Subsystem Integration and Reliability in a High-Power Rocket Payload

Honors Thesis

Presented In Partial Fulfillment of the Requirements for the
University Honors Program
Colorado State University

By

Alex Benson
Department of Electrical and Computer Engineering

Kari Cowden, *Department of Mechanical Engineering*
Lucy Krips, *Department of Electrical and Computer Engineering*

Spring Semester 2026

Contents

Abstract

This thesis investigates subsystem integration and its impact on reliability in a high-power rocket payload developed for NASA's Undergraduate Student Launch Initiative (USLI). High-power student rockets operate under severe mechanical, thermal, and electrical constraints, where failures frequently emerge at subsystem interfaces rather than with individual components. As part of a multidisciplinary USLI team, an electrical payload subsystem was designed, integrated, and tested within a full-scale competition rocket.

The work focuses on the interaction between electrical, mechanical, and software subsystems, with particular attention to power distribution, sensing, communication, and structural interfaces. A system-level integration and verification methodology was applied, including iterative design, environmental testing, and failure analysis. Observed failure modes were documented and traced to interface-level design decisions, motivating targeted redesigns and integration improvements.

Results demonstrate that subsystem integration quality is the dominant factor governing payload reliability in USLI-class rockets. The findings highlight the importance of interface definition, verification-driven design, and cross-disciplinary coordination in small-scale aerospace systems. In addition to technical outcomes, this project contributed to the development of professional engineering competencies in systems thinking, documentation, and reliability-focused design practices.

Introduction

High-power rocket systems operate in extreme environments characterized by high acceleration, intense vibration, transient thermal loading, and strict mass and power constraints. In such conditions, overall system reliability is often governed not by the performance of individual components, but by the quality of integration between subsystems. Failures in student-built aerospace systems frequently originate at the interfaces between electrical, mechanical, and software elements, where mismatches in assumptions, constraints, or implementation can propagate into mission-critical faults.

NASA's Undergraduate Student Launch Initiative (USLI) is a national engineering competition that challenges university teams to design, build, and launch a high-power rocket while adhering to NASA-defined safety, performance, and verification requirements. The competition emphasizes formal systems engineering practices, including subsystem documentation, interface control, design reviews, and verification testing, within the constraints of limited resources and accelerated development timelines. These characteristics make USLI competition rockets a representative and instructive platform for examining subsystem integration and reliability in small-scale aerospace systems.

This thesis examines the electrical payload subsystem of a competition rocket and its interaction with the broader vehicle architecture. The electrical subsystem includes power distribution, sensing, communication, and embedded control elements that must operate reliably while being mechanically constrained by the rocket structure and coordinated with software and recovery systems. Propulsion and aerodynamic performance are treated as external constraints rather than primary subjects of analysis. The focus is placed on how subsystem interfaces were defined, implemented, and iteratively refined throughout the design and testing process.

This work argues that subsystem integration quality is the dominant factor governing payload reliability in high-power rocket systems. Electrical–mechanical interfaces and power connec-

tion instability were found to be primary drivers of observed failure modes. Through system-level testing and failure analysis, this thesis demonstrates that reliability improvements are achieved most effectively through interface-focused design and verification rather than component-level optimization alone.

The primary contributions of this thesis are: (1) the design and implementation of an electrical payload subsystem suitable for high-vibration rocket environments; (2) an interface-centered integration and verification methodology applied within a multidisciplinary team setting; and (3) an empirical analysis of integration-driven failures and design iteration outcomes. Together, these contributions provide insight into both the technical challenges of small-scale aerospace system integration and the engineering practices required to address them effectively.

Background and Related Work

Embedded Systems in Aerospace

Embedded systems are the computational backbone of modern aerospace vehicles. From orbital satellites to suborbital sounding rockets, reliable embedded computing is what closes the loop between sensing the environment and acting on it. In an aerospace context, an embedded system is not simply a microcontroller running code. It is a tightly constrained integration of processing hardware, firmware, sensors, actuators, and communication interfaces, all operating within strict budgets for mass, power, and volume. These systems must execute deterministically in real time while surviving mechanical shock, vibration, thermal cycling, and electromagnetic interference. The margin for failure is low, and the cost of integration errors is high.

Liu et al. (2021) establish the foundational design concerns for embedded aerospace systems: hardware-software co-design, resource efficiency, and environmental constraint management. Their analysis emphasizes that system reliability is not achieved through component selection alone, but through disciplined co-development of hardware and firmware from the earliest design stages. Shalf et al. (2018) extend this further into high-performance embedded computing for spaceflight, examining processor selection, thermal design, and communication interface architecture. Their work demonstrates that system performance under harsh operational conditions depends heavily on how well computing, memory, and sensor layers are coordinated, not just how capable each layer is individually.

The OPS-SAT nanosatellite mission, documented by Zeif et al. (2022), provides a concrete case study of these principles applied to a real mission. The embedded computing subsystem for OPS-SAT was designed around real-time control, fault tolerance, and interface integrity under environmental stress, all priorities that translate directly to any constrained aerospace platform regardless of scale. Their approach to hardware-software integration and environmental test

campaigns reflects the same challenges faced in student rocketry, where resources are limited but operational requirements remain demanding.

More recently, the field has moved toward modularity as a guiding design philosophy. Saponara et al. (2025) present a multi-payload integration framework for nanosatellites built around standardized interfaces, coordinated power management, and modular subsystem boundaries that simplify both integration and verification. Zhang (2023) reinforces this direction, arguing that modular and reconfigurable payload architectures improve reliability by isolating subsystem failures and reducing interface complexity. Together, these works reflect a broader shift in aerospace embedded design: the reliability problem is increasingly treated as an integration problem, and the solution increasingly lies in how subsystems are defined and connected, not just how they are built. These principles, developed largely in professional and orbital contexts, are now directly informing how high-power student rocketry systems are designed and verified.

Embedded Systems in the USLI Context

NASA's Undergraduate Student Launch Initiative is a national engineering competition that challenges university teams to design, build, and fly a high-power rocket to a target altitude while satisfying a mission-specific payload requirement defined annually by NASA. The competition is structured around formal systems engineering practices, including a proposal, preliminary, critical, and final design reviews, interface control documentation, hazard analysis, and a verification and validation test campaign. These requirements impose a professional engineering framework on student teams operating under academic timelines and constrained budgets. This combination makes USLI a meaningful and representative platform for studying embedded systems integration: it replicates the core constraints of professional aerospace at a scale accessible to undergraduate engineers.

The electrical subsystem work for this project began at the subscale vehicle level. Two Arduino Nano-based subscale launches established the first integration lessons: the system survived flight loading, but incomplete data recovery and a shutdown after 5.5 hours of pre-launch standby

showed that logging reliability and battery endurance had to be treated as operational constraints rather than bench-only requirements. Those lessons directly informed the competition payload architecture discussed in later chapters.

The competition payload embedded system was designed and built in response to those lessons. The central controller is the Arduino UNO Q, running Zephyr RTOS to support concurrent sensor monitoring, data logging, and actuation control. The sensor suite combines I^2C and SPI devices for altitude, inertial state, timekeeping, and pH measurement, while motor control is handled through a Pololu VNH5019 dual motor driver shield and a relay-based safety interlock.

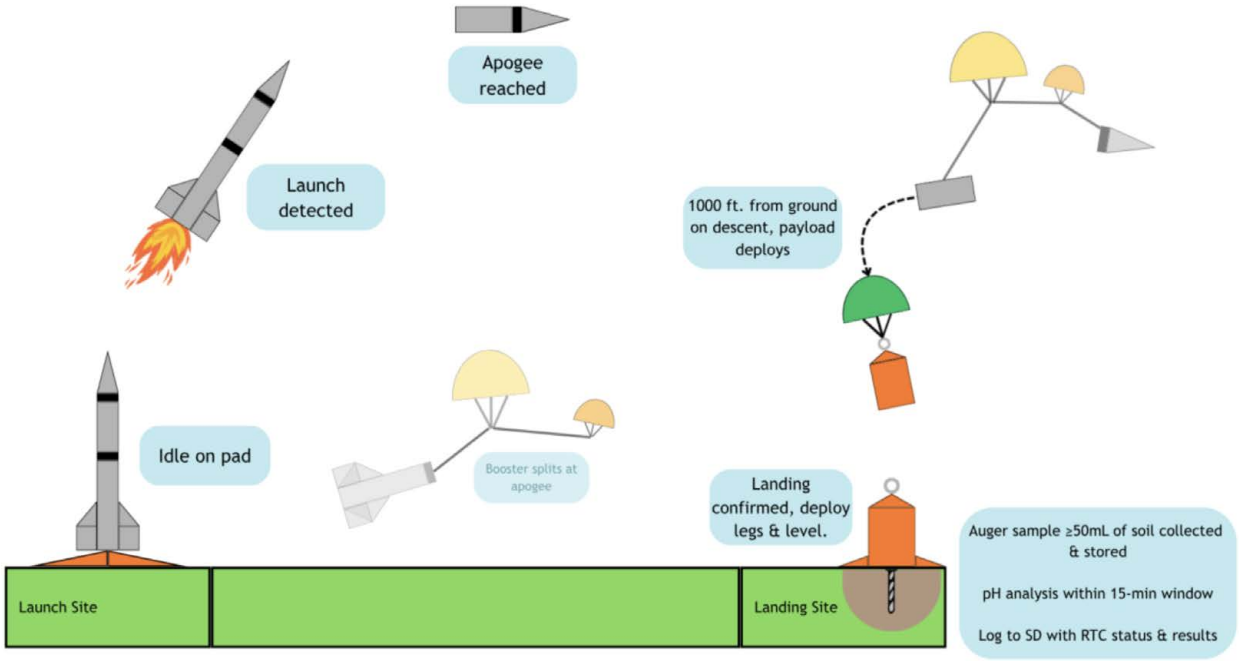


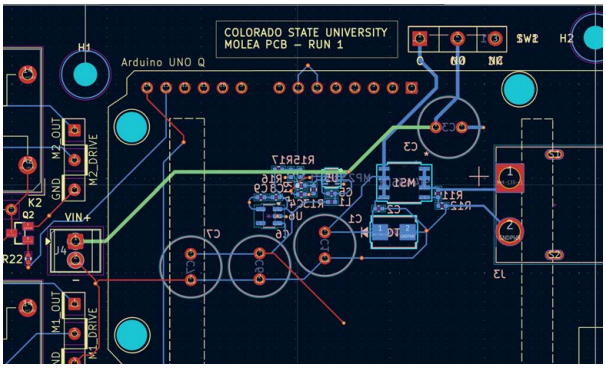
Figure 2.1: MOLEA Mission Concept of Operations

The concept of operations defines the embedded system's behavioral requirements. The payload is activated via a key switch on the exterior of the rocket prior to flight. From that point on, the fully autonomous system monitors altitude, inertial state, and time concurrently. Launch is detected through sensor fusion, and the system then waits for confirmed landing before any actuation is permitted. Landing confirmation uses a combination of IMU and barometric data to distinguish a stable ground state from other low-velocity flight phases. Once landing is confirmed, the relay is activated, Motor 1 deploys the exterior legs and drives the payload to a vertical ori-

entation, and Motor 2 engages the auger sequence to collect approximately 50 mL of soil for pH measurement. The mission concept of operations is illustrated in Figure 2.1.

The custom PCB, manufactured and assembled by PCBWay, consolidates the prototype's protoboard assemblies and loose wiring into a single integrated board. It incorporates the motor relay circuit, reduces the overall form factor to fit within the payload bay volume constraints, and eliminates the discrete microSD breakout used in the subscale version by leveraging native storage on the UNO Q. Initial validation identified a power trace continuity fault, shown in Figure 2.2, which later became an important case study in PCB debuggability and interface-centered design.

(a) Resistor-level view of broken trace location



(b) Wider context showing trace path and affected net

Figure 2.2: PCB power trace continuity error identified during bench validation

A repair attempt restored microcontroller boot behavior but did not recover the external sensor interfaces, indicating an additional continuity issue on the sensor ground path. As a result, the PCB could not be qualified for flight and the competition launches used the original protoboard assembly. The detailed outcome and design lessons are analyzed in Chapter 6.

The body of literature reviewed in Section 2.1 addresses embedded system reliability primarily at the architecture and component level. What it does not address in depth is how reliability is shaped by the quality of physical and logical integration between subsystems in small-scale, resource-constrained aerospace systems built by multidisciplinary student teams. The subscale launches and ground testing conducted in this project produced empirical failure data rooted in integration decisions: power architecture, battery endurance, interface robustness, and firmware

state management. This gap between component-level reliability literature and integration-level failure behavior in student aerospace systems is what this thesis addresses directly.

System Architecture and Design

The RAMS rocket consists of three independent sections: the nosecone, payload section, and booster section. The forward section houses the forward avionics bay, the payload integration subassembly, and the MOLEA payload. The payload integration subassembly retains the payload during flight using a Kevlar restraint cord routed through a compression spring. At 1,000 feet AGL, an electrically actuated line cutter severs the Kevlar cord, releasing the spring and ejecting the payload from the airframe. Three aluminum guide rods maintain axial alignment of the payload during ascent and through the deployment event. The payload then descends under its own dedicated recovery parachute before landing and initiating its autonomous surface operations sequence.

The MOLEA payload was constrained to a final flight mass of 8 lbs, reduced from an earlier 10 lb limit during the design process. This mass constraint directly influenced component selection, PCB layout, and overall packaging of the electronics within the payload enclosure. The payload must comply with the NASA requirement that no protrusions extend beyond one quarter inch from the airframe prior to apogee, which required all mechanisms to remain fully retracted through apogee.

