Project MESSI

(Microgravity Experiment for Spirulina as Superfood In-Vitro)

Student Team:

Menno Boogaard, Edmundo Colorado, Donald Comstock, Kody Kidder, Sean Kim, Jessy Law, Frederick Missel, Michelle Munasinghe, Justin Nebel, Carlos Perez, Morgan Shilling, Philip Strauss, Adam Townsend and John Trzinski

Faculty Member:

Dr. Pedro Llanos

1. List of Acronyms Used in Report

ABS	Acrylonitrile Butadiene Styrene
APDS-9301	Luminosity Sensor
CAD	Computer Aided Design
CG	Center of Gravity
EZI 65	Rocket name
FRS	Filament Runout Sensor
GRD	Ground
IMU	Inertial Measurement Unit
LED	Light Emitting Diode
MPH	Miles Per Hour
MPU-950	Motion Processing Unit
MUX	Sensor breakout
NFF	NanoRacks Feather Frame
PLA	Polylactic Acid
SCL	Serial Clock
SDA	Serial Data
SD Card	Secure Digital Card
Si7021	Temperature Sensor
USB	Universal Serial Bus

Structures Team with Documentation

Progress and Procedures

Goals from previous progress report:

- Reduce weight by 20 grams
- Stress test Version 4
- Print 3 final versions for flight hardwear, backup and control
- Identify any new needs from the Zebrafish, Spirulina, and Electronics team

Print Versions:

- Version 1
  - Print 1
    - Issues and changes:
      - The print was done with the incorrect material, Polylactic acid.
      - (PLA) was used instead of Acrylonitrile butadiene styrene (ABS) Premium.
      - We decided that the divider between the avionics and the Zebrafish was not enough to protect the Avionics from foam debris that would likely crossover.

- Version 2
  - Print 2
    - Issues and changes:
      - Divider had the wire holes in the wrong location.
      - Electronics identified that the divider made it hard to install the avionics.
      - We found that the dividers for the Spirulina were not strong enough and when we tried to remove the infill from the structure it punctured the dividers.
      - We adjusted the lid mechanism and our plans for how we were attaching the lid. We went from a taped design (lid version 1) to a sliding design Lid Version 2).
      - Found that the divider between the spirulina/Zebrafish and the avionics was not strong enough to put sensors on.
      - The length of bolt holes were too long and they needed to be shorted to reduce weight.
      - Avionics were placed too close together for the team to easily assemble the avionics.

- <u>Version 3</u>
    - Print 3
        - Issues and changes:
            - The only issues with Version 3 and Print 3 was weight, we were under our goal of 200 g for structure mass, at 148 g for the base, but avionics and Spirulina had grown in weight so we had to remove 20 g.

- <u>Version 4</u>
    - Print 4
        - Issues and changes:
            - Print 4 had an unknown issue with the printer during the print. It is possible that x-axis stepper motor was failing.
            - We rounded the external corners to reduce the mass.
            - We changed the skin thickness to reduce the mass.

    - Print 5
        - Issues and changes:
            - Found that the inside dimensions from avionics structure and the zebrafish structure side walls of print 93.3mm in print verses 93.9mm.
            - We also found that the interior of the lid bracket was 80.9mm in print 5 verse 83.0mm in print 3.

- <u>Version 5</u>
    - Print 6
        - Issues and changes:
            - We found that the print shifted when it was being printed, this was due to the BuildTak surface moving during the printing process.
            - We had issues with the mass being much higher than expected and this was assumed to be from a software update.
            - We changed the size of the screw holes to better accommodate the screws.
            - We changed the width of the screws for the accelerometer in the avionics bay.
            - We turned the screws 90 degrees to better accommodate the new sensor.
            - We adjusted the location of the screws in the Spirulina section so that they were better centered.
    - Print 7
        - Issues and changes:

- There were no issues observed for print 7 and the mass was reduced to 130g.

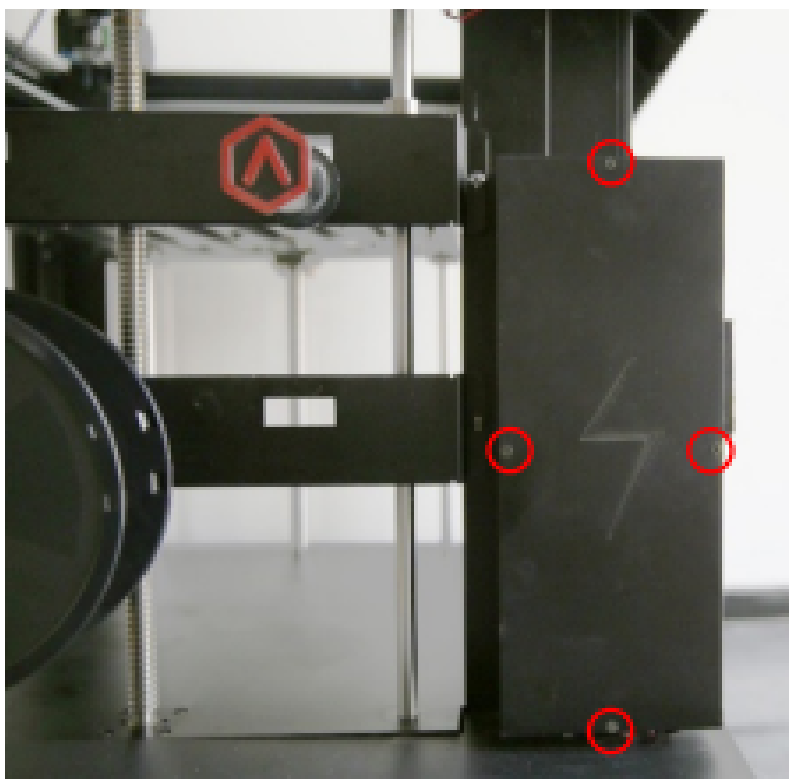
<u>Stress Tests Setup:</u>
The minimum structural requirements for the structure test are:

- A 7 kg weight was on the top and bottom with top being as the square side with the zebrafish and avionics and the bottom being defined as the square side with the Spirulina.
- A 5 kg weight was on the sides, the sides are defined as the base that is opposite of the lid, lid where the lid structure slides on , avionics which is the long rectangular side that the avionics bay is on and zebrafish which is the long rectangular side that the zebrafish experiment is on. The 5 kg is placed at the center of the side, to the left of the side and to the right of the side. The left is defined as the side of the structure with the Avionics and zebrafish, the right is defined as the side with the Spirulina.
- If the 5 kg on the sides works we then test the Print with 7 kg on the sides as at the center, left, and right as well.

