SSC25-RAIV-01

Launch Fast Learn Fast: Lessons Learned from the R5-S2 and R5-S4 Missions

James Berck, Sam Pedrotty, Jack Wisbiski, Morgan Alexander, Kathryn Knesek,
Ruth May, Jacob Nunez-Kearny, Rob Stoddard, Thomas Kirven, Robin Onsay
NASA Johnson Space Center
2101 E NASA Pkwy
Houston, TX 77058; 281-678-9761
james.w.berck@nasa.gov

Roger Hunter
NASA Ames Research Center
M/S 213-13, Moffett Field, CA 94035; 650-604-5004
roger.c.hunter@nasa.gov

ABSTRACT

The Realizing Rapid, Reduced-cost high-Risk Research (R5) project is pushing the envelope on minimizing cost and schedule for complex spacecraft that support low Technology Readiness Level payload demonstration in Low Earth Orbit (LEO). R5 has recently designed, built, launched, and operated multiple spacecraft that are iteratively and incrementally delivering improved payload hosting capabilities while maintaining cost targets (< $100k in materials per bus) and schedule targets (launching at least two spacecraft per year). This paper will describe the unique architecture, approach, technological advancements, and lessons learned from the project. The primary stakeholder for R5 is the Small Spacecraft Technology program within the Space Technology Mission Directorate at NASA with the project based out of Johnson Space Center.

To achieve the aggressive cost and schedule targets, R5 is redefining the understanding of how much rigor, quality, and robustness are required to succeed in high-risk LEO missions. Starting with a clean-slate for process and system design, R5 has charted a path, subsystem by subsystem, through commercial, industrial, automotive, and medical-grade components to arrive at a functional baseline bus that includes high-performance onboard computing, six degree-of-freedom control from a cold-gas propulsion system, automated mission operations, and other capabilities. R5 has also accounted for regulatory schedule impacts in its architecture development, which includes using communication systems that are faster to approve through the regulatory process. R5 Spacecraft 2 (R5-S2) and R5 Spacecraft 4 (R5-S4) are the first two R5 spacecraft to successfully make it to orbit. These spacecraft, which contain no space-rated components, quickly exceeded their minimum mission success criteria and operated throughout their 10 month mission lifetime in orbit. The data gathered during the mission and its consideration for the next generation of R5 spacecraft is covered below.

By its very nature, R5 is constantly learning lessons regarding component suitability, process, and architectural design decisions and their impact on the project's goals. R5 has captured more than 70 lessons learned that span all subsystems and phases of spacecraft development and operations. R5 tracks the infusion of the lessons learned into the broader space community as a success metric. This paper provides details on lessons learned with the goal of enabling teams across the industry to leverage these lessons to improve their own cost, process, and schedule.

INTRODUCTION TO THE R5 PROJECT

The Realizing Rapid, Reduced-cost high-Risk Research (R5) project is primarily intended to integrate, launch, and then operate payloads in Low Earth Orbit (LEO). While commercial capabilities exist for demonstrating payloads in LEO, their cost and lead time prevent many low Technology Readiness Level (TRL) or responsive payloads from being demonstrated. The R5 project is purposefully built to fill this gap. R5 enables responsive payload demonstration with a highly flexible architecture that can be changed very late in the design flow with a schedule target of less than 12 months between payload identification and on-orbit operation. R5 also enables low TRL payload demonstration by minimizing the cost of demonstration, which is realized by minimizing process, minimizing bus parts cost with a cost target of less than $100k per bus, minimizing payload development effort by providing standard commercial interfaces and accepting non-flight-qualified payload hardware (allowing the payload to accept the risk of failure while R5 ensures personnel safety and launch and regulatory compliance), and leveraging non-standard or “higher risk” launch opportunities.

R5 has many secondary objectives including proliferating lessons learned in the hopes that they enable groups across government and industry to reduce their costs and schedules and accelerate technology demonstration, reevaluating many standard processes and protocols surrounding spaceflight hardware development, accelerating assessment of modern commercial-off-the-shelf (COTS) components for spaceflight, and developing early career engineers. As R5 is based out of Johnson Space Center (JSC) and is proximal to human spaceflight, R5 is also interested in advancing technologies relevant to in-space inspection of complex spacecraft to eventually enable autonomous inspection of crewed vehicles. R5’s primary stakeholder is the Small Spacecraft Technology (SST) program within the Space Technology Mission Directorate (STMD) at NASA.

To minimize project cost and schedule, the R5 project uses a minimum viable product approach to scope definition and uses an iterative and incremental development paradigm. Each new spacecraft contains increased capability and addresses lessons learned from prior spacecraft. To maximize the value of each mission, secondary payloads and flight objectives are added as able after the identification of a primary payload and objective. The spacecraft architecture, already designed to be inherently flexible to accommodate a broad range of payloads while minimizing the non-recurring engineering (NRE) needed to integrate them, simplifies the addition of these secondary payloads and objectives.

R5 started in 2021 as an internal research and development project at JSC. The project was able to build from the lessons learned in JSC’s Seeker CubeSat project. In late 2021, R5 Spacecraft 1 (R5-S1) was assembled, tested, and delivered for launch onboard Astra’s LV0008. R5-S1 was made entirely from COTS parts with a custom aluminum structure. LV0008 launched in early 2022 with R5-S1 onboard but failed to reach orbit. Even though a launch vehicle anomaly prevented R5-S1 from completing its mission, many lessons were still learned during development, assembly, integration, and testing (AIT) of the vehicle. During the development of R5 Spacecraft 2 (R5-S2) and R5 Spacecraft 4 (R5-S4), STMD SST became the primary stakeholder for the project.

R5-S2 AND R5-S4 MISSION OVERVIEW

R5-S4 was originally intended to build on the lessons of R5-S1 while also adding a prototype cold-gas propulsion system that was an evolution of the system flown on Seeker. With R5-S1 failing to reach orbit, R5-S4 carried the same primary objective of gathering data on the results of the ultra-lean process that R5 was pioneering. Additional details of the applied process and its results are detailed in sections below. Late into the mission schedule, a spacecraft slot opened on the launch that R5-S4 was manifested on. NASA’s Launch Services Program, the group responsible for the mission, would typically fill late-opening slots with a mass simulator. After discussing R5’s unique ability to deliver a spacecraft along a similar timeline to a mass simulator, NASA’s Launch Services Program requested the project build and deliver a second spacecraft for the mission as it would provide substantial value over a mass simulator. R5-S2 and R5-S4 were co-manifested on the NASA Educational Launch of Nanosatellites (ELaNa) 43 mission. Both were given nearly identical missions, intended to gather more flight data compared to a single spacecraft.

Several schedule changes enabled R5 to completely rebuild R5-S2 and R4-S4 between October 2023 and April 2024 to incorporate improvements from lessons learned during the initial AIT efforts. R5-S2 and R5-S4 were both 6U (2x4U, with “Units” of 10 x 10 x 10 cm) spacecraft and can be seen in Figure 1.

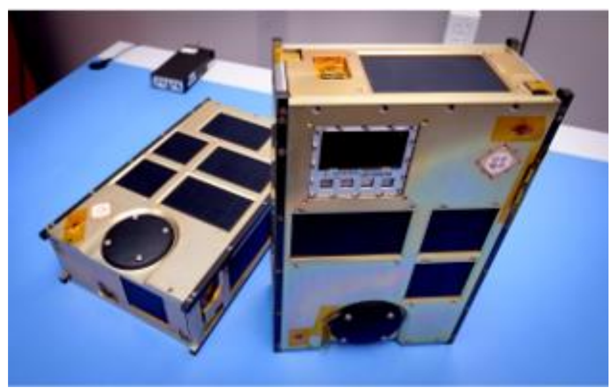

Figure 1: R5-S2 (left) and R5-S4 (right)

Objectives and Concept of Operations

The primary goal of R5-S2 and R5-S4 was to demonstrate the viability of the R5 process to rapidly and affordably deliver a functional spacecraft despite the associated increased risk. The primary objective and minimum success criteria for these initial R5 missions was for each spacecraft to power on, automatically start operations, and transmit telemetry to the ground. The secondary objectives included:

- Demonstrating spacecraft commanding over the Iridium® network
- Enveloping the performance of Iridium Short Burst Data® (SBD) links
- Demonstrating the functionality of the cold gas propulsion system
- Automatically shutting down, recharging batteries from solar panels, rebooting, and resuming the mission from the previous operational period
- Validating environment assumptions
- Imaging the launch vehicle upper stage upon deployment and downlinking the imagery
- Exercising the guidance, navigation, and control (GNC) system software and hardware

R5-S4 also hosted an Extremely Low Resource Optical Identifier (ELROI) from Los Alamos National Laboratory (LANL). An ELROI is effectively a spacecraft license plate, allowing for the independent identification of spacecraft from the ground, even if the spacecraft fails to power on or is otherwise unable to communicate.[1]

The mission ConOps was intentionally developed to be simple and sequential, enabling the mission to be completed automatically by the spacecraft without ground commanding or intervention (with the exception of the command enabling the propulsion system). Details of the high-level concept of operations (ConOps) are as follows:

1. Deployment and initial operations
    a. Deploy from upper stage
    b. Immediately power on, initialize systems, and begin capturing images
    c. After expiration of RF transmission delay timer, begin transmitting telemetry
    d. Begin operating star tracker
    e. Demonstrate commanding
    f. Wait for command to start propulsion system demonstration
    g. Complete propulsion system demonstration
    h. Continue gathering and downlinking data until spacecraft powers off
2. Nominal operations
    a. Recharge battery
    b. Power on and automatically resume last mission phase
    c. Resume gathering data
    d. Receive and execute commands
    e. Continue gathering and downlinking data until spacecraft powers off

R5-S2 and R5-S4 System Overview

The R5 spacecraft is functionally organized into subsystems: avionics, communication, flight software (FSW), GNC, propulsion, and structures. A detailed description of each subsystem is covered in the subsystem section below. This section describes the R5-S2 and R5-S4 systems at a high level to provide sufficient context for the intervening sections.

