SSC21-IX-06

Development of a COTS-Based Propulsion System Controller for NASA's Lunar Flashlight CubeSat Mission

Nathan Cheek
Georgia Institute of Technology
Jet Propulsion Laboratory, California Institute of Technology
620 Cherry St NW, Atlanta, GA 30332; (404) 941-6264
ncheek@gatech.edu

Nathan Daniel, E. Glenn Lightsey, Sterling Peet, Celeste Smith
Georgia Institute of Technology
620 Cherry St NW, Atlanta, GA 30332; (615) 829-0897
ndaniel9@gatech.edu

Daniel Cavender
NASA Marshall Space Flight Center

ABSTRACT

The Lunar Flashlight mission is designed to send a 6U CubeSat into lunar orbit with the aim of finding water-ice deposits on the lunar south pole. The Glenn Lightsey Research Group (GLRG) within Georgia Tech's Space Systems Design Laboratory (SSDL) is developing a low-cost propulsion system controller for this satellite using commercial-off-the-shelf (COTS) parts, with an emphasis on overcoming the harsh environment of lunar orbit through careful architecture and testing. This paper provides in-depth coverage of the Lunar Flashlight Propulsion System (LFPS) controller development and testing processes, showing how an embedded system based on COTS parts can be designed for the intense environment of space. From the high-level requirements architecture to the selection of specific hardware components and software design choices, followed by rigorous environmental testing of the design, radiation and other environmental hardening can be achieved with high confidence.

INTRODUCTION

Advances in manufacturing technologies, along with the continued miniaturization of electronics, are enabling new possibilities within the CubeSat form factor. Classified as a technology demonstration, the Lunar Flashlight mission will substantiate the value of CubeSats in conducting planetary science with demonstrations of green propulsion and active laser spectroscopy.[1]

The Lunar Flashlight Propulsion System is a green monopropellant thruster-based propulsion subsystem of the spacecraft that is designed to deliver over 2500 N s of total impulse. It occupies approximately $2 \times 1 \times 1.5$U of the spacecraft's total 6U volume (1U is approximately 10 cm×10 cm×10 cm).[2] This small size is enabled by the use of additive manufacturing, which allows fluid passages to be built directly into structural supports in ways that are not possible with traditional manufacturing methods. The Lunar Flashlight spacecraft will use this propulsion system to perform a lunar orbital insertion and desaturate momentum wheels.[2]

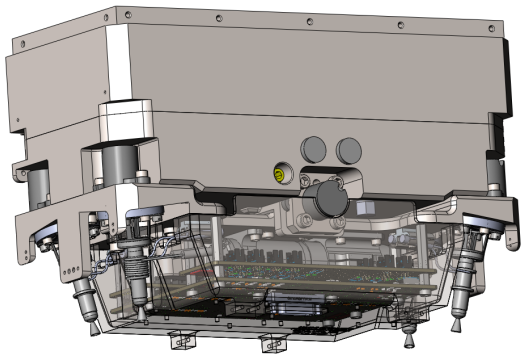

Figure 1: LFPS Model with Controller Visible at Bottom

LFPS includes an electronic control system responsible for managing the Electrical, Electronic and Electromechanical (EEE) components of the propulsion system. This responsibility includes driving valves, heaters, and a pump. It also includes measuring system thermocouple temperatures and pressure transducers as well as monitoring devices for fault conditions. Certain automated and commanded op-

erations require temperature and pressure feedback to drive valve and heater outputs. One example is the ability to thermostatically control the temperature of each thruster catalyst bed using thermocouple inputs to drive heater outputs.

SYSTEM REQUIREMENTS

When Georgia Tech (GT) began the LFPS controller design process, many of the relevant spacecraft interfaces had already been defined. This included connector pinout, operational voltage ranges, command and telemetry protocols, and thruster thermocouple types. Higher-level propulsion system requirements were used to determine the external component interface requirements given in Table 1.

Table 1: External Component Interfaces

External Component	Channels
Valve	6
Heater	10
Pump	1
Thermocouple (K-Type)	5
Thermocouple (R-Type)	4
Pressure Transducer	2

The entire propulsion system was subject to rigid external volume constraints, but the precise volumetric envelope of the controller resulted from careful coordination between the electrical and mechanical design teams. The addition of a pump to the design helped to alleviate high pressure safety concerns,[2] but introduced new volume and layout constraints on the controller. This led to the characteristic shape seen in Figure 2 which was designed to fit precisely over the pump, while still allowing for access to connectors during integration and rigid attachment to the additively manufactured manifold structure via eight mounting points.

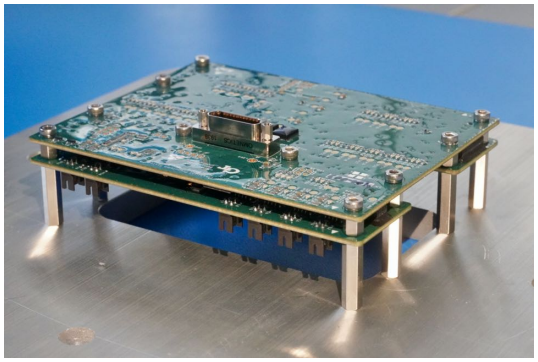

Figure 2: Lunar Flashlight Propulsion System Controller

Power to the controller was required to be segregated for safety purposes. First, a regulated 5 V line allows sensors and digital electronics to function. Second, an unregulated 9 to 12.3 V line allows the heaters, valves, and pump to operate. Splitting the power inputs in this way allows the spacecraft to read sensor data from the propulsion system without a risk of inadvertently energizing the heaters, valves, or pump.

An operational temperature range of at least 5 to 40 °C was required, along with a non-operational range of at least -15 to 60 °C. These values were used during component selection and when developing the thermal vacuum environmental test.

Detailed random vibration load factors were also given as requirements at qualification and acceptance levels. The composite G_{rms} qualification level was 35.7 g, while the composite G_{rms} acceptance level was 17.9 g.

While there were no project guidelines specifically requiring high-reliability space-rated parts, it was required that the controller be designed to withstand the expected radiation profile of the mission. The propulsion system sits at one end of the spacecraft, so one face of the controller is inherently shielded by the bulk of the spacecraft. The other five faces are shielded by a 2 mm-thick titanium 6Al-4V shell. The lifetime total ionizing dose (TID) with this configuration was estimated to be under 10 kRad. Latch-up tolerance requirements were also defined, requiring the controller design to be tolerant of linear energy transfer (LET) of 37 MeV cm^2 mg^{-1}.

HARDWARE DESIGN APPROACH

Component Selection

The LFPS controller was designed to meet MSFC-STD-3012A Grade 4 parts standards.[3] This parts grade allows for commercial parts that have not undergone the stringent screening, inspection, and sourcing constraints required of other grades. The use of this standard allowed for quicker development due to immediate parts availability and low parts cost. Prior to testing, care was taken to maximize reliability of the design through careful component selection.

First, components with flight heritage were selected when possible. The GLRG has delivered a variety of flight electronics for various missions, and this provided a selection of components to choose from. It is, however, very rare for comprehensive flight electronics parts lists to be distributed publicly, especially lists of COTS parts. This makes it

hard to find COTS solutions with flight heritage, and represents an opportunity for further development in the field.

Second, components with automotive AEC-Q100 qualifications were given preference. Such components have generally undergone more stringent testing than non-qualified parts. The AEC-Q100 qualification process subjects components to a variety of test conditions in order to demonstrate their reliability. Mechanical strength is scrutinized through solder ball and wire bond shear tests. Electrostatic discharge tests are performed using human-body and charged-device models. Integrated circuits are subjected to latch-up tests.[4] While there is no central certification board for AEC-Q100, many common suppliers internally qualify their parts to this standard.

Third, components with higher temperature ranges were preferred. The spacecraft may see large temperature fluctuations depending on whether the sun is visible. During sun-pointing operations, the propulsion system will directly face the sun, leading to higher temperatures on the LFPS controller compared to the rest of the spacecraft. The preference for higher temperature ratings extended past the initial component selection phase. Later in the design process, a component audit determined that an in-use Abracon ABMM2-7.3728MHZ-E2-T oscillator had the lowest maximum operating temperature rating of all components on the controller. This was replaced with the ABMM2-7.3728MHZ-D1-T model of the same oscillator family, giving the controller components a most restrictive operating temperature range of -30 to 85 °C, computed by performing an intersection of all component operating temperature ranges.

Selecting the proper microcontroller was a specific concern due to the inherent complexity of such components and its position as a central failure point within the propulsion system. The commercially available Microchip ATmega128 was chosen in part due to the existence of the ATmegaS128, a radiation-tolerant version of the microcontroller. The ATmegaS128 is available in a larger ceramic package, but the ATmegaS128-MD-HP Hirel Plastic version is available in the same TQFP64 package as the COTS ATmega128. The ATmega128 has wider voltage and clock speed tolerances than the ATmegaS128, but it is possible to design a circuit that meets the requirements of both versions of the chip. The LFPS controller was specifically designed to meet the requirements of both of these chips, allowing for a "drop-in" upgrade if needed, with no board design changes. This enabled the decision regarding selection of the COTS ATmega128 or space-qualified ATmegaS128 to be deferred until after radiation testing of the ATmega128 had been performed, while keeping redesign risk low. It also opened up opportunities for the controller design to be reused in future missions with more stringent radiation tolerance requirements. The design was finalized with a normal COTS ATmega128 after an evaluation confirmed that the component survived TID testing.