<u>Steps for Adding the Filament Runout Sensor:</u>

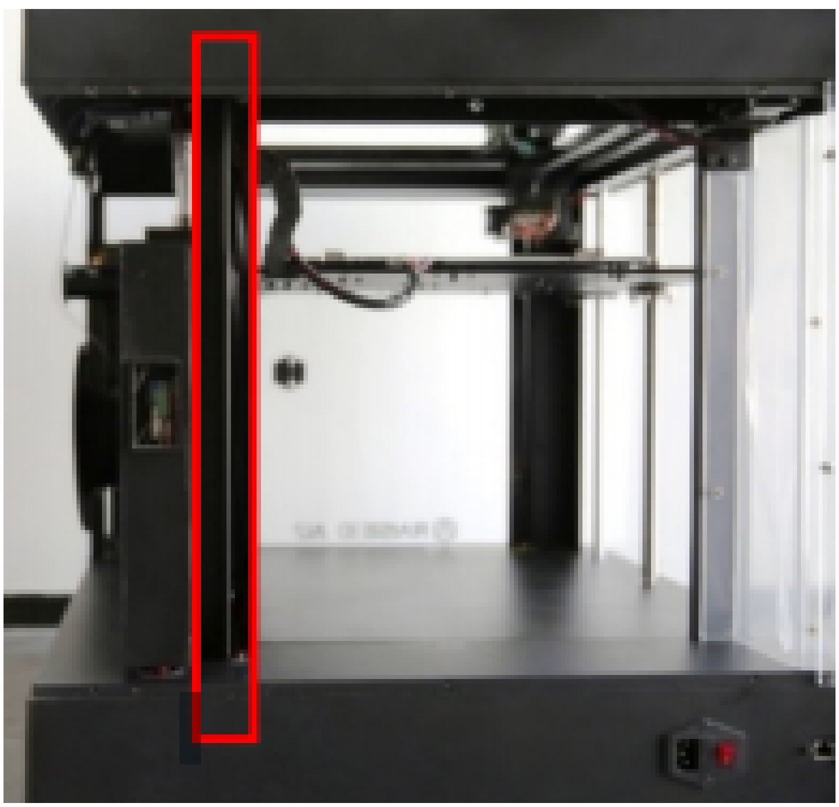
1. Fully disconnect power to the printer by turning the switch to the off position and unplugging the power cable.

2. Use a 2mm hex wrench to remove the 4 fixing screws on the electronic box cover. Remove the cover and set aside.

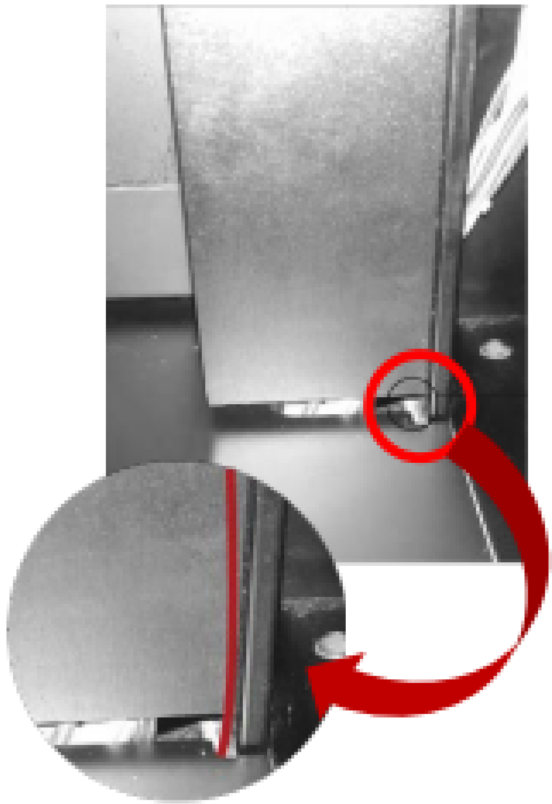


3. Use a phillips head screwdriver to remove the screws that secure the side door and rear panel. Remove the panel assembly and set aside.

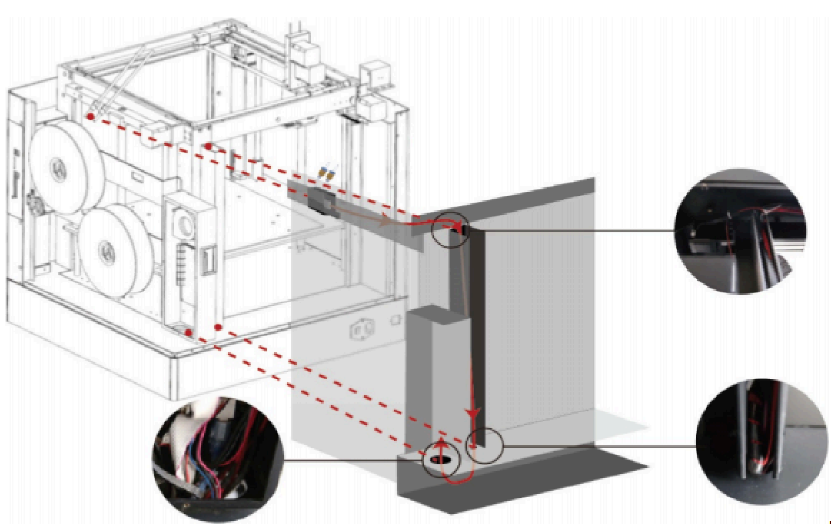
4. Remove cover on cable path.