R5-S2 and R5-S4 utilized nearly identical designs apart from the ELROI payload on R5-S4. ELROI is a completely self-contained unit with only a structural interface to the system (no power or data interface). As such, its inclusion only affected the structural layout of spacecraft systems and the number of solar panels its host could accommodate as ELROI requires line of site from itself to the exterior of the spacecraft. Given the minimal differences between R5-S2 and R5-S4 and the desire to simplify the description of their mission, further discussion of the vehicles below will discuss their design as if they're identical.

Figure 2 shows the exploded view of R5-S4. Both spacecraft contained:

- Industrial-grade COTS flight computer (quad-core intel Atom, 4 GB DDRL3 RAM, 64 GB eMMC) for all data processing and system control
- COTS 18650 lithium ion cells in a custom battery pack for energy storage
- Rugged COTS solar panels for energy generation
- Custom Power Distribution Unit (PDU) and propulsion controller for power distribution, regulation, and propulsion system actuation
- Rugged COTS Iridium radio for telemetry and commanding
- COTS and custom components integrated into a cold-gas propulsion system for attitude control
- COTS cameras, GPS, and IMU to provide information to enable spacecraft position and attitude estimation

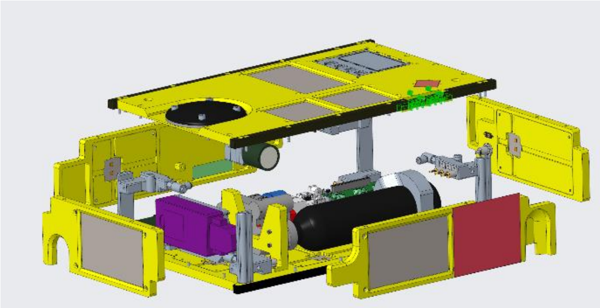

Figure 2: Exploded view of R5-S4

THE R5-S2 AND R5-S4 PROCESS HYPOTHESIS

Many traditional spaceflight projects spend considerable resources performing analysis on expected environments and resulting system performance. This analysis is typically time consuming and expensive. In situations where the environments can't be precisely quantified, conservatism is applied to the analysis which then drives system requirements (and therefore system cost). Components qualified for these expanded environments are then typically procured. These components are typically built to order with long lead times and are very expensive.

The "human element" during AIT (the inevitability of mistakes) is rarely considered as an aspect of project success even though it can have substantial consequences, like the failure to open a parachute or the toppling of the entire spacecraft.[2, 3] When these issues arise during AIT, with the large cost and lead time of space-qualified components that may have been damaged or destroyed, project schedule and cost can be substantially impacted.

R5's aggressive cost and schedule goals are incompatible with the use of many traditional aerospace products and processes. As part of the goal set of R5, the project sought data on process to show what's minimally required for successful high-risk LEO missions. R5 assessed standard NASA practices, space industry practices, and practices of adjacent industries in an attempt to correlate process and success. For R5-S2 and R5-S4, R5 hypothesized the following tenets were required to enable the successful delivery of the low cost, high performance, low lead time spacecraft.

Paramount Personnel Safety

An emphasis on personnel safety is common across most spaceflight projects. Lost personnel time due to injury is especially impactful on lean projects as there is often no backfill for personnel during their recovery, causing substantial schedule increases, and an associated cost increase with the schedule increase. Every system and process must be designed such that human error or environmental issues do not lead to harm to personnel. Risk to hardware and mission success can be accepted, but not substantial personnel risk. This is especially important to clarify when running a project perceived as high-risk. On R5, this emphasis on personnel safety manifests in several ways, including only allowing certified pressure system operators to perform hazardous high pressure operations.

Make Data-driven Decisions and Consider Context

New projects often consider historic examples as they start to define their process. Historic lessons learned can easily be incorporated into modern projects even though they may no longer be relevant. Projects also thought to be similar may have vastly different requirements or risk profiles which decrease their relevance.

As R5 defines (and refines) its process, it emphasizes data gathering above anecdotes and best practices. Prior experience is gathered and verified before being converted into process. Comparing and contrasting the context of the prior experiences versus R5's context is a useful exercise that can highlight the applicability of the experience. All additional process comes with cost, even though it can sometimes be hard to quantify. For R5 to achieve its aggressive cost and schedule goals, it requires a willingness to change and to look for direct evidence that more process is worth the effort for a particular element.

Simultaneously, a vigilance against the natural tendency to add more process for any given fault or discrepancy is required. There are instances where the resource trade favors a focus on anomaly recovery as opposed anomaly prevention. The value of process addition must be carefully considered as all process addition carries cost.

Existing best practices for process are often derived from direct experiences from failures on another projects. This is valuable perspective. However, it is important to consider these lessons within the context of the project. For example, lessons learned 30 years ago for spaceflight computers in deep space may not be relevant to a project utilizing a modern commercial flight computer in LEO. Likewise, suggestions based on models or analysis not anchored in sufficient test data can be misleading. Contextualize lessons learned or best practices for the project at hand and its risk profile to best determine relevance.

Agile Philosophy

R5's philosophy strongly aligns with the original Agile Manifesto Principles, without the formal modern Agile process.[4] These three points from the manifesto illustrate R5's prioritization philosophy:

- "Individuals and interactions over processes and tools"
- "Working systems over comprehensive documentation"
- "Responding to change over following a plan"

Hardware Rich

Traditional spaceflight projects typically have very limited hardware stock, resulting in substantial cost and schedule impacts when hardware is damaged. R5 operates with substantial hardware stock, minimizing lost schedule when hardware is damaged. R5 is able to maintain this hardware-rich approach by utilizing low-cost and low-lead-time hardware. This enables the R5 project to greatly accelerate AIT as the project can accept the increased risk with reduced process.

Part Selection

Traditional spaceflight projects usually select components based on conservative estimates of environments and extended mission lifetimes. Environmental aspects considered can include random vibration and mechanical shock environment during launch, thermal vacuum cycles, cumulative radiation effects, and single event radiation effects.

Selecting parts qualified for the above considerations usually results in low-volume or one-off highly specialized components that are very expensive and slow to produce. R5's aggressive cost and schedule targets prevent it from using traditional components and result in a focus on COTS components. R5 leverages lessons learned regarding the survivability of COTS components in the space environment along with its high risk acceptance to baseline the lowest-grade parts that meet performance and limited environmental requirements.[5] Technical performance of components is verified by performing functional checkouts of components or subsystems.

Limited environmental testing includes random vibration and thermal (and vacuum on a case by case basis). All R5 components must survive the system-level random vibration test required by the launch vehicle. These levels can vary substantially, but are usually enveloped by the GEVS qualification spectra.[6] R5 also attempts to select parts that operate between -40 to +85 C, though will accept parts that operate between 0 to +65 C if no extended temperature options are available within the project's cost and schedule limitations. If there's a substantial and credible concern about vacuum survivability or performance, vacuum testing is used to verify component suitability. As a result, R5 uses a combination of component grades including commercial, industrial, automotive, and medical. R5 uses no components that are space grade.

Component cost and schedule also drive R5 part selection. R5 intends to keep each spacecraft bus cost far below $100,000 and requires components to have a lead time less than 6 months (usually targeting parts that are stocked and ship immediately). Schedule aspects beyond sourcing are also considered in component selection, like their associated regulatory schedule implications.

To identify components that meet the aforementioned requirements, R5 performs a market survey of COTS offerings. R5 also performs a literature review of recent missions to identify any components in the survey with flight heritage and to identify other COTS components that should be included in the survey.

Make vs Buy

R5 baselines buying all components and systems. In some cases, the market survey does not identify components or systems that meet R5's performance requirements within the project's strict budget and schedule limits. In these cases, the project will develop these systems internally. These design of these systems utilize the same general methodology described in the part selection section above.

Although detailed descriptions are outside the scope of this paper, R5 is pursuing low-cost reaction control wheels made from hard disk drives, a novel modular flight software framework bult on Core Flight System, a custom avionics suite, a relative navigation bearing and segmentation algorithm utilizing Deep Learning, an open source star tracker using COTS cameras, and space to ground optical communications.

Hardware Handling Optimization

Traditional spaceflight projects build process around protecting the hardware as hardware damage carries a large cost and schedule penalty. R5's novel paradigm emphasizing robust and low-cost and low-lead-time hardware enables a relatively dramatic reduction of process around hardware interaction that greatly accelerates both development and AIT.

R5 develops and integrates hardware in a shirt-sleeve environment as opposed to a clean room, maintaining a foreign object debris (FOD) Awareness level of FOD control. Most components, especially enclosed COTS units or assembled systems, are handled without gloves or electrostatic discharge (ESD) protection. This reduces the time taken to don/doff coveralls, verify the function of protective equipment, and other standard activities performed when entering traditional AIT environments. Similarly, performing AIT in an office-like environment removes the significant cost of installing, operating, and maintaining traditional AIT facilities.

Some activities require more rigorous controls and those are applied as needed. For example, propulsion system components are sensitive to even the low level of FOD in the AIT area. As such, they are assembled within a laminar flow bench and capped until they're installed into the spacecraft. Pressure systems, including ground support equipment, also utilize filters to support the adaptive level of FOD management.