Samtec FSI-series connectors were selected for board-to-board use due to the simplicity of the connection method and the ample random vibration, mechanical shock, and mating cycle durability specifications provided by the manufacturer. Only one spring-loaded component is needed to make the connection between boards, which allows for low-profile connections as small as 3 mm between boards. The ability to choose such small board-to-board spacing was especially valuable when the controller volume allocation was being initially determined. Two concerns were later identified with this component during testing and flight board integration, but they were not serious enough to prompt a redesign. First, it was noted that the protruding spring-loaded contacts are particularly vulnerable to damage. Care must be taken to protect these parts from becoming bent. Second, one out of nine controller stackups presented an alignment issue that caused intermittent connectivity for some contacts. While this may have been due to a PCB manufacturing tolerance defect, it seems that this connector design is especially vulnerable to such defects.

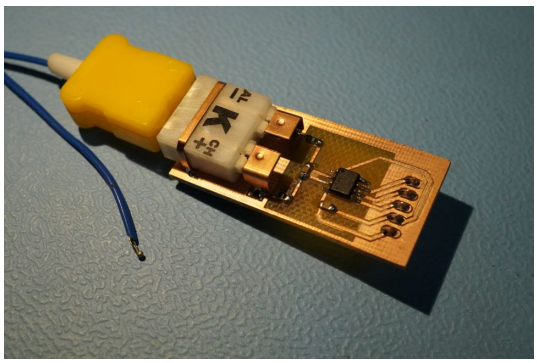

Figure 3: Thermocouple Amplifier Development Board

Proof of Concept Modules

After components were selected, modular proof-of-concept printed circuit boards (PCBs) were developed. These simpler circuits allowed specific controller functionality to be demonstrated without requiring the entire controller functionality to

be simultaneously implemented. For example, a development PCB was created specifically to filter and digitize thermocouple measurements using the MAX31855 thermocouple amplifier (Figure 3). A separate development PCB was created to test the selected heater/valve driver and current sensing functionality. Yet another development PCB was created to test the initial pump motor controller circuit design (Figure 4).

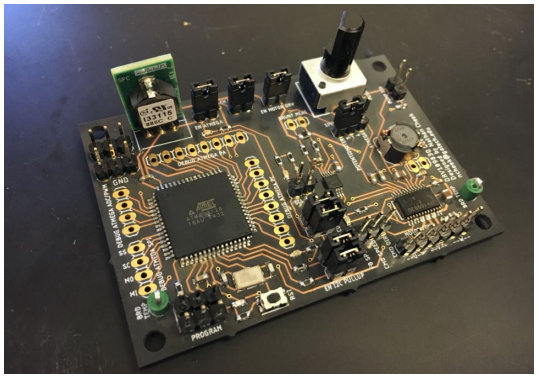

Figure 4: Pump Motor Controller Development Board

Each of these modules was tested separately and schematic, layout, and component updates were made as necessary. Once reliable operation of each module was achieved, the working modules were integrated into a unified design. This modular approach also proved helpful during the consolidated PCB layout process, since the layouts of multi-channel circuits could be duplicated using the Design Blocks feature of Autodesk EAGLE.

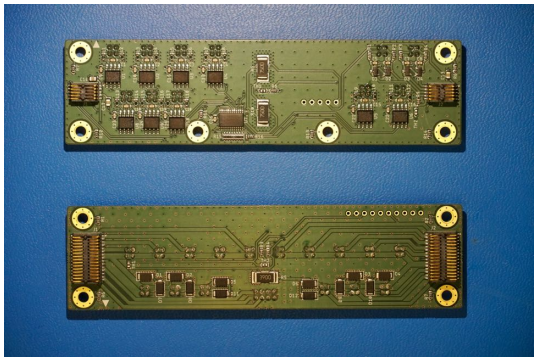

Figure 5: Layout of Controller: Sensor Connect Board (Top), Driver Connect Board (Bottom)

Board Layout

Numerous layout strategies were employed to maximize controller performance and ease of integration. While the three-board stackup was chosen to maximize usable volume around the central pump, it also presented an opportunity to distance high-current driver circuitry from precision analog circuitry. The *Sensor Connect Board* (Figure 5) provides connections to external sensors and includes pressure sensor input protection as well as thermocouple signal digitization components. The *Driver Connect Board* (Figure 5) provides high-current connections to the external heaters, valves, and pump, and includes valve flyback protection components. These components protect the controller by suppressing the voltage spikes generated when inductive valve loads are interrupted. The *Main Board* (Figure 6) connects these boards, with high-current drivers on one end, analog pressure sensor signal amplifiers and microcontroller on the other, and the spacecraft connector in the middle.

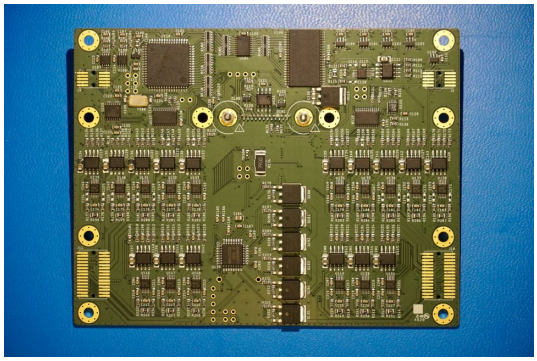

Figure 6: Layout of Controller: Main Board

The shape of the structural manifold guided the placement of connectors on the two auxiliary boards. Connectors were placed directly over gaps in the manifold, allowing for easier access during the assembly process. Figure 7 shows how connector placement exploits these openings.

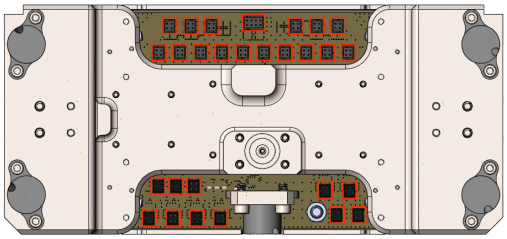

Figure 7: Placement of Connectors (Red) Utilizing Gaps in Manifold

Integrated circuit components were preferentially placed on internal faces of the stackup. This was done to better shield the components from both ionizing and non-ionizing radiation. One disadvantage of this approach was that many components were harder to probe during debugging sessions. This could be resolved, if desired, by creat-

ing adapter boards that allow for communication between all controller boards without requiring them to be stacked vertically.

Four-layer boards were chosen for their performance benefits over two-layer boards. While the interior layers provided additional space for traces in a few congested areas of the *Main Board*, the primary goal was to reduce both the reception and emission of electromagnetic interference (EMI). One internal layer was allocated for use as a power plane. The other internal layer was used as a ground plane. Since the *Main Board* includes both analog and digital ground, the ground plane was split such that the proper ground plane sits underneath both analog and digital portions of the circuit (Figure 8).

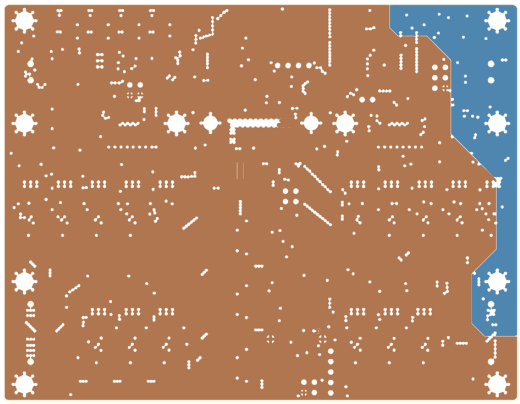

Figure 8: Split Analog and Digital Ground Plane on Main Board

On each board, unused portions of the top and bottom layers were filled with ground pours and connected to the ground plane with a dense array of through-hole vias. Similar vias were also placed heavily around digital communication lines and sensitive analog lines to reduce parasitic coupling. Via fences were also employed around the edges of each board.

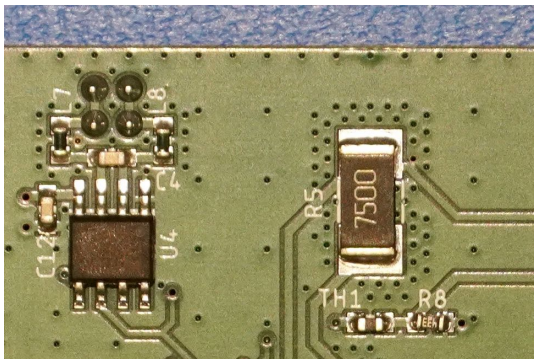

Figure 9: Via Fencing Employed Near Analog Signals and Board Edge; Vias Used for Thermal Dissipation Around Heater

Vias were also used for thermal purposes, placed heavily near components inclined to reach high temperatures such as heaters and flyback diodes. Initial testing of the DRV103H heater driver circuit showed that using thermal vias without corresponding thermal pours on other layers would result in the driver overheating within seconds. By adding copper pours on all layers under the DRV103H and nearby thermal through-hole vias connected to ground, the same layout was able to operate indefinitely without overheating in ambient conditions. Careful use of vias was also considered when placing high-current traces. In some cases, as many as nine vias were placed at the spot where a trace was routed between layers. This approach creates a lower voltage drop across the change in layers, resulting in better electrical and thermal performance while lowering sensitivity to the quality of the via plating process.

Latch-Up Mitigation

Integrated circuits are susceptible to Single Event Effects (SEE), a category of phenomena resulting from interactions between the component and energetic radiated particles.[5] The relatively high level of particle radiation in space makes SEE a significant concern for electronics on satellites, especially those which are operating beyond low Earth orbit. These effects can sometimes lead to undefined behavior, for example due to a bitflip in memory, and in extreme cases can be destructive to hardware.

