

# Design of an Experiment to Evaluate High-Power Rockets as a CubeSat Qualification Platform

Author: Peter Tanner

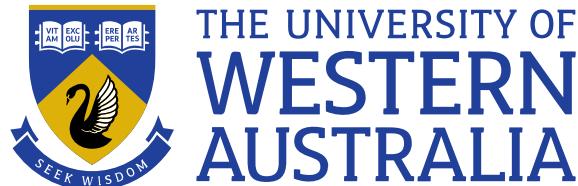
Supervisor: Dilusha Silva

ATTENTION: THIS IS A DRAFT VERSION. TODO: CHECK CHECKLIST  
BEFORE SUBMITTING *This thesis is presented in partial fulfilment of the  
requirements for the degree of Bachelor of Philosophy (Honours) at the University  
of Western Australia*

Faculty of Engineering and Mathematical Sciences

Word count: TODO:

Submitted: November 12, 2024



## 0.1 TODO: (!! REMOVE BEFORE FINAL RELEASE !!)

- [ ] Add schematics and PCB gerbers to appendix using `pdfpages`
- [ ] Add links to source repository <https://git.petertanner.dev/hpr-evaluation/>
- [ ] Use L<sup>A</sup>T<sub>E</sub>Xtemplate for Honours papers
- [ ] Resolve all warnings except badness warnings because I don't know how to control that
- [ ] Make BibTeX consistent
- [ ] Bad citation format for datasheets and websites (Date is after access date).
- [ ] Check grammar and spelling
- [ ] Check presentation/typesetting
- [ ] Resolve all TODO: comments
- [ ] ADD ENGINEERING COVERSHEET WITH HONORS DECLARATION.
- [ ] Finish all items on this checklist

### Wts

- PROJECT BODY (ASSUME EXPERIMENTAL PROJECT)
  - 20% INTRODUCTION AND LITERATURE REVIEW
  - 15% EXPERIMENTAL DESIGN
  - 35% RESULTS AND DISCUSSION
- 10% CONCLUSION AND FUTURE WORK
- 10% SCOPE
- 10% PRESENTATION (SHOULD BE GUARANTEED...)

### USING FORMULA WC \* SECTION/80%

- INTRODUCTION AND LITERATURE REVIEW: 3000 TO 4500 WORDS
- EXPERIMENTAL DESIGN: 2250 TO 3375
- RESULTS AND DISCUSSION: 5250 TO 7875
- CONCLUSION AND FUTURE WORK: 1500 TO 2250

## Abstract

The CubeSat is a type of small satellite created to reduce the cost of access to space for universities due to its standardised  $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$  form factor. The total number of CubeSats launched into space is growing exponentially due to their low cost, doubling every 2.5 y, however the mission success rate has stalled at 75% since 2018 [1], [2].

Vibration and shock tests are qualification procedures which aim to emulate launch conditions, however they cannot perfectly replicate them [3]. Testing of CubeSats on suborbital high-power rockets (HPR) is a novel qualification method that can potentially better replicate launch conditions compared to a shaker table, and therefore better detect issues and improve the likelihood of mission success. While there have been tests of university CubeSats on HPRs [4], there are no direct comparisons to shaker tables to evaluate their effectiveness for qualification.

This paper outlines the design of a data acquisition system (DAQ) to measure vibration from the HPR launch and shaker table tests and finally makes a direct comparison of the vibration environment on the HPR launch and vibration table.

## Acknowledgements

I'd like to thank all the people and organisations who have supported me throughout this project. Dilusha Silva for being a wonderful mentor and for coordinating the project. Michal Zawierta for his expertise flying drones for the drone tests of the CubeSat. Jamir Khan for being a wonderful friend and engineer who worked on the mechanical side of this project, including construction of the high-power rocket, and for putting up with all my delays. Timothy Ludovico for designing the camera payload and being all around wonderful to work with. Jeremy Marelich and AVI for providing their shaker table facilities and conducting the tests. UWA Aerospace for being a wonderful institution who has been with me from first year through my growth as an engineer and has supported me through this project. Space Angel for creating this project and providing expertise and connections to the Indian Institute of Space Science and Technology (IIST). Dr. Priyadarshnam Hari and the Indian Institute of Space Science and Technology for providing their launch expertise and opportunity to launch on POEM. International Space Centre for supporting this project with funding.

# Contents

0.1 TODO: (!! REMOVE BEFORE FINAL RELEASE !!) . . . . .	1
<b>1 Introduction</b>	<b>10</b>
1.1 Background . . . . .	10
1.2 Problem Identification . . . . .	11
<b>2 Literature Review</b>	<b>12</b>
2.1 Standard Satellite Qualification Methods . . . . .	12
2.1.1 Vibration . . . . .	12
2.1.2 Shock . . . . .	15
2.1.3 Functional testing . . . . .	15
2.2 Rocket Testing of CubeSats . . . . .	16
2.2.1 Sounding Rockets . . . . .	16
2.2.2 High-Power Rockets (HPR) . . . . .	17
2.2.3 Avionics Systems . . . . .	19
<b>3 Design Process</b>	<b>21</b>
3.1 Identification of Project Goals . . . . .	21
3.2 Design Group and Scheduling . . . . .	22
3.3 High-Level Constraints and Requirements . . . . .	22
3.3.1 Electrical Power System (EPS) . . . . .	23
3.3.2 Environmental Requirements . . . . .	24
3.3.3 Camera Payload HPR Test Requirements . . . . .	24
3.3.4 HPR Vibration Experiment Requirements . . . . .	24
3.3.5 Other Requirements . . . . .	25
3.4 Selection of Components and High-Level System Design . . . . .	25
3.4.1 High-Level Design . . . . .	25
3.4.2 Components Sourcing . . . . .	26
3.4.3 High-Level EPS and Power Components Selection . . . . .	27
3.4.4 Power Electronics . . . . .	28
3.4.5 Radio Downlink . . . . .	30
3.4.6 Payload Communications . . . . .	31
3.4.7 GNSS Tracking . . . . .	32
3.4.8 Accelerometers . . . . .	33
3.4.9 Onboard Data Handling Unit (OBDH) . . . . .	33
3.5 Design Tools . . . . .	34

3.5.1	Altium Designer 24 . . . . .	34
3.5.2	circuit.js . . . . .	36
3.5.3	LTspice . . . . .	36
3.6	Implementation of High-Level Design . . . . .	36
3.6.1	Design Verification and Schematic Capture . . . . .	37
3.6.2	PCB Setup . . . . .	38
3.6.3	PCB Layout . . . . .	38
3.6.4	Finalisation of Design and Manufacturing . . . . .	38
3.6.5	Software Design Process . . . . .	39
3.6.6	Integration Testing . . . . .	39
3.7	Design Evaluation Framework . . . . .	39
3.7.1	Environmental Testing . . . . .	40
3.7.2	Shaker Table Test . . . . .	41
3.7.3	Vehicle Tests . . . . .	43
3.7.4	Experiment Evaluation . . . . .	43
3.8	Version One of DAQ . . . . .	43
3.8.1	Overview of First Design . . . . .	44
3.8.2	Evaluation of First Design . . . . .	44
3.8.3	First Revision Implications . . . . .	45
<b>4</b>	<b>Final Design of DAQ</b>	<b>46</b>
4.1	Overview of PCBs . . . . .	46
4.1.1	PCB Sizes . . . . .	46
4.1.2	PCB Stackups . . . . .	46
4.2	Electrical Power System (EPS) . . . . .	47
4.2.1	Power Budget . . . . .	47
4.2.2	Battery Selection . . . . .	48
4.2.3	Battery Pack . . . . .	49
4.2.4	5 V Power Conditioning Design and Simulation . . . . .	50
4.2.5	Internal 3.3V Power Conditioning . . . . .	51
4.2.6	EPS Subsystem Design . . . . .	52
4.2.7	Integration into DAQ . . . . .	52
4.3	Onboard Data Handling (OBDH) . . . . .	54
4.3.1	Camera Communications Interface and Camera Data Downlink . . . . .	54
4.3.2	Accelerometer Data Acquisition System . . . . .	60
4.3.3	GNSS Tracking . . . . .	65
<b>5</b>	<b>Final Experiment Design for HPR Evaluation</b>	<b>66</b>
5.1	High-Power Rocket Flight . . . . .	66
5.1.1	Simulation . . . . .	67
5.1.2	HPR Experiment Setup . . . . .	69
5.1.3	HPR Data Analysis . . . . .	71
5.1.4	Shaker Table Experiment Setup . . . . .	73
5.2	Shaker Table . . . . .	73
5.2.1	Random . . . . .	73
5.2.2	Quasi-static Acceleration Test . . . . .	73

5.2.3	Shock . . . . .	74
5.3	Evaluation of HPR as a CubeSat Qualification Platform . . . . .	74
<b>6</b>	<b>DAQ Evaluation and Discussion</b>	<b>76</b>
6.1	Drone Tests . . . . .	76
6.1.1	First Drone Test . . . . .	76
6.1.2	Second Drone Test . . . . .	78
6.1.3	Third Drone Test . . . . .	79
6.1.4	Discussion . . . . .	82
6.2	Temperature Testing . . . . .	83
6.2.1	High-Temperature Test . . . . .	83
6.2.2	Low-Temperature Test . . . . .	84
6.2.3	Discussion . . . . .	85
6.3	High-Power Rocket Test Flight . . . . .	85
6.3.1	Discussion . . . . .	87
6.4	Shaker Table . . . . .	87
6.4.1	Random . . . . .	87
6.4.2	Shock . . . . .	89
6.4.3	Discussion . . . . .	90
<b>7</b>	<b>Evaluation of HPR Flight as a Qualification Platform</b>	<b>91</b>
7.1	Random . . . . .	91
7.2	Shock . . . . .	92
7.3	Discussion . . . . .	94
<b>8</b>	<b>Conclusions and Future Work</b>	<b>95</b>
8.1	Future Work . . . . .	95
8.2	Conclusions . . . . .	96
8.3	Appendix . . . . .	97
8.3.1	Code . . . . .	97
8.3.2	References . . . . .	97

## List of Figures

2.1	Random vibration test [13] . . . . .	13
2.2	Quasi-static acceleration test. The input profile high acceleration from 20 Hz to 21 Hz, resulting in the response having a force of 10.8 $g$ acceleration around this frequency [13] . . . . .	14
2.3	Coupled loads model [17] . . . . .	14
2.4	Shock response spectrum of and time-domain shock response. Left: near-field (close to shock source). Right: far-field (distant from shock source) [19] . . . . .	15
2.5	Acceleration in time domain (Left), Angular velocity in time domain (Right) during the launch of FloripaSat-I [8] . . . . .	16
2.6	Random vibration (Left) and sine sweep (Right) tests on a shaker table during the qualification of FloripaSat-I [8] . . . . .	17
2.7	Typical launch of a single stage high-power rocket . . . . .	18
2.8	Shock response spectrum from computer modelling of an igniter based on the low explosive aluminium potassium perchlorate [26] . . . . .	19
3.1	High-level overview of design process . . . . .	21
3.2	Responsibilities of members on the CubeSat design project and the information required to be communicated between each member. . . . .	22
3.3	Block diagram of the CubeSat, including connections to the camera payload and ground station over radio downlink. . . . .	26
3.4	(Left) 1P battery pack using $2N = 4$ batteries, (Right) 2P battery pack using $2N = 4$ batteries. . . . .	29
3.5	Example of the hierarchical schematic sheet format for the main DAQ PCB. . . . .	35
3.6	Workflow for integrating a design into a PCB. . . . .	37
3.7	High-temperature testing setup . . . . .	40
3.8	Low-temperature testing setup . . . . .	41
3.9	IIST recommended random vibration test profile for qualification of CubeSat for launch on POEM (profile defined in 3.12). . . . .	42

4.1	S21 attenuation parameters of several dielectrics for a $50\Omega$ microstrip. Compared to the high-performance Rogers 4350 dielectric, there is negligible difference in attenuation at low frequencies. Adapted from [44] . . . . .	47
4.2	Battery mounting on reverse side of DAQ PCB. . . . .	49
4.3	LTspice schematic representation of TPS61022. . . . .	50
4.4	Simulation of the TPS61022 in LTspice using component values close to the datasheet as shown in figure 4.3. . . . .	51
4.5	Electrical power system (EPS) and output voltages . . . . .	52
4.6	Layout of TPS61022 implemented in EPS. . . . .	53
4.7	PCB layout of internal power conditioning, includes the layout of the AMS1117 regulator and another instantiation of the TPS61022 layout. . . . .	53
4.8	Pinout of DE-9 connector. . . . .	55
4.9	Message structure. . . . .	56
4.10	Example of an image received with errors resulting in an unrecognisable image due to dropped bytes causing the image to shift at each error. Only a small proportion of the image is lost ( $< 3\%$ ), represented by red pixels at the bottom and contiguous sections of the received image are error-free. . . . .	57
4.11	Image chunk protocol. . . . .	58
4.12	Comparison of two image reception techniques. Image (a) used the blocking protocol with a block size of 400 pixels. Image (b) is an example of what the image would look like without the image blocking. . . . .	60
4.13	Accelerometer PCB. . . . .	61
4.14	Block diagram of accelerometers and Pi Zero data acquisition system. . . . .	62
4.15	Mounting positions of accelerometers on the CubeSat chassis. The top and bottom accelerometer share a common $x$ axis but have opposite $y$ and $z$ axes. . . . .	63
5.1	OpenRocket diagram of <i>UNO</i> . . . . .	67
5.2	Flight profile of <i>UNO</i> using a K1100T motor. Simulated in OpenRocket. . . . .	67
5.3	Stability of <i>UNO</i> using a K1100T motor. Simulated in OpenRocket. . . . .	68
5.4	Acceleration of <i>UNO</i> using a K1100T motor over (top) the whole flight and (bottom) the thrust phase. Simulated in OpenRocket. . . . .	69
5.5	CubeSat in <i>UNO</i> before launch. . . . .	70
5.6	Data processing workflow for HPR accelerometer data. . . . .	72
5.7	CubeSat setup in on the shaker table. . . . .	73
5.8	Realised shock profile. . . . .	74
6.1	DJI M600 drone with CubeSat mounted sideways before flight. . . . .	77
6.2	Mounting the CubeSat to the DJI M600 drone for the second drone test. . . . .	79
6.3	Grayscale image of sheds taken at 120 m AGL. The red pixels are dropped bytes. . . . .	80
6.4	Grayscale image of trees and roads from 120 m AGL. The red pixels are dropped bytes. . . . .	81
6.5	Screen capture of the ground station during live image download from the CubeSat. The terminal shows the periodic uptime messages which were sent to indicate that the system was online while the camera was capturing an image. A local Python script reassembles the image from the download stream. Red pixels have yet to be downloaded at this point. . . . .	82
6.6	High-temperature testing setup . . . . .	84

6.7	Low-temperature testing setup . . . . .	85
6.8	Time domain flight data from LSM6DSOX accelerometer. . . . .	86
6.9	Noisy time domain flight data from ADXL375 accelerometer. . . . .	86
6.10	Response of payload from shaker table as recorded by DAQ and IEPE accelerometers. . . . .	88
6.11	Noisy accelerometer from the DAQ. . . . .	88
6.12	Response of 8ms 50 g half-sine shock. The two discontinuities in the shaker table response are a result of the table moving to ensure it stays within the maximum displacement limit. . . . .	89
6.13	Battery becoming unmounted after shock testing . . . . .	90
7.1	Comparison of shaker table and HPR responses. Shaker table response was captured using IEPE accelerometers, the HPR responses were captured using DAQ accelerometers. . . . .	92
7.2	Shock response of pyroshock and half-sine profile. The two discontinuities in the shaker table response are a result of the table moving to ensure it stays within the maximum displacement limit. . . . .	93
7.3	Shock response of pyroshock and half-sine shock. . . . .	94

## List of Tables

3.1	High-level requirements . . . . .	23
3.2	Operating voltage and current consumption of devices connected to EPS in the final DAQ design. . . . .	27
3.3	Battery cell requirements . . . . .	28
3.4	Comparison of 1S and 2S battery packs. . . . .	29
3.5	Boost converter requirements . . . . .	30
3.6	Linear regulator requirements . . . . .	30
3.7	Radio downlink requirements . . . . .	31
3.8	Payload communications requirements . . . . .	32
3.9	GNSS tracking requirements . . . . .	33
3.10	Accelerometer requirements . . . . .	33
3.11	OBDH requirements . . . . .	34
3.12	IIST recommended random vibration test profile for qualification of CubeSat for launch on POEM. . . . .	42
3.13	IIST recommended shock test profile for qualification of CubeSat for launch on POEM. . . . .	43
3.14	Data sources and their data rates. . . . .	44
3.15	CrystalDiskMark benchmark of DAQ v1. . . . .	45
4.1	Operating voltage and current consumption of devices connected to EPS. . . . .	48
4.2	UART settings. . . . .	56
5.1	Rocket motor impulse classes [56] . . . . .	66
5.2	Comparisons between the two tests and their representation. . . . .	74
6.1	Lost bytes statistics for an image with 1920000 pixels. . . . .	82

# Chapter 1

## Introduction

### 1.1 Background

The University of Western Australia (UWA) Microelectronics Research Group (MRG) is developing a 2U CubeSat to measure the health of vegetation through an infrared camera array [5]. The CubeSat is a type of small satellite created to reduce the cost of access to space for universities due to its standardised  $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$  form factor. This CubeSat will launch on an Indian Polar Satellite Launch Vehicle (PSLV) in the PSLV Orbital Experiment module (POEM), which will host multiple CubeSats in orbit and will provide services including power and communications to the CubeSat.

The total number of CubeSats launched into space is growing exponentially due to their low cost, doubling every 2.5y, however the mission success rate has stalled at 75% since 2018 [1], [2], which implies a need for novel qualification methods. For most single-launch satellites, increased testing is the optimal strategy to minimise failure [2]. Qualification of the CubeSat is required by the launch provider to minimise the risk of damage to the launch vehicle or other payloads from mission failure. The MRG is planning to qualify this CubeSat on a suborbital high-power rocket (HPR) in combination with traditional vibration and shock tests on a single degree of freedom (SDOF) electrodynamic shaker table.

Vibration and shock testing are typical tests for CubeSats which are intended to replicate the conditions of launch [1]. Despite their widespread use, SDOF vibration and shock tests do not perfectly replicate the conditions at launch as [3], [6]:

1. The peak flight responses are not able to be achieved since a vibration table cannot simulate steady-state thrust forces since they only can simulate dynamic forces [3].
2. A SDOF test can only excite one axis at a time which is not representative of the launch environment [3], [6].
3. A vibration table tests a "fixed-base" case which has different modes compared to the case where the satellite is fixed to the launch vehicle [3].

A HPR has a higher total impulse than model rockets but a lower impulse than sounding rockets, with a range of 36 N s up to 163 840 N s, and have a sub-orbital trajectory unlike launch vehicles [7].

Suborbital rockets have been used for testing several CubeSats [8], [9], however this qualification method is not in widespread use in the industry.

## 1.2 Problem Identification

For institutions with limited budget, shock and random vibration tests using a SDOF vibration table is the current state of the art (SOTA) method for qualification. HPRs are a potential qualification method which can complement SDOF vibration tests, however there is no prior studies comparing both HPRs and SDOF vibration tests against the qualification level set by the launch provider. If HPRs can produce a vibration environment similar to the qualification level, HPRs may be a useful complement to SDOF vibration tests and may increase success rates.

Since HPRs are not frequently used, another issue is the lack of tooling for making HPRs an effective test platform. This research will involve design and evaluation of a combined test and data acquisition platform which:

1. Measures the vibration response of the rocket on the CubeSat required for evaluating the HPR platform and
2. Emulates the POEM satellite bus services to ensure the payload-under-test has access to the same environment as on launch.

An experiment will be designed using the DAQ data to evaluate HPRs.

# Chapter 2

## Literature Review

This literature review will cover the current testing methods used in CubeSats, the use of suborbital rockets as a qualification method and cover the types of sensors and systems required to record these tests.

### 2.1 Standard Satellite Qualification Methods

There are multiple satellite qualification standards, an example is the NASA General Environmental Verification Specification (GEVS) which is a panel of tests including electromagnetic compatibility (EMC), thermal, acoustic and vibration tests that are required for all NASA Goddard Space Flight Center projects [10]. Other standards include ISO-15864, JERG-2-002, NASA-STD-7002A, ECSS-E-ST-10-03C and SMC-S-01 [11]. While these standards have flight heritage, being used on many successful payloads, they were designed for medium or large satellites, and therefore fully complying with these standards are out of the budget of most university CubeSat programs [11]. While is no widely used test standard for CubeSats currently in use, since most CubeSat projects are budget constrained, there is a de facto minimum series of tests which are random vibration, shock and thermal vacuum testing [1].

#### 2.1.1 Vibration

Vibrations are experienced by satellites during transportation and loading, and most prominently during launch [12]. The purpose of vibration testing is to ensure that the satellite will survive transportation and launch conditions, and to find workmanship errors [3], [12].

##### Random Vibration Test

In the random vibration test, a vibration profile is applied to the satellite which tests all the resonant frequencies of the satellite [13]. This range includes frequencies on the order of 100 Hz to 1000 Hz, since higher frequencies couple to the satellite through acoustic means rather than through the structure [3]. An example of a random vibration profile and its response is shown in figure 2.1. A sine sweep vibration test is similar, but instead of the frequency being randomly sampled it is swept through sequentially from either low to high frequency or vice versa.

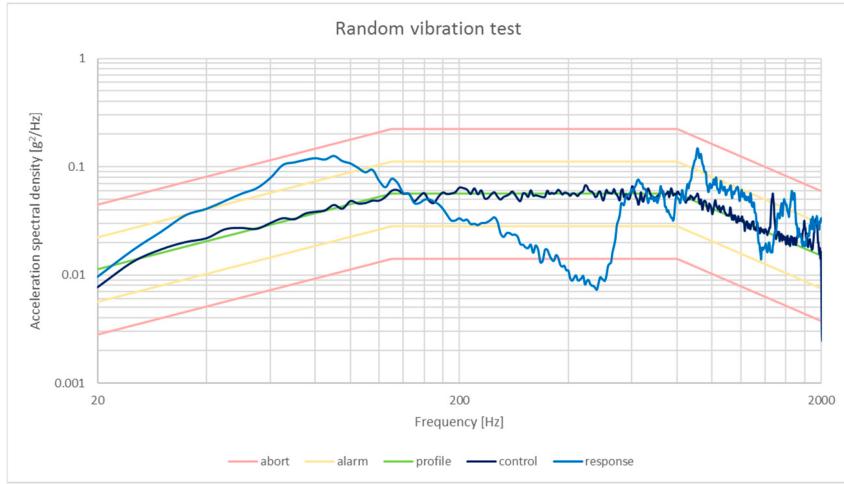


Figure 2.1: Random vibration test [13]

The limitations of random vibration tests is that the shaker and table will have different modes than the launch vehicle and payload mount, resulting in the test response not perfectly matching the flight response [3], [14]. Gordon and Kern argue that this difference is not a factor in practice since shaker tests are "not intended to be a strength test" [3, p. 7] and that components "should have been strength qualified prior to integration" [3, p. 7]. Component level is argued as a best practice in the CubeSat community [15], however some argue that component level testing is not suited to the short timeline of university CubeSat projects and that more effort should be put into integration testing [16]. If a testing program focuses on integration testing, then this mismatch between shaker table and flight response could result in the CubeSat not being properly qualified.

While 6 degrees of freedom (DOF) shaker tables exist which can replicate the vibrations experienced in all dimensions during launch, most satellites are still tested with single-axis shaker tables [3], [6], [14]. While Gordon and Kern [3] state that these limitations are adequately managed by testing in all three orthogonal axes separately, Aglietti and Nath [6] created a model of three, two and single axis vibration tests and found that to match the 3 DOF response with a single DOF table, the satellite needed to be subjected to 2.5 times the  $g_{\text{rms}}$  forces than in 3 DOF testing, leading to the satellite being over designed [6].

### Quasi-Static Acceleration Test (QAT)

A quasi-static test simulates liftoff, where there is a combination of random vibration from engines and quasi-static acceleration from the engine and other external forces on the launch vehicle [12], [13], which are approximated as constant forces at selected frequencies as shown in figure 2.2. The QAT is usually compared to results from coupled loads analysis, where all forces are assumed to be applied to the satellite through the launch vehicle as shown in figure 2.3 [17].

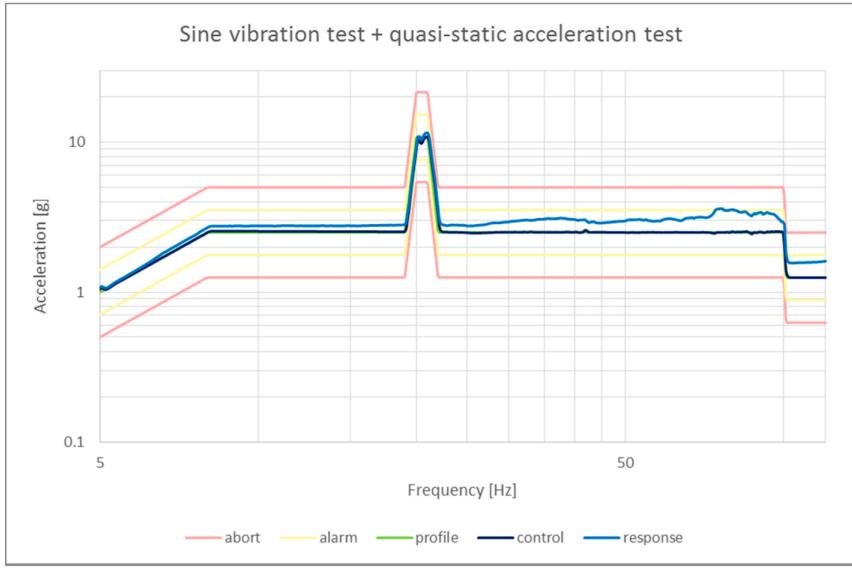


Figure 2.2: Quasi-static acceleration test. The input profile high acceleration from 20 Hz to 21 Hz, resulting in the response having a force of  $10.8\text{ g}$  acceleration around this frequency [13]

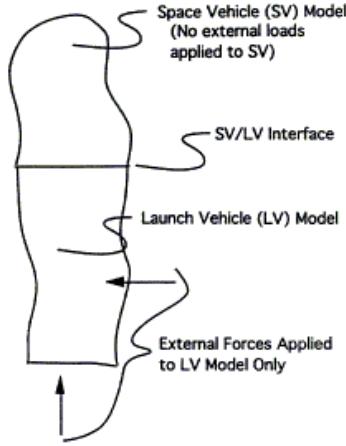


Figure 2.3: Coupled loads model [17]

The first limitation of a quasi-static acceleration test is that the shaker table cannot apply the peak response evenly that is predicted by coupled loads analysis (CLA) [3]. Again, Gordon and Kern state that these limitations are addressed by component-level strength qualification. They also state that applying the peak response evenly is not necessary, since if an item does not fail, the correctly applying the response evenly does not greatly increase its likelihood of failing [3]. The second limitation is there is a difference in modes, since a quasi-static acceleration test also contains random vibrations [3].

### 2.1.2 Shock

Shock is experienced by satellites when pyrotechnics are detonated or deflagrated during events such as staging and ignition, the response appears as a range of decaying sinusoids in the 100 Hz to 10 kHz frequency range [12], which decay in 5 ms to 15 ms [12]. The spectrum extends up to 40 kHz, however for analysis frequencies above 10 kHz are assumed to be non-damaging [18], [19]. Pyroshock may cause peak accelerations of up to 10000  $g$  [19]. High explosives are primarily used for explosive elements on rockets in combination with some low explosives for initiators [18].

Shock is tested using a shock-generating device which is applied to the satellite along all three axes [10], [19], the shock generating device for a CubeSat can be an electrodynamic shaker table [13] with a half-sine profile [13]. The shock test has similar limitations as the random vibration test, since it also uses a shaker table to affect the satellite.

Shock are simulated to create a shock response spectrum (SRS), which plots the maximum acceleration per frequency bin. The SRS contains an octave slope which rises to the first resonant frequency called the "knee frequency". The octave slope can be approximately  $9 \text{ dB oct}^{-1}$  to  $12 \text{ dB oct}^{-1}$  depending on distance to the source.

