A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Christopher Gerdom

December 2018
COMMITTEE MEMBERSHIP

TITLE: Nanosatellite Launch Data-logger (Sync)

AUTHOR: Christopher Gerdom

DATE SUBMITTED: December 2018

COMMITTEE CHAIR: Andrew Danowitz, Ph.D.
Assistant Professor of Electrical Engineering

COMMITTEE MEMBER: John Bellardo, Ph.D.
Associate Professor of Computer Science

COMMITTEE MEMBER: Joseph Callenes-Sloan, Ph.D.
Assistant Professor of Electrical Engineering
ABSTRACT

Nanosatellite Launch Data-logger (Sync)

Christopher Gerdom

CubeSat designers are increasingly looking to incorporate delicate structures and optics into their payloads. These delicate payloads, however, may not survive the required absolute-worst-case launch vibration testing needed for flight certification. To help address this problem, and to better match testing conditions to real-world launch environments, this thesis introduces Sync, a compact 1/4U CubeSat payload designed to collect data on the vibrations and thermal environments CubeSats experience inside a deployer on the way to orbit. This data can be used to better understand the launch environment for different vehicles, and help develop new, more realistic testing guidelines that could enable more delicate payloads to be launched.
ACKNOWLEDGMENTS

Thanks to:

- My parents, for their love and support.
- Justin Sherrell, Justin Nguyen, Max Selna, and Junior Ramirez for help with mechanical design and data visualization.
- My panel members Dr John Bellardo, Dr Andrew Danowitz, and Dr Joseph Callenes-Sloan for their advice and feedback on this thesis.
# TABLE OF CONTENTS

| LIST OF TABLES | x |
| LIST OF FIGURES | xi |

## CHAPTER

1. **Introduction** | 1
   1.1 **CubeSat Payload Background** | 1

2. **Previous Works** | 3
   2.1 **LEO** | 3
      2.1.1 **Mission Requirements** | 5
         2.1.1.1 *Accelerometer Requirements* | 6
         2.1.1.2 *LEO Thermocouple Requirements* | 7
         2.1.1.3 *Timing and Recording* | 7
         2.1.1.4 *Launch Abort* | 8
      2.1.2 **Lessons Learned** | 8
         2.1.2.1 *LEO Trigger Testing Problems* | 9
         2.1.2.2 *Accelerometer Saturation* | 9
         2.1.2.3 *Gswitch Orientation* | 10
   2.2 **Alaska Research Cubesat 1 (ARC-1)** | 10
      2.2.1 **ARC-1 Mission Requirements** | 11
      2.2.2 **ARC-1 Mission Requirements Justification and Fulfillment** | 12
         2.2.2.1 *Launch Detection* | 12
         2.2.2.2 *Vibrations Requirement* | 12
         2.2.2.3 *Temperature Requirement* | 13
         2.2.2.4 *Time Requirement* | 13
   2.3 **Previous Work Conclusions** | 13

3. **System Design** | 15
   3.1 **Sync Mission Goals** | 15
   3.2 **Mission Requirements** | 16
      3.2.1 **GEVS Qualification and Accelerometers 25,100, and 200G** | 17
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Overview of possible microcontroller/microprocessors for the Sync payload</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Memory availability and capacity generated from in stock selection at DigiKey at time of writing</td>
<td>35</td>
</tr>
<tr>
<td>3.3</td>
<td>Non volatile memory with a SPI interface available at DigiKey at the time of writing</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Estimated standby current draw</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Estimated active current draw</td>
<td>43</td>
</tr>
<tr>
<td>5.1</td>
<td>Predicted vs Actual current</td>
<td>75</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.1</td>
<td>Dimensionless CAD model of LEO, internal components visible.</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>The ARC-1 data logger payload[8]</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>The working block diagram of the system</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>NASA GEVs profile for spacecraft vibrations testing used to guide Cal Poly vibrations tests.</td>
<td>20</td>
</tr>
<tr>
<td>3.3</td>
<td>Threshold setting behavior for GSwitch interrupt</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>the filter shape of the anti-aliasing filter. Component tolerance effect indicated by shadow</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>The general schematic design of one anti-aliasing filter</td>
<td>31</td>
</tr>
<tr>
<td>3.6</td>
<td>Two images of the SD card cage. The left image shows the gold spring contacts exposed without the SD card. The right image shows a micro SD card in the card cage.</td>
<td>35</td>
</tr>
<tr>
<td>3.7</td>
<td>A rendering of the Sync backing plate with half of the test structure hidden.</td>
<td>38</td>
</tr>
<tr>
<td>3.8</td>
<td>The backing plate to keep the board rigid</td>
<td>39</td>
</tr>
<tr>
<td>3.9</td>
<td>Exaggerated displacement visualization from FEA report, credit Gerardo Ramirez</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>the schematic layout of the always on digital regulator and the switched analog regulator</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>NAND flash chip array with demultiplexing chip</td>
<td>48</td>
</tr>
<tr>
<td>4.3</td>
<td>the SD card cage schematic as implemented</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>Accelerometer and anti-aliasing filter arrangement.</td>
<td>49</td>
</tr>
<tr>
<td>4.5</td>
<td>the internal elements of the anti-aliasing/low-pass blocks. Note that two filters are present in this design.</td>
<td>50</td>
</tr>
<tr>
<td>4.6</td>
<td>The schematic connection of the MSP430FR5739 microcontroller.</td>
<td>50</td>
</tr>
<tr>
<td>4.7</td>
<td>Board stack-up showing 4 conducting layers and supporting nonconducting layers</td>
<td>52</td>
</tr>
<tr>
<td>4.8</td>
<td>Negative Z board side with analog area shaded in red</td>
<td>53</td>
</tr>
<tr>
<td>4.9</td>
<td>The code flow of the Sync payload board. Interrupts not shown.</td>
<td>55</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.1</td>
<td>The Sync payload inside a 1U TestPOD on the Cal Poly vibes table.</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>NASA GEVS profile for spacecraft vibrations testing used to guide Cal Poly vibrations tests.</td>
<td>67</td>
</tr>
<tr>
<td>5.3</td>
<td>Peak hold of the power spectral density created from 1 second portions of the random x axis vibrations test</td>
<td>68</td>
</tr>
<tr>
<td>5.4</td>
<td>The control sensor’s Y-axis data which corresponds to the payload sensors X-Axis data.</td>
<td>69</td>
</tr>
<tr>
<td>5.5</td>
<td>Plot of the sampled signal of the x axis channel for the three accelerometers. Note the clipping in the 25g signal.</td>
<td>70</td>
</tr>
<tr>
<td>5.6</td>
<td>The location of the three-axis control accelerometer on the opposite side of the Sync payload plate</td>
<td>71</td>
</tr>
<tr>
<td>5.7</td>
<td>Breakdown of signal from debug pin during data rate test</td>
<td>73</td>
</tr>
<tr>
<td>5.8</td>
<td>Captured waveform from the debug pin during data rate test</td>
<td>73</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

The Sync payload is a compact launch dynamics data logger. Sync records vibrations and temperature within the CubeSat deployer while the rocket is launching. The data from Sync gives insight into the environment within the CubeSat deployer on launch which may help more advanced and delicate CubeSat payloads survive future launches. The compact form factor of the Sync payload (9x9x2cm) allows it to fit into spare space within CubeSats as small as 1U(approximately 10x10x10cm). The mission life of a launch dynamics data logger is short, 90 min plus data transfer time, compared to the time on orbit, 1–24 years. The short mission time makes Sync more suitable to secondary payload status rather than designing an entire CubeSat around recording this data.

1.1 CubeSat Payload Background

CubeSats are nanosatellites (nanosats) that conform to the CubeSat standard [3]. The smallest CubeSats are 10x10x10cm or 1u. CubeSats range from 1u to 3u or 10x10x30cm in 1u increments. CubeSats larger then 3u exist but are much less common than 1–3u CubeSats. CubeSats typically launch as secondary payloads on rockets that launch much larger satellites. The rocket carries CubeSat deployers that will launch the CubeSat from the rocket on orbit. CubeSats share rides to space and must conform to safety standards that keep the rocket and main payload safe. The CubeSat standards involve environmental testing, limitations on power and radio communications, and limits to hazardous materials[3]. A CubeSat goes through vibrations testing, thermal vacuum testing, and other safety tests before it is attached
to a rocket. Once in orbit CubeSats typically perform some science experiment and communicate the data back to Earth. CubeSat payloads are the instruments that perform these science experiments. The CubeSat may orbit for months to years collecting data and sending it to the ground. CubeSats may be in orbit for multiple years. Missions that have a longer operational period are often seen as having more merit than those that finish shortly after deploying. Many CubeSats perform more than one experiment on orbit to take advantage of the time they are in space. [16]
Chapter 2

PREVIOUS WORKS

Sync is not the first platform designed to record the environment inside a CubeSat deployer. Two other CubeSats, LEO and ARC-1, are designed to record the environment within the CubeSat deployer on launch. The Launch Environment Observer, LEO, is a Cal Poly designed 2u CubeSat that has not launched at the time of writing. ARC-1 is a 1u CubeSat designed by The University of Alaska Fairbanks which has launched but failed to turn on[11]. Unlike LEO and ARC-1, Sync is designed as a CubeSat payload allowing it to be integrated into other CubeSats. LEO and the requirements used for its mission serve as the starting point for the Sync Payload design. Due to a lack of information at the time of design, ARC-1 did not influence the Sync Platform. It is included here as an alternative implementation of a launch dynamics data logger.

2.1 LEO

The Launch Environment Observer (LEO) is a 2U (10x10x20cm) CubeSat that records launch environment dynamics inside the CubeSat deployer while the rocket travels to space. The mission records the vibration, mechanical shock, and temperature experienced by the CubeSat within the deployer. LEO turns on when a launch starts and records environment data from liftoff to deployment +60 seconds. The system records data independently of the rocket and does not have a telemetry feed to the ground during the launch. LEO is manifested on the SpaceX Falcon Heavy. As of publication, LEO has undergone environmental testing but has not launched. The results of the environmental testing are the focus of section 2.1.2 as they influence
some of the design parameters for Sync. LEO has two three-axis accelerometers with different sensitivities, and a thermocouple. Two different accelerometer sensitivities detect different events. A low sensitivity accelerometer measures high acceleration experienced during mechanical shock events. Mechanical shock events include impacts or the detonation of ballistic bolts. A high sensitivity accelerometer measures low acceleration from vibrations due to the rocket engines and aerodynamics. LEO uses a thermocouple to monitor the structure temperature on the way to orbit.

LEO uses two Cal Poly Intrepid Boards computers to process data and control the CubeSat. The Intrepid Board is the standard flight computer used by Cal Poly CubeSats. One Intrepid Board functions as the flight computer and the second board functions as a payload processing board. The primary board runs a custom Linux operating system while the payload board uses a bare runtime instead of an operating system. The payload board responds to the launch and processes samples from the
sensors. The payload board does not run an operating system reducing the time to boot. Without an operating system, the payload board boots much closer to launch.

2.1.1 Mission Requirements

The mission requirements for LEO focus on the sampling rates and sensitivities of the sensors. Development of these mission requirements was completed in conjunction with NASA Launch Services Program (LSP) [15] and the sensor requirements are set to capture the expected range of frequencies from the NASA General Environmental Verification Standard (NASA GEVS) [22]. The following mission requirements come directly from internal documentation for LEO. The requirements are a combination of derived and fundamental requirements. The requirements are listed here and expanded upon later.

1. Each accelerometer shall have a minimum usable frequency response of 20–2,000 Hz, a minimum data acquisition rate of 4,000 Hz, minimum 8-bit resolution, with 1 three-axis accelerometer having a usable range of $\pm 100G$ and the second three-axis accelerometer having a usable range of $\pm 10G$.

2. LEO shall operate and record a wired thermocouple within the 2U CubeSat with a minimum range of 0–71$^\circ$C $\pm 1^\circ$, with a minimum sample rate of 25 Hz

3. LEO shall record around events of interest in the period of time from as close to ignition as possible and to 60 seconds after CubeSat deployment.

4. LEO shall be capable of 10 minutes of total data acquisition time

5. LEO shall record continuously from T-0 through fairing separation

6. LEO shall partition remaining recording time to capture subsequent engine shutdown/startup events and CubeSat Deployment.
7. LEO shall be capable of terminating data generation if a launch abort is detected.

2.1.1.1 Accelerometer Requirements

LEO records frequencies in the range used by the NASA GEVS vibrations standard [22]. NASA GEVS defines a frequency range between 20 and 2000Hz. To properly record vibrations from the GEVS test, the sensors must be able to respond to vibrations with spectral components in the 20–2000Hz. The Nyquist rate of the system is 4000Hz. The sensors used in LEO have a 6kHz bandwidth and sensitivities of ±25g and ±200g. The frequency response range of the sensors is from 2–6000Hz. LEO samples with an external 12 bit analog to digital converter at 140kHz. The LEO mission requirements call for ±10G and ±100G accelerometers with an 8-bit sample. The mission specification for LEO define higher sensitivity accelerometers sampled with a lower sensitivity analog to digital converter. Equations 2.1–?? show that the designed sensitivity of LEO exceeds the original LEO mission requirements.

\[
system
text{sensitivity}[G/bit] = \frac{ADC
text{sensitivity}[mV/bit]}{accelerometer
text{sensitivity}[mV/G]} \tag{2.1}
\]

The original LEO mission requirements call for 0.8G/bit and 0.08G/bit for the low and high sensitivity accelerometers. LEO as implemented has 0.1G/bit and 0.01G/bit sensitivities.