Single Event Latch-up (SEL) is a type of potentially destructive SEE that affects CMOS devices. It is typically triggered by the deposition of charge from a high energy ion or proton. This event creates a short-circuit path inside the chip, leading to high current draw. Power must be reset to clear this fault condition. Some latch-up scenarios may be inherently non-destructive, but in destructive cases, the power reset must occur quickly to prevent permanent damage to the circuit.[6]

A multi-faceted approach was taken to mitigate the concerns of latch-up within the controller. Certain aspects mirror the "Careful COTS" approach outlined by Sinclair and Dyer.[7] For example, the propulsion system was designed to normally be powered off during flight. The controller will be powered on only occasionally for health monitoring and propulsive maneuvers. This limits the window of opportunity for latch-up to occur. Also, current-limiting resistors were commonly placed in series with logic pins. This can mitigate latch-up and pre-

vents devices from damaging each other in the event that different voltages are asserted concurrently on the same line.

The LFPS controller also includes an overcurrent protection scheme meant to automatically clear a latch-up fault. The 3.3 V power rail is split into five sub-rails each protected by a TPS2553-Q1. If a short-circuiting latch-up scenario occurs, the high current triggers the open-drain $\overline{FAULT}$ output of the relevant TPS2553-Q1, which pulls down the EN line, temporarily disabling the power rail. Recharging of the EN line is resistor-limited, providing time for the parasitic latch-up to reset before the power rail is re-enabled. Care must be taken to ensure that the circuit delays the retry mechanism long enough for decoupling capacitors to discharge, ensuring the component experiencing latch-up fully resets. It is also important to determine the appropriate overcurrent threshold for such a circuit, such that it never triggers during normal operations, but reliably triggers in a SEL scenario.

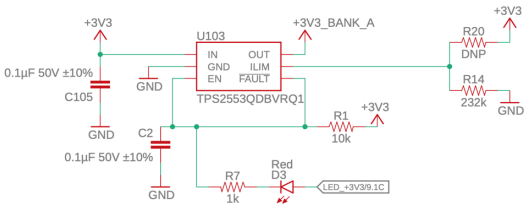

Figure 10: TPS2553-Q1 Overcurrent Protection Circuit

One of these circuits is shown in Figure 10, while the resulting waveform in the presence of a non-clearing overcurrent fault can be seen in Figure 11. It is generally expected that a latch-up fault will clear after the first power reset, but this circuit will continue resetting as long as the fault condition remains.

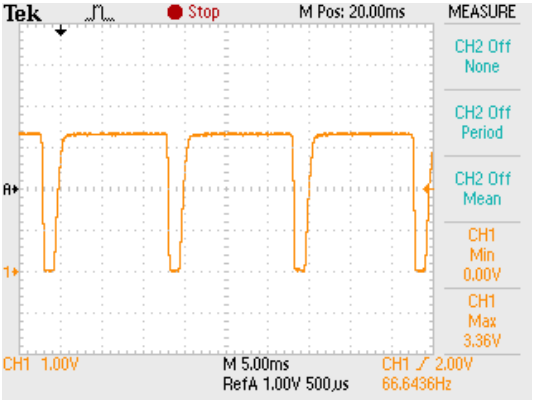

Figure 11: Waveform of Overcurrent Protection Circuit in the Presence of a Non-Clearing Fault

FIRMWARE DESIGN APPROACH

The propulsion system controller firmware utilizes the F Prime framework. F Prime is an open-source flight software framework developed by the Jet Propulsion Laboratory (JPL) that allows for modular development of reliable flight software. Its component-based development process simplifies architectural decisions during the software design phase and allows the work to be easily reused in future projects.[8]

Utilizing F Prime Features

F Prime provides a number of features, such as autocoded interfaces (called "ports"), modules (called "components"), and connections between component ports. It comes with many components and features relevant to operating a spacecraft, including systems for handling incoming commands and outgoing telemetry. Since the LFPS command interface was predefined, Georgia Tech developed a component that converts incoming command frames into the format expected by F Prime. The code for distributing the commands to relevant components was then autogenerated by the framework. Similarly for telemetry, components use the internal F Prime system to distribute all telemetry channels to a single GT-developed component. This component then packages the outgoing telemetry into the standardized frames required by the project.

A similar approach allowed GT developers to take advantage of F Prime's built-in parameter system. F Prime supports hardcoded default parameter values, as well as the ability to store new parameter values in volatile or non-volatile memory. After specifying parameter names, datatypes, and identifiers in XML component definition files, the F Prime autocoder generated the appropriate code allowing components to query and update these parameters. The ATmega128 microcontroller includes 4 KiB of non-volatile EEPROM. A component was developed to interact with the EEPROM and provide a parameter database interface to F Prime's parameter system, allowing the operator to store new parameter values that persist between reboots.

Many of these components are triggered to run data-handling procedures and control algorithms via F Prime rate groups. A rate group driver uses an ATmega hardware timer to occasionally trigger rate groups, which call specific component input ports, triggering time-based actions.

Reliable Firmware

A number of techniques were utilized to promote reliable controller operation. First, static memory allocation was preferred. When memory was allocated, it was done during startup. This reduces the risk of memory leaks, which happen when memory is not properly freed for reuse, leading the system to run out of usable memory.

Second, assertions were rigorously used throughout the codebase. Assertions check for logically impossible situations, and reset the microcontroller if they are found. During the development and testing phase, this helped identify bugs such as when unexpected values were passed between components. It will also allow the system to detect errors during flight and revert to a safe mode.

Third, a watchdog component was developed to utilize the ATmega128's built-in watchdog hardware. This component is triggered by a rate group to occasionally stroke the watchdog. If this does not occur, which may happen if the microcontroller locks up or crashes, the watchdog will perform a hardware reset.

Fourth, little trust was placed in the state of peripheral devices. Generally, at startup, configurations are written to peripherals such as general purpose input/output (GPIO) expanders or motor controllers. If these components suffer single event effects or other related issues, their configuration registers could change or even revert to defaults, leading to unexpected behavior. The ATmega128 is programmed to occasionally rewrite these configuration values, to ensure that unexpected states are quickly reverted.

F Prime on the ATmega128

The open-source F Prime code repository has brought support to various devices and operating systems including Raspberry Pi and Linux.[9] However, the GT development team had to implement specific changes to port the framework to the ATmega128 platform. For example, a platform-specific *HardwareRateDriver* component was created, utilizing the ATmega's *TIMER1* hardware timer to drive rate groups via an interrupt.

The F Prime framework generally requires more program and data memory than other frameworkless approaches. This represented a serious concern as the ATmega128 only provides 128 KiB of flash program memory and 4 KiB of internal SRAM. Fortunately, the ATmega offers support for external SRAM, which allows the microcontroller to take advantage of 64 KiB of SRAM data memory. While the addition of external SRAM solved the data memory limitation, other work was required to reduce the program size to fit on the ATmega. F Prime provides a number of configurable options to reduce this size. For example, the *FW_OBJECT_NAMES* and *FW_OBJECT_TO_STRING* options which store debugging strings were disabled. Also port serialization, which uses network byte order to communicate between components, was disabled by modifying the *FW_PORT_SERIALIZATION* option. Identifier datatypes used for command, telemetry, parameter, and other systems were minimized in many cases from 32-bit values to 16-bit values.

Two compiler optimization flags were enabled which also had a large effect on lowering the program size. The *-mrelax* flag decreases program size by shuffling instructions around such that *JMP* instructions can be replaced with shorter *RJMP* instructions. While *JMP* instructions must be able to change the program counter to any valid address, *RJMP* instructions use less space to describe a relative jump to a nearby address. The *-mcall-prologues* flag bundles function prologues and epilogues into reusable subroutines which lessens program size in exchange for slower function calls.[10]

LFPS Topology

A simplified view of the LFPS F Prime topology is given in Figure 12, where components and connections related to rate groups and parameter handling have been hidden. Control of the heaters, valves, and pump are split among three components. Each of these components is responsible for processing relevant sensor inputs and operator commands, as well as generating driver outputs based on internal algorithms.

GROUND SUPPORT EQUIPMENT

It would not be possible to effectively test and iterate such a complex design without the help of custom ground support equipment. Various equipment was designed and manufactured specifically to aid in the maturation process of the controller. This approach eased the process of hardware and firmware debugging, and enabled specific functional tests to be more easily completed.

Portable Firmware Development Boards

The COVID-19 pandemic necessitated that much of the LFPS firmware development occur remotely. While entire controller development units could be

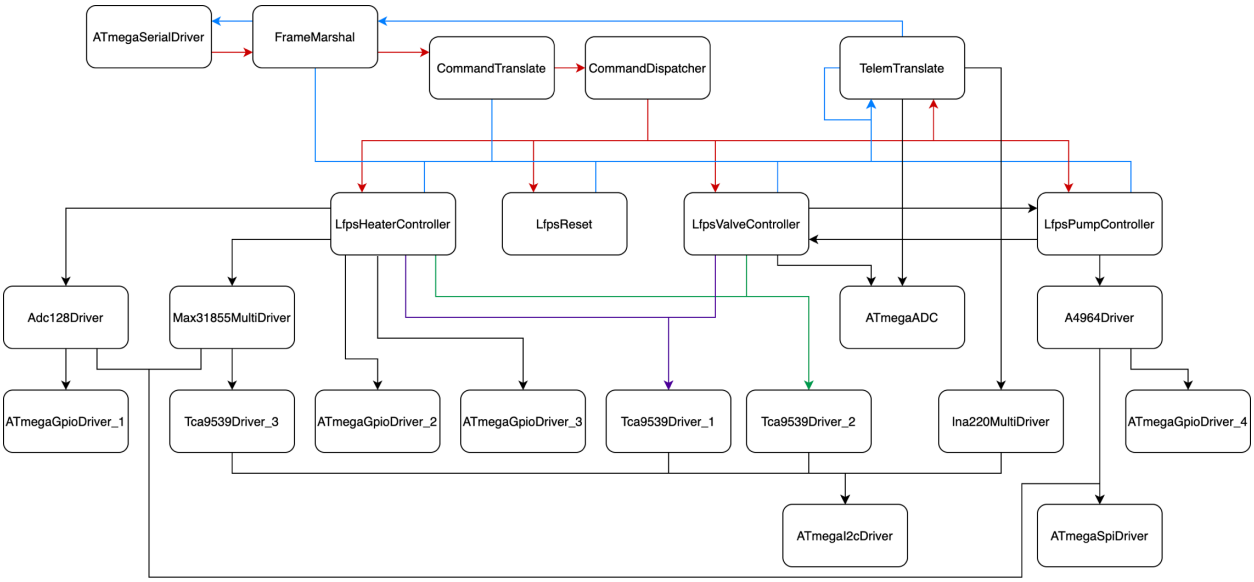

Figure 12: Simplified F Prime Firmware Component Topology

temporarily borrowed for remote testing, this approach did not scale efficiently as new developers were onboarded to the project, due to the limited number of development units and the effort required to assemble new units.