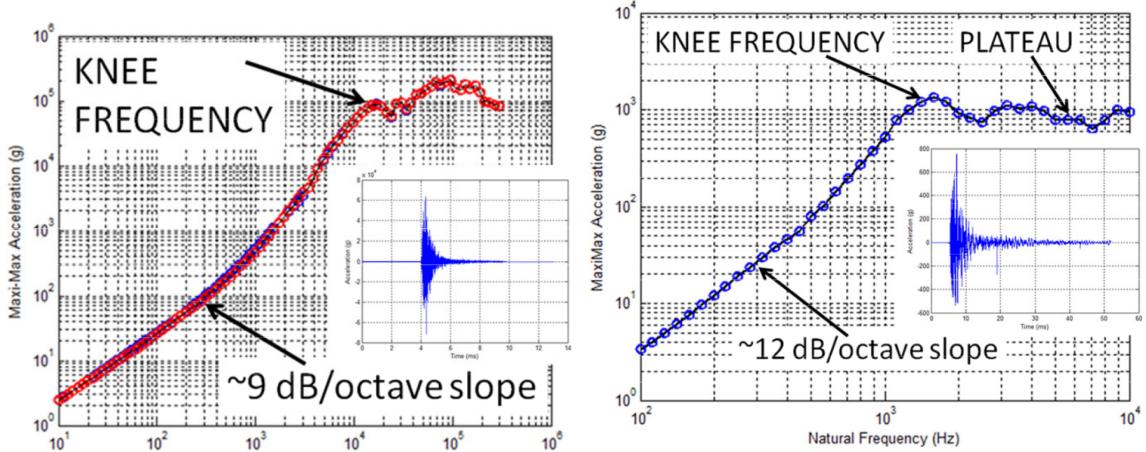


Figure 2.4: Shock response spectrum of and time-domain shock response. Left: near-field (close to shock source). Right: far-field (distant from shock source) [19]

### 2.1.3 Functional testing

The CubeSat's functionality is typically tested using "day-in-the-life" testing where a 24-hour period of the CubeSat's on-orbit life is tested on the ground [20]. Simulating the on-orbit conditions is an important factor in this test, for example GPS and star field simulators are commonly used [20].

Doyle et al. propose an extension to day-in-the-life testing called mission testing which involves simulating the long term CubeSat mission on the ground instead of one 24-hour cycle [21]. This requires accurately simulating the on-orbit conditions of the CubeSat for an extended and uninterrupted period of time (several weeks), and must also simulate nominal and emergency events that may be encountered in orbit [21].

On-orbit conditions that may be simulated include interacting with the CubeSat only through its radio and providing the same amount of power that would be seen in orbit.

## 2.2 Rocket Testing of CubeSats

### 2.2.1 Sounding Rockets

Sounding rockets are a class of suborbital rocket used between 40 km and 200 km, above where weather balloons operate [22]. While sounding rockets have been used to launch many CubeSats with suborbital trajectories, such as in the REXUS-25 mission [23], there has been only one instance of sounding rockets being used as an additional qualification platform for a CubeSat [4]. The FloripaSat-I CubeSat was tested on a VSB-30 sounding rocket [4] to qualify the CubeSat under launch conditions. This qualification method was intended not to replace, but to complement standard vibration and shock qualification methods [4]. The test measured these launch conditions through the micro-electromechanical systems (MEMS) based MPU6050 inertial measurement unit (IMU) [4].

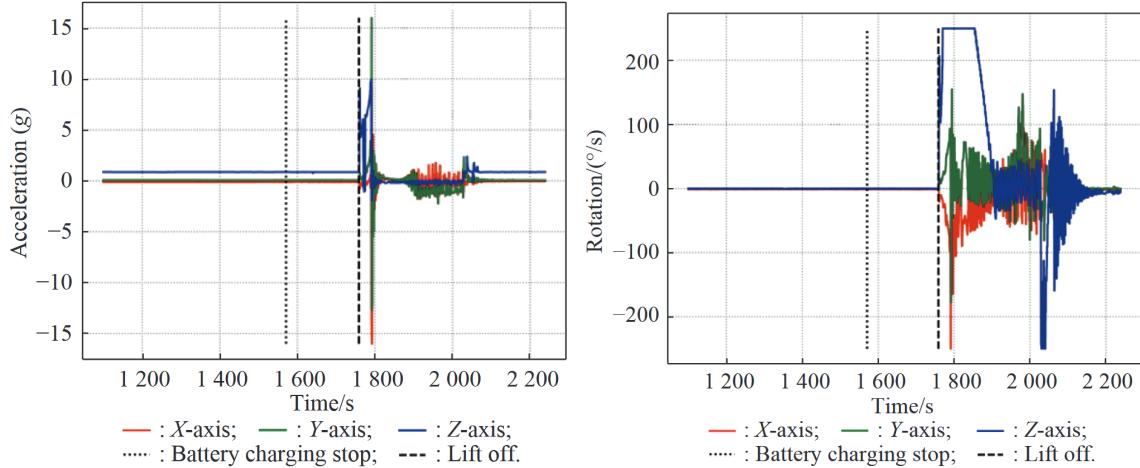


Figure 2.5: Acceleration in time domain (Left), Angular velocity in time domain (Right) during the launch of FloripaSat-I [8]

While this study does show the time-domain accelerometer and gyroscope measurements from the sounding rocket launch in figure 2.5, it does not compare the data to other qualification tests in the FloripaSat-I campaign, including random vibration and shock testing. Additionally, the launch data was not presented in the frequency domain through the boost and coast phases of the flight, meaning they could not be compared to the acceleration spectra which was shown for the shaker table testing in figure 2.6.

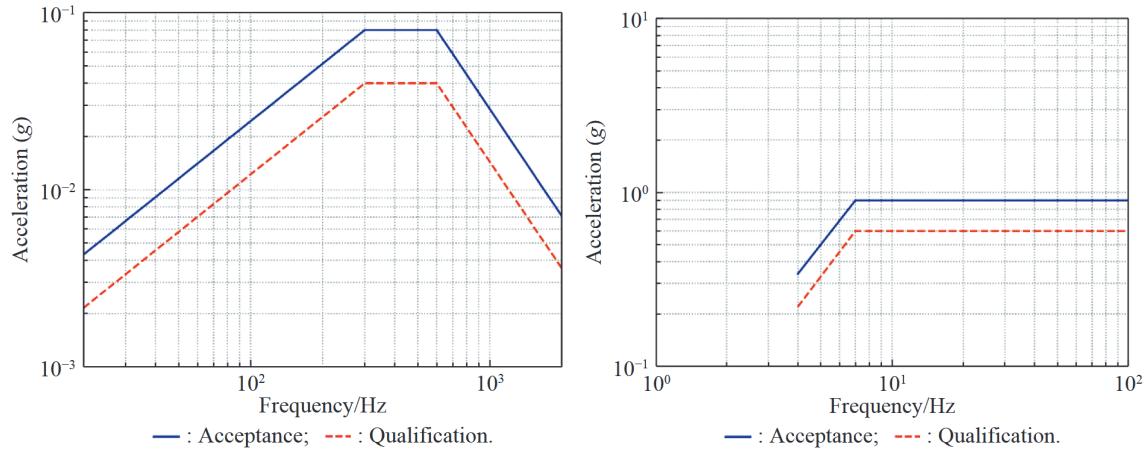


Figure 2.6: Random vibration (Left) and sine sweep (Right) tests on a shaker table during the qualification of FloripaSat-I [8]

Another shortcoming of the study is that a shock test using a half-sine pulse was not performed. The use of a sounding rocket is a potential method of qualifying the CubeSat’s ability to tolerate shocks since there will be shock events when pyrotechnics are lit to stage the rocket, although the forces will have lower intensity than on a orbital launch vehicle.

### 2.2.2 High-Power Rockets (HPR)

While sounding rockets have a significantly lower cost compared to an orbital launch vehicle, they cost \$1 million USD per launch to launch 200 kg on average [24], resulting in a specific cost of \$5000 USD/kg, which is still a large amount for university CubeSat programs. High-power rockets (HPR) are a lower-performance but cheaper alternative to sounding rockets, which can leverage the expertise of university rocketry teams while having qualification potential as sounding rockets. A single stage level 3 certification rocket can reach altitudes above 10 000 ft [25] for a cost of only \$1200 USD [25]. Despite the potential cost benefits, there have not been any published instances of a HPR being used to qualify a CubeSat.

The typical phases of a HPR launch are [25]:

- Boost phase: The HPR is being powered by a solid rocket motor. In most HPR launches, this phase only lasts several seconds.
- Coast phase: After motor burnout, the rocket follows a ballistic trajectory to its apogee.
- Apogee: This is the maximum altitude the rocket will reach. At apogee the drogue parachute is deployed, which limits the rocket’s descent velocity.
- Main parachute deployment: At a set altitude the main parachute is deployed, which has a higher surface area than the drogue parachute and further slows the rocket to landing velocity. A main parachute should not be deployed at apogee since this results in the rocket drifting further which complicates recovery efforts.

- Landing: The rocket lands on the ground and is recovered by the rocketry team for safing (disarming of live energetics) and transportation. While the landing occurs minutes after launch, finding the rocket is a harder task and may occur hours after landing.

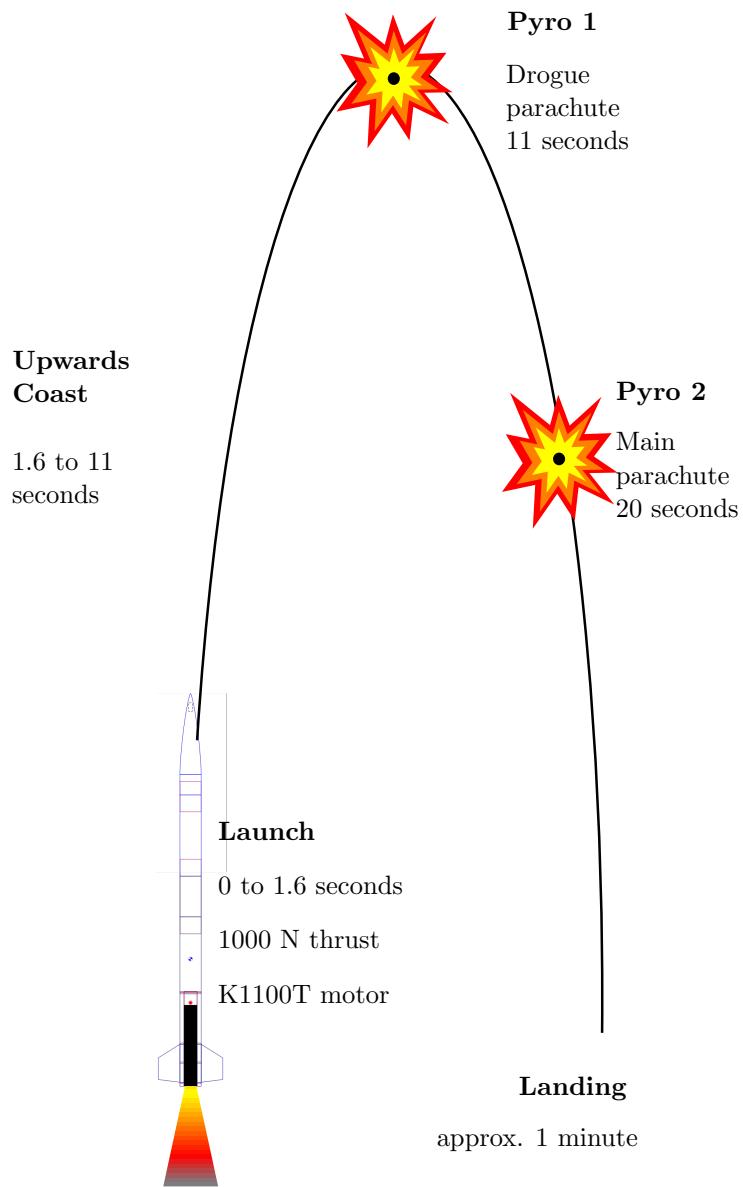


Figure 2.7: Typical launch of a single stage high-power rocket

One potential issue with HPRs as a qualification platform for shock is that low explosive black

powder is used [25] which has different explosive characteristics, such as a subsonic flame front, compared to the high-explosives used in launch vehicles [18] and will therefore produce different shock responses. One study [26] performed finite element analysis of igniters filled with low explosives including aluminium potassium perchlorate and boron potassium nitrate and determined the SRS, shown in figure 2.8. Compared to the SRS of high-explosives in figure 2.4, where at a frequency of 1 kHz the acceleration is over  $1000\text{ g}$  [19], in these low explosive simulations the acceleration at 1 kHz is only  $10\text{ g}$  [26]. Therefore, it is hypothesised that HPRs will not be useful for shock qualification since the response of low explosives is different from the high explosives used on launch vehicles.

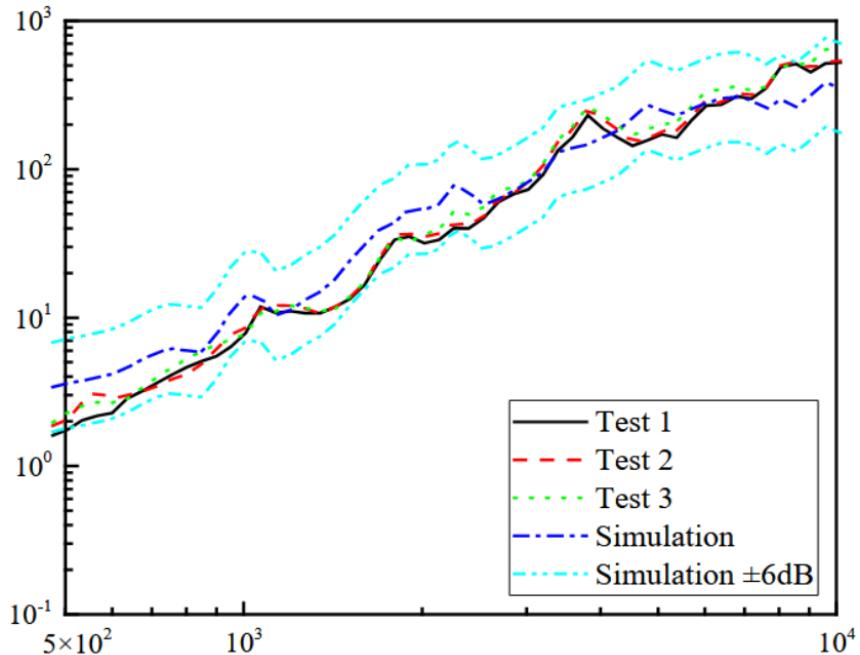


Figure 2.8: Shock response spectrum from computer modelling of an igniter based on the low explosive aluminium potassium perchlorate [26]

### 2.2.3 Avionics Systems

Some HPR flights use avionics systems which may contain [25], [27]:

- Radio and GNSS receiver for tracking,
- Accelerometer and barometer for measuring acceleration and altitude,
- Relays or transistors for firing e-matches to deploy parachutes (known as an electronic deployment).

An example of a mature avionics system widely used in HPR is the Telemetrum [27], [28], which contains a GNSS antenna and receiver, ISM radio, microcontroller, MEMS accelerometer, barometer and pyrotechnic channels [28].

Most HPR avionics use MEMS accelerometers, including the MPU6050, MPU9250, ADXL375 and BMX160 [27]. One of the most popular are the MPU6050 [27], [29], however this accelerometer is out of production and should not be used in future designs.

# Chapter 3

## Design Process

This section outlines in detail the process, shown in figure 3.1, used to create two designs: an initial design and a final design.

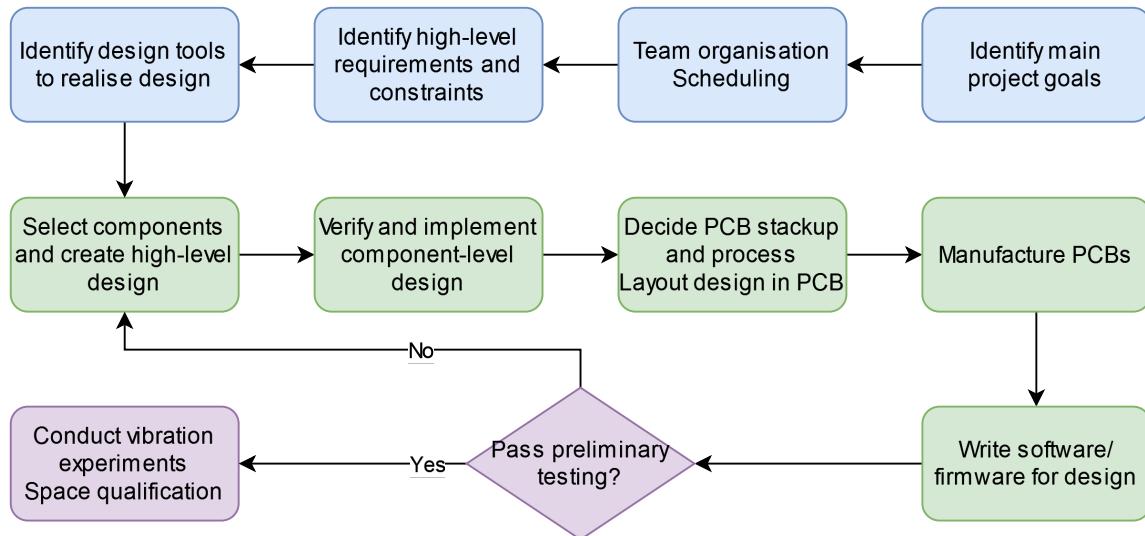


Figure 3.1: High-level overview of design process.

### 3.1 Identification of Project Goals

The ultimate goal of the CubeSat project is to launch on the POEM and receive at least one image from orbit during its 6-month lifespan.

Before launch the CubeSat must be tested to minimise the risk of mission failure. A novel HPR testing method was planned for two launches (one private and one at the Australian Universities Rocket Competition (AURC) 2024), in addition to shaker table qualification. To pass testing, the camera payload should capture at least one image on a preliminary drone test flight, the HPR

launch and during vibration testing.

This design project will have two main goals:

1. Design an experiment and a data acquisition system (DAQ) to evaluate the effectiveness of the HPR testing method.
2. Design and build an "emulation platform" as part of the DAQ which provides the same services to the camera payload as provided by POEM. While this testing campaign does not have resources for day-in-the-life or mission testing, the principle of replicating on-orbit conditions will be used in the tests to reduce the risk of integration failure. This emulation platform will need to emulate most of the specifications of POEM, but some systems will be designed to a minimum standard required for the tests.

## 3.2 Design Group and Scheduling

The CubeSat design group was made of Peter Tanner, Jamir Khan and Timothy Ludovico. Each person's dependencies are shown in figure 3.2. The scheduling of the project is agreed upon at this stage, since each part of the CubeSat requires integration before tests.

The dimensions of the DAQ were to be negotiated to fit it inside the CubeSat. While the camera payload should be built to POEM standards, some aspects such as the data rate, format and connector pinouts had to be negotiated.

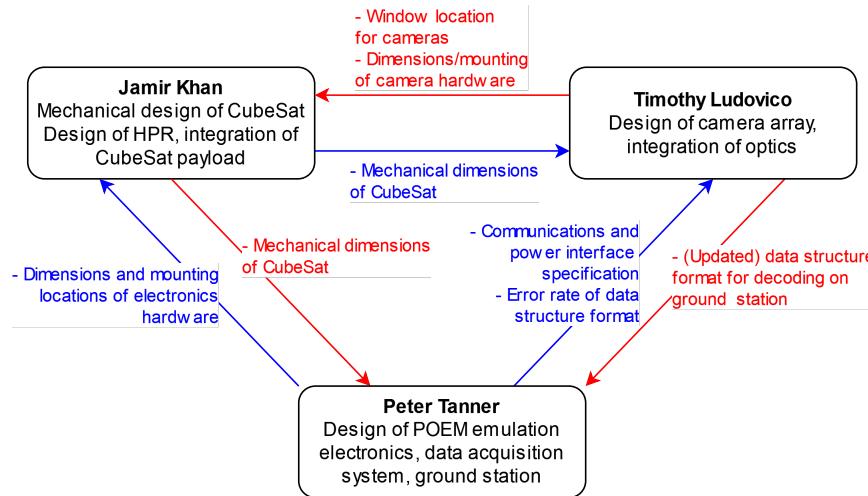


Figure 3.2: Responsibilities of members on the CubeSat design project and the information required to be communicated between each member.

## 3.3 High-Level Constraints and Requirements

After defining the goals for the project, a list of high-level requirements and constraints were created. A summary is shown in table 3.1.

ID	Name	Description
<b>E1</b>	Emulation power and voltage	EPS must emulate a 5 V bus with at least 3 A current capacity.
<b>E2</b>	Battery life	DAQ must sustain camera payload operation for at least 2 h.
<b>P1</b>	Shock and vibration	DAQ must pass shock and random vibration tests as described in section 3.7.2 Shaker Table Test.
<b>P2</b>	Hot and cold operation	DAQ must pass temperature qualification range of $-20^{\circ}\text{C}$ to $80^{\circ}\text{C}$ .
<b>P1</b>	Physical dimensions	DAQ must fit within 1 CubeSat unit.
<b>R1</b>	GNSS tracking	Must have GNSS tracking suitable for a 10 000 ft HPR launch.
<b>R2</b>	Radio downlink	Radio link must be stable to receive one image from a 10 000 ft HPR launch.
<b>V1</b>	Testing controlled environment	A direct comparison of the two methods requires the construction of the satellite to be controlled.
<b>V2</b>	Vibration Sampling Rate	DAQ must sample vibrations at a high frequency ( $> 1 \text{ kHz}$ ).
<b>V3</b>	Maximum Acceleration	DAQ must sample vibrations with a high acceleration range ( $> 200 g$ ).
<b>A1</b>	AURC Regulations	Must comply with AURC 2024 rules to perform second launch on UWA Aerospace competition rocket.
<b>A2</b>	Budget	DAQ cost must be below A\$1500.
<b>A3</b>	Project Deadline	System and experiments to be finalised by October 18, 2024.

Table 3.1: High-level requirements

### 3.3.1 Electrical Power System (EPS)

The emulation platform will contain an electrical power system (EPS) which will provide power to the camera payload and the data acquisition system.

#### Voltage and Current

A requirement of the EPS is to emulate the voltage and current provided by the POEM to one CubeSat to facilitate testing of the camera payload's power electronics. POEM contains a 28 V and 42 V bus, however IIST states that a 5 V connection with a maximum current of 1.5 A is provided to CubeSats, with the option to expand to 3 A. The EPS will have to emulate at least one of these busses.

#### Battery Life

POEM outputs a consistent amount of power to each CubeSat while on orbit due to its solar panels and battery system, however in the testing it will not be possible to deploy a solar panel, therefore the DAQ must have adequate battery life to power the camera payload throughout the length of

the HPR test. To account for setup times, this should be a minimum of 2 h.

### **3.3.2 Environmental Requirements**

If this research is continued, it is possible a future version of this payload will fly with the camera payload on POEM to make a direct comparison between the vibration environment on POEM to the conditions on both a HPR and the shaker table tests. Therefore, the DAQ must go through the same qualification campaign as the camera payload.

#### **Shock and Random Vibration Test Pass**

The DAQ must remain functional during the vibration environment of the rocket. This means it must pass the IIST recommended qualification procedure, which involves shock and random vibration tests. These tests are described in more detail in section 3.7.2 Shaker Table Test.

#### **Cold and Hot Temperature Test Pass**

The DAQ must be able to survive at temperatures of  $-20^{\circ}\text{C}$  to  $80^{\circ}\text{C}$  as described in section 3.7.1 High-Temperature Test and section 3.7.1 Low-Temperature Test. This will restrict the components able to be used to only those with industrial temperature ranges.

#### **Physical Dimensions**

The DAQ must fit inside a 1U CubeSat.

### **3.3.3 Camera Payload HPR Test Requirements**

DAQ needs to recover one image from the camera in flight. These systems do not need to exactly emulate the systems on POEM since these tests are not going to orbit, but they must function to an altitude of at least 10 000 ft, which was the intended height of the HPR.

#### **GNSS Tracking**

A launch apogee of 10 000 ft will result in the rocket drifting away from the launch site, potentially out of sight so a GNSS tracking solution is required.

#### **Radio Link**

A radio link needs to be designed to allow reception of at least one image during flight. The link needs to be designed so that data loss still results in a recognisable image.

### **3.3.4 HPR Vibration Experiment Requirements**

#### **Experiment Design**

To make a direct comparison between the two methods, the physical construction of the CubeSat and accelerometer placement should be controlled.

### **Maximum Measurable Acceleration**

Pyroshock events and motor launch are high- $g$  events that require accelerometers with measurement scales above these events, otherwise they will saturate at the maximum scale. As shown in the literature review, MEMS accelerometers struggle to measure high acceleration ranges however characterising the knee frequency would be sufficient for the experiment, therefore an accelerometer with a range over 200  $g$  would be appropriate.

### **3.3.5 Other Requirements**

#### **2024 Australian Universities Rocket Competition (AURC) Regulations**

This payload was intended to fly on the UWA Aerospace rocket *Svengeance* in the AURC 2024 competition, as part of a collaboration with UWA Aerospace. AURC has additional rules for electronics systems, relevant rules include (but not limited to) [30]:

- Lithium-polymer batteries are not allowed (unless using COTS equipment)
- Connectors must have a positive locking mechanism
- Electronics must be mounted using rigid fixing methods

#### **Budget**

The cost of the DAQ system must not exceed A\$1500.

#### **Time**

The project must be completed before the end of semester 2 2024 (October 18, 2024)

## **3.4 Selection of Components and High-Level System Design**

### **3.4.1 High-Level Design**

The high-level requirements and constraints identified in section 3.1 Identification of Project Goals were used to create a high-level design, since some subsystems depend on how others are configured. For example, using batteries in parallel requires additional balancing components to be selected in this stage.

Only the minimum amount of detail was considered at this stage in the design for flexibility. For example, the types of capacitors to be used for power supply decoupling were not part of the high-level design since they can be treated on a subsystem level later in the design process.

A high-level block diagram of the system using the parts chosen is shown in figure 3.3.

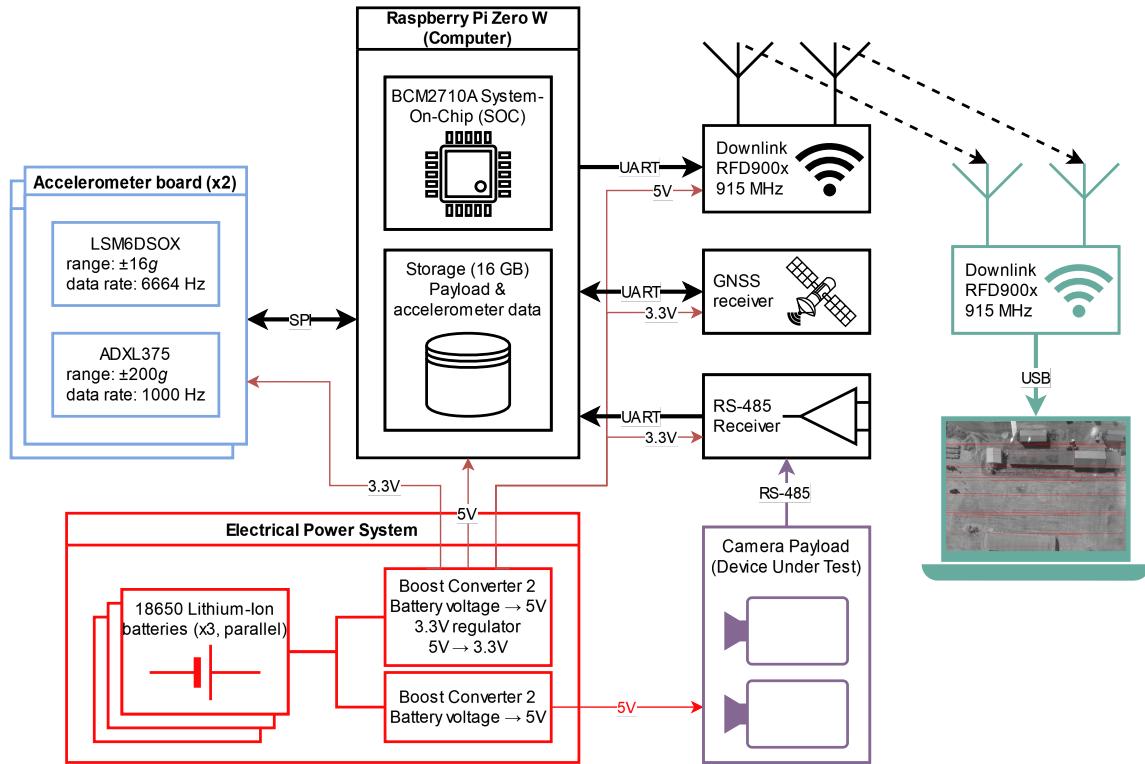


Figure 3.3: Block diagram of the CubeSat, including connections to the camera payload and ground station over radio downlink.

Lower level requirements and constraints were created for each class of component or subsystem, since the high-level requirements are quite general.

