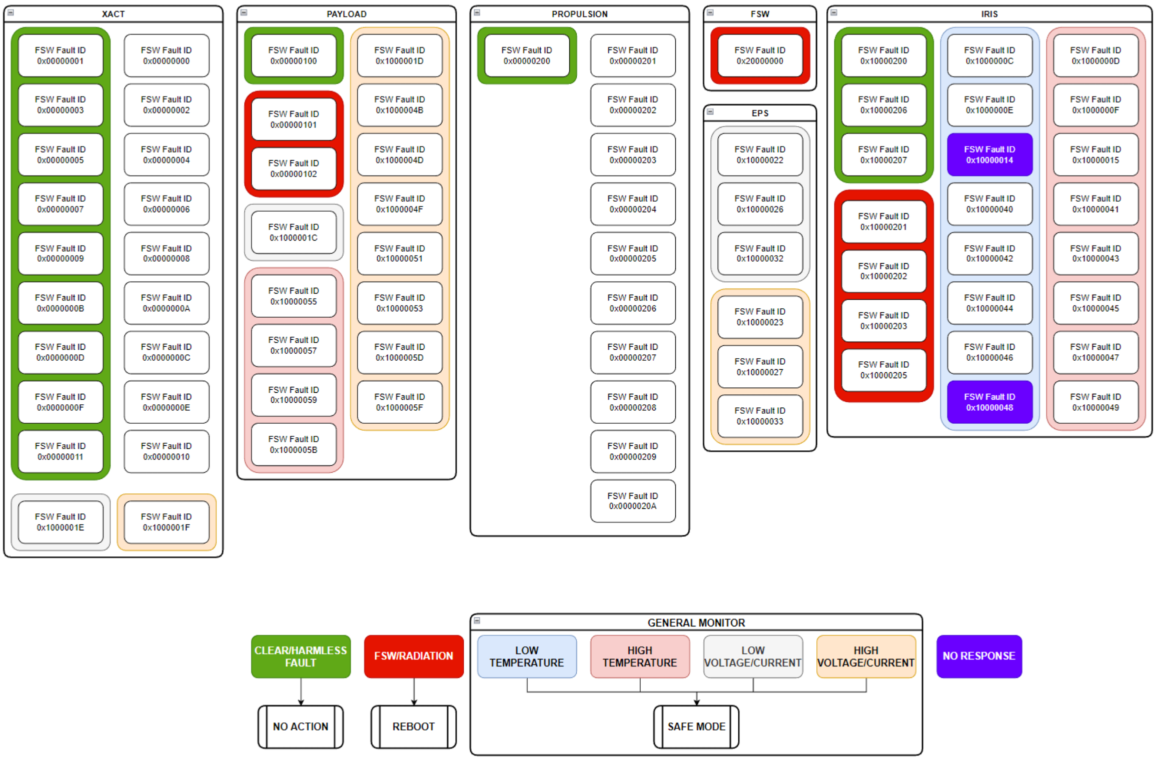


Figure 19: $GT - LFL - 036$ Lunar Flashlight Ground Response Tree Overview

For each of these faults there is an associated tree of response in the document. Figure 20 shows a sample response tree from the 0x00000200 and 0x00000201 fault IDs. The 0x00000200 fault ID response is quite simple: No action, as the fault is harmless. 0x00000200 specifically is the fault ID that the propulsion system sends when a maneuver is completed.

0x00000201 on the other hand has an associated tree. First, the response of the fault is to put the spacecraft into safe mode. The mission operator should then check the tank temperatures and ensure that they are in the appropriate ranges. If the fault continues to happen, then an anomaly report is created and further investigation is required. If the fault is not reoccurring, then the operators should log the fault, restore spacecraft mode to normal and may continue the contact. Note that a logged fault will also have an anomaly report, but will likely be closed immediately if it does not impact the mission directly.

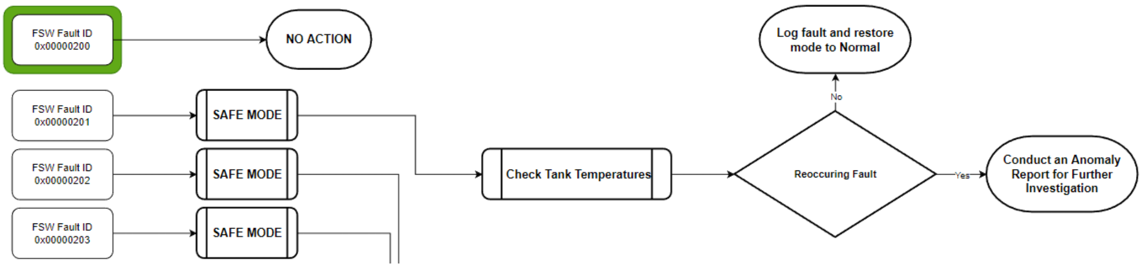


Figure 20: Ground Response Tree Propulsion Fault Sample from $GT - LFL -$ 036$GroundResponseFaultTree$

6.5 Post-Pass Reporting

After each contact, the mission operations team communicates the activities that occurred within that contact for project awareness.

6.5.1 Nominal Post-Pass Reporting

For nominal contacts, a post-pass report is released at the end of the contact. These reports are sent as an email to the operations group and external parties. They included

- Contact Number
- Personnel on shift
- Contact Description
- Contact Activities
- Matters (Any hiccup throughout the contact)
- Staffing for next contact
- Description for next contact

6.5.2 Spacecraft Anomaly Reporting and Reconstruction

When an anomaly with the Spacecraft occurs, it should immediately be noted, and recorded once the contact is over by the Mission Planner as an Spacecraft Anomaly Record and saved in the appropriate Box directory. Each anomaly record follows a defined naming scheme to enable ease of use for tracking the documents. The project and other parties should be notified in the post contact report with an indication of the criticality of the incident.