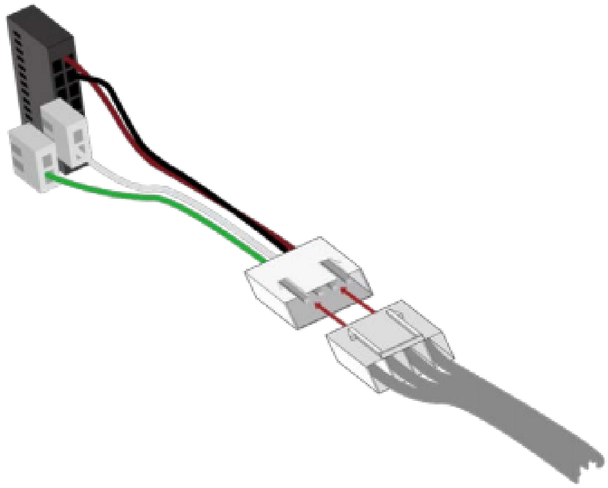
5. Insert the filament runout sensor cable into hole between the wire path and electronic box. Pull wire through the electronic box.



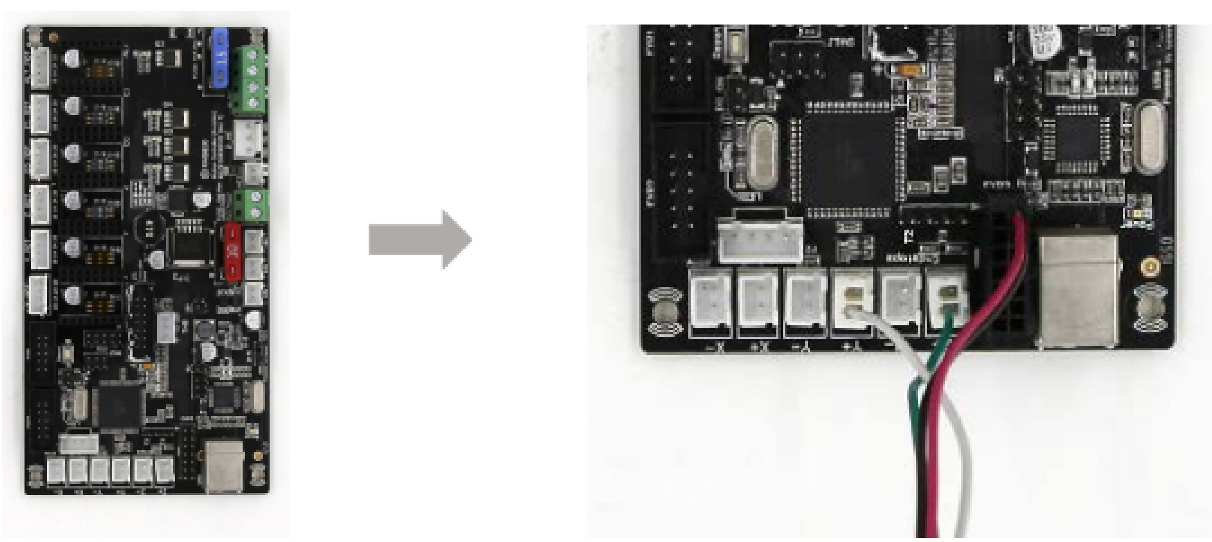
6. Pass the filament runout sensor cable throughout the cable path along the printer and loop into the electronic box.



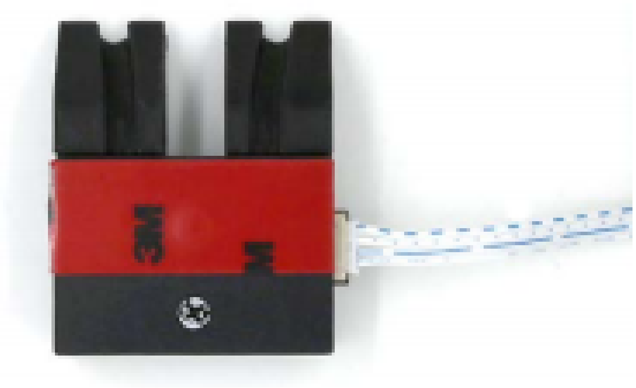
7. Use the adapter (C) to connect the filament runout sensor to the motion controller boar with the 4 pin end.



8. Connect the other three ends of the adapter to the motion controller board as follows:
    a. Connect the red-**black** cable to EXP3
    b. Connect the **white** cable to Y+
    c. Connect the green cable to Z+