### 3.4.2 Components Sourcing

Components were sourced from at least one of the following reputable suppliers:

- JLCPCB
- Mouser
- DigiKey

JLCPCB was a PCB manufacturing service used to manufacture the PCBs for the design. In addition to manufacturing they provided a service to automatically place and solder components to the PCBs. This component source has advantages over component distributors as they:

- automatically place components, which reduces assembly time and reduces risk of poor soldering compared to manual assembly,

- price components at a lower cost that is closer to the wholesale price compared to component distributors

JLCPCB was not used for expensive components that needed fine control over the amount being purchased, since JLCPCB could only assemble either two or five full boards. This was the case for the chosen GNSS receiver, which had a high unit price and should not be assembled on all five boards.

Mouser and DigiKey were component distributors which allow the purchasing of individual components, with a higher markup compared to JLCPCB. If a part cannot be purchased from JLCPCB for any reason, it was purchased from Mouser or DigiKey and manually assembled.

### 3.4.3 High-Level EPS and Power Components Selection

A power budget was created which accounts for the worst-case current consumption of the payload under test and DAQ components, this was used to specify the EPS performance. An example of a power budget for the final design is shown in table 3.2.

Item	Voltage (V)	Max unit current (mA)	Quantity	Max current (mA)
Payload-under-test	5	3000	1	3000
Pi Zero	5	1300 [31]	1	1200
RFD900x	5	1000 [32]	1	1000
NEO-M9N	3.3	100 + 50 [33]	1	150
LSM6DSOX	3.3	0.55 [34]	2	1.1
ADXL375	3.3	0.145 [35]	2	2.9
SP3485EN	3.3	2 [36]	1	2
XR20M1172	3.3	0.5 [37]	2	1
<b>Total (3.3 V)</b>	-	-	-	157
<b>Total (5 V)</b>	-	-	-	5200

Table 3.2: Operating voltage and current consumption of devices connected to EPS in the final DAQ design.

### Battery Selection

Batteries were selected based on the following criteria shown in table 3.3:

Name	Description	HL constraints
<b>Stable in vacuum</b>	The DAQ will be vacuum tested, therefore the cells need to be stable under vacuum	P1
<b>Wide temperature range</b>	Batteries must survive temperature testing	
<b>Low cost</b>	Due to limited budget, space qualified batteries would be cost prohibitive	
<b>High energy density and specific energy</b>	The system needs to support the payload for at least 2 h at the worst-case power budget without recharging	
<b>Reasonable power density and specific power</b>	The overall system needs to be able to provide at least 3 A of current in a small space/mass, so the batteries should have similar current capacities.	
<b>Must not be a lithium-polymer battery</b>	This is an AURC rule.	
<b>Space heritage</b>	Ideally the batteries would be used in space to maximise the chance of passing vacuum and temperature testing.	

Table 3.3: Battery cell requirements

The battery chemistry was selected to design other parts of the high-level design such as the charging and power conditioning electronics, since these were dependent on the nominal cell voltage. Lithium-ion (Li-ion) cells were selected since in general they satisfy many of the criteria.

### 3.4.4 Power Electronics

Power electronics were used to condition the battery voltage, since the voltage of batteries decreases as it was discharged.

For the high-level design, two cell configurations were considered, as shown in figure 3.4:

- 1S (one set of  $2N$  batteries in parallel).
- 2S (two sets of  $N$  batteries in parallel, each set connected in series).

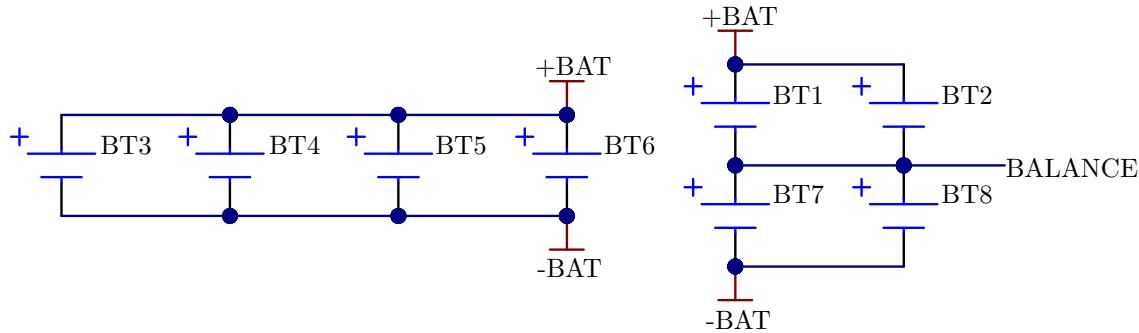


Figure 3.4: (Left) 1P battery pack using  $2N = 4$  batteries, (Right) 2P battery pack using  $2N = 4$  batteries.

Category	1S	2S
Nominal voltage (Li-ion)	3.6 V	7.2 V
Power loss	Higher $I^2R$ losses	Lower $I^2R$ losses
Components required for 5 V output	Boost converter	Buck converter and balancing circuitry
Cost of system	Lower due to lack of balancing	Higher due to balancing circuitry

Table 3.4: Comparison of 1S and 2S battery packs.

A 2S battery pack would make sense if the electronics operate on a higher voltage (for example, if there are motors or high-power radios). However, in this application the highest voltage used was 5 V which was not significantly higher than 3.7 V. Additionally, the  $I^2R$  losses between the battery and the power converters is low since they were integrated on the same board. A preliminary analysis shows that a power system using 2S batteries resulted in a higher cost compared to a 1S system due to the additional battery balancing electronics required. Therefore, a 1S battery pack was selected for the high-level design.

Based on the high-level description of the battery pack and the power budget, a boost converter and regulator would be used to condition the battery voltage. A regulator is used instead of a buck-boost converter due to the low current consumption of the 3.3 V rail and low cost compared to a buck-boost converter. These components will have the following requirements:

## Boost Converter

Name	Description	HL constraints
<b>Input voltage range</b>	It must have an input voltage range of at least $< 2\text{ V}$ and $> 4.4\text{ V}$ to support a Li-ionbattery.	TODO:
<b>Output voltage range</b>	It must be able to output 5 V.	TODO:
<b>Output current range</b>	It must be able to output at least 3 A.	TODO:
<b>High efficiency</b>	A lower efficiency converter would require more batteries and therefore more space and mass to meet the battery life requirement compared to a higher efficiency converter.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.5: Boost converter requirements

## Linear Regulator

Name	Description	HL constraints
<b>Dropout voltage</b>	This linear regulator will receive a constant 5 V, it must have a dropout voltage lower than 1.7 V so that it can power a 3.3 V system.	TODO:
<b>Output voltage range</b>	It must be able to output 3.3 V.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.6: Linear regulator requirements

### 3.4.5 Radio Downlink

The POEM contains a radio downlink which allows experiments to transmit data to the ground at a speed of  $5\text{ kbit s}^{-1}$ . The radio cannot be used to control the CubeSats from the ground. Since the HPR test is relatively short, the downlink speed was increased to allow one image to be received over the time span of one flight.

Name	Description	HL constraints
<b>Data rate</b>	Must have adequate data rate to transmit one image over the time of a 10 000 ft HPR test flight	TODO:
<b>Range</b>	Must have adequate link budget to allow reception of picture data over 10 000 ft.	TODO:
<b>Common interfaces</b>	Must use a non-proprietary interface, such as UART or SPI to simplify development.	TODO:
<b>USB ground station</b>	Must be able to be connected to a laptop on the ground for live visualisation of picture data.	TODO:
<b>Class license operation</b>	Radio must be able to be legally operated using a free class license [38], since obtaining an amateur radio license would require significant time.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.7: Radio downlink requirements

Note that this radio transceiver will not be used on the space launch, since POEM will provide a radio downlink, but it must pass environmental testing since wireless communication is the only method of monitoring the system during environmental testing.

### 3.4.6 Payload Communications

For the payload communications, a physical layer and data layer was chosen. The physical layer determined the representation of data as (in this case) electric signals. The data layer determined how data was transmitted and received as frames.

CubeSats use the RS-485 physical layer specification and UART data layer specification to transmit data to POEM at a maximum speed of  $5 \text{ kbit s}^{-1}$ . Therefore, this system has no choice but to emulate the physical and data layer specification used on POEM to satisfy design goals.

Name	Description	HL constraints
Same data layer as POEM (UART)	This needs to emulate the same interfaces on POEM.	TODO:
Same physical layer as POEM (RS-485)	This needs to emulate the same interfaces on POEM.	TODO:
Locking connectors	Locking connectors must be used between the payload and DAQ due to the high vibration environment.	TODO:
Data rate of $5 \text{ kbit s}^{-1}$ or greater	This needs to emulate the same interfaces on POEM.	TODO:
Industrial temperature range	The RS-485 transceiver is a component that must pass temperature testing.	TODO:

Table 3.8: Payload communications requirements

RS-485 was chosen since it is a differential bus which increases immunity to electromagnetic interference from other elements on the launch vehicle, therefore reducing the bit error rate [39]. A combination of RS-485 and UART allows large amounts of data to be transferred since the UART data link specification is simple compared to other protocols such as CAN bus [39].

Since the DAQ needed to emulate POEM services as a constraint, payload communications had to use the combination of RS-485 and UART. The Pi Zero has one hardware UART, this single-ended UART signal needs to be converted to the differential RS-485 signal using an RS-485 transceiver. The SP3485 transceiver was chosen due its low unit cost of A\$0.32 and high availability on JLCPBCB. The transceiver supports data rates up to  $10 \text{ Mbit s}^{-1}$ , which was adequate for the  $5 \text{ kbit s}^{-1}$  downlink speed. It used a 3.3 V logic level on the single-ended side, which was required to interface with the Pi Zero without level shifting [36]. The EN-L variant was chosen due to its extended temperature range of  $-40^\circ\text{C}$  to  $85^\circ\text{C}$  required to pass environmental testing [36].

### 3.4.7 GNSS Tracking

The DAQ had to track the HPR throughout the full launch to enable recovery as stated in section 3.3.3 Camera Payload HPR Test Requirements. This will be achieved through a Global Navigation Satellite System (GNSS) receiver, which received signals from GNSS satellites and determined the position and altitude of the receiver.

Name	Description	HL constraints
<b>Metre-level precision</b>	Metre-level precision is adequate for tracking and recovery.	TODO:
<b>Common interfaces</b>	Must use a non-proprietary interface, such as UART or SPI to simplify development.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.9: GNSS tracking requirements

### 3.4.8 Accelerometers

MEMS accelerometers were chosen for the DAQ due to their low cost and low power consumption compared to traditional piezoelectric accelerometers.

Name	Description	HL constraints
<b>High output data rate</b>	Output data rate should be over 1 kHz to maximise range of the shock response spectrum.	TODO:
<b>High acceleration range</b>	Maximum measurable acceleration should be over $200\ g$ to maximise range of the shock response spectrum.	TODO:
<b>SPI with FIFO and hardware interrupt</b>	To reduce load on the processor SPI should be used over other interfaces (such as I <sup>2</sup> C), and a FIFO queue with interrupt should be available.	TODO:
<b>PCB with adequate mounting points</b>	The accelerometers need to be mounted to a PCB with a strong mounting point to maximise the resonant frequency.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.10: Accelerometer requirements

### 3.4.9 Onboard Data Handling Unit (OBDH)

The OBDH unit acquired data from sensors and the payload and saved it to a storage device and relayed relevant data to the ground station through the radio. The two main requirements of the OBDH is that it has enough resources to be able to process and save the sensor and payload data without losing data, and that it has adequate storage to hold sensor data from one HPR flight.

Name	Description	HL constraints
<b>Data storage</b>	OBDH should be able to store several images for post-flight analysis if the radio downlink underperforms. It should be able to store readings read from the accelerometer at a high rate.	TODO:
<b>SPI and UART interfaces</b>	These interfaces are requirements for the GNSS receiver, payload under test and the accelerometers.	TODO:
<b>Adequate processor speed</b>	The OBDH must not "drop" image frames or data samples due to inadequate processing speed.	TODO:
<b>Industrial temperature range</b>	This is a component that must pass temperature testing.	TODO:

Table 3.11: OBDH requirements

## 3.5 Design Tools

These tools were used to realise the design so that it can be manufactured.

### 3.5.1 Altium Designer 24

Altium Designer 24 (Altium) is an electronics design automation (EDA) tool which is widely used in industry and has been used for design of CubeSat and space hardware [40].

#### Schematic Editor

A circuit is first implemented using schematic symbol representations of components in the schematic editor. In the schematic view the connections between the components are abstracted using net labels and wires. The schematic view does not necessarily represent the physical layout of the PCB but is intended to convey the connections between components in a format that can easily be read.

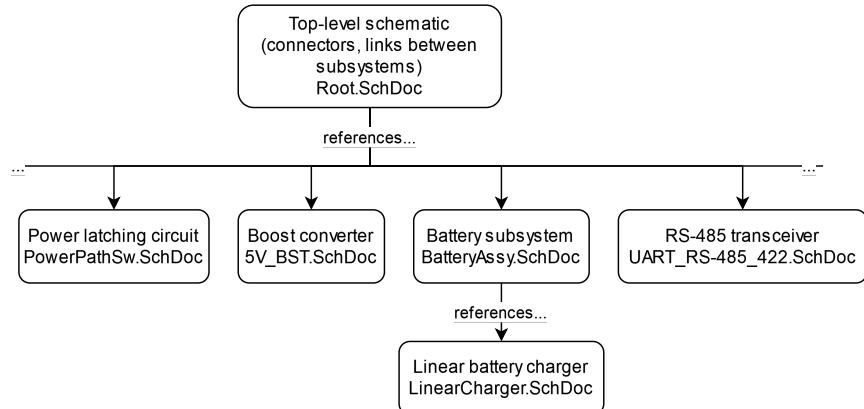


Figure 3.5: Example of the hierarchical schematic sheet format for the main DAQ PCB.

A root schematic contains references to other schematics which are abstracted as sheet symbols with ports. Each sheet symbol represents a particular subsystem of the DAQ. The hierarchical sheet symbol representation has several benefits, including that it facilitates reuse of designs and allows larger systems to be decomposed into multiple schematics which are easier to modify and read. This is shown in figure 3.5.

## PCB Editor

After a design is implemented using schematics, its physical layout defined in the PCB editor. A full "net list" of connections between each symbol in the schematic representation is transferred to the PCB editor. Each symbol is linked to a footprint, which is the physical representation of the component and contains information such as

- The land pattern, which is the layout of pads or holes required for mounting the component on the PCB,
- The component's 3d model

The PCB editor contains automated design rule check (DRC) tools which is used in the design process to reduce the likelihood of a faulty PCB. The DRC uses rules set in the project and if a rule is violated, it is reported. This feature is used to ensure components have adequate clearance, geometry such as vias and tracks have manufacturable dimensions, and to control the impedance of microwave frequency tracks. From a design process standpoint, this reduces the risk of receiving PCBs with design issues and increases the likelihood of meeting project budget constraints.

## Output Jobs

Once a PCB is ready to be manufactured, an automated "outjob" ensures that the required design files are automatically generated with the right settings for manufacturing. From a design process standpoint, this reduces the risk of receiving boards with incorrect specifications due to incorrect export settings. The artefacts generated include:

- Bill of materials

- Gerber files
- Drill location files
- Pick-and-place component locations

### **PCB Project and Version Control System (VCS)**

The schematics, PCBs and output jobs were integrated into an Altium PCB project, which was versioned using the Git version control system (VCS) and pushed to a remote Git server on every commit.

#### **3.5.2 circuit.js**

Circuit.js was a simple browser-based analog circuit simulator [41]. Circuits in this simulator can be edited and interacted with in real-time, whereas in traditional SPICE simulators the circuit cannot be edited once the simulation starts. Circuit.js uses a numerical method which is prone to error however, therefore this simulator was used for rapid, real-time prototyping of designs. After these designs were finalised they were simulated in traditional SPICE-based simulators.

#### **3.5.3 LTspice**

LTspice was used to the DC-DC boost converter for this project, which was required to power the internal DAQs and the payload. A simulation was performed to characterise the ripple voltage and to validate its performance over a range of input voltages. LTspice has been used for simulation of boost converters in the past and is free which makes it a suitable circuit simulator for this project [42].

Ultimately LTspice was chosen over other freeware SPICE simulators such as PSpice since LTspice contains an "alternate" solver which has less error at the trade-off of simulation time [43]. The reduced error results in the solver converging on a solution, whereas in PSpice or in LTspice normal solver mode it was not able to converge on a solution and the simulation could not be completed.

## **3.6 Implementation of High-Level Design**

After the high-level design was created and components finalised, they were integrated into the design as shown in figure 3.6 using the design tools from section 3.5 Design Tools

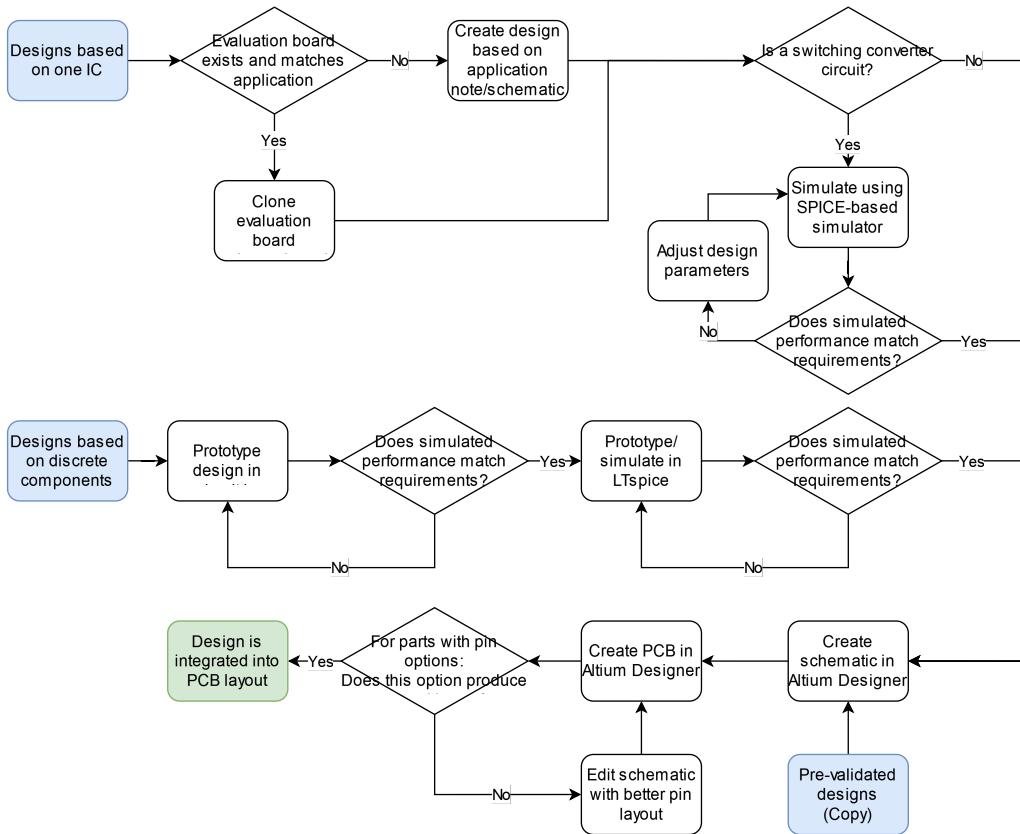


Figure 3.6: Workflow for integrating a design into a PCB.

### 3.6.1 Design Verification and Schematic Capture

The high-level design shown in 3.3 was separated into multiple subsystems, this practice was useful since it allowed for design reuse, improved readability of the project and allowed different verification strategies to be used for different subsystems.

There were three cases for integration of one subsystem into the design:

1. Subsystems based around single ICs, for example power converters, were implemented by using the process outlined in the application note or by copying a reference design or evaluation board design if it exactly matched the application in the DAQ. As an additional precaution, switching converters are simulated in a SPICE simulator to verify the output was stable over the range of input voltages and output currents.
2. Discrete components were used to implement some simple circuits to save on part cost. These were prototyped in circuit.js, then once working in circuit.js were simulated in a SPICE simulator to verify their functionality.
3. Some subsystems have already been verified on other projects and are able to be directly applied to the DAQ, the schematics for these designs are copied.

Once verified, a subsystem was implemented as a single schematic sheet in an Altium PCB project and connected to other schematics using the hierarchical sheet system as described in section 3.5.1 Schematic Editor.

### 3.6.2 PCB Setup

Before any subsystems are laid out in the PCB, several parameters about the PCB are agreed upon.

PCB rules are set according to the JLCPCB capabilities page. This was set according to the board manufacturing process used and ensures that the board does not contain geometry which cannot be manufactured or assembled.

The dimensions of the PCB and position of connectors were negotiated with the rest of the design team to ensure it would fit in the CubeSat. The maximum size was negotiated so that there was as much room possible for simpler routing.

Each PCB had a stackup defined, which set the number of metal layers in the PCB, the thickness of the board, the insulating substrate used and the type of routing on each layer. For example, a standard PCB stackup would have four copper layers (signal-ground-power-signal) with a 7628 substrate. Parameters which determined what stackup to use included cost (since large or less common stackups have additional fees), the number of nets and components required to route and the RF performance of the substrate.

### 3.6.3 PCB Layout

Once a subsystem was finished in schematic capture, the components were routed in the PCB using general PCB design rules including:

- Using the manufacturer's sample layout and PCB layout guidelines, if they exist
- Placing the majority of components on one side to allow cheaper PCB component assembly, and placing large or manually soldered components such as connectors on the opposite side
- Minimising track distance
- Using large polygons for power routing
- Placing decoupling capacitors close to the part's power pins

Additional PCB rules are created where necessary, such as to enforce the geometry of RF tracks or clearance rules.

### 3.6.4 Finalisation of Design and Manufacturing

After the PCB design was finished, an automatic design rule check (DRC) was run to find any errors that would affect manufacturing, based on the PCB rules set in section 3.6.3 PCB Layout. The manufacturing outjob was run to generate artefacts such as Gerber files, the bill of materials for automated and manual manufacturing, and component locations for pick-and-place. These artefacts were sent to JLCPCB to design and partially assemble the PCBs. The manual manufacturing BOM was used to purchase components from component distributors.

After the boards were received some basic tests are conducted (such as ensuring that voltage domains and ground are not short-circuited). After this, additional components were manually

assembled either using hot-air or a soldering iron. After soldering, the manual solder joints were inspected to ensure they are not cold joints, and the boards were again tested for short-circuits.

### 3.6.5 Software Design Process

The Raspbian OS was flashed on to an SD card with settings to allow the Pi Zero to connect to a local Wi-Fi network. An external computer was used to connect to the Pi's secure shell (`ssh`) server and log into the Pi. After this, the USB Ethernet gadget was configured on the Pi and host computer sides which allows the host computer to `ssh` into Pi without needing a Wi-Fi network, which was useful for field debugging.

A combination of Python scripts and C programs were used for different parts of the DAQ. Python is used for tasks which do not require many system calls and do not require high optimisation, such as transferring data from the payload to the radio. The advantage of Python for these tasks was speed of development. Some tasks, such as reading data from the accelerometers, require many system calls and has considerable performance impact when using Python. For these tasks, a program was written in C and compiled on the Pi Zero. C has less overhead compared to Python as it is a compiled directly to ARM assembly, whereas Python is interpreted to an intermediate representation which is executed by the Python virtual machine which adds overhead.

A Samba server was set up on the Pi Zero to allow code developed from a PC to be copied over to the Pi, and to allow an integrated development environment on the PC to access libraries and headers present on the Pi Zero.

### 3.6.6 Integration Testing

To ensure the DAQ performs at a basic level with the camera payload before conducting any large tests, the systems were integrated, and the following pass/fail tests were performed.

#### 1. Integration testing with camera system

Success criteria: one image from the payload needs to be re-transmitted over radio and saved on the OBDH.

#### 2. Ground distance testing

Success criteria: one image from the payload needs to be re-transmitted over radio and saved on the OBDH over a distance equal to or greater than the first drone test (120 m).

This stage was where a decision needs to be made on whether the design was final or required another iteration.

## 3.7 Design Evaluation Framework

The design evaluation framework consisted of three major types of tests:

- Environmental tests.

Hot and cold temperature testing, which qualifies requirement P2 (hot and cold operation).

Shaker table, which qualifies requirement P1 (shock and vibration), V1 (testing controlled environment), V2 (vibration sampling rate) and V3 (maximum acceleration).

- Vehicle tests

Drone flight (120 m), which qualifies requirements R2 (radio downlink), R1 (GNSS tracking).

HPR flight, which qualifies requirements P1 (shock and vibration).

- Experiment evaluation.

Evaluation of accelerometer performance, which qualifies V1 (testing controlled environment), V2 (vibration sampling rate) and V3 (maximum acceleration).

Each test was assessed on a pass/fail basis. Note that this evaluation framework is not based on whether the HPR flight is a viable alternative to shaker table testing, but instead aimed to evaluate the effectiveness of the experiment itself and the DAQ.

### 3.7.1 Environmental Testing

As stated in 3.3.2, it is possible the DAQ will fly with the CubeSat on the PSLV in future work, therefore there are environmental requirements for part of the DAQ. These environmental tests would evaluate the resilience of the DAQ in the environment of space and during launch.

#### High-Temperature Test

IIST recommended a qualification test where the CubeSat was placed in a thermal vacuum chamber for 2.5 h and heated to 70 °C. The CubeSat electronics were turned on and tested during the final 30 min of the test.

Due to time restrictions it was only possible to do a preliminary high-temperature test at atmospheric pressure with a consumer oven on only the electronics section of the payload (the combined camera and DAQ assembly) without the CubeSat chassis. The preliminary test used the same heating temperature and timeline as the recommended test.



Figure 3.7: High-temperature testing setup

A failure of this test occurs when the DAQ loses radio connection or transmits corrupted data for more than one minute, or if the system is not functional after the test.

### **Low-Temperature Test**

IIST recommended a qualification test where the CubeSat was placed into a thermal vacuum chamber for 2.5 h and was cooled to  $-20^{\circ}\text{C}$ . The CubeSat electronics were turned on and tested during the final 30 min of the test.

Due to time restrictions it was only possible to do a preliminary low-temperature test with a consumer freezer. To prevent condensation from developing on the electronics during the test, which would not occur in the thermal vacuum chamber, the electronics were placed in an airtight bag prior to the test and pressurised with pure nitrogen gas for 5 min to displace air containing moisture. Note that this would not replicate the thermal transfer conditions in a vacuum, but would at least qualify the temperature range of the components used.



Figure 3.8: Low-temperature testing setup

A failure of this test occurs when the DAQ loses radio connection or transmits corrupted data for more than one minute, or if the system is not functional after the test.

### **3.7.2 Shaker Table Test**

IIST recommended that the CubeSat be mechanically qualified using a single-axis electrodynamic shaker table using random vibration and half-sine shock tests.

#### **Random Vibration**

In the random vibration test, the CubeSat was fixed rigidly to an electrodynamic shaker table, then the table was programmed with the IIST recommended vibration profile specified in table 3.12. The vibration test occurred for 60 s per axis and was repeated for all three axes.

Frequency (Hz)	PSD ( $g^2 \text{ Hz}^{-1}$ )	$g$ (RMS)	Duration ( $\text{s axis}^{-1}$ )	Axis
20	0.002	13.5	60	Three axes
60	0.002			
250	0.138			
1000	0.138			
2000	0.034			

Table 3.12: IIST recommended random vibration test profile for qualification of CubeSat for launch on POEM.

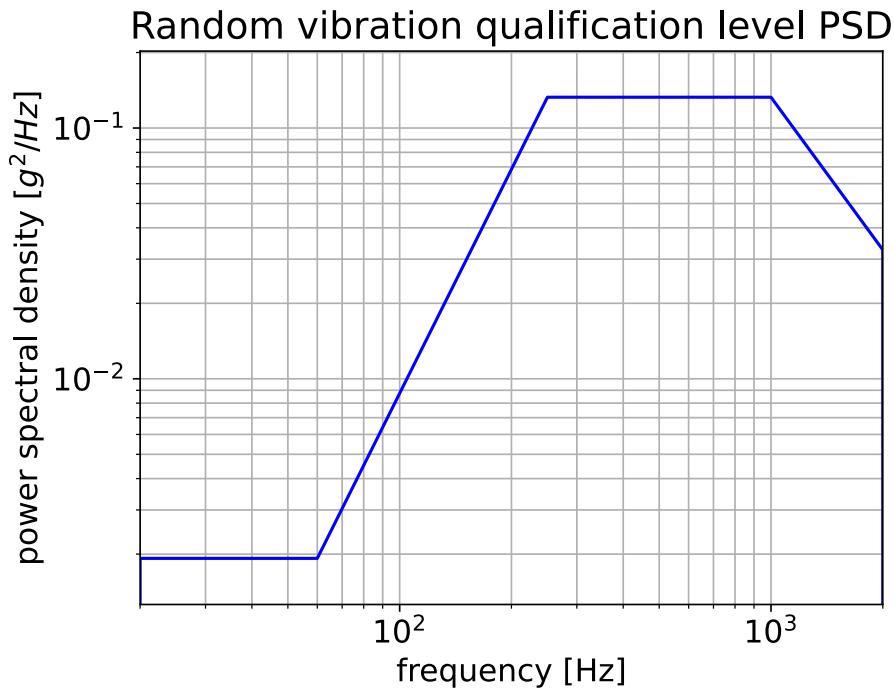


Figure 3.9: IIST recommended random vibration test profile for qualification of CubeSat for launch on POEM (profile defined in 3.12).