Flight avionics hardware is generally cleaned and conformal coated before integration to prevent damage from FOD or handling. Even with these preventative measures, the R5 process has resulted in a few instances of FOD related issues on test boards. However, for R5, the time and money required to maintain a truly FOD free environment is far greater than the cost of dealing with these isolated issues. This focus on improving hardware robustness and being hardware rich allows for considerably higher fidelity integrated testing, such as physically bringing the spacecraft or hardware testbeds outside to test the connection with the spacecraft communication constellation, GPS, and solar charging.

Uncompromising Emphasis on Integrated Testing

As explained above, traditional spaceflight projects typically have limited hardware stocks. This typically necessitates building and maintaining analog test environments for hardware such as simulations and analysis tooling, which can be time intensive and insufficient for substantially reducing risk. The use of analog systems for development can also lead to unexpected issues during AIT due to even subtle differences between the analog and the flight system.

In R5's hardware rich environment, engineers can perform development and gather data directly on their own set of target hardware. This significantly reduces the risk inherent in models and analysis not matching the true behavior of the hardware. Additionally, it greatly increases development and iteration speed since engineers have immediate access to hardware.

With the approximations and simplifications in many analytical and numerical models used to assess system performance, unexpected system behavior in AIT is common on most spaceflight projects. As AIT occurs relatively late in the project lifecycle, if the development period overruns, integrated testing may be cut. R5 emphasizes a "data first" philosophy that prioritizes acquiring test and flight data over thorough analysis to avoid these pitfalls. This testing occurs throughout the development cycle with a variety of hardware configurations until the entire integrated system can be tested.

To accelerate system integration, R5 utilizes Flat Satellites (FlatSats) instead of Engineering Development Units (EDUs). To gain the benefits EDUs provide to the assessment of the spacecraft structure (not including environmental qualification), 3D printed structural prototypes are used to verify tool access and component fit before metal fabrication. FlatSats are built on optical breadboards to support fast system reconfiguration and to provide a robust platform that supports the system as it is reconfigured and taken to various locations for testing. All components are easily accessible in their FlatSat configuration, accelerating iterative testing and troubleshooting (unlike EDUs).

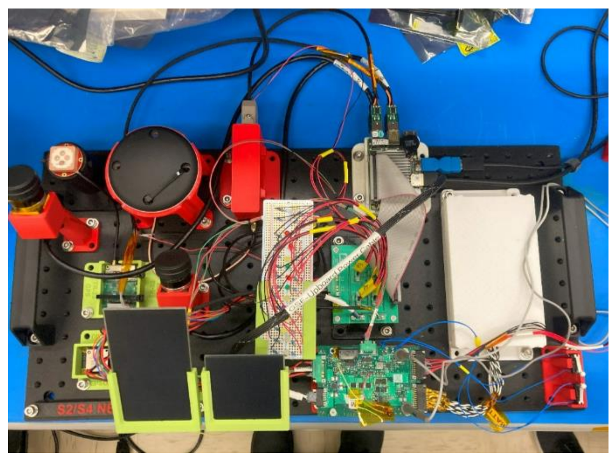

Figure 3: FlatSat used for R5-S2 and R5-S4

R5 uses FlatSats for system development and verification. FlatSats are used for FSW development, supporting rapid iteration and enabling the identification of interface issues and hardware nuances early in development. FlatSats also enable power system verification, which is accomplished by running all subsystem loads in their most demanding duty cycle and ensuring the system supports the power draw. Day-in-the-life (DITL) testing, which operates the system through mission phases in a flight-like configuration, is also performed on FlatSats as the final verification before the FSW load is deployed onto the spacecraft.

On the spacecraft themselves, R5 repeats the functional checkouts demonstrated on the identical FlatSat hardware to verify consistent performance. R5 then performs the DITL testing on the spacecraft, which includes live-sky portions to verify RF connectivity, star tracker function, and battery system charging from the solar panels. These tests move risk from the technical performance of the system to its environmental survivability.

Aside from limited component thermal cycling and separate vacuum testing, the only environmental testing that R5 baselines is random vibration. R5 uses either the launch vehicle-specified spectra or GEVS acceptance, whichever is greater. This environmental test approach leverages experience from other projects and results in a savings of approximately $50k and 10 weeks of schedule.[7]

Figure 4: DITL live-sky testing of R5-S2 (upper right) and R5-S4 (center right)

When schedule overruns inevitably occur, R5 reduces the system's technical scope as opposed to missing deliveries or reducing integrated testing. Rideshare manifests often prevent manifest changes close to launch and remanifesting is a substantial cost that's incompatible with R5's low cost goals. R5 minimizes the overall impact of the changes to the system's technical scope as instead of outright descoping, the capability can be moved to a future R5 launch. This is possible as the minimal cost and schedule enables multiple launches.

Figure 5: Integration of R5-S2 into dispenser

Refocused Reviews

Traditional NASA research and technology projects can have 13 or more reviews. R5 significantly reduces formal reviews while keeping peer reviews to maximize the technical benefit from reviews while minimizing their overhead. R5 has three formal gating reviews (Mission Concept Review, System Acceptance Review/Flight Readiness Reviews, and After Action Review) meant to ensure that stakeholders concur that the mission concept is valid, the completed spacecraft is acceptable, and finally that lessons learned are captured and incorporated.

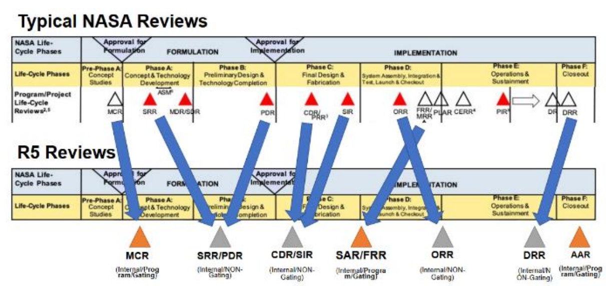

Figure 6: R5 review process

Abandon Spacecraft Preconceptions

Traditional space systems share many commonalities in their construction and ConOps, including using direct-to-Earth communications, relying on ground commanding for many mission activities, operating in a power-positive way, and designing systems to be optimal for the environment or mission.

R5 has carefully evaluated what payloads minimally require for successful demonstration and has worked aggressively to adapt the less critical aspects of the spacecraft and mission architecture to minimize cost and schedule as opposed to maximizing technical performance.

Most payloads don't require active commanding from the ground and can accomplish their demonstrations with automated input managed onboard the spacecraft. With a moderate scope increase for the addition of the appropriate level of spacecraft automation for the payload, real-time operations can be descoped. This substantially reduces the complexity of planning operations and enables the mission to proceed with minimal ground control. Most demonstration payloads also don't require substantial amounts of data to be downlinked to prove they have successfully operated. Reduced data requirements coupled with the lack of a real-time control requirement result in the ability to use alternative communication architectures, like Iridium. R5 uses Iridium for data downlink and commanding. Instead of scheduling a ground pass (a scarce resource) and managing data downlink across a short pass window, R5 continuously broadcasts and is able to consistently downlink data and receive commands even without actively pointing the bus. While data rates are generally much lower than direct-to-Earth methods, contact/coverage durations are much greater.

Most payload demonstrations can usually be completed within a few orbits and don't require continuous uptime. Since the payload doesn't have to operate continuously, the spacecraft also doesn't have to. R5 leverages the benefit of being power negative as much as possible. R5 does not have any expensive or mission critical deployment mechanisms and is able to use far lower efficiency solar cells at great cost and schedule savings. Another benefit of being power negative is the consistent power cycling of the bus, which can clear system errors.

R5 further leans into the constraints and benefits of the Iridium communication architecture and being power negative with the level of automation implemented on the spacecraft. The spacecraft are able to power on, automatically start or resume their mission, send data, receive and execute commands, and cease operations all without any active ground intervention. This, coupled with the above, precludes the need for a dedicated mission operations center, reducing project facility requirements and cost.

While many traditional spacecraft optimize their design for the environment and/or mission, R5 instead explicitly accounts for human interaction with the hardware on the ground during development and AIT in its design. This consideration enables deeper and easier interaction with the system in all phases of design and AIT by a greater variety of personnel. For example, by using a desktop operating system (Ubuntu) on the primary flight computer, more project personnel could create, collect, and save data intuitively through the graphical user interface without having to learn how to use a command line interface.

R5-S2 AND R5-S4 MISSION RESULTS

R5-S2 and R5-S4 completed their full success criteria and several additional stretch goals during their 300 day mission duration. This was far beyond their expected performance given the lean process, speed, and bus parts cost of less than $38,000 (per bus). The details of on-orbit performance are captured in each subsystem's section below, though salient highlights include:

- Powering on and automatically started the mission without ground input/command
- Demonstrating Iridium radios operate in the space environment with minimal modifications, including sending telemetry and receiving commands, and repeatedly sending one kilobyte SBD messages
- Demonstrating the Seeker-derived propulsion system holds pressure and successfully actuated all jets, addressing the suspected Seeker failure mode and demonstrating an undetectably low leak rate across the mission duration
- Successfully hosting the LANL ELROI payload, which was used to identify R5-S4 from ground observations alone
- Estimating a ~6 deg/s tip off rate from downlinked camera images on R5-S2
- Estimating a 3 deg/s or 6 deg/s tip off rate for R5-S4 from ELROI observations
- Validated environmental assumptions, hardware robustness, and minimal environmental test plan
- Demonstrated graceful failure of the GNC system
- Identified performance limitations of the star tracker system

The successful delivery and operation of R5-S2 and R5-S4 proved the R5 hypothesis, validating many of its assumptions and assertions.

