Preliminary Information for Deployment Switches, Antenna Deployment and Vibe sensor testing.

Claire Lizotte

Contents

[Interface Document 2](#_Toc17203878)

[Deployment switch investigation & preliminary research 2](#_Toc17203879)

[Antenna Deployment investigation & preliminary research 8](#_Toc17203880)

[Vibe testing sensor placement 12](#_Toc17203881)

[Fixture 12](#_Toc17203882)

# Interface Document

The first chunk of the summer was spent reviewing the capstone team’s work to familiarize myself with the cubesat as a project, as a system, and to determine the requirements for it.

Then, the interface document was prepared with Stephen and Andrew. It was a group effort.

For the interface document, there is a tab for each sub system. Each sub system has each component in it, which then gets broken down into what connections it has. For example, the camera in the payload has a connection to the ESP, the OBC, and the structure.

I am not sure how up to date this is with the PC 104 stack and component changes.

There are two ADCS tabs. I am not sure what the difference between the two are.

It does not include a layout with the custom boards.

The pinouts for the OBC were included and assigned in the *wiring interface* tab.

# Deployment switch investigation & preliminary research

The requirements from Nanoracks for the deployment switches state:

**4.1.4 Deployment Switches**

1) The CubeSat shall have a minimum of three (3) deployment switches that correspond to

independent electrical inhibits on the main power system (see section on electrical interfaces).

2) Deployment switches of the pusher/plunger variety shall be located on the rail end faces of the

CubeSat’s -Z face.

3) Deployment switches of the roller/lever variety shall be embedded in the CubeSat rails (+/- X

or Y faces).

4) Roller/slider switches shall maintain contact with 75% of the NRCSD rail width along the entire

length of the rail.

5) The CubeSat deployment switches shall reset the payload to the pre-launch state if cycled at

any time within the first 30 minutes after the switches close (including but not limited to radio frequency transmission and deployable system timers).

6) The CubeSat deployment switches shall be captive.

7) The force exerted by the deployment switches shall not exceed 3N.

8) The total force of all CubeSat deployment switches shall not exceed 9N.

I thought it would be easiest to implement plunger switches as:

* We have more physical space at the ends of the CubeSat, compared to on the rails.
* It follows what Endurosat does, so we can mimic their set up, and we know that set up works.
* There are already several holes in the rails and I do not really want to add more holes to them, which may further mechanically weaken them.

Below are pictures of the Endurosat system and the bottom cover plate for our CubeSat;

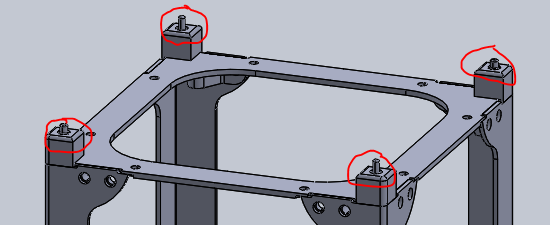


Figure : Endurosat plunger deployment switches

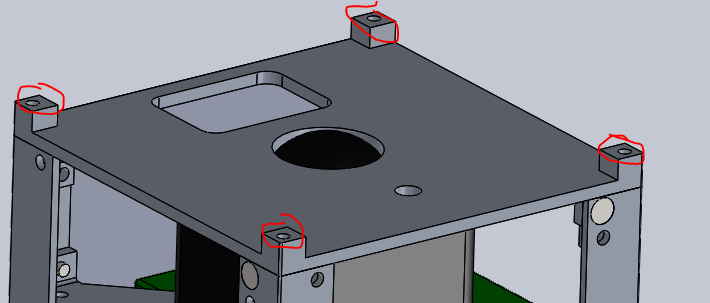


Figure : Holes for the plunger switches for our CubeSat.

As you can see, the plunger switches are not actually in the model yet, as their selection has not been finalized. They can have a diameter of up to 4 mm.

The cross section of the plate is shown below:



Figure : Cross sectional view of the bottom plate

The diameter marked by the red dimension is currently 3 mm, as we found a plunger that was 3 mm.

I don’t think that you’d want this to be too much wider though as most of the loading during launch will be carried by those end faces and they need enough material to support the loads. The hole could also be counterbored if needed to accommodate a switch.

There are a two main methods by which we could do the plunger switch:

1. Have a plunger switch, as Endurosat does. The holes could be counterbored if this was the method chosen, as in figure 4.
2. Have a plunger trigger another switch – push button, plunger switch, roller switch, etc, as in Figure 5. For this option, we would not have to have the lever, we could have the plunger directly press a button or something. Alignment here is key.