The IIST recommended random vibration test profile was used without modifications in the final shaker table testing.

A failure of this test occurs when the DAQ loses radio connection or transmits corrupted data for more than one minute, or if the system is not functional after the test.

### Shock

In the shock test, the CubeSat was fixed rigidly to an electrodynamic shaker table, then the table was programmed with the IIST recommended half-sine shock profile specified in table 3.12. The amplitude of the shocks had to be reduced in the test due to limitations of the shaker table.

The IIST recommended qualification level for the shock test is specified in table 3.13.

Amplitude	Duration (ms)	Shock profile	Axis
50 g	10	Half sine	Single-axis shocks, for all three axes

Table 3.13: IIST recommended shock test profile for qualification of CubeSat for launch on POEM.

A failure of this test occurs when the DAQ loses radio connection or transmits corrupted data for more than one minute, or if the system is not functional after the test.

### 3.7.3 Vehicle Tests

#### Drone Test Flights

Drone tests were used as a qualification platform for the HPR launch since drone tests:

1. Use the expertise of the UWA Aviation Laboratory, which is participating in the project.
2. Are repeatable, whereas the rocket test can only feasibly be done once per launch season.
3. Have greater control over the position compared to the suborbital rocket launch and will better qualify the machine vision processes on the camera payload.

The drone test evaluated the communications between the camera payload and the communications downlink stability in real time. A successful test involved receiving at least one frame from the camera payload at a reasonable quality.

#### High-Power Rocket Test Flight

The high-power rocket (HPR) test flight was used as an experimental qualification method for the CubeSat. This DAQ is used to evaluate the effectiveness of the HPR flight by using accelerometers, but the HPR flight also serves as a milestone for evaluating the effectiveness of this DAQ for this type of application.

### 3.7.4 Experiment Evaluation

The first condition for the experiment to be successful was if a conclusion could be reached on if HPR is a viable alternative qualification method using only the response of the HPR and shaker table tests.

Since this design used lower cost MEMS accelerometer as the vibration measurement device, this had to be validated. It will be assumed that the accelerometers on the shaker table have been calibrated adequately. By comparing the response observed by the shaker table vibration sensors to the response observed by the MEMS accelerometer, it will be possible to evaluate the effectiveness over the spectrum of vibrations the DAQ encounters during flight.

## 3.8 Version One of DAQ

The design process allows for multiple versions of the DAQ to be designed, subject to time and budget constraints.

### 3.8.1 Overview of First Design

The first design had three boards: EPS, radio and OBDH.

The OBDH was based on a STM32L476 microcontroller and the 8 Gbit THGBMJG6C1LBAIL embedded multimedia card (eMMC). All connections out of the microcontroller going to other boards were using the RS-485 physical layer. This design also was intended to be used in a dual redundant pair. A ZED-F9P differential GNSS receiver was used in this design. Other elements of the design were not changed between the first and second revision.

The microcontroller was programmed in C using the STM32 hardware abstraction library (HAL) and using FreeRTOS as a realtime operating system to manage scheduling.

### 3.8.2 Evaluation of First Design

- The STM32L476 did not have enough resources to move data from the sensors and camera payload to the payload at an adequate speed. A benchmark using CrystalDiskMark, in table 3.15 shows that the maximum throughput is  $0.84 \text{ MB s}^{-1}$ , and while only 60% of the throughput is being used as shown in table 3.14, between reading from the data sources and writing to the storage there is not enough resources in practice to do this at an adequate speed, resulting in the maximum sampling rate of the sensors to be limited.
- There were issues using the 4-bit mode on the eMMC, which limited write and read speeds.
- Due to space limitations on the rocket, it was not possible to have two redundant systems. The next revision would use only one DAQ.
- By the end of this section, it was understood that centimetre level positioning was not required to obtain good results from the camera system.
- A preliminary test with the camera payload showed that the RS-485 transceiver was able to successfully receive image data from the camera payload, but it was not able to write the data to the eMMC or the radio at an adequate rate.
- The preliminary test also showed that the EPS was adequate for delivering power to the camera payload.
- Good GNSS data was received from the GNSS board.
- The RFD900x performed well when tested alone without writing to the eMMC.
- At the end of this revision it was concluded that the STM32 platform was not flexible enough to complete the research objectives in time.

Data source	Data rate	Notes
LSM6DSOX	$0.41 \text{ MB s}^{-1}$	16 byte structs are generated at 6664 Hz for both acceleration and gyroscope data for two sensors.
ADXL375	$0.038 \text{ MB s}^{-1}$	20 byte structs generated at 1 kHz for two sensors.
Camera payload	$0.054 \text{ MB s}^{-1}$	460 800 bd
<b>TOTAL</b>	$0.502 \text{ MB s}^{-1}$	60% of maximum sequential write bandwidth.

Table 3.14: Data sources and their data rates.

Test	Read [MB/s]	Write [MB/s]
SEQ1M Q1T1 (1 task, 1 thread)	0.84	0.84
RND4K Q1T1 (1 task, 1 thread)	0.75	0.66

Table 3.15: CrystalDiskMark benchmark of DAQ v1.

### 3.8.3 First Revision Implications

The EPS, GNSS receiver and radio designs were left unchanged since they performed well during preliminary testing.

However, the OBDH was completely changed to use a Raspberry Pi Zero W in the second revision. This was because a microcontroller was hard to debug due to a limited number of breakpoints and a lack of isolation between tasks in FreeRTOS which resulted in difficult to debug issues. A Raspberry Pi has a Linux-based operating system, which is easier to develop for since tools such as GDB can be used for debugging, and processes isolate errors from affecting other processes. The Raspberry Pi also has more resources and should not result in the poor write speeds observed on the first OBDH.

Some systems were simplified, such as the differential GNSS not being used on the final DAQ. RS-485 was also removed for board to board communication, except between the camera payload and the DAQ, since it was unnecessary given the short distance between DAQ boards.

# Chapter 4

# Final Design of DAQ

The final DAQ design is implemented using the design process described in section 3 Design Process.

## 4.1 Overview of PCBs

Three PCB projects are created for the three boards in the final design:

- Power and OBDH (referred to as DAQ)
- GNSS receiver
- Accelerometers

As stated, PCB rules were set according to the JLCPCB capabilities page.

### 4.1.1 PCB Sizes

A PCB size of 80 mm × 80 mm was used for the boards since this was the maximum size available for the PCB inside the CubeSat. The DAQ board uses this size since the batteries are mounted on the reverse side of the board, and require a lot of board area to mount. The GNSS receiver board uses this size since larger PCB areas result in higher gain for the patch antennas used for receiving GNSS signals. For the MEMS accelerometers, a PCB of 22 mm × 22 mm was chosen as it was the minimum size possible to fit both accelerometers and mounting holes, and minimising the PCB area maximises the resonant frequency of the accelerometers.

### 4.1.2 PCB Stackups

Four layer stackups are used for the DAQ and GNSS boards since these were cheap to manufacture at JLCPCB for boards under 100 mm × 100 mm, and they were simpler to route due to more layers. The DAQ used a standard signal-ground-power-signal stackup. This was chosen since the DAQ board contains many parts and having a dedicated power and ground plane made routing simpler and improved the EMI performance of the PCB. The GNSS board contained few components and used a stackup of four ground layers which covered the full PCB to maximise the gain of the patch

antennas. The accelerometer board used a two-layer signal-signal stackup since there were few components to route and to save cost.

All boards used a 7628 substrate since this was the lowest cost substrate and acceptable since the highest frequencies routed on the PCBs were the L1 signal for GNSS, which is a relatively low 1575.42 MHz. As shown in figure 4.1, the loss for a microstrip is under 1 dB at 2 GHz [44], which is acceptable.

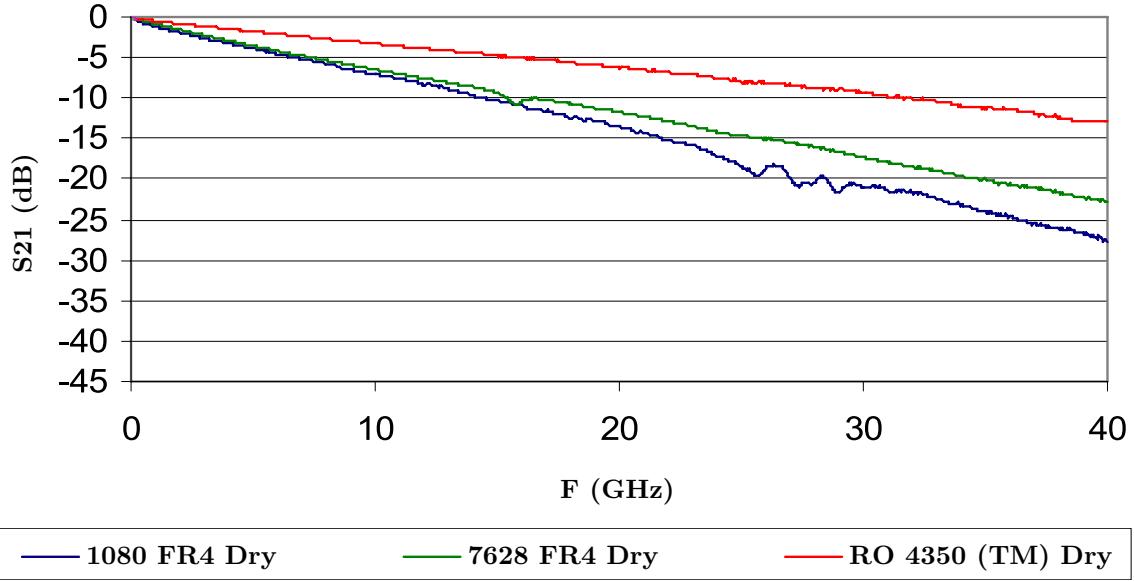


Figure 4.1: S21 attenuation parameters of several dielectrics for a  $50\Omega$  microstrip. Compared to the high-performance Rogers 4350 dielectric, there is negligible difference in attenuation at low frequencies. Adapted from [44]

All boards used hot air solder levelling (HASL) lead-tin coating due to its low cost, and since this PCB did not require flat surface finishes since relatively large components were being soldered on to the PCB, and since these boards were not commercial boards that were subject to environmental requirements.

## 4.2 Electrical Power System (EPS)

### 4.2.1 Power Budget

The power budget of the DAQ and the required power supply for the payload was created for the final design, as shown in table 4.1.

Item	Voltage (V)	Max unit current (mA)	Quantity	Max current (mA)
Payload-under-test	5	3000	1	3000
Pi Zero	5	1300 [31]	1	1200
RFD900x	5	1000 [32]	1	1000
NEO-M9N	3.3	100 + 50 [33]	1	150
LSM6DSOX	3.3	0.55 [34]	2	1.1
ADXL375	3.3	0.145 [35]	2	2.9
SP3485EN	3.3	2 [36]	1	2
XR20M1172	3.3	0.5 [37]	2	1
<b>Total (3.3 V)</b>	-	-	-	157
<b>Total (5 V)</b>	-	-	-	5200

Table 4.1: Operating voltage and current consumption of devices connected to EPS.

#### 4.2.2 Battery Selection

COTS 18650 lithium-ion batteries were chosen as the energy source for the DAQ. The advantages of this battery satisfy the criteria in section 3.4.3 Battery Selection and include:

- The 18650 format encases the battery in a rigid metal cylinder which is well-suited for the space environment [45]. Battery formats which use a flexible pouch, like most lithium-polymer cells, are more prone to outgassing in the vacuum of space [45].
- Compared to other rechargeable battery solutions such as Ni-Cd and Ni-H<sub>2</sub>, Li-ion batteries have improved temperature range, energy density and specific energy and cycle life [46].
- COTS Li-ionbatteries are a mature battery format due to widespread use in consumer products [46].
- Extensive flight heritage as they have been proven in other CubeSat missions [45], and are being used on flagship NASA missions, such as Europa Clipper [47].
- COTS batteries are produced at scale for commercial markets and are lower cost compared to specialised aerospace batteries.

Manufacturers produce a variety of types of Li-ionbatteries with different chemistries, which affect parameters including internal resistance, discharge and charging temperature range and capacity.

The Samsung INR18650-25R Li-ionbattery was chosen for the DAQ platform due to

- Previous flight heritage on CubeSats [48].
- Operation over a large temperature range of -20 °C to 75 °C and has been proven to be stable at 130 °C [49].
- Good capacity of 2500 mA h and high maximum discharge rate of 20 A [49].
- Low cost.

### 4.2.3 Battery Pack

As stated in the high-level design (Section 3.4.4 Power Electronics), a 1S lithium-ion battery pack was selected. In the final design, three batteries were placed in parallel to form a 1S3P battery pack. Three 18650 cells were selected since this is the maximum number of cells which can fit on a 80 mm × 80 mm PCB using a 3 cell 18650 battery holder. This connector holds batteries using a leaf spring.

The 1S3P battery pack was mounted on the reverse side of the PCB due to the amount of space it consumes, as shown in figure 4.2.

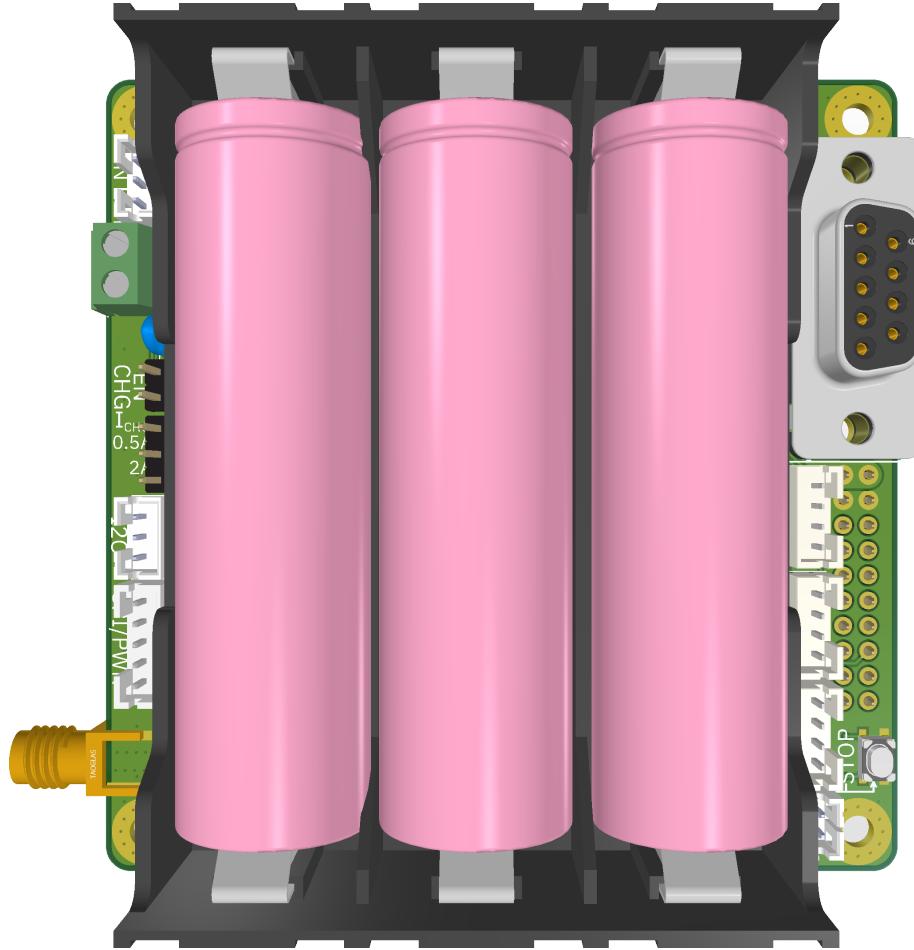


Figure 4.2: Battery mounting on reverse side of DAQ PCB.

The battery pack can be charged by connecting 5 V power to a charging terminal block. Two 1 A TP4056 linear Li-ion battery chargers were connected in parallel to the battery pack to allow a maximum charging current of 2 A.

#### 4.2.4 5 V Power Conditioning Design and Simulation

The Texas Instruments TPS61022 boost converter was selected for this 1S system. It had a working voltage range of 1.8 V to 5.5 V which was suitable for a 1S3P Li-ion battery pack with a working voltage range of 2.5 V to 4.2 V, a maximum output current of over 3 A and can be configured to have an output voltage of 5 V which was ideal for the DAQ and the camera payload [50]. A DC-DC switching converter was selected since it has high efficiency above 90% throughout the input voltage of a standard 1S battery pack [50]. The TPS61022 had an operating temperature range of  $-40^{\circ}\text{C}$  to  $150^{\circ}\text{C}$  which was adequate for thermal qualification [50]. This satisfies the requirements described in section 3.4.4 Power Electronics.

The inductor selected for the converter subsystem is the ASPI-0630HI-1R0M-T15, which was a  $1\text{ }\mu\text{H}$  coil with a low DC resistance of  $10\text{ m}\Omega$ . The coil had a saturation current of 20.5 A which was above the 8 A maximum switching current. To make manufacturing simpler, the voltage divider was changed from  $R_1 = 732\text{ k}\Omega$  and  $R_2 = 100\text{ k}\Omega$  to use  $R_1 = 750\text{ k}\Omega$  and  $R_2 = 100\text{ k}\Omega + 2\text{ k}\Omega$  since these resistor values are available on JLCPCB. A  $1\text{ pF}$  compensation capacitor was also added. To evaluate the effects of these choices, the converter was simulated in LTspice with a single fully charged lithium-ion battery as the input.

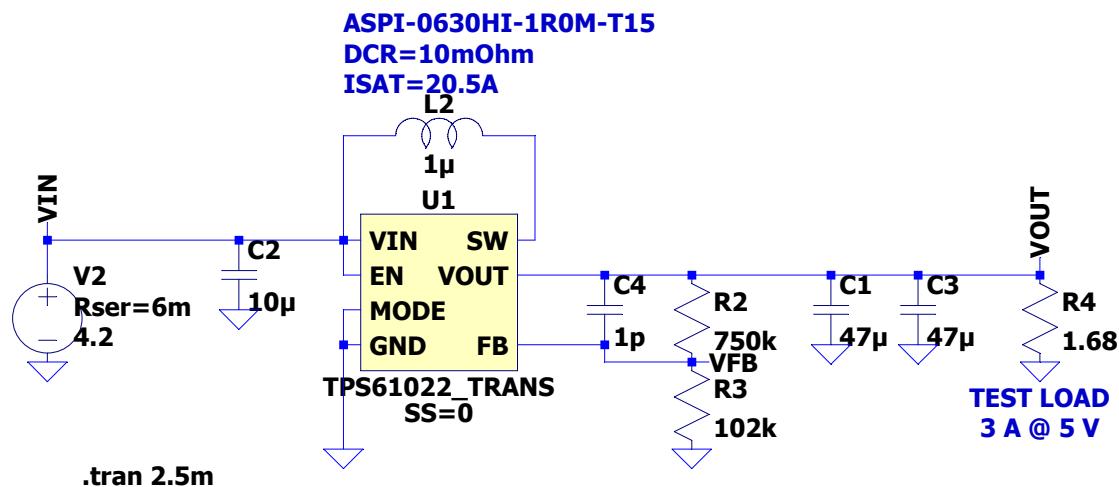


Figure 4.3: LTspice schematic representation of TPS61022.

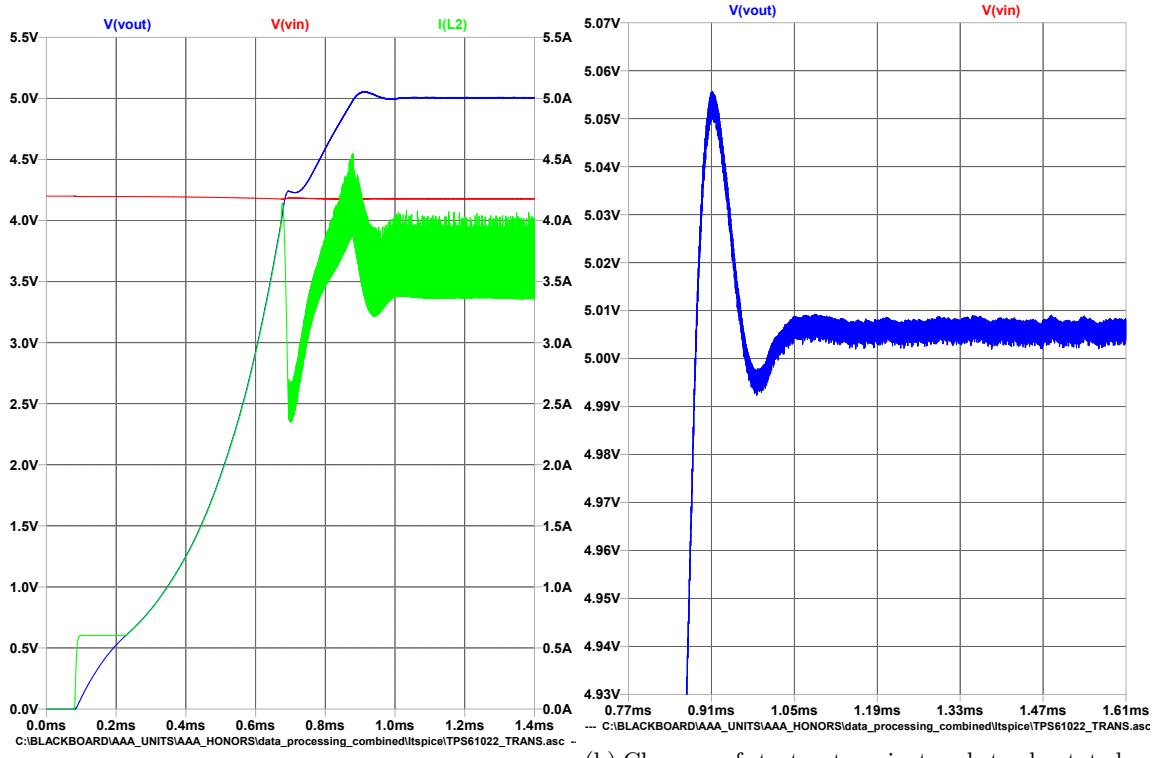


Figure 4.4: Simulation of the TPS61022 in LTspice using component values close to the datasheet as shown in figure 4.3.

As shown in figure 4.4, during startup there was a small voltage overshoot of 0.06 V, but this was negligible and well within the tolerance of the components. The steady-state output voltage has a mean of 5.006 V and a peak to peak ripple voltage of 7 mV, or a ripple of 0.15%, which was more than adequate for this application since it was mainly digital logic.

The simulation also showed that the power system would work up to 3 A, which was the output current required to emulate the power capabilities of POEM. Two TPS61022 boost converters were used in the final design: one was dedicated for the payload-under-test, and the other was for the DAQ. This allowed the payload to be able to receive the maximum current able to be provided by POEM, and also isolated failures in either system from affecting the other.

#### 4.2.5 Internal 3.3 V Power Conditioning

The DAQ internally used 3.3 V for some components, however this system only uses a maximum of 157 mA therefore efficiency is a lower factor and a linear regulator was chosen. This solution resulted in a power loss of 270 mW, which was considered insignificant compared to the maximum power output of the 5 V system (26 W).

An Advanced Monolithic Systems AMS1117-3.3 linear regulator was chosen due to its cheap

pricing on JLCPCB of only A\$0.20 and since it has been used in past designs with success. It has a high dropout voltage of 1.1 V, which was acceptable for the 5 V input [51]. The AMS1117 had an operating temperature range of  $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$  which was adequate for thermal qualification [51].

This regulator had been validated in previous boards, therefore implementing this subsystem only required cloning the schematic and layout to the design.

#### 4.2.6 EPS Subsystem Design

Each of the subsystems were arranged as shown in figure 4.5. The payload and DAQ were powered off separate boost converters, since the payload can potentially use the full 3 A capacity of a single boost converter. The output of the two converters are not combined to ensure a failure on one power rail would not affect the other. Regulation of the 3.3 V supply occurs after boosting to 5 V since the high dropout voltage of the regulator prevents it operating directly from the battery.

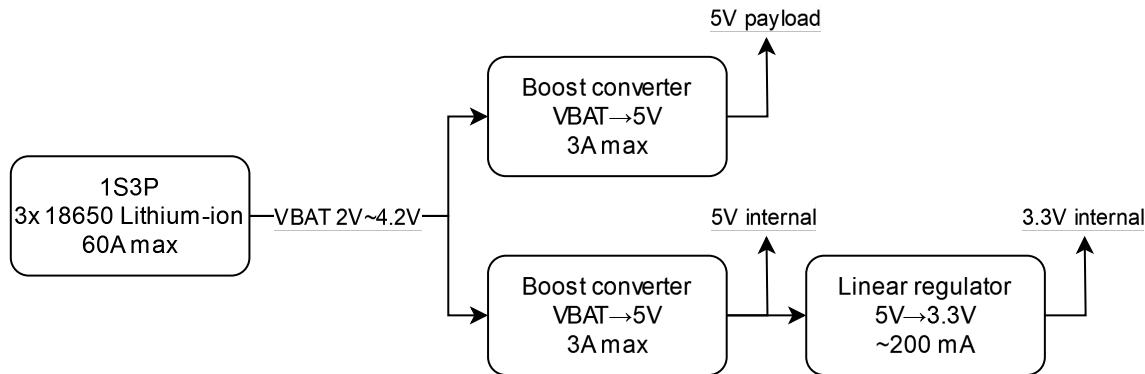


Figure 4.5: Electrical power system (EPS) and output voltages

#### 4.2.7 Integration into DAQ

A single schematic was created for the TPS61022 based off the successful LTspice simulation and instantiated twice in the root schematic. A PCB layout based on Figure 10-1 from the datasheet was used and is shown in figure 4.6 [50].

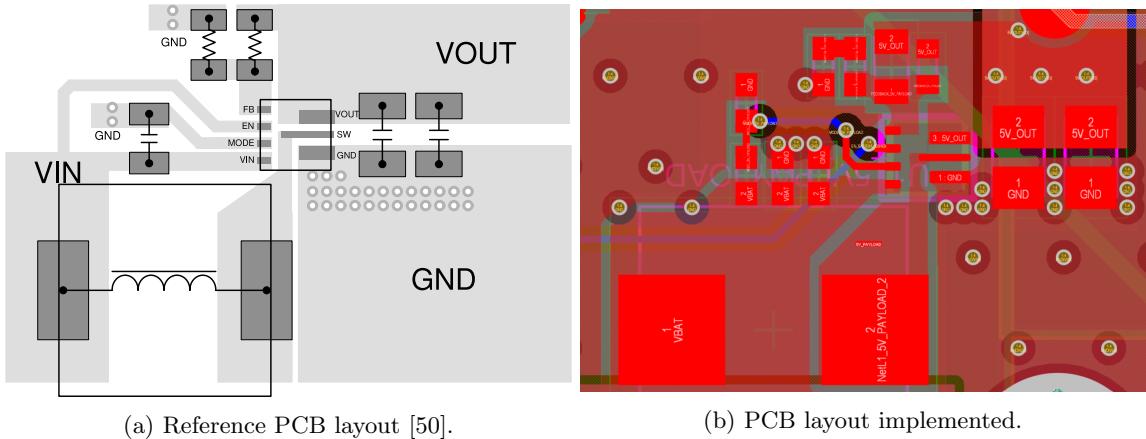


Figure 4.6: Layout of TPS61022 implemented in EPS.

The 3.3 V regulator was implemented from the reference design. The two ceramic capacitors were placed close to the regulator inputs and outputs to minimise noise. The output of the regulator was connected to the internal 3.3 V power plane through multiple vias.