The Georgia Tech team worked through this limitation by designing a credit-card sized development board specifically for firmware testing (Figure 13). This basic board includes an ATmega128 microcontroller along with necessary support components including a crystal oscillator and external SRAM. It also includes the spacecraft RS-422 and development UART serial interfaces, along with I2C and SPI headers for interfacing with other development boards. Finally, an ADC and two pressure transducer amplifier circuits are included.

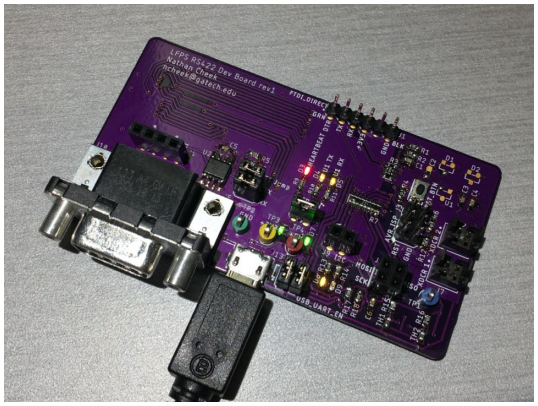

Figure 13: Firmware Development Board

This development board was especially useful during the initial F Prime platform porting activities, allowing developers to quickly test functionality of the ATmega128 microcontroller running F Prime. Following that initial phase, the development boards were used to test deployments as the LFPS firmware topology was built out. Very few F Prime components communicate externally, so even though this board does not include all of the driver and sensor components of the real controller, most components could be effectively tested.

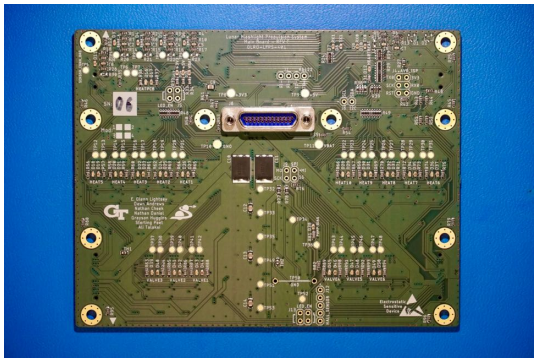

Figure 14: Top of Controller with Circular Pogo Pin Landing Pad Test Points

Controller Interface Boards

Numerous test points were designed into the LFPS controller to enable debugging and functional test procedures. Early designs utilized Harwin S2761-46R test points soldered onto pads, but it

was discovered that the force from standard spring-loaded test clips would often delaminate these pads from the PCB substrate. This approach was discarded in favor of circular pads (Figure 14) that could accept spring-loaded pogo pins from separate controller interface boards.

Controller interface boards were designed to stack directly onto the controller stackup and connect small component-less test points on the controller to larger sturdier interfaces more accessible to the user (Figure 15). Keystone 5XXX-series test clips expose many such test points. Voltage rail connections are made available via banana jacks, while communications buses are available via D-sub connections. Polarized AVR ISP connectors provide an interface for burning the ATmega bootloader. An FTDI FT232RL UART to USB interface allows the user to communicate directly with the microcontroller and flash firmware updates.

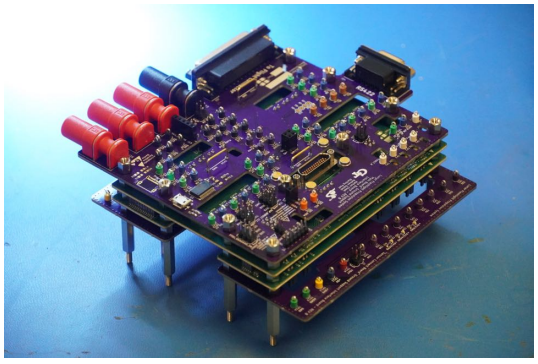

Figure 15: Controller Interface Board Stackup

Since communications and power connections are made via pogo pins (Figure 16), these interface boards offer the ability to use and test development controller boards without installing expensive flight connectors. The main controller interface board offers a 25-pin D-sub connector that mimics the pinout of the 25-pin Micro-D flight connector. The controller programmer board also contains connectors with this pinout. This standardized approach combined with the availability of off-the-shelf Micro-D to D-sub adapters allowed for reuse of test harnesses and boards throughout the development process. For example, a spacecraft emulation testbed could connect to an LFPS controller development unit via the D-sub connector on the interface board. If a controller was later provided with a Micro-D connector, it could be connected through that connector without modifying the pinout.

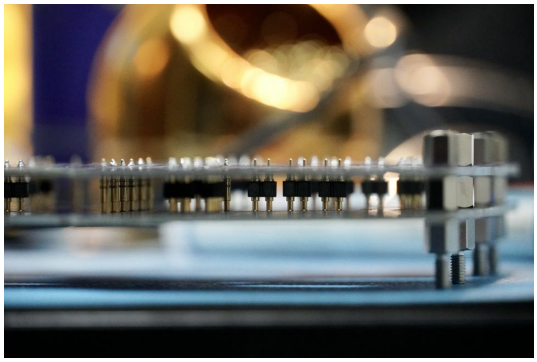

Figure 16: Pogo Pins Assembled on Interface Board

Controller Programmer Board

A programmer board was created to allow firmware flashing and debugging whenever the controller is in hard-to-reach positions. By utilizing the 25-pin flight connector pinout, this board enables new firmware to be installed via USB as long as a connection to the flight connector can be made. For example, during thermal vacuum environmental testing, it was helpful to be able to send functional test commands and install updates without having to break the vacuum. A harness was created to pass the 25-pin connection from the controller to the programmer board through a chamber pass-through connector, allowing for quick program changes during early testing.

This board will also be useful later in the spacecraft integration stage. Once the propulsion system is integrated, there is no way to program the LFPS controller except through the spacecraft connector. As long as this is accessible, new firmware can be flashed by connecting it to this board.

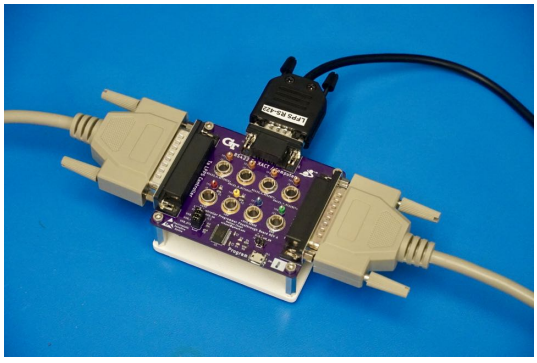

Figure 17: Firmware Flashing Board

FlatSat

Development of a FlatSat platform was critical to enabling efficient functional testing, especially during the firmware development phase. This platform (Figure 18) was specifically designed to emulate all inputs and outputs of the controller. 50 W chassis-mounted resistors emulate heater loads, with resistance values matching the expected values of the heaters. Relays act in place of valves, providing an inductive load with closely matched parameters.

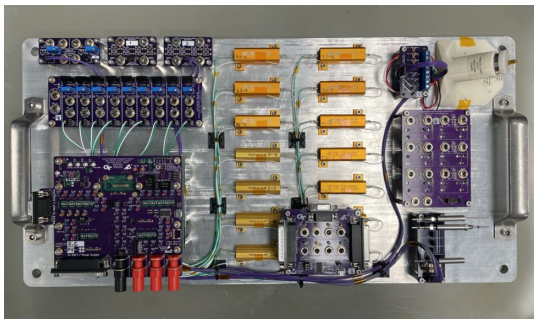

Figure 18: Controller FlatSat

Pressure transducers are emulated with adjustable Wheatstone bridges, made of static resistors and potentiometers (Figure 19). Resistor values were chosen to closely match the documented input and output impedance of the XP5 pressure sensors. This allowed for high-confidence testing of the pressure sensor amplification circuit prior to arrival of the actual sensors.