The Anomaly Record Detail includes:

- Date of Anomaly
- Time of Anomaly
- Project
- Subsystem/s
- Report Number
- Short Title
- Mission Phase
- Hardware affected/Involved
- Level of Criticality

This report also includes:

- Description
- Current Spacecraft State
- Decision to Continue Contact
- Path Forwards
- Lessons Learned
- Photos/Snapshots

Given the anomaly time listed in the Anomaly Record document, the EVRs and telemetry from the spacecraft should be logged and added to the Anomaly Record folder. If the Anomaly is fault based and of Criticality level 1 or 2, then the following additional steps should be executed:

- The spacecraft state before the anomaly should be established from the telemetry logs
- Then the state should then be re-created on the testbed, and the predicted anomaly condition should be triggered

This will serve to confirm the predicted anomaly condition. If the anomaly is not replicated by the previous steps then the proposed cause should be confirmed by analysis. The results of the reconstruction will be appended to the anomaly record and/or the anomaly closure memo.

6.5.3 Mission Operations Center Anomaly Reporting

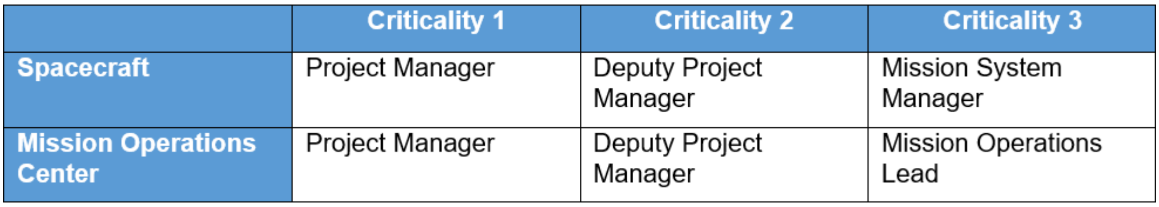
A Mission Operations Center Anomaly Record is created from the template document, and then the internal tracking log of MOC Anomaly Records is updated to include the Anomaly Record. This report differs from the typical spacecraft report due to the additional sections for submitting tickets with the Aerospace Engineering Information Technology (IT) group.

- Anomaly record name
- MOC State
- Additional Report Required?
- Submitted IT Ticket?
- IT Ticket Resolved?
- Contact Name
- Date of occurrence
- Date of Resolution
- Level of Criticality

For anomalies that require IT support, IT tickets submitted for anomalies promptly after AR is completed.

6.6 Anomaly Closure

For each level of criticality and type of anomaly there are specific individuals that are required to assess the anomaly and respond for closure. These individuals are identified in Figure 21.



	Criticality 1	Criticality 2	Criticality 3
Spacecraft	Project Manager	Deputy Project Manager	Mission System Manager
Mission Operations Center	Project Manager	Deputy Project Manager	Mission Operations Lead

Figure 21: Criticality Personnel Contact

If the anomaly is of Criticality 1 or 2, then there may be an associated memo that will be written as a part of the anomaly closure process. These memos will be utilized to inform the JPL PRS anomaly reporting system.

7 GT MOC Risk Analysis

Thus far the majority of the discussion has been about fault management associated with the spacecraft. However, it is critical that the MOC be involved in the Verification and Validation (VV) process to ensure that Georgia Tech is capable of performing mission operations for Lunar Flashlight.

7.1 Methodology

A key component of evaluating a system's failure modes is the Functional Hazard Assessment (FHA). The FHA considers functions and the different conditions required to cause a fault, as well as the severity of the identified fault in the different mission phases. First each function is identified, and then each failure or malfunction condition can be considered. Hazard levels are then assigned to each failure condition, and the assessment is iterated as the system is developed. Typically FHAs cover the subsystem/system level of any given system.

For this analysis the levels of severity assigned were:

1. Insignificant
2. Minor
3. Moderate
4. Major
5. Critical

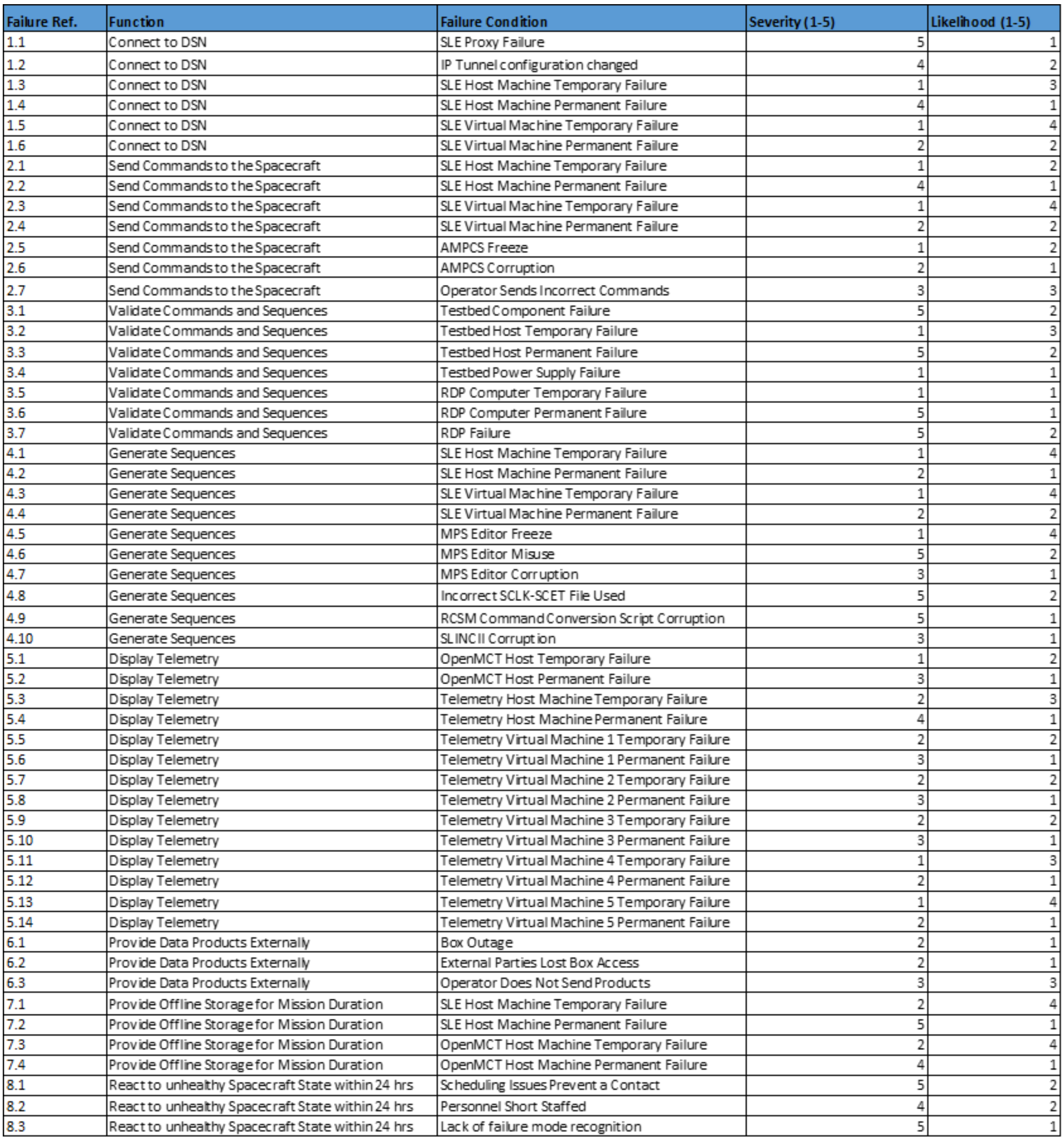
And the levels of likelihood were identified as:

1. Very Low (0% - 1%)
2. Low (1% - 30%)
3. Moderate (30% - 50%)
4. High (50% - 70%)
5. Very High (70% - 100%)

For this FHA, first the functions of the MOC were identified. These were found to be the following:

1. Connect to the DSN
2. Send Commands to the Spacecraft
3. Validate Commands and Sequences
4. Generate Sequences
5. Display Telemetry
6. Provide Data Products Externally
7. Provide Offline Storage for Mission Duration
8. React to Unhealthy Spacecraft State within 24 hrs

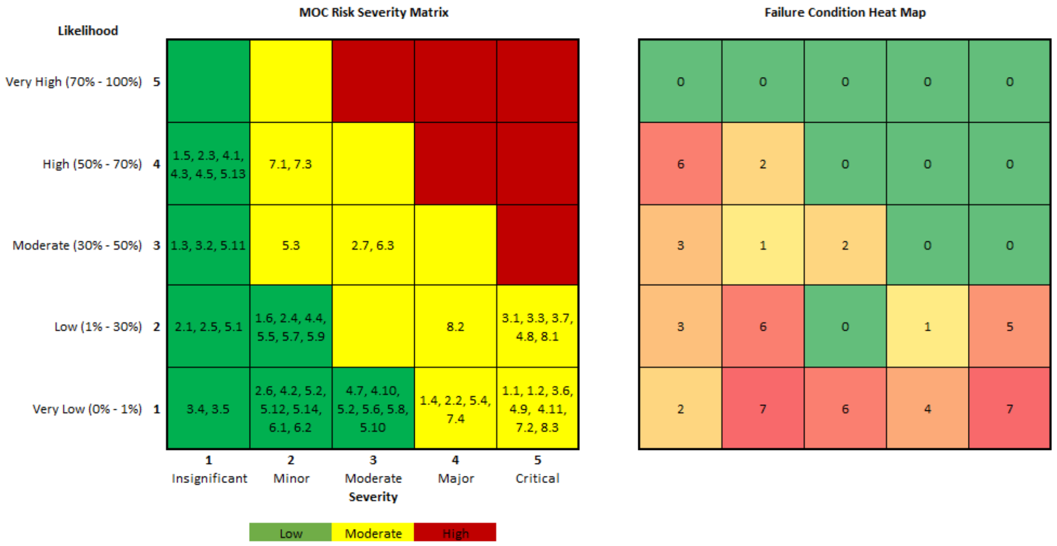
Once the MOC Functions were identified then the individual failure conditions were identified and then rated. These failure conditions can be seen in Figure 22. The FHA is an important tool that can be used to rank VNV items and help provide a clear understanding of the weak links in a system.