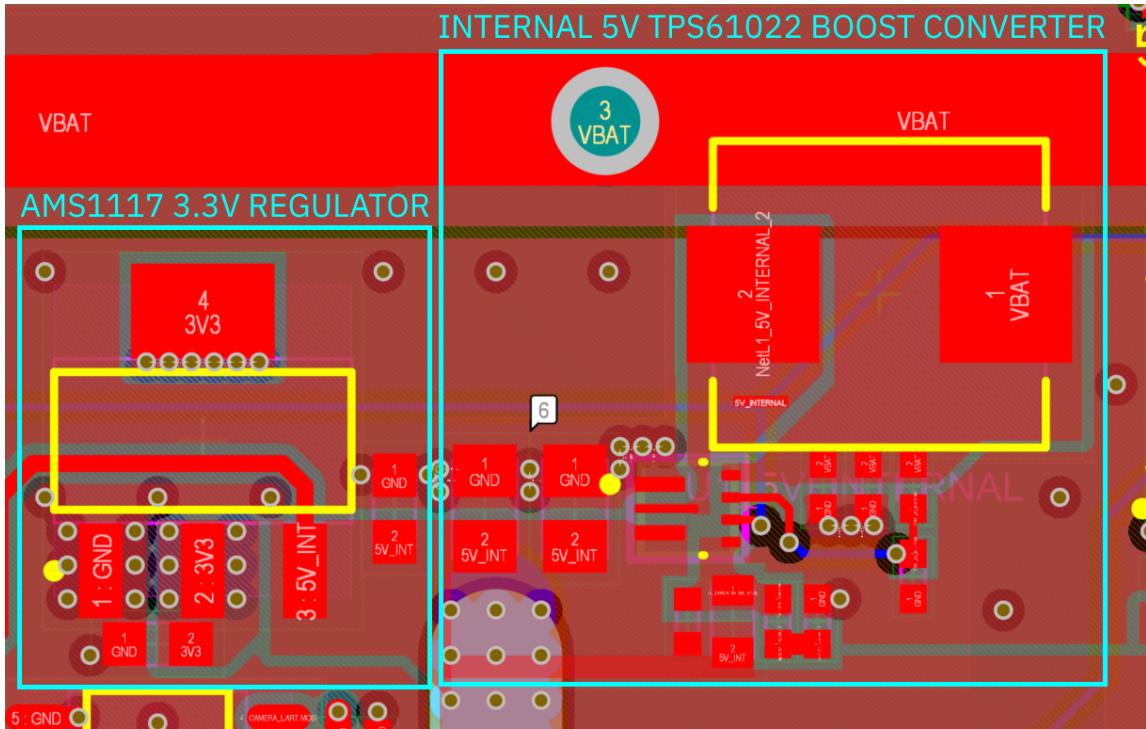


Figure 4.7: PCB layout of internal power conditioning, includes the layout of the AMS1117 regulator and another instantiation of the TPS61022 layout.

## 4.3 Onboard Data Handling (OBDH)

A first revision of the DAQ used an STM32L476 microcontroller, which had similar peripherals, including UART, I<sup>2</sup>C and SPI, however as a microcontroller it did not have an operating system, nor significant storage. Storage was provided in the form of a 4 GB embedded multimedia card (eMMC) chip, which was chosen since it is directly soldered to the PCB and was more resistant to vibration than a micro SD card connector.

Due to the issues encountered with the first revision and due to the time constraints, the second revision uses a Raspberry Pi Zero W v1.3 (referred to as "Pi Zero"). This was a development board which integrates the BCM2835 Broadcom system on chip (SoC), 512 MB of RAM and contained a micro-SD card slot, USB interface and peripherals such as UART, I<sup>2</sup>C and SPI [52]. This board was chosen due to its small form-factor compared to larger Raspberry Pis, simplicity of integration compared to the Raspberry Pi compute modules and low cost. The Raspberry Pi platform has been used in low-cost low Earth orbit (LEO) CubeSat applications [53]. It was predicted that in polar orbits that the Pi Zero has a lifespan of 5 years [53], which was adequate for this project since the POEM would cease to maintain its orbit after several months. The thermal performance of a Raspberry Pi Zero W was adequate for space applications [53]. While a Raspberry Pi Zero 2W was initially selected, which would have had higher performance compared to the Zero v1.3, it was not possible to acquire one due to supply chain issues.

Typically, a Raspberry Pi runs the Raspbian operating system (OS), which is a Debian fork [52]. Compared to developing for a microcontroller, an OS is easier to develop for as it allows the use of standard Linux utilities for interacting with the system, including `ssh` for remote control, and having each DAQ task in a separate process with separated memory makes debugging easier. This ease of development was required to meet the time constraint of the project.

Due to limitations of the BCM2835 SoC only one hardware UART was available [52], but more are required for receiving data from GNSS receivers. Receiving data through a software implementation of UART is not ideal since it uses a large amount of CPU resources and is prone to missing bits especially on the non realtime OS of the Raspberry Pi which severely limits its usable baud rate. The XR20M1172 dual hardware UART to add more UART ports to the Raspberry Pi through the SPI bus. This UART has a 64-byte first-in first-out (FIFO) buffer and interrupts, which eliminates the need for expensive polling, and has a maximum data rate of 16 Mbit s<sup>-1</sup> [37], which was more than adequate for GNSS data which has a maximum baud rate of only 38 400 bd. This UART has a temperature range of -40 °C to 85 °C, allowing it to survive temperature testing.

### 4.3.1 Camera Communications Interface and Camera Data Downlink Connector

The final design of the camera payload interface used a DE-9 connector. This was chosen due to its low cost and ability to be secured using bolts instead of by friction alone. The pinout of the connector is shown in figure 4.8. The RS-485, power and ground lines were duplicated for redundancy.

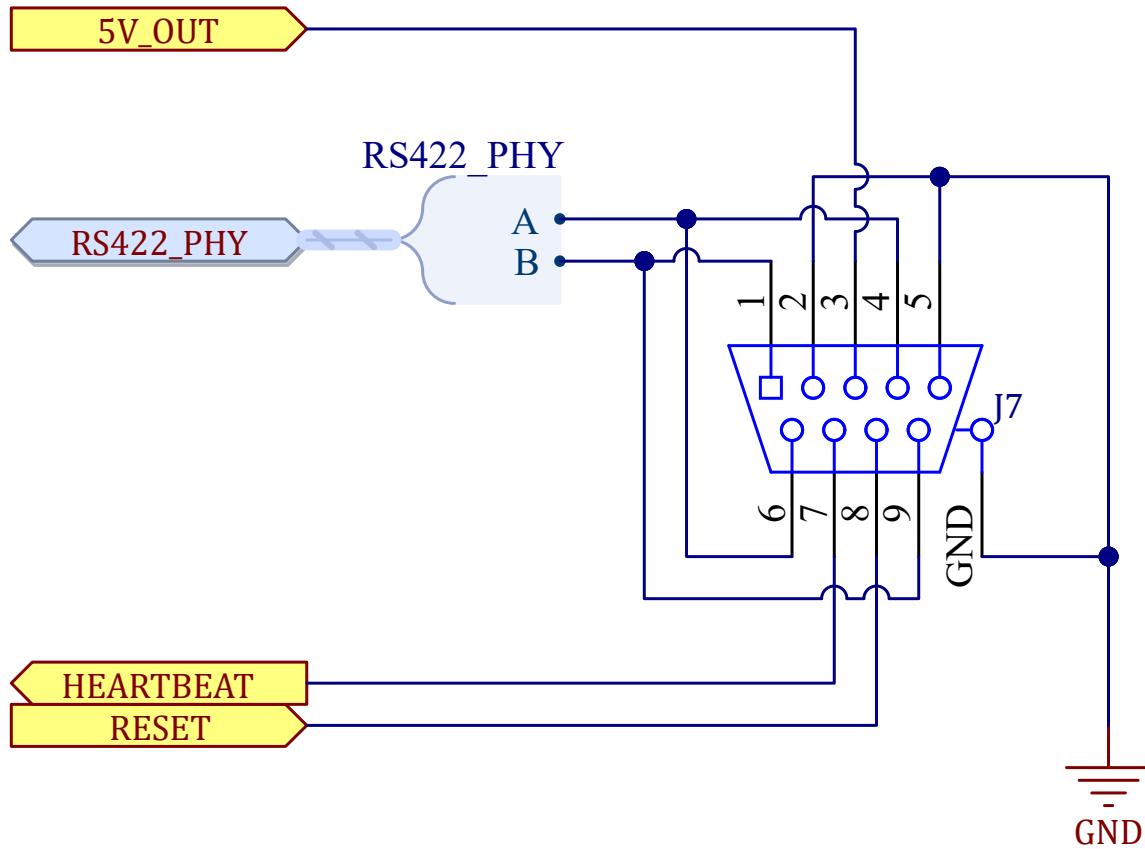


Figure 4.8: Pinout of DE-9 connector.

A female DE-9 connector was soldered to the DAQ, a male solder cup DE-9 terminal was soldered to 22 AWG copper cable to form the wire harness to the camera payload.

### RS-485 Receiver

The SP3485EN-L/TR RS-485 transceiver was used to convert the RS-485 signalling to single-ended UART. The transceiver was fixed in receive mode since there is no way for the POEM to send data to the payload. The transceiver contains a  $120\ \Omega$  d.c. termination as this was the end of the RS-485 bus, the resistor is sized to handle the power dissipation of a 5 V RS-485 bus. The receiver contains a PSM712-LF-T7 transient voltage suppressor (TVS) diode array, which was designed to prevent the transceiver from being damaged by electrostatic discharge (ESD) due to manual handling of the connector.

### Transport Layer

The camera payload communicates with the DAQ through UART at the transport layer. The settings in table 4.2 were agreed upon and are used on both the payload and DAQ sides:

Attribute	Value	Note
Speed	57 600 bd	This is significantly more than the 5000 bd connection provided by POEM, but the drone and HPR tests have limited time and require faster transmission to receive any picture during the test.
Data bits	8	
Stop bits	1	
Parity	None	
Flow control	None	This is a one-way connection, so software flow control is not possible.

Table 4.2: UART settings.

### Presentation Layer

Several simple formats are used in the presentation layer for this project. The data stream may contain either messages or image data.

Messages consist of a header, the message data of a variable length and an end of message marker as shown in figure 4.9. The header is sufficiently long and unique that it should not appear in the image data stream. Since messages are not often transmitted, this overhead is acceptable.

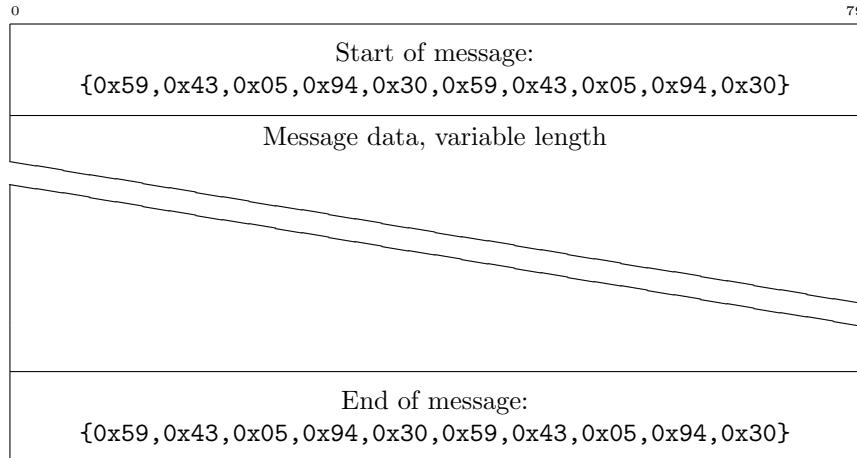


Figure 4.9: Message structure.

A chunk protocol is used for the transmission of images in the final design. A preliminary design simply sequentially transmitted each byte of the image, which was adequate when the radio was placed close to each other ( $> 10\text{ m}$ ). However, when testing past this range and at any significant distance, the errors caused the received image to become unrecognisable. The type of error observed consists of contiguous bytes being lost occasionally, rather than frequent corruption of individual bytes as shown in 4.10.



Figure 4.10: Example of an image received with errors resulting in an unrecognisable image due to dropped bytes causing the image to shift at each error. Only a small proportion of the image is lost (< 3%), represented by red pixels at the bottom and contiguous sections of the received image are error-free.

A chunk protocol was effective against this type of error while being simple to implement. It involved adding a header to signify the start of a chunk, followed by the chunk number and 400 bytes of pixel data as shown in 4.11. A  $1600 \times 1200$  pixel image where each pixel is represented by one byte required 4800 chunks to be sent in this implementation. The number of bytes per chunk is a trade-off between pixels lost per error and effective data rate.

A start of chunk marker of {0x00, 0xFF, 0x00, 0xFF} was chosen since the image data being transmitted are raw pixels, and it was unlikely for the image to contain pixels that are full or zero intensity, and even more unlikely for them to be adjacent to each other.

When an error occurs causing a string of bytes to be dropped, it causes the next chunk to not be encountered after 400 bytes. When this error occurs, the receiver searches for the next occurrence of the header and restarts receiving from that point. The chunks which were skipped have their pixels filled in the received image to prevent the error causing the image to become unrecognisable.

The end of an image is signified by the ASCII string "EOFEOFEOFEOF". The start of chunk

of the next image immediately starts after this string.

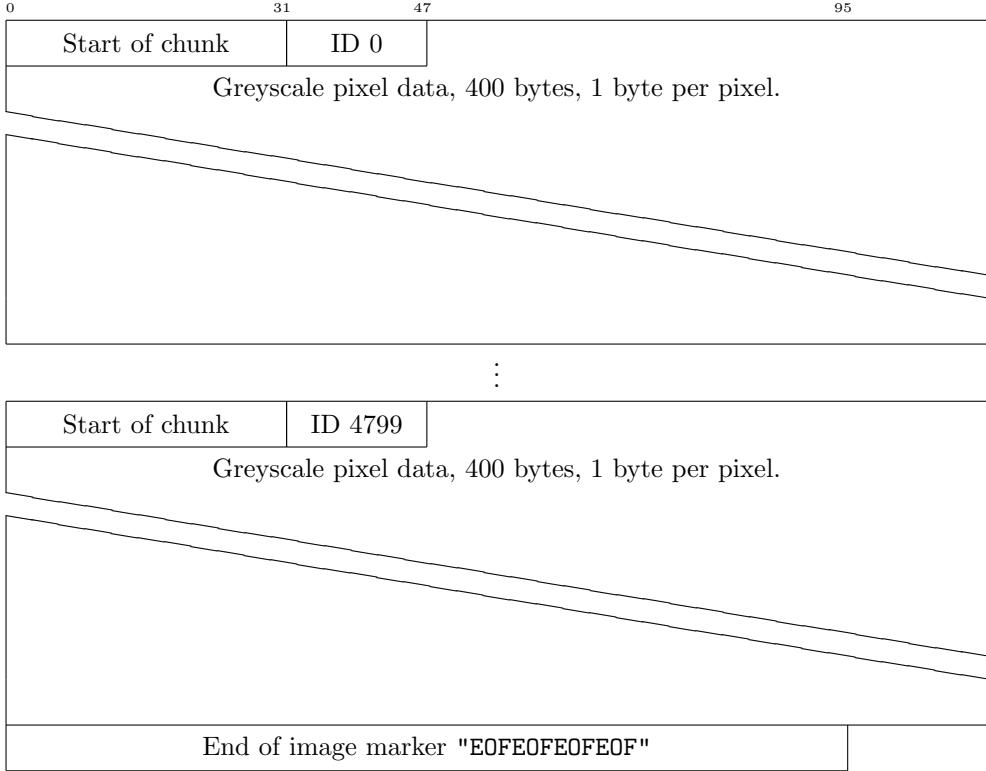


Figure 4.11: Image chunk protocol.

### Radio Downlink

The RFD900x radio transceiver was chosen to emulate the POEM radio service. This transceiver uses the 915 MHz industrial, scientific and medical (ISM) band and transmits with a maximum power of 1 W using a frequency hopping spread spectrum (FHSS) technique [32]. Data rates from  $12 \text{ kbit s}^{-1}$  to  $224 \text{ kbit s}^{-1}$  are available with the default firmware [32]. The RFD900x uses a UART to transmit and receive data [32].

The RFD900x satisfies several constraints. It reduced the time to test since it uses the ISM band, which can be used by anyone provided they follow the Low Interference Potential Devices (LIPD) Class License legislation. The use of FHSS allowed the RFD900x to transmit at the maximum power of 1 W that is allowable by the class license under the frequency hopping transmitters section [38].

Distances of 40 km line-of-sight were possible using the RFD900x [32], which was far greater than the maximum distance achievable with the rocket and drone tests. The maximum drone test scheduled had an altitude of 500 m, and the rocket was intended to fly to 10 000 ft (3 km).

The radio transceiver had a temperature range of  $-40^\circ\text{C}$  to  $85^\circ\text{C}$ , which satisfied the range of temperatures required to pass the temperature testing [32].

## **Camera Communications and Downlink Software**

A Python script was developed for the DAQ which:

1. Opens the serial port `/dev/serial0` with the settings described in table 4.2.
2. Create and open a log file with a unique filename to dump the received bytes from the camera payload.
3. Periodically read a block of 2048 from the UART receive buffer and write the block to the file and UART output.
4. Every 3 seconds write a message to the UART output containing the current system time, the script's uptime and the number of UART errors.

This Python script is managed by a `systemd` service which reboots the script if it crashes.

The ground station is a laptop running Windows with an RFD900x connected to it over USB serial. There are two Python scripts on the laptop, the first saves the data from the radio to a file and prints any messages to the console.

1. Opens a serial port to the RFD900x with the UART settings of the ground RFD900x.
2. Create and open a log file to store the received bytes from the radio.
3. Periodically:
  - Read the bytes in the serial buffer,
  - Save the bytes to the log file,
  - Find any start of message mark, and if encountered print all bytes until the end of message mark is received

A second script periodically reads the latest log file and splits it into images.

1. Allocate a `bytearray` for a full  $1600 \times 1200$  pixel image.
2. Allocate a  $1600 \times 1200$  `bytearray` where each byte is the state of each pixel (received or missing).
3. Read the whole file and strip all messages (bytes enclosed by the start and end of message markers).
4. Then find the indices of the end of image makers and split the file into several bytearrays, one for each image.
5. For each image bytearray:
  - Find the next start of chunk
  - Extract the chunk ID from the next two bytes.
  - Extract image bytes starting from after the chunk ID to either the next header index, or the end of file if this doesn't exist.
  - Add all bytes to the image byte array, starting at index `ID * 400` and set the state of the added pixels to "received".
  - Repeat until the image array is full or there are no more bytes left in the image bytearray.

**Results** This system was tested in the drone test at a distance of 120 m line of sight. The image received is shown in 4.12 with comparison to an image received without blocking:

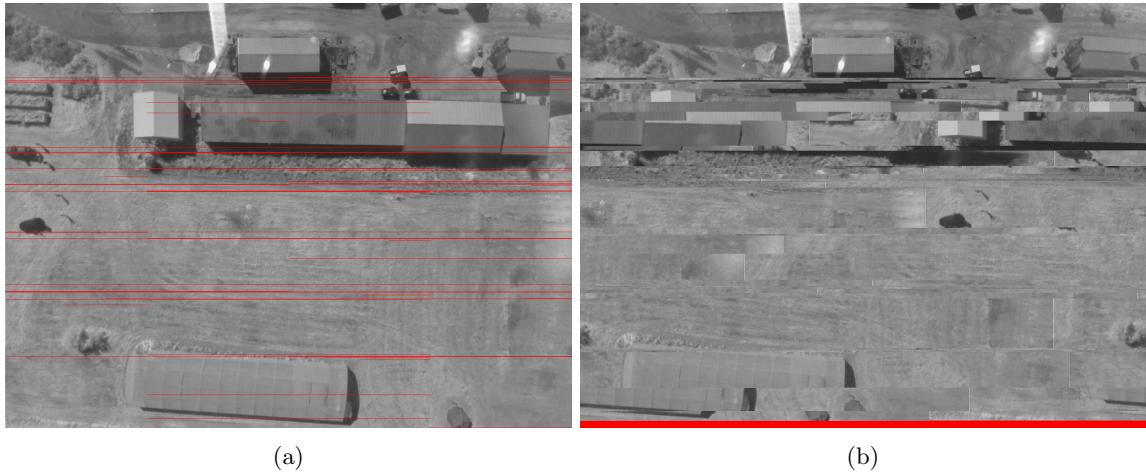


Figure 4.12: Comparison of two image reception techniques. Image (a) used the blocking protocol with a block size of 400 pixels. Image (b) is an example of what the image would look like without the image blocking.

### 4.3.2 Accelerometer Data Acquisition System

#### Accelerometer Selection

Two accelerometers MEMS were chosen, the ADXL375 and LSM6DSOX.

The STMicroelectronics LSM6DSOX is a MEMS inertial measurement unit (IMU) which has a  $\pm 16\text{ g}$  accelerometer and  $\pm 2000\text{ m}^\circ\text{s}^{-1}$  gyroscope, both with a sampling rate of 6666 kHz. This accelerometer was used to characterise the random vibration spectrum of launch due to its high sampling rate.

Due to the low full-scale of the LSM6DSOX of only  $\pm 16\text{ g}$ , the ADXL375 was chosen to characterise the shock response of the payload to pyroshock due to its significantly higher full-scale range of  $\pm 200\text{ g}$ .

Based on the accelerometer specifications, these accelerometers satisfy the requirements outlined in table 3.10.

#### Accelerometer PCB

The accelerometer PCB (shown in figure 4.13) contained both the ADXL375 and LSM6DSOX accelerometers and mounted them to the chassis.

To maximise the resonant frequency of the assembly, the PCB was mounted on its flat side to the chassis and was fixed with four steel M3 bolts on each corner. Additionally, the dimensions of the PCB was minimised to 22 mm  $\times$  22 mm, which was the minimum dimension that can fit all components. Maximising the resonant frequency past the sampling frequency would minimise distortion of the response. The PCB is depicted in figure 4.13.

The PCB exposed one interrupt and chip select pin for each accelerometer, an SPI connection and 3.3 V power bus.

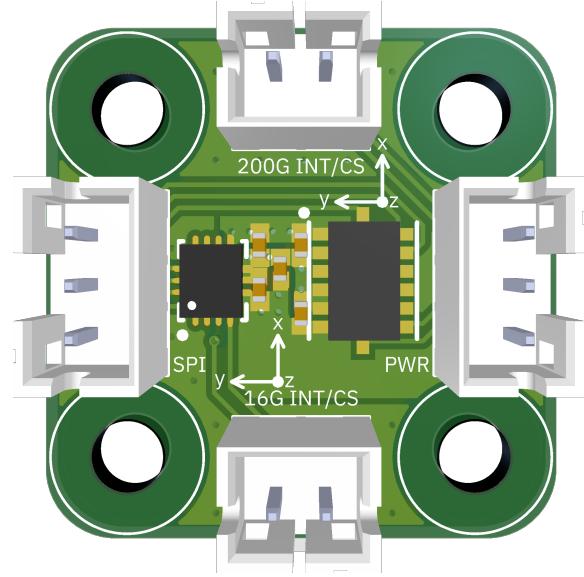


Figure 4.13: Accelerometer PCB.

### Connection to DAQ

The accelerometers were connected to the DAQ using the SPI0 bus of the Pi Zero. The chip select ( $\overline{CS}$ ) and interrupt (INT) pins were connected to GPIOs and handled by software, therefore it was not necessary to only use the hardware SPI chip select/chip enable (CE) pins for all chip selects. This is depicted in figure 4.14. Although SPI requires more wiring, in the case of these sensors SPI has a higher clock speed compared to I<sup>2</sup>C. For example, the LSM6DSO has supports the I<sup>2</sup>C Fast Mode+ extension (FM+) which gives the I<sup>2</sup>C bus a maximum clock rate of 1 MHz, but the SPI has a maximum clock speed of 10 MHz which is higher. SPI also has higher data rate for a given frequency than I<sup>2</sup>C since I<sup>2</sup>C adds additional flow control bits such as start, stop and acknowledgement bits.

This SPI bus was dedicated for only accelerometers to maximise the number of accelerometers connected, and was not shared with non-accelerometer peripherals. Two accelerometers were used as redundancy.

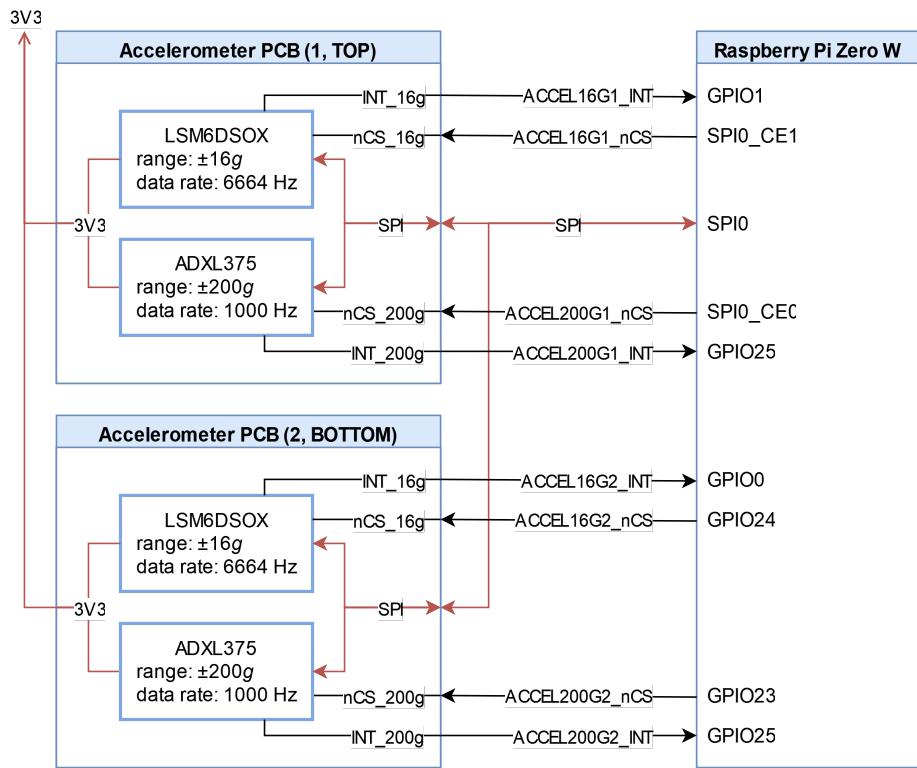


Figure 4.14: Block diagram of accelerometers and Pi Zero data acquisition system.

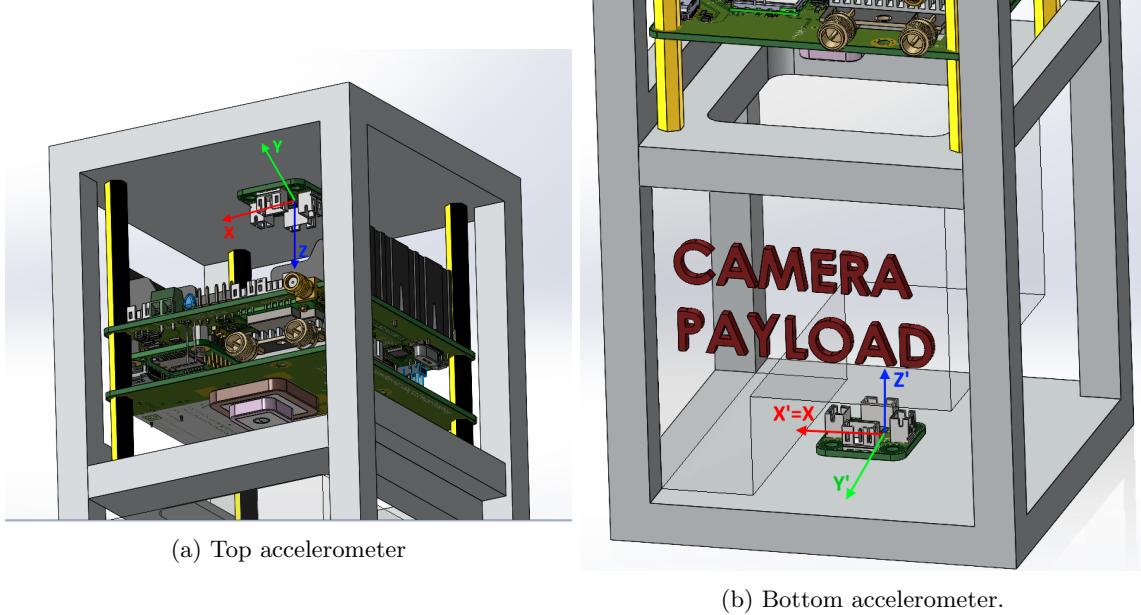


Figure 4.15: Mounting positions of accelerometers on the CubeSat chassis. The top and bottom accelerometer share a common  $x$  axis but have opposite  $y$  and  $z$  axes.

### LSM6DSOX Software

A C program was written to read values from the LSM6DSOX, package them into a binary flat file format and store them to disk. A C program was required since preliminary tests with a Python script with the same functionality resulted in 100% of the CPU being used to read from one accelerometer.

The `lsm6dso-pid` platform independent (PID) library was used to provide simple access to the registers of the accelerometer [54], in combination with WiringPi which provided bindings to GPIO, SPI and other peripherals on the Pi [55].