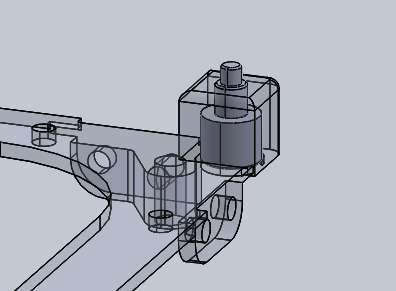


Figure : Endurosat's plunger switch. It fits into the space in the rail end.

From the CAD that Endurosat sent me.

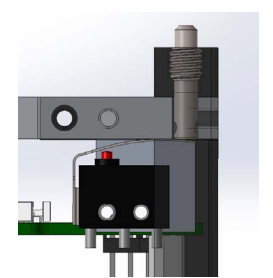


Figure : Plunger pressing a lever switch. From [1]

Both options have their advantages and disadvantages. The Alaskan team [1] had issues with their plunger getting stuck in the retracted position during the vibration testing. This would be a system failure. However, it is very difficult to find an actual plunger switch that is space rated that is small enough (I have not been able to find one so far…). Endurosat did not want to disclose which one was used for their system.

Option 1 would mechanically be much simpler as there would be no additional mounting requirements for the switches, they would just have to be screwed into the cover plate before mounting it.

The circuit below was provided by Nanoracks in the deployment interface document, and is a requirement. From this, I am inferring that the deployment switched plug into the ESP.

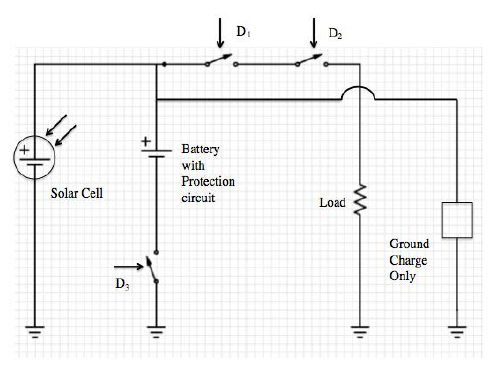
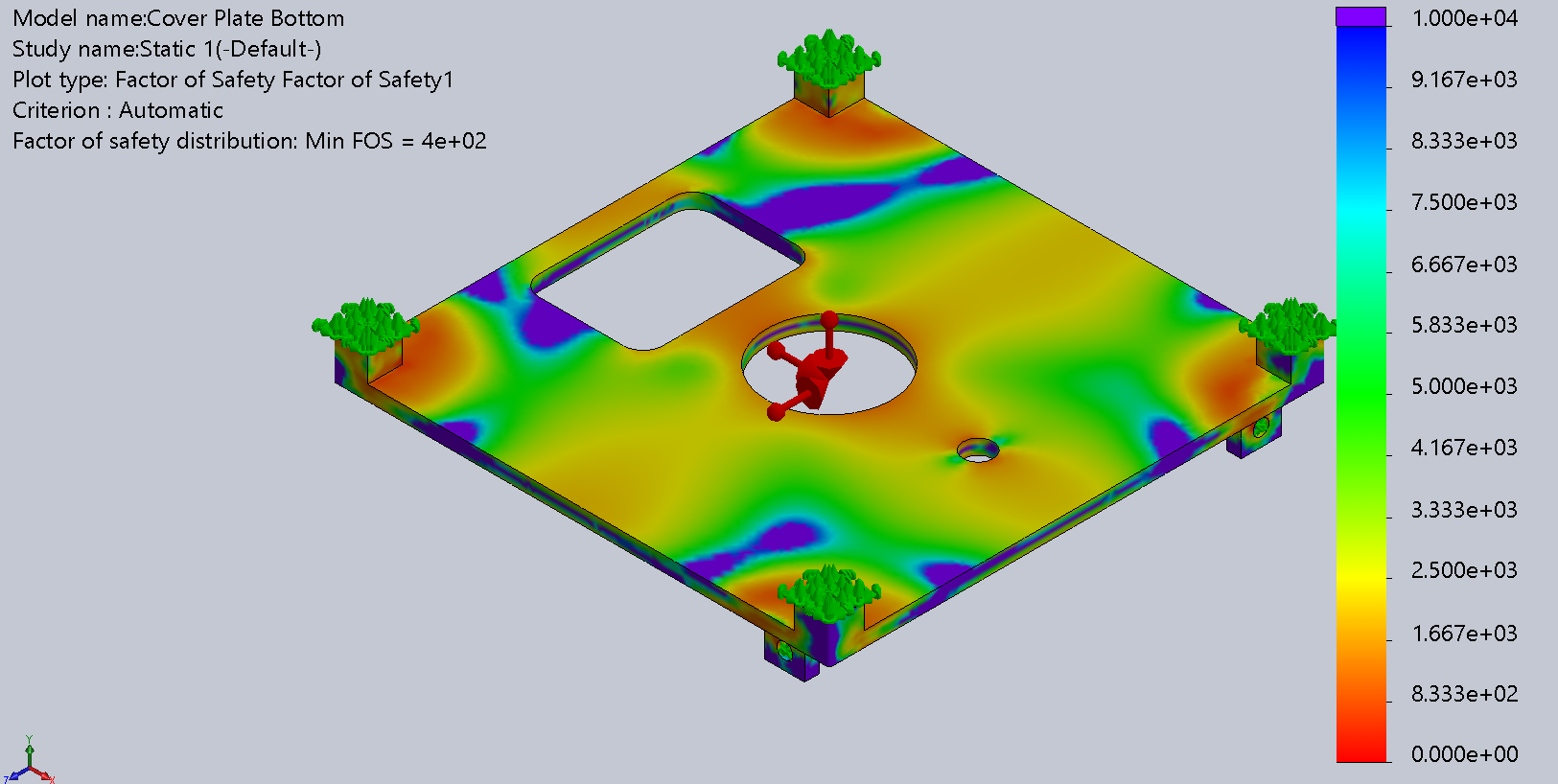