Failure Ref.	Function	Failure Condition	Severity (1-5)	Likelihood (1-5)
1.1	Connect to DSN	SLE Proxy Failure	5	1
1.2	Connect to DSN	IP Tunnel configuration changed	4	2
1.3	Connect to DSN	SLE Host Machine Temporary Failure	1	3
1.4	Connect to DSN	SLE Host Machine Permanent Failure	4	1
1.5	Connect to DSN	SLE Virtual Machine Temporary Failure	1	4
1.6	Connect to DSN	SLE Virtual Machine Permanent Failure	2	2
2.1	Send Commands to the Spacecraft	SLE Host Machine Temporary Failure	1	2
2.2	Send Commands to the Spacecraft	SLE Host Machine Permanent Failure	4	1
2.3	Send Commands to the Spacecraft	SLE Virtual Machine Temporary Failure	1	4
2.4	Send Commands to the Spacecraft	SLE Virtual Machine Permanent Failure	2	2
2.5	Send Commands to the Spacecraft	AMPCS Freeze	1	2
2.6	Send Commands to the Spacecraft	AMPCS Corruption	2	1
2.7	Send Commands to the Spacecraft	Operator Sends Incorrect Commands	3	3
3.1	Validate Commands and Sequences	Testbed Component Failure	5	2
3.2	Validate Commands and Sequences	Testbed Host Temporary Failure	1	3
3.3	Validate Commands and Sequences	Testbed Host Permanent Failure	5	2
3.4	Validate Commands and Sequences	Testbed Power Supply Failure	1	1
3.5	Validate Commands and Sequences	RDP Computer Temporary Failure	1	1
3.6	Validate Commands and Sequences	RDP Computer Permanent Failure	5	1
3.7	Validate Commands and Sequences	RDP Failure	5	2
4.1	Generate Sequences	SLE Host Machine Temporary Failure	1	4
4.2	Generate Sequences	SLE Host Machine Permanent Failure	2	1
4.3	Generate Sequences	SLE Virtual Machine Temporary Failure	1	4
4.4	Generate Sequences	SLE Virtual Machine Permanent Failure	2	2
4.5	Generate Sequences	MPS Editor Freeze	1	4
4.6	Generate Sequences	MPS Editor Misuse	5	2
4.7	Generate Sequences	MPS Editor Corruption	3	1
4.8	Generate Sequences	Incorrect SCLK-SCET File Used	5	2
4.9	Generate Sequences	RCSM Command Conversion Script Corruption	5	1
4.10	Generate Sequences	SLINCII Corruption	3	1
5.1	Display Telemetry	OpenMCT Host Temporary Failure	1	2
5.2	Display Telemetry	OpenMCT Host Permanent Failure	3	1
5.3	Display Telemetry	Telemetry Host Machine Temporary Failure	2	3
5.4	Display Telemetry	Telemetry Host Machine Permanent Failure	4	1
5.5	Display Telemetry	Telemetry Virtual Machine 1 Temporary Failure	2	2
5.6	Display Telemetry	Telemetry Virtual Machine 1 Permanent Failure	3	1
5.7	Display Telemetry	Telemetry Virtual Machine 2 Temporary Failure	2	2
5.8	Display Telemetry	Telemetry Virtual Machine 2 Permanent Failure	3	1
5.9	Display Telemetry	Telemetry Virtual Machine 3 Temporary Failure	2	2
5.10	Display Telemetry	Telemetry Virtual Machine 3 Permanent Failure	3	1
5.11	Display Telemetry	Telemetry Virtual Machine 4 Temporary Failure	1	3
5.12	Display Telemetry	Telemetry Virtual Machine 4 Permanent Failure	2	1
5.13	Display Telemetry	Telemetry Virtual Machine 5 Temporary Failure	1	4
5.14	Display Telemetry	Telemetry Virtual Machine 5 Permanent Failure	2	1
6.1	Provide Data Products Externally	Box Outage	2	1
6.2	Provide Data Products Externally	External Parties Lost Box Access	2	1
6.3	Provide Data Products Externally	Operator Does Not Send Products	3	3
7.1	Provide Offline Storage for Mission Duration	SLE Host Machine Temporary Failure	2	4
7.2	Provide Offline Storage for Mission Duration	SLE Host Machine Permanent Failure	5	1
7.3	Provide Offline Storage for Mission Duration	OpenMCT Host Machine Temporary Failure	2	4
7.4	Provide Offline Storage for Mission Duration	OpenMCT Host Machine Permanent Failure	4	1
8.1	React to unhealthy Spacecraft State within 24 hrs	Scheduling Issues Prevent a Contact	5	2
8.2	React to unhealthy Spacecraft State within 24 hrs	Personnel Short Staffed	4	2
8.3	React to unhealthy Spacecraft State within 24 hrs	Lack of failure mode recognition	5	1

Figure 22: Fault Hazard Analysis of the MOC

From this FHA, a likelihood-concequence risk severity matrix was created as seen in Figure 23. This risk severity matrix was matched to the JPL project risk severity matrix. From this matrix it is clear that Lunar Flashlight does not have any critical items in the high risk area, indicated in red.



Likelihood					
Very High (70% - 100%) 5					
High (50% - 70%) 4	1.5, 2.3, 4.1, 4.3, 4.5, 5.13	7.1, 7.3			
Moderate (30% - 50%) 3	1.3, 3.2, 5.11	5.3	2.7, 6.3		
Low (1% - 30%) 2	2.1, 2.5, 5.1	1.6, 2.4, 4.4, 5.5, 5.7, 5.9		8.2	3.1, 3.3, 3.7, 4.8, 8.1
Very Low (0% - 1%) 1	3.4, 3.5	2.6, 4.2, 5.2, 5.12, 5.14, 6.1, 6.2	4.7, 4.10, 5.2, 5.6, 5.8, 5.10	1.4, 2.2, 5.4, 7.4	1.1, 1.2, 3.6, 4.9, 4.11, 7.2, 8.3
Severity	1 Insignificant	2 Minor	3 Moderate	4 Major	5 Critical

0	0	0	0	0
6	2	0	0	0
3	1	2	0	0
3	6	0	1	5
2	7	6	4	7

Figure 23: MOC Risk Severity Matrix and Heatmap

7.2 High Risk Elements and Resolution

From the Fault Hazard Assessment, the most concerning MOC related faults are the 3.1, 3.7, 4.8, 8.1, and 8.2. These risks correspond to

- 3.1: Testbed Component Failure, impact to the validating commands functionality of the MOC
- 3.3: Testbed Host Permanent Failure, impact to the validating commands functionality of the MOC
- 3.7: RDP Failure, impact to the validating commands functionality of the MOC
- 4.8: Incorrect SCLK-SKET file used, impact to the sequence generation functionality of the MOC
- 8.1: Scheduling issues preventing a contact, impact to the reaction to unhealthy spacecraft state within 24 hrs functionality of the MOC
- 8.2: Personnel Short Staffed, impact to the reaction to unhealthy spacecraft state within 24 hrs functionality of the MOC

These issues already have resolutions, laid out in the table below.