Equation 2.1 defines the system sensitivity to acceleration leading to 0.8G/bit and 0.078G/bit that LEO exceeds with a sensitivity of 0.1G/bit and 0.012G/bit. LEO is able to sense vibrations in the original 10G range as well as vibrations with up to 25G amplitude.
2.1.1.2 LEO Thermocouple Requirements

LEO uses a k-type thermocouple[19] soldered to the sensor board. The thermocouple connects to the structure to increase the thermal conductance to the probe. Limiting board to structure contact area thermally isolates the board from the structure. In vacuum heat transfers between objects by radiation and conduction. Conduction between connecting surfaces is more effective at transferring heat than radiation. The structure directly contacts the deployer while the sensor board does not. Attaching the probe remotely on the structure reflects the temperature of the deployer better. The Sync payload uses the same method of thermocouple mounting.

2.1.1.3 Timing and Recording

The mission specifications state that LEO must record as close as possible to the start of the launch. The specifications include events of interest including liftoff, maximum aerodynamic pressure (Max Q), stage separation, and payload deployment. These events encompass major changes to the CubeSat environment and are generally where the worst environmental conditions are experienced. LEO records data at events of interest using the expected time of those events. LEO does not record the full launch.

LEO uses a mechanical GSswitch to trigger as close to launch as possible. The mechanical GSswitch is an acceleration sensitive component that enables the CubeSat when it senses acceleration in a set direction. A mechanical GSswitch triggers when the vertical acceleration of the rocket pushes two electrical contacts together. LEO’s design relies on a calibrated mass-spring system to close an electrical circuit and turn on the data logger.
2.1.1.4 Launch Abort

Aborting from false launch detections allows LEO to record the launch. Recording data from false triggers will consume power and memory, potentially preventing the recording of the actual launch. The CubeSat handles launch detection and abort by using a barometer with the mechanical GSwitch. Lower pressure after a launch detection indicates that the detection is legitimate and the data is valid. If the pressure does not drop then LEO stops recording. The abort and re-trigger method mitigates false triggers caused by handling of the CubeSat and conserves power.

2.1.2 Lessons Learned

Testing LEO has revealed problems ranging in severity from inconvenient to mission critical. These problems were identified at the system level testing stage. They are still known issues, multiple of which are not fixable without substantial changes to LEO’s system design. These problems are as follows:

1. Testing the trigger mechanism involves a cumbersome process of shaking the spacecraft and placing it in a vacuum.

2. The higher sensitivity accelerometer experiences clipping while testing. The clipping removes information on the vibrations experienced by the CubeSat and cannot be recovered.

3. Proper placement of the trigger mechanism requires knowledge of the orientation of the CubeSat on the rocket.
2.1.2.1 LEO Trigger Testing Problems

The process for testing LEO involves one technician shaking the spacecraft and then placing it into a bag. The bag is sealed, and a vacuum is provided by a shop vacuum. The deployment testing is a risk to the CubeSat. The technician who is holding the CubeSat must do so by the rails and not any of the faces as the CubeSat has solar cells on five of six sides. With limited grip, there is a greater risk that a cell might be broken or the CubeSat might be dropped. Since the pressure must drop within 10 seconds of the trigger event. Testing the trigger mechanism typically involves multiple retests.

2.1.2.2 Accelerometer Saturation

The higher sensitivity $\pm25g$ accelerometer saturates while on the shake table. When the sensor saturates it distorts the amplitude and frequency components of the vibration signal. Saturation of the higher sensitivity accelerometer prevents proper recording of the vibrations experienced by the spacecraft, removing information about vibrations in the launch environment. The low sensitivity accelerometer does not provide enough resolution to properly capture the vibrations. The NASA GEVS standard for qualification testing of satellites under 50Kg has an amplitude of $14.1G_{RMS}$ the expected peak to peak amplitude assuming white noise driving is $6 \times 14.1G_{RMS}$ or $84.6G_{pp}$ requiring a 42.3G accelerometer to sense 99.7% of the signal. The 200G is well above the required 42.3G accelerometer and will lose lower amplitude information in the noise of the signal.
2.1.2.3 Gswitch Orientation

The mechanical Gswitch senses acceleration along one axis. The mechanical Gswitch must align with the rocket’s direction of travel to sense the launch. To mount the mechanical Gswitch the designer must know the orientation of the CubeSat within the deployer, the deployer on the rocket, and the rocket while being launched. If the mechanical Gswitch is not mounted properly then the CubeSat will not detect the launch. The orientation of the CubeSat within the deployer is well known. Coordination with the launch integrator will ensure that this information is correct. The orientation of the deployer on the rocket is much less known. This information is not publicly available and is not often shared with CubeSat developers. The orientation of the rocket during launch and storage may not be vertical. SpaceX integrates payloads into the Falcon 9 horizontally and then lifted to vertical when at the pad[20]. Other rockets launch from the horizontal position from the bellies of planes as with the Launcher One from Virgin[35] or the Pegasus from Northrup Grumman[18]. LEO flies on a Falcon Heavy and will not launch horizontally. LEO will not have a problem detecting the vertical launch but this design cannot be used for the Sync payload. Sync is not designed for one specific rocket so the launch orientation may not be known. LEO’s design leaves room for improvement. These problems may be reduced with a change to the design of the accelerometer and trigger mechanisms. Sync addresses problems found in the LEO design process.

2.2 Alaska Research Cubesat 1(ARC-1)

ARC-1, a 1U CubeSat designed by the University of Alaska Fairbanks, has a launch environment data logger to record the environment inside a CubeSat deployer on launch. The ARC-1 CubeSat launched in 2015 on an Atlas V but did not send its radio beacon, indicating a power-on failure[11][6].
2.2.1 ARC-1 Mission Requirements

The data logger for ARC-1, like LEO, records the environment inside the CubeSat deployer on the way to orbit[8]. The mission requirements for ARC-1 as follows:

1. The ARC-1 CubeSat shall detect initiation of launch within 120 seconds of ignition.

2. The ARC-1 CubeSat shall measure vibrations from the NASA General Environmental Verification Standard (GEVS) using an 18G accelerometer. The standard covers the range of 20–2000Hz at a maximum level of $10G_{rms}$ for acceptance testing.

3. The ARC-1 CubeSat shall measure mechanical shock with a 70G accelerometer.

4. the ARC-1 CubeSat shall measure temperature in the range of -40°C to 105°C.

5. The ARC-1 CubeSat shall record for up to two hours to capture events from launch to 60 seconds after deployment.
2.2.2 ARC-1 Mission Requirements Justification and Fulfillment

The requirements vary slightly from LEO but have similar justifications. ARC-1 meets these requirements using different methods than LEO. The following sections expand on the mission requirements and their justification.

2.2.2.1 Launch Detection

ARC-1 detects the launch using piezoelectric sensors. The sensors flex in one axis and produce a voltage that triggers an interrupt in the microcontroller. Due to space constraints, ARC-1 only uses two piezoelectric sensors to detect the launch. The sensors resonate at 240Hz, a frequency within the expected range of vibrations at launch. When the rocket vibrates at 240Hz the launch interrupt triggers. ARC-1 uses two axes instead of three with cross-axis coupling. Cross-axis coupling occurs when vibrations in one axis have smaller components in a second axis. The smaller amplitude vibrations in the secondary axis will still trigger the launch detection circuitry of ARC-1 if the primary axis vibrations are large. The ARC-1 spacecraft verifies that the launch has occurred by checking the static acceleration of the CubeSat. The static acceleration is the acceleration that is not due to vibrations, it appears as an offset vibrations data. The static acceleration at any given time depends on the rocket. ARC-1 on the Atlas V should have an acceleration of 2G 120 seconds after launch. ARC-1 is designed for an Atlas V and will experience a different acceleration than LEO on the Falcon 9.

2.2.2.2 Vibrations Requirement

ARC-1 uses the levels for the NASA GEVS acceptance testing as an approximation of the environment on the rocket. The NASA GEVS acceptance testing amplitude is
Acceptance testing is a lower intensity GEVS test than used on LEO and Sync. LEO and Sync use the GEVS qualification test which uses vibrations with a 2–2000Hz frequency range and 14$G_{rms}$ amplitude. ARC-1 assumes a sine driving force while calculating the sensitivity of the accelerometer for monitoring vibrations. The GEVS acceptance test is randomly driven not sine driven. For the random noise, the peak to peak for 99.6% of the signal is $6 \times G_{rms}$ requiring a $60G_{pp}$ or $\pm30G$ accelerometer. The 70G accelerometer is selected to measure shock because it can read much higher accelerations than the 18G accelerometer. ARC-1 lists acceleration sensitivities of 0.009G/bit and 0.034G/bit. The acceleration sensitivity of ARC-1 is higher than LEO but uses an accelerometer that will clip in the GEVS acceptance test. The test results for random vibrations are not given.

### 2.2.2.3 Temperature Requirement

The temperature inside the deployer is measured with the AD7410[1], an IC-based temperature sensor. The ARC-1 sensors read the temperature of boards in separate locations of the CubeSat.

### 2.2.2.4 Time Requirement

The recording time requirement for ARC-1 is two hours. ARC-1 launched on an Atlas V and has a different launch timeline than LEO. Launch timelines are dependent on the orbit and launch vehicle.

### 2.3 Previous Work Conclusions

LEO and ARC-1 sense similar environments. Both CubeSat designs target the NASA GEVS standard. LEO tests to the higher severity GEVS Qualification standard and
ARC-1 test to the lower severity GEVS Acceptance standard. Both CubeSats record vibrations, temperature, and mechanical shock. The primary differences between LEO and ARC-1 are the launch detection mechanism and the sensitivity of accelerometers. LEO uses a mass-spring system while ARC-1 uses tuned piezoelectric sensors. Both methods allow the CubeSat to detect the launch without an electrical signal from the rocket. LEO confirms the launch with a barometer while ARC-1 uses the acceleration of the rocket from the launch timeline. These methods require mission specific parameters for pressure or acceleration, but will not detect false launches from handling. The recording time for each mission is different. LEO and ARC-1 are designed for different rockets and will take different times to get to their final orbit and deploy. LEOs mission targets 10 minutes of recording with the recording timed around events of interest. ARC-1 records for a two-hour period. LEOs recording method needs less data capacity but will miss information when not recording. ARC-1 will produce more data but capture unexpected events. LEO and ARC-1 both contain two accelerometers; one high sensitivity accelerometer to measure the vibrations that shake the CubeSat, and one low sensitivity accelerometer to sense mechanical shock. LEO uses 25G and 200G accelerometers whereas the ARC-1 CubeSat uses an 18G accelerometer and 70G accelerometer. The accelerometer sensitivity is dependent on the expected environment. LEO has lower sensitivity accelerometers than ARC-1 because it measures more severe vibrations than ARC-1. LEO and ARC-1 have different designs to suit different rockets. Both CubeSat designs must be on during the launch, trigger without electrical signals from the rocket, and record vibrations and mechanical shock. Rocket independent design criteria are incorporated into the Sync design along with lessons learned from LEO to make a compact launch environment data logger.
Chapter 3

SYSTEM DESIGN

Figure 3.1: The working block diagram of the system

3.1 Sync Mission Goals

Sync is a CubeSat payload that records vibrations and temperature from the launch environment inside a CubeSat deployer. The Sync payload is a secondary payload for CubeSats and is intended to fit into extra space in a CubeSat. The system must meet four design goals to function properly as a secondary payload for CubeSats. It must record data from the launch, have a long shelf life, function independently of the main CubeSat during launch, and be as compact as possible. The Sync payload must record data from the liftoff of the rocket to the deployment of the CubeSat. The primary goal of the sync payload is to collect environmental data from within the CubeSat deployer.
CubeSat launches occasionally delay by weeks to months, and the CubeSat developer may not have access to the CubeSat during the delay. The Sync payload must support a yearlong standby period to reduce the impact of launch slips on launch recording. A yearlong standby time also enables early integration of the Sync payload into the host CubeSat, without the need to open the CubeSat to replace batteries.

The Sync Payload shall function independently from the CubeSats main controller during launch. The Sync payload must be active while the rocket is traveling to space. Accidental radio transmissions may interfere with the radio communications of the rocket. Accidental deployment of antennas, booms, or other mechanisms may damage or jam a CubeSat in the deployer. Waivers are needed for any CubeSat to be powered while the rocket travels to space. The waiver process for Sync will be easier if the radio and deployment mechanisms are not powered when the payload is recording the launch. Maintaining the payload system as independent from the host CubeSat removes the risk of powering radios and deployable mechanisms.

The Sync Payload shall be as small as possible. As a secondary CubeSat payload, the Sync payload must fit in the remaining space within a CubeSat. Sync is not designed to fit within one CubeSat, but to be as small as possible to fit within as many CubeSats as possible. the volume consumed by the payload increases the number of possible CubeSats the payload will fit in.

3.2 Mission Requirements

The mission requirements listed in this section build upon the LEO mission requirements in section 2.1.1. The changes from LEO mission requirements are based on feedback from students and staff working on LEO. Each requirement is listed here then elaborated in the following subsection.
1. The Sync payload shall detect the rocket launch and begin recording as soon as possible.

2. The Sync payload shall be able to record using three accelerometers with sensitivities of 25, 100, and 200G.

3. The Sync Payload must record vibrations data with a for 90 minutes.

4. The Sync Payload must be electrically isolated from the host system during launch.

5. The Sync Payload must be smaller than 0.5U (10x10x5cm).

6. The Sync Payload should have a standby time of one year or greater.

3.2.1 GEVS Qualification and Accelerometers 25,100, and 200G

The GEVS qualification standard contains frequencies between 20–2000Hz at an intensity of $14.1G_{RMS}$ (See Figure 3.2). The high sensitivity accelerometer saturates during vibrations testing on the LEO CubeSat. From the $G_{RMS}$ value and the use of random vibrations, the peak to peak accelerometer range needed is $84.6G_{pp}$. Section 3.5.1 discusses this calculation. A 100G accelerometer is suggested to provide additional sensing range and prevent saturation. A 25G accelerometer is used to measure lower amplitude vibrations when lower intensities are expected. Lower intensities may be seen on some rockets or in some locations. The Atlas V aft bulkhead carrier, a location that CubeSat deployers have been mounted to, tests to $7G_{RMS}[10]$. The 25, 100, and 200G accelerometers provide a range of sensitivities to allow Sync to record on multiple different launch vehicles.