The central controller is the Arduino UNO Q, running the Zephyr real-time operating system. Zephyr supports structured, concurrent task execution, which is necessary for a system that must simultaneously monitor multiple sensors and coordinate timed actuation sequences without blocking on any single operation. The UNO Q also provides native onboard storage, eliminating the discrete microSD breakout board used in the subscale design and removing one additional point of potential connection failure.

The sensor suite interfaces are documented in Table 3.1. The BMP280 barometric altimeter operates at I^2C address 0x76 and provides altitude and internal payload temperature. The PCF8523 real-time clock operates at I^2C address 0x68 and timestamps all logged events. The

Atlas Scientific EZO pH carrier board operates at I^2C address 0x63 and provides calibrated pH readings via the EZO circuit. The SparkFun ICM-20948 IMU communicates over SPI rather than I^2C, selected for its higher data throughput appropriate to the continuous three-axis accelerometer and gyroscope sampling required during flight. The Atlas Scientific pH probe uses an impedance-matched cable and an electrically isolated EZO carrier board, protecting signal integrity against motor switching transients on shared ground planes.

Table 3.1: Electronics Sensor Interface

Item	**Description**	**Purpose**	**Interface / Address**
ICM-20948	IMU Breakout Board	Accelerometer/gyroscope data	SPI
BMP280	Barometric pressure sensor	Altitude data	I^2C 0x76
EZO/pH Suite	EZO isolated board, pH carrier board, probe	pH data requirement	I^2C 0x63
PCF8523	Real-time clock	Timestamp requirement	I^2C 0x68

Motor actuation is handled by a Pololu VNH5019 dual motor driver shield. Two Pololu 2829 DC motors with 150:1 gearboxes provide the torque required for leg deployment and auger actuation within the payload enclosure. The demonstration flight power architecture used two 2200 mAh 7.4 V LiPo batteries wired in series, providing 14.8 V to the VIN of the Pololu shield, which powered the Arduino UNO Q. The Arduino provided a 5 V rail to the shield and powered peripheral sensors from its native regulated lines. For the competition launch, the power architecture transitions to a PCB-based solution with dedicated buck regulator ICs producing independent 5 V and 3.3 V regulated rails, with 3300 mAh batteries providing the extended standby capacity required to accommodate longer pad wait times. The power architecture is shown in Figure 3.1.

Under bench testing, the system performed within all defined limits. Idle current draw measured 69 mA against a limit of 247 mA. The leg motor under no-stall conditions measured 193 mA against a 200 mA limit, and the auger motor measured 160 mA. Under stall, the leg motor drew 630 mA and the auger motor drew 400 mA, both within the 700 mA stall limit. Estimated operational duration under these conditions was validated at 6.88 hours. Current draw is automatically

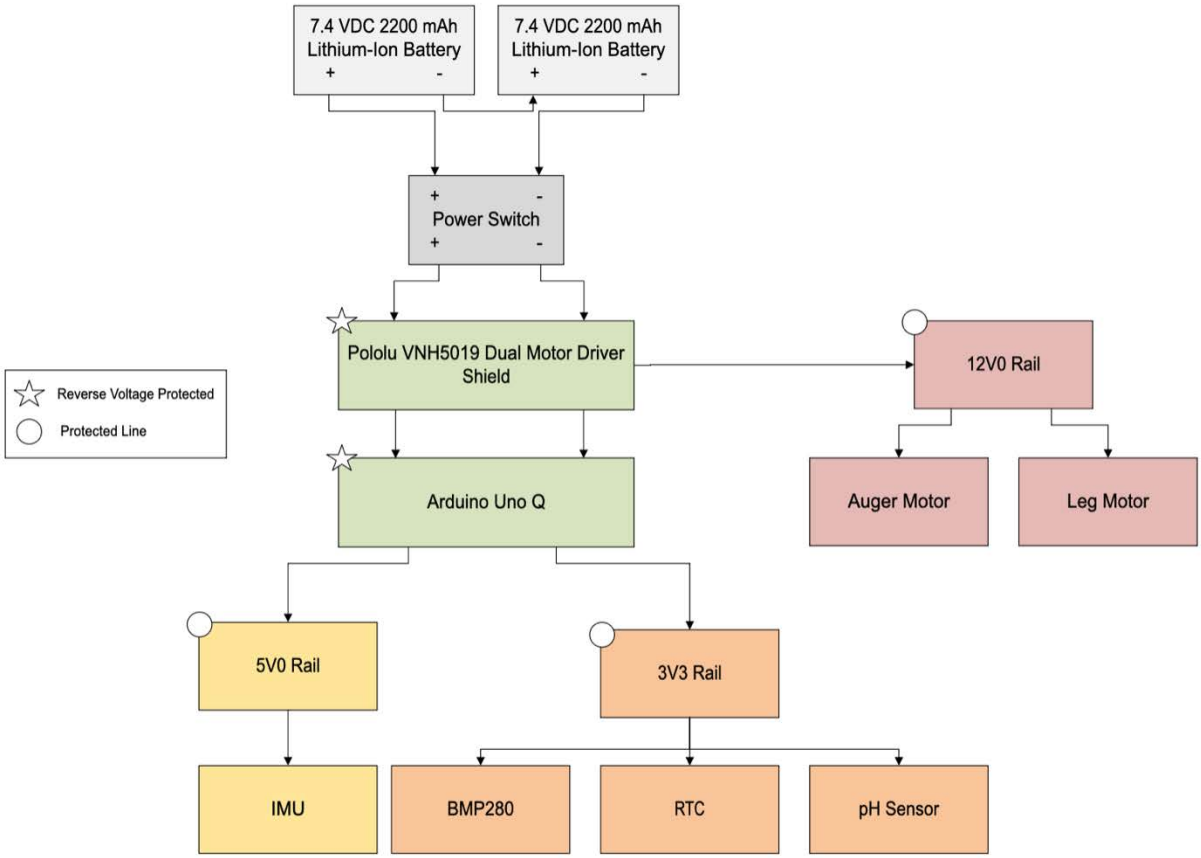


Figure 3.1: Power Architecture Block Diagram

timestamped and logged via the VNH5019 current sense feedback pins during all motor operations, providing post-flight auditability of motor operating conditions. Full results are summarized in Table 3.2.

Table 3.2: Power Structure Under Bench Testing

Condition	**Measured Current**	**Expected / Limit**	**Verification Status**
System idle	69 mA	$\leq$247 mA	PASS
Leg motor running, no stall	193 mA	$\leq$200 mA	PASS
Auger motor running, no stall	160 mA	$\leq$200 mA	PASS
Leg motor running, stall	630 mA	$\leq$700 mA	PASS
Auger motor running, stall	400 mA	$\leq$700 mA	PASS
Estimated operational duration	6.88 hrs	$\geq$6.8 hrs	PASS

The Pololu shield motor enable pins are pulled to ground such that the shield treats motors as always enabled. Motor power is gated exclusively through a hardware relay on the custom PCB. No power can reach either motor until the relay is explicitly closed by the microcontroller in the correct mission state. This ensures that even if a software fault or noise event produces an inadvertent PWM command to the VNH5019, the motors cannot be energized outside the permitted operational window. This circuit was added in direct response to the risk identified during prototyping that unintended motor actuation inside the rocket body could cause mechanical damage to the leg deployment mechanism. The motor control circuit and relay interlock are shown in Figures 3.2 and 3.3.

The custom PCB was designed in KiCad and fabricated and assembled by PCBWay. The as-built protoboard assembly measures 92.5 mm $\times$ 77.1 mm $\times$ 41.2 mm, and the projected PCB envelope is 98.52 mm $\times$ 98.68 mm $\times$ 45 mm. The PCB consolidates the protoboard assemblies and loose wiring into a single manufacturable board, integrates the relay interlock circuit, regulated power tree, sensor interfaces, and motor control, and specifies precise standoff mounting geometry for mechanical attachment within the electronics bay. PCB 3D renders are shown in Figure 3.4.

The as-built electronics integrated with the payload body in the protoboard configuration are shown in Figure 3.7.

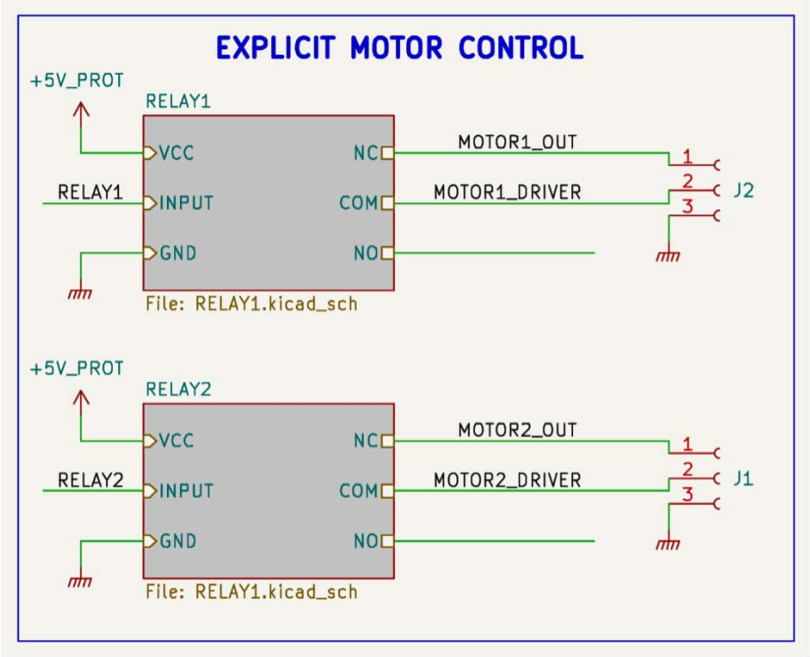


Figure 3.2: Explicit Motor Control Schematic

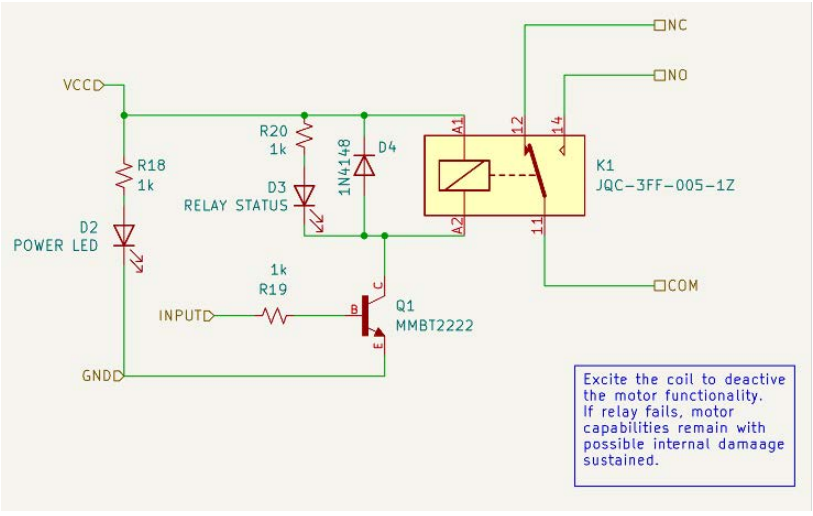
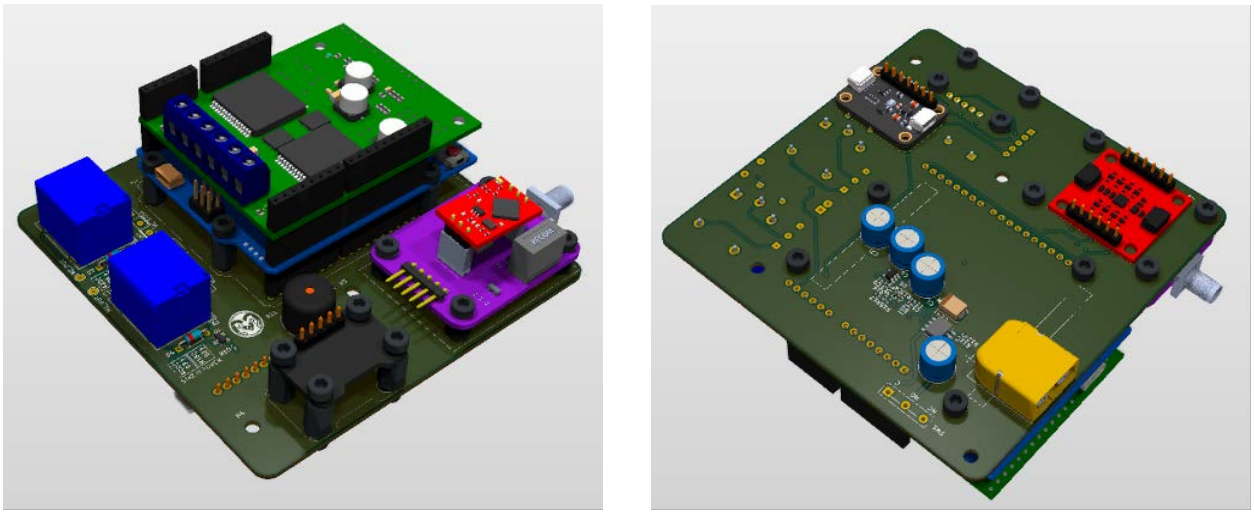
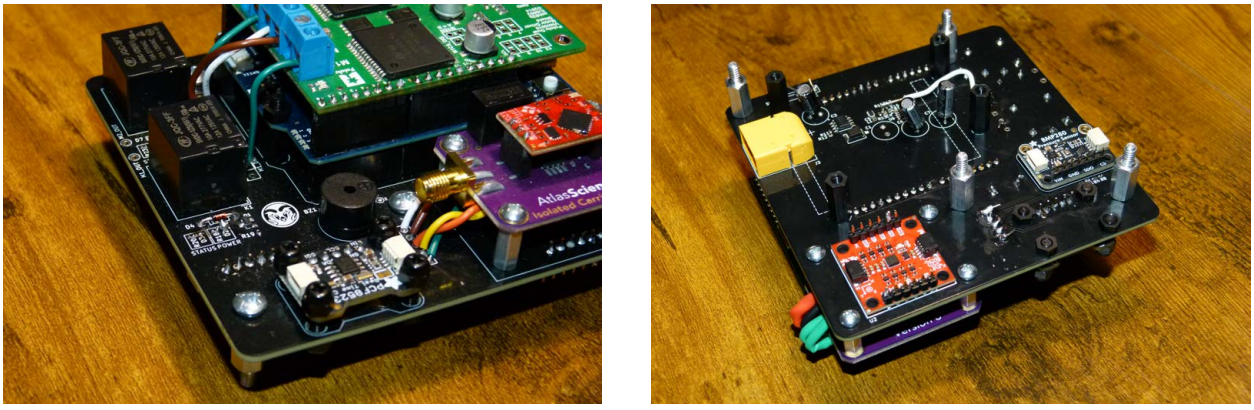


Figure 3.3: Relay Interlock Schematic (RELAY1.kicad_sch)



(a) Top Face (b) Bottom Face

Figure 3.4: MOLEA PCB 3D Renders



(a) Front face (b) Back face

Figure 3.5: As-built MOLEA PCB as received from PCBWay. Physical inspection confirmed component placement and board dimensions matched the KiCad fabrication files. The broken power trace and ground continuity fault identified during bench validation are not visible on the board surface.