Risk	Mitigation
3.1	Additional Sphinx procured and being tested as a back-up. Testbed upgraded with interface board to make more flight-like. IRIS EM integrated for RF capability.
3.3	Created a back-up of the testbed host to ensure testbed host recovery given permanent failure.
3.7	Open discussion about purchase of backup RDP.
4.8	Automated SCLK-SCET file selection and automated sequence generation pipeline to prevent user error.
8.1	Consistent communication with the DSN coordinator and Project about available contacts.
8.2	GT Ops team has hired more personnel and is in the processing of training new operators.

8 Ground Station Reliability

Not only are Small satellites error prone but, university ground station infrastructures are also prone to malfunction and are often teetering on the edge of disrepair. From personal experience with university ground stations, they tend to fail frequently due to a combination of poor design, lack of experienced maintenance, and lack of use outside of the mission operations window. Most CubeSat missions tend to ignore the operations window until they have shipped the flight unit. Unfortunately, ground station development is typically not a hot topic and does not have much funding. The result is a ground segment that is dominated by individuals developing in free time.

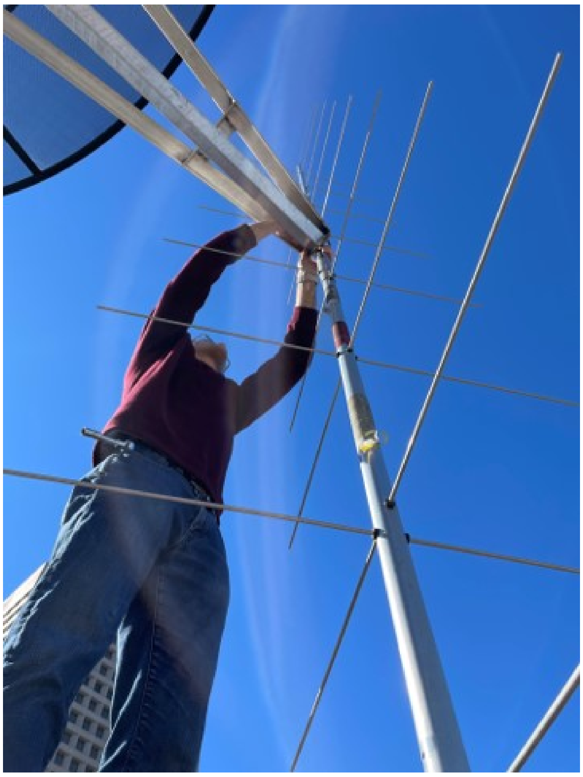


Figure 24: Montgomery Knight Ground Station Under Maintenance

8.1 Historical Performance

The Georgia Tech Ground Station Network aims to solve these mission to mission ground segment concerns. The aim is to provide an interface to the system that enables each mission to plug in and perform the mission. The current state of the GT Ground Station Network does not fully represent that vision. As it stands, currently each ground station operates independently.

For a given mission to utilize the ground station network, one must develop a packet decoder if they wish to have their health beacon decoded as a part of the post-pass script. They must also develop a commanding tool that will get called in the pre-pass script. This is effective if the spacecraft has a guaranteed link, which is almost never the case. The short term solution is to spam a given command with the TX spammer. Again, this works, but if the command is to request a data downlink, then the downlink can get overridden with command acknowledgements.

This was the implementation used for GT-1 and ARMADILLO. A packet decoder was implemented to ensure that valid health beacons were received. TX spammer was called to send commands. ARMADILLO reached the limit of the implementation with the development of commanding software that required both transmit and reception. This was to send acknowledgement commands and confirm the reception of that command before moving on to the next command. It also solved the timing issue, as the ground control software could update the relative commanding time based on the SCLK.

8.2 Resolution

The resolution to this issues is the implementation of a more advanced ground station network. The building blocks for which have already been developed. The architecture that the ground station network is moving towards is shown in Figure 25. The link component between each node has been developed. The next step is to integrate the GS-Link into the existing station nodes that are standalone. Once that is complete, then a Clearinghouse can be developed to provide the interface to any party to schedule ground station network assets.

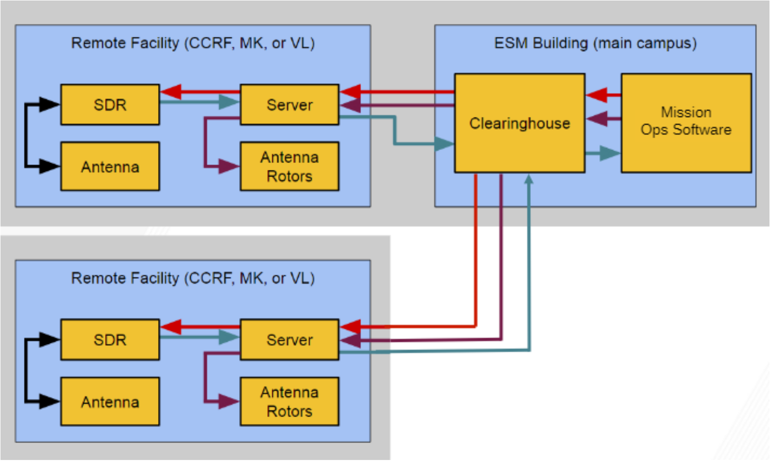


Figure 25: Planned Ground Station Network Architecture

The clearinghouse is critical, as it will act as the distributor of the schedule to the different assets, as well as the pipe through which telemetry and commands flow. This will also enable the Mission Operations Center to schedule either "Live" contacts or automated contacts. Automated contacts will leverage the existing ground station chain, and can have a piece of operations software receive and transmit commands without an operator in the loop. The intention with this architecture is to make the individual stations mission operations software agnostic.