The full scale range of reach accelerometer is derived from the LEO fundamental requirements of 0.8G/bit and 0.08G/bit. The system sensitivity is 0.064, 0.258, and 0.516 G/bit for the 25,100, and 200G accelerometers respectively.
3.2.1.1 Bit Resolution

The resolution is more precise than the original LEO requirements (section 2.1.1) but below that of the implemented analog to digital converter on LEO. Reducing the number of bits reduces the resolution of the sampled signal but allows for the Sync Payload to use internal rather than external analog to digital converters.

3.2.1.2 90 Minute Recording

LEO records for a period of ten minutes broken up across multiple events of interest. These are pre-scheduled events and LEO will not show any information from unexpected events. The 90-minute recording period for Sync covers the full launch from liftoff to the deployment of the CubeSat. The payload must have sufficient power and memory to record for 90 minutes. The recording and standby time define the total amount of energy that the payload must store. The data rate and the recording length set the needed memory size.

3.2.1.3 Electrical Isolation

Sync must be electrically isolated from the host CubeSat for the duration of the launch. Isolating the host CubeSat from Sync on launch prevents Sync from back powering the host CubeSat.

3.2.1.4 Small Overall Size

CubeSats have extremely limited internal space. There must be space to successfully fit Sync as a secondary payload. If the payload is as small as possible in the vertical direction, then it will fit into more CubeSats. Many of the payloads that Cal Poly use fit between the rails of the CubeSat. The payloads do not have spare room alongside
them. The CubeSat frame limits payload X and Y dimensions and the primary payload limit the Z height. The Sync Payload targets 0.5U of volume (10x10x5cm) as the maximum volume for the payload.

3.2.1.5 One Year of Standby

The LEO spacecraft, as well as many other Cal Poly CubeSats, have experienced multiple month launch delays. The delays are due to many factors and access to the satellite is not always available during a delay. A long shelf life limits the maintenance required by the Sync payload. A year of standby allows the payload to remain ready for recording through a number of launch delays. This requirement drives the majority of battery capacity requirements. Reducing the standby power as much as possible improves the payload’s ability to meet this requirement.
Figure 3.2: NASA GEVs profile for spacecraft vibrations testing used to guide Cal Poly vibrations tests.
3.3 Microcontroller Selection

The processor on the Sync payload controls the payload state, and data transfer from sensors to memory. The processor must process data fast enough to sample data consistently while storing the data in payload memory.

CubeSats and the payloads inside them use Microprocessors or Microcontrollers to process data[7]. Complex CubeSat payloads have separate microprocessor that are tailored to the science missions while simpler CubeSat payloads share the processor with the rest of the CubeSat. Processors manage science instruments and process data before sending it to the ground. The primary differences between microprocessors and microcontrollers are the speed, computational power, and included peripherals. Microprocessors are typically faster, see table 3.1, but consume more power. Microprocessors also need external memory and persistent storage. Microcontrollers are less powerful but have integrated ram and persistent memory. Microcontrollers require less complicated board setup and are used when low power and small size are needed. The sync payload does not require a large amount of processing power and can benefit from the smaller board area of a microcontroller.

The Sync payload uses a microcontroller for its smaller board area, lower power, and shorter parts list. Cal Poly uses the MSP430FR5739 for sensors and health monitoring. The MSP430FR5739 is compact and uses FRAM memory. The FRAM memory is more radiation tolerant than EEPROM, making it more reliable in the space environment[32].

3.3.1 Microcontroller Selection

The MSP430FR5739 has twelve internal analog to digital converters, three universal serial interfaces, and a real-time clock. The Sync payload needs nine analog interfaces
for the three three-axis accelerometers and one analog interface for the thermocouple. The three serial interfaces transfer data between the MSP430, the memory, and the CubeSat processor. The real-time clock functionality replaces an external real-time clock IC to track the time and date. The time and date of each trigger event are referenced to a known launch date.

### Table 3.1: Overview of possible microcontroller/microprocessors for the Sync payload.

<table>
<thead>
<tr>
<th>Property</th>
<th>MSP430FR5739</th>
<th>PIC18F6720</th>
<th>AT91SAM9G20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Microcontroller</td>
<td>Microcontroller</td>
<td>Microprocessor</td>
</tr>
<tr>
<td>Speed</td>
<td>24MHz</td>
<td>4MHz</td>
<td>400Mhz</td>
</tr>
<tr>
<td>GPIO</td>
<td>32</td>
<td>52</td>
<td>96</td>
</tr>
<tr>
<td>ADC</td>
<td>12, 10-Bit</td>
<td>12, 10-Bit</td>
<td>4, 10-Bit</td>
</tr>
</tbody>
</table>

#### 3.4 Electrical GSwitch Design

The LEO mechanical GSwitch has multiple limitations that the Sync electrical GSwitch improves. The LEO mechanical GSwitch uses a tuned mass-spring system to make an electrical connection when the rocket accelerates upwards. The mechanical GSwitch functionality is limited by the orientation of the CubeSat on the rocket (see section 2.1.2). The new design uses the LIS3DH[21] a low power MEMS accelerometer. The LIS3DH has multiple settings that allow for extremely low current consumption while monitoring the acceleration of the payload[21]. The active consumption of 11 µA allows the accelerometer to remain on continuously. The LIS3DH has 3 parameters that improve its performance over the mechanical sensor. The LIS3DH has a programmable sensitivity, threshold, and duration. Also, the sensor has a configurable high pass filter. These parameters can be fine-tuned to the launch vehicle. The duration parameter prevents handling from triggering the system, Figure 3.3. The high...
pass filter removes DC acceleration components (DC acceleration components in this context are the effects of gravity measured by the sensor). The High pass filter also allows the payload to have sub-1G thresholds without orientation considerations.

![Diagram of interrupt status and threshold setting behavior for GSwitch interrupt](image)

**Figure 3.3: Threshold setting behavior for GSwitch interrupt**

The electrical GSwitch may be tuned by software rather than mechanical corrections. Software tuning allows the trigger adjustment after integration into the CubeSat. When the launch Vehicle is known the electrical GSwitch is set with a threshold and duration. Shorter durations are more sensitive to handling than long durations but will trigger closer to a launch. Similarly, lower amplitude thresholds will trigger with fewer vibrations but may be triggered by handling the CubeSat.
3.5 Accelerometer Design

The accelerometers are the sensors that translate the vibrations of the payload to electrical signals. This section covers the selection of the three accelerometers in the payload. These sensors fulfill the requirements listed in Section 3.2. Specifically that:

The Sync payload must measure vibrations with frequencies between 20-2000Hz using three three-axis accelerometers. The accelerometers should have sensitivities of 25, 100, and 200G.

3.5.1 Expected Vibrations

The NASA GEVS standard tests to 14.1\textit{G}_{RMS} \cite{22}. Because the noise is random, the peak to peak amplitude is dependent on the standard deviation of a normal distribution. The standard deviation of the normal distribution is the \textit{G}_{RMS}. For 99.7\% of the signal, the peak to peak amplitude is six times the standard deviation or six times the \textit{G}_{RMS} of the test. The accelerometers are rated in magnitude so three times \textit{G}_{RMS} indicates the accelerometer needed. For the Nasa GEVS Qualification test Sync needs an accelerometer $84.2\textit{G}_{PP}$ For the GEVS Acceptance test the level is $10\textit{G}_{RMS}$ so Sync would need an accelerometer of 30G to properly measure the signal.

3.5.1.1 Accelerometer Types

The most common type of accelerometer in electronics is the DC MEMS accelerometer. The DC MEMS accelerometer is designed to measure relatively low-frequency accelerations. DC MEMS accelerometers are often used in navigation to measure orientation and static accelerations. Many MEMS accelerometers feature a digital interface. These interfaces communicate over an array of protocols including SPI and
I2C. In general, these components are low power, low cost, and easy to integrate into electronic devices.

AC accelerometers provide signals without a bias from static accelerations. These signals are useful when the orientation is not needed but the vibration is. These sensors are most common in vibrations measurement and primarily exist in two forms, charge mode and voltage mode. Charge mode piezoelectric sensors produce a charge from a piezoelectric element that is transferred through a cable to a remote charge measuring device. Voltage mode piezoelectric sensors use voltage to signal the acceleration of the sensor. Voltage mode sensors are charge mode sensors with the supporting circuitry to convert the charge based signal to a voltage inside the sensor. Voltage mode sensors are easier to interface with but are often larger due to the internal analog circuitry[23]. The charge mode sensor’s small size is the result of using external processing circuitry. The Sync payload uses voltage mode sensors because the supporting circuitry to convert the charge to voltage is much smaller when implemented in the sensor than with discrete components.

3.5.1.2 Sensor Bandwidth

The accelerometer bandwidth defines the mechanical frequency range in which the sensor is able to sense. The accelerometer is not able to convert vibrations outside the accelerometer bandwidth. It is important that the accelerometer has a bandwidth of at least double the highest frequency to be measured. If the accelerometer bandwidth is not large enough aliasing and data loss will occur. The desired bandwidth is 20-2000Hz from Section 3.2.
3.5.2 Accelerometer Mounting

The accelerometers can be mounted using epoxy, bolts, or solder. The first and most common method for Cal Poly vibrations testing is to mount the sensor to the structure with glue or epoxy. Attached sensors have cables mounted to them to transfer the signal to a processing board. LEO uses this method to connect the accelerometers to the structure. Mounting the accelerometers directly to the structure allows the user to measure the vibrations at any point of the structure large enough to mount the sensor to. Alignment is an issue when mounting accelerometers with epoxy. The sensor may not be placed in the right orientation when placed on the structure or the sensor coordinate system may not align with the CubeSat coordinate system.

Mounting with bolts generally allows for larger accelerometers. Bolts scale to the size of the accelerometer. The mounting holes must also be selected early in the design. The proper placement of the holes removes the chance for technicians to mount the accelerometer improperly. In general, many of the benefits for glued accelerometer placement are also benefits for bolted accelerometers.

In most, mass-produced products, accelerometers are board mounted with solder. The board mounted accelerometer moves with the PCB, not the structure. If the PCB flexes relative to the structure then the sensor readings will be distorted by the flex of the board. If the board flex is mitigated, then the PCB mounted sensor is easier to design and test because the accelerometer does not need a structure to mount to.

3.5.3 Design Choice And Justification

The sync payload uses three 3 axis accelerometers that surface mount to the board. Integration into multiple spacecraft is simplified by using board mounted accelerom-
eters. Mounting the sensors to the board standardizes the performance of the Sync payload because the performance is not dependent on the CubeSat structure. The selected accelerometers are the 832M1-25, 832M1-100, and 832M1-200[25]. These three accelerometers are ac coupled voltage mode piezoelectric sensors. Each sensor reads acceleration along three axes.

The accelerometers cover three different dynamic ranges to reduce the impact of saturation. Each accelerometer has the same voltage range for its output and, by extension, the same range of digital values used in the analog to digital converter. The dynamic range of each sensor allows three different types of events to be observed. The lowest sensor reads from 0 to 25G. This sensor is intended for the lower level vibrations of the payload. A second sensor with range 0 to 100G provides readings when the 25G sensor clips. A third sensor that reads up to 200G is used for mechanical shock. Shock is the result of sharp high acceleration events. Examples of shock include the sharp acceleration at ignition of the rocket, or the acceleration felt from stage separation.

3.6 Analog To Digital Converter Design

The Analog to Digital Converter (ADC) converts analog values from the accelerometers to digital values. The digital values are then processed and stored by the microprocessor. The analog values are sampled at discrete times at quantized values, sacrificing some signal detail to allow the microcontroller to process the data. The speed and resolution of the ADC affect the fidelity of the digital signal. sampling a fast signal with a slow analog to digital converter results in information loss and signal distortion. The Nyquist frequency of the ADC is double the frequency of the signal. The signal must be sampled at this rate to properly recover the frequency component of the signal. The resolution of the ADC affects the amplitude reconstruction of the signal. many ADCs quantize signals to fewer bits to sample faster. Higher resolution
ADCs represent smaller voltage steps between digital values.

### 3.6.0.1 Expected Signal

The Sync mission requirements define 2000Hz as the highest frequency of interest (Section 3.2). The bandwidth of the accelerometers is 6000Hz, exceeding the required 2000Hz and producing signals above the 2000Hz that are not of interest. The Sync Payload must sample at a minimum of 12kHz to avoid aliasing any signals in the bandwidth of the accelerometer. An anti-aliasing, or low pass, filter attenuates signals higher than a set passband. Sync uses an anti-aliasing filter with a passband of 0–2000Hz to attenuate signals with frequencies higher than 2000Hz. A low pass filter with a stop band at 6000Hz and a passband up to 2000Hz minimizes the aliasing of signals outside of the band of interest.

The transition band between the pass band and stop band is 4000Hz. A higher order filter is needed to reduce the transition width between the start and stop bands. Increasing the filter order increases the number of components in each filter but reduces the required sampling rate by moving the stop band closer. The MSP430FR5739 can support higher sampling rates so the system is optimized to reduce components.

The MSP430FR5739 samples nine accelerometer channels and one thermocouple channel at 12kHz each. The MSP430FR5739 samples each channel sequentially [33]. The MSP430FR5739 will produce samples at a rate of 120ksps when sampling only ten channels at 12KHz each.

The MSP430FR5739 must sample across all 16 analog to digital channels due to the arrangement of external ADC channels. Four internal channels and two unused external channels are intermixed with the ten channels to be sampled.

To reduce code complexity and execution time, the four internal channels and two
extra external channels are sampled sequentially with the ten channels for Sync sensors. The MSP430FR5739 produces 192ksps when Sampling across 16 total channels at the expected channel sampling rate of 12ksps.