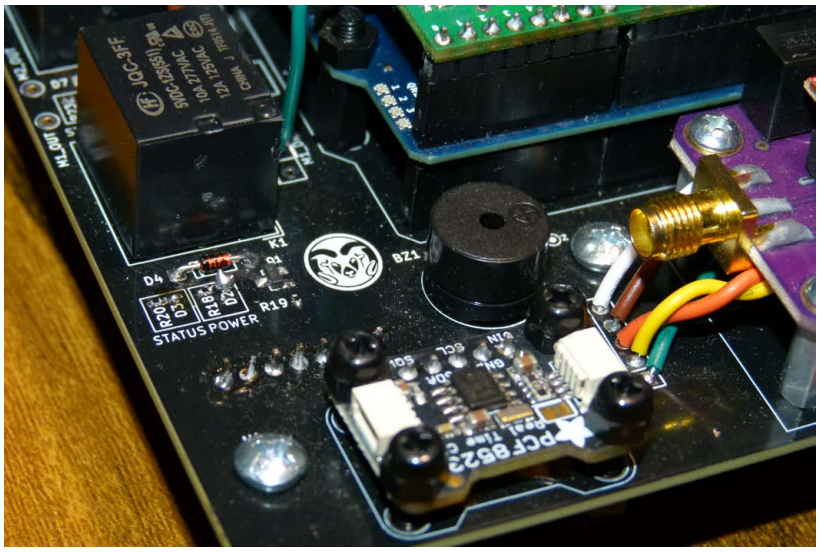


Figure 3.6: As-built MOLEA PCB, close-up of front face showing 0201 and 0402 passive component footprints. The small component scale that complicated hand-rework and continuity probing during bench debugging is visible at this magnification.

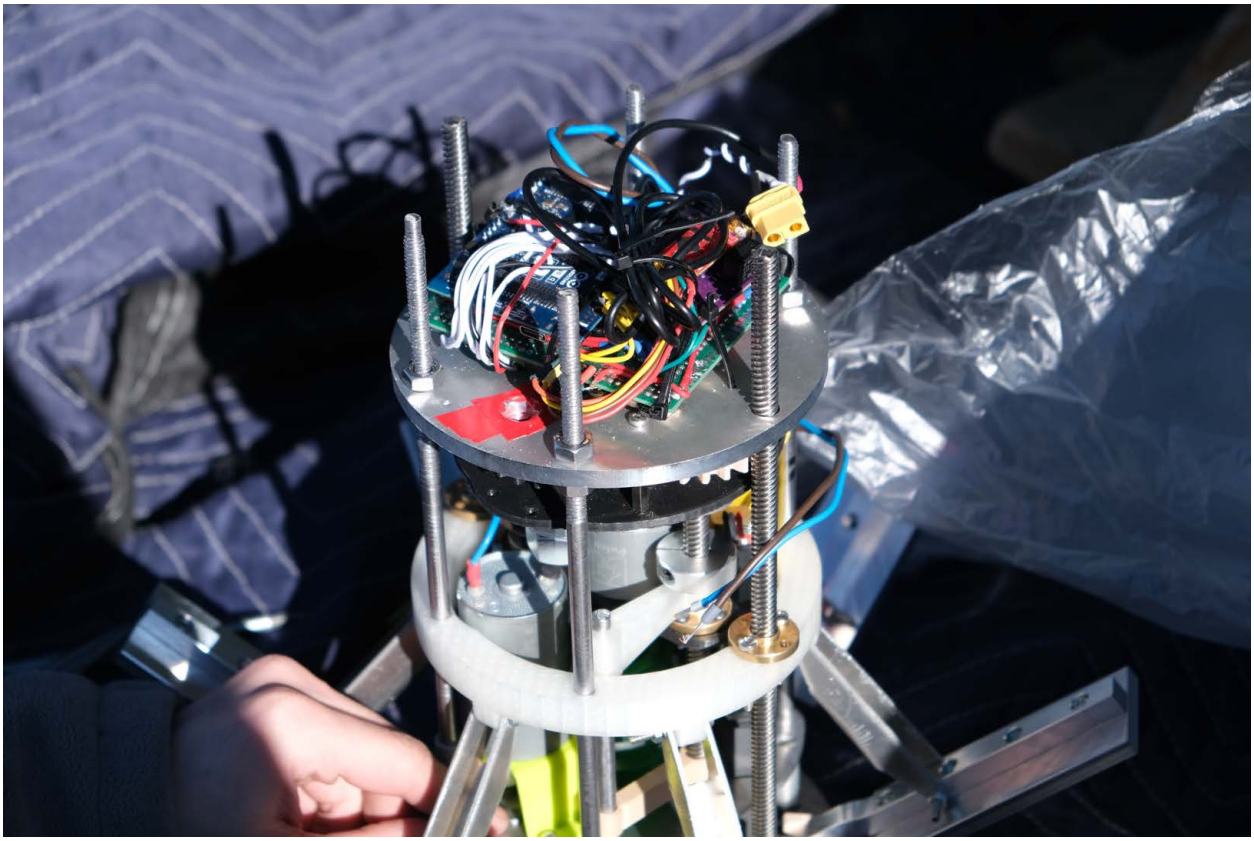


Figure 3.7: As-built Electronics Integrated with Payload Body (Protoboard Configuration)

The Arduino UNO Q incorporates an integrated dual-band Wi-Fi and Bluetooth antenna. No active RF communication is used during flight operations and all wireless interfaces are disabled in the flight software configuration. This was disclosed formally in the FRR in compliance with NASA RF requirements. The recovery system avionics are housed in a separate bay and are fully electrically isolated from all payload circuits as required by the USLI handbook.

Integration Methodology

The integration methodology applied to the MOLEA payload was shaped by three overlapping forces: the mechanical constraints of the payload enclosure, the NASA design review cycle, and the failure data produced by the subscale launches and ground testing. The approach was neither a strictly planned test campaign nor purely reactive, but varied by subsystem depending on how well the interface could be anticipated before hardware existed.

Interface control between the electrical and mechanical subsystems was defined primarily through CAD and physical fitting rather than formal interface control documents maintained separately from the review deliverables. The mechanical team designed the payload body, electronics bay envelope, bulkhead geometry, motor mounts, and lead screw assembly, and the electrical subsystem was designed to fit within those constraints. The electronics bay dimensions, standoff mounting points, connector locations, and wire routing paths were negotiated iteratively as the physical design matured. The PDR, CDR, and FRR submissions served as the primary documentation of interface decisions as they were made, rather than as records of interfaces defined in advance. This approach placed significant coordination burden on the design review writing process itself, as producing each report forced explicit documentation of assumptions that had previously existed only in conversation or in CAD files.

The NASA review cycle did not primarily drive circuit design decisions through technical feedback, but scoring pressure influenced the sequencing of what got built and tested by each milestone deadline. Testing that needed to be reported at CDR was prioritized over testing with longer lead times, and the PCB fabrication timeline was structured around the FRR submission date. This created a situation where the competition payload PCB was designed, reviewed, and submitted to fabrication while the protoboard system was simultaneously being used for functional validation and flown in the vehicle demonstration flight. All thirteen software autonomy tests documented in the FRR were completed on the protoboard system using a mix of bench testing

and state injection simulation before the FRR was submitted.

The most significant integration challenge was the rigidity and connection reliability of the electronics assembly under flight loads. During the vehicle demonstration flight, a recovery system failure produced a harder landing than the protoboard assembly was prepared to survive, and the payload lost power at a physical connection after entering the FLIGHT state. The detailed failure chain is analyzed in Chapter 6; from an integration methodology standpoint, its importance is that a mechanical event propagated through an electrical interface and terminated the mission before surface operations began.

This failure directly motivated the PCB design. The transition from protoboard to PCB did not change the circuit logic, but it replaced press-fit connections, jumper wires, and breadboard contacts with soldered joints on a rigid substrate with defined standoff mounting points. In this sense, the PCB is itself an integration artifact: its primary value is not in what it computes but in how reliably it maintains its connections under the conditions the payload actually experiences.

The software-mechanical interface at the torque switching gearbox presented a different kind of coordination challenge. The gearbox passively switches the single drive motor between leg deployment and auger actuation based on physical position: once the legs reach full extension, a mechanical arm engages the auger drivetrain. The firmware does not receive a sensor signal confirming this transition. Instead, a timed motor run validated through repeated physical testing with the assembled payload determines when full leg deployment and gearbox engagement can be assumed. This design places the reliability of the state transition on the consistency of the mechanical system rather than on a closed-loop electrical signal. It was a pragmatic decision given the constraints of the enclosure, but it represents an open interface assumption: if the legs deploy slower than expected due to soil resistance or mechanical friction, the firmware will command auger actuation before the gearbox has fully engaged. This was identified as an integration risk and was mitigated through testing across the range of expected deployment conditions, but it remains an area where the electrical and mechanical subsystems share an assumption that neither independently verifies.

Cross-disciplinary coordination throughout the project was handled through team meetings, shared CAD access, and the NASA review documents. No formal interface control document was maintained independently of those review artifacts. The practical consequence of this was that interface assumptions were sometimes discovered late, when physical assembly revealed a conflict between what the electrical design expected and what the mechanical design provided. The most consistent example was connector and wire routing, where paths that were clear in CAD became constrained once the full assembly was integrated and recovery harnesses, motor leads, and sensor cables competed for the same space within the electronics bay.

Testing and Verification

Testing on the MOLEA payload spanned five distinct activities completed before FRR submission: bench-level sensor and firmware tests, full payload ground tests of the complete leg, auger, and pH sequence, drop and vibration testing of the electronics assembly, two subscale launches, and the full-scale vehicle demonstration flight. Together these activities produced the failure data and verification evidence that shaped the competition payload design.

Bench-level testing was the first layer of verification and covered individual sensor interfaces, power rail behavior, and firmware state transitions. Each sensor was verified for correct I^2C and SPI communication, address conflicts were checked, and power consumption was measured across idle and active motor states. The thirteen software autonomy tests documented in the FRR, covering launch detection, false trigger rejection, descent bypass, landing confirmation, drilling sequencing, pH acquisition, data logging completeness, and safe shutdown, were completed on the protoboard system using a mix of direct bench testing and state injection simulation. State injection was used where replicating the physical flight condition on the bench was impractical, such as simulating a barometric climb profile to trigger the PREFLIGHT-to-FLIGHT transition without actually launching the payload.

The auger mechanism is shown during torque validation testing in Figure 5.1.

The leg deployment and orientation system is shown during ground testing in Figure 5.2.

Soil collection and pH sensor testing is shown in Figure 5.3.

Full payload ground tests validated the complete autonomous sequence from power-on through pH data collection. The 15-minute operational window was validated through repeated ground runs, confirming that active data collection could operate for 13.5 minutes with 90 seconds reserved for launch and landing phases. The system consistently completed within the window across these tests. Flight confirmation of this result was not achieved during the competition year, as neither full-scale flight produced surface operations data. The soil moisture content testing,

Figure 5.1: Auger mechanism during torque validation testing. The auger was driven into representative soil samples to validate that the 150:1 gearbox provided sufficient torque for penetration and sample collection without stalling the motor beyond its rated current limit.

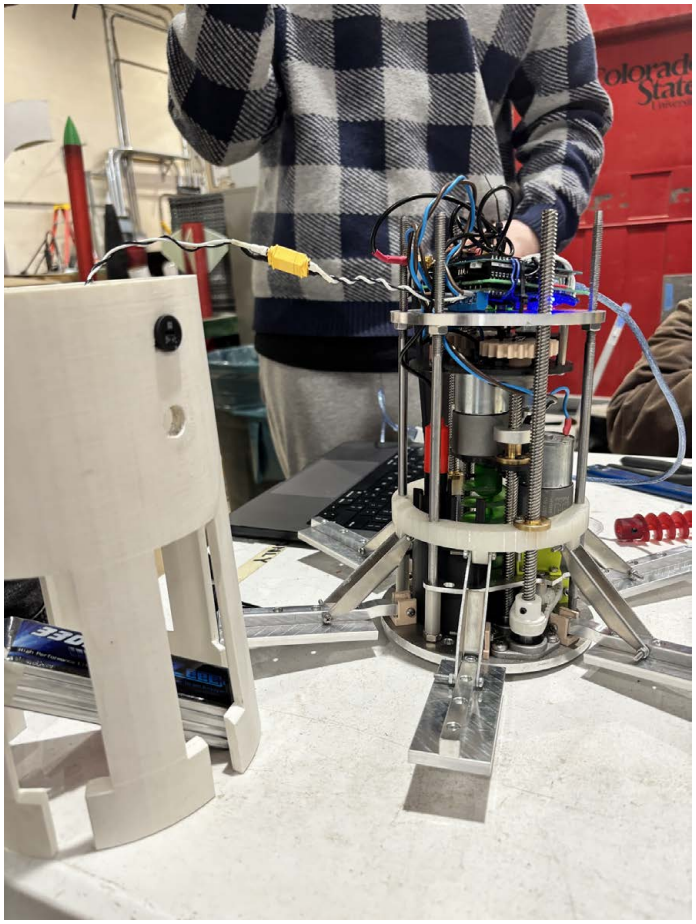


Figure 5.2: Leg deployment and orientation system during ground testing. The payload was placed at multiple starting angles to verify that the leg deployment sequence reliably brought the body to a stable vertical orientation across the range of expected landing attitudes. Torque switch engagement was confirmed audibly and by current signature.