Another advantage to this architecture is the capability to command assets while in the MOC. This would enable easier satellite communications testing with missions developing their testbeds. The existing infrastructure requires ground station operators to manually command each station they wish to test. If this can be routed through the mission operations center, it will allow for faster and more robust communications system developement.

9 Recommendations

From the implementation of fault protection on ARMADILLO and experience with numerous CubeSats in their operational phases there are a few key components to a successful spacecraft.

First and foremost, design for failure. Establish the worst case scenario for the spacecraft in terms of power early in the development of the system. Ensure that, in the case that solar arrays fail to deploy, or otherwise, that the spacecraft radio and command and data computer can survive with minimal power consumption. More often than not, it is critical to ensure these components of the spacecraft bus are as robust as possible.

This also means defining a spacecraft safe mode early and understanding the interactions of that safe-mode with your EPS under-voltage lockout. The UVLO enable voltage and restoration voltage are to be carefully considered. If your safe-mode start-up has a transient that will immediately place the spacecraft back into UVLO, then the restoration voltage needs to be elevated. The goal of the EPS UVLO restoration limit is to provide the spacecraft enough power to have it return to a power-positive state. For deployment in particular, the safe mode can consider the Power On Reset (POR) count to ensure the automated burn-wire sequence or any other high-current devices are not triggered repeatedly, or over a maximum POR count.

Second, implement error handling. The spacecraft should be able to able to restore function in the case that the FSW is in an undefined state. If the CDH computer is no longer responding, ensure that there is a watchdog implementation that will power cycle the CDH in the case the FSW is not sending a heartbeat. Then, ensure that if any field is populated with an invalid argument then an Event Verification Record is produced and a default argument can be sent in its place. This approach is critical for spacecraft state management, ensure that there is always a defined state.

The most critical fault to implement in every spacecraft is the command loss timer. Ensure that the timer is of an appropriate length, in the case that something goes wrong with the radio or otherwise, the spacecraft will reboot itself and hopefully restore that functionality. In addition, ensure that there is a firecode implementation. A firecode does not require any CDH, the radio should recieve the command

and set a pin that causes the EPS to hard-reset the spacecraft. This proved to save ARMADILLO in an instance where the command loss timer soft-resets were not restoring the beacon functionality.

Unless the particular mission requires a more specialized state flow, the simplistic spacecraft state flow shown in Figure 26 should be implemented. The safe mode, as previously discussed, is the lowest power state of the spacecraft and only has the functionality to transmit a health beacon, and receive commands. Once commanded out of safe mode, only then can FSW faults have any impact. Idle is the nominal state where anything can be commanded with the fault protection in place to put the spacecraft back into safe mode. It is critical to note that the spacecraft command loss timer should perform a soft reset even when the spacecraft is in safe mode.

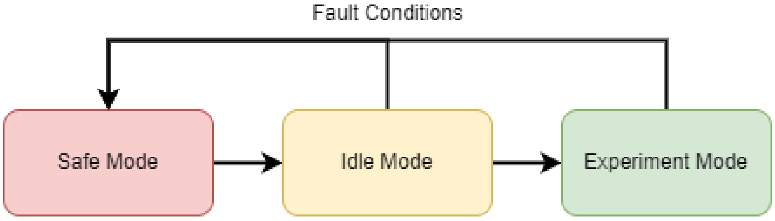


Figure 26: Simple Spacecraft State Diagram

Third, the spacecraft experiment state should allow for the experiment and experiment specific faults to be enabled. This is in the case that the experiment has particular fault responses that can contain branching logic to ensure data validity. These faults get overridden with the safe mode condition call, and prevent damage to the payload.

Offload as much of the fault management as possible onto the subsystems for specific control scenarios. For example, if the EPS needs to maintain the battery within a thermal range, then ensure that the EPS has it's own thermal response to maintain the battery within the operational range, and in the case it exceeds the range, do not charge it and inform the CDH. This will simplify the central fault protection implementation, and ensures that subsystems maintain their health in the case that the CDH fails.

Fourth, establish the interactions of faults with sequences. During mission critical events it is essential that any given fault does not jeopardize the mission as a whole. Figure 27 shows the recommended spacecraft fault interaction rules. Lunar Flashlight's fault protection utilizes a single interaction rule, and that is the non-critical sequence implementation. This implementation in particular can be worked around by disabling particular faults within each mission phase. However, if the LF faults had different interaction levels, then the fault protection implementation would have required significantly less work to manage for operations. The key concerns associated with implementing different interaction levels are additional complexity adding risk, and increasing the difficulty testing the system.