Each sample from the analog to digital converter is 10 bits. The system sensitivity using Equation 2.1, is 0.064, 0.258, and 0.516 G/bit for the 25, 100, and 200G accelerometers respectively.

3.7 Anti-Aliasing Filter

Not all signals on the analog lines will be produced by the accelerometers; Higher frequency noise from the digital circuits may contaminate the signals. Anti-aliasing filters are low pass filters designed to prevent high-frequency noise from interfering with the intended measurement. Any signal above half the sampling frequency will alias into the primary frequency range and irreparably corrupt the signal of interest. Anti-aliasing filters remove the signals outside of the recording range before the analog to digital conversion.

The Sync payload uses a low-pass filter on each of the channels before the ADC to prevent high-frequency interference. The anti-aliasing filters are placed as close to the analog to digital converter as possible to reduce the coupling of high-frequency signals.

\[ f_{\text{nyquist}} = \frac{f_{\text{sample}}}{2} \]  

(3.1)

The anti-aliasing filter is an active filter to increase the attenuation of high-frequency signals. The attenuation in the passband is kept as flat as possible with unity gain. This passband is from 2–2000Hz as targeted by the mission requirements in section 3.2. The stop band is set to attenuate by 20dB at 6000Hz to reduce the number of components on the board for each filter. An attenuation of 20dB reduces the signal magnitude by ten. The low pass filter attenuates at \(5.75 \times 10^{-3} \text{dB/Hz}\) starting at
2000Hz, the end of the pass band.

At ten bits, the least significant bit of the analog to digital converter represents 3.3mV. Signals at 6kHz must be greater than $3.3mV \times 20dB$ or 33mV to alias into the sampled range. If the out of band signal has a full-scale range of 3.3V then the anti-aliasing filter must attenuate by 60dB for the signal not to alias. Full-scale signals are not expected from the accelerometers above the rated 2–6000kHz band. The filter does attenuate at 60dB or higher for frequencies above 60kHz.

The analog anti-aliasing filters are designed to minimize component count so that they can fit onto one side of the board. The filters must repeat for each of the ten analog channels, each additional component in the filter requires ten additional components when implemented for every channel. A Chebyshev low-pass filter with a passband to 2000Hz, stop band is at 6000Hz is used to filter each channel before sampling. A Chebyshev[13] low-pass filter is used because it has a steeper transition after the end of the passband when compared to Butterworth or Bessel low pass filters [13]. The passband ripple of this filter is 0.07dB resulting in no more than 0.07dB difference between the attenuation and gain in the passband. Low pass band ripple limits the distortion of the signal in the pass band. The steep drop-off allows the passband and stopband to be closer without needing a higher order filter. The Chebyshev filter trades passband ripple for transition band slope. The higher the ripple in the passband the steeper the slope in the transition band[13]. The Chebyshev low-pass filter is implemented with a second order Sallen Key topology[13], as shown in figure 3.5.

3.7.1 Analog To Digital Design Conclusion

Sync uses a 10-bit analog to digital converter to sample signals with a 3.3V full-scale range and frequencies from 2–6000Hz. Mission requirements in Section 3.2 require
Figure 3.4: the filter shape of the anti-aliasing filter. Component tolerance effect indicated by shadow

Figure 3.5: The general schematic design of one anti-aliasing filter
20–2000Hz. The oversampling at 12kHz combined with an anti-aliasing filter prevent signal corruption from signals in the 2000–6000Hz range. The major limit to the filter is board space. The filter is duplicated ten times on the PCB. Given more PCB space, higher order filters should be used. Higher order filters will reduce the transition band and enable a lower sampling rate. A lower sampling rate requires less memory to store a signal for a given amount of time.

3.8 Memory

The Sync mission requirements (section 3.2) for bit width, sampling rate, and recording time define the size and speed of memory needed. The capacity and MPS430 interface requirements limit the memory selection to NAND flash and SD card memory. The following section discusses the limitations and part selection.

3.8.1 Memory Properties Derived From Mission Requirements

The requirements for the Sync payload impact the size and speed of the memory. The general capacity required is calculated with Equation 3.2.

\[
\text{capacity}[\text{bits}] = \text{bit width}[\text{bit}] \times \frac{\text{data rate}[\text{sample}]}{\text{sample}[\text{second}]} \times \text{recording time}[\text{seconds}]
\]

Equation 3.2 does not account for any overhead for the time stamps or other memory items that improve the usability of the system. For this approximation, it is assumed that the time-stamps will not be frequent enough to be a sizable portion of the memory. The Sync mission requirements define a 10-bit sample of the accelerometer signal. Without bit packing, the 10-bit samples will be stored in 16-bit variables; the native data size of the MSP430.

The MSP430FR5739 samples at 200ksps[33]. Each sample is 16 bits producing
3.2Mbps. The memory must support a write speed of 3.2Mbps or faster to continuously capture data. The required recording time is 90 minutes (Section 3.2). The resulting minimum capacity required from equation 3.2 is shown in equation 3.3

\[
\text{capacity[bits]} = 16 \left[ \frac{\text{bit}}{\text{sample}} \right] \times 200 \times 10^3 \left[ \frac{\text{sample}}{\text{second}} \right] \times 5400 \left[ \frac{\text{seconds}}{} \right]
\]

\[
\text{capacity} = 17.28 \times 10^9 [\text{bits}]
\]

Or

\[
\text{capacity} = 2.16 \times 10^9 [\text{bytes}]
\]

The Sync payload needs at least 17.28Gbits of memory with a minimum write speed of 3.2MBps to properly record samples for 90 minutes at 200ksps.

### 3.8.1.1 Memory Type

Memory components vary in capacity and speed with the memory type. Table 3.2 contains the max capacity and number of components available from the DigiKey selection of memory parts. The availability of the components, as well as the maximum capacity available, is shown in the table. EEPROM is very common but with small capacities, while NAND flash is much less common but five orders of magnitude higher in capacity. The Sync payload memory requirements prevent EEPROM and FRAM as options due to the limited capacity. The cost of manufacturing the PCB prohibit using packages with ball grid arrays. Ball grid array packages connect to the PCB using balls of solder under the chip, making the printed circuit board assembly difficult. The available memory chips are further limited since the MSP430FR5730 supports only SPI [33] peripherals and not dual or quad SPI used by higher capacity memory. Limiting the selection by type of interface reduces the available parts to those shown in table 3.3. The memory available with SPI interfaces does not have the capacity to record for 90 minutes on a single chip. At the time of board design the highest capacity SPI memory chip without ball grid array pins is the W25N01GVZEIG[36] a 1Gb
NAND flash chip. The W25N01GVZEIG supports write speed up to 104Mbps[36], exceeding the required 3.2Mbps write speed. The capacity of the W25N01GVZEIG is 1Gbit. One 1Gbit chip limits the recording time to approximately five minutes. The SPI bus does not limit the number of connected components, so eight SPI memory chips are used to increase the recording time to 40 minutes. The number of chips is limited to eight due to the available space on the board.

The capacity of NAND or NOR flash memory chips is not high enough to meet the 90 minute recording time. A micro SD card has much higher capacity with a SPI interface with capacities ranging from 4–32GB for the Kingston high capacity industrial grade cards[9]. The SD card SPI interface supports up to 25Mbps over SPI[9], which exceeds the expected data rate of 3.2Mbps. Industrial grade cards are used for the extended temperature range of -40–85°C. The primary concern with using a micro SD card is the electrical connection to the card. The micro SD card does not solder to the board and uses a metal cage to mount as shown in figure 3.6. The spring contacts may disconnect under the vibrations of the launch and stop data recording.

3.8.2 Sync Selection

Both soldered NAND flash and a micro SD card are included in the Sync payload design. The Sync payload will use the micro SD card to record data and the NAND will only be used if the SD card fails. Preliminary vibrations tests in Section 5.1 demonstrate that the SD card maintains contact with the microcontroller through X, Y, and Z-axis testing. Removing the NAND flash array is suggested for future board revisions. The space used by the array may be used for extra battery capacity.
Figure 3.6: Two images of the SD card cage. The left image shows the gold spring contacts exposed without the SD card. The right image shows a micro SD card in the card cage.

Table 3.2: Memory availability and capacity generated from in stock selection at DigiKey at time of writing.[4]

<table>
<thead>
<tr>
<th>Type</th>
<th>Max capacity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEPROM</td>
<td>2Mb</td>
<td>4922</td>
</tr>
<tr>
<td>FRAM</td>
<td>4Mb</td>
<td>187</td>
</tr>
<tr>
<td>NAND FLASH</td>
<td>512Gb</td>
<td>382</td>
</tr>
<tr>
<td>NOR FLASH</td>
<td>2Gb</td>
<td>1066</td>
</tr>
</tbody>
</table>

Table 3.3: Non volatile memory with a SPI interface available at DigiKey at the time of writing

<table>
<thead>
<tr>
<th>Type</th>
<th>Max Capacity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND</td>
<td>2Gb</td>
<td>21</td>
</tr>
<tr>
<td>NOR</td>
<td>2Gb</td>
<td>172</td>
</tr>
</tbody>
</table>
3.9 Databudget

The Sync payload produces 3.2Mbps of data while monitoring the launch environment. This section verifies that the Sync payload can properly record data for the 90 minute launch period.

3.9.0.1 Internal Data Transfer

Data transfers internally from the ADC peripheral to the internal FRAM memory where it held until it is written to the SD card. The transfer process to and from the FRAM uses the DMA. A DMA transfer takes two clock cycles to transfer data from one register to the other. Under normal operations, the transfer speed is 12MHz and transfers two bytes per transaction. This internal transfer speed (192Mbps) is greater than the 3.2Mbps production rate of the analog to digital converters. The internal transfer does not limit the data rate of the system.

3.9.0.2 External Transfer To The SD Card

The data is stored in an SD card. The interface to the SD is limited by the SPI clock speed of the MSP430. The SD card needs a token before each block in the transfer in multiblock transfer mode. The token adds one byte to each block transferred as well as two bytes of cyclic redundancy check (CRC) and at least one byte of dummy data to allow the SD card to respond[9]. The bus efficiency is then

\[
\frac{512B}{1B_{token} + 512B_{data} + 2B_{CRC} + 1B_{response}} \times 100\% = 99.2\%
\]

This calculation assumes that the commands for starting and stopping the transfer are not significant compared to the data transferred, which is the case for most multi-block transfers.
3.9.0.3 SD Limits

The SD card write speed depends on the class of card and the state of the block being overwritten. The industrial grade 8 GB Kingston Micro SD card selected in section 3.8.1.1 supports write speeds of 20MB/s or 160Mbps[9]. The bus write speed is limited by the speed of the controller and the physical process of writing to memory cells. The MSP430 limits the bus transfer rate to 24Mbps which does not exceed the 160Mbps write speed of the SD card. The transfer to SD card memory is limited by the MSP430, not the SD card.

3.9.0.4 Data Budget Summary

The Sync payload produces data at a rate of 3.2Mbps. The data production rate is lower than the transfer rate between the ADC and internal FRAM memory. The data production rate is lower than the transfer rate between the internal FRAM memory and the SD card. The Sync payload supports recording at the full rate and does not need to throttle the sample rate to record to the SD card.

3.10 Structure

The Sync Payload structure prevents the board from flexing independent of the structure. The board must move with the structure to measure the accelerations properly.

3.10.0.1 Sync Plate

If the payload flexes, it will move separately from the structure and change the recorded data. By limiting the flex of the board, the accelerometers on the board better match the movement of the structure. The design goal of the plate is to reduce flex in the board and increase the accuracy of the system. The plate must be ridged.
Figure 3.7: A rendering of the Sync backing plate with half of the test structure hidden.

enough to support the board and prevent flexing. The thickness of the plate provides the stiffness and contributes a significant portion of the mass of the system.

Typical boards manufactured for Cal Poly are mounted at the four corners with screws. Four more mounting points are added to this board to reduce flexing in the center of the Sync payload.

The board design and mounting are analyzed in Abacus by Gerardo Ramirez. Abacus provides a Finite Element Analysis (FEA) modeling of various objects. The FEA model simulates the board and screws holding it down to simulate how much it will flex when vibrated. The model does not simulate the copper layers or the masses of each electrical component as a volume. The simulation uses point masses in the center of mass for each component to reduce the complexity of the simulation. It does, however, provide a rough approximation and shows that the board does
The greatest area of concern is the area under the three accelerometers. These areas are monitored during simulation and shown to shift by micrometers. The test performed assumed no backing plate and fixed the mounting holes in space. The simulation uses the NASA GEVS qualification standard as a stimulus. These levels are designed to show that hardware will survive the launch. The full structure will be tested to NASA GEVS before flight to confirm that it will survive launch.
3.11 Power

The Sync payload is designed with a one year standby time and 90 minute recording period. To reach one year of standby the Sync payload uses low power components and powers down sections of the board to conserve power when not in use.

Two power modes exist in Sync, standby and active. In the standby state, the payload draws microamps of power to maintain a long battery life. In active mode, all of the components are drawing power to record vibrations data. This section addresses the methods used to reduce power consumption and noise in the analog circuitry.

3.11.0.1 Regulator Separation

Total system power draw is reduced by switching off components that are not needed in the current payload mode. In the active power state, the SD card memory and analog components are powered to record data. In standby mode, only the MSP430 and electrical GSwitch are powered.
The components operating in standby use low quiescent current linear regulators to reduce current draw. Quiescent current is the current consumed by a regulator when active. Reducing quiescent current makes the total current draw of the system closer to the current drawn by the components behind the regulator.

The components that are not needed in the standby power state are supplied by a separate regulator with an enable line. The enable line allows the regulator to be powered down in the standby mode, significantly reducing power consumption.

3.11.0.2 Analog Considerations

The board design of the Sync payload isolates the digital and analog sections of the board reducing the likelihood of digital noise in the analog signal. Separate ground and power planes isolate the digital and analog areas of the board. Separate linear regulators supply the two regions of the board to remove switching noise from the supply side. The digital and analog grounds connect through one zero-ohm resistor following the star method. The star method[28] connects analog and digital power and ground at one point to prevent digital noise from entering the analog circuitry.