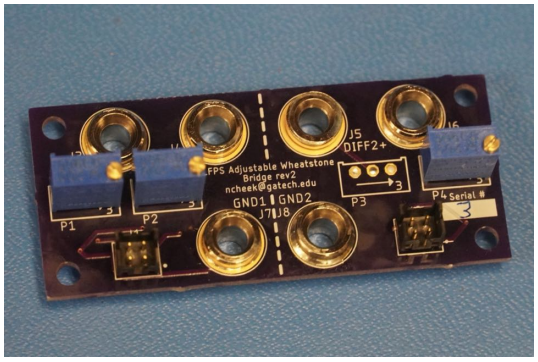

Figure 19: 2-Channel Adjustable Wheatstone Bridge Showing a Paired Potentiometer Configuration and a Single Potentiometer Configuration

Thermocouples are emulated using a similar adjustable Wheatstone bridge circuit (Figure 20). The largest difference is that each channel must be electrically isolated for proper operation of the MAX31855 thermocouple amplifiers. This was accomplished by the use of CUI PDSE1-S5-S3-S isolating 3.3 V DC-DC converters, and had the unexpected side-effect of lowering thermocouple measurement noise significantly.

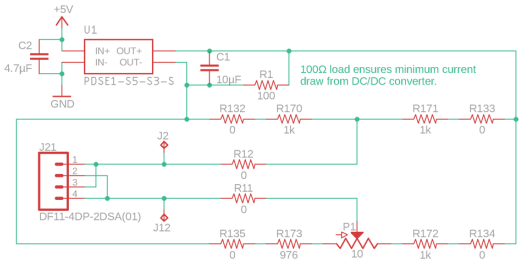

Figure 20: Thermocouple Emulator Channel

The first design of this emulator board did not use isolating DC-DC converters. It was discovered that an isolated power supply was necessary to power the board, but this only allowed a single channel to be used at a time. Noise was also an issue with this design. Through testing with fixed resistor values, it was discovered that the resistor values played a crucial role in reducing noise. Figure 21 shows the reduction in variance from using 1 kΩ resistors instead of 10 kΩ resistors. Variance reduced from 87.49 to 12.58 °C with this change.

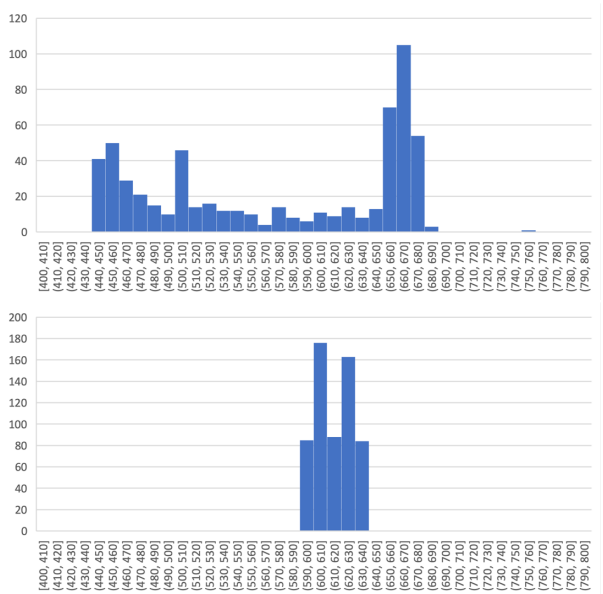

Figure 21: Thermocouple Measurement Histograms Using 10 kΩ (top) vs 1 kΩ (bottom) Resistors in a Wheatstone Bridge-Based Emulator, 600 Samples Each, 10 °C Buckets

The addition of isolating DC-DC converters allowed all channels to operate concurrently and further decreased the MAX31855 measurement variance. The design seen in Figure 20 reliably measures down to 0.25 °C with occasionally differing readings

within 1 °C of the nominal reading (Figure 22). Variance was 0.12 °C. This makes the emulator an excellent choice for testing how the controller responds to changes in temperature.

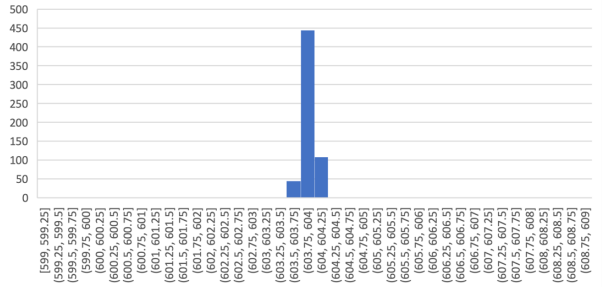

Figure 22: Thermocouple Measurement Histogram Using 1 kΩ (bottom) Resistors in an Isolated Wheatstone Bridge-Based Emulator, 600 Samples, 0.25 °C Buckets

The MAX31855 can safely accept either polarity of differential voltage on the thermocouple inputs, which meant the emulator design did not need to ensure a specific polarity. This allows the operator to set as low of a voltage as they wish, as a negative polarity output from the bridge will not damage the MAX31855.

Figure 23 shows the final design of the thermocouple emulator board with isolating DC-DC converters. Potentiometers allow the user to change the differential output voltage. Banana jacks are used to power the board, as well as measure the differential output of each channel.

Figure 23: Thermocouple Emulator

Graphical User Interface

A web-based Graphical User Interface (GUI) was developed to aid in controller testing (Figure 24). This GUI provides an interface for sending commands to the controller, as well as reading back telemetry from the controller. A Python server communicates with the controller via a serial connection, verifying and parsing incoming telemetry frames as well as generating outgoing command frames. This program includes a webserver so local and remote browsers can connect to the command and telemetry console. Live data is passed between the browser and webserver via the Socket.IO library.[11]

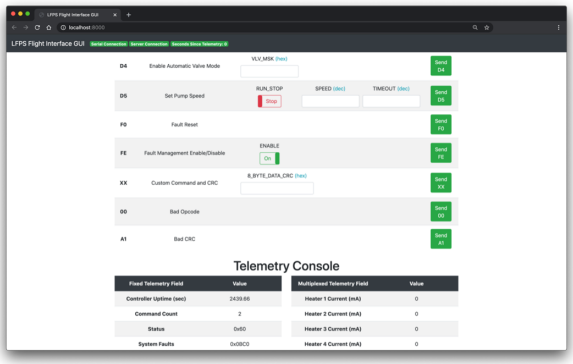

Figure 24: Controller Command and Telemetry GUI Webpage

This program logs incoming telemetry and outgoing commands allowing for later analysis. It also provides a command recording and playback feature which allows the operator to record a command sequence and repeat it back indefinitely. This was developed to enable the sending of repeated thruster fire commands, a useful test for confirming that the controller continues to respond properly over long periods of time.

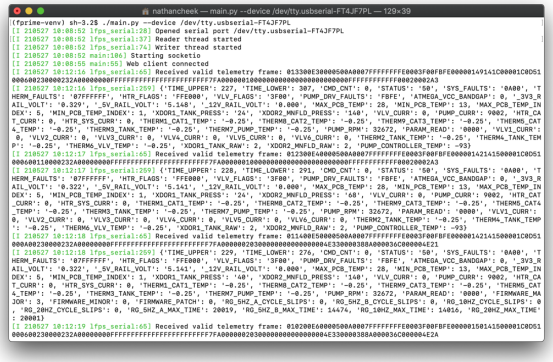

Figure 25: Controller Command and Telemetry GUI Backend with Placeholder Data

ENVIRONMENTAL TESTING

A range of tests were compiled into an environmental testing campaign with the goal of showing the LFPS controller could withstand the radiation, vibration, vacuum, and temperatures inherent to launch and operation in space. SEL radiation testing was performed on individual components. TID radiation testing was performed on component development boards as well as the first prototype controller. Random vibration and thermal vacuum tests were carried out on an engineering development unit (EDU) after staking and conformal coating.

Single Event Latch-Up

Based on the LFPS controller bill of materials, JPL provided guidance on the components most susceptible to SEL. This was used to develop a list of components to test. JPL performed SEL testing on these parts with an effective LET of $42.14\,\mathrm{MeV\,cm^2\,mg^{-1}}$ at a 45° angle, using a krypton ion source. The results of these tests are summarized in Table 2.[12,13]

Table 2: SEL Test Results

Component	Result
TPS73733DCQ	No SEL observed
REF3033	Non-Destructive SEL observed at 60 °C
REF3012	No SEL observed
MCP6N16	Non-Destructive SEL observed at 60 °C
DRV120APWR	Destructive SEL observed above 12 V
INA220-Q1	No SEL observed
TPS2553Q	No SEL observed

After noting the destructive SEL seen in the DRV120APWR, the DRV103H was identified as a potential replacement. With its higher voltage rating of 32 V compared to 28 V, it was hoped that this device would be less susceptible to SEL at the maximum system voltage of 12.3 V. Both drivers were included in the initial prototype controller, but the DRV103H was ultimately chosen as the sole heater and valve driver due to its higher voltage and current ratings.

Total Ionizing Dose

After assembling a prototype controller, members of the Georgia Tech controller development team visited JPL to perform TID testing of the prototype with a cobalt-60 source. Testing was performed on the prototype controller (Figure 26) as well as a number of component-specific breakout boards. The inclusion of breakout boards allowed data collection to continue, independent of the state of the controller. For example, a design error required the removal of the DS89C21 transceiver from the controller, yet results were obtained by testing the same component on a breakout board. The test plan in Table 3 was followed for each device under test (DUT), for a total dose of 30 kRad on each board.