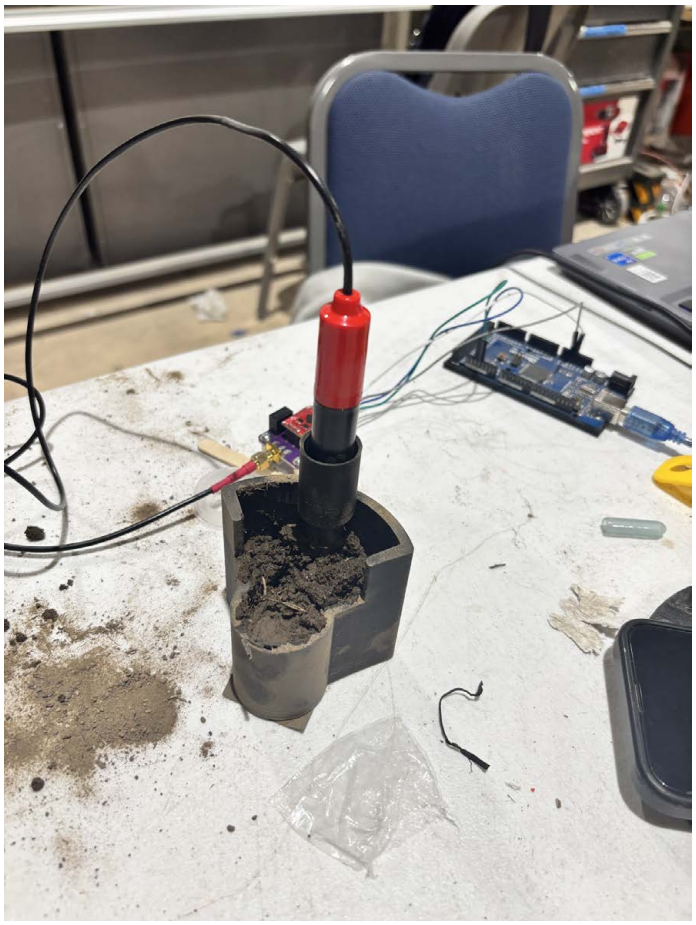


Figure 5.3: Soil collection and pH sensor testing. Soil moisture content was characterized to identify the minimum water-to-soil mass ratio required for reliable pH readings. Readings below approximately 2% moisture content were unstable; consistent and repeatable readings required a minimum of 5% moisture content. Testing was conducted using representative soil types sourced from the CSU Soil Lab.

which characterized the minimum water-to-soil mass ratio required for reliable pH readings at 2% and stable repeatable readings at 5%, was conducted by a mechanical engineer on the team under the electrical subsystem's sensor and data logging infrastructure.

Drop and vibration testing of the electronics assembly was conducted to characterize connection stability under mechanical loading. This testing was added to the verification campaign specifically in response to concerns about protoboard connection robustness identified during assembly, and it preceded the vehicle demonstration flight. The results informed wire management decisions and connector retention practices on the protoboard build that flew.

Two subscale launches were conducted on an Arduino Nano-based flight computer with no hardware differences between the two events. The first launch demonstrated hardware survivability but produced no recovered dataset. The second produced approximately 5.5 hours of standby data before shutting down roughly 30 minutes before launch, making battery endurance under actual pad timelines a competition payload design constraint.

The full-scale vehicle demonstration flight is shown in Figure 5.4. It was the most significant test event and produced the primary failure data driving redesign decisions. The payload lost power after apogee and did not advance past the FLIGHT state, meaning no surface operations data was collected. Post-flight analysis traced the failure to an impact-induced loss of power at the protoboard assembly following a recovery system malfunction. The logged data confirmed that the system had correctly detected launch and entered the FLIGHT state before power was lost, validating the state detection firmware up to the point of failure.

The failure was not attributable to the electrical circuit design or to the firmware. It was attributable to the mechanical robustness of the electrical assembly under an unanticipated loading condition. This finding sharpened the motivation for the custom PCB, where soldered joints on a rigid substrate with mechanically defined standoff mounting represent a direct structural response to the failure mode observed in flight.

The complete verification status at FRR submission reflects this test history. All software autonomy requirements were verified on the protoboard system. Power architecture bench testing

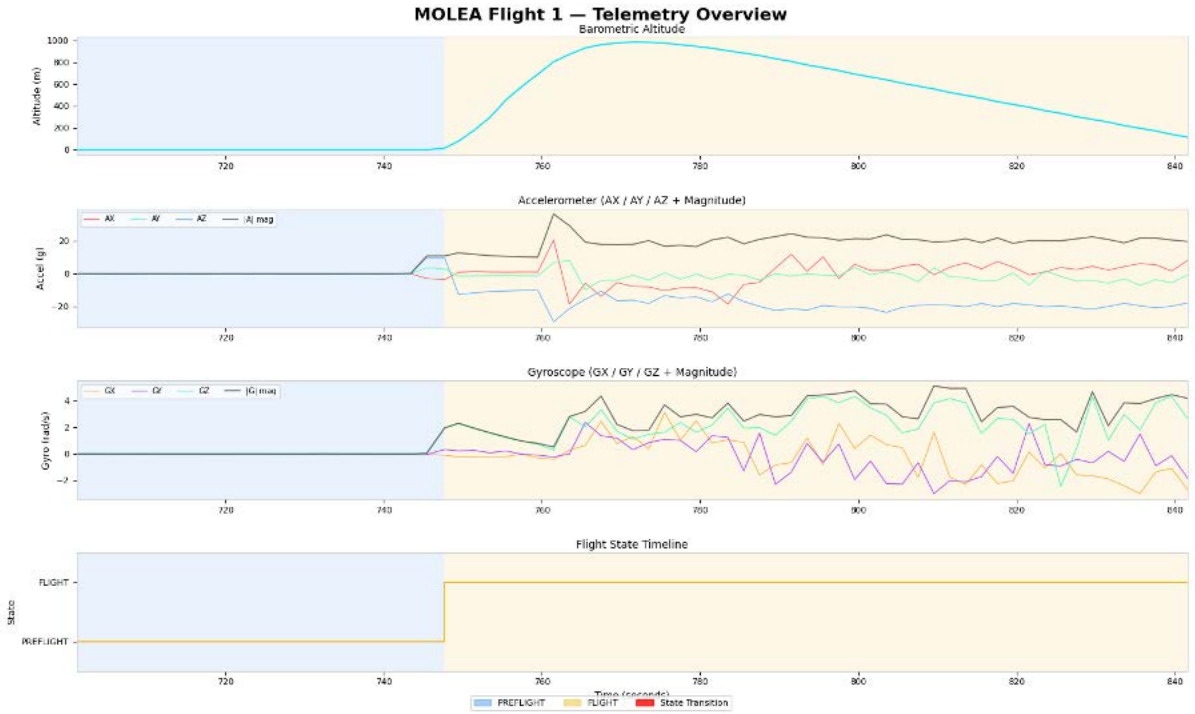


Figure 5.4: MOLEA Full-Scale Flight 1 (Vehicle Demonstration Flight)

confirmed nominal current draw across all operating conditions within defined limits. The full autonomous sequence was validated through ground testing. The one open item was the PCB itself, which was in fabrication at FRR submission. The protoboard system served as the surrogate for all functional verification, and the PCB was treated as a form-factor and mechanical robustness improvement that does not alter the verified electrical behavior.

Flight data from the vehicle demonstration flight is presented in Appendix C (Figures 9.11–9.14), including the altitude profile, accelerometer and gyroscope traces, and the firmware state timeline showing FLIGHT state entry prior to power loss.

Results and Analysis

The results of this project are organized around four distinct test events: the two subscale launches, the vehicle demonstration flight, bench-level validation testing, and ground testing of the complete autonomous sequence. Each event produced either verified performance data or traceable failure data. Together, they form an empirical record of how integration decisions shaped payload reliability at each stage of development. Raw flight logs, sensor data traces, and test records are collected in Appendix C; this section interprets the results in terms of their integration significance.

Subscale Launches

Two subscale flights were conducted using identical Arduino Nano-based hardware. Neither flight produced a complete dataset, but each contributed distinct integration information.

The first subscale launch was scoped solely to validating timestamped sensor logging to a microSD card during a live launch event. No autonomy firmware was under test. The microSD card was not recovered after landing, and no data was retrieved. The flight confirmed hardware survivability under launch dynamics but provided no logged data for analysis.

The second subscale launch introduced an extended pre-launch standby period as a test of power endurance. The system logged approximately 5.5 hours of timestamped sensor data before shutting down, about 30 minutes before launch. Although no flight data was captured, the standby record showed that pad timelines could exceed the capacity margin available in the subscale build. This directly motivated the 3300 mAh battery specification for the competition payload.

No sensor characterization data, altitude profiles, or flight dynamics data were recovered from either subscale event. The subscale launches therefore validated hardware integration and power endurance in a bounded way but did not produce flight performance data. The vehicle demonstration flight was the first event to yield logged flight data and the first to expose integration failures under actual launch and landing loads.

Vehicle Demonstration Flight

The vehicle demonstration flight was the most significant test event and produced the primary failure data driving redesign decisions. The protoboard assembly lost power after apogee and did not advance past the FLIGHT state, meaning no surface operations data was collected. Flight data in Appendix C (Figures 9.11–9.14) confirms that launch detection and FLIGHT-state entry occurred correctly before power was interrupted.

Post-flight analysis traced the failure to a loose power connection on the protoboard assembly after the Piranha line cutter did not actuate and the payload descended with the rocket section rather than under its own parachute. The important result is not simply that a connection failed, but that a mechanical recovery failure propagated through a physical electrical interface and ended the payload mission. This is the clearest example in the project of integration quality determining mission outcome.

Second Full-Scale Flight and End of Competition Campaign

Following the vehicle demonstration flight, the team received a standard NASA reflight extension, providing approximately one additional month to address the recovery system failure and attempt a second qualifying flight. The protoboard electrical assembly survived the first flight impact with no major damage and was repaired without modification to the circuit. The second full-scale flight was conducted within the extension window.

The second flight experienced a parachute entanglement event in the recovery system. Unlike the first flight, the line cutters functioned as designed and severed the Kevlar retention cord, but parachute entanglement prevented nominal deployment and the rocket again descended without a controlled recovery. The electrical payload survived the second landing and again suffered only a loss of power to the protoboard connections on impact. The onboard state detection firmware once again logged successful launch detection and transition to the FLIGHT state before power was interrupted, but no surface operations data was collected from either full-scale flight.

With two flight attempts exhausted under the NASA extension deadline, the team was unable to meet the qualifying criteria for competition launch. The RAMS team did not attend the 2026 USLI competition launch in Huntsville, Alabama.

Bench Testing

Bench-level validation confirmed that the electrical subsystem performs within all defined operational limits. The measured current draw and estimated operational duration, summarized in Table 3.2, validate the power architecture decisions made in response to the subscale failure data. The 3300 mAh battery specification provides meaningful margin over the 6.88-hour estimated duration, and the independent regulated 5 V and 3.3 V rails on the custom PCB provide cleaner separation between logic and motor power domains than the Arduino's native regulated lines allowed.

Ground Testing of the Autonomous Sequence

Ground testing of the complete autonomous sequence validated the finite state machine across all thirteen software autonomy requirements documented in the FRR and summarized in Appendix D. The 15-minute operational window was consistently met across repeated ground runs, with 13.5 minutes available for active data collection and 90 seconds reserved for launch and landing phases.

Flight confirmation of this result was not achieved during the competition year. Neither the vehicle demonstration flight nor the second full-scale flight produced surface operations data. Ground testing therefore represents the most complete validation available for the autonomous sequence, and the flight data that was recovered from both flights, showing correct launch detection and state transition, is consistent with the ground test results up to the point of power loss.

PCB Design Outcome

The custom PCB was received from PCBWay fully populated and physically conformant to the KiCad design files. Bench validation revealed two trace faults: a broken trace in the power distri-

bution network isolating the sensor power rail, and a suspected ground continuity fault affecting the sensor ground net. A partial repair bridging the power trace restored microcontroller boot behavior, but the sensors remained unresponsive due to the ground fault.

The ground continuity fault was not resolved within the project timeline. The design used 0201 and 0402 imperial footprints throughout, which made probing and hand rework impractical with the available equipment. A minimum 0603 footprint standard, test points on critical nets, and redundant paths in power-critical areas would have preserved the form factor while making the first-spin board more recoverable.

As a result, the PCB was not flown. Both full-scale flights used the protoboard assembly, which, despite its connection robustness limitations, represented the only fully functional hardware available.