3.11.0.3 Power Budget

A power budget defines the expected current in active and standby mode. Table 3.4 lists the power draw of the Sync payload components in standby mode, omitting the unpowered memory and analog components. The estimate includes the current consumed by disabled regulators to improve the estimate. In active mode, the power estimate in table 3.5 includes the memory, analog components, and the higher power draw of the MSP430FR5739. The estimated current draw in the active mode assumes that the SD card is constantly written to and that the NAND flash is inactive. The actual power draw in active mode will be lower because the SD card will be written
Table 3.4: Estimated standby current draw.

<table>
<thead>
<tr>
<th>Item</th>
<th>Current (A)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>msp430</td>
<td>6.3E-6</td>
<td>Low Power Mode 3 MSP430fr5739 datasheet</td>
</tr>
<tr>
<td>electrical gswitch</td>
<td>11.0E-6</td>
<td>LIS3DH datasheet</td>
</tr>
<tr>
<td>always on regulator</td>
<td>100.0E-6</td>
<td>LP2981 Datasheet</td>
</tr>
<tr>
<td>off digital switched regulator</td>
<td>10.0E-6</td>
<td>Shutdown mode TL1963A Datasheet</td>
</tr>
<tr>
<td>off analog switched regulator</td>
<td>1.0E-6</td>
<td>Shutdown mode LP2981 Datasheet</td>
</tr>
<tr>
<td>total</td>
<td>128.3E-6</td>
<td>Amps</td>
</tr>
</tbody>
</table>

in bursts. Using the NAND flash instead of the SD card reduces the current draw by 100mA, but decreases system memory.

Sync is expected to be in standby for at most 8760 hours, and in active mode for one and a half hours. The total required battery capacity for each mode is 1.124AHrs and 0.49AHrs for standby and active mode respectively. Batteries for sync must have a capacity of 1.615AHrs at 3.3V to meet the required year of standby and 90-minute recording.

3.11.0.4 Batteries

The Sync payload is independently powered from the CubeSat to ensure that the CubeSat radios and deployment mechanisms are not powered during launch. Volume and mass are limited in the Sync payload, and rechargeable batteries do not provide the energy density to remain on standby for one year. The high capacity per volume and mass of CR123 lithium battery is ideal for the Sync payload. Sync’s lithium cells are not needed after launch and do not need to be rechargeable.

Underwriter’s Laboratory(UL) listing demonstrates that the batteries have been tested and demonstrated safe for many conditions. The use of UL listed batteries meet
Table 3.5: Estimated active current draw

<table>
<thead>
<tr>
<th>Item</th>
<th>Current (A)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>msp430</td>
<td>2.0E-3</td>
<td>81.4E-6*24 from the datasheet for full power at 24MHz</td>
</tr>
<tr>
<td>electrical gswitch</td>
<td>11.0E-6</td>
<td>LIS3DH datasheet</td>
</tr>
<tr>
<td>always on regulator</td>
<td>100.0E-6</td>
<td>LP2981 Datasheet</td>
</tr>
<tr>
<td>On digital switched regulator</td>
<td>100.0E-6</td>
<td>Shutdown mode TL1963A Datasheet</td>
</tr>
<tr>
<td>On analog switched regulator</td>
<td>100.0E-6</td>
<td>Shutdown mode LP2981 Datasheet</td>
</tr>
<tr>
<td>SD card</td>
<td>300.0E-3</td>
<td>Kingston datasheet for industrial cards</td>
</tr>
<tr>
<td>memory chips x8</td>
<td>200.0E-6</td>
<td>W25N01GVZEIG datasheet standby</td>
</tr>
<tr>
<td>accelerometers x3</td>
<td>66.0E-6</td>
<td>832m1 datasheet</td>
</tr>
<tr>
<td>AA filter x10</td>
<td>25.0E-3</td>
<td>AD8656 datasheet</td>
</tr>
<tr>
<td>total</td>
<td>327.5E-3</td>
<td>Amps</td>
</tr>
</tbody>
</table>

the Orbital Debris Assessment Report and Range Safety requirements [12]. Non UL listed batteries might be used in CubeSats but then need to be tested to ensure that they meet standards.

The batteries recommended for the Sync payload are UL listed CR123 cells with a capacity of at least 1500mAH. These cells provide 3V and 1500mAH in a compact cylindrical shape. Using two cells in series provides six volts at full charge and will maintain at least four volts until discharged.

3.12 CubeSat Interface

Sync is designed as a CubeSat payload and must electrically interface with the CubeSat main board to transfer data after the CubeSat deploys. Before the CubeSat is deployed, Sync must be isolated from the payload to avoid back powering the CubeSat. The sections that follow address the Isolation, power and data connections
between the Sync Payload and the CubeSat

3.12.1 Power

The CubeSat and Sync power system are connected through a diode on the Vdd side and MOSFET on the ground. The Sync power system is independent of the CubeSat power system while the CubeSat is not deployed. When the CubeSat deploys and connects the grounds of the payload and CubeSat, the CubeSat powers the Payload. The CubeSat is designed to power the Sync payload after deployment so that the CubeSat can retrieve data if the Sync batteries die.

3.12.2 Isolation

Isolation is important because the Sync payload is powered while the CubeSat is inside the deployer and the mainboard is not powered. The Sync payload must avoid back powering the CubeSat to prevent CubeSat components from turning on while connected to Sync. Back powering the CubeSat is not safe and may damage components. The payload will need a waiver from the launch integrator for violating the CubeSat specification by remaining "on" during launch. An isolated payload design will help receive a waiver for the Sync Payload.

The system uses diodes to stop reverse current flow on power lines from the payload into the CubeSat. The mounting holes are not connected to ground and the vias are covered with solder mask to prevent accidental shorts on the plate. The sync interface board contains a low side switching MOSFET that breaks the connection between the Sync payload and CubeSat ground. When the MOSFET breaks the circuit, the ground has no return path to the rest of the CubeSat. The low side switch is activated by the CubeSat mainboard only, preventing the Sync from connecting the ground.
3.12.3 Data

The payload must transfer data to the CubeSat to send it to the ground. The data transfers to the CubeSat after the CubeSat has deployed and powered on. The electrical environment is unknown, and the cable length may be as long as 30cm. Cal Poly CubeSats have had problems with cables inside CubeSats picking up noise from the radio, processors, and power electronics inside the CubeSat. In Sync, the data connection between the host CubeSat and the Sync payload uses Low Voltage Differential Signaling (LVDS) SPI. LVDS SPI improves the signal integrity with high common mode noise rejection.

Better signal quality enables higher transfer speeds and reduces the time to transfer data off the payload. The differential signal provides lower interference to components around the payload cable due to the magnetic fields of the balanced signals canceling.[34]

The Payload and the CubeSat interface board use matching LVDS converters to change single ended SPI into differential SPI. The SN65LVDT14 and SN65LVDT41[30] are compact LVDS converters for SPI. The SN65LVDT14 is the SPI slave converter on the Sync payload board and the SN65LVDT41 is the SPI master converter on the CubeSat interface board.
Chapter 4
SYSTEM IMPLEMENTATION

The system implementation section contains the schematic, layout, and software design of the system. In this section the practical aspects of the system implementation are addressed including component selection, board layout choice, and software implementation.

4.1 Schematic

The schematic defines the connections made between the electrical components. Many computer aided drawing programs provide tools that make the process of defining the connections or nets easier. In the schematic creation process, multiple tools are used to make sure that all nets are connected properly and that any missing connections are intentional.

4.1.1 Component Selection

Many of the main components were selected in the design phase. These components are critical to the mission and are needed to meet specific requirements laid out in the design phase. The pre-selected components include the accelerometers, the microcontroller, and the memory. Where possible the same component will be used in similar configurations. Reusing multiple components reduces the BOM\(^1\) cost as well as the cost of assembly by reducing the number of unique components.

\(^1\)Bill Of Materials
4.1.2 regulators

The regulators convert the 6V from the batteries to the 3.3V that all payload components need. The design of the power system in Section 3.11 defines three separate regulators. The analog regulator must be able to supply relatively high power with a small amount of noise, the digital always on regulator must be as efficient as possible, and the digital switched regulator must be able to handle the 500mA power needed by the memory table 3.5.

![Figure 4.1: the schematic layout of the always on digital regulator and the switched analog regulator](image)

The TL1963A[31] is selected for the memory regulator as it supports up to 1.5A of current with a low 1mA quiescent current. The shutdown current is 1\(\mu\)A[31], The on-current capacity is able to support the SD card and memory chips at the same time.

The LP2981 [29] is used for both the analog regulator and the always-on regulator. This regulator has a low quiescent current of 100\(\mu\)A and high ripple noise rejection of 63dB of attenuation[29]. The low quiescent current reduces the power used by the system in standby. The high ripple noise rejection makes the regulator ideal for isolating the analog supply line from the digital one.

4.1.2.1 Supporting Memory Circuits

Sync uses an array of NAND flash and an SD card to store vibrations data. The NAND and SD card are selected in Section 3.8.1.1 and need circuitry to support
them. Both the SD card and the NAND flash array connect to the MSP430FR5739 using SPI. A demultiplexer is used to select the proper chip select from the array and reduce the number of MSP430FR5739 pins needed to control eight NAND chips. Figure 4.2 contains the schematic layout of the NAND flash. The micro SD card

![Figure 4.2: NAND flash chip array with demultiplexing chip](image)

selected in Section 3.8.1.1 conforms to a standard electrical connection[9]. The SD card is not directly soldered to the board and uses an SD card cage (Figure 3.6) to make electrical contact with the board. The card cage selected is used on Cal Poly system boards as they have been shown to survive the vibrations of the rocket launch. The SD card cage is electrically connected in SPI interface mode rather then the standard SD card interface. Note that the SD card data outline requires a pull-up to a logic high to function properly.

### 4.1.3 Analog Signal Chain

The analog signal chain extends from the analog accelerometers through the anti-aliasing filters, and ends at the analog inputs of the MSPFR5739. The anti-aliasing filters contain identical opamp circuits that match the design from Section 3.7. The opamps are designed for low noise high bandwidth rail to rail performance to allow
Figure 4.3: the SD card cage schematic as implemented for an analog design with only one power regulator. The AD8656 low noise opamp meets the general design requirements with a rail to rail performance and a 28MHz bandwidth[2]. The AD8656 contains two opamps in each package.

Figure 4.4: Accelerometer and anti-aliasing filter arrangement.

Each analog component references the analog ground and 3.3V supply. The anti-aliasing filters are labeled as low pass in figure 4.4. Low pass filter blocks are hierarchical schematic elements that internally contain the passive components, op-amp, and analog supplies needed for the active filtering. Hierarchical blocks allow for faster
more readable design by copying the same elements without cluttering the page they are used on.

Figure 4.5: the internal elements of the anti-aliasing/low-pass blocks. Note that two filters are present in this design.

4.1.4 MPS430

The microcontroller has the most connections out of any component. Only the Vcore pin is left floating in this design. In Figure 4.6 all of the pins are referenced.

Figure 4.6: The schematic connection of the MSP430FR5739 microcontroller.

4.1.4.1 Design Challenges

Mistakes that were made in the original design are documented here in this section. These include separating the digital and analog references, missing capacitors on the
crystal, and shorting Vcore to the supply voltage.

In the original design, different regulators supply the analog and digital supply pins. The analog reference pin was supplied with analog 3.3V which is set to 0V in the standby power state. In the standby power mode, the analog reference pin became the source of power for the analog 3.3V net. Aside from damaging the microcontroller, supplying analog 3.3V from the microcontroller also increased power draw in the standby power mode. The increase in power drastically reduced the standby time of the payload. This error is corrected by referencing the analog supply to the always-on regulator.

The crystal oscillator was originally missing the 22pF load capacitors preventing the system from oscillating. Without a crystal oscillator, the msp430 real-time clock stops keeping time. The timing feature is key to tracking the mission launch date.

Shorting Vcore to the supply pins bypasses an internal regulator and causes higher power draw in low power states. Vcore is not intended to connect to the supply voltage. Under normal operations, this core will pull low in some low power modes.

4.1.5 Schematic Verification

Before routing all components were verified to be in stock from electronic component distributors. The design was also checked for any components that might be obsolete or near the end of their production cycle. At this stage, most passive components can be substituted out for drop-in replacements, but any larger obsolete components will require a design change.

The schematic is also checked for single node nets. Single node nets represent pins that do not connect to anything. On large designs, pins may be overlooked resulting in missing functionality for chips.
4.2 Board Layout

The layout is the physical arrangement of components on the PCB as well as the design of the copper shapes that connect them. The sync board uses 136 components with 106 unique nets and 368 connections. The board design attempts to keep as many components on the negative Z side of the board as possible. The negative Z side meets with the plate to prevent board flex. The fewer components on the negative Z board side, the better the physical connection to the backing plate.

4.2.0.1 Board Stack-Up

The board uses four layers to improve routing and allow for internal power and ground planes. All electrical connections between the layers (Vias) go through each layer.

![Figure 4.7: Board stack-up showing 4 conducting layers and supporting nonconducting layers](image)

The digital and analog sections of the board are isolated to avoid noise from the switching circuits. The analog area is isolated with a via fence and grounds are connected through one 0Ω resistor. The supply side is connected through a low noise low dropout regulator. These practices were previously discussed in the power design
section 3.11.0.2.

Figure 4.8: Negative Z board side with analog area shaded in red

By limiting the intermixing of analog and digital components the noise on the analog tracks are reduced. Using analog ground planes as shielding allows for digital tracks to run under the analog section of the board on the opposite side. The board uses via fences to limit the noise from the digital planes. The MSP430 is the only component with digital and analog sides, and as such, it bridges the analog and digital planes.
To reduce the length of all traces, analog and digital, the MSP430 is placed as close to centerboard as possible. Placing the MPS430 close to center prevents digital and analog traces from crossing. The digital traces are kept to a minimum length to increase the maximum speed.