R5-S2 AND R5-S4 LESSONS LEARNED

After delivering and operating R5-S2 and R5-S4, various process changes were made for the next set of spacecraft in development, R5 Spacecraft 3 (R5-S3) and R5 Spacecraft 5 (R5-S5). Cross-cutting lessons learned are described below, though specific lessons learned for each subsystem are captured in each subsystem's section.

Increase Inter-Subsystem Communication

Improving inter-subsystem communication reduces interface disconnects during assembly. After some subsystem-to-subsystem interface mismatches on the R5-S2 and R5-S4 spacecraft, regular “system engineering tags” were introduced, which greatly increased inter-subsystem communication and reduced interface errors. This is an example of how R5 applies the Agile Manifesto principle of "Individuals and interactions over processes and tools.” R5 continues to maintain minimal documentation, which exacerbates these disconnects, but does reduce project schedule. R5 also attempts to collocate the team as much as possible to increase organic communication.

Work Towards System Traceability and Control

For R5-S2 and R5-S4, there were build configuration management errors during assembly. Some propulsion components were installed into the incorrect spacecraft. The FlatSat hardware and software could change without notice, which lead to assumptions about its configuration that prolonged troubleshooting. For these spacecraft, basic paper procedures generated in word processor software and stored in common electronic file storage systems and spreadsheet-based inventory tracking was used. Using this system was very time intensive and provided poor version control. After the delivery of R5-S2 and R5-S4, the project assessed several modern tools that supported system traceability and selected Epsilon3 for use. With these modern tools, projects can now develop and execute procedures for all interactions with hardware that automatically update the part history for the associated inventory. Actions are automatically timestamped, artifacts (like images) can be directly added to procedures, and multiple users can simultaneously collaborate on the same procedure (including commenting on steps and having discussions on anomalies). This greatly improved tracking also reduces overhead compared to traditional approaches as these web-based tools provide more intuitive version control and interfaces than the traditional file storage systems.

Track Discrepancies and Create an Open Reporting Culture

R5-S2 and R5-S4 encountered several discrepancies during final AIT. These issues were all quickly caught and addressed due to the open reporting culture and Discrepancy Report (DR) process. Improvements in the data captured in the DR providing more context and detail on the anomalous event itself has further improved troubleshooting outcomes and reduced troubleshooting time. As a project, R5 consistently communicates with its personnel to speak up if they believe there’s a problem and describes DRs in relation to hardware instead of personnel. R5 also ensures there is no reprimand for reported DRs. This emphasis allows personnel to come forward more easily and readily to report issues, allowing them to be caught earlier.

Rather than focusing on blaming individual engineers, R5 uses a more systematic view of the problem. As soon as a DR occurs, all context surrounding it is captured immediately. Engineers are praised for creating thorough descriptions of the event, including images, videos, written documentation, thought process, initial root causing, and comprehensiveness for a DR they produce. This approach significantly disincentivizes engineers from hiding or ignoring anomalous behavior in the system, resulting in greater information sharing as the team iterates. This also allows for considerations to changes of process that include root causes outside of the technical, such as fatigue.[8]

Questions supporting DR resolution include:

- What was the exact context of this DR? This includes the physical state of the system, mental state of the user, and state of the process.
- What can be changed about the system design or process to prevent this issue in the future?
- Are there any more broadly applicable lessons or changes that can be applied here? Has normalization of deviance occurred?
- Is that change necessary? Is it more or less effort than tolerating this type of DR occurring? In the R5 hardware rich environment, not all DRs should create process additions.

This approach reenforces team psychological safety while maximizing the amount of information that can be gained while quickly iterating on a high-risk system.

For R5, using collaboration tools like GitLab Issues allow the team to quickly collect context surrounding a DR and actively track troubleshooting steps in a single place. Actions can be dispositioned and traced back to the DR and to a corresponding e-procedure or inventory item. Collecting or linking all the relevant information maximizes the context the team can pull from and apply to the solution. This also allows for continuation of the data capture and discussion if the DR repeats in the future.

Verify All Relevant Component Specifications

Whether it’s an incorrect coordinate frame, misplaced decimal, optimistic maximum power draw number, or other oversight, many interface control documents or specification sheets contain errors. Assuming they are correct and developing hardware/software based on these documents without verifying them inevitably leads to schedule delays during AIT as issues become apparent. R5 attempts to identify these discrepancies by integrating and testing hardware as early as possible.

Account for Human Error in AIT Planning

Human error cannot be completely eliminated and should be accounted for in project planning just like technical performance and environmental robustness. Design the system and process to be tolerant to things like rough hardware handling. Maintaining a hardware rich environment and selecting components robust to handling minimizes schedule impacts due to error. Accounting for these challenges and as a result cultivating a hardware rich environment saved the schedule on multiple occasions.

Test Like You Fly

Utilize flight-like configurations on test systems (e.g., FlatSats), especially for final testing. This includes things that are perceived as subtle including harnesses lengths, grounding, and live skies. Harness lengths and differences between operating the system with and without connectivity to the Iridium constellation proved significant to R5's integrates system performance.

Verify Component and System Polarity

Polarity check tests vehicle sensors and effectors highlight sign errors, frame mismatches, and other subtle yet critical errors that can result in the loss of a mission. This type of test caught incorrect thruster mapping on R5-S2 and R5-S4 prior to delivery, enabling the successful propulsion demonstration.

DETAILED SUBSYSTEM DESCRIPTIONS

Avionics Subsystem Description

A detailed walkthrough of the avionics subsystem and associated lessons learned can be found in an associated paper.[9] The R5-S2 and R5-S4 avionics subsystems were divided into the following functional groups: flight computer, power, propulsion control, and peripherals and payloads.

R5-S2 and R5-S4 each used an UP Board UP-CHT01-A20-0464-A11 as their primary flight computer. This single board computer (SBC) features an Intel Atom x5-Z8350 system on a chip with an Intel Gen8 HD400 internal GPU, 4 GB of DDR3 memory, and 64 GB of eMMC storage. This SBC features many common interfaces, including USB 2.0, HDMI, USB 3.0, ethernet, UART/SPI/I2C serial interfaces, and up to 28 general-purpose input/output (GPIO) pins.

A custom power distribution unit (PDU), COTS solar panels, and COTS battery cells integrated into a custom pack comprised the power system. Figure 7 describes the architecture of this system. The PDU enabled charging and monitoring the batteries, protection against power faults, and providing 5 V, 12 V, and 3.3 V power rails for the rest of the avionics. The PDU could support a current draw of 10 A. Each spacecraft contained a custom tab-welded 1-series, 6-parallel cell pack consisting of Samsung 30Q 18650 3,000 mAh Li-ion batteries.

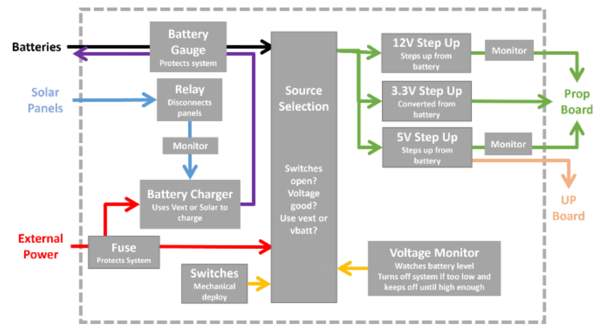

Figure 7: Power architecture

Propulsion system control and electrical interfaces were managed by a custom propulsion controller board, colloquially known as the "prop board." The board took signals from the flight computer and commanded corresponding valves and systems.

The spacecraft each contained a Ximea MQ013 camera, a Ximea MC023 camera, a NAL Research A3LA-RS Iridium modem, and a JAVAD TR-2S GPS connected to the flight computer by USB. The Epson M-G370PDS0 IMU onboard used a UART serial interface routed through the prop board to connect to the flight computer.

Avionics On-Orbit Performance

In their first operational period, both spacecraft responded to commands from the ground and completed a propulsion system demonstration, successfully demonstrating the function of all parts of the propulsion system. Upon running the battery down to the PDU's shutdown threshold, the spacecraft powered off and the batteries began to recharge. Hours later, R5-S2 and R5-S4 automatically powered back on and resumed the mission, completely demonstrating the avionics systems. R5-S4 would complete a total of 103 operational cycles with more than 170 hours of operations and R5-S2 would complete a total of 48 operational cycles with more than 130 hours of operations.

Fewer avionics onboard measurements were downlinked than expected. Limited fault analysis suggests this is due to a software error that inconsistently caused the FSW application that collected this telemetry to crash. Despite this, sufficient data has been gathered to draw conclusions about the environment and the system's performance.

The spacecraft gathered temperature data from the PDU, flight computer, and battery pack, validating environment assumptions. R5-S2's battery pack temperature was measured to be between 19 °C and 54 °C, while R5-S4's was measured to be between 22 °C and 55 °C, both of which are within the Samsung 30Q's rated discharge temperature range of -20 °C to 75 °C. The batteries did not have any active thermal control systems. See Figure 8 for downsampled battery temperature data.

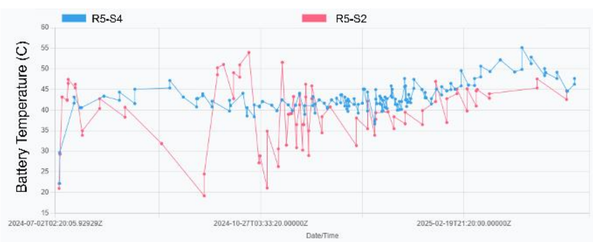

Figure 8: R5-S2 and R5-S4 battery temperature vs time

Across the mission, R5-S4's maximum CPU temperature (read from the operating system) was 90 C with an average temperature of 82.65 C. R5-S2's maximum CPU temperature was 90 C as well with an average temperature of 82.69 C. The maximum temperature and elevated average suggest that the CPU was thermal throttling, though explicit data to confirm this was not recorded or downlinked. See Figure 9 for downsampled CPU temperature data.