To maximise sampling rate and minimise CPU usage, the accelerometer is operated using the FIFO and interrupt features.

Samples from the accelerometer were stored in binary flat file format, consisting of the `struct datapoint` for each 3-axis sample. A header of `0x59` was used which signifies the start of a sample. The type field was set to either 1 for accelerometer samples or 2 for gyroscope samples.

```
struct datapoint
{
    char header;
    char type;
    float x;
    float y;
    float z;
};
```

To improve reliability, the LSM6DSOX program contains a watchdog, implemented using a POSIX timer, which restarts and re-initialises the accelerometer if no samples have been received for 1 s. The process is managed by a `systemd` service which initialises the program on system boot and restarts the program if it crashes.

The following is a description of the program

1. Read the first argument, which describes the sensor that the process will manage (1 or 2), and open the appropriate SPI bus using WiringPi.
2. Open a new unique binary file to store the samples to.
3. Create a watchdog POSIX timer and debug printing POSIX timer.
4. Attach the FIFO read interrupt handler on the interrupt pin.
5. Initialise the LSM6DSOX by issuing the following commands:
6. Ensure the `WHO_AM_I` register matches the expected value,
7. Software reset the device,
8. Wait for the device to be reset,
9. Disable the I<sup>3</sup>C interface,
10. Enable block data update,
11. Set the scale for the accelerometer to 16 *g*, the maximum full scale possible,
12. Set the sampling rate to 6666 Hz and the batching rate to 12.5 Hz,
13. Set the FIFO to continuous mode (old samples are automatically discarded),
14. Set FIFO watermark level to 384 samples,
15. Set interrupt pin 1 to pulse on FIFO watermark being reached (this results in a pulse being generated on the INT1 pin when a large amount of data is present in the FIFO to be read, resulting in the interrupt handler being triggered.).

The interrupt handler has the following functionality and is based off sample code from the LSM6DSOX PID library [54]:

1. Immediately reset the watchdog timer to 1 second (to prevent it triggering since a read is started)
2. Read the number of samples in the FIFO. For each sample:
3. Read the sample's tag and if it is accelerometer data then read the sample from the FIFO into a 6-byte array
4. Convert each of the three axes (represented by two byte words) to mg.
5. Store the reading in a struct with the correct header and type and write to the log file.
6. For debug reasons, print the current sample every 3332 samples (every half second).

### **Unpacking Data for Data Analysis**

The accelerometer binary files were read on a PC using a Python script which uses the `struct` library to unpack the file and converted it to a traditional comma-separated values (CSV) file.

### **ADXL375 Software**

A separate process was created for the ADXL375, however the code uses the same procedure for the LSM6DSOX. The only changes are the initialisation process, where it sets registers to put the accelerometer into  $200\text{ g}$  full scale and 1600 Hz sampling rate, and changes in the watermark level. The addresses of the registered were changed for the ADXL375, but the high-level procedure is similar.

#### **4.3.3 GNSS Tracking**

The u-blox NEO-M9N was used for tracking [33]. This is a multi-GNSS receiver which is able to receive from multiple GNSS constellations simultaneously, which results in a faster acquisition time and greater interference immunity [33]. The receiver can report position with an accuracy of 2.0 m (circular error probable), which is adequate for a HPR tracking application [33]. The NEO-M9N was used instead of other u-blox receivers due to its high navigation update rate of 25 Hz which is useful due to the high speed of a HPR flight [33].

Since POEM provides the location of the CubeSat and due to the speed and altitude restriction of the NEO-M9N of  $500\text{ m s}^{-1}$  and 80 km respectively, this receiver will not be present on the space launch and is only required for the HPR launch.

A differential GNSS (DGNSS) solution was considered and tested based on the u-blox ZED-F9P, however the drone test did not require the centimetre level precision of the ZED-F9P, so it was not used in the final design.

## Chapter 5

# Final Experiment Design for HPR Evaluation

The design of the DAQ was a precursor for the experiment to compare HPR to shaker table testing. Note that the results of the HPR and shaker table tests will also be used for evaluating the DAQ in addition to the goal of evaluating the usefulness of HPR launches for qualifying CubeSats.

### 5.1 High-Power Rocket Flight

A custom rocket named *UNO* was designed and built by another project member (Jamir Khan) from scratch, it had a height of 290 cm, diameter of 16.3 cm and a dry mass of 14.42 kg without a motor. It was designed to fly with an M impulse class motor, however due to changes in United States export regulations it was not possible to obtain this motor in time, therefore it was only possible to launch with a K impulse class motor which has about 1/10th of the total impulse of the N motor as shown in table 5.1. All analysis following this assume a K1100T motor.

Total impulse [Ns]	Motor impulse class
160.01 - 320.00	H
320.01 - 640.00	I
640.01 - 1,280.00	J
1,280.01 - 2,560.00	K
2,560.01 - 5,120.00	L
5,120.01 - 10,240.00	M
10,240.01 - 20,560.00	N
20,560.01 - 40,960.00	O
40,960.01 - 81,920.00	P
81,920.01 - 163,840.00	Q

Table 5.1: Rocket motor impulse classes [56]

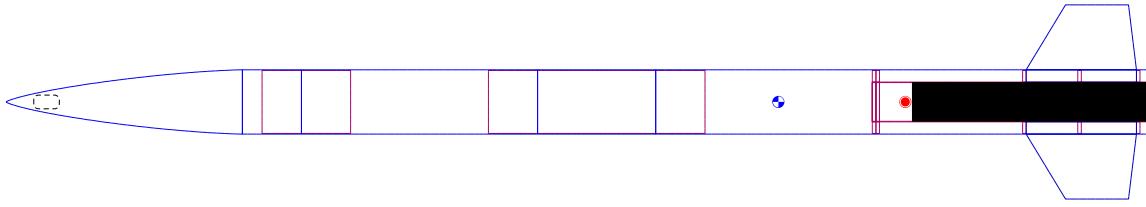


Figure 5.1: OpenRocket diagram of *UNO*.

### 5.1.1 Simulation

The rocket was simulated using OpenRocket [57], [58], an open-source simulator which can predict parameters such as stability and acceleration based on empirical methods which use the rocket's shape and basic environment parameters such as constant wind [58], [59]. OpenRocket was used to ensure the rocket design was stable throughout launch and flight, which was important to ensuring the CubeSat payload does not become damaged by this qualification method. However, as it used a simple empirical model of the flight, it was not designed to model the effect of the motor and aerodynamic forces on the vibration environment in the rocket. It also does not simulate pyroshock events, instead modelling parachute deployment events as simple changes in the aerodynamics of the rocket [58].

#### Flight Profile

As shown in 5.2 the rocket was anticipated to reach an apogee of 413 m at 9.74 s and a total flight time of 30 s.

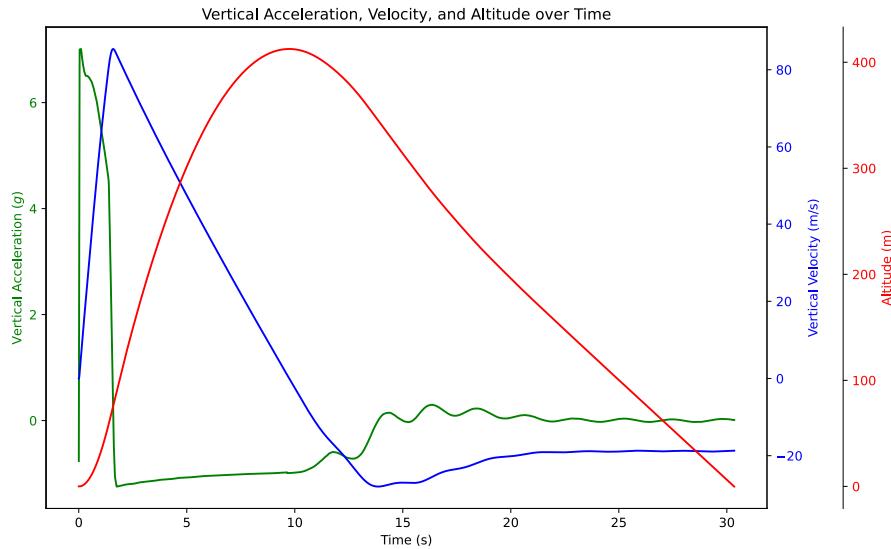


Figure 5.2: Flight profile of *UNO* using a K1100T motor. Simulated in OpenRocket.

## Stability

As shown in figure 5.3, the stability was above 2.0 calibres for the coast and launch phase, which was a general rule to ensure the rocket was stable and will not veer off course [25]. The short moment of stability below 2.0 occurs when the rocket reaches apogee, which was not an issue since the parachutes are immediately deployed at this point.

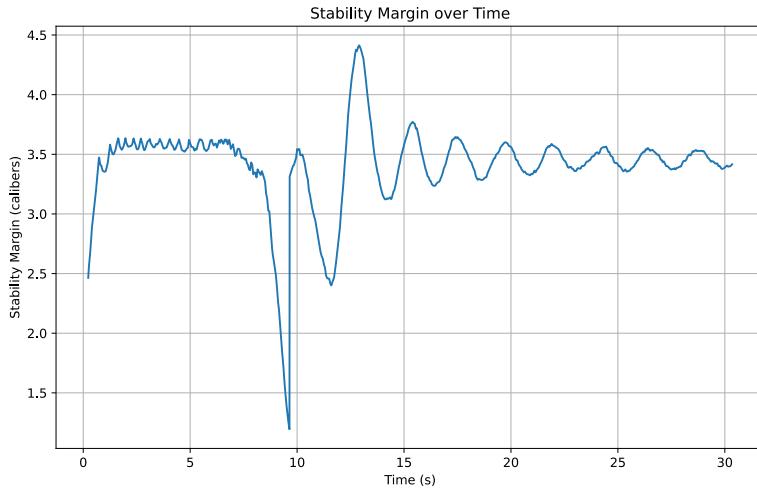


Figure 5.3: Stability of *UNO* using a K1100T motor. Simulated in OpenRocket.

## Acceleration

As stated, since OpenRocket does not model the vibration environment in the rocket and models the rocket as one solid body, only the acceleration of the whole rocket can be modelled. Pyroshock events are not modelled by OpenRocket. The launch phase lasts only 1.6s and has a high average acceleration of  $5.77\ g$ , as shown in 5.4. During the coast phase, the rocket is decelerated by gravity as expected and after parachute deployment the rocket only has a small deceleration force.

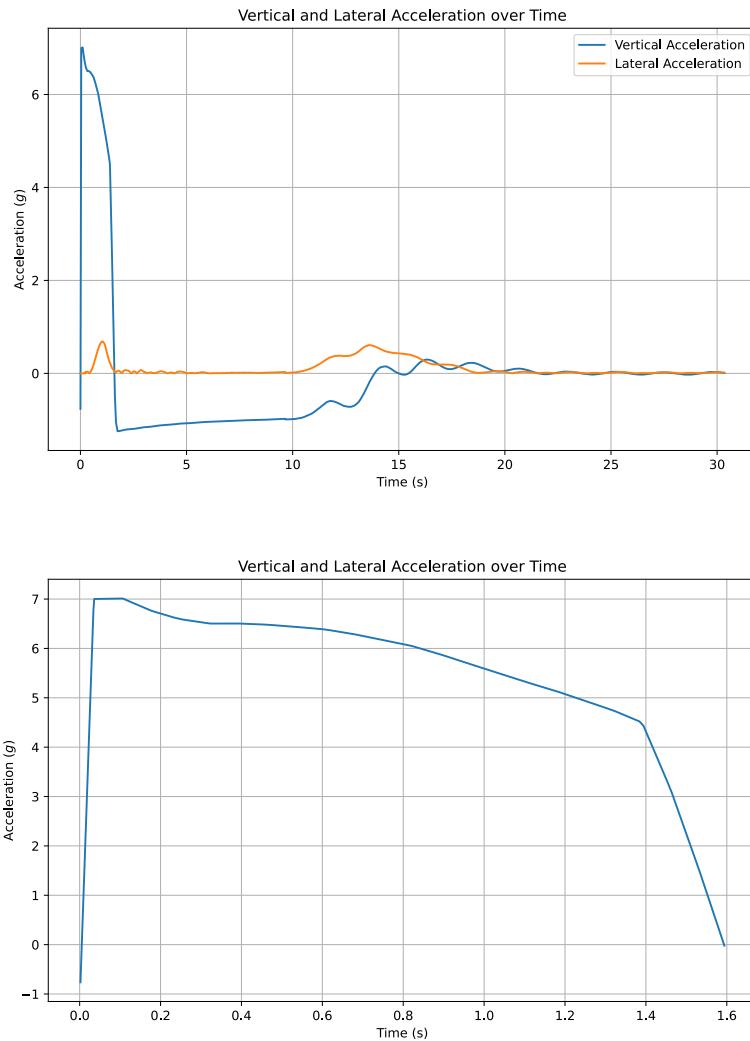
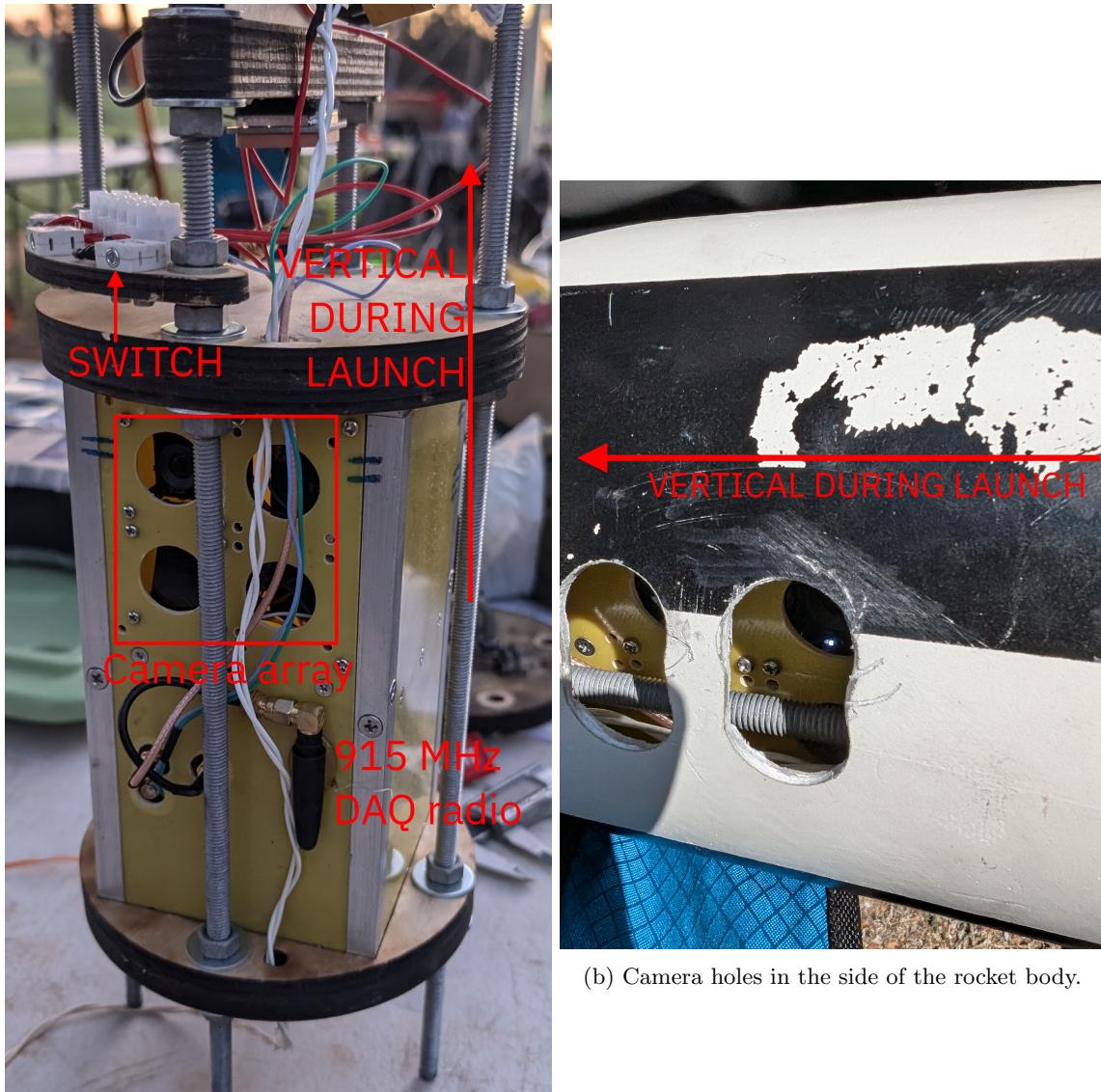


Figure 5.4: Acceleration of *UNO* using a K1100T motor over (top) the whole flight and (bottom) the thrust phase. Simulated in OpenRocket.

### 5.1.2 HPR Experiment Setup

The CubeSat was mounted in the rocket *UNO* payload bay through two wooden plates as shown in figure 5.5. The rocket splits into multiple parts on the ground for assembly: the payload bay was in the middle of two parts, both have wooden bulkheads epoxied to the body. The payload assembly and the payload bay tube was placed in between the two rocket parts. The payload assembly was fixed to the top and bottom wooden bulkheads to fix it in place.



(a) CubeSat mounting hardware before mounting in *UNO*.

Figure 5.5: CubeSat in *UNO* before launch.

After mounting the DAQ payload and setting up recovery electronics and parachutes, the K1100T motor was mounted in the motor tube. The rocket was then transported to the launch pad where it was mounted vertically on a launch rail and a motor igniter was attached.

The rocket was launched remotely, and the DAQ continued to save accelerometer data during the flight. A video was taken during flight to label the response to certain flight events.

Once the rocket landed, it was safed by switching off power to the recovery electronics and the DAQ. The recovered rocket was transported back for disassembly and retrieval of the CubeSat.

### 5.1.3 HPR Data Analysis

After the rocket landed, the accelerometer data files were extracted from the Raspberry Pi's SD card and transferred to a computer. A Python notebook was created to process the data as follows:

- Unpack the binary flat file into a CSV.
- Plot the CSV.
- Visually identify the start of the flight which is characterised by a sustained increase acceleration for a few seconds.
- Using video of the flight, identify the end of the flight and all key phases (including boost, coast and pyrotechnic events).
- Separate the CSV into smaller CSVs which contain the boost and coast phases and pyroshock events.
- For the boost and coast phases, which represent quasi-static acceleration and random vibration respectively, use Welch's method to obtain an estimation of the power spectral density.
- For the pyroshock events, originally Welch's method would be used, however it was found that plotting the time-domain shock response was more valuable for this analysis due to frequency limitations of the both accelerometers in the DAQ and shaker table.

This process is summarised in figure 5.6.

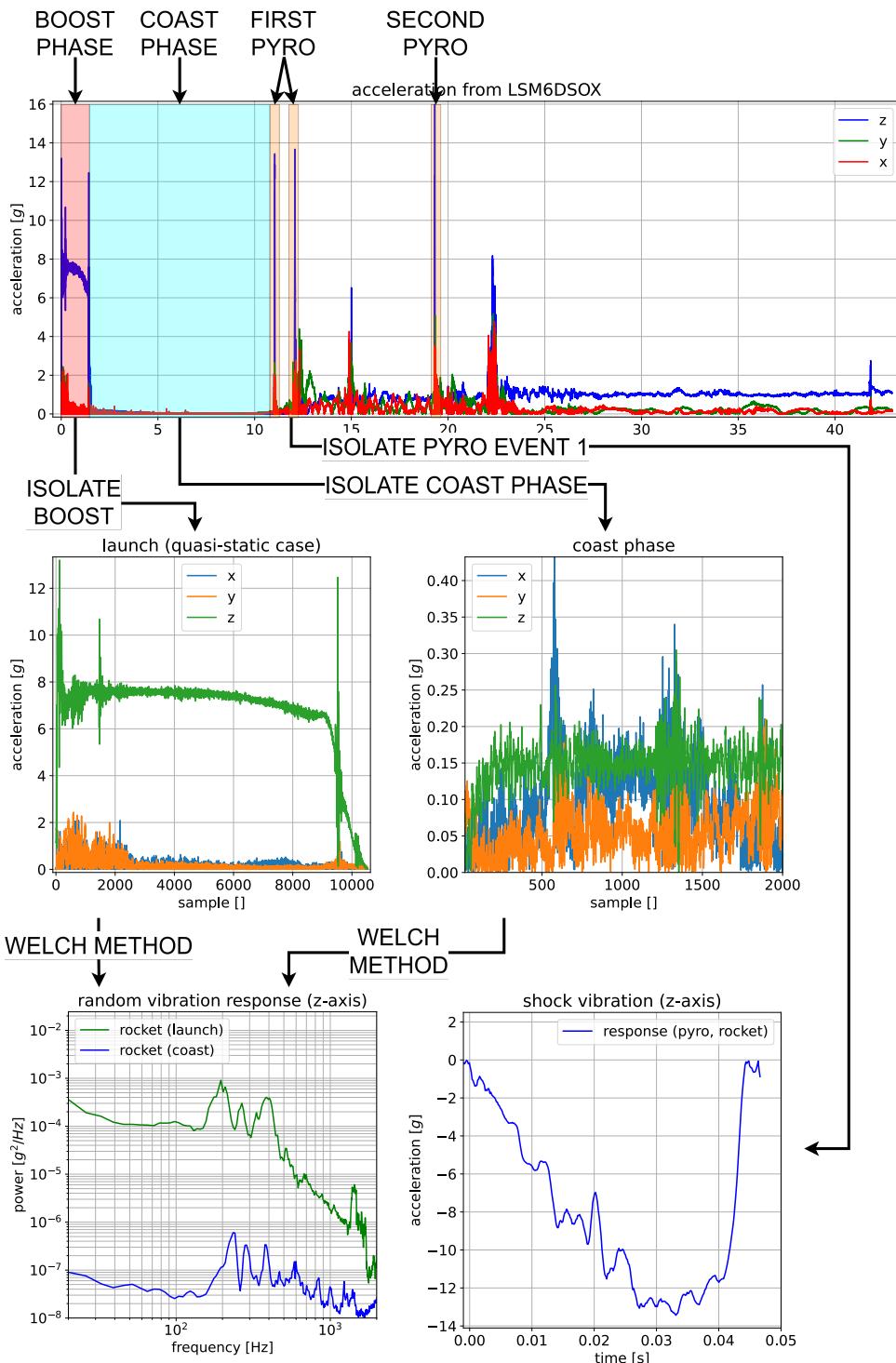


Figure 5.6: Data processing workflow for HPR accelerometer data.

### 5.1.4 Shaker Table Experiment Setup

## 5.2 Shaker Table

The shaker table tests were performed at AVI on the September 25, 2024 using a Brüel & Kjær LDS V8800 electrodynamic shaker table. The CubeSat was fixed to the shaker table using three bolts which went through the bottom plate.

A Brüel & Kjær type 4533-B integrated electronics piezoelectric (IEPE) accelerometer was used as the control and data accelerometer, which were mounted to the shaker table and the payload respectively. This accelerometer has a frequency range of 0.016 Hz to 1250 kHz and a resonance frequency of 36.7 kHz, which is well above the range of the tests. The accelerometers were attached to mounting studs which were fixed to the CubeSat using bisphenol-A epoxy.

The table was first mounted in the vertical configuration and the CubeSat was mounted axially as shown in figure 5.7. Random and shock tests were performed, then the table was rotated manually 90° and the CubeSat re-mounted to conduct the same tests in the x-axis. The CubeSat was rotated 90° to finally test the y-axis.

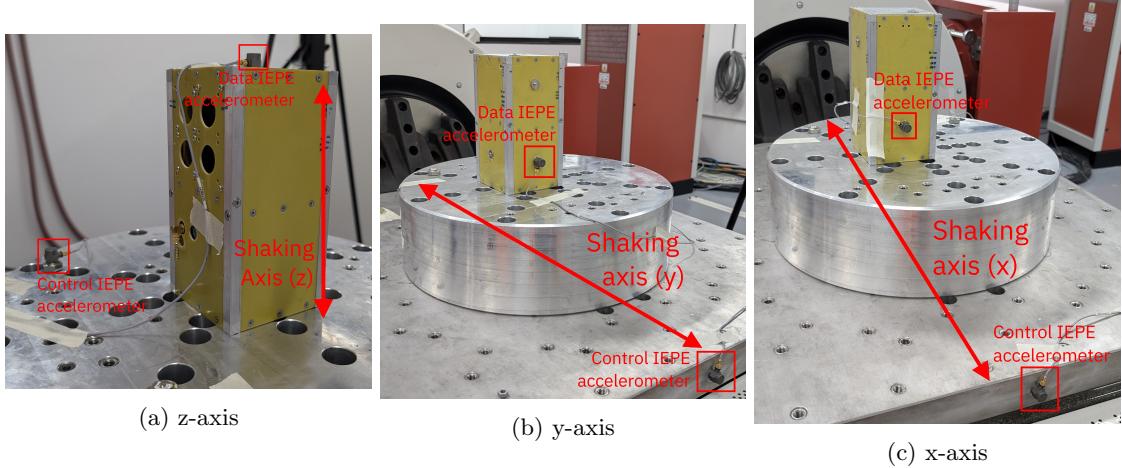


Figure 5.7: CubeSat setup in on the shaker table.

### 5.2.1 Random

The IIST recommended random vibration profile described in table 3.12 was used.

### 5.2.2 Quasi-static Acceleration Test

A QAT was not specified by IIST, therefore one was created which has a profile of  $18.9\ g$  from 30 Hz to 35 Hz. However, this test was not realisable with the type of shaker table after testing. Since IIST did not specify this test, it was excluded from testing. For later analysis, the boost phase would be compared to random vibration.

### 5.2.3 Shock

Due to limitations of the shaker table, the shock time had to be reduced from 10 ms to 8 ms, which produced a shock which would over-qualify the CubeSat compared to the IIST recommended profile. The profile is shown in figure 5.8

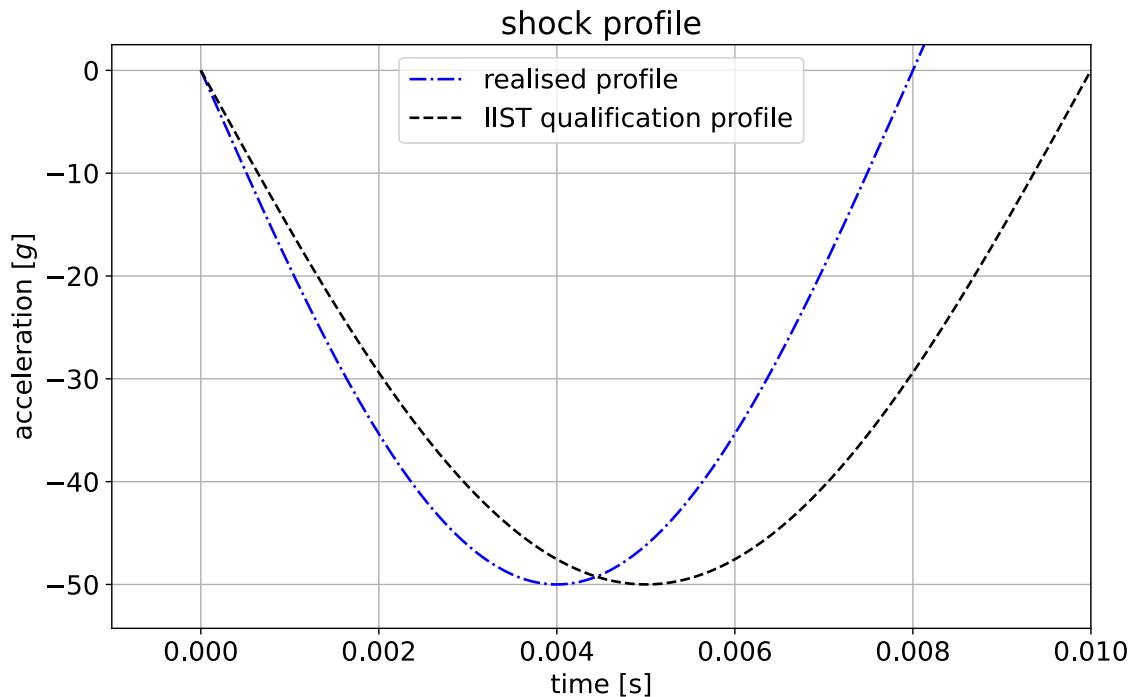


Figure 5.8: Realised shock profile.

## 5.3 Evaluation of HPR as a CubeSat Qualification Platform

Each part of flight is compared to one type of shaker table test as shown in table 5.2:

HPR flight	Shaker table	Representation
Boost/launch	Random vibration	PSD
Coast	Random vibration	PSD
Pyroshock/parachute deployment	Shock	SRS, Time domain

Table 5.2: Comparisons between the two tests and their representation.