Integration Failure Pattern

Across all observed failure modes, the pattern is consistent. No failure was attributable to a component operating outside its specifications. Each failure originated in an interface assumption that was not fully validated against the actual environment: pad standby duration in the subscale launches, impact loading at landing in the full-scale flights, and board-level debuggability under the project timeline. The design responses, higher battery capacity, the custom PCB, independent regulated rails, and the relay interlock circuit, all address interfaces rather than the components alone. This is the operational definition of interface-centered design, and the results support it as the most effective reliability strategy for a system of this complexity and constraint.

Discussion

The failure modes documented in this project: the battery depletion identified through the subscale launches, the impact-induced power loss on both full-scale flights, and the PCB trace faults that prevented qualification of the custom board, are individually unremarkable. Batteries drain during long standby periods. Loose connections fail under impact. First-spin PCBs routinely require iteration. What makes these failures instructive is that each occurred after the system had passed the verification activity considered appropriate at the time. The gap between bench performance and operational performance was therefore not a gap in component quality. It was a gap in how completely the operating environment had been defined before verification was declared complete.

The project responses were strongest when they treated reliability as an interface problem. The 3300 mAh battery specification addressed the mismatch between bench endurance and actual pad wait duration. The custom PCB addressed the physical connection between the power supply and the circuit. The independent regulated rails improved separation between logic and motor power domains. The relay interlock constrained the boundary between software state decisions and mechanical actuation. None of these changes fundamentally altered what the circuit computed or what the firmware decided. They changed the conditions under which those computations and decisions could be trusted.

The relay interlock is worth discussing separately because it represents a different mode of integration-driven design. The PCB, the battery, and the power rails were all motivated by observed failures. The relay was motivated by a risk that was identified before any failure occurred, through the act of thinking carefully about what would happen if the firmware commanded motor actuation at the wrong time while the payload was still inside the rocket body. That kind of prospective risk identification is harder than reactive redesign because there is no failure data to point to. It requires enough cross-disciplinary understanding to imagine failure modes that have not yet happened. In this case it worked, and the corresponding failure was never observed. Future teams building

actuated payloads with motors that could cause mechanical damage if triggered during flight should treat the relay interlock pattern as a default starting point rather than an optional addition.

The FSM architecture deserves the same treatment. Building the autonomy as a finite state machine with explicit states, defined transitions, and timeout-driven abort behavior made the software directly verifiable. Each of the thirteen software tests in the FRR maps to a specific state or transition in the FSM. Reviewers could follow the cause-and-effect logic without needing to read the full codebase. Fault conditions forced the system into known safe states rather than leaving behavior undefined. For any student team building an autonomous payload intended to operate without human intervention after landing, an FSM is not merely a software architecture choice. It is a verification strategy, and one that scales well to the NASA review framework where the ability to demonstrate deterministic behavior to a panel of engineers matters as much as the behavior itself.

The PCB design and fabrication experience warrants dedicated reflection. Producing a custom PCB within the constraints of a student competition project is an ambitious undertaking, and the board returned from PCBWay was physically consistent with the fabrication files. The failures were design failures, not manufacturing failures, and they were present before the board was ever sent to fabrication. Both faults, the open power trace and the ground continuity error, reflect the same underlying issue: no redundant path existed on power-critical nets, and the design was not structured to be debugged when something did not work on the first spin. Using 0201 and 0402 footprints throughout the board removed much of the practical ability to probe, bridge, or rework the affected nets. Designing for debuggability is therefore not a secondary concern in student hardware; it is what makes iteration possible.

The NASA review cycle functioned as a documentation forcing function throughout the project. PDR, CDR, and FRR deadlines created pressure to articulate interface assumptions that would otherwise have remained informal. The act of writing the FRR electronics section, for example, required producing a complete sensor interface table, a power architecture description, a current draw test table, and a motor safety interlock schematic, none of which existed as standalone

documents before the report was written. This is not an efficient way to produce documentation, but it is an effective one. The review cycle ensured that integration decisions were recorded at each stage of the design, which made failure analysis after each flight possible in a way that would not have been if the only record of the design was the hardware itself.

The most significant limitations of this work fall into three categories. First, the timed motor run used to confirm torque switch engagement between leg deployment and auger actuation has no closed-loop sensor confirmation. Adding a limit switch or current-signature detection would close this interface assumption and remove the most significant unverified dependency in the autonomous sequence. Second, the custom PCB was not successfully debugged or flown, so the transition from protoboard wiring to a soldered, rigid-substrate assembly was not validated under flight conditions. Third, the complete surface operations sequence, including leg deployment, soil collection, and pH analysis in a real post-flight environment, was never validated by either full-scale flight, leaving the end-to-end system performance unconfirmed beyond the ground test campaign.

Conclusion

The MOLEA payload developed by the RAMS team at Colorado State University for the 2025–2026 NASA USLI competition provides an empirical basis for the central claim of this thesis: subsystem integration quality is a dominant factor governing reliability in small-scale aerospace systems. The project produced a functional autonomous embedded payload subsystem, an interface-centered record of integration and verification decisions, and a failure analysis rooted in actual subscale, full-scale, bench, and ground-test outcomes.

The team completed two full-scale flights under the NASA reflight extension. Recovery system failures on both flights prevented nominal payload deployment and kept the project from meeting NASA's qualifying criteria for the 2026 competition launch. Even so, the electrical system survived with minor damage, and the state detection firmware successfully registered launch and entered the FLIGHT state before power was lost at the protoboard connections. The project was ultimately presented at Colorado State University's Engineering Days showcase, where the full payload, integrated rocket, and auger mechanism were exhibited for the CSU engineering community.

The unresolved technical items are clear: the custom PCB must be redesigned for debuggability and requalified, the leg-to-auger torque switch should be closed with sensor feedback or current-based confirmation, and the complete surface operations sequence still requires validation in a real post-flight environment. For future RAMS teams and other USLI programs building autonomous payloads, the lasting lesson is to treat every disciplinary boundary as a potential failure site, characterize the environment on both sides of that boundary, and verify the interface before relying on the subsystem it connects.

E-Days Presentation

Following the conclusion of the USLI competition campaign, the RAMS team presented the MOLEA project at Colorado State University's Engineering Days (E-Days), an annual engineering showcase held in and outside the Lory Student Center at CSU. E-Days brings together senior design teams, research groups, and engineering student organizations to exhibit their work to the broader CSU community and visiting guests. The RAMS team presented the complete integrated rocket and the MOLEA payload to attendees across the event.

The display included the full rocket body, split to allow visitors to see the internal section layout including the payload bay, recovery system, and avionics compartments. A live demonstration of the auger mechanism was conducted at the table using the protoboard-based payload system, with the auger driven into a soil sample to illustrate the soil collection concept. The full autonomous actuation sequence was not run during the demonstration; the auger operation was manually commanded to show the physical hardware capability. The pH measurement process was explained verbally to visitors, as live sensor readings were not displayed during the event. The team also brought the as-built custom PCB, which was exhibited as a design artifact alongside the protoboard assembly it was intended to replace, providing a visible illustration of the progression from prototype wiring to manufacturable hardware.

The E-Days presentation is documented in Figures 8.1, 8.2, and 8.3.

Figure 8.1: RAMS rocket displayed split at the E-Days table, showing the internal section layout including the payload bay and recovery system.

Figure 8.2: RAMS team E-Days display table in the LSC Ballroom, Colorado State University.

Figure 8.3: RAMS team in front of the E-Days display.

A Personal Note to my Senior Design Team

To my senior design team, what a year it has been. I am deeply grateful to have worked alongside each of you, and I learned more from this project than I could have imagined at the start. Working as the sole electrical engineer with a team of mechanical engineers was, at times, challenging, but it also became one of the clearest lessons I have had in what engineering really means: bringing different disciplines, schedules, perspectives, and ways of thinking together to make something real.

We chose one of the most demanding senior design projects available, with rapid design cycles, constant documentation, long nights preparing NASA reviews, and no shortage of problems that had to be solved quickly and together. Through all of it, we built more than a rocket and a payload. We built trust, resilience, and memories that I will carry with me long after this project is over.

Thank you, (from left to right in Figure 8.3), Isa Fontana, Adrian Ramos Jimenez, Thomas Miller, Tyler Shukert, Connor Morgan, Jack Stern, Adam Ruelas, Phillip Nelson, Ethan Milligan,

Daniel Fisher, and Lilly Perez, for choosing this senior design project with me and for making the year unforgettable.

References

Liu, Z., Zhang, J., & Wang, H. (2021). Embedded systems for space exploration and satellite applications. *World Journal of Advanced Research and Reviews, 12*(3), 238–247. https://wjarr.com/sites/default/files/WJARR-2021-0341.pdf

Shalf, J., et al. (2018). High-performance embedded computing in space. *Journal of Aerospace Information Systems, 15*(8), 422–433. https://doi.org/10.2514/1.I010555

Wang, X., Zhang, L., & Liu, Q. (2020). Design optimization of solid rocket propulsion: A survey of methods. *AIAA Journal, 58*(6), 2325–2344. https://doi.org/10.2514/1.A34594

Saponara, S., et al. (2025). A paradigm for smallsats multi-payload integration. *Acta Astronautica, 222*, 123–135. https://doi.org/10.1016/j.actaastro.2025.05.018

Zeif, R., Kubicka, M., & Hörmer, A. (2022). Development and application of an embedded computer system for CubeSats exemplified by the OPS-SAT space mission. *e & i Elektrotechnik und Informationstechnik, 139*(3), 123–135. https://doi.org/10.1007/s00502-022-00991-9

Zhang, Z. (2023). Modularity, reconfigurability, and autonomy for the future in payload systems. *Aerospace Systems Engineering Journal*. https://doi.org/10.1016/j.aessy.2023.100139

NASA Marshall Space Flight Center Office of STEM Engagement. (2025). *2026 Student Launch handbook and request for proposal*. National Aeronautics and Space Administration.

Appendices

A Electrical Schematics

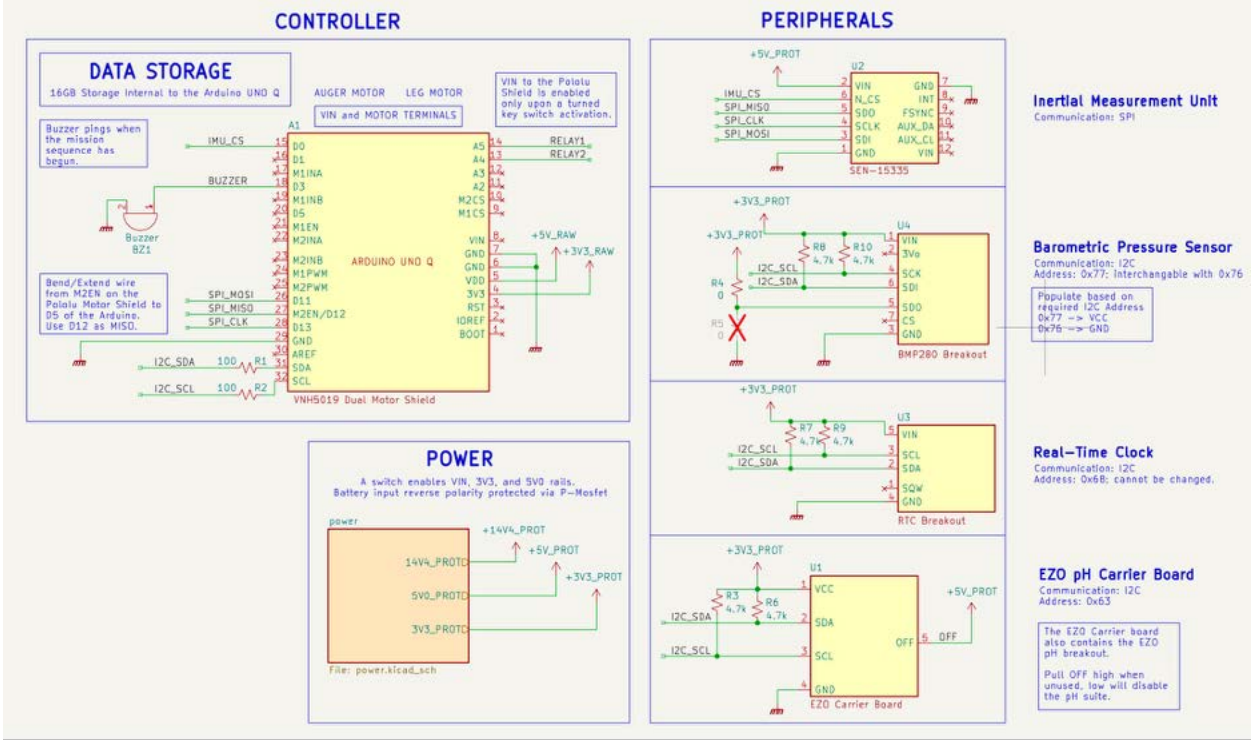


Figure 9.1: MOLEA Electrical Schematics as-built for Full-Scale Flight 1 (Protoboard Configuration)