Figure 9: R5-S2 and R5-S4 CPU temperature vs time

During the spacecraft's initial operational period, where their batteries were at their highest state of charge for the entire mission, R5-S4 operated for approximately 5.4 hours and R5-S2 operated for approximately 5.9 hours. This data is the time of the last telemetry message from the spacecraft during the first operational period and not the true duration of the period as it wasn't recorded. This is the best data available to assess a minimum operational time for each operational period, though it must be evaluated with the caveat that many other factors (like Iridium constellation geometry) can impact successful data transmission beyond spacecraft power. Evaluating all subsequent operational periods and binning the data, R5-S2 had a minimum operational duration mode of 3.1 hours and R5-S4 had a minimum operational duration mode of 2.85 hours.

Across the mission, the spacecraft had their battery recharge time calculated by differencing the time their last received telemetry message of the prior operational period was received with the time the first telemetry message of the current operational period was received. This overestimates the recharge time as the last telemetry message received isn't necessarily when the spacecraft powers off, the first telemetry message doesn't always immediately get delivered, and there are five instances where a spacecraft operational period did not successfully telemeter any data. With these caveats, the average estimated recharge time for R5-S4 was 74 hours (standard deviation of 66.4 hours) with a maximum of 360 hours and the average estimated recharge time for R5-S2 was 135.5 hours (standard deviation of 130 hours) with a maximum of 645.5 hours.

Utilizing the onboard time recorded in the telemetry packets sent vs the ground time at which the packets were received over the course of the mission, R5 was able to conservatively bound the observed clock drift of the spacecraft over time. At the start of the mission (after keeping time for approximately 80 days), both spacecraft had less than 80 seconds of clock drift. By the end of the mission (after keeping time for approximately 380 days), R5-S4 had less than nine minutes of drift and R5-S2 had less than seven minutes of drift.

Avionics Subsystem Lessons Learned

A complete list of avionics system design and lessons learned can be found in the associated paper.[9] What follows is a small selection of the most important lessons.

- Some common printed circuit board (PCB) design and analysis software assumes convective cooling environments which are not applicable to spaceflight designs
- Connectorized harnesses accelerate AIT more than the time taken to design and fabricate them
- Thoroughly inspect all custom components upon receipt to identify nonconformances

- before integration (even those with inspection or test reports)
- Thermal cycling can identify solder workmanship errors quickly
- Spring clips are not sufficient for battery encapsulation and connectivity in GEVS vibration environments
- Power protection systems (like fuses) should be placed as close to power sources (like batteries) as possible
- Specification sheets provide a useful starting place for design, but their information should be verified in integrated testing as soon as possible to identify and address errors/omissions
- In environments where designs can change quickly (like iterative environments), out-of-date hardware and information must be carefully managed to ensure it doesn't get used in the wrong context

Communication Subsystem Description

The communication system consisted of a NAL Research A3LA-RS modem (based on the Iridium 9523) and a single NAL Research SAF2040-B antenna. This was the only radio onboard and was used for telemetry and commanding. The Iridium network's SBD system allows for low-rate bidirectional data without having to schedule time on the network. The network is also able to successfully transfer SBD even while the bus isn't actively being pointed. R5-S2 and R5-S4 would attempt to send telemetry packets (80-150 bytes in size) approximately every 30 seconds. The spacecraft were also able to transmit variable packet sizes on command and able to complete a packet size demonstration sequence that would send, in order, a 50, 100, 300, 500, and finally 1000 byte message. While the 9523 advertises a maximum SBD message size of 1960 bytes, the 1000 byte maximum in the demonstration sequence was implemented due to a lack of time required to test the larger message size. The network endpoints for the spacecraft were configured to be email with the telemetry data transferred as attachments. Sending commands (mobile terminated messages) to the spacecraft utilized a similar mechanism, which the Iridium network would then buffer and deliver when the spacecraft powered on and successfully completed an SBD session (note that there is a limit on the time during which the Iridium network stores the messages).

The A3LA-RS had its power routed through a switching circuit for power control with a power resistor to reduce inrush current to a level compatible with the switching circuit. The radio, used in the 3-wire configuration, sent and received data from the flight computer using a serial-to-USB converter. The FSW interacted with the modem using AT commands and ensured the radio only received power if the RF license duration (stored onboard) had not expired. The A3LA-RS had its internal components staked/conformally coated with DOWSIL 3145, but otherwise had no modifications (the electrolytic capacitors were not replaced).

Communication Subsystem On-Orbit Performance

Both spacecraft successfully transmitted telemetry and received commands across their entire 300-day mission duration, downlinking a combined total of 897 kB (including nearly 100 images). With the exception of the brief propulsion system demonstration at the start of the mission, the spacecraft never controlled their attitude. This successfully demonstrated not only the use of a COTS Iridium 9523-based device over a 300-day mission, but also the robustness of the system and network to poor pointing/tumbling. With all spacecraft telemetry routed to email, spacecraft insight and operations were able to be completed anywhere email was available.

Across the mission, R5-S2 attempted approximately 46000 SBD sessions and successfully completed approximately 3200, while R5-S4 attempted approximately 76000 and completed approximately 3700. In addition to uncontrolled attitude and unknown Iridium satellite constellation geometry (optimized for terrestrial connectivity), the R5 FSW attempted to create SBD sessions faster than recommended by Iridium best practices and may have triggered traffic management, all of which likely contributed to the low SBD session completion rate in space. For both spacecraft, the vast majority of these session attempt failures were outliers with 77% of all SBD session attempts succeeding on the first attempt. See Figure 10 for the attempts required per successful SBD session represented as a histogram and Figure 11 for the associated time between successful SBD sessions similarly represented.

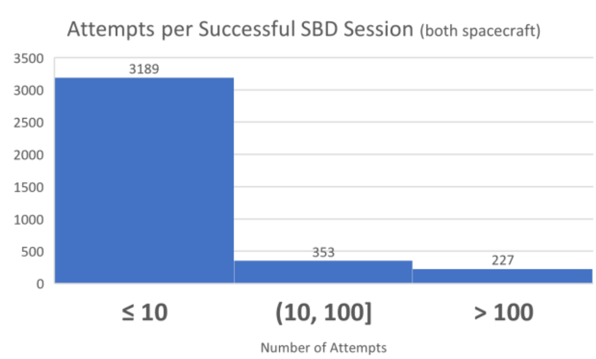

Figure 10: Attempts per successful SBD session

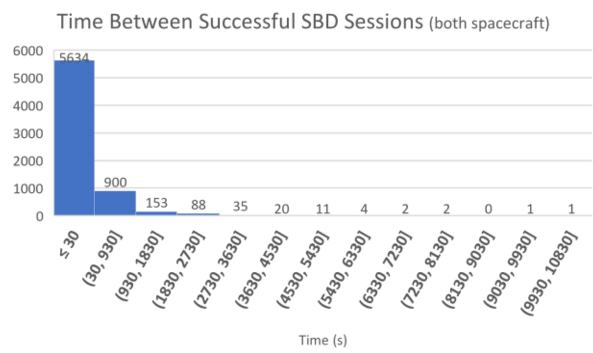

Figure 11: Time to complete a successful SBD session

Seventeen SBD packet size demonstration sequences were completed across both spacecraft between July of 2024 and January of 2025. Most of these sequences completed downlink of all five messages in their sequence within two minutes. Notably, there was no strong correlation between message size and SBD session failure. At the time of the first demonstration sequence in July of 2024, the successful transmission of a 1 kB message is believed to be the largest single mobile originated Iridium message ever sent from space.

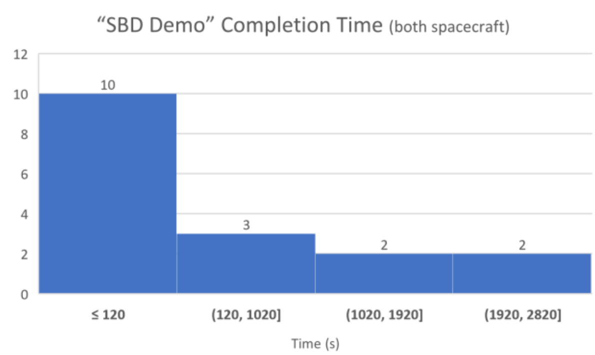

Figure 12: "SBD Demo" completion time

A signal strength command was sent to the modem and the result added to the SBD message loaded onto the modem before each SBD session attempt. Sessions that successfully completed had their data recorded when received on the ground. Signal strengths measured while SBD sessions failed to complete were not recorded. Signal strengths measured before successful SBD sessions varied substantially, with 15% of all successful SBD sessions reporting a signal strength of 0 bars immediately before the SBD session initiation, as shown in Figure 13.