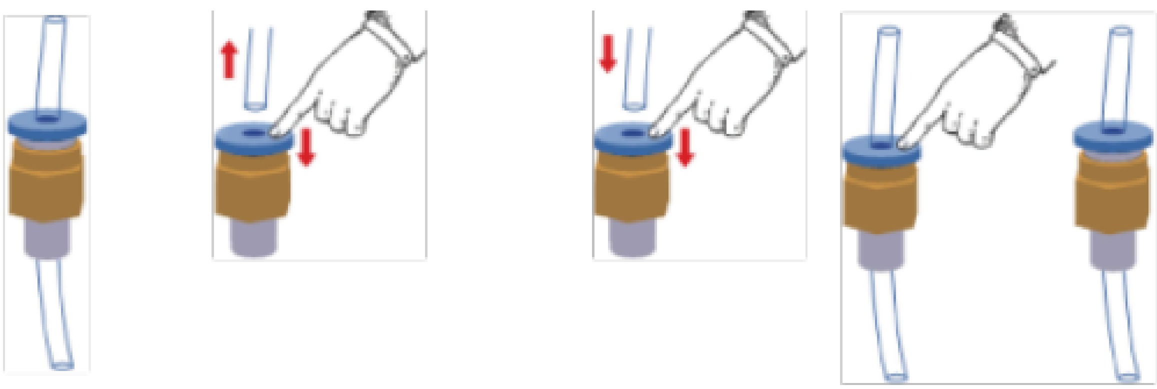


9. Connect the filament runout sensor cable to the filament runout sensor.

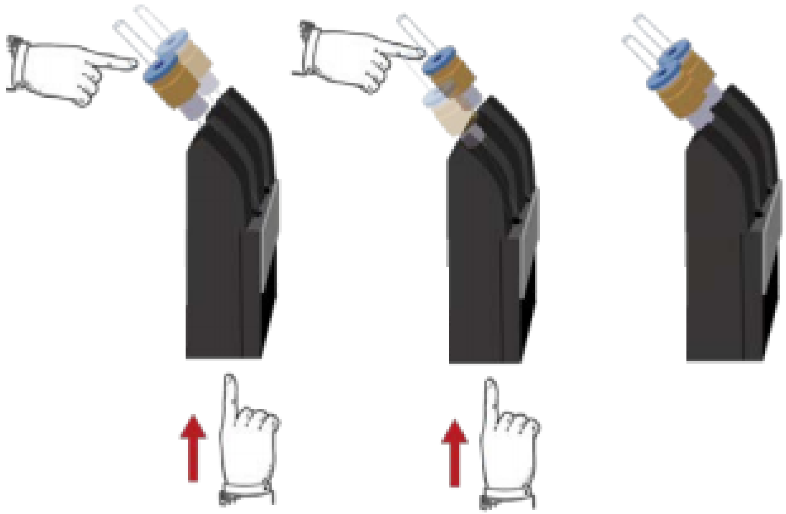


10. Restore power to the printer to ensure that the red light on the filament runout sensor is functioning. If the light turns on, power off the printer. If the light does not turn on, contact support at help.raise3d.com.

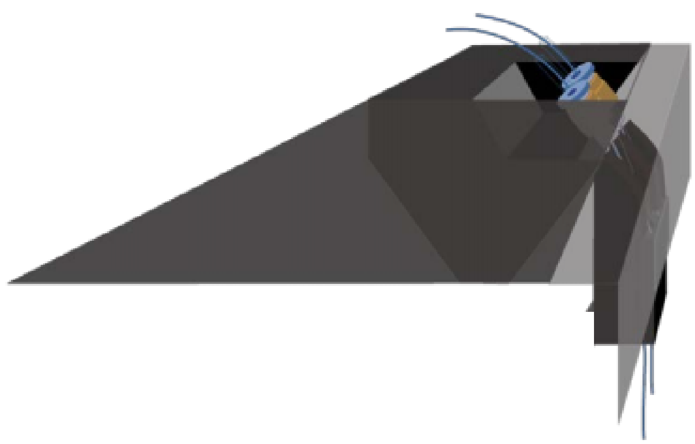
11. Remove the bowden tubes by pressing the blue ring on the coupling. Push the new bowden tubes into the coupling. 1 to 2 inches should be protruding from the bottom of the coupling, in order to attach to the filament runout sensor.



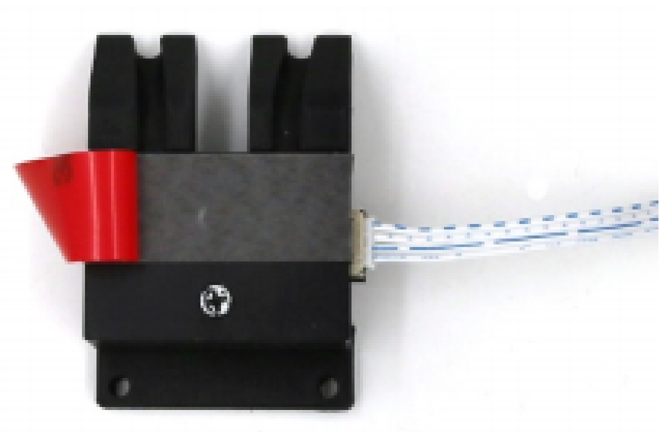
12. Insert the bowden tube ends into the two holes in the filament runout sensor.



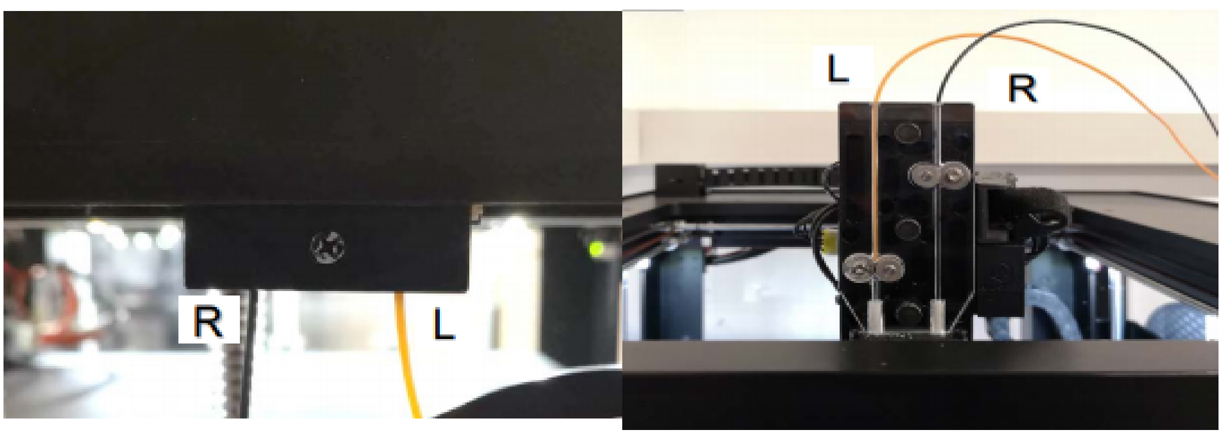
13. Move the bowden tubes upwards by pushing upwards on the filament runout sensor. Adjust the height until the top of the bottom screw is flushed with the edge of the panel. The top screw of the filament runout sensor should be completely hidden.