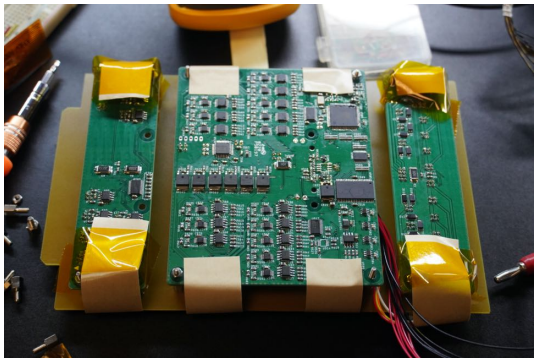

Figure 26: Controller between TID Irradiation Phases, Pins Grounded with Conductive Tape

Table 3: 30 kRad TID Test Plan

Test	Description
1	Functional Test of DUT
2	10 kRad dose at 25 Rad/s
3	Functional Test of DUT
4	10 kRad dose at 25 Rad/s
5	Functional Test of DUT
6	5 kRad dose at 25 Rad/s
7	Functional Test of DUT
8	5 kRad dose at 25 Rad/s
9	Functional Test of DUT

Between each irradiation segment, a functional test was performed on the DUT, as seen in Figure 27. For each device, this generally included measuring the current drawn by the device, then as relevant, measuring output voltages, checking its ability to communicate, logging measurements made by the device (ideally measurements of some external reference), and checking that the device responded properly to external stimuli. General results of this test are given in Table 4. Note that current meter resolution was provided down to 1 mA during this test, so recorded changes of 1 mA are not generally considered to be significant.

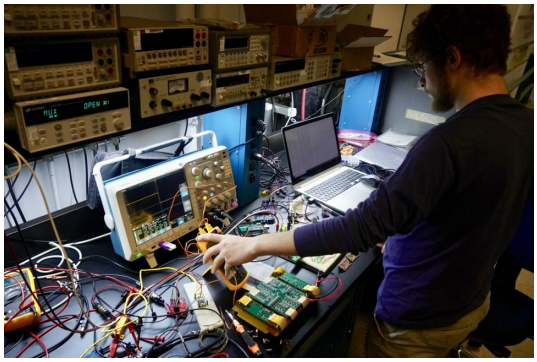

Figure 27: TID Functional Test Setup

Table 4: TID Test Results

Component	Degradation
74HC573PW-Q100	None observed
A4964KJPTR-T	Possible temperature measurement degradation
AS7C4096A-12TCN	None observed
ATMEGA128A-AUR	None observed but limited functionality tested
CY7C1049GE30-10ZSXI	None observed
DRV103H	None observed
DRV120APWR	None observed
DS89C21TMX/NOPB	No functional degradation, possible change in current draw
INA220BQDGSRQ1	No functional degradation, possible change in current draw
IRFR2405PBF	None observed
MAX31855KASA+T	Internal and external temperature measurement degradation
MCP6N16-010E/MS	None observed
REF3012AIDBZT	None observed
TCA9539QPWRQ1	None observed
TLE4284DV33ATMA1	Output voltage degradation
TPS2553QDBVRQ1	None observed

74HC573PW-Q100, AS7C4096A-12TCN, and CY7C1049GE30-10ZSXI

The Nexperia 74HC-series latch was tested as part of three SRAM test circuits which resided on two SRAM breakout boards and the controller. One breakout board utilized an Alliance AS7C4096A-12TCN. The other breakout board and the prototype controller utilized a Cypress CY7C1049GE30-10ZSXI. Testing with an Arduino Mega at 16 MHz, both breakout boards passed all memory read/write tests. Similar tests on the controller at 8 MHz were inconclusive due to a continuous memory paging error likely caused by a misconfigured firmware library. On the breakout boards, current draw remained stable at 83 mA for the Alliance chip and 82 mA for the Cypress chip.

A4964KJPTR-T and IRFR2405PBF

The Allegro A4964 sensorless motor controller was tested along with six International Rectifier IRFR2405PBF MOSFETs on an A4964 development board. The motor driver circuit was able to successfully drive the motor at 3000, 6000 and 9000 RPM after each irradiation phase. The internal temperature measured by the A4964 during 9000 RPM testing decreased from 38.47 to 37.12 °C, but a confounding factor was that later tests were likely performed more quickly, allowing less time for the component to heat up.

ATMEGA128A-AUR

Basic functionalities of the ATMEGA128A microcontroller were tested on the prototype controller. Communication with I2C peripherals was successful during the entire test. UART communication was also successful, although occasional data corruption was seen. This corruption was later determined to be due to timing error inherent to the ATmega when communicating at a 115200 baud rate with an 8 MHz crystal oscillator, and prompted a change from 8 to 7.3728 MHz in the next revision.

DRV103H and DRV120APWR

Both Texas Instruments drivers were tested as part of the prototype controller. Two of each component were tested, with one driving a heater and the other driving a valve. All four tested channels operated nominally. The DRV103H heater channel consistently measured 199 mA, while the valve channel consistently measured 116 mA. The DRV120APWR heater channel consistently measured 196 mA, while the valve channel consistently measured 59 mA.

The startup delay times of both components were also tested. The DRV103H heater channel operated at 100% duty cycle with no delay, as configured, while the heater channel operated at 100% duty cycle for 11.5ms each time before switching to hold mode. The DRV120APWR heater channel also operated at 100% duty cycle with no delay, and the valve channel operated at 100% duty cycle for 1.2ms each time before switching to hold mode.

DS89C21TMX/NOPB

This RS422 transceiver was tested on a breakout board. Across the 30 kRad cumulative dose, the transceiver continued to successfully send and receive data between a Teensy 3.2 development board and test computer via an FTDI USB-RS422-WE-1800-BT cable assembly. After the 30 kRad cumulative dose, current drawn during no communication dropped from 29 to 27 mA, while current during communication dropped from 29 to 28 mA.

INA220BQDGSRQ1

This current and bus voltage sensor was tested as part of the prototype controller as well as on a separate breakout board and as part of the A4964

development board. On the breakout board, the measured supply current jumped from 1 to 2 mA after a 25 kRad cumulative dose. On the controller and breakout board, device measurements of a 155 mA load remained constant across the entire test, while voltage measurements dropped from 12.00 V to 11.95 V. On the A4964 development board, current measurements remained stable, for example with the 3000 RPM current draw remaining steady at 146 mA. Voltage measurements trended lower over time, going from 11.98 to 11.93 V when running the motor at 3000 RPM.

MAX31855KASA+

Significant degradation was seen among the tested Maxim MAX31855 thermocouple amplifiers over the 30 kRad dose. First, error was seen in the internal cold junction reference temperature measurements (Figure 28). Second, error was noted in the compensated hot junction thermocouple measurements (Figure 29). These results showed that over the course of the mission, temperature measurements may become increasingly lower than the actual temperature. With the expected mission dose of up to 10 kRad, measurement errors of up to -15% can be expected.

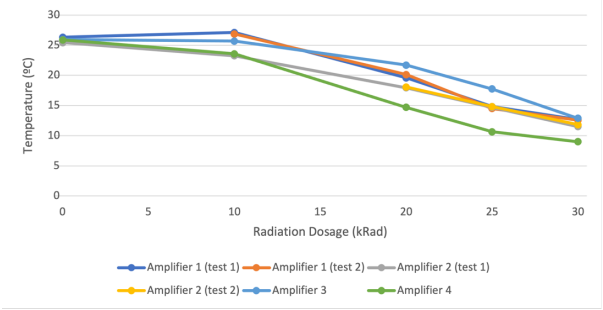

Figure 28: MAX31855KASA+ Thermocouple Amplifier TID Test Results: Internal Cold Junction Reference Temperature

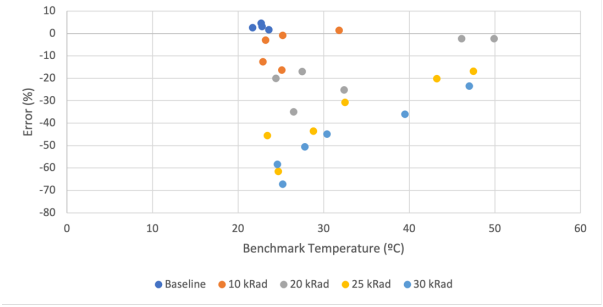

Figure 29: MAX31855KASA+ Thermocouple Amplifier TID Test Results: Hot Junction Temperature Error as Compared to Benchmark

It can be seen in Figure 29 that hot junction temperature measurement error is inversely correlated with temperature. In colder temperatures, cold junction compensation plays a larger role in determining an accurate temperature. This indicates that the cold junction measurement degradation likely plays a large role in the overall hot junction measurement degradation.

It is critical that appropriate temperatures are maintained during operation of the propulsion system. Fortunately, the results of this test can be used to generate compensated temperature setpoints which can be applied during the mission.

Current measurements of the MAX31855 breakout board remained constant at 2 mA, both in idle and measurement modes.

MCP6N16-010E/MS

The Microchip MCP6N16 instrumentation amplifier was tested via a breakout board. Functional tests between irradiation cycles evaluated the chip's performance with inputs disconnected as well as with 0.025 V, 0.05 V, and 0.075 V differential voltages. Across all tested differential input voltages, current drawn by the breakout board remained constant at 1 to 2 mA. These values remained steady across the 0 to 30 kRad test range.

Output voltages were recorded at each step and are given in Figure 30. Note that the baseline 0 kRad test results differ widely from the rest. This was likely due to a change in voltage meter averaging settings, and not due to actual degradation. The remaining values show that the component continued to give stable outputs all the way up to a 30 kRad dose.

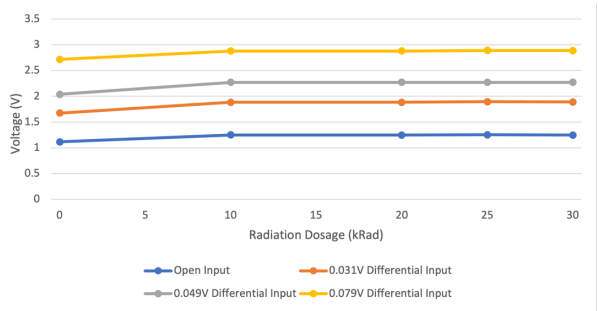

Figure 30: MCP6N16-010E/MS Instrumentation Amplifier TID Test Results: Output Voltage

REF3012AIDBZT

A Texas Instruments REF3012AIDBZT voltage reference was included on the prototype controller. The output voltage of this device was tested between irradiation phases, and found to remain constant between 1.250 and 1.251 V.