Since the shaker table profiles are at or above the IIST recommended qualification level, for this research it will be assumed this was adequate to qualify for launch conditions.

HPR will be a successful qualification platform if:

- For PSD measurements the HPR PSD trace is equal to or greater than the PSD trace from the corresponding shaker table test.
- For shock, the HPR pyroshock events should produce a shock response which has a half-sine profile time which is less than the shaker table response, and a peak acceleration which is greater than the shaker table response.

# Chapter 6

# DAQ Evaluation and Discussion

## 6.1 Drone Tests

### 6.1.1 First Drone Test

The first drone test was performed on the July 8, 2024 at the UWA Shenton Park Field Station. At this stage in development, the first iteration of the design was partially completed. This was mounted to a 3d-printed mockup of the final CubeSat chassis, then tied to the drone, as shown in figure 6.1. In this configuration, the camera payload is in the top half of the CubeSats and the DAQ in the bottom half. The camera payload is mounted so that the camera is pointing towards the ground. A DJI M600 drone was used to raise the CubeSat to 120 m.

Since at this point in the design process it was evaluated that the DAQ needed a second new design, most of the systems were not tested, and the focus was on testing the camera payload. The DAQ's EPS was tested in this flight since it was needed for the camera to capture an image.

As a test of the DAQ's EPS, this test was a complete success since it was able to deliver power to the camera payload throughout the duration of the flight, therefore the EPS was used in the final DAQ design. While not relevant to the evaluation of this system, it should be noted that the camera payload did not capture any images due to a faulty connection between the Raspberry Pi 3 compute module and the camera module on the camera payload.

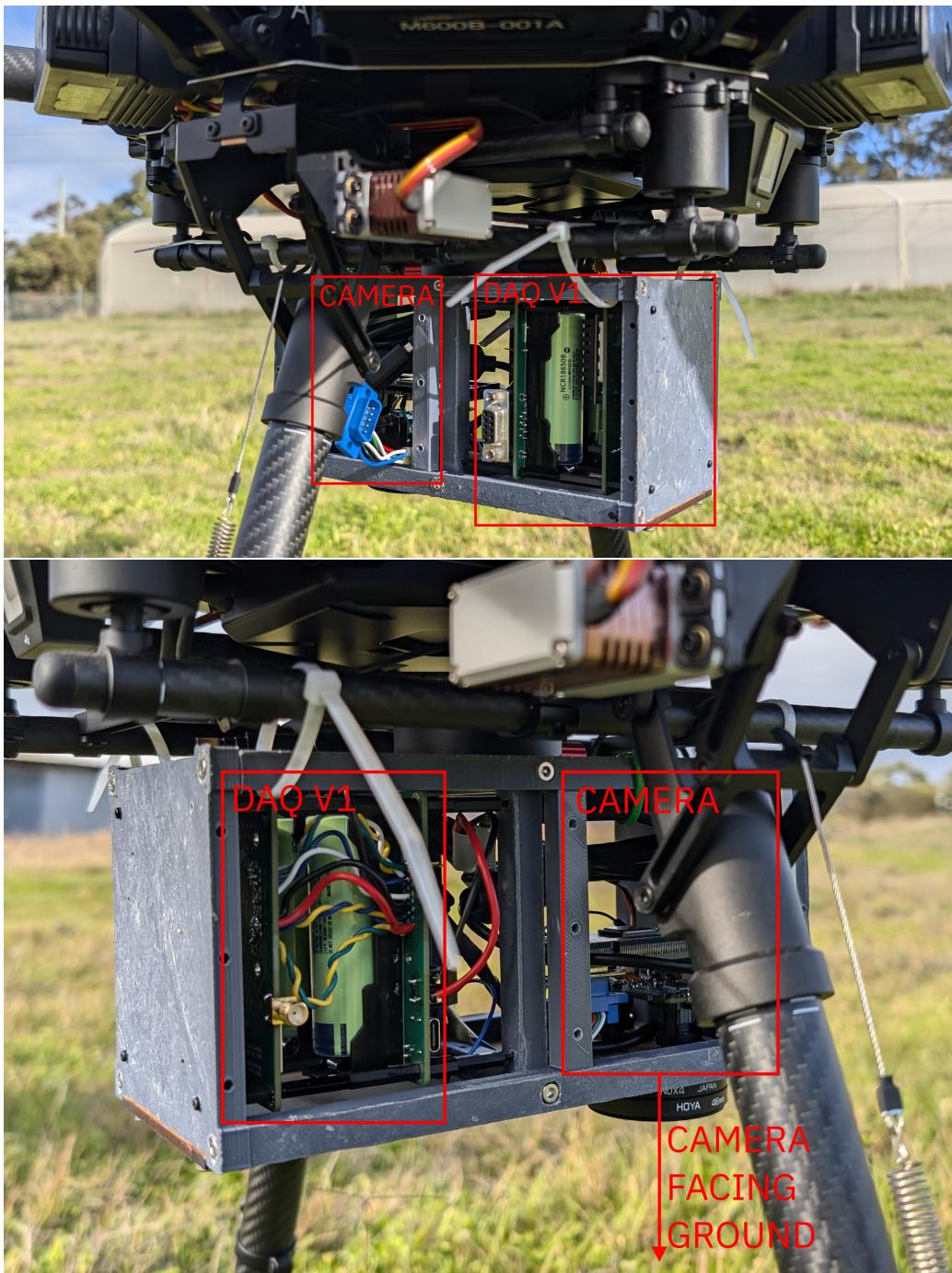


Figure 6.1: DJI M600 drone with CubeSat mounted sideways before flight.

### **6.1.2 Second Drone Test**

The second drone test was performed on the September 2, 2024 at the UWA Shenton Park Field Station. This drone test uses the final design of the DAQ, final metal CubeSat chassis and a modified camera payload. The CubeSat was mounted to the bottom of the DJI M600 drone in an identical orientation to the first drone test. The drone raised the CubeSat to 120 m.

As of this test, part of the final design was programmed, including the radio and CubeSat communications. A message is periodically sent to indicate that the OBDH system is alive.

Due to an issue on the camera payload, no picture data was saved to the OBDH's storage, nor was any picture data was received over the radio link. The periodic messages were received on the ground throughout the flight with the correct system uptime, indicating that the OBDH and radio downlink was stable throughout the flight.



Figure 6.2: Mounting the CubeSat to the DJI M600 drone for the second drone test.

### 6.1.3 Third Drone Test

The third drone test was performed on the September 12, 2024 at the UWA Shenton Park Field Station. All electronic and mechanical hardware was identical between the second test, the only changes were software. This test features the blocking image transmission method after validation of the method in ground testing. The camera payload had other software improvements, including a process watchdog to ensure the system restarts if it encounters an error, and better logging.

This test was a complete success, as two good pictures were captured from the camera system and downloaded to the ground station over the radio link.

A first image was captured at 120 m above some sheds on the property during the first flight as shown in figure 6.3. The drone was landed in between image captures since the download process

uses significant time, and keeping the drone in the air would use battery energy that would be useful for a second image.



Figure 6.3: Grayscale image of sheds taken at 120 m AGL. The red pixels are dropped bytes.

The second 120 m flight captured another image near roads on the edge of the property, as shown in figure 6.4. Figure 6.5 shows the ground station during an image download.

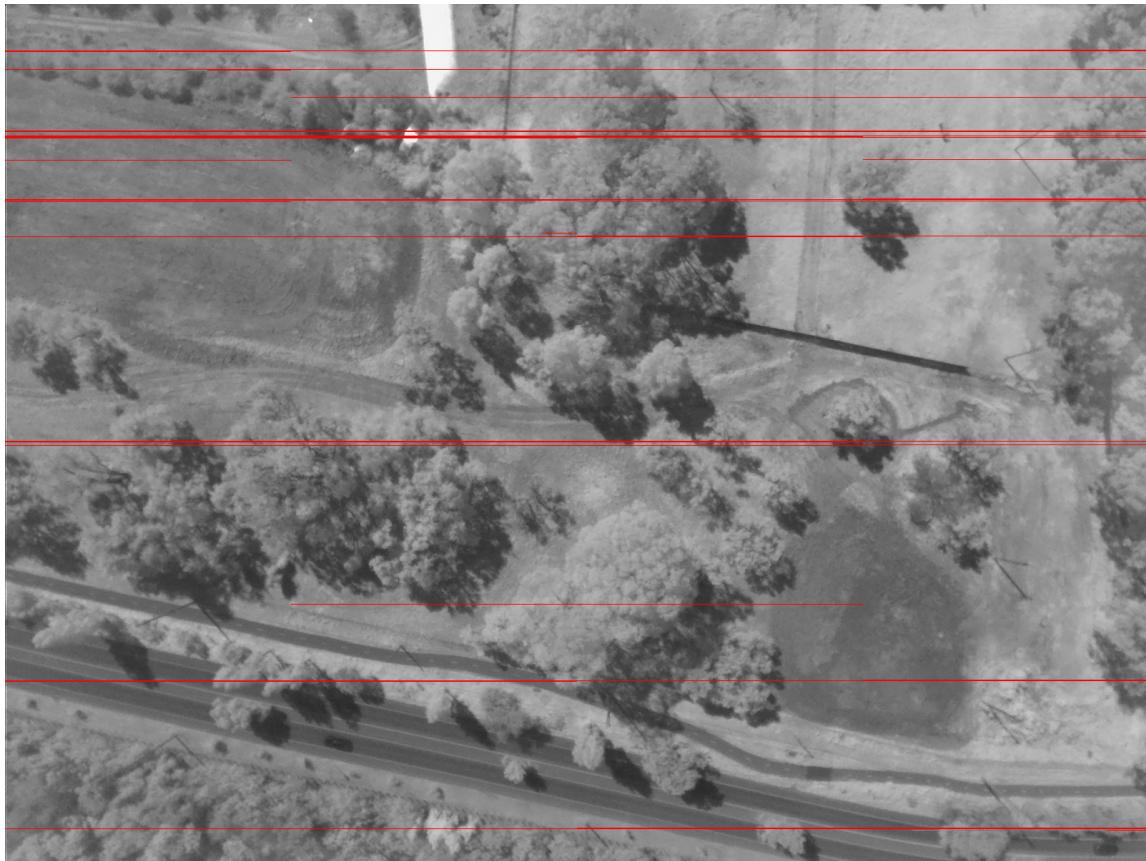


Figure 6.4: Grayscale image of trees and roads from 120 m AGL. The red pixels are dropped bytes.

The screenshot shows a software interface for managing a ground station. On the left, there is a code editor with Python scripts named `corrupt.py`, `process_images.py`, and `output_data_20.py`. The `process_images.py` script contains functions for stripping messages from a buffer and reading binary files. The `output_data_20.py` script appears to be a test script. On the right, there is a preview window showing a camera feed with a red rectangular overlay indicating missing data. Below the preview is a terminal window displaying log messages from the system, including uptime and transmission status. The terminal also shows the results of saving processed images as `image_0.png` and `image_naive_0.png`.

Figure 6.5: Screen capture of the ground station during live image download from the CubeSat. The terminal shows the periodic uptime messages which were sent to indicate that the system was online while the camera was capturing an image. A local Python script reassembles the image from the download stream. Red pixels have yet to be downloaded at this point.

At this distance, less than 3% of the image data was not downloaded as shown in table 6.1. In the current implementation, the error manifests as "strips" of lost image pixels since the image is transmitted sequentially.

Image	Lost bytes	Byte error rate
Sheds	45474	2.4%
Roads	31824	1.7%
Mean	38649	2.0%

Table 6.1: Lost bytes statistics for an image with 1920000 pixels.

#### 6.1.4 Discussion

The first two drone tests were only partial successes but were useful in validating parts of the final DAQ such as the EPS and downlink. However, in the future it would be useful to perform less drone

tests since the outcomes of the first two tests could have been achieved through ground testing.

The results of the final drone test were a success, showing the recovery of an image of acceptable quality from a drone at a height of 120 m. As shown in figure 6.4 and 6.3, the errors are mainly burst errors rather than single byte errors, since an error manifests as multiple sequential bytes being lost. Due to time constraints, it was only possible to implement the blocking image protocol. This protocol is very limited and does not feature state of the art forward error correction (FEC) techniques such as LDPC, turbo codes and BCH [60]. These coding schemes are used as part of standard satellite protocols including DVB-S2 and CCSDS file delivery protocol [60]. An interleaver changes the order of bits to more evenly distribute burst errors over a bit stream, which allows LDPC and FEC codes to handle burst errors [61]. Another novel technique would be to use generative adversarial networks on the received image [62], since burst errors result in similar banding errors which are caused by faulty satellite image sensors. The use of FEC would allow compression algorithms to be used, since compressed data is more sensitive to errors. JPEG-LS and CCSDS 123.0-B-2 are image compression algorithms used on imaging satellites [63].

The state of the art for CubeSat testing does not consider a drone test as part of the typical qualification campaign, however this successful result shows that a drone test is a useful test especially for imaging satellites. Typical CubeSat testing campaigns use ground services to emulate supporting equipment available on the launch vehicle, however for the drone test this was not possible due to the limited mass able to be carried. Designing the DAQ as a launch vehicle emulation platform is a useful improvement over the state of the art since it can be used in drone and rocket tests where the launch services need to be emulated, but large equipment is not able to be used.

## 6.2 Temperature Testing

### 6.2.1 High-Temperature Test

The high-temperature was performed as described in section 3.7.1 High-Temperature Test. Thermocouples were used to monitor the temperature of the fibreglass panels on the CubeSat frame, which was  $(80 \pm 2)^\circ\text{C}$  at steady-state. Since the tests were not performed at a specialised facility with Li-ionbattery firefighting equipment, the Li-ionbatteries were not used. A DC power supply was used to substitute the batteries, and was set to 3.7 V DC to emulate the average battery voltage.

The DAQ transmitted picture data from room temperature until 72 °C, at which point the camera system failed, and no images were captured. However, the DAQ was still functional to the 80 °C target temperature as periodic uptime messages were sent.



Figure 6.6: High-temperature testing setup

### 6.2.2 Low-Temperature Test

The low-temperature test was performed as described in section 3.7.1 Low-Temperature Test on September 12, 2024 using a consumer fridge. A thermocouple was placed on the ground plane of the GPS receiver. After 2 h, the batteries were clipped into the holder and the system was powered for half an hour. After the test the system was turned off and thawed while in the nitrogen bag, to prevent condensation.

The minimum temperature reached in the test was 0 °C, which was short of the –20 °C required for qualification. When powered on, the DAQ sent messages indicating the system was online and waiting for picture data from the camera payload. This picture data was not received from the camera payload. It was concluded after test that a USB switch on the camera payload was not rated for temperatures below 0 °C, resulting in no image data being saved on the camera payload. Following the test, the device was thawed and rebooted successfully. This indicates that the DAQ, excluding the communication link since this was not tested due to the camera issue, was functional to 0 °C.



Figure 6.7: Low-temperature testing setup

### 6.2.3 Discussion

These two temperature tests demonstrate that the DAQ can operate from 0 °C to 80 °C at atmospheric pressure. One of the major goals is to compare the vibration environment on a HPR launch to shaker table and the final space launch, which requires part of this system to be adapted for launch conditions and to be stable in orbit.

While these results were useful as a preliminary measure, it does not show that the DAQ is suitable for a space launch since:

- The lowest temperature reached was 0 °C, while the qualification target is –20 °C for the cold testing,
- The tests were performed at atmospheric pressure, meaning there was convection which would more effectively remove heat from hotspots compared to in a vacuum where only radiative heat transfer is present. It is possible that in a vacuum a hotspot, such as the processor of the Raspberry Pi, could heat up faster than heat can be radiated which could cause loss of performance or device failure.

Thermal vacuum testing would be required to qualify the DAQ for space operations.

## 6.3 High-Power Rocket Test Flight

*UNO* launched with the CubeSat on board on September 22, 2024 at 12:52pm. The launch and recovery was successful and the DAQ did capture accelerometer data and periodically broadcast uptime messages to the ground station. However, an issue is that the camera payload appears to

have booted in an invalid state, resulting in no image being captured or transmitted. An LED on the camera payload was found to be constantly on, which indicates power that is being received, however this LED is only meant to be monetarily on at the beginning of an image capture.

Good acceleration data was captured from the LSM6DSOX accelerometer as shown in figure 6.8, however the ADXL375 accelerometer did not produce good data therefore its data was not used in the final analysis. As shown in figure 6.9, the data from the ADXL375 is not correlated with any major flight events and is highly noisy. Due to time limitations the cause of the bad data has not been determined.

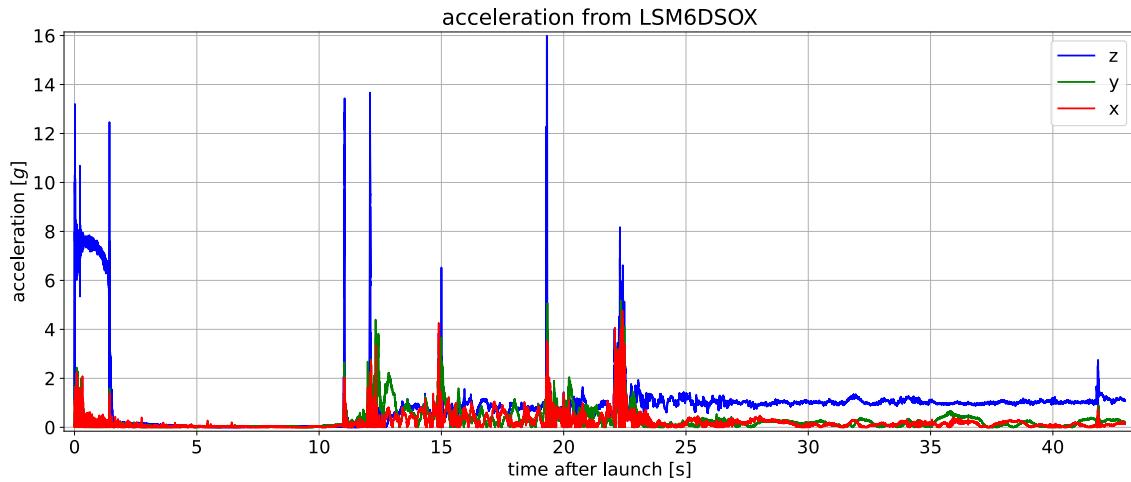


Figure 6.8: Time domain flight data from LSM6DSOX accelerometer.

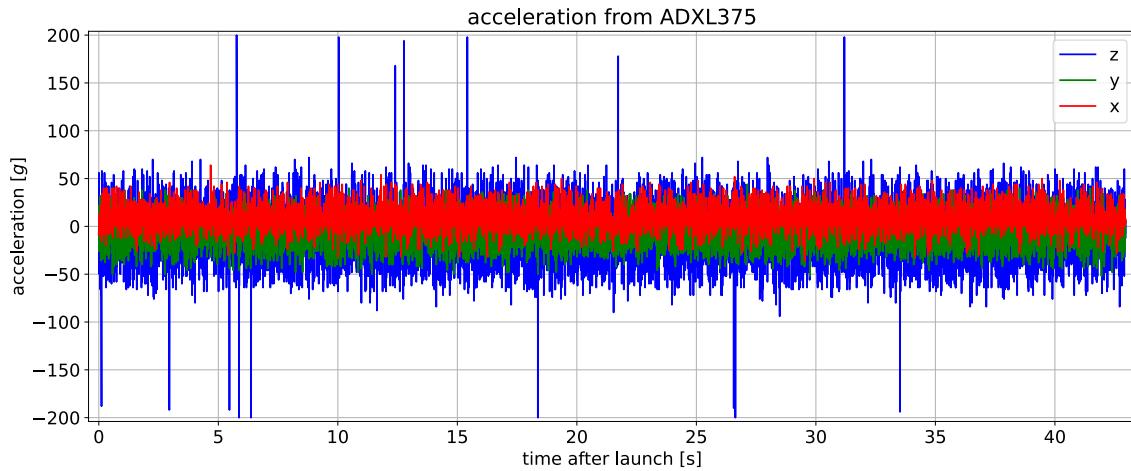


Figure 6.9: Noisy time domain flight data from ADXL375 accelerometer.

### **6.3.1 Discussion**

The issues with the camera were not simple to debug, since they worked during ground testing before the flight but refused to during and after the flight. This result indicates a need at the very minimum a dual redundant power supply in an OR-ing configuration, where both supplies are active at the same time and if one supply fails, the other automatically powers the payload due to an OR-ing ideal diode. The reliability of the POEM satellite bus is achieved through redundancy, therefore the emulation platform should ideally have some form of redundancy. This was removed from the scope of the final design iteration due to time and budget constraints, but clearly more effort should have been focused on it.

It is speculated in post flight discussion that the camera failure was caused by a race condition in the startup of the camera and Raspberry Pi on the camera payload's side. The payload should have its own watchdog to issues from causing complete failure, especially on the final design launched on POEM, however to be an effective testing platform the DAQ should have a watchdog feature which restarts the camera payload if it does not respond, since this test could have been used to test other parts of the camera system had the startup issue not occurred.

## **6.4 Shaker Table**

The shaker table tests were performed at AVI on the September 25, 2024 using the procedure described in section 5.2 Shaker Table.

### **6.4.1 Random**

The random vibration profile described in table 3.12 was successfully realised on the shaker table, resulting in a response as shown in figure 6.10. A result from the random vibration test was that it resulted in the DAQ momentarily not responding, then automatically rebooting after 6 minutes.

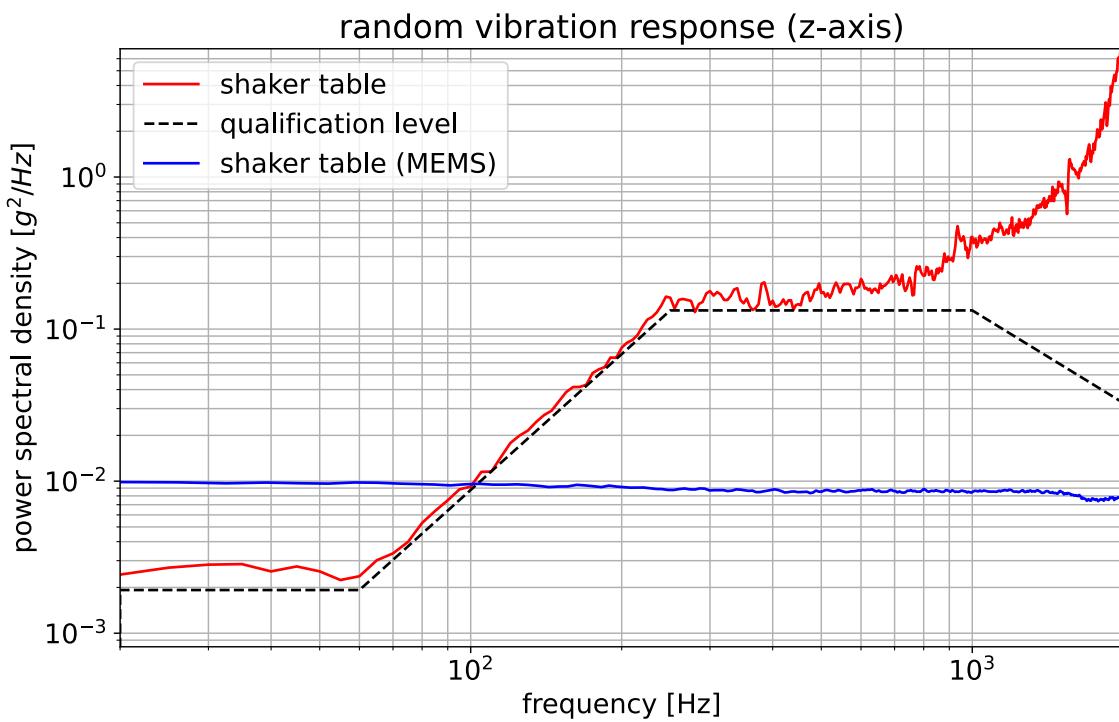


Figure 6.10: Response of payload from shaker table as recorded by DAQ and IEPE accelerometers.

A time-domain sample of the acceleration recorded by the DAQ is shown in figure 6.11.

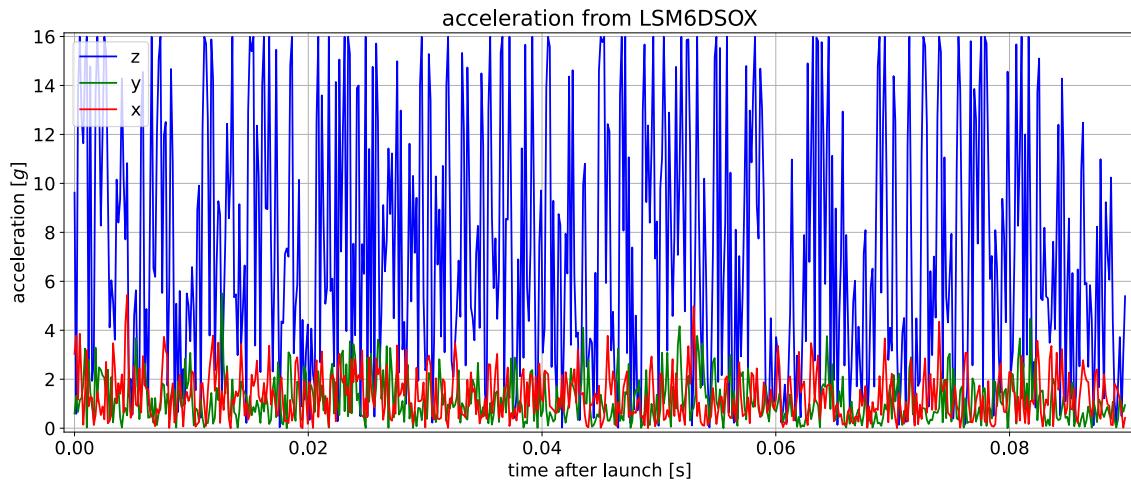


Figure 6.11: Noisy accelerometer from the DAQ.

#### 6.4.2 Shock

The shock produced the response as shown in figure 6.12.

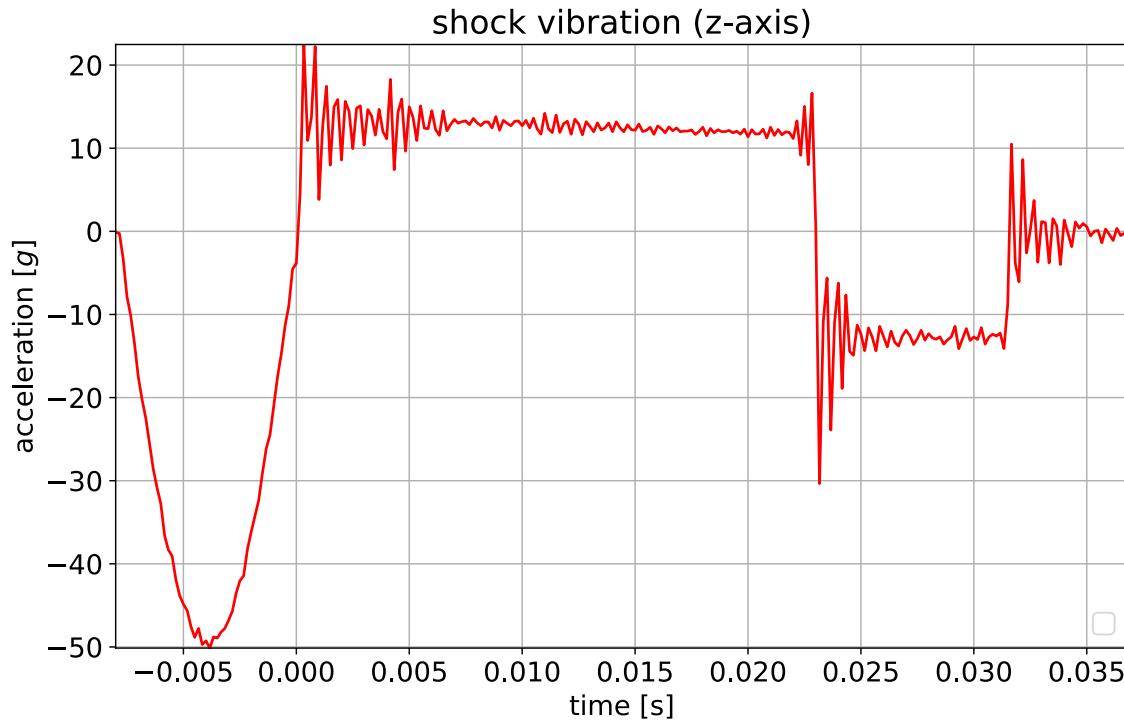


Figure 6.12: Response of 8 ms 50  $g$  half-sine shock. The two discontinuities in the shaker table response are a result of the table moving to ensure it stays within the maximum displacement limit.