14. Remove the tape covering on the filament runout sensor and adhere to the panel.



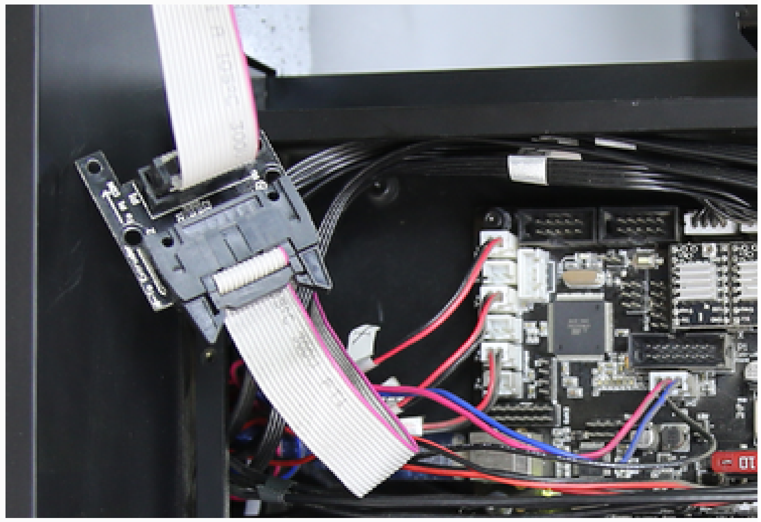
15. Load the filament through the filament runout sensor and up through the bowden tubes.

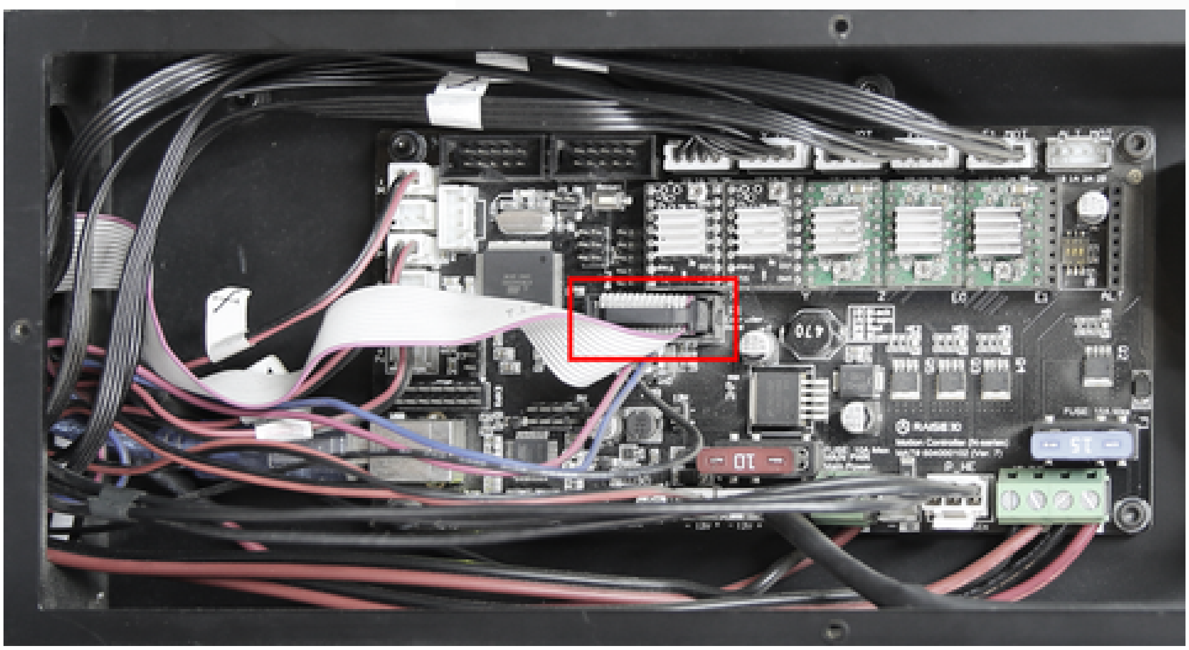


16. Reattach the wire path cover, electronic box cover, and side door/ rear panel assembly with the respective screws.
17. Download Motion Board Firmware with Filament Runout Sensor (FRS) version and version 0.9.7 (Raise 3D). This is found at: https://www.raise3d.com/pages/filament-run-out-sensor-installation-instruction.
18. Download version 0.9.7 of the Motion Board Firmware. This is found at: https://www.raise3d.com/pages/download.
19. Save the two firmwares to a USB drive and insert it into the printer. Reboot the printer and go to “Motion Controller Firmware Update,” for instructions on how to update the motion board firmware.