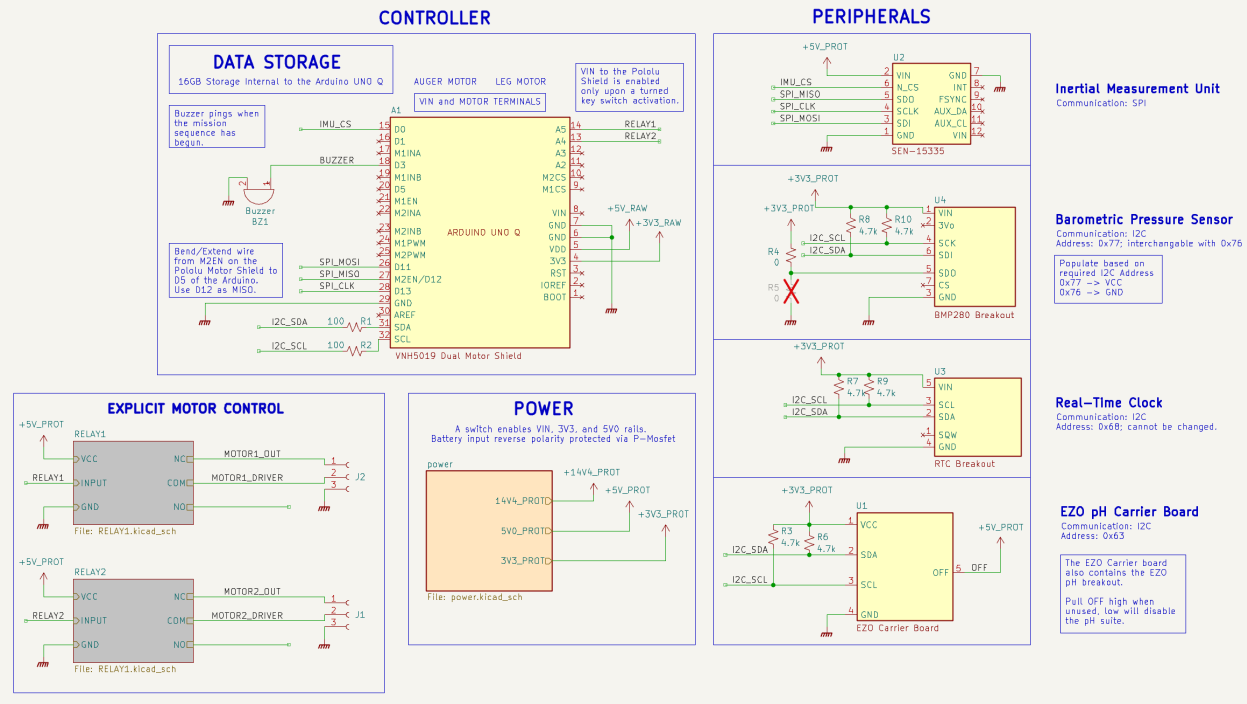


Figure 9.2: MOLEA Electrical Schematics as-built for Full-Scale Flight 2 (PCB Configuration)

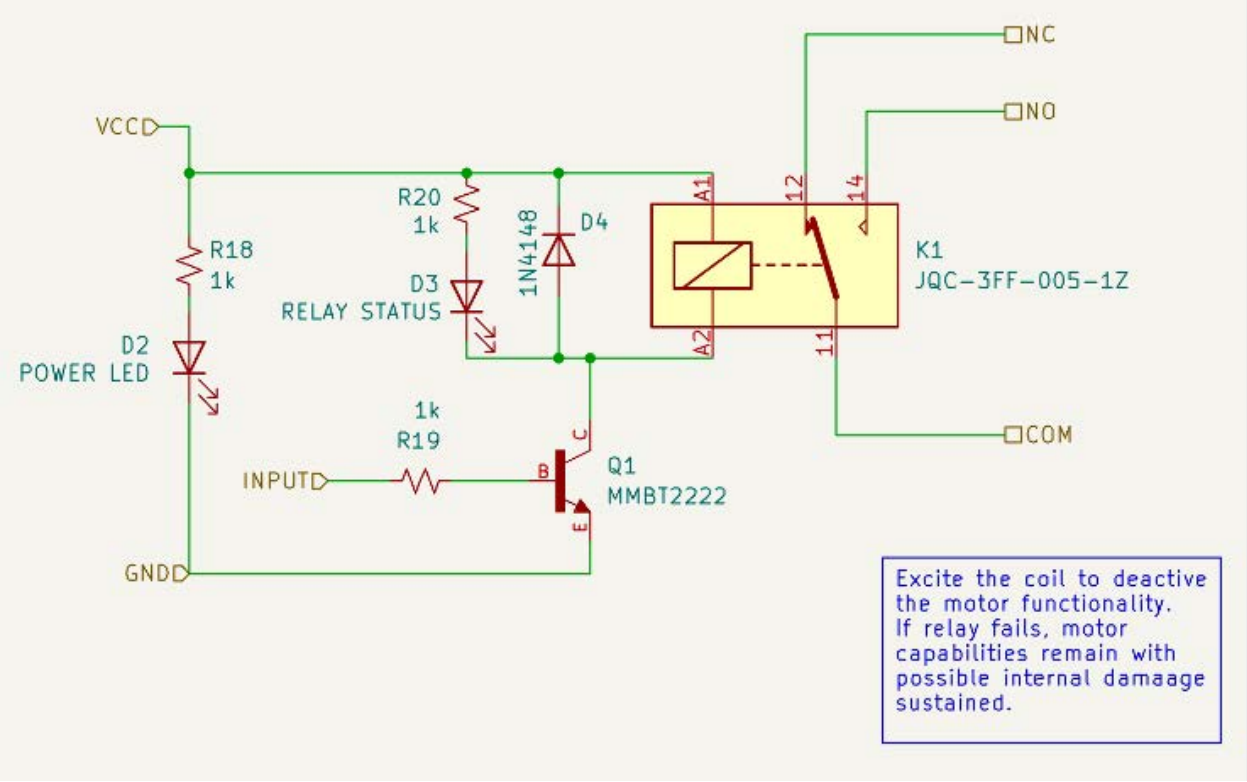


Figure 9.3: Relay Control Schematic (RELAY1.kicad_sch)

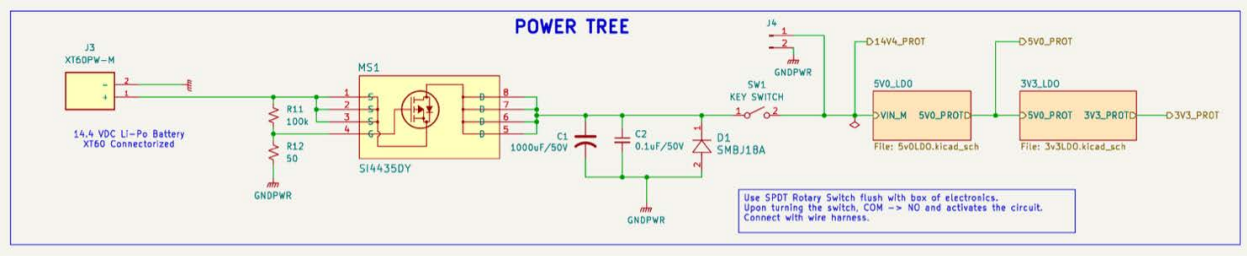


Figure 9.4: Power Tree Schematic (power.kicad_sch)

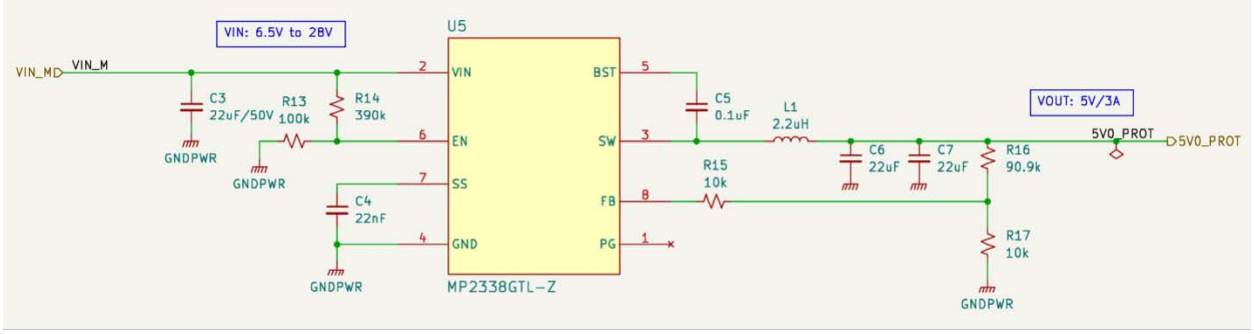


Figure 9.5: 5V0 Regulated Line Schematic (5v0LDO.kicad_sch)

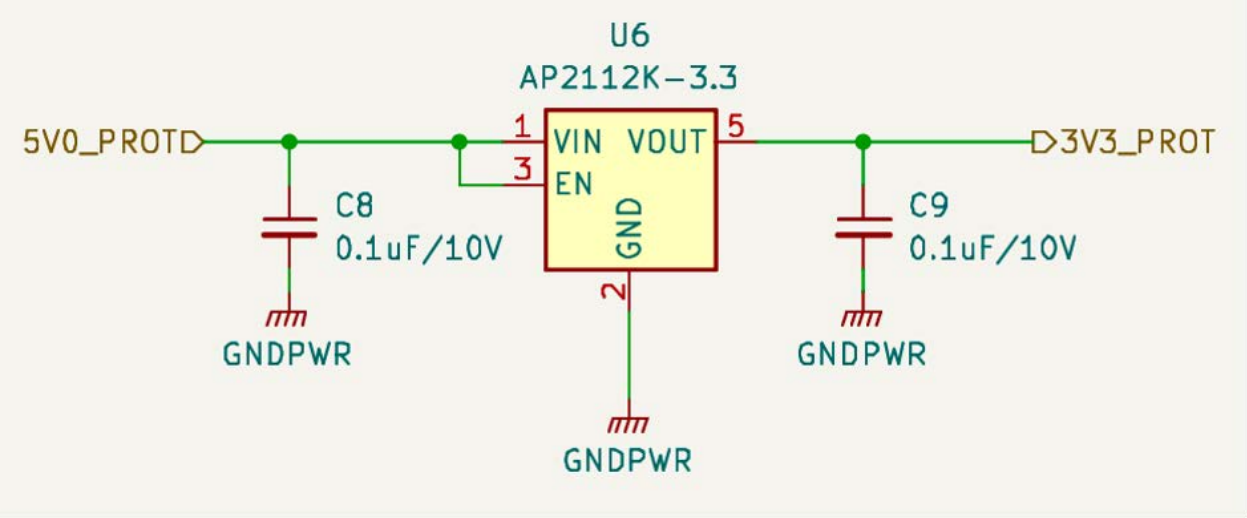


Figure 9.6: 3V3 Regulated Line Schematic (3v3LDO.kicad_sch)

B PCB Design

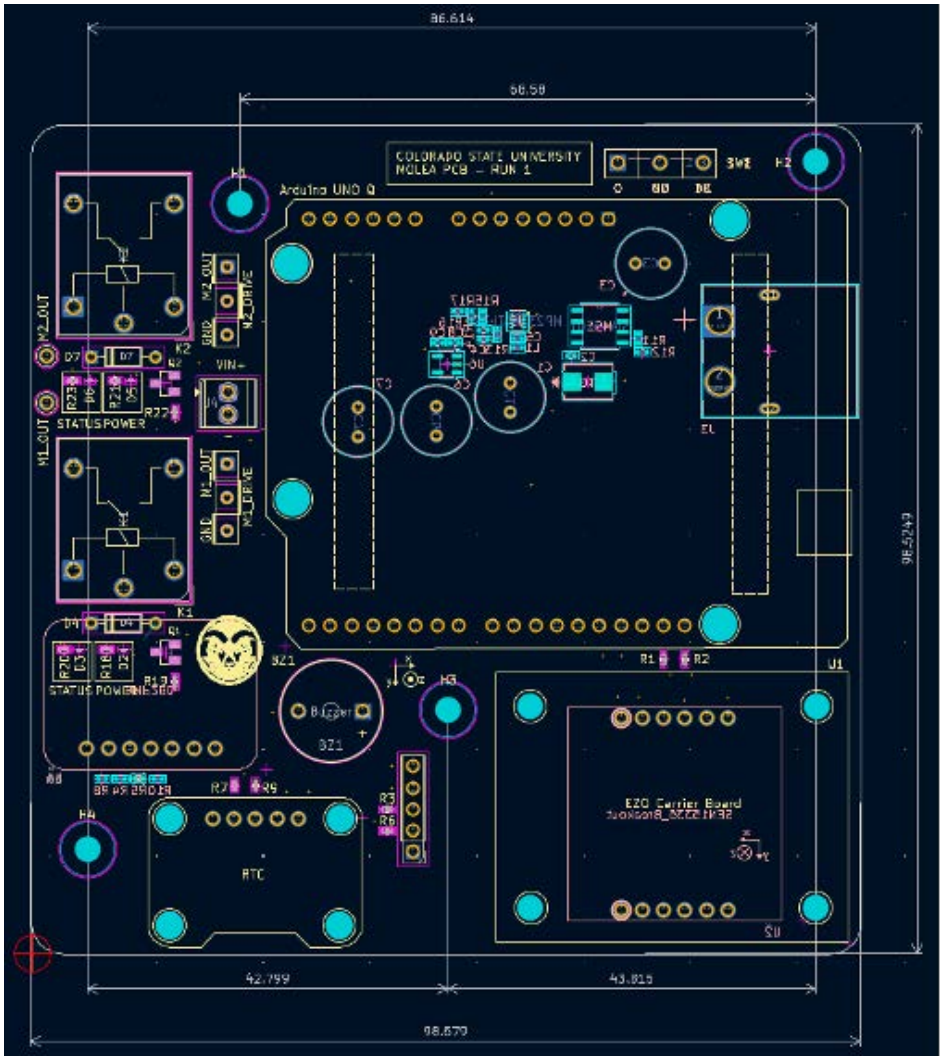
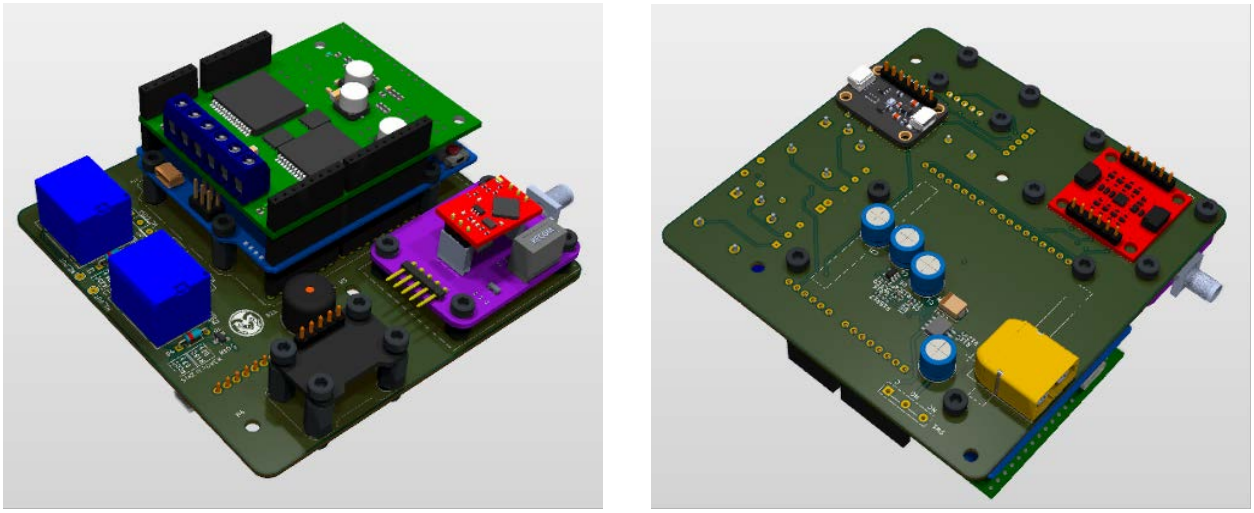