Figure : Nanoracks requirements on how to wire the deployment switches.

The next steps for the deployment switches would be to finalize the hardware that will be used, add it to the SolidWorks model, and verify where the switches connect electrically. It is important to verify that the forces match the nanoracks requirements for whatever hardware are selected. It also should be confirmed if the switches must be NO or NC… the circuit from Nanorack looks like NO, but that does not make sense to me.



The cover plate was tested under the gravitational loads with the ends fixed and hinge restraints where the bolts would fix the plate to the rails. The minimum FOS is 400, when the holes for the deployment switches are 5 mm, indicating that this is acceptable.

Despite the FEA still having a large FOS, I would not have the holes for the plungers larger than 5 mm, as that only leaves 1 mm of material on either side, and this seems small.

TO DO

This needs to have a circuit developed for it that integrates with a timer. See the correspondence with CSA in Appendix BB.

From the CSA information, we can see that we need the timer in the deployment switch circuit before the load:

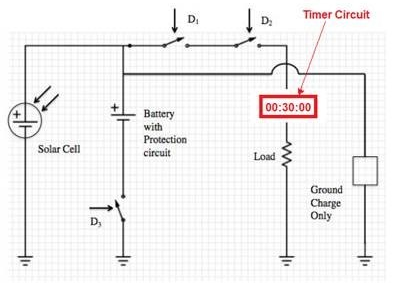


Figure : Deployment switch circuit with placement of timer

For the timer, the CSA said that we could use a real time clock, a 555 timer (with added redundancy), or a microcontroller (for a software solution).

<https://www.electronics-tutorials.ws/connectivity/real-time-clocks.html>

<https://www.electronics-tutorials.ws/waveforms/555_timer.html>

I’m still not sure about the deployment switches – they have to physically break the power circuit, yet they also have to be able to reset the timer if any are pressed again. The timer can only start when all three are released.

Next steps:

* Find deployment switches – either plunger switches or plungers to hit a microswitch, and put it in the CAD
* Select a timer
* Design the circuit
* Determine how the timing circuit is going to be powered
* Determine if the deployment switches need to plug into the ESP
* Built a prototype of the circuit and test the timer

# Antenna Deployment investigation & preliminary research

The first system I looked at for the antenna deployment was GOMSpace’s interstage platform with the antenna deployment option, as we were going with their antenna.

However, when I put the model into the CAD of the CubeSat, it did not fit with the rest of the design.

So, after discussing with the rest of the team, it was determined that it should be straight-forward enough to build our own, as the Carleton team did so [2].

The Carleton team’s antenna design was different – they had their antenna coiled within the cubesat. However, the release mechanism could be similar. They decided to use a burn wire (same as the GOMSpace innerstage platform would have used) to release a door on their antenna structure. In our case we would just release the antenna. We could follow a similar circuit to the one they (Carleton) developed.

The circuit is quite simple – the only component required for the burn wire is a resistor at the location you want the burn wire to snap. The Carleton team included a MOSFET for each antenna ‘branch’ so that the power could come directly from the ESP and would not be drawn from the microcontroller. (They decided to use a separate microcontroller for this circuit due to limited GPIO, I am not sure if this is necessary for us or if we could use the OBC). They also included an overall MOSFET that allos the others to be active or not.