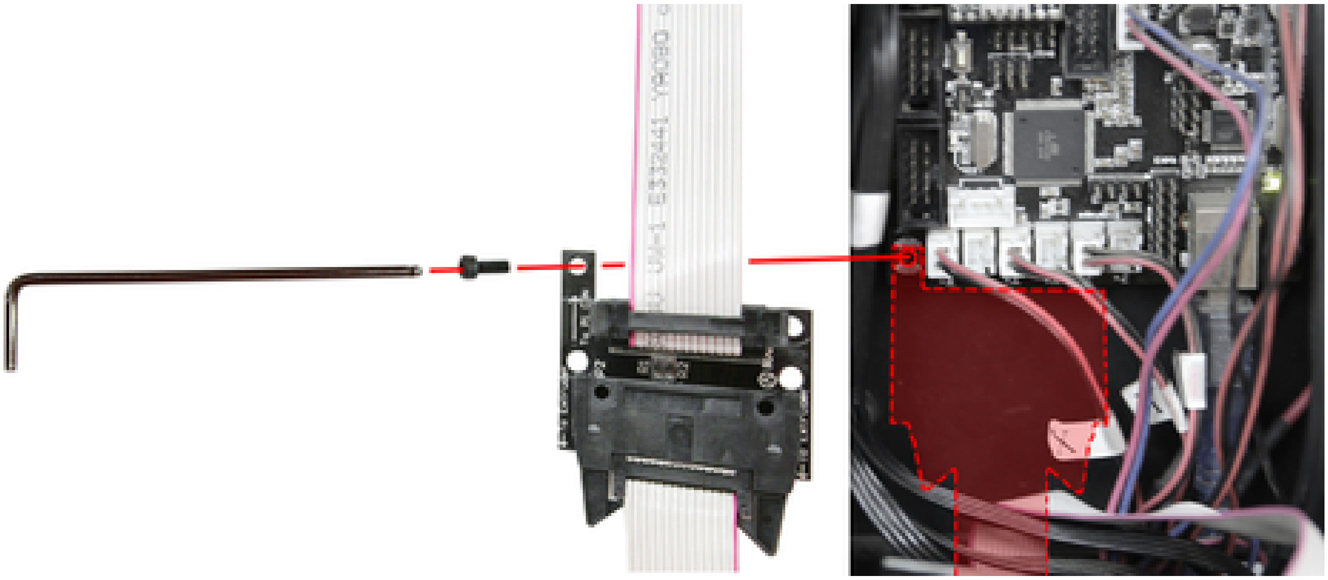
Steps for Adding the Thermal Protector:

1. If the electronic box is not open, complete steps 1 and 2 of filament runout sensor installation.
2. Remove the ribbon cable from the motion controller board and connect the end to the thermal protector.

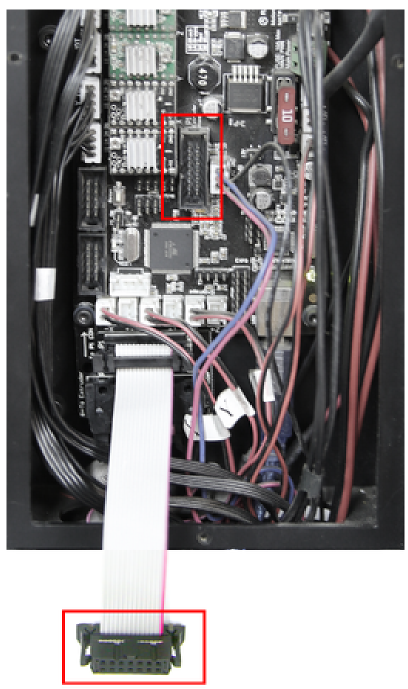




3. Use the existing screw on the bottom left of the motion controller board to attach the thermal protector.



4. Connect the other end of the thermal protector to the space where the ribbon cable was previously connected.



5. Reattach the electronic box cover with its respective screws.

Steps for Degreasing and Greasing the Printer:

1. The first step in degreasing the printer is to grab put on gloves.
2. After gloves are placed on, grab a medium sized cloth, preferably one that is not very fiberios. This will decrease the potential for stings being stuck to the metal rods and lead screws.
3. Take the cloth and wrap one end to the end of your pointer finger. The other end will be held by your other hand.
4. In a slow motion move one hand forward as the other hand moves backwards This will help remove all grease.
5. As the cloth is being moved forward and backwards, move your hands up and down the side of the lead screw or rods. This will ensure that you cover the entire surface.
6. When removing the the grease make sure the places where grease is taken away is memorized. This ensures that when reallying the same amount of grease is on the rod or lead screws.
7. The first step to regreasing is to take get a small amount of grease with you finger and apply it to the bottom of the rod or lead screw.
8. After a small amount of grease is on, evenly distribute the grease along the surface of the rod.
9. This can be done by pinching the rod and moving your fingers up.
10. Most important step is to make sure that the same grease removed earlier is applied in the same area and same amount.

Steps for Cleaning the Printer:

1. After removing and reapplying grease, the printer must be dusted off.
2. To dust off the printer, make sure it is turned off.
3. Next, remove the top cover, his give you better reach.
4. Then take the DataVac ED500 Electric Duster and plug it in.
5. Move the duster slowly over the surface. This will ensure that all dust and particulate matter is removed.
6. Another step in cleaning a printer is to ensure that the BuildTak surface is removed of all glue and plastic residue.
7. To remove such substances, take a cloth and apply alcohol to it.
8. Then place the cloth on the surface and rub the cloth in a circular motion.
9. After the surface is cleaned and is not sticky anymore, let the surface dry.
10. Take the BuildTak surface and place it in printer again.

10/16/18:

- Removed Print 5 Version 4 from the printer.
- Removed extra structural pieces added by the printer on Print 5 Version 4.
- Before all extra structural pieces were removed the the main structure is 180g.
- After removal of the extra structural pieces were removed the the main structure is
- Changes requested by the electronics team:
    - Wire channels leading to the Spirulina to allow for power and sensor placement.
    - The size of the wire channels is not a major factor for the electrical team so it will be chosen to maximize weight reduction and minimize structural integrity loss.
    - Placement should be on the avionics side of the payload with one hole in each of the three dividers, this allows for the shortest route for the wires.
- Updated bolt placement.
- Place 4 bolt holes on the Zebrafish side of the divider for a humidity sensor.
- Shrink size of the bolt holes to properly fit the bolts.
- Issues found with Print 5:
    - The lid doesn't fit on Print 5 and it looks as if the structure has been printed smaller than it should have been in the x-axis. This issue is likely caused from the 3D Printers issue with the x-axis motor.