Figure 9.7: MOLEA PCB Layout (KiCad Design File)



(a) Top View

(b) Bottom View

Figure 9.8: MOLEA PCB 3D Renders

C Flight and Test Data

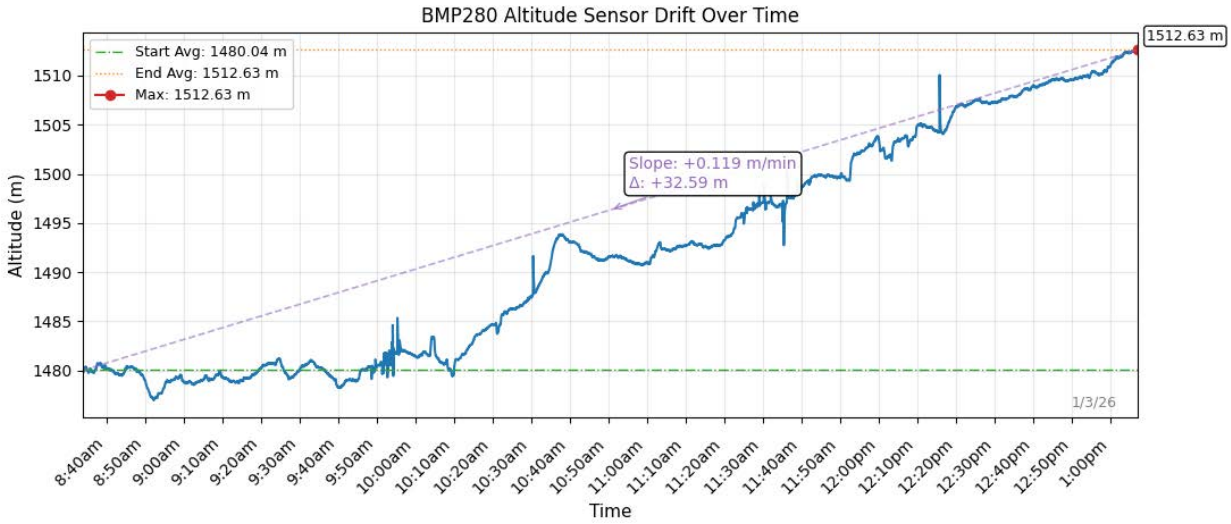


Figure 9.9: Subscale Flight 2 BMP280 Barometric Altitude Profile over the 5.5-hour pre-launch standby window. Data was collected continuously from power-on until the system shut down approximately 30 minutes before launch. The trace characterizes sensor drift and baseline stability under extended standby conditions.

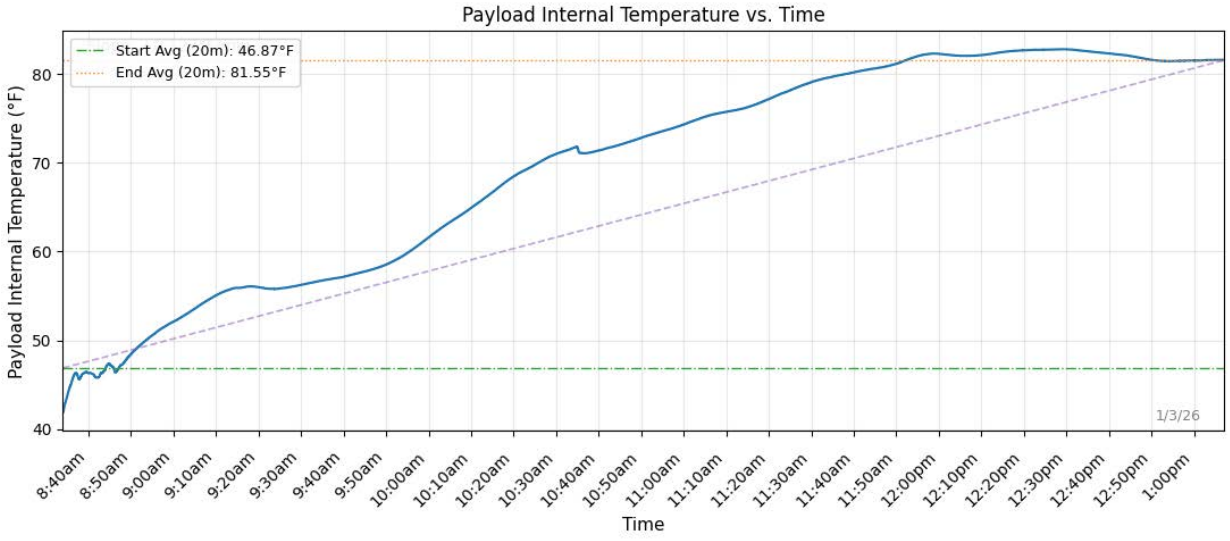


Figure 9.10: Subscale Flight 2 BMP280 Internal Payload Temperature Profile over the 5.5-hour pre-launch standby window. Temperature data was logged concurrently with altitude to characterize thermal behavior of the electronics bay during extended pad wait operations.

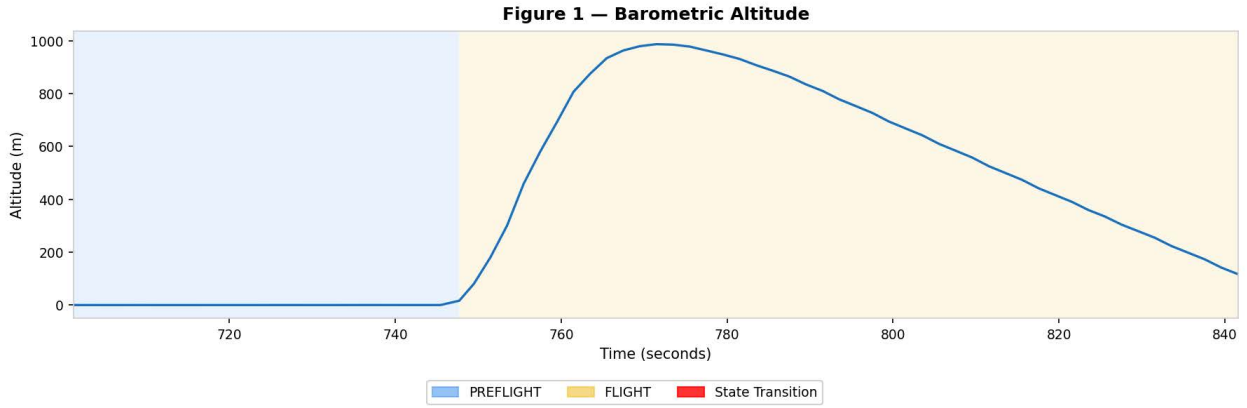


Figure 9.11: Vehicle Demonstration Flight BMP280 Barometric Altitude Profile. The trace captures the full flight profile from launch through apogee. Data logging terminates at the point of power loss following ground impact, consistent with the protoboard connection failure identified in post-flight analysis.

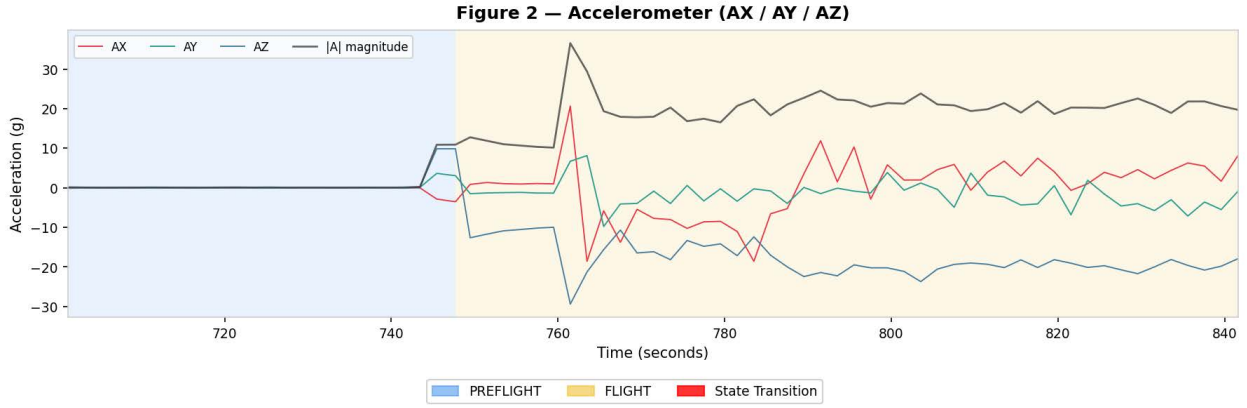


Figure 9.12: Vehicle Demonstration Flight ICM-20948 Three-Axis Accelerometer Data. High-amplitude acceleration signatures at launch and at the moment of power loss are visible. The data confirms correct IMU operation and logging continuity up to the point of the connection failure.

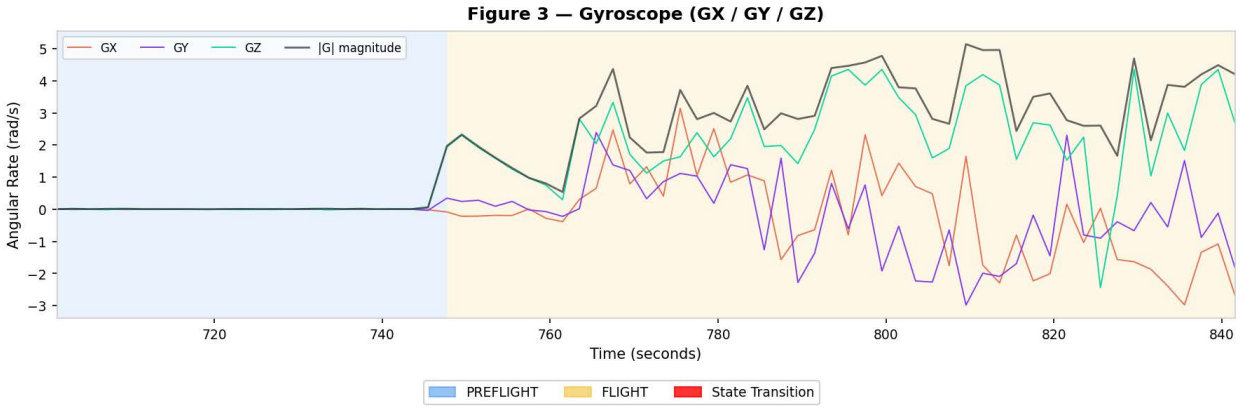


Figure 9.13: Vehicle Demonstration Flight ICM-20948 Three-Axis Gyroscope Data. Angular rate data captures vehicle rotation during ascent and descent phases. Logging terminates concurrent with the accelerometer and altitude data at the point of power loss.