The NAND flash chips greatly increase the complexity of the high-speed routing. The general large size of the array coupled with the limited board space prevented the optimal placement of the NAND flash. The sub-optimal placement of the NAND flash involves vias in the traces for data increasing the noise in the signal lines.

The clock line is routed as cleanly as possible. The clock line routing avoids vias where possible and is routed with a fixed impedance. The impedance of the board is not controlled, but the width is selected to roughly match a 50Ω impedance. The fixed impedance reduces noise from miss-matched connections.

An attempt is made to maintain unbroken ground and power planes. Breaking the ground plane under any major current path or signal line results in much higher noise as current is forced to take a longer path to close the current loop.

4.2.0.3 Board Area And Keepouts

Most components are placed on the top of the board to reduce board flex; reducing the number of components on the plate side of the PCB decreases the machined pockets in the plate and increases the contact area. Figure 4.8 illustrates the density of the board design. The usable surface area of the board is 11.24in², with 5.76in² of the component area on the negative z side of the board. The components cover 51.3% of the negative Z board area. The components on the positive Z (plate side) of the board cover 0.5in² of the board or 4% of the total area. The components placed on the bottom of the board are localized to one area away from the accelerometers to provide a solid backing surface to the accelerometers.
4.3 Software

This section covers the software implemented on the Sync payload. First, an overview of the software structure is provided, then, an overview of the software of individual components. All of the software on the Sync payload is written in C and run on the MSP430FR5739. The Code Composer Studio development environment, a custom eclipse distribution by Texas Instruments[27], was used while developing the code.

![Figure 4.9: The code flow of the Sync payload board. Interrupts not shown.](image)

The microcontroller starts by initializing the real-time clock peripheral and the GSwitch. This initialization does not re-run unless the system is rebooted. At this point, all general-purpose input and output pins (GPIO) are set to input to reduce the system power draw. After setting the real-time clock and GSwitch, the system enters a low power state.

When the electronic GSwitch triggers, an interrupt is sent to the MSP430 and the system leaves the low power state. The interrupt is cleared and the current date in the real-time clock is checked against the launch date. If the Launch date has not passed, then the system will sleep again. If the launch date has passed, then the system will increase the clock speed, initialize the micro SD card, and begin taking samples. The system will continue to record until the observation period has elapsed.

When the observation period is over the system enters an idle state. Optionally the system may enter sleep mode again and wait for another event to trigger it. The re-sleep re-trigger function allows the system to capture a launch even if a false
detection previously occurred.

At any point, after initialization, the payload may enter a command mode. The command mode is triggered by a user command over UART. The command mode allows the user to change payload parameters. Editable parameters include time and date, launch date, and recording length.

### 4.3.0.1 Initialization State

The initialization stage prepares the Sync payload to trigger when it detects a launch and configures the UART console for user interface. The initialization state programs the real-time clock with the time set by the user and configures the electronic GSwitch with predefined thresholds. The GSwitch is configured at this stage with the programmed thresholds for the specific test or rocket. After configuration the GSwitch will toggle an interrupt line when vibrations exceed the programmed thresholds. The UART interface is initialized here, allowing the user to reconfigure the payload. The initialization state also powers down the memory and analog regulators to prepare to enter sleep mode.

### 4.3.0.2 Sleep State

The sleep state puts the microcontroller into low power mode 3 (LPM3). LPM3 turns off the core clock but keeps the real-time clock and GPIO interrupts enabled. Sync can exit LPM3 from an external interrupt, from the real-time clock, or a UART command. The UART commands cause Sync to leave the sleep state and enter the command state. The GSWitch interrupt triggers a date check to compare the current date with the expected launch date; if the date has not passed then the interrupt is cleared, and the system enters the sleep state again.
4.3.0.3 Recording Initialization State

This state powers on the memory and analog chain, initializes the SD card in SPI mode, and initializes the analog to digital conversion. The SD card initialization configures the card for the writes in the recording state. The analog to digital converter is configured at this stage to record at the sample rate defined in Section 3.6, and a DMA is configured to transfer each sample into the SD card block buffer. Sync exits the Recording Initialization state when an SD card and analog to digital converter initialize.

4.3.0.4 Recording State

The recording state monitors four buffer blocks that are filled by the DMA with samples from the analog to digital converter. When the DMA fills a buffer, the buffer is marked as full and the DMA begins filling the next buffer with samples. The main process polls the status of the four buffers and cues a DMA transfer to the SD card when a buffer is full. When the SD card transfer is finished the main process marks the buffer as empty and will begin emptying the next full buffer. When there are no full buffers, the main process does nothing.

One interrupt routine runs separately from the main process in the recording state. The interrupt routine cycles between four buffer blocks filling one at a time with DMA transfers from the analog to digital converter. Interrupts are triggered by the end of a 240 sample transfer from the analog to digital converter. The interrupt adds the date, time, and long duration counter to the remaining space in the 512-byte buffer, then marks it as full. The next DMA transfer to a free buffer is initialized and the interrupt returns to the main process.
4.3.0.5 Closing State

The Closing state ends the sequential write to the SD card and disables the analog to digital converter. The SD card protocol requires that sequential writes end with a write end command to the SD card. The Closing State sends this command to prevent the SD card from entering an unknown state. The analog to digital converter is shut down to reduce power consumption of the Sync payload.

4.3.0.6 Idle State

The idle state waits for commands from the CubeSat or user after the recording process ends. This state optionally prepares the system for the next recording.

The code for the Sync payload is written to use the above states while recording data. More state may be added as more features are added.

4.3.1 SD Card Software Interface

The MSP430 communicates with the SD card using SPI. The SD card boot process initializes the card in SPI mode rather than SD Express mode. SD Express is the standard interface for SD cards and uses four data lines instead of one[9]. SD Express is not supported by the MSP430 serial peripheral interfaces, so the card is initialized in SPI mode. The SPI commands must be sent at a lower data rate of 400KHz while the card is initializing. After initializing the SD card in SPI mode, commands exchange to establish high capacity card support by Sync. The SPI data rate increases after the SD card initialization is complete.

Card commands are short, about 6-8 bytes, and typically have 1-byte responses. The SD card responds in a variable number of bytes after the command. The SD card response requires sending 0xFF over SPI until the response byte is read. The
variable response time prevents DMA transfers for commands as they cannot respond to the contents of the SD card response.

Without the DMA, SPI transfers are processed in the main code and block other functions. Data blocks are sent with DMA transfers because the SD card responds only after 512 Bytes of data.

### 4.3.1.1 SD Card Memory Structure

The Sync payload does not use a FAT file system on the SD card. The microcontroller writes data blocks to the SD card that contain serialized data from the ADC, a time stamp, and a cycle counter. Each block consists of 240 samples from 16 channels; each channel is sampled 15 times per block. The time stamp and a cycle counter are used to reconstruct the timing of each sample.

The cycle counter increments every 64 clock cycles to provide sub-second resolution for the start of the sampling of each block. The time between each sample is fixed by the ADC, peripheral but the time between block writes fluctuates due to interrupts. The cycle counter is used to adjust the timing of each block of samples in post processing to account for the variable timing between blocks.

### 4.3.2 Electrical GSwitch

The LIS3DH used for the electrical GSwitch does not require constant monitoring or control from the MSP430FR5739. The LIS3DH is controlled with software-defined I2C. Software-defined I2C uses two GPIO pins controlled by software to emulate a hardware I2C peripheral. Using software-based I2C leaves the hardware serial peripherals to the SPI and UART communications. The SPI and UART interfaces need the serial peripherals because they use the generated interrupts to speed up communication. After configuration, the I2C interface is only used to reset the interrupt output.
of the LIS3DH.
Testing the Sync payload verifies that it meets the mission requirements. Vibrations testing is the most complex test and verifies multiple mission requirements. The data rate test and power tests are needed to verify the remaining requirements.

5.1 Vibration Testing

The NASA GEVS qualification test is designed to ensure that the CubeSat will make it to orbit without components breaking or screws coming loose. A CubeSat breaking inside the deployer may damage other CubeSats inside the deployer, or the worse, may damage the rocket. Therefore, CubeSats must be built to survive this test. The Vibrations test verifies the following mission requirements:

1. The Sync payload shall detect the rocket launch and begin recording as soon as possible.

2. The Sync payload shall be able to record the GEVS qualification test using three accelerometers with sensitivities of 25, 100, and 200G.

Launch detection verification involves placing Sync into the standby mode past the launch date, beginning the test, and verifying that the recording begins. Sync’s ability to record over the NASA GEVS qualification test is verified if Sync successfully records the data of a full single axis test.
5.1.1 Test Setup

Sync bolts to a 1U test structure for testing. The structure, designed and manufactured by Justin Sherrell and Max Selna of Cal Poly PolySat, provides a realistic test setup of a 1U CubeSat containing the Sync payload. The structure is integrated into a 1U TestPOD used for mounting CubeSats to the shake table (Figure 5.1). The shake table is an electronically controlled table designed to shake an object at specific amplitudes and frequencies and only shakes along one axis at a time. Any vibration in the directions besides the tested axis is uncontrolled and incidental to the test. A control accelerometer on the bottom of the Sync backing plate records the vibrations experienced by the payload. A visual inspection confirms the pass or fail status of the mechanical structure while the control accelerometer provides a reference to compare to the payloads recorded data.

5.1.2 Testing Process

The testing process consists of bolting the TestPOD to the shake table, programming Sync to record for a set period of time, and then shaking it with the shake table. The payload should then detect the vibrations, prepare the SD card, and record data on the accelerations experienced. This process repeats for each of the X, Y, and Z-axis.

The testing process involves increasing acceleration values in stages. The test first runs at -6dB of full scale then -3dB then at 0dB of full scale. The test increases in severity to minimize the damage if the structure begins to fail. The testing process generally follows a sine sweep, -6dB, -3dB, and 0dB random vibration, and then sine sweep.

The sine sweeps are low amplitude and follow a time-dependent frequency change. The pre and post sine sweeps indicate if the structure has changed after the random
Figure 5.1: The Sync payload inside a 1U TestPOD on the Cal Poly vibes table.

vibrations test. If the sweeps are comparable, than the structure is undamaged. The random vibrations phase is much higher in amplitude than the sine sweeps. It is the random vibrations phase that mechanically strains the system to ensure it is able to survive vibes. The random vibrations conform to the NASA GEVS standard for qualification (figure 5.2). This is the most strenuous test spacecraft will experience before launch.
5.1.3 testing results

5.1.3.1 Results Of Payload Accelerometer Readings

The payload successfully recorded across all three axes and all three sensors confirming that the payload can detect a launch and begin recording. The largest concern was the reliability of the SD card electrical connection. If the SD card fails, then the backup NAND must be used capture data. The SD card reliability is verified by preventing SD card restarts during the vibrations testing. The SD card succeeded in the three separate runs for each axis.

To verify the recording the data was transferred from the SD card and visualized in open MCT[17]. Open MCT is a NASA data visualization tool that allows users to inspect data from databases. Open MCT was used to separate the sine sweeps from the random vibes.

The separated data is converted to power spectral density in python. The vibrations analysis looks at the power spectral density of the accelerations. The power spectral densities plotted in figure 5.3 are created using the FFT on approximately one second slices of the time domain signal.

FFTs assume that the signal is periodic. The time domain signals seen in figure 5.5 have stepped amplitudes for each shift in intensity. Sampling over a shorter range reduces the impact of larger low-frequency trends on the power spectral density such as the changes in the severity of the random test.

The resolution of the frequency conversion is dependent on the sampling rate and the number of samples in each conversion. Each FFT calculation used $2^{14}$ samples, or approximately one second of data. Using Equation 5.1.3.1 the bin width is 0.9Hz.

$$\Delta f_{bin} = \frac{\text{channel sampling frequency}}{\text{number of samples}}$$ (5.1)
The number of samples is set to a power of two to increase the processing speed. The FFT calculation performs best on powers of two length inputs. The power spectral density is computed from the FFT of the input signal using:

$$PSD(x_t) = \frac{1}{N \times F_s} \times |FFT(x_t)|^2$$  \hfill (5.2)

where $N$ is the number of samples, $x_t$ is the time domain signal and $F_s$ is the sampling frequency of the channel. The $G_{RMS}$ of a given range of frequencies is:

$$G_{RMS} = \sqrt{\int_{f_1}^{f_2} PSD(f) \, df}$$  \hfill (5.3)

The $g_{rms}$ is used to represent the magnitude of the acceleration sensed in a given frequency range. The power spectral density and the $g_{rms}$ are used to represent vibrations experienced in vibes due to the nature of the signal being measured. The input signal is a stationary band limited white noise.

### 5.1.3.2 Control and Payload Data Comparison

Figure 5.3 contains the power spectral density from each x-axis accelerometer channel. The Sync X-axis aligns with the control sensor y-axis during the vibrations testing. The amplitude of the PSDs in figure 5.3 is higher then the control in Figure 5.4. Figure 5.4 shows the power spectral density of multiple points of the test setup. The control data for the Sync payload is the Y-axis plate channel. The control sensor is located on the plate below the board and not on the board(figure 5.6). The payload data displays the same frequency peaks but does not have the same amplitude. The control data is an average of multiple runs where as the payload data is the maximum value for each frequency bin over the duration of the test. Part of the vertical shift
is due to comparing the average (control values) with the maximum (Sync) values.

The time series control data is not available. In the payload time series data, amplitudes over 100G were detected. The increased amplitude in the time domain data is potentially caused by board flex and may be solved with adhesive between the plate and PCB.