10/23/18:

- Stress test of Print 5 conducted with a 2kg and 5kg weight with a maximum of 7kg placed on the top, bottom, and each of the four sides. This mass was also placed on left, right and center of each of the four sides.
    - Stress test on the base is not accurate as it has extra structure from printer error and the the bonding between the structure and the infill.
- Pulled as much of the remaining base off of the structure as we could The act of pulling the base of took the floor of the structure as well.
- Look for cause of x-axis issue that caused Print 5 to be smaller than intended in the x-axis.
- Measured the width of the print 5, from avionics to zebrafish, and the width of the inside edge of the lid bracket and compared it to print 3 and the CAD model to determine if the issues was with the CAD model or the printer.
    - Dimensions of CAD:
        - Width: 93.9mm
        - Lid Bracket: 83.0mm
    - Dimensions of Print 3:
        - Width: 93.9mm
        - Lid Bracket: 83.0mm

    - Dimensions of Print 5:
        - Width: 93.3mm

        - Lid Bracket: 80.9mm
- We lost the CAD drawing for Version 4 and we had to remake the CAD drawing for Version 4
    - We adjusted the skin from Version 3 to Version 4 from 1.5mm to 1mm.
    - We adjusted the partition between the avionics and the zebrafish from 3.5mm to 2mm.
    - We decreased the size of the slot that the partition fits into from 3.9mm to 2.4mm and we maintained the .4mm of tolerance.
    - We adjusted the length of the partition by adding .1 mm to each side to maintain the .4mm of tolerance between the partition and the walls.
    - We adjusted the skin on the lid to 1 mm from 1.5mm and we added the fillet to the edges of the lid so that it matches the the fillet on the body of the structure. We added a fillet of 6.35mm to the lid.
    - We adjusted the width of the nut holes to 5.4mm to better accommodate the 5mm nuts that the team is using. We left .4 mm of tolerance between the nut and the nut hole so that they can fill it with adhesive.

10/25/18:

- The new printer parts arrived for the 3D printer and they were installed.
    - We replaced the stepper drivers
        - After calling Raise 3d we were informed that the stepper motors never go bad and they are a lifetime part. They told us that it is typically the stepper drivers that go bad and cause issues similar to what we have been experiencing.
    - We added a filament runout sensor
        - We did this to prevent a print from being lost if the filament runs out, this sensor will pause the print when it senses that it is out of filament and it will allow us to change the filament before continuing.
    - We added a new glass plate
        - The old plate was worn to the point that we had to use glue, hairspray and a boat to make a print. This new plate should allow us to make prints without the need of glue, hairspray or a boat. It was advised that we still put hairspray down but it does not need to be cleaned after every use.
    - We attempted to add a new temperature sensor to prevent parts from burning out.
        - We were unable to instal the new temperature sensor as the ports on the board had some type of sealant installed in them to prevent them from being removed. After contacting Raise 3D we were told that they due this when they have to ship a printer.
    - We removed all of the old grease from all of the rods and moving parts of the printer and added new grease from to all of the rods and moving parts.
        - After removal of the old grease and the addition of the new grease it was found that the X and Y axis' were hard to move. The incorrect grease may

        - have been added, the type that was added was silicon lubrication grease and it was suggested by Raise 3D.
    - We cleaned out all of the dust and cleaned the glass housing.
        - We found that you should not use an alcohol based cleaner on the walls as it starts to melt the walls. No significant damage was done but if used to clean the printer it could cause damage in the long term.
    - A camera was added that the team can access at any time so we can monitor a print
        - We have had issues with getting the camera to work on school wifi as it is an open network but we are looking for options to get around that.
- We printed a test cube to see if the printer was functioning properly the test cube should have been 20mm x 20mm x 20mm
    - After the print completed successfully we measured the cube and found it to be 17.8mm on the X axis by 20mm on the Y axis by 20mm on the Z axis.
    - It is possible that this could be due to the lower voltage that was given to the X axis motor earlier as a fix for the issue.

10/30/18:

- The main goal was to take apart the printer and find the issue that is causing the decrease size in the x-axis.
    - After pulling off all of the side panels and removing the top panel it was discovered that there were issues with the homing sensors on the X, Y, and Z axis.
        - Once the repair was completed on 10/25/18 the voltage of the motor was restored to its original value so this was not the cause.
        - We found that when the printer goes to home itself by finding the edges it hits the homing sensor at a high rate of speed and it bends the sensor. This was thought to be a possible issue that could be causing the issue. We bent the homing sensor back to its original location.
        - We found that the X-axis motor was lose from the rod that controls the x-axis. After discovering this we tightened the X-axis motor and thought that this might be the cause of our issues.
            - This was likely caused from the dismantling of the printer in search of the issues.
- Started printing test cubes to verify that the adjustments made to the printer solved the last of the printer issues that we were having.
- Attempted to connect the wireless camera for us to check on the print remotely.
    - We were unable to get the camera to connect to network due to an unknown issue.

11/01/18:

- Prep the printer for Print 6.
    - We sprayed the glass bed with Aqua Net hairspray to prevent the print from moving and prepared the print with the new printer software.

- Attempted to connect the wireless camera for us to check on the print remotely.
    - The camera settings were completely reset and we were able to get the camera to connect to a hotspot from a computer.
    - We found out that the camera could not function on the schools network as there are multiple access points and the hardware and software cannot support this.
- Started Print 6 and and informed the team of the print progress and time so that people could check on the print.

<u>11/06/18:</u>

- Identify the cause of the issues from Print 6.
    - In print 6 we observed issues with the print shifting mid print, and with the print being much heavier than is should have been.
        - The mass of print 6 ended up being 156 g when print 3 weighed 152 g. Print 6, according to the computer estimates, was supposed to be just over 20 g less than print 3.
            - The software the for the preparing the print on the computer was updated and after the update we could not get the prints mass to decrease.
                - We changed the software back to the old software that we have used on the old prints.
        - The new glass plate was suspected to have moved. We added more clips to the glass bed to help prevent it from moving.