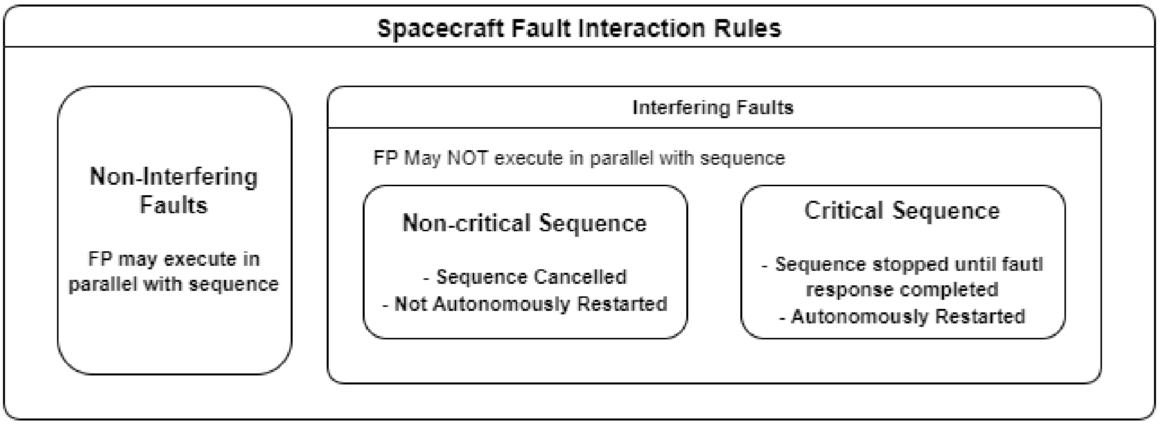

Spacecraft Fault Interaction Rules
Non-Interfering Faults
FP may execute in parallel with sequence
Interfering Faults
FP May NOT execute in parallel with sequence
Non-critical Sequence
- Sequence Cancelled
- Not Autonomously Restarted
Critical Sequence
- Sequence stopped until fautl response completed
- Autonomously Restarted

Figure 27: Spacecraft Fault Interaction Rules [6]

Fifth, record fault protection data, heartbeat data, and state data per subsystem and provide that information within EVRs or the health beacon. This data in particular is critical for troubleshooting, and can provide insight on the ground for subsystem failures. Table 1 shows a simple and general implementation of typical faults and paired with downlinked EVRs and current spacecraft performance, once can deduce what occurred.

Example Fault Implementation	Fault	Response
CDH Faults		
	Command Loss Timer	Soft Reset, Safe Mode
	Invalid State	Soft Reset, Safe Mode
EPS Faults		
	Battery Voltage Low/High	Safe Mode
	Battery Current Low/High	Safe Mode
	Subsystem 1 Current Low/High	Safe Mode
	Subsystem 2 Current Low/High	Safe Mode
	Subsystem 3 Current Low/High	Safe Mode
ADCS Faults		
	Sunpoint Mode Failure	Restart ADCS
	Invalid Time/References	Restart ADCS, set time/refs
	Fine hold failure	Restart ADCS
Thermal Faults		
	Subsystem 1 Temperature High	Safe Mode
	Subsystem 2 Temperature High	Safe Mode
	Subsystem 3 Temperature High	Safe Mode
	Subsystem 1 Temperature Low	Enable Neighboring Components (If possible)
	Subsystem 2 Temperature Low	Enable Neighboring Components (If possible)
	Subsystem 3 Temperature Low	Enable Neighboring Components (If possible)
COM Faults		
	Radio Firmware Fault	Restart radio until response
	Radio Heartbeat Fault	Restart radio until response

Table 1: Sample Fault Protection Implementation

Finally, the general architecture that is recommended is the same implementation that Lunar Flashlight implemented. Separate applications for monitoring and response. The general monitor watching the telemetry, and calling the fault protection manager (FPM) if a telemetry point exceeds that limit. Then the fault protection manager having the defined fault responses and which are enabled. If the response is enabled, the FPM checks the interaction rule of the sequence/fault and runs the fault response. Where the fault can either be synchronous or not, and the sequence can be critical or not.

10 Acknowledgements

I would like to thank the folks that helped me along the way. Thank you Dr. Lightsey for trusting me with such an important mission to Georgia Tech, and for letting me hop around on different projects to satisfy my many different interests. I honestly don't think I would have enjoyed my graduate school experience at any other school, in any other lab.

Thank you to Sterlng Peet, and ground station contributors. Without their commitment to the ground segment, many missions would not see remote success. Thank you for commiserating with me about the seemingly endless ground station woes.

Thank you to the mission operations team. Jishnu Medisetti, Mason Starr, Michael Hauge, and John Cancio thank y'all for keeping me sane. In addition, thanks to Lacey Littleton, Celeste Smith, Nathan Cheek, and Conner Awald for the endless support, both on Lunar flashlight and otherwise. Thank you to all of the undergraduate researchers, and new graduates that are supporting Lunar Flashlight, your contribution is unquantifiable. Without all of your support in testing, procedure development, tool generation, and additional miscellaneous tasks, our original group of four on this project would have probably lost our minds.

Thank you Anthony Shao-Berkery, Sam Mouradian, Stuart Demcak, Ted Sweetser, Aadil Rizvi, Kevin Lo, Philippe Adell, Shannon Statham, Kris Buckmaster, and everyone else from the JPL team who have supported us. We started this project without much experience in mission operations, and you all have provided valuable guidance along the way.

Finally, thank you to Mom for consoling me during my lowest of lows, Dad for celebrating my highs, and my brothers for keeping me humble.

Figure 28: The GT Lunar Flashlight Team

References

[1] Barbara A. Cohen, Paul O. Hayne, Benjamin Greenhagen, David A. Paige, Calina Seybold, and John Baker. Lunar Flashlight: Illuminating the Lunar South Pole. *IEEE Aerospace and Electronic Systems Magazine*, 35(3):46–52, March 2020. Conference Name: IEEE Aerospace and Electronic Systems Magazine.

[2] Melissa R Jones, Kristin A Fretz, Sanae D Kubota, and Clayton A Smith. Integrating Reliability Engineering with Fault Management to Create Resilient Space Systems. *Johns Hopkins APL Technical Digest*, 34(4), 2019.

[3] Martin Langer and Jasper Bouwmeester. Reliability of CubeSats - Statistical Data, Developers' Beliefs and the Way Forward. *30th Annual AIAA/USU Conference on Small Satellites*.

[4] Dillan McDonald and E. Glenn Lightsey. Recovery of a Lost Satellite: The ARMADILLO Mission. 2022. Conference Name: Small Satellite Conference, Logan, Utah.