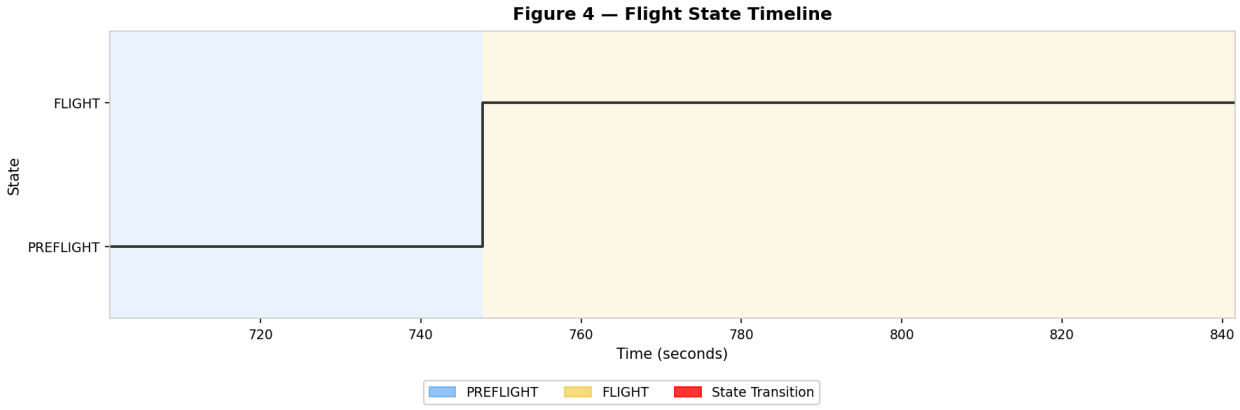
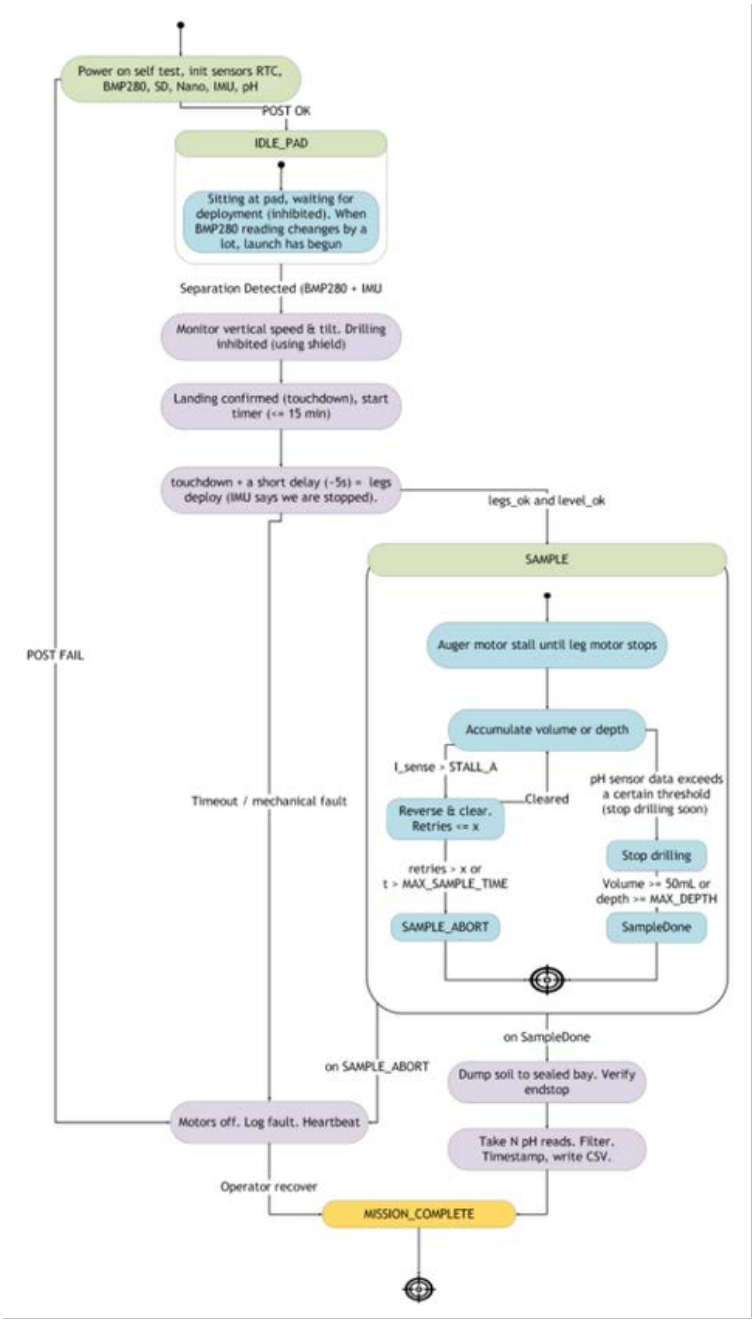


Figure 9.14: Vehicle Demonstration Flight Firmware State Timeline. The log confirms successful transition from PREFLIGHT to FLIGHT state following launch detection before power was interrupted. No further state transitions were recorded, consistent with power loss prior to landing confirmation.

D Software Architecture



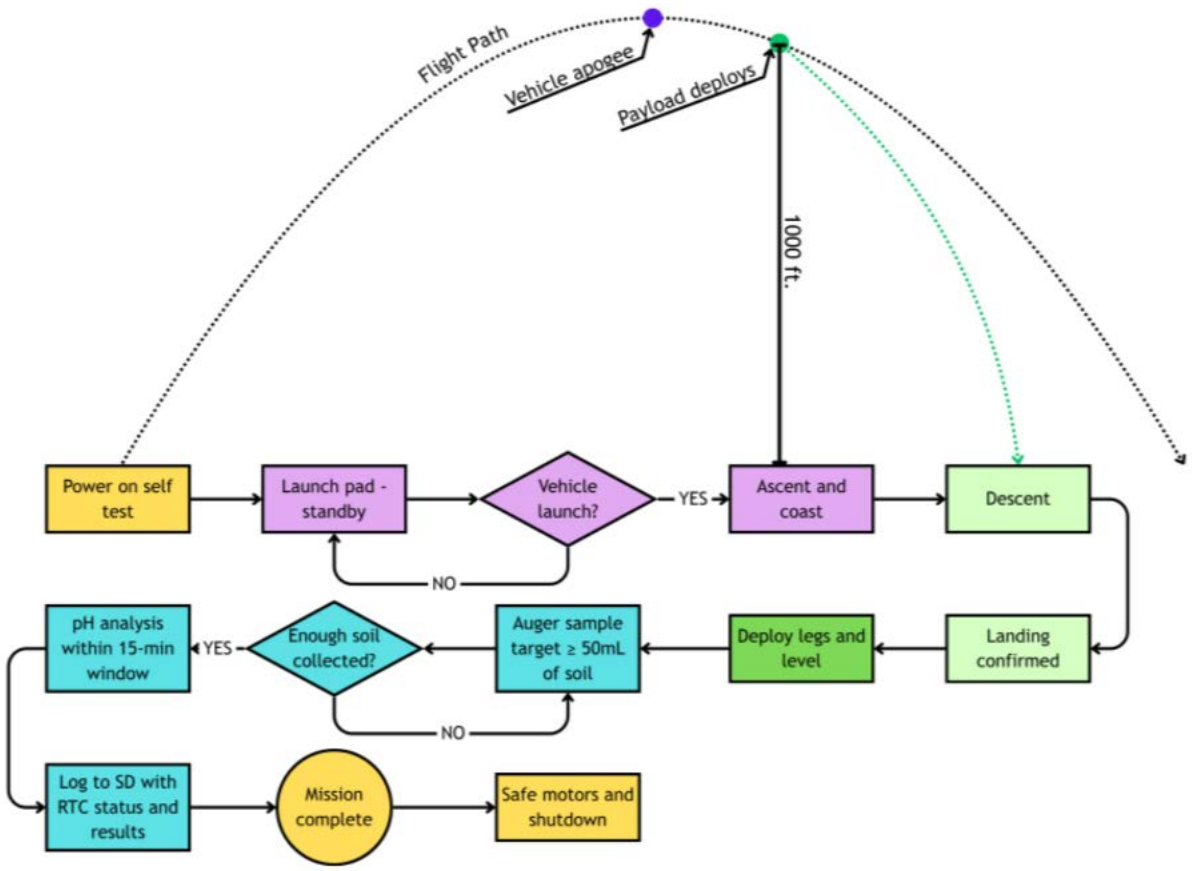

Flight Path
Vehicle apogee
Payload deploys
1000 ft.
Power on self test
Launch pad - standby
Vehicle launch?
YES
NO
Ascent and coast
Descent
Landing confirmed
Deploy legs and level
Auger sample target ≥ 50mL of soil
Enough soil collected?
YES
NO
pH analysis within 15-min window
Log to SD with RTC status and results
Mission complete
Safe motors and shutdown

Table 9.1: Software Autonomy Test Verification (SW-T01 through SW-T13)

Code	Requirement	State	Method	Success Criterion	Status
SW-T01	Launch detection triggers FLIGHT	PRE → FLIGHT	Bench/sim	Logs "LAUNCH DETECTED" and transitions	Complete
SW-T02	No false launch from noise/handling	PREFLIGHT	Bench	State remains PREFLIGHT for entire test	Complete
SW-T03	Descent bypass triggers if launch missed	PRE → DESC_BYP	Drop/sim	Logs "DESCENT BYPASS" and transitions	Complete
SW-T04	Landing detection enters LANDING_CONFIRM only when stable	FLIGHT → LAND_CONF	Bench/sim	Logs "LANDING BYPASS" and transitions	Complete
SW-T05	Landing confirm requires stable hold time	LAND_CONF → DRILL	Bench/sim	Logs "LANDING CONFIRMED → DRILLING" only after timer	Complete
SW-T06	LANDING_CONFIRM resets timer if unstable	LAND_CONF	Bench/sim	Confirm timer resets; does not enter DRILLING	Complete
SW-T07	Drill starts only in DRILLING	DRILLING	Bench	Motor command issued only in DRILLING	Complete
SW-T08	Drilling forces PH_TEST after motor activation	DRILL/PH → PH_TEST	Bench/sim	Logs timeout message and transitions	Complete
SW-T09	Soil collected condition forces PH_TEST	DRILL → PH_TEST	Bench/sim	Logs timeout message and transitions	Complete
SW-T10	pH sampling produces valid readings	PH_TEST	Bench/sensor	pH values at expected interval; filtered output stable	Complete
SW-T11	15-min window enforces stop/abort	DRILL/PH → DISARMED	Bench/sim	Transitions to DISARMED and stops actuators	Complete
SW-T12	Data logging completeness	All	Bench/full	Log contains timestamps & states for entire run	Complete
SW-T13	Safe state turns off actuators	DISARMED	Bench	Motor commanded off, no further actuation	Complete

E Payload Hardware

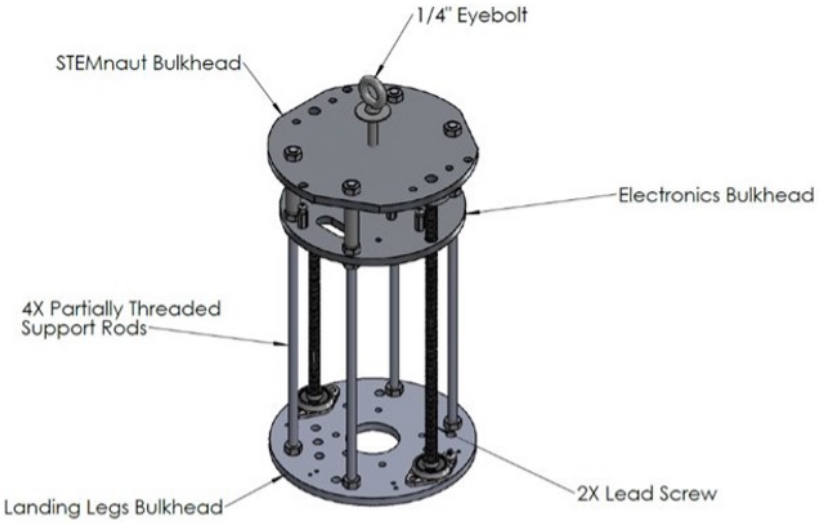


Figure 9.15: MOLEA Payload Structural Layout CAD

Figure 9.16: MOLEA Payload as-built with Legs Extended

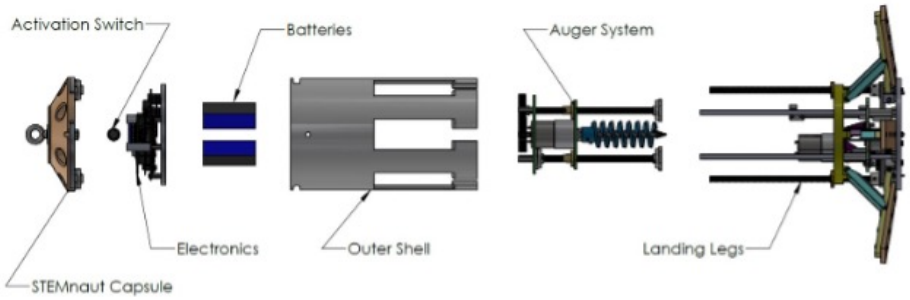


Figure 9.17: MOLEA Payload Exploded View

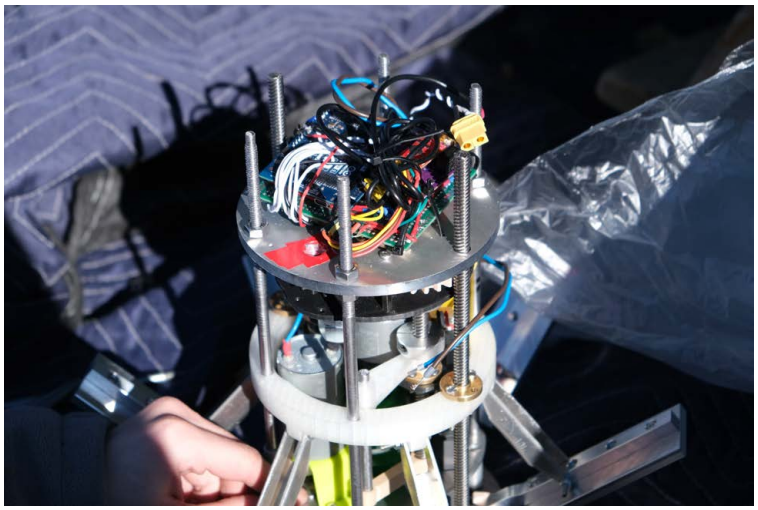


Figure 9.18: As-built Electronics Integrated with Payload Body (Protoboard Configuration)

F Individual Contributions Statement

This thesis reflects individual contributions as the sole electrical engineer on the RAMS USLI team. Responsibilities included all electrical subsystem design, component selection, sensor integration, firmware development, PCB design in KiCad, bench testing, ground testing coordination, and all electrical content across the PDR, CDR, and FRR submissions. Mechanical design, structural fabrication, recovery system design, and soil testing were the work of other team members. The autonomous sequence firmware, including the finite state machine and all state detection logic, was written independently. The GitHub repository for the MOLEA automation codebase is available at https://github.com/bens-on/molea-automation.