5.1.4 Test Conclusions

the vibrations test demonstrates that the Sync payload properly triggers and begins recording data when it detects a launch. The data that the payload records does not match the control data in amplitude but contains the same frequency peaks. It may be possible to correct the offset between control and payload data using an adhesive between the plate and PCB. This correction requires retesting the payload but may improve the accuracy of the data. Alternatively the data may be corrected without further testing by developing a model of the transfer function between the PCB and the backing plate. Both of these options are left as future work for individuals with a better understanding of vibrations testing.
Figure 5.2: NASA GEVS profile for spacecraft vibrations testing used to guide Cal Poly vibrations tests.
Figure 5.3: Peak hold of the power spectral density created from 1 second portions of the random x axis vibrations test
Figure 5.4: The control sensor’s Y-axis data which corresponds to the payload sensors X-Axis data.
Figure 5.5: Plot of the sampled signal of the x axis channel for the three accelerometers. Note the clipping in the 25g signal.
Figure 5.6: The location of the three-axis control accelerometer on the opposite side of the Sync payload plate
5.2 Data Rate Testing

This section covers the test process for the data transfer internal to the MSP430 to SD card pair. The data creation rate from the analog to digital converter is verified by measuring the time to create and transfer a block of 240 samples to the SD card. The sampling rate impacts the time to frequency conversion used to understand vibrations data.

5.2.1 Testing Methodology

The sync payload board has one pin available for general debugging. This pin is toggled high at the completion of a block write to the SD card and is toggled low when a block cache is filled in the MSP430 FRAM. The writes to the SD card and internal cache happen concurrently so the high and low timing is not strictly the timing of the individual components. One cycle is completed for each block written to memory. By finding the frequency the number of blocks written to the SD card per second is found. Given that each block contains 240 samples per second (Section 4.3.1.1) the sampling rate can be extrapolated. The sampling rate is slightly higher than this value to accommodate the overhead of interrupt handling, ADC configuration, and DMA setup.

5.2.2 Testing Results

The frequency is obtained over approximately 37 blocks using the measure function of the oscilloscope. A visual inspection of figure 5.8 shows that the period of each pulse is generally consistent. The full test procedure and write up is present in appendix C with the other as run documents.

At 240 samples per block, the lower bound of the sampling rate is computed by
Equation 5.4 and 5.5 as 225.456ksps.

\[
\text{sample rate} = \text{block rate} \times 240 \quad (5.4)
\]

\[
225.456\text{ksps} = 939.4\text{Hz} \times 240 \quad (5.5)
\]

This sample rate is higher than the expected 200ksps. Across each channel, the sample rate is 14.1ksps versus the expected channel sampling rate of 12.5ksps. 939.4 blocks are created every second.

The original timing expected 833.3 blocks every second. The block rate is 113%
of the original expected value. The system is running at 113% speed of the designed system. Setting the end block without adjusting for the higher speed will result in the recording ending early.

The increased speed relates to the use of the sub-main clock to feed the analog to digital converter. The original design used the asynchronous peripheral clock for the ADC. The clock was switched to the sub main clock due to stability issues. The sub-main clock runs at 24MHz and is divided by 5. Since approximately 20 clock cycles to process a 10-bit sample, the resulting sampling rate is 240ksps. This sampling rate is much higher than expected. Setting the clock divider to 6 rather than five brings the sampling rate back to the expected value of 200ksps.

5.2.3 Summary

The data rate test indicates the current system runs faster than expected. Recording samples faster than expected will result in improperly set recording times while also increasing the frequency range that can be sensed. The extra bandwidth is not needed for the Sync payload.

5.3 Power Budget Verification

The verification of the power budget ensures that the payload meets the intended standby time as well as recording time. Each state has an expected power draw and excessive power draw in either mode reduces the total standby and run time.

The power draw for most components is well defined in the datasheet. Any excess power draw indicates a design oversight that may damage components or shorten the runtime.
5.3.1 Pin States and Power Draw

The configuration of each pin has a dramatic effect on the power consumption of the board. From testing the lowest power state is achieved by defining all unused pins as inputs with pull-down resistors. By defining every inactive or unused pin this way, the power consumption of the MSP430 is decreased. This is because in the lowest power state, the majority of IC’s on the board are unpowered. Leaving pins floating high or undefined on the MSP430 causes excess power to leak into these IC’s through the data and gpio lines. Defining the pins as pulled down and input removes the current flow from the MPS430 to the unpowered components saving 300\(\mu A\) of current in idle modes.

| Table 5.1: Predicted vs Actual current. |
|-----------------|-----------------|-----------------|
| Mode            | Predicted Current | Actual Current  |
| Standby         | 128\(\mu A\)     | 113\(\mu A\)   |
| Active          | 527.3\(mA\)      | 109\(mA\)      |

5.3.2 Measured Power vs Expected power

The sync payload draws less current in both modes than predicted in the design. The power predicted of each mode is dependent on the max value listed in the datasheet. The performance improvement is greatest in the active mode as the maximum power draw is least representative of the actual performance. The maximum current draw occurs only when data is being written to external memory. Including write duty cycles for the memory and adding a separate profile for changing memory types can increase the accuracy of the power estimate. At the data production rate of 225.456ksp/s found in section 5.2 writes for an average of 36% of the time. At a standby power draw of 960\(\mu A\), the SD card draws an average power of 108mA. This
improves the power estimate to 135mA. While not matching the measured power draw, the new estimate of 135mA is much more accurate.

5.4 Conclusion

The results of the power draw test indicate that the power consumed by the board is lower than expected. One year of standby and 90 minutes of recording requires 1.15AHr of capacity while the designed battery capacity provides 1.5AHr using the measured power of 113µA in standby and 109mA in active mode. The remaining capacity may be used for 129 more days of standby, 3 more 90-minute recordings, or some combination of both. The Sync payload power consumption performs better than required by the original mission requirements 3.2.
The design and results of tests on the Sync payload show that Sync meets the established requirements in Section 3.2. By meeting these requirements Sync demonstrates that it is a functional data logger implementable as a secondary payload in CubeSats as small as 1U. The Sync payload meets each of the original mission requirements as follows.

6.0.1 Launch Detection

The Sync payload requirement for detecting launch is:

_The Sync payload shall detect the rocket launch and begin recording as soon as possible._

The Sync payload uses a MEMS accelerometer as an electronic GSwitch to monitor the acceleration of the payload in a low power state. The thresholds on the accelerometer prevent false triggers while handling the CubeSat and reduce false launch detections. In testing the electronic GSwitch triggered the recording at the start of each vibrations test demonstrating its ability to detect the launch. The acceleration thresholds are configurable in software to meet the unique requirements of each rocket.

6.0.2 GEVS Qualification and Accelerometers 25, 100, and 200G

The Sync payload requirement for recording vibration is:
The Sync payload shall be able to record the GEVS qualification test using three accelerometers with sensitivities of 25, 100, and 200G.

The Sync payload used three accelerometers to measure vibration during the vibrations test which tested to the GEVS qualification standard. The Sync payload recorded through the full test of each axis with all accelerometer channels. The amplitude of the power spectral density was higher than the control, but displayed peaks at similar frequencies. The recorded data indicates that the payload board experienced vibrations with greater intensity at a higher frequency than the control. Mechanical adjustments to the interface of the PCB and the plate may improve the payload’s response to vibrations. A transfer function may also be derived to compensate for the interface between the plate and the board.

6.0.2.1 90 Minute Recording

The Sync payload must record vibrations data with a 10-bit resolution for 90 minutes.

The Sync payload records with 10-bit resolution across nine accelerometer channels and stores data to an SD card. The system needs a minimum of 2.16GB to record data for 90 minutes. The system supports SDHC micro SD cards which have capacities up to 32GB or 22 hours of recording. Sync exceeds the recording requirements.

6.0.2.2 Electrical Isolation

Electrical Isolation separates the payload electronics from the CubeSat electronics for the duration of the launch. The specific requirement is that:

The Sync payload must be electrically isolated from the host system during launch.
This requirement is met by the design of the sync payload and interface board. Sync uses low side switching, insulated mounting holes, and power diodes to prevent back powering the host CubeSat. The Sync ground is not connected with the host system until switched by the host system. The board design and manufacturing prevent accidental shorts where the board contacts the structure.

6.0.2.3 Small Overall Size

The size of the Sync payload is limited by the requirement that:

*The Sync payload must be smaller than 0.5U (10x10x5cm)*

The payload is 2.5cm tall with batteries included, making the approximate volume of the payload 0.25U (10x10x2.5cm). The payload is visible from the side in Figure 3.7. The payload is shorter than required potentially allowing Sync to fit within more CubeSats.

6.0.2.4 One Year of Standby

The battery capacity and power draw in standby are defined by the requirement that:

*The Sync payload should have a standby time of one year or greater.*

The Sync payload draws 113µA of current in standby. The Sync battery pack has a capacity of 1500mAh. After 90 minutes of recording the battery pack has 1337mAh or 492 Days of standby. With two recording periods, Sync has 432 days of standby. The primary batteries in the Sync payload have the capacity to maintain standby longer than the required 365 days.

Sync meets each of the specified requirements allowing it to record vibrations inside the deployer. The payload is smaller, consumes less power, and can record
longer than required. Exceeding the mission requirements enables Sync to fit more CubeSat missions and record information from more rockets.

6.1 Future Work

Sync requires more work on the host interface. The MSP430FR5739 supports SPI slave functionality to transfer data to the host system. The LVDS converters on the host and payload side function, but the host software interface with the payload is not developed. The host interface must support large data transfers to retrieve the data from the Sync payload memory.

The development of a few more payload functions will be useful for reducing the amount of data to the downlink. Performing the PSD calculations on the MSP430 or host CubeSat can reduce downlink size by limiting the number of frequency bins. Averaging bins together or only downlinking the maximum PSD data will reduce the time to downlink data from the spacecraft.

The second revision of the Sync payload board does not require the NAND flash on the existing board. Removing the NAND flash frees board surface area that can be used by a higher bit analog to digital converter and/or higher order anti-aliasing filters. Higher order anti-aliasing filters enable a lower sampling rate and total memory requirements. Higher bitrate samples increase the system sensitivity to those seen in LEO without consuming more memory. Continuing the above work will make Sync a more efficient and effective data logging platform for CubeSats.
BIBLIOGRAPHY


[19] REOTEMP. Thermocouple Types - Types of thermocouples - Comparison of Thermocouple Types.


Appendices

A FEA Analysis of Sync Payload Board
MEMORANDUM

To: Christopher Gerdom, PolySat, Cal Poly SLO
cgerdom@calpoly.edu

From: Gerardo Ramirez
gramir17@calpoly.edu

Date: 5/29/2018

RE: Finite element analysis for the deflection of a PCB board undergoing GEVS vibration

Introduction

This finite element analysis study was performed to determine whether the three accelerometers on Chris’ Sync board for a 1U CubeSat are located at the anti-nodes when under vibration. Additionally, this study was performed to determine the maximum displacement of the three accelerometers. This analysis was chosen because of the concern that the vibrational loading conditions a satellite faces when launching may cause the accelerometers to resonate at the boards natural frequencies and deflect significantly.

Model Development, Boundary and Loading Conditions

To perform this analysis initially a SolidWorks CAD model was imported from a STP file. A 2D surface was used to create an extruded model of the board from the face of the STP file. This was then imported into ABAQUS.

FR-4, a woven composite fiberglass, is the material selected for the board. Table 1 shows the mechanical properties used for the model. Note that modeling of composites for FEA is significantly more difficult than homogeneous materials. FR-4 is typically woven in such a manner to improve mechanical properties in the fiber orientation of two directions, lengthwise and crosswise, which are both in XY plane of the board. The properties for lengthwise and crosswise young’s modulus do not vary significantly, with a typical value of 24 GPa and 21 GPa respectively. Since our board is predominantly two-dimensional, since deflections in the direction normal to the board are of interest, and since the lengthwise and crosswise properties are close in magnitude, the model for this analysis will conservatively use the smaller young’s modulus to define homogeneous material properties.

Table 1. Mechanical Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young’s Modulus</td>
<td>21 GPa</td>
</tr>
<tr>
<td>Poisson’s Ratio</td>
<td>0.12</td>
</tr>
<tr>
<td>Density</td>
<td>1850 kg/m³</td>
</tr>
</tbody>
</table>

The damping ratio for the circuit board was taken from the experimental results of the paper Modal Analysis of a Printed Circuit Board. Damping ratios of 0.012, 0.0072, and 0.0032 were used for the modal ranges of 1 to 2, 2 to 3, and 3 to 4 respectively.

To simulate the inertial mass of the three accelerometers, the special point mass inertia feature was utilized in ABAQUS. The location and dimensions of the accelerometers was determined in SolidWorks, and the
mass of one accelerometer was measured to be 0.00363 kilograms using a scale. The mass of each accelerometer was evenly distributed over a range of point masses to simulate the loading distribution.

The boundary conditions on the model are located at the eight holes of the boards bolt pattern. Each hole was constrained in all three degrees of translational freedom and all three degrees of rotational freedom.

A steady state dynamic modal loading was used in the initial step of the analysis. This dynamic loading analysis was initially used to determine the natural frequencies of the board over a range of 0-2000 Hz. Then, to simulate the real loading of a CubeSat under launch conditions, a linear perturbation acceleration base motion was used for the second step of the analysis. Data for the acceleration and frequency data was taken from NASA’s General Environmental Verification Standard (GEVS) for the qualification level of amplitude spectral data vibration.

**Analysis and Results**

Figures 1, 2, 3, and 4 show the natural frequencies of the board over the range of 0-2000 Hz. The first, second, third, and fourth mode were found to have frequencies of 1261.2 Hz, 1390.9 Hz, 1419.4 Hz, and 1525.6 Hz respectively. Note that the displacement is labeled U3, in the Z direction normal to the plane of the board. The displacement of the board is normalized to a maximum displacement of 1.0, shown in red in the figures. The location of red in each figure is associated with the anti-node of each natural modal frequency.