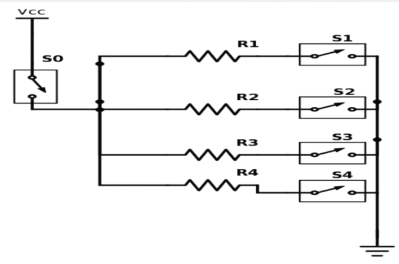


Figure : First prototype circuit. Each resistor is for the burn wire for a separate antenna branch.

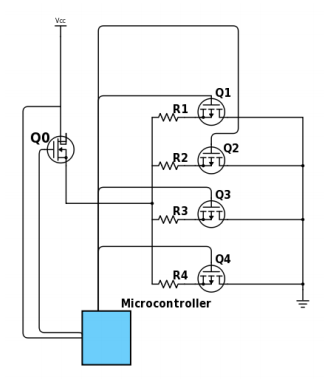


Figure : Actual Circuit - mechanical switches replaced with MOSFETs

The microcontroller would have to interface with the OBC through digital IO or a serial communication channel.

Ideally, there is a back up circuit for all antenna branches in case the first one fails, so the circuit depicted in figure 8 would be duplicated. There also has to be feedback from the antenna deployment (Stephen was looking into switches for this) to know if the antenna was successfully deployed or not. The GPIO required is summarized in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | Pin | Quantity | Total |
| Allow burn wire circuit to be active | 1 output | 2 | 2 O |
| Burn wire for branch | 1 output | 8 (4 branches, 2/branch) | 8 O |
| Check antenna release | 1 input | 4 | 4 I |
| Coms with OBC (if not using separate MC) – trigger signal | 1 input | 2 | 2 I |
| Coms with OBC (if not using separate MC) – response signal | 1 output | 2 | 2 O |
|  |  | TOTAL | 12 Output  6 Input  = 18 GPIO |

My thought was that the OBC would trigger the microcontroller, it would enable the first burn wire circuit, then one at a time, burn the wires for each antenna branch. If it didn’t work for any of the branches, the second burn wire circuit would be attempted. Then it would respond to the OBC with the status of the antenna.

The Carleton team also performed an experiment with the burn wire to determine how much time it requires with what current to burn. The results, summarized below, show that the smaller the resistor value, the more power used, but the less time it takes to burn through the wire. They used Dyneema wire. We should perform our own version of this experiment with our own set up and wire and antenna deployment method. Then we can know how long the microcontroller needs to keep the switch active for.

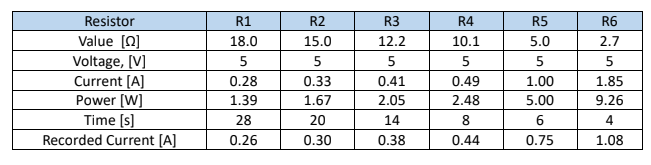


Figure : Results of different resistors with burn wire using Carleton's set up.

Mechanically, the antenna needs to be held against the side of the Cubesat. This interface still needs to be developed. The following is taken from the Gomspace interstage datasheet:

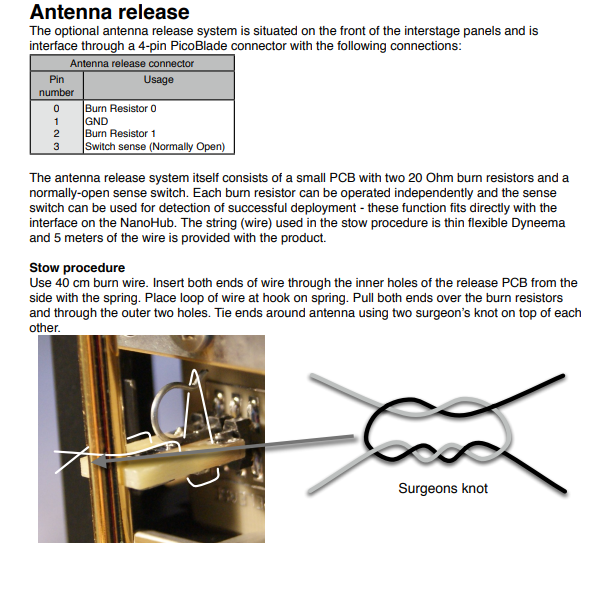


Figure 11: <https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanoutil-interstage-12.pdf>

Next steps:

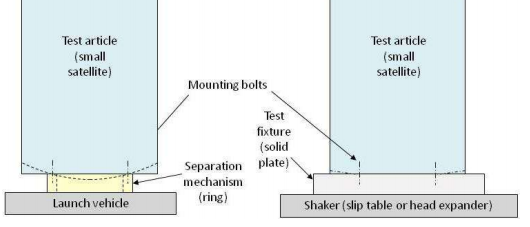
* Finalize the circuit design
* Mechanical design
* Prototype circuit
* Perform experiment to determine time it takes to burn the wire
* Prototype mechanical deployment

# Vibe testing sensor placement

<https://www.instarengineering.com/pdf/Instar_VibVCCVBration_Testing_of_Small_Satellites_Part_2.pdf>

## Fixture

We will likely have to design our own fixture for the cubesat. We will have to do this with a couple points in mind:

* MATCH THE BOLT PATTERN OF THE TEST FACILITY
  + You do not want to get there, only to find out you cannot attach the fixture.
* High stiffness
  + Ideal fixture has no modes below 2000 Hz.
* Light weight
  + The shaker will have a max force rating, F = ma. There will be a target test acceleration. The fixture mass may or may not be an issue. Need to check with the testing facility. [3]
  + [3] recommends using Al 6061 T651 to make the fixture
* Flight like interface
  + Cannot be too rigid
  + Mimic the flight interface geometry - 
  + EX. If the launch vehicle’s interface doesn’t cover the entire satellite, the test one shouldn’t, otherwise the natural frequencies will be affected and the satellite won’t be tested properly.

The CSA said that they would be posting CAD for a interface that is similar to the Nanoracks deployer. They have not done that yet at this point. We should use this model though (we will still have to build our own), as the design should be adequate if it’s from them.

**Mass Simulation**

* Everything with mass greater than 1% of total predicted mass should be simulated
* Items with distributed mass (ex harnessing) can be distributed to other items (ex batter that weighs 150g will be 180g)
* The masses should have appropriate center of gravities and mass moment of inertias

It should be noted that the panels and the bolts are not likely to see true flight loads because the stiffness of the loads are not accurate.

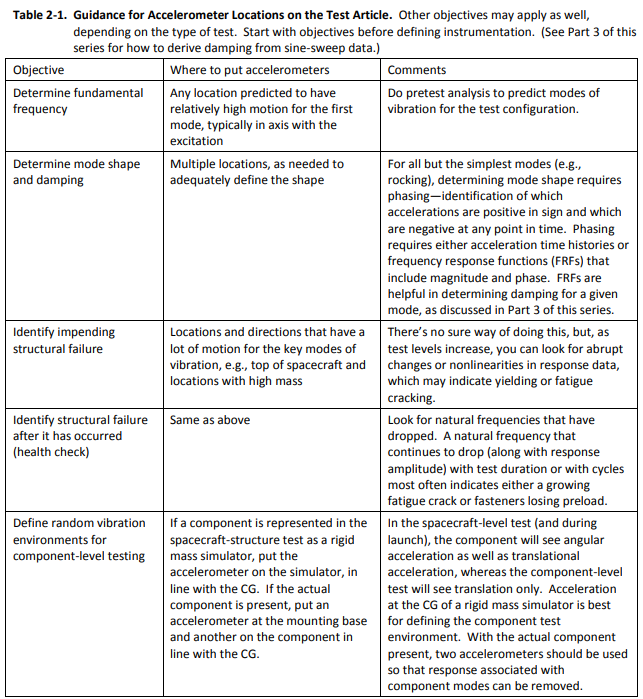
For testing the cubesat’s structure, the dummy masses will be

**Accelerometers**

There should be at least 2 control accelerometers mounted on the test fixture, close to the test item. These provide the control feedback for the shaker table to ensure that the desired profile is being met. These are for the axis that is being excited.

There should be 2 control accelerometers, one for each other axis that are not being excited, just in case something is wrong, it could potentially be detected. Again, these are on the fixture, close to the test item.

[3] provided a nice table that summarized where to put the accelerometers on the satellite itself. The location of the accelerometers depends on the objectives.



We should be putting accelerometers in all three axis on:

* The cameras
  + These are really critical
* OBC
* Center of the rails

These could be in the center of the plate… that would give the most movement, or by the bolt holes.

To mount the accelerometers, use glue and Kapton tape.

Nick, Lauren and I came up with spots to put the accelerometer.

Next steps would be:

* Determine the mass of the dummy masses
  + It should be slightly higher than the mass of the component it’s representing to help account for harnessing and what not.
* Create drawings for the dummy masses so that they can be machined
* Determine what testing facility we are using so that we can confirm how many data acquisition points there are, and we can confirm our selection of the accelerometer locations.
* Get the bolt pattern from the facility and modify the deployer model from CSA to account for that.