A result of the shock test is one of the 18650 batteries becoming unmounted from its holder as shown in figure 6.13.



Figure 6.13: Battery becoming unmounted after shock testing

#### 6.4.3 Discussion

Typically, during shaker table tests on the ground, a CubeSat may be supported with external equipment when it is supported by services provided by the satellite bus. This test shows that the concept is viable as an alternative to external equipment, however the DAQ needs more work to protect it from high vibration environments, including spot welding batteries to the PCB instead of relying on leaf spring terminals. In this particular test, the batteries were not a problem since the two batteries on the sides were held by a cable tie. The reboot of the computer during the random vibration test likely indicates an issue with the quality of wiring harness or some loose element short-circuiting and causing issues.

As shown in figures 6.10 and 6.11, the MEMS accelerometers did not perform well at recording the response. The recorded PSD is almost white noise and was not useful for analysis. Therefore, for the rest of this analysis the IEPE accelerometer data will be used.

## **Chapter 7**

# **Evaluation of HPR Flight as a Qualification Platform**

This chapter will answer if a HPR launch is a viable qualification platform and will evaluate the success of the experiment design.

### **7.1 Random**

The coast and launch phases were compared to the random vibration shaker table test through a PSD, the results are shown in figure 7.1.

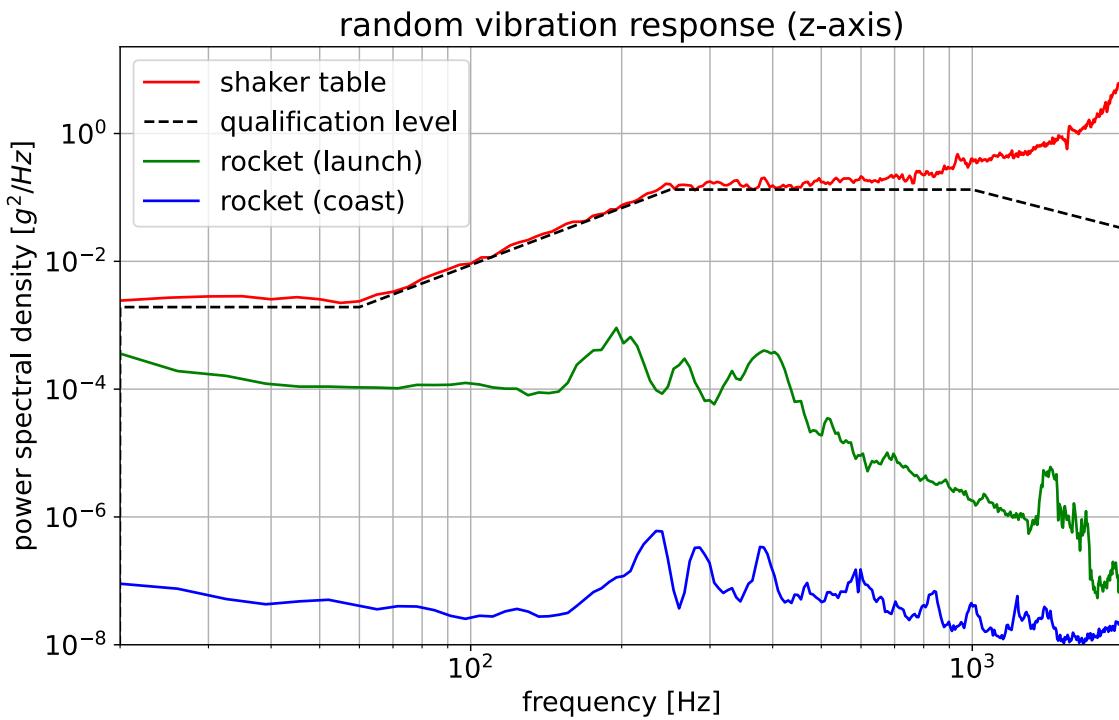


Figure 7.1: Comparison of shaker table and HPR responses. Shaker table response was captured using IEPE accelerometers, the HPR responses were captured using DAQ accelerometers.

## 7.2 Shock

The half-sine shock test on the shaker table was compared to one parachute deployment event, where a pyrotechnic shock occurred. The two shock responses are shown in figure 7.2.

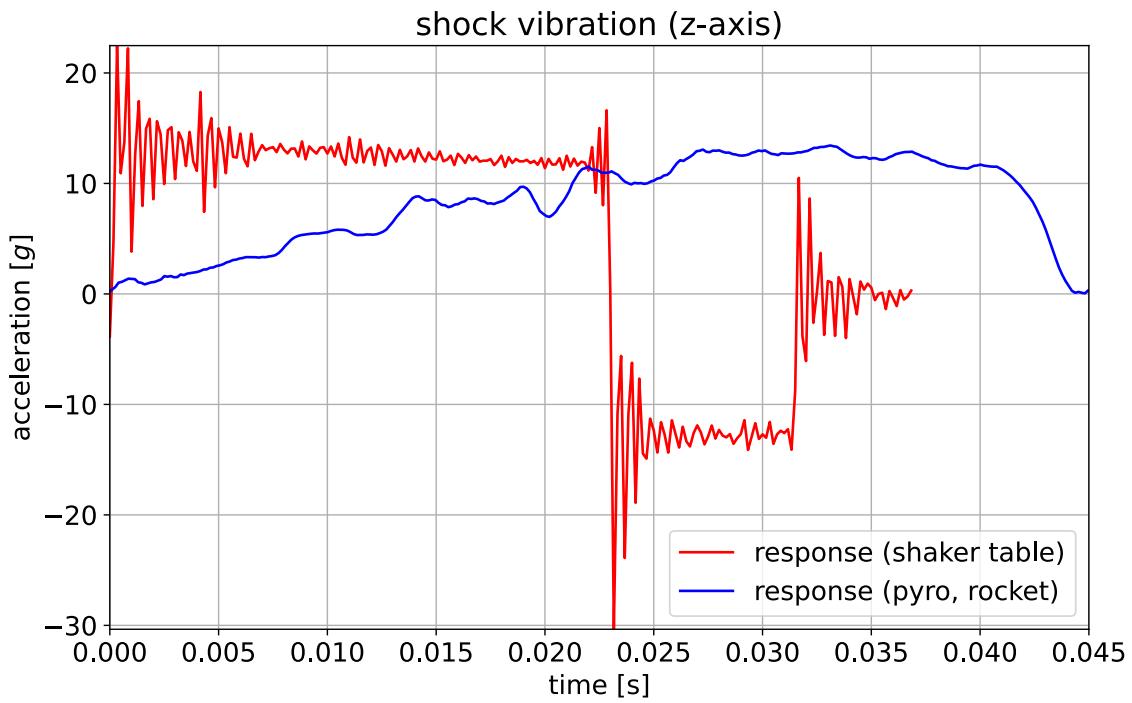


Figure 7.2: Shock response of pyroshock and half-sine profile. The two discontinuities in the shaker table response are a result of the table moving to ensure it stays within the maximum displacement limit.

The SRS was created using the time-domain data in figure 7.2, using a damping factor of  $Q = 10$ . As shown in 7.3, the SRS of the rocket consistently has a lower peak acceleration than the SRS of the half-sine shock for all frequencies between 10 Hz to 10 000 Hz

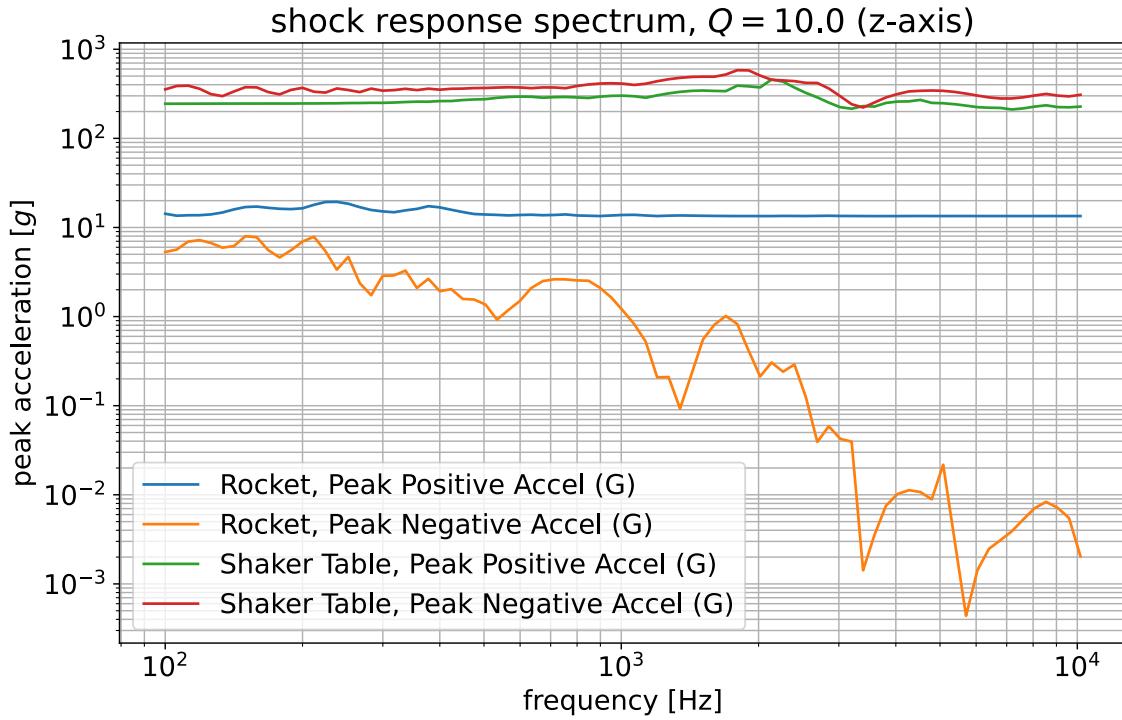


Figure 7.3: Shock response of pyroshock and half-sine shock.

### 7.3 Discussion

The experiment was successful in showing that HPR was not a suitable vibration qualification procedure for a CubeSat using a K1100T motor. The pyroshock events generated on the HPR flight had lower peak accelerations (Consistently 1 order of magnitude less) across the spectrum compared to the qualification response from the IIST profile. The random vibration response from both the launch and coast phases were consistently below both the shaker table response, and the qualification profile.

This experiment design was adequate for this case, since the goal was to answer a simple yes or no question about if HPRs are suitable for qualification. However, future tests may require a more comprehensive comparison since it is possible that with a different HPR motor or components that there are portions of the test which meet the qualification level. This type of comparison may include determining what frequencies the HPR test are effective at comparing, or using coherence to compare the PSDs numerically.

## Chapter 8

# Conclusions and Future Work

### 8.1 Future Work

While this design was successful in showing the inadequacy of HPR launches for qualification, another question is whether current shaker table tests are over-qualifying CubeSats. A useful extension of this work would be to put the DAQ on the PSLV POEM and measure the real vibration environment during launch. This would allow direct comparison between both HPR and shaker table testing to the real conditions.

However, to launch the DAQ on POEM a future design would need to be created which would incorporate all the learnings from this design including:

- Machine a heatsink to connect all chips on the Pi Zero module to the chassis or heatsink to ensure better thermal performance in vacuum [53].
- Batteries should be spot welded to tabs that are soldered fixed to the PCB to make pack resistant to random vibration rather than using the current leaf springs.
- Use IEPE accelerometers and integrate an analog front end for the accelerometers on the DAQ instead of using MEMS accelerometers, since this work has shown MEMS accelerometers to underperform for vibration analysis.
- Investigate the use of a real-time operating system instead of Raspbian.
- Implement a watchdog to increase reliability.

To qualify the future design, the testing campaign would need to be repeated. While the fridge and oven tests were found to be useful preliminary tests for partially qualifying the temperature range of components, a final test should be performed using a thermal vacuum chamber.

Other design improvements that would be useful for future HPR launches include:

- Use of error correcting codes such as LDPC codes with interleaving and compression to provide higher downlink speeds and reduced error rates, which would be good since drone and HPR tests are relatively short [61].

- Implement a watchdog to reset the payload-under-test if it fails, to get more value out of the tests.
- Use two independent power supplies in an OR-ing configuration to implement power supply redundancy.

This work was limited by motor sourcing issues, therefore the experiment should be repeated for more significant M impulse class motors, which have ten times the total impulse and may be able to provide the random vibration environment seen in shaker table testing.

The experiment developed in this work could only answer if HPR launches were useful for qualification or not, but it could not say to what degree they differ from shaker table testing (and potentially the real POEM launch). An experiment which quantifies the difference will have to be developed in the future.

## 8.2 Conclusions

This work developed a novel data acquisition and emulation platform for the POEM intended to be used in tests on a HPR or drone qualification platform and an experiment to evaluate the usefulness of HPR launches as a CubeSat qualification method. The DAQ was used to evaluate whether the HPR is a viable alternative to shaker tables as a vibration qualification platform. The vibration profile from the HPR launch, captured using the DAQ, was compared to the profile from the shaker table test. While the shaker table performed at, or above, the IIST recommended qualification level for POEM for all tests except the quasi-static acceleration test, the HPR launch produced a vibration profile that was consistently below the qualification level, therefore the conclusion is that HPR launches using a K1100T motor or lower are not a good vibration qualification platform. While the motor may be able to be changed in future flights to produce a stronger random vibration spectrum that may meet the qualification level, it is unlikely that HPR launches will ever be a substitute for shock testing since a larger launch does not use significantly larger black powder charges.

Since HPR testing was a novel qualification method, there were few designs on how a CubeSat should be supported during the launch. Traditionally, a satellite bus would be emulated using heavy ground equipment such as power supplies and computers. This work developed a novel test platform which emulated the final satellite bus the CubeSat will use, at specifications required for HPR flight. While HPR launches below a K1100T motor are not an ideal qualification platform, the drone tests were especially useful for qualifying the camera hardware. This emulation platform has a future in drone testing of CubeSats with optical payloads, since the mass restrictions of the HPR launch also apply to the drone.

## 8.3 Appendix

### 8.3.1 Code

Code and material from this research is available at the following git repositories:

- <https://git.petertanner.dev/peter/Honours-DAQ-software>
- <https://git.petertanner.dev/peter/Honours-DAQ-PCB>
- <https://git.petertanner.dev/peter/Honours-Data-Processing>
- <https://git.petertanner.dev/peter/Honours-Thesis>

### 8.3.2 References

- [1] R. P. Welle, “Overview of cubesat technology,” *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*, pp. 1–17, 2020.
- [2] J. Bouwmeester, A. Menicucci, and E. K. Gill, “Improving cubesat reliability: Subsystem redundancy or improved testing?” *Reliability Engineering & System Safety*, vol. 220, p. 108 288, 2022.
- [3] S. Gordon and D. L. Kern, “Benefits of spacecraft level vibration testing,” in *Aerospace Testing Seminar*, 2015.
- [4] L. K. Slongo, J. G. Reis, D. Gaiki, P. V. H. Seger, S. V. Martínez, B. V. B. Eiterer, T. G. Pereira, M. B. Neto, M. dos Santos Frata, H. D. Hamisch, *et al.*, “Pre-flight qualification test procedure for nanosatellites using sounding rockets,” *Acta Astronautica*, vol. 159, pp. 564–577, 2019.
- [5] T. Ludovico, “Targeting low earth orbit vegetation indexing for home grown sensing,” Master’s thesis, The University of Western Australia, 2024.
- [6] N. Nath and G. S. Aglietti, “Study the effect of tri-axis vibration testing over single-axis vibration testing on a satellite,” in *2022 IEEE Aerospace Conference (AERO)*, IEEE, 2022, pp. 1–10.
- [7] D. T. Pierce, “Development of a rocket test platform capable of delivering standard dimension payloads to near-space altitudes,” Ph.D. dissertation, Monterey, CA; Naval Postgraduate School, 2019.
- [8] M. G. Mariano, F. E. Morsch, M. S. Vega, D. M. A. M. Pio, S. L. Oriel, S. L. Kessler, and B. E. Augusto, “Qualification and validation test methodology of the open-source cubesat floripasat-i,” *Journal of Systems Engineering and Electronics*, vol. 31, no. 6, pp. 1230–1244, 2020. DOI: [10.23919/JSEE.2020.000103](https://doi.org/10.23919/JSEE.2020.000103).
- [9] G. Minelli, L. Magallanes, N. Weitz, D. Rigmaiden, J. Horning, J. Newman, M. Scott, S. Brady, C. Watkins, J. Christensen, *et al.*, “The mobile cubesat command and control (mc3) ground station network: An overview and look ahead,” 2019.
- [10] NASA Goddard Space Flight Center, *General Environmental Verification Standard (GEVS) for GSFC Flight Programs and Projects*, NASA Technical Standard, Document date: April 28, 2021., NASA Goddard Space Flight Center, Apr. 2021. [Online]. Available: <https://standards.nasa.gov/standard/GSFC/GSFC-STD-7000>.

- [11] M. Cho, H. Masui, T. Hatamura, S. Horii, S. Obata, *et al.*, “Overview of nano-satellite environmental tests standardization project: Test campaign and standard draft,” 2012.
- [12] C. D. Brown 1930, *Elements of spacecraft design*, English, ser. AIAA education series. Reston, VA: American Institute of Aeronautics and Astronautics, 2002, ISBN: 978-1-60086-051-5 1-60086-051-6 978-1-60086-179-6 1-60086-179-2. [Online]. Available: <https://doi.org/10.2514/4.861796>.
- [13] C. Nieto-Peroy and M. R. Emami, “Cubesat mission: From design to operation,” *Applied Sciences*, vol. 9, no. 15, p. 3110, 2019.
- [14] G. S. Aglietti, M. Remedia, M. Appolloni, and A. Kiley, “Spacecraft structure model validation and test philosophy,” *AIAA Journal*, vol. 57, no. 5, pp. 2109–2122, 2019.
- [15] W. Rawson and E. G. Lightsey, “Best practices and considerations for planning and conducting integration of university cubesats,”
- [16] Z. S. Decker, “A systems-engineering assessment of multiple cubesat build approaches,” Ph.D. dissertation, Massachusetts Institute of Technology, 2016.
- [17] J. Dickens, M. Wittbrodt, M. Gate, L. Li, and A Stroeve, “Coupled loads analysis accuracy from the space vehicle perspective,” *Acta Astronautica*, vol. 48, no. 1, pp. 21–28, 2001.
- [18] L. J. Bement and M. L. Schimmel, “A manual for pyrotechnic design, development and qualification,” Tech. Rep., 1995.
- [19] National Aeronautics and Space Administration, *Pyroshock test criteria*, NASA Technical Standard, National Aeronautics and Space Administration, Dec. 2011. [Online]. Available: {<https://s3vi.ndc.nasa.gov/ssri-kb/static/resources/NASA-STD-7003A.pdf>}.
- [20] C. Venturini, B. Braun, D. Hinkley, and G. Berg, “Improving mission success of cubesats,” 2018.
- [21] M. Doyle, R. Dunwoody, G. Finneran, D. Murphy, J. Reilly, J. Thompson, S. Walsh, J. Erkal, G. Fontanesi, J. Mangan, *et al.*, “Mission testing for improved reliability of cubesats,” in *International Conference on Space Optics—ICSO 2020*, SPIE, vol. 11852, 2021, pp. 2699–2718.
- [22] G. Seibert and B. T. Battrick, *The history of sounding rockets and their contribution to European space research*. ESA Publications division Noordwijk, 2006.
- [23] B. Pont, J. Beurskens, J Dalderup, P. Dolron, J Gubbels, J Horandel, R. Jordans, H. Pourshaghghi, D Szálas-Motesiczky, T. v. Vliet, *et al.*, “Rexus-25 rocket flight of a cubesat cosmic-ray detector,” 2019.
- [24] J. M. Jurist, “Commercial suborbital sounding rocket market: A role for reusable launch vehicles,” *Astropolitics*, vol. 7, no. 1, pp. 32–49, 2009.
- [25] M. Canepa, *Modern high-power rocketry*. Trafford Publishing, 2005, vol. 2.
- [26] J. Wang, X. Ren, X. Li, Y. Wen, L Cheng, and Q Guo, “Numerical simulation of the effect of combustion characteristics of main charges on the output shock of a typical igniter,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2478, 2023, p. 072024.
- [27] J. T. Telaak and W. De Backer, “Designing an arduino-based rocket flight computer for embedded systems education,” in *2023 Regional Student Conferences*, 2023, p. 71965.

- [28] Altus Metrum, *Telemetrum*, <https://altusmetrum.org/TeleMetrum/> (accessed Oct. 15, 2024), 2023.
- [29] P. Moschidis and P. S. Bithas, “Arduino rocket flight computer,” in *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*, IEEE, 2022, pp. 1–6.
- [30] Australian Youth Aerospace Association, *2024 aurc rocket specifications*, version Draft A, <https://aurc.ayaa.com/wp-content/uploads/2023/12/2024-AURC-Rocket-Specifications-Draft-A.pdf> (accessed Oct. 15, 2024), Dec. 2023.
- [31] Raspberry Pi Ltd, *Raspberry pi hardware - raspberry pi hardware*, <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (accessed Oct. 15, 2024), 2024.
- [32] RF Design, *Rfd900x and rfd868x radio modem datasheet*, <https://files.rfdesign.com.au/Files/documents/RFD900x%20DataSheet%20V1.2.pdf> (accessed Oct. 15, 2024), Dec. 2020.
- [33] u-blox, *Neo-m9n-00b - data sheet*, [https://content.u-blox.com/sites/default/files/NEO-M9N-00B\\_DataSheet\\_UBX-19014285.pdf](https://content.u-blox.com/sites/default/files/NEO-M9N-00B_DataSheet_UBX-19014285.pdf) (accessed Oct. 15, 2024), Mar. 2023.
- [34] STMicroelectronics, *LSM6DSO: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope*, DS12140 - Rev 2 - January 2019, STMicroelectronics, Jan. 2019. [Online]. Available: <https://www.st.com/resource/en/datasheet/lsm6dso.pdf>.
- [35] Analog Devices, *Adxl375 data sheet*, <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL375.PDF> (accessed Oct. 15, 2024), Apr. 2014.
- [36] MaxLinear, *Sp3485 data sheet*, <https://www.maxlinear.com/ds/sp3485.pdf> (accessed Oct. 15, 2024), Aug. 2021.
- [37] ——, *Xr20m1172 two channel i2c/spi uart with 64-byte fifo*, <https://www.maxlinear.com/ds/xr20m1172.pdf> (accessed Oct. 15, 2024), Feb. 2022.
- [38] Department of Infrastructure, Transport, Regional Development, Communications and the Arts, *Radiocommunications (low interference potential devices) class licence 2015*, <https://www.legislation.gov.au/F2015L01438/latest/text> (accessed Oct. 15, 2024), 2015.
- [39] A. Cratere, L. Gagliardi, G. A. Sanca, F. Golmar, and F. Dell’Olio, “On-board computer for cubesats: State-of-the-art and future trends,” *IEEE Access*, 2024.
- [40] N. Meirambekuly, B. A. Karibayev, T. A. Namazbayev, G.-G. A. E. Ibrayev, S. O. Orynbassar, S. A. Ivanovich, and A. A. Temirbayev, “A high gain deployable 1/s band conical helix antenna integrated with optical system for earth observation cubesats,” *IEEE Access*, vol. 11, pp. 23 097–23 106, 2023. DOI: 10.1109/ACCESS.2023.3253556.
- [41] P Falstad, “Falstad circuit simulator,” <https://falstad.com/circuit/circuitjs.html> (accessed Oct. 14 2024).
- [42] M. Giesselmann and V. Roy, “Modeling of power supplies for power modulators with ltspice,” *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 26, no. 2, pp. 508–514, 2019.
- [43] Analog Devices, *Ltspice help*, <https://ltwiki.org/files/LTspiceHelp.chm/html/SPICE.htm> (accessed Oct. 8, 2024), 2022.
- [44] P. Hamilton, G. Brist, G Barnes, and J. Schrader, “Humidity-dependent loss in pcb substrates,” *Printed Circuit Design and Manufacture*, vol. 24, no. 6, p. 30, 2007.

- [45] V. Knap, L. K. Vestergaard, and D.-I. Stroe, “A review of battery technology in cubesats and small satellite solutions,” *Energies*, vol. 13, no. 16, p. 4097, 2020.
- [46] A. D. Pathak, S. Saha, V. K. Bharti, M. M. Gaikwad, and C. S. Sharma, “A review on battery technology for space application,” *Journal of Energy Storage*, vol. 61, p. 106792, 2023.
- [47] F. Krause, J.-P. Ruiz, S. Jones, E. Brandon, E. Darcy, C. Iannello, and R. Bugga, “Performance of commercial li-ion cells for future nasa missions and aerospace applications,” *Journal of The Electrochemical Society*, vol. 168, no. 4, p. 040504, 2021.
- [48] G. M. Marcelino, E. Morsch Filho, S. V. Martinez, L. O. Seman, and E. A. Bezerra, “In-orbit preliminary results from the open-source educational nanosatellite floripasat-i,” *Acta Astronautica*, vol. 188, pp. 64–80, 2021.
- [49] Samsung SDI Co., Ltd., *Specification of product: Lithium-ion rechargeable cell for power tools (model: Inr18650-25r)*, version 1.0, Mar. 2014.
- [50] Texas Instruments, *Tps61022 8-a boost converter with 0.5-v ultra-low input voltage*, D, <https://www.ti.com/lit/ds/symlink/tps61022.pdf> (accessed Oct. 15, 2024), Jul. 2021.
- [51] Advanced Monolithic Systems, Inc., *Ams1117 1a low dropout voltage regulator*, <http://www.advanced-monolithic.com/pdf/ds1117.pdf> (accessed Oct. 15, 2024), 2007.
- [52] E. Upton and G. Hafacree, *Raspberry Pi user guide*. John Wiley & Sons, 2016.
- [53] S. M. Guertin, “Raspberry pis for space guideline,” *NASA: Washington, DC, USA*, 2022.
- [54] STMicroelectronics, *Lsm6dsox-pid*, <https://github.com/STMicroelectronics/lsm6dsox-pid> (accessed Oct. 15, 2024), 2024.
- [55] Grazer Computer Club, *Wiringpi library*, <https://github.com/WiringPi/WiringPi> (accessed Oct. 15, 2024), 2024.
- [56] National Fire Protection Association, *NFPA 1127 Code for High Power Rocketry*, 2018 edition. NFPA, 2018.
- [57] Sampo Niskanen and others, *Openrocket simulator*, <https://openrocket.info/index.html> (accessed Oct. 10, 2024), 2024.
- [58] S. Niskanen, “Development of an open source model rocket simulation software,” Master’s thesis, Aalto University, Espoo, May 2009. [Online]. Available: [https://github.com/openrocket/openrocket/releases/download/Development\\_of\\_an\\_Open\\_Source\\_model\\_rocket\\_simulation-thesis-v20090520/Development\\_of\\_an\\_Open\\_Source\\_model\\_rocket\\_simulation-thesis-v20090520.pdf](https://github.com/openrocket/openrocket/releases/download/Development_of_an_Open_Source_model_rocket_simulation-thesis-v20090520/Development_of_an_Open_Source_model_rocket_simulation-thesis-v20090520.pdf).
- [59] W. Brown, M. Wiesneth, T. Faust, N. Huynh, C. Montalvo, K. Lino, and A. Tindell, “Measured and simulated analysis of a model rocket,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 4, pp. 1397–1411, 2019. DOI: [10.1177/0954410017752730](https://doi.org/10.1177/0954410017752730). eprint: <https://doi.org/10.1177/0954410017752730>. [Online]. Available: <https://doi.org/10.1177/0954410017752730>.
- [60] A. Zeedan and T. Khattab, “Cubesat communication subsystems: A review of on-board transceiver architectures, protocols, and performance,” *IEEE Access*, vol. 11, pp. 88161–88183, 2023. DOI: [10.1109/ACCESS.2023.3304419](https://doi.org/10.1109/ACCESS.2023.3304419).
- [61] Sonali, N. Gupta, A. Dixit, and V. K. Jain, “Capacity and ber analysis of bch and ldpc coded fso communication system for different channel conditions,” *optical and quantum electronics*, vol. 53, pp. 1–25, 2021.

- [62] Z. L. Paola, L. S. Jesús, A. H. Christian, and R. U. Sonia, “Correction of banding errors in satellite images with generative adversarial networks (gan),” *IEEE Access*, vol. 11, pp. 51 960–51 970, 2023. DOI: 10.1109/ACCESS.2023.3279265.
- [63] M. Hernández-Cabronero, A. B. Kiely, M. Klimesh, I. Blanes, J. Ligo, E. Magli, and J. Serra-Sagristà, “The ccstds 123.0-b-2 “low-complexity lossless and near-lossless multispectral and hyperspectral image compression” standard: A comprehensive review,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 102–119, 2021. DOI: 10.1109/MGRS.2020.3048443.