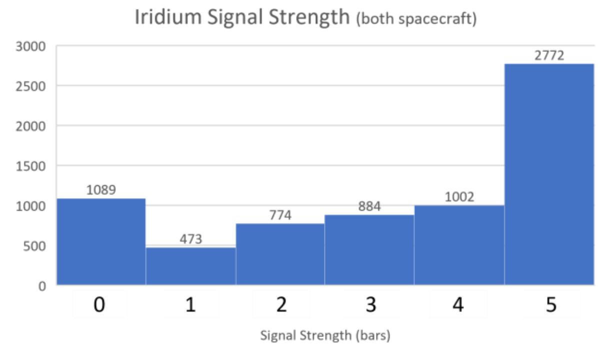

Figure 13: Signal strength before successful SBD sessions

Communication Subsystem Lessons Learned

- The type of communication system selected can drive the licensing timeline. Licensing timelines are the most important selection criteria for short timeline spacecraft projects.
- Constellation-based communication systems can be highly robust to poor pointing or tumbling
- COTS Iridium devices may be suitable for space use with minimal modification
- Iridium 9523-based systems can repeatedly successfully send 1000 byte SBD in orbit

Flight Software Subsystem Description

The FSW subsystem for R5-S2 and R5-S4 utilized a custom Python-based flight software framework called Python Flight Software (PyFS). This framework was created to imitate aspects of the NASA Core Flight System (cFS) framework utilized extensively at NASA.[10] The core objectives were to accelerate FSW development, especially for non-expert users/developers, and to provide a path to transition to cFS in the future.

Architecturally, PyFS is designed to be similar to cFS. They both utilize a publish subscription architecture between applications in the framework. All applications can publish or subscribe to messages on a common software bus. This includes telemetry, output data, and commands. In many cFS systems, a scheduler application is used to “wake up” applications periodically at a given frequency. This is replicated in PyFS with its own scheduler. PyFS similarly attempts to replicate other cFS patterns.

This replication was intended to allow for transitioning from a PyFS codebase to a cFS codebase after sufficient project maturity and support. This was done for the R5-

S3 and R5-S5 vehicles, which utilize cFS but leverage the application logic, GNC algorithms, and IO handling from their PyFS predecessor. Some applications, such as the Star Tracker (ST) application, were designed to be directly called by a cFS application to ease the transition between the two architectures.

The FSW contained various GNC applications for pointing, hardware input/output (IO), and high-level managers such as the Scheduler and Automated Flight Manager (AFM). The AFM automatically configured the system through all mission phases, allowing for automated execution of modes. FSW modes that enabled actuation of the propulsion system were gated by ground command.

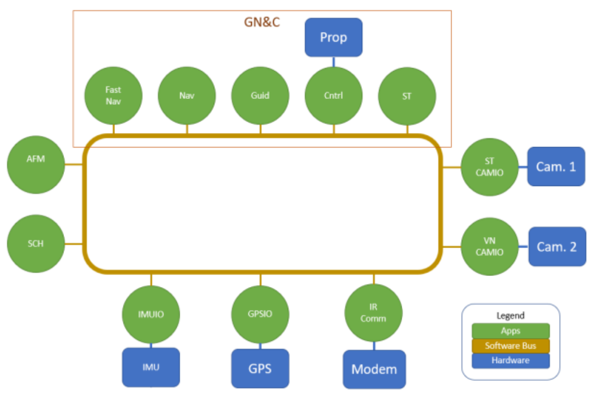

Figure 14: R5-S2 and R5-S4 PyFS Architecture

Flight Software Subsystem On-Orbit Performance

The FSW for R5-S2 and R5-S4 successfully executed its primary objective in addition to various secondary objectives. All flight software applications with the exclusion of the GPSIO and IMUIO applications executed as expected. IMUIO did not receive any nonzero IMU data in orbit, which may be from rough hardware handling during environmental testing or an intermittent bug resulting from poor hardware configuration. GPSIO had mixed results, running without issue for a short period of time at the beginning of each operational cycle, but eventually ceasing to output data. This was likely due to a software bug, but was not root caused due to the team transitioning to cFS for the subsequent flights.

RAM and CPU utilization was highly variable, with RAM utilization between 30%-85% and CPU utilization typically around 95% after boot and then quickly stabilizing to between 35% to 70%.

All commanding, moding, and telemetry (other than telemetry sent from the GPSIO application) successfully routed through the system and down to the ground.

Flight Software Subsystem Lessons Learned

- PyFS tripled the number of contributing developers on the FSW team due to Python’s inherent ease of use and memory guardrails compared to C. GNC, Project Management, Avionics, and interns could all make contributions to FSW code under the stewardship of the FSW Lead. This accelerated all aspects of development, but especially bug identification and fixing during AIT.
- Custom frameworks require more time debugging fundamental infrastructure.
- PyFS continues operating if a single application crashes (unlike cFS).
- Many bugs that would be illuminated at compile time are not caught. A lack of compilation and static typing also leads to non-deterministic performance as all the various applications run together and dynamically allocate memory.
- Even though Python has significant performance overhead compared to other languages like C, there was effectively no impact on the system due to the relatively high performance of the primary flight computer.
- Utilizing a desktop environment, such as an Ubuntu Linux operating system allowed for greater accessibility to non-expert users for integrated testing and troubleshooting. More usable debugging and software development environments such as VS Code could be utilized directly on the flight computer. Simple operations such as accessing files were easier for the team due to a desktop file explorer being present.
- The code present in the FSW and ground software can be easily shared due to the nature of Python. Scripts used for checkout can be directly pulled from FSW or vice versa without requiring substantial build system knowledge. This greatly improved ground testing software development.
- Software version control should be present on the flight computer to enable better traceability of changes.

Guidance, Navigation, and Control Subsystem Description

On R5-S2 and R5-S4, the goal of the GNC system was to reduce risk for future R5 missions by demonstrating navigation hardware, rotational rate damping, and inertial attitude pointing. These GNC modes could be activated by ground command.

The R5-S2 and R5-S4 GNC system was comprised of several sensors and FSW applications with the propulsion subsystem owning the effectors. A JAVAD TR-2S GPS receiver connected to a Tallysman TW1421 antenna provided latitude, longitude, and altitude information from the unit's GPGGA message. An Epson M-G370PDS0 IMU provided the change in angle and the change in velocity of the unit. A Ximea MC023 camera with a Fujinon HF35HA lens collected images which were then processed by the NASA "COTS Star Tracker" software.[11] While it wasn't "in the loop" for the GNC system, a Ximea MQ013 with a Fujinon HF35HA camera was used to path find image collection for relative navigation. Aside from the associated FSW IO applications, GNC software was split into a "nav" application that updated vehicle state estimates with a Multiplicative Extended Kalman Filter (MEKF), a "fastnav" application that propagated the state estimates using IMU data, a "guid" application that provided spacecraft guidance, and a "control" application that commanded thruster actuation based on the guidance output.

The Star Tracker system was intended to explore on-orbit performance and was configured to process images with exposure times of 50 ms, 100 ms, and 250 ms every 20 seconds. The camera had its distortion parameters measured prior to launch and parameterized with the Brown model. The imagery was converted from greyscale to binary, the resulting blobs centroided, and the largest 20 blobs' centroids were then processed in sets of three to match to the star catalog. Details of this process can be found in the paper describing the algorithm and some preliminary performance.[12]

The MEKF implementation was heavily influenced by the Orion spacecraft's attitude navigation filter design. The MEKF had an update and a propagate step. Propagation was done by directly measuring the IMU at 50Hz. The update step was dependent on the star tracker system.

For rate damping mode, guidance simply sent a command to the control system to target a null rate. For the celestial object pointing mode, the guidance system utilized an ephemeris of the target (specifically, in this case, Earth's moon) location and the system clock. The difference between the sensor boresight vector and the spacecraft-cg-to-target vector was calculated and sent to the control system.

The control system utilized a phase plane control scheme. The rate limit of this system was determined based on the slew rate at which star tracker performance degraded during ground testing. Once a target rate and orientation error was provided by the guidance system, the controller would attempt to reduce the error by commanding paired jet firings by the propulsion system. A jet pair per control cycle were fired for up to 200ms. This simplified the jet selection logic to simply pick the jet combination with the greatest alignment with the error in the phase plane for each control cycle.

Guidance, Navigation, and Control Subsystem On-Orbit Performance

Approximately 30 seconds after deployment, both R5 spacecraft captured 30 images over approximately 30 seconds with their deployer-facing relative navigation cameras (Ximea MQ013). The spacecraft were likely more than 60m away from the Firefly Alpha upper stage by the time they initialized. The cameras were configured with automatic gain and exposure, resulting in the first two images being completely black (as expected). R5-S4 eventually downlinked all 30 relative navigation images, which contained light artifacts sweeping across the field of view. R5-S2 eventually downlinked 24 relative navigation images (skipping images 1,2, 25, 26, 27, and 28), which also contained light artifacts, but also showed a white spot that moved consistently frame-to-frame. This spot may be the upper stage, but there isn't insufficient information to say with certainty. The motion of the dot within the field of view was used to calculate a tumble rate of approximately six degrees per second. To downlink the images, they were compressed via JPEG and downscaled.

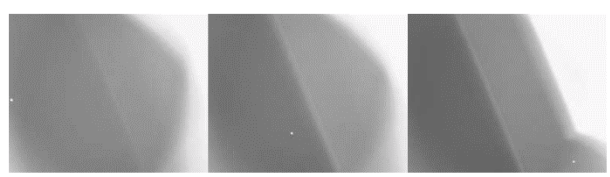

Figure 15: R5-S2 image sequence showing what may be the Firefly Alpha upper stage

Throughout the mission, the star tracker was actively acquiring images and attempting to produce quaternions from them. R5-S4 acquired more than 99000 star tracker images with an average solve attempt time of 2.2 seconds and R5-S2 acquired more than 82000 star tracker images with an average solve attempt time of 1.9 seconds. R5-

S2 downlinked 1623 star tracker solutions of which only 6 were successful. R5-S4 downlinked 1846 star tracker solutions of which only 11 were successful. These solutions were chosen at random across the mission and also represented a fairly even distribution across all image exposures (with no obvious relationship between exposure level and solve success or failure). Sufficient supporting data required to verify the accuracy of the successful solves was not downlinked. In general, this performance was far worse than the system's performance during live sky testing on the ground. At least part of the poor performance is likely due to the tumble rate. Additional analysis and improvements to the star tracker algorithms (like the addition of more detailed failure insight to the telemetry) is ongoing.