TCA9539QPWRQ1

A breakout board for the Texas Instruments TCA9539QPWRQ1 IO expander underwent TID testing. These components were also used on the prototype controller. Each device operated nominally throughout the TID tests. On the controller, IO expanders successfully enabled and disabled heater and valve drivers. On the breakout board, various IO expander channels were connected to loads and inputs for testing. When driving an LED on an output channel, 8 to 9 mA were consistently measured. When reading 3.3 and 0 V inputs, supply currents of 0 to 1 mA were consistently measured.

TLE4284DV33ATMA1

Degradation was noted in the output of the Infineon TLE4284DV33ATMA1 3.3 V linear regulator (Figure 31) which was installed on the prototype controller. This indicates that over the course of the mission, the voltage supplied to the controller's 3.3 V rail may drop. While this voltage is not directly used for any analog measurements other than the telemetered 3.3 V rail voltage, downstream components could behave improperly if the 3.3 V rail voltage drops sufficiently. The INA220 and MAX31855 components have the least margin, requiring a minimum of 3.0 V to operate. With the expected mission dose however, the output voltage of the TL4284DV33 should not drop below 3.2 V, leaving sufficient margin.

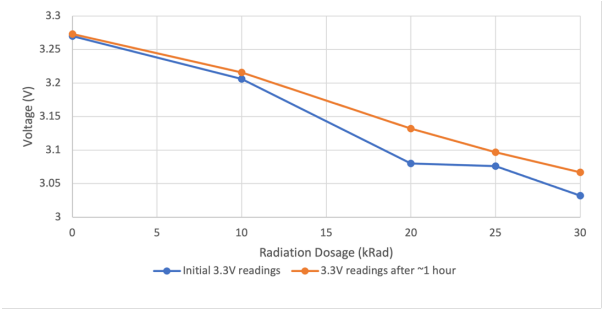

Figure 31: TLE4284DV33ATMA1 Voltage Regulator TID Test Results: Voltage Out

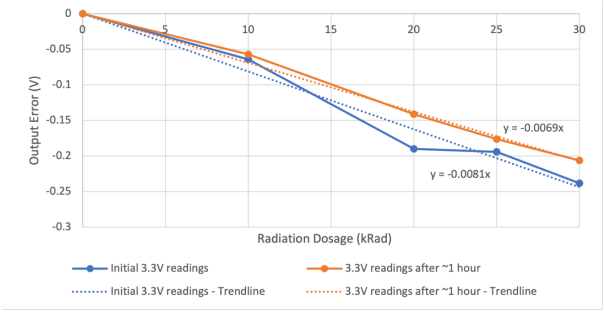

Figure 32: TLE4284DV33ATMA1 Voltage Regulator TID Test Results: Error from Baseline

TPS2553QDBVRQ1

The Texas Instruments TPS2553QDBVRQ1 was tested as part of the prototype controller. Two of the five current-interrupting channels were explicitly tested. At all times, each channel continued to operate under nominal loads without tripping. A resistive load of approximately 16.5 Ω was then added, which successfully tripped each circuit every time and cleared once the load was removed.

Random Vibration

Random vibration testing provides confidence that the device will be able to withstand the high vibration loads seen during launch. An EDU version of the controller underwent acceptance and qualification testing at the Georgia Tech Research Institute. The acceptance and qualification parameters for the LFPS controller were developed by NASA MSFC to cover the loads expected on the Space Launch System (SLS) vehicle. The test plan in Table 5 was followed.

Table 5: Random Vibration Test Plan

Test	Description
1	Full Functional Test of DUT
2	Axis 1 at Acceptance Levels
3	Limited Functional Test of DUT
4	Axis 2 at Acceptance Levels
5	Limited Functional Test of DUT
6	Axis 3 at Acceptance Levels
7	Full Functional Test of DUT
8	Axis 1 at Qualification Levels
9	Limited Functional Test of DUT
10	Axis 2 at Qualification Levels
11	Limited Functional Test of DUT
12	Axis 3 at Qualification Levels
13	Full Functional Test of DUT

A fixture was used to mount the controller to the vibration table (Figure 33). Accelerometers were mounted to the fixture and controller for system feedback and data logging (Figure 34). Functional tests performed between vibration tests showed that the controller was capable of withstanding the tested levels without any functional degradation.

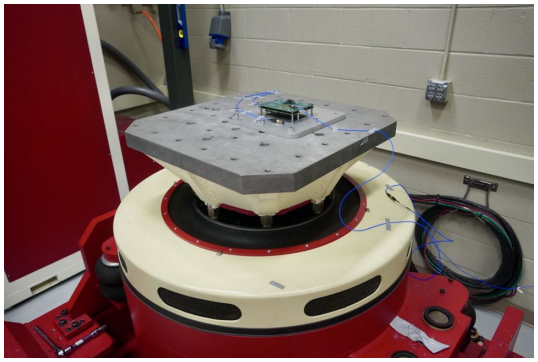

Figure 33: Random Vibration Z-Axis Setup

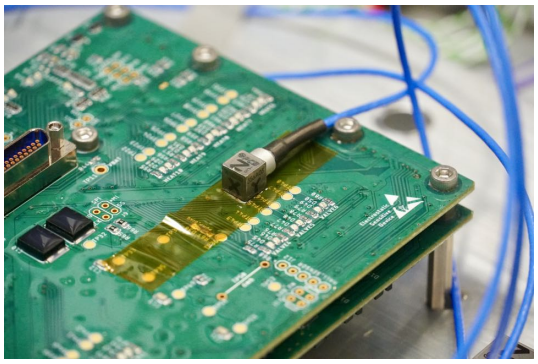

Figure 34: Detail of Random Vibration Setup

Thermal Vacuum

Thermal vacuum cycling is a method of confirming that the device will properly operate in space. First, a vacuum is drawn in the test chamber, simulating the vacuum of space. Second, the device is thermally cycled. This simulates the thermal stress seen due to operation and cycling in and out of view of the sun. A thermal vacuum test was designed around the minimum and maximum operational and non-operational temperature limits provided as project requirements (Figure 35). A rough vacuum upper limit of 7.5×10^{-1} torr was provided as a requirement, but the chamber was seen to reach 2.75×10^{-4} torr during the test.

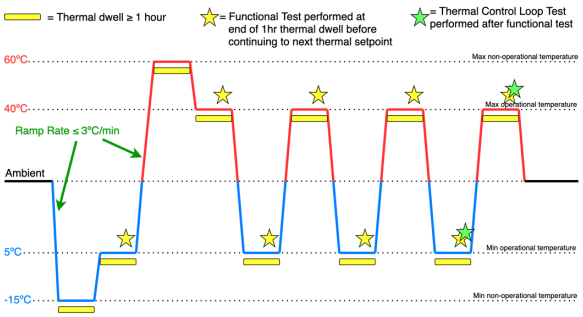

Figure 35: Thermal Vacuum Test Procedure

Initial attempts to run the thermal vacuum cycle at Georgia Tech failed due to an inability to bring the device down to the −15 °C temperature. With a minimum coolant temperature of −20 °C, the chamber could not extract enough heat from the device. At first, the controller would not fall below 7.8 °C even after a 7 hour attempt. An assortment of techniques followed which successfully brought the device down to the necessary temperature (Figure 36).

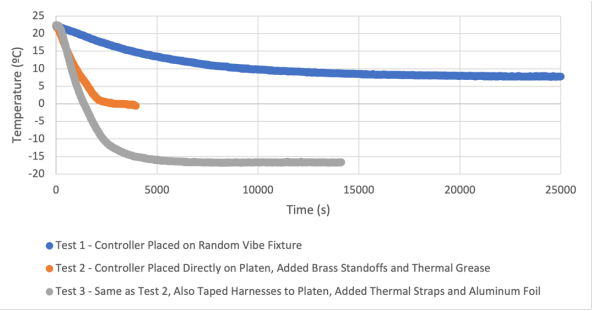

Figure 36: Thermal Vacuum Temperature Response to Various Thermal Management Approaches, Measured at Base of Controller with Platen Coolant Lowered to −20 °C

First, the controller was removed from an intermediate aluminum fixture, stainless steel standoffs were replaced with brass standoffs, and thermal grease was added between the standoff/platen interface. This setup allowed the controller to reach −0.6 °C after about 1 hour.

Second, copper thermal straps were added, with thermal grease applied to all strap interfaces. Chamber pass-through harnesses were taped down to the platen to reduce their ability to act as a heat pipe bringing in external heat. Aluminum foil was also placed over the controller to reflect heat radiated from the chamber walls. This setup allowed the controller to reach the necessary −15 °C in about 1 hour. Figure 37 shows the controller in this configuration but without the aluminum foil.