[5] Jishnu Medisetti and E. Glenn Lightsey. Development of a Lunar Mission Operations Center for the NASA JPL Lunar Flashlight Mission. 2022.

[6] Paula S. Morgan. Fault Protection Techniques in JPL Spacecraft. 2005. Conference Name: First International Forum on Integrated System Health Engineering and Management in Aerospace (ISHEM), Napa, California.

[7] Aadil Rizvi, Kevin F Ortega, and Yutau He. Developing Lunar Flashlight and Near-Earth Asteroid Scout Flight Software Concurrently using Open-Source F Prime Flight Software Framework. 2022. Conference Name: Small Satellite Conference, Logan, Utah.

11 Appendix A: Fault Protection Testing Campaign Flow

11.1 Flight Unit Testing

11.1.1 Spacecraft Setup

- Run functional procedure for spacecraft initialization
- Establish connection with the spacecraft
- Safe Mode Sequence Trigger
- Load a safe mode sequence onto the spacecraft
- Set the spacecraft mode to normal
- Enable fault responses
- Trigger a fault with a safe mode response
- Verify that the safe mode sequence executes as expected

11.1.2 Category 1 Fault Response Validation

- Remove the safe mode sequence
- Load the test interrupt sequence
- Per each category 1A fault
  - Create the scenario for fault trigger
- For XACT faults, unique to each fault
- For Gen Mon faults, setting the range out of bounds
  - Enable Gen Mon
  - Enable fault id
  - Monitor for Fault response trigger EVR
  - Verify fault response
- Per each Category 1B fault
  - Enable the fault id
  - Trigger the fault id
  - Monitor for Fault response trigger EVR
  - Verify fault response

11.1.3 Category 2A Fault Response Validation

- Create the scenario for fault trigger by setting the range out of bounds
- Enable Gen Mon
- Enable fault id
- Monitor for Fault response trigger EVR
- Verify fault response

11.1.4 Fault Persistence Test

- Remove the safe mode sequence
- Enable fault responses
- Enable fault id for XACT fault
- Run the persistence script that sends a reoccurring fault trigger
- Verify that the persistence response is as expected
- Kill the persistence script
- Disable fault id for XACT fault
- Clear the XACT fault
- Turn on payload
- Soft reset payload
- Enable fault id for payload fault
- Run the persistence script that sends a reoccurring fault trigger
- Verify that the persistence response is as expected
- Kill the persistence script
- Disable fault id for payload fault
- Clear the payload fault
- Turn off payload
- Enable fault id for IRIS fault
- Run the persistence script that sends a reoccurring fault trigger
- Disable IRIS telemetry query
- Verify fault trigger
- Update onboard time to be 45 minutes in the future
- Verify for the persistence response for the IRIS
- Disable IRIS fault ID
- Re-enable IRIS telemetry query
- Disable fault responses

11.1.5 Fault Clear Test for the XACT, IRIS, and Payload

- Enable fault responses
- Enable fault id
- Trigger fault
- Verify fault response
- Trigger fault
- Verify no fault response
- Send clear fault
- Trigger fault
- Verify fault response

11.1.6 IRIS Fire Code Test

- Run functional procedure for spacecraft initialization
- Setup CASSY
- Establish connection with the spacecraft
- Verify CASSY lock
- Send fire code
- Verify IRIS restarts as expected

11.1.7 Flight Unit UVLO Test

- Power on Spacecraft
- Setup power supply to charge at minimum solar panel charge
- Setup CASSY
- Upload Safe Mode Sequence
- Clear unwanted fault IDs
- Set APIDs to high
- Discharge battery until at 9 volts
- Trigger safe mode
- Monitor safe mode sequence execution
- Charge battery to storage voltage
- Run test breakdown procedure

11.2 Testbed Testing

11.2.1 Testbed UVLO Test

- Mate the EPS to the testbed battery harness
- Connect the battery monitor to the testbed battery harness
- Connect voltage and current probes in-line with the battery monitor
- Mate the battery to the testbed battery harness
- Check the battery voltage
- Configure power supply to voltage match the battery
- Mate the power supply to the battery charge ports on the testbed battery harness
- Ensure that the battery is just above 9V
- Enable logging on the voltage and current probes
- Upload safe mode sequence to the spacecraft
- Perform standard power-on checks
- Issue ATB cleanup
- Set APIDs to maximum
- Discharge to 9V

- Monitor until UVLO is triggered
- Unplug the UART on the testbed host
- Stop UART redirect script
- Set the power supply charging current to minimum solar charge
- Verify EM battery is charging
- Once Sphinx LEDs are lit
  - Plug in UART
  - Restart UART redirect
- Monitor EVRs until safe mode sequence is complete
- Charge battery to storage voltage
- Shutdown testbed
- Export session data

11.2.2 XACT Time and Refs Characterization Test

- Perform standard power-on checks
- Set Chebyshev polynomial coefficients
- Set Chebyshev time references
- Enable XACT fault
- Trigger XACT fault
- Wait 1 minute and verify that invalid time and ref faults are not radiated

11.3 Thermal Vacuum Fault Testing

11.3.1 High Temperature Faults

- Verify that the spacecraft has reached the high plateau temperature
- Verify that the tested fault channel is at a triggerable temperature
- Update the Gen Mon to have the current temperature trigger the fault
- Enable the Gen Mon channel
- Enable the Spacecraft Fault ID

11.3.2 Low Temperature Faults

- Verify that the spacecraft has reached the low plateau temperature
- Verify that the tested fault channel is at a triggerable temperature
- Update the Gen Mon to have the current temperature trigger the fault
- Enable the Gen Mon channel
- Enable the Spacecraft Fault ID