During AIT, the IMUs occasionally failed to establish communication with the flight computer, but at the time no root cause was determined. After spacecraft delivery, a software bug was identified which made successful initialization of the IMU dependent on initial conditions of the UP Board's UART lines. This likely resulted in the failure to communicate with the IMUs. On the ground, this behavior was seen more often in R5-S2. In orbit, this led to the decision to command the pointing demonstration on the R5-S4 spacecraft and not the R5-S2 spacecraft. After commanding the demonstration, the spacecraft downlinked GNC data captured every 10 seconds throughout. The data shows that the system gracefully failed—the IMU never provided any information and as a result the GNC system did not actuate any effectors.

As mentioned above, much less GPS data was downlinked than expected, likely due to a software bug. Of the 489 valid GPS telemetry packets downlinked from R5-S2, only 102 contained position fixes. On R5-S4, of 539 valid packets, only 39 contained position fixes. For both spacecraft, the number of satellites averaged 8-10 for each position fix and the fastest time to first fix was approximately 490 seconds on R5-S4 and 75 seconds on R5-S2.

Guidance, Navigation, and Control Subsystem Lessons Learned

- Tumble rates can significantly degrade star tracker performance and deployment tipoff rates may be greater than what star trackers can handle
- Inter-subsystem communication regarding interfaces is critical. For the GNC subsystem, this includes software and hardware considerations.
- Polarity verification for all sensors and effectors is critical

Propulsion Subsystem Description

R5-S2 and R5-S4 utilized a cold-gas propulsion system for attitude control. This is an unusual choice for small spacecraft attitude control, but it's strongly aligned with JSC's longer-term goal of developing small extravehicular free-flying inspection capabilities compatible with crewed spacecraft. On this mission, the goal of the propulsion system was to demonstrate its core functionality to reduce risk for future missions and inform future design iterations.

The design for the R5 propulsion system heavily leveraged lessons learned from the Seeker CubeSat launched in 2019.[13] A scripted series of thruster firings was executed on-orbit to exercise the propulsion system by firing jet pairs so that every jet was actuated, jets were held open long enough to register a propellant tank pressure drop, and that there was no resulting translational motion. The propulsion system was also designed to support the moon-pointing demonstration.

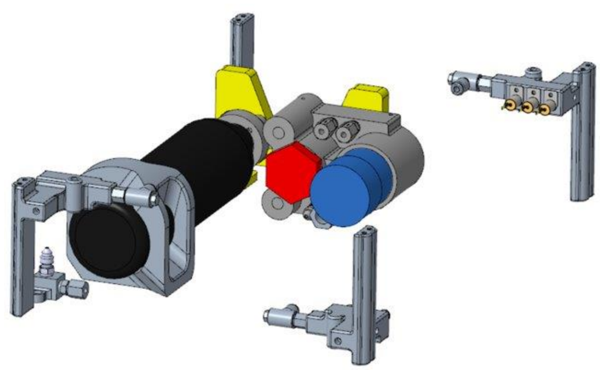

Figure 16: Propulsion system components

The R5-S2 and R5-S4 propulsion systems were a high-pressure cold gas N2 blowdown system using a mini-regulator with 12 thrusters for six degree of freedom control, high performance isolation valve, relief valves, and an ISO/DOT-certified paintball tank. All components upstream of the thruster valves were mounted in a NASA-designed manifold, and each set of three thrusters were mounted in a NASA-designed 3D printed aluminum thruster pod. Thrust is achieved by the actuation of the isolation valve and thruster valve providing flow through a jet insert, which is a set screw with a conical converging-diverging nozzle laser drilled inside. Placement of the thruster pods and thruster directions enforces pairs of thrusters to be fired for pure rotational or translational motion. This is intended to reduce the consequence of a single stuck-on thruster during proximity operations.

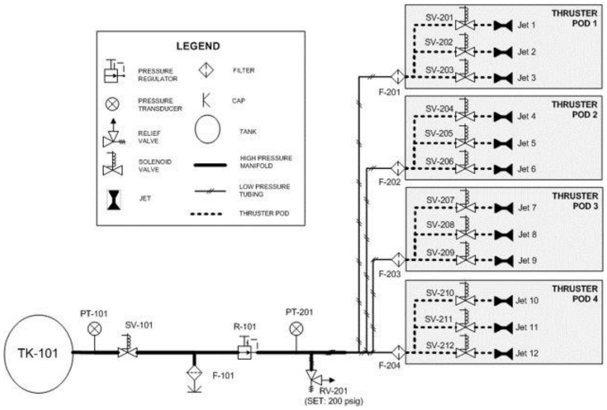

Figure 17: Propulsion system fluid schematic

For R5-S2 and R5-S4 and all other currently planned R5 vehicles, a Tippmann 13 cubic inch aluminum paintball tank with a flight load pressure of 2500 +/- 100 psig is used as the propellant tank. This keeps the maximum expected pressure at the maximum expected temperature in the tank below the 3000 psig design pressure. With the tank's ISO/DOT rating, R5 can claim, on a lot-by-lot basis, that the tank is built to 2.4x burst with high confidence. Using a tank built to federal/international standards saves a significant amount of time compared to developing and qualifying a novel high pressure tank. The largest downside is the unoptimized tank mass, but it can be easily tolerated within the R5 bus as the bus is otherwise well below the mass limit for 6U spacecraft.

The following three steps are required for integration:

1. Purchase and receive tank
2. Remove the regulator from the tank
3. Clean interior

This greatly simplifies what is typically a lengthy process of checkouts and verification that would be required for non-ISO/DOT-rated tanks.

The isolation valve, regulator, relief valve, thruster valves, tubing, and fittings were similarly all COTS parts. Many were identical or similar to the Seeker vehicle, leveraging their verified performance in orbit. Others borrowed from high performance terrestrial analogs such as the medical grade Parker C7 valves used for the thruster valves. Those unqualified components were purchased in large quantities and screened with acceptance testing.

The manifolds, thruster pods, and jets were custom designed at JSC. These components were designed to interface with the COTS hardware in the propulsion system, be leak tight, have a 4x factor of safety to burst, and be easily manufacturable by publicly available manufacturing companies. The designs are currently going through the NASA Technology Transfer Program to be publicly available.

The system was extensively tested, including:

- All GFE components were proofed and precision cleaned
- All components passed wear-in, thermal, and functional checkouts
- Jet insert testing characterized flow rate for thruster performance
- Thruster valve lot testing eliminated out-of-family valves
- Integrated testing verified analysis for system performance. Analyzed total on-time, lock-up, and regulator performance
- Isolation valve temperature testing verified valve will not overheat during usage
- Final assembly leak check verified no gross overboard leakage
- Functional checkouts prove all 12 jets operational and provided data to compare to on-orbit performance

Propulsion Subsystem On-Orbit Performance

The propulsion demonstration was completed by both spacecraft in orbit. After the demonstration, the telemetry received from both vehicles showed that the pressure drop during each thruster firing was in family with expectations from ground testing. This proved that the entire system, including every jet, actuated as expected.

Over 300 days in orbit, the propulsion system had a leak rate under the detectable limit of the pressure transducer of 1E-4 Standard Cubic Centimeter per Minute (SCCS).

Propulsion Subsystem Lessons Learned

- Being hardware rich enabled the rebuild, extra testing, and selection of best components.
- Design iteration has led to steady improvements such as the addition of filters where possible, smaller pressure transducers, and easier access to the fill port
- Replace filters with a "Last Chance" style design. The current filters with tiny, thin threads, are a larger FOD threat than no filter at all.

- Fill line, manifold, and fill port should be mocked up early in AIT, fabricated, and then sent to precision cleaning. The fill line was created late in the process and could have been cleaner if it was made earlier.
- Make two of everything to have spares: Manifold, thruster pod, fill port, fill line, any GFE
- Have a standardized system for testing in terms of procedures, data collection, reporting, etc. A process was developed for the thruster valves that can be extended to other components
- Bench testing of valves should be done with flight like computers to avoid extra confusion in the data collected (like switch bounce back)
- Prop needs more communication with FSW to build into testing how we plan to use the valves in flight.
- Thermal testing of the isolation valve was done due to concerns regarding the solenoid coil overheating. This turned out to be a non-issue for R5-S2 and R5-S4, but may return with new manifold designs.

Structures Subsystem Description

The R5-S2 and R5-S4 structure consisted primarily of Al 6061-T651/T6. The spacecraft used a "sandwich" design where the majority of components were mounted to a large "top" and "bottom" plate with the intent of simplifying access during AIT compared to traditional CubeSat assembly designs. Given R5's extensive use of COTS components, there was a wide variety of fastener types and mounting patterns across the components. To simplify assembly, 3D printed polyetherimide brackets and adapters were used to ensure all components were adequately attached to structure. Components consisting of or substantially featuring PCBs (e.g., primary flight computer) were only connected to structure at their mounting holes (usually in the corners of their rectangular form factors) and without any additional thermal conductivity (e.g., no thermal straps). In order to maintain critical dimensional tolerances, shims were used. Most fasteners were designed to have two locking features, typically using helical thread inserts in addition to preload.