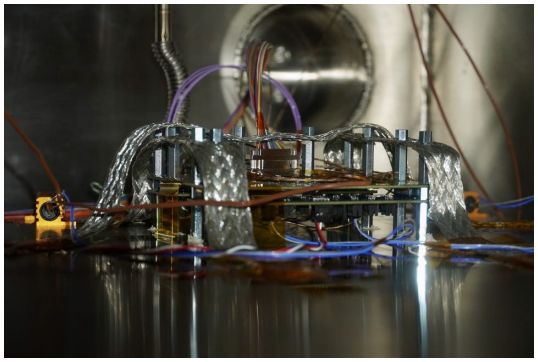

Figure 37: Controller Undergoing Thermal Vacuum Test

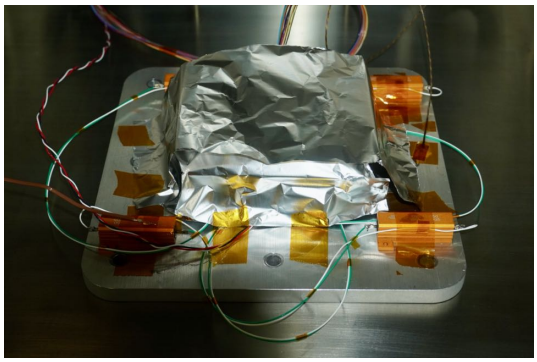

Figure 38: Setup of Heater Driver Temperature Characterization Test in Vacuum Chamber

Throughout the prescribed thermal cycles while under vacuum, simplified functional tests returned positive results. After removing the device from vacuum, a full functional test was performed to identify any degradation and none was found. R-type thermocouple readings initially looked concerning, but replacing the externally attached test thermocouple with a new one showed that the device continued to read temperatures nominally.

Operational Temperature Characterization

The 0.5 Ω nominal on-resistance of the DRV103H prompted a concern that the controller would overheat when warming up the thruster catalyst beds. A test was developed to characterize the expected temperatures in this operational mode, and was performed on the PCB version expected for flight use.

Four 8.2 Ω chassis mount resistors were selected to emulate the catalyst bed heaters. The actual heaters were expected to quickly rise above 9 Ω. It was also expected that the catalyst heaters would reach operational temperatures within 30 minutes of startup, so for each temperature characterization test, the heaters were run for 1 hour to provide sensible margin.

Various test runs inside a vacuum chamber were performed with slightly modified parameters in an effort to best simulate the expected environment. Aluminum foil was added around the entire test article to better simulate the controller surrounded by its thermally reflective titanium shield. This was then modified so the foil only covered the controller (Figure 38), instead of also reflecting heat from the chassis mount resistors back onto the controller.

In each test, different maximum temperatures were reached due to changes in the test setup. In a particularly overconservative test with resistor heat reflected back onto the controller with aluminum foil, a controller temperature of approximately 125 °C was reached. The DRV103H drivers began to thermally limit themselves when the controller reached 120 °C after 45 minutes. The final test configuration most accurately emulated the expected environment, and showed the controller reaching 60 °C at 13 minutes, 70 °C at 21 minutes, 80 °C at 30 minutes, and 95 °C at 55 minutes (Figure 39). During all of these tests, the only noted performance degradation was internal rate-limiting by the DRV103H when the controller reached 120 °C.

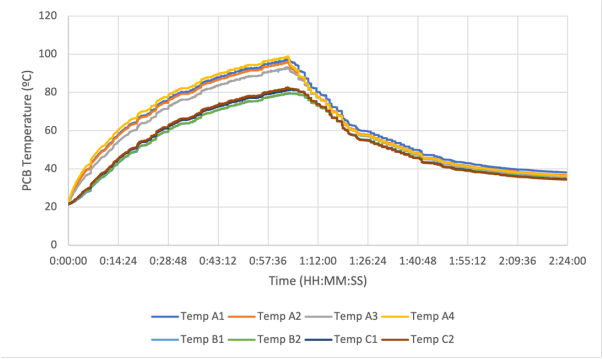

Figure 39: Controller PCB Temperature During Catalyst Bed Heater Operational Temperature Characterization Test with Four 8.2 Ω Loads

As the expected warmup time was lowered from 30 minutes due to later analysis, it became clear that the controller should be able to withstand this heatup sequence without exhibiting degraded performance or exceeding individual component operating temperature ratings.

CONCLUSION

Following the conclusion of functional and environmental testing on the EDU controllers, flight boards were procured and assembled. After passing functional tests, the boards were then staked and conformal coated (Figure 40). These flight boards then underwent random vibration and thermal vacuum acceptance testing similar to the tests performed on the EDU controller. One unit has now been installed into a flight propulsion system by Georgia Tech and is awaiting integration into the spacecraft (Figure 41).

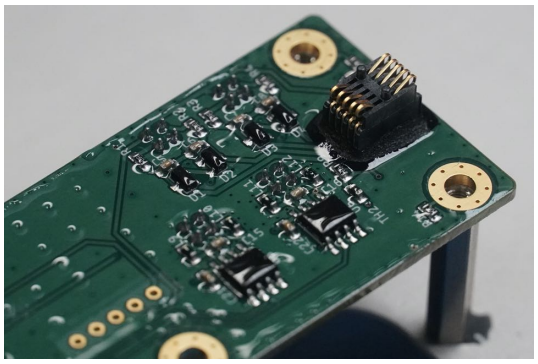

Figure 40: Flight Board after Staking and Conformal Coating

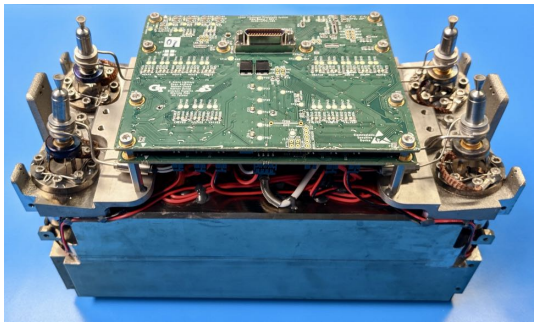

Figure 41: Flight Boards Integrated into Propulsion System

Georgia Tech is finishing development of a COTS-based green monopropellant propulsion system controller that is expected to play a critical role in enabling the first CubeSat to reach the Moon. The success of this mission will provide evidence that this low-cost approach to electronics development can be used for systems of increasing complexity, enabling new possibilities for small satellites.

ACKNOWLEDGEMENTS

This work was made possible through NASA contracts 1631930 and 80MSFC19M0044. A portion of this research was also carried out at the Jet Propulsion Laboratory, California Institute of Technology, under NASA contract 80NM0018D0004. Additionally, the design, manufacturing, and testing of this system would not have been possible without the help of many individuals.

At the NASA Marshall Space Flight Center, Donald McQueen, Carlos Diaz, Christopher Burnside, Hunter Williams, and Mignon Thames all offered valuable insights and logistical support throughout the development and testing of this system. Additionally, Clark Teems and Oren Kornberg provided excellent thermal modeling support.

At the NASA Jet Propulsion Laboratory, John Baker, Philippe Adell, and Matthew Kowalkowski provided guidance and operational support on many occasions. Additionally, Greg Allen, Sergeh Vartanian, and Farokh Irom supported this project through SEL radiation component testing. Recognition is also due to Bernard Rax and Jason Thomas for the long days and late evenings spent running the TID radiation testing facility.

At the Georgia Institute of Technology, Dawn Andrews, Grayson Huggins, Ali Talaksi, Brandon Colón, and Lacey Littleton provided excellent support from the structural design perspective, working hard to ensure the controller electronics would integrate properly into the system. On the firmware front, Scott Gilliland offered valuable support in building out functionality and squashing virtual bugs. William Jun and Ryan Lewis helped make environmental testing possible at Georgia Tech by supporting thermal vacuum and random vibration testing, respectively.

References

[1] B. A. Cohen, P. O. Hayne, B. Greenhagen, D. A. Paige, C. Seybold, and J. Baker, "Lunar Flashlight: Illuminating the Lunar South Pole," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 3, pp. 46–52, 2020.

[2] D. Andrews, G. Huggins, E. G. Lightsey, N. Cheek, N. Daniel, A. Talaksi, S. Peet, L. Littleton, S. Patel, L. Skidmore, M. Glaser, D. Cavender, H. Williams, D. McQueen, J. Baker, and M. Kowalkowski, "Design of a Green Monopropellant Propulsion System for the Lunar Flashlight CubeSat Mission," in *Proceedings of the Small Satellite Conference*, 2020.

[3] NASA MSFC, "MSFC-STD-3012: Electrical, Electronic, and Electromechanical (EEE) Parts Management and Control Requirements for MSFC Space Flight Hardware," 2012.

[4] Automotive Electronics Council, "AEC-Q100-REV-H: Failure Mechanism Based Stress Test Qualification for Integrated Circuits," 2014.

[5] NASA GSFC, "Single Event Effects." `https://radhome.gsfc.nasa.gov/radhome/see.htm`. (Accessed 2021-05-23).

[6] M. Johnson, R. Cline, S. Ward, and J. Schichl, "Texas Instruments White Paper SCAA124: Latch-Up," 2015.

[7] D. Sinclair and J. Dyer, "Radiation Effects and COTS Parts in SmallSats," in *Proceedings of the Small Satellite Conference*, 2013.

[8] R. L. Bocchino Jr., T. K. Canham, G. J. Watney, L. J. Reder, and J. W. Levison, "F Prime: An Open-Source Framework for Small-Scale Flight Software Systems," in *Proceedings of the Small Satellite Conference*, 2018.

[9] "F Prime: A Flight-Proven, Multi-Platform, Open-Source Flight Software Framework." `https://github.com/nasa/fprime`. (Accessed 2021-05-23).

[10] "AVR Options." `https://gcc.gnu.org/onlinedocs/gcc/AVR-Options.html`. (Accessed 2021-05-23).

[11] "Python Socket.IO Server and Client." `https://github.com/miguelgrinberg/python-socketio`. (Accessed 2021-05-23).

[12] G. Allen and S. Vartanian. Private Communication, 2019.

[13] F. Irom. Private Communication, 2021.