![Figure 1. Contour plot for U3 deflection, normalized to 1.0, first mode with a natural frequency of 1261.2 Hz.](image)

Figure 1. Contour plot for U3 deflection, normalized to 1.0, first mode with a natural frequency of 1261.2 Hz.
Figure 2. Contour plot for U3 deflection, normalized to 1.0, second mode with a natural frequency of 1390.9 Hz.

Figure 3. Contour plot for U3 deflection, normalized to 1.0, third mode with a natural frequency of 1419.4 Hz.

Figure 4. Contour plot for U3 deflection, normalized to 1.0, for the fourth mode with a natural frequency of 1525.6 Hz.
Figures 5, 6, 7, and 8 show contour plots for the U3 deflection of the board when accelerated under NASA’s General Environmental Verification Standard (GEVS) qualification level for amplitude spectral data vibration. The U3 deflection is normal to the plane of the board and is shown in meters on the scale. Figures 5, 6, 7, and 8 show the board under the first, second, third, and fourth mode natural frequencies from figures 1, 2, 3, and 4 respectively.

Figure 5. Contour plot for U3 deflection of board when accelerated under GEVS qualification vibration under a frequency of 1261 Hz. Units on the plot are shown in meters. Note that the maximum deflection of the entire board is -0.008977 mm. The order of magnitude of the deflection near the accelerometers is smaller, and closer to 0.0006555 mm.

Figure 6. Contour plot for U3 deflection of board when accelerated under GEVS qualification vibration under a frequency of 1391 Hz. Units on the plot are shown in meters. Note that the maximum deflection of the entire board is -0.02541 mm. The order of magnitude of the deflection near the accelerometers is smaller, and closer to 0.0008617 mm.
Figure 7. Contour plot for U3 deflection of board when accelerated under GEVS qualification vibration under a frequency of 1419 Hz. Units on the plot are shown in meters. Note that the maximum deflection of the entire board is -0.02302 mm. The order of magnitude of the deflection near the accelerometers is smaller, and closer to 0.002474 mm.

Figure 8. Contour plot for U3 deflection of board when accelerated under GEVS qualification vibration under a frequency of 1526 Hz. Units on the plot are shown in meters. Note that the maximum deflection of the entire board is -0.003794 mm. The order of magnitude of the deflection near the accelerometers is smaller, and closer to 0.0001569 mm.

Table 1 shows the results for U3 deflections of the board under GEVS qualification vibration loading. The deflections are tabulated for the four natural frequency modes of interest. For each mode the maximum deflection of the entire board is listed, as well as the maximum deflection of the accelerometers.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Frequency (Hz)</th>
<th>Maximum Deflection of Entire Board (mm)</th>
<th>Maximum Deflection of Accelerometers (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>1261</td>
<td>0.008977</td>
<td>0.004624</td>
</tr>
<tr>
<td>Second</td>
<td>1391</td>
<td>0.02541</td>
<td>0.003318</td>
</tr>
<tr>
<td>Third</td>
<td>1419</td>
<td>0.02302</td>
<td>0.003242</td>
</tr>
<tr>
<td>Fourth</td>
<td>1526</td>
<td>0.003794</td>
<td>0.002045</td>
</tr>
</tbody>
</table>
B Assembly Procedure
## Changelog

<table>
<thead>
<tr>
<th>Rev</th>
<th>Name</th>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Chris Gerdom</td>
<td>10/16/18</td>
<td>Creation</td>
</tr>
</tbody>
</table>

## Sync Payload assembly

### Materials/Consumables

<table>
<thead>
<tr>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal zip ties</td>
</tr>
<tr>
<td>8 2-56 screws</td>
</tr>
<tr>
<td>Sync payload board</td>
</tr>
<tr>
<td>Sync payload plate</td>
</tr>
</tbody>
</table>

### Tools

<table>
<thead>
<tr>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/64 hex driver</td>
</tr>
<tr>
<td>5/64 Bit</td>
</tr>
<tr>
<td>In-oz torque wrench</td>
</tr>
<tr>
<td>Tin snips</td>
</tr>
<tr>
<td>Pliers</td>
</tr>
<tr>
<td>#</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
3. SCREW IN 8 2-56 SCREWS

4. SET THE TORQUE WRENCH TO 40IN-OZ
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td><strong>TORQUE ALL SCREWS. USE A STAR PATTERN</strong></td>
</tr>
<tr>
<td>6.</td>
<td><strong>FEED METAL ZIP TIE THROUGH THE OUTER SLOT AND THEN THE INNER SLOT.</strong></td>
</tr>
<tr>
<td>7.</td>
<td><strong>FEED THE ZIP TIE THROUGH THE THICKER END AND PULL TIGHT</strong></td>
</tr>
<tr>
<td>8.</td>
<td><strong>CUT THE ZIP TIE SHORT. LEAVE ABOUT 1 CM</strong></td>
</tr>
</tbody>
</table>
9. Curl the end of the zip tie back to hide the sharp end
# Changelog

<table>
<thead>
<tr>
<th>Rev</th>
<th>Name</th>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Chris Gerdom</td>
<td>10/16/18</td>
<td>Creation</td>
</tr>
</tbody>
</table>

## Sync Payload assembly

<table>
<thead>
<tr>
<th>Materials/Consumables</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 5/16” 4-40 socket caps</td>
<td></td>
</tr>
<tr>
<td>x-y panels</td>
<td></td>
</tr>
<tr>
<td>Top hat</td>
<td></td>
</tr>
<tr>
<td>Shoe</td>
<td></td>
</tr>
<tr>
<td>Payload assembly</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tools</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/32 hex bit</td>
<td></td>
</tr>
<tr>
<td>3/32 hex driver</td>
<td></td>
</tr>
<tr>
<td>Torque wrench</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>INSTRUCTION</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1.</td>
<td>ATTACH -$Y$ TO THE PAYLOAD USING TWO 4-40 SOCKET CAPS</td>
</tr>
<tr>
<td>2.</td>
<td>ATTACH +$Y$ TO THE PAYLOAD USING TWO 4-40 SOCKET CAPS</td>
</tr>
</tbody>
</table>
3. ATTACH SHOE WITH 4 4-40 SCREWS

4. ATTACH THE TOPHAT WITH 4 4-40 SCREWS
5. Set the torque wrench to 80in-oz

6. Torque all screws on the +Y. Work in a star pattern

7. Torque all screws on the -Y. Work in a star pattern

8. 

9. 
C As Run Assembly Procedures
## Sync Payload assembly

<table>
<thead>
<tr>
<th>Materials/Consumables</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 5/16&quot; 4-40 socket caps</td>
<td>CG</td>
</tr>
<tr>
<td>x-y panels</td>
<td>CG</td>
</tr>
<tr>
<td>Top hat</td>
<td>CG</td>
</tr>
<tr>
<td>Shoe</td>
<td>CG</td>
</tr>
<tr>
<td>Payload assembly</td>
<td>CG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tools</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/32 hex bit</td>
<td>CG</td>
</tr>
<tr>
<td>3/32 hex driver</td>
<td>CG</td>
</tr>
<tr>
<td>Torque wrench</td>
<td>CG</td>
</tr>
</tbody>
</table>

Chris Gerdon
10/23/2018
20:03
<table>
<thead>
<tr>
<th>#</th>
<th>INSTRUCTION</th>
<th>INITIAL</th>
<th>DATE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ATTACH -Y TO THE PAYLOAD USING TWO 4-40 SOCKET CAPS</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>ATTACH +Y TO THE PAYLOAD USING TWO 4-40 SOCKET CAPS</td>
<td>CG</td>
<td></td>
</tr>
</tbody>
</table>
3. ATTACH SHOE WITH 4 4-40 SCREWS

4. ATTACH THE TOPHAT WITH 4 4-40 SCREWS
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>SET THE TORQUE WRENCH TO 80IN-OZ</td>
</tr>
<tr>
<td>6.</td>
<td>TORQUE ALL SCREWS ON THE +Y. WORK IN A STAR PATTERN</td>
</tr>
<tr>
<td>7.</td>
<td>TORQUE ALL SCREWS ON THE -Y. WORK IN A STAR PATTERN</td>
</tr>
<tr>
<td>8.</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td></td>
</tr>
</tbody>
</table>
Sync Payload assembly staking(-Z)

Materials

<table>
<thead>
<tr>
<th>Materials</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Syringe Plunger + Tip</td>
<td>Scale</td>
</tr>
<tr>
<td>Mixing Sticks</td>
<td>Staking Compound</td>
</tr>
<tr>
<td>Tri-Pour Mixing Cups</td>
<td>Razorblade / Exactoknife</td>
</tr>
</tbody>
</table>

Part 1: Prepare Staking

<table>
<thead>
<tr>
<th>#</th>
<th>Instruction</th>
<th>Initial</th>
<th>Date time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gather all hardware and components from Materials.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Clean all parts with IPA before bringing into the cleanroom.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Clean the workbench with IPA and equip your ESD bracelets.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Place a mixing cup on the scale and zero the scale.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Pour 1-2 grams of Scotch Weld Part A Staking into the mixing cup. Record the mass.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mass of Part A 1.0 58 g</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Calculate and record the required mass of Part B by multiplying the mass of Part A by 5/7.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Calculated mass of Part B 0.7 557 g</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Zero the scale again.</td>
<td>CG</td>
<td></td>
</tr>
</tbody>
</table>
8. Add the calculated amount of Scotch Weld Part B to the same mixing cup. Record the mass.
   Actual Mass of Part B 0.767

9. Stir the mixed parts until the color of the mixture is uniform, then stir for 1 more minute.

10. Pour the completed staking into a syringe slowly and evenly in order to ensure that no air bubbles form inside the syringe.

11. Turn the syringe so that the tip is facing up. Carefully squeeze the plunger until there is no air left in the tip.

**Part 2: Stake**

12. Stake the components in the locations specified by red dots in the figure above.

13. Bake at 90°C for 30 min

140 mm 67°C
## Changelog

<table>
<thead>
<tr>
<th>Rev</th>
<th>Name</th>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Chris Gerdon</td>
<td>10/16/18</td>
<td>Creation</td>
</tr>
</tbody>
</table>

## Sync Payload assembly

<table>
<thead>
<tr>
<th>Materials/Consumables</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal zip ties</td>
<td></td>
</tr>
<tr>
<td>8 2-56 screws</td>
<td></td>
</tr>
<tr>
<td>Sync payload board</td>
<td></td>
</tr>
<tr>
<td>Sync payload plate</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tools</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/64 hex driver</td>
<td></td>
</tr>
<tr>
<td>5/64 Bit</td>
<td></td>
</tr>
<tr>
<td>In-oz torque wrench</td>
<td></td>
</tr>
<tr>
<td>Tin snips</td>
<td></td>
</tr>
<tr>
<td>Pliers</td>
<td></td>
</tr>
</tbody>
</table>

*Chris Gerdon 10/23/18*
<table>
<thead>
<tr>
<th>#</th>
<th>INSTRUCTION</th>
<th>INITIAL</th>
<th>DATE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>PLACE PLATE ON THE TABLE</td>
<td>L6</td>
<td>10/23/18</td>
</tr>
<tr>
<td>2.</td>
<td>ALIGN PAYLOAD BOARD WITH PAYLOAD PLATE. NOTE CUTOUT FOR COMPONENTS UNDER SD CARD</td>
<td>L6</td>
<td></td>
</tr>
</tbody>
</table>
### 3. SCREW IN 8 2-56 SCREWS

![Image of screws being installed](image1)

### 4. SET THE TORQUE WRENCH TO 40IN- OZ

![Image of torque wrench](image2)
5. TORQUE ALL SCREWS. USE A STAR PATTERN

6. FEED METAL ZIP TIE THROUGH THE OUTER SLOT AND THEN THE INNER SLOT.

7. FEED THE ZIP TIE THROUGH THE THICKER END AND PULL TIGHT

8. CUT THE ZIP TIE SHORT. LEAVE ABOUT 1 CM
<table>
<thead>
<tr>
<th></th>
<th>CURL THE END OF THE ZIP TIE BACK TO HIDE THE SHARP END</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image_url" alt="Image of zip tie" /></td>
</tr>
</tbody>
</table>

LG
## Sync Structure Staking

### Materials

<table>
<thead>
<tr>
<th>Material</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syringe Plunger + Tip</td>
<td>Scale</td>
</tr>
<tr>
<td>Mixing Sticks</td>
<td>Staking Compound</td>
</tr>
<tr>
<td>Tri-Pour Mixing Cups</td>
<td>Razorblade / Exactoknife</td>
</tr>
</tbody>
</table>

### Part 1: Prepare Staking

<table>
<thead>
<tr>
<th>#</th>
<th>Instruction</th>
<th>Initial</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Gather all hardware and components from Materials.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Clean all parts with IPA before bringing into the cleanroom.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Clean the workbench with IPA and equip your ESD bracelets.</td>
<td>CG</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Place a mixing cup on the scale and zero the scale.</td>
<td>CG</td>
<td></td>
</tr>
</tbody>
</table>
| 5. | Pour 1-2 grams of Scotch Weld Part A Staking into the mixing cup. Record the mass.  
   Mass of Part A: 0.97  
   Calculated mass of Part B: 0.69  | CG      |           |
| 6. | Calculate and record the required mass of Part B by multiplying the mass of Part A by 5/7.                                                                                                                   | CG      |           |
| 7. | Zero the scale again.                                                                                                                                                                                        | CG      |           |
8. Add the calculated amount of Scotch Weld Part B to the same mixing cup. Record the mass.

   Actual Mass of Part B  0.65

9. Stir the mixed parts until the color of the mixture is uniform, then stir for 1 more minute.

10. Pour the completed staking into a syringe slowly and evenly in order to ensure that no air bubbles form inside the syringe.

11. Turn the syringe so that the tip is facing up. Carefully squeeze the plunger until there is no air left in the tip.

**Part 2: Stake**

12. Stake the components in the locations specified by red dots in the figure above.

13. Bake at 90°C for 30 min / 20 min

   67°c