All of the structure was a custom NASA design and the aluminum plates were CNC machined at a commercial fabrication center. The 3D printed brackets were done in-house. All custom aluminum structure was passivated by a commercial coating vendor to JSC process specification PRC-5005 Type I Class 3 (alodined) with the outer surfaces that interface with the dispenser rails anodized per JSC PRC-5006 Type III. Fasteners were acquired from a variety of vendors including McMaster-Carr Supply Company. Where two locking features weren't possible, fasteners were staked with DOWSIL 3145. Even without any consideration for mass optimization, the spacecraft were far under the 6U CubeSat limit with R5-S2 weighing 7.26 kg and R5-S4 weighing 7.44 kg (loaded with propellant).

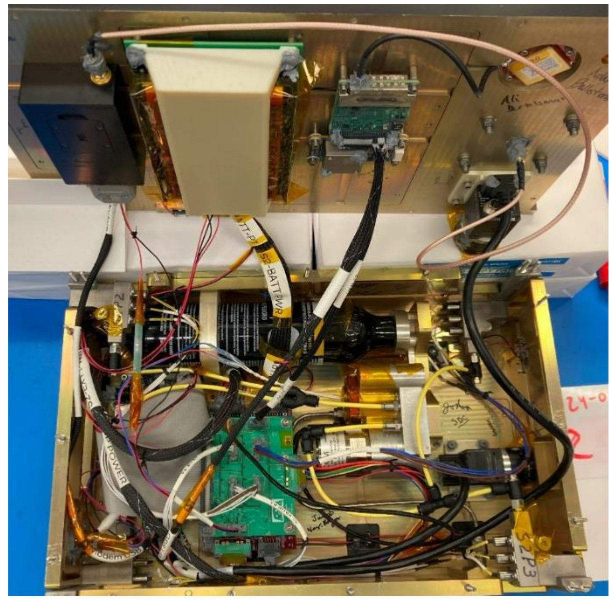

Figure 18: R5-S2 interior

For the R5-S2 and R5-S4 launch, the required random vibration testing levels were less than the GEVS Acceptance test definition, so the GEVS Acceptance test definition was used. A sine sweep was also performed before and after each axis. These tests were performed at a DynaQual Test Labs, a commercial testing vendor. R5 uses a custom vibration test fixture that is intended to mimic a CubeSat dispenser, which allows the project to perform vibration testing with fewer constraints across a variety of missions and dispensers. During their development, R5-S2 and R5-S4 went through six random vibration tests each, with one done to GEVS Qualification levels and duration.

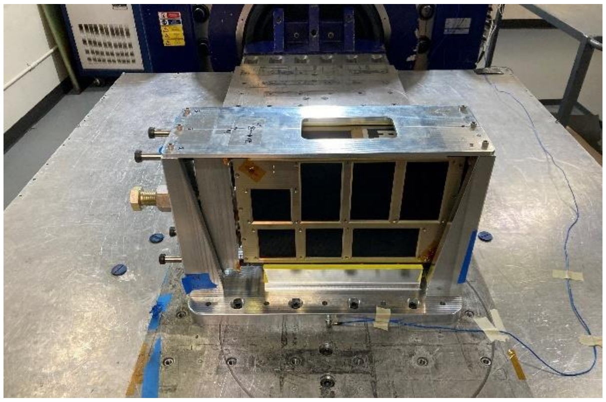

Figure 19: R5-S4 during random vibration testing

Structures On-Orbit Performance

The structure on both spacecraft survived the launch environment and successfully deployed, meeting all mechanical requirements. As both spacecraft operated across their entire 300 day mission, the approach to thermal management was validated.

Structures Subsystem Lessons Learned

- Identify design/ICD errors/mismatches by performing fit checks as early as possible. One way to do this is to complete a 3D printed fit check with as many real components as possible. This identifies differences between CAD and actual component dimensions, tool access challenges, and other integration concerns. A 3D printed fit check also provides an opportunity to draft assembly procedures and to provide physical measurements of required wire harness lengths.
- Avoid using shims wherever possible. Shims take a considerable amount of time to apply and drove critical path when the spacecraft were opened and closed. Shims may also not have been required as each dispenser handles CubeSat geometry tolerances differently and R5 used the CubeSat Design Specification instead of the dispenser ICD. Future R5 vehicles will attempt to accept looser tolerances on vehicle bounding dimensions (as long as they conform to the dispenser requirements).
- Inter-subsystem communication regarding interfaces is critical. Ensure that components or assemblies developed by other subsystems are inspected from a structures' point of view to catch process escapes with fasteners or other areas within structures' domain.
- PTFE washers compress significantly under load. During installation of the GPS and IMU boards, the PTFE washers in the IMU fastener stack and GPS fastener stack deformed when torqued even though the load was distributed by steel washers in the stacks. PTFE washers were replaced with fiberglass washers, which have a higher compressive strength and resolved the issue while providing similar electrical insulation.
- Helical thread inserts require careful consideration in design and manufacturing, especially with small threads. R5 selected shorter inserts than the recommended length due to space constraints. This, combined with the small size of the 4-40 and 2-56 fasteners used across the spacecraft and what may have been tolerance challenges on the insert holes resulted in many instances during assembly where the running torque on the fasteners was too low or where the holes were too small for the inserts. In the future, R5 will consider using larger fasteners, different methods to provide a secondary locking feature for fasteners, longer helical thread inserts, and considering manufacturers that can provide tighter tolerances where required.
- Consider AIT spacecraft configurations when defining harness lengths (as opposed to just the flight configuration). The cabling lengths were not sufficient to enable spacecraft disassembly without disconnecting harnesses and resulting in challenging assembly steps with limited access.

CONCLUSION

R5 has successfully rapidly and affordably designed, built, tested, delivered, launched, and operated two spacecraft for their full mission lifetime of 300 days. These spacecraft contained no space-qualified parts and all of their components cost less than a total of $38,000 (per bus). This suggests that the lean process R5 is pioneering may be suitable to enable risk-accepting programs to substantially reduce their cost and schedule for getting nascent technologies demonstrated in orbit. R5 was able to do this with an iterative and incremental development paradigm creating many lessons learned along the way.

R5 will continue to share its findings with the community with the goal of enabling others to enhance their spacecraft capabilities while also reducing their cost and schedule. R5 is highly motivated to partner across government and industry in order to accelerate the infusion of lessons learned. Seven additional spacecraft are planned over the next few years, which will demonstrate additional bus capabilities and refinements of the R5 process.

Acknowledgments

R5-S2 and R5-S4 were funded by the Small Spacecraft Technology program within NASA's Space Technology Mission Directorate and Johnson Space Center's Engineering Directorate. R5 is actively pursuing new public-private partnerships and mission-focused funding partners across the US government. The authors would like to thank our families for the support through the late nights and weekends. In addition, the authors would like to thank the many CubeSat Launch Initiative, KULR Technology, Firefly Aerospace, Voyager Technologies, Iridium, and DynaQual personnel that helped get these vehicles across the finish line and into orbit. The authors would also like to thank all R5 team members and support personnel past and present for enabling the success of the mission.

References

1. Palmer, D.M., et. al., "Progress Towards the ELROI Satellite License Plate", Proceedings of the Small Satellite Conference, Logan, Utah, 2020 https://digitalcommons.usu.edu/smallsat/2020/all2020/134/
2. Ryschkewitsch, Michael; et. al., "GENESIS Mishap Investigation Board Report Volume I," National Aeronautics and Space Administration, 2005 https://www.nasa.gov/wp-content/uploads/2015/01/149414main_Genesis_MIB.pdf
3. Scolese, Christoper; et. al., "NOAA N-PRIME Mishap Investigation Final Report," National Aeronautics and Space Administration, 2004 https://www.nasa.gov/wp-content/uploads/2015/01/65776main_noaa_np_mishap.pdf?emrc=8ee322
4. Beck, Kent; et. al., "Manifesto for Agile Software Development," 2001 https://agilemanifesto.org/
5. Leitner, Jesse; et. al., "COTS EEEE Parts: What Do the Data and Experiences Tell Us?", National Aeronautics and Space Administration, 2023 https://ntrs.nasa.gov/citations/20230001450
6. "GSFC-STD-7000 GENERAL ENVIRONMENTAL VERIFICATION STANDARD (GEVS) For GSFC Flight Programs and Projects.", National Aeronautics and Space Administration, 2013 https://explorers.larc.nasa.gov/2019APSMEX/MO/pdf_files/gsfc-std-7000a_final_3-28-18.pdf
7. Estes, Jay; et. al., "SPLICE Flight Hardware Environmental Test Qualification Program," AIAA Scitech 2021 Forum, Virtual, 2021 https://doi.org/10.2514/6.2021-0257
8. Dekker, Sidney (2014). *The Field Guide to Understanding 'Human Error'*. Ashgate.
9. Knesek, Kathryn.; et. al., "Avionics Design Architecture for Low-Cost CubeSat Missions and Lessons Learned from R5-S2 and R5-S4" Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, 2025 https://ntrs.nasa.gov/api/citations/20240016467/downloads/2025_IEEE_Aerospace_R5_avionics_final.pdf
10. NASA, "Core Flight System (cFS)," [Online]. URL: https://github.com/nasa/cFS. [Accessed 4 June 2025].
11. NASA, "COTS Star Tracker [Online]. URL: https://github.com/nasa/COTS-Star-Tracker. [Accessed 4 June 2025].
12. Pedrotty, Sam.; et. al., "DESIGN AND PERFORMANCE OF AN OPEN-SOURCE STAR TRACKER ALGORITHM ON COMMERCIAL OFF-THE-SHELF CAMERAS AND COMPUTERS," Proceedings of the AAS GNC Conference, Breckenridge, Colorado, 2020
13. Pedrotty, Sam.; et. al., "SEEKER FREE-FLYING INSPECTOR GNC FLIGHT PERFORMANCE," Proceedings of the AAS GNC Conference, Breckenridge, Colorado, 2020