<u>11/09/18:</u>

- Removed Print 7 from the printer.
    - We observed no issues with the print from the printer.
    - The mass of Print 7 is 130 g and this is 18 g less than Print 3.

<u>Goals Going Forward:</u>

- Print 1 more final versions for flight hardware, backup and control.
- Print 2 dividers (Version 3)
- Print 2 lids (Version 3)
- Make changes to the Spirulina faceplate for wire channels,bolt hole locations adjustments and the diameter of the tube holders.
- Print 3 Spirulina faceplates (Version 2)

Modifications

- Changed the thickness of the divider between the of the Spirulina and the Zebrafish experiment.
    - Old Thickness: 2.5 mm
    - New Thickness: 2 mm
    - Old Width: 95.5 mm
    - New Width: 96 mm
    - Old Hight: 95.85 mm
    - New Hight: 96.1
- Changed the slot depth that the divider between the of the Spirulina and the Zebrafish experiment slide into.
    - Old width: 95.5 mm
    - New width: 96 mm
- Changed the skin thickness on the lid.
    - New skin thickness: 1 mm
    - Old skin thickness: 1.5mm
- Rounded the corners of the lid to match the corners on the structure.
    - Radius of 6.35 mm
- Changed bolt holes on the experiment to better fit the bolts that we have.
    - Old bolt holes inner dimensions: 7.159 mm
    - New bolt holes inner dimensions: 6.004 mm

Summary of 3D Inputs:

	Nanolab 1	Nanolab 2	Nanolab 3	Nanalab 4	Nanolab 5	Nanolab 6

Layer height	0.2mm	0.2mm	0.2mm	0.2mm	0.2mm	0.1mm
Shell width	0.8mm	0.8mm	0.8mm	0.8mm	0.8mm	0.8mm
Infill	10%	10%	15%	15%	15%	15%
Infill speed	40 mm/s	40 mm/s	40 mm/s	40 mm/s	40 mm/s	70 mm/s
Estimated time	32h 45m	29h 10m	30h 6m	29h 1m	33h 23m	69h 34m
Estimated filament used	257g	192.8g	210.3	227.8g	226.4g	193.9g
Heated bed temperature	110 °C	110 °C	110 °C	110 °C	110 °C	110 °C
Extruder temperature	240 °C	240 °C	240 °C	240°C	250 °C	240 °C
Infill pattern type	Rectilinear	Rectilinear	Rectilinear	Rectilinear	Rectilinear	Grid

Payload Mass Manifest:

Key
Not Changed
Changed

1st Generation:

Part	Mass (g)
USB Connector	16
Humidity and Temperature Sensor	1
Accelerometer	2.4
Light Sensor (x3)	3
Arduino	7.5
Multiplexer	7.8
Wires	24
Housing (V3) without lid	148
Lid (V3)	28
Partition (V3)	20
Velcro	3
Zebrafish (Including Container) and	100
Tube Housing	12
Test Tubes (x18)	36
1.5 ml of water (x18)	27
Battery Housing	20
Battery (AA) (x2)	30
LED Lights	xx
Total	**485.7**

2nd Generation:

Part	Mass (g)
USB Connector	16
Humidity and Temperature Sensor	1
Accelerometer	2.4
Light Sensor (x3)	3
Arduino	7.5
Multiplexer	7.8
Wires	24
Housing (V3) without lid	148
Lid (V3)	28
Partition (V3)	20
Velcro	3
Zebrafish (Including Container)	100
Tube Housing	12
Test Tubes (x18)	36
1.5 ml of water (x18)	27
Battery Housing	12
Battery (AA) (x2)	30
LED Lights	1
Total	**478.7**

3rd Generation

Part	Mass (g)
USB Connector	16
Humidity and Temperature Sensor	1

Accelerometer	2.4
Light Sensor (x3)	3
Arduino	7.5
Multiplexer	7.8
Wires	24
Housing (V4) without lid	130
Lid (V4)	30
Partition (V3)	20
Velcro	3
Zebrafish (Including Container)	100
Tube Housing	12
Test Tubes (x18)	36
1.5 ml of water (x18)	27
Battery Housing	12
Battery (AA) (x2)	30
LED Lights	1
Total	**462.7**

Level 1 Rocket Configuration:

Part	**Mass (g)**
Humidity and Temperature Sensor	1
Accelerometer	2.4
Light Sensor	1
Arduino	7.5
Multiplexer	7.8
Wires	10

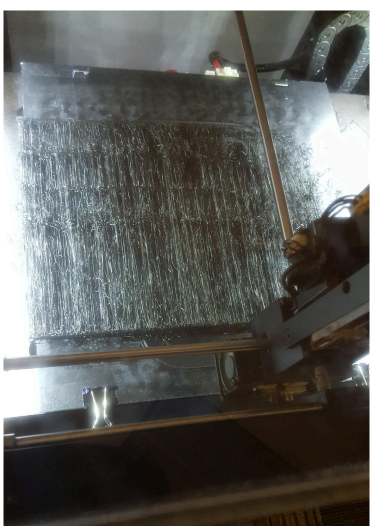
Battery Housing	12
Battery (AA) (x2)	30
LED Lights	1
Total	**72.7**

Photos

<u>Current Version Progression</u>

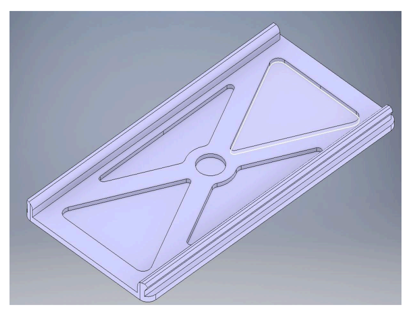
Version 1	Version 2	Version 3	Version 4	Version 5
				
Starting Design And we needed to minimize the mass	Added divider to protect avionics from foam debris.	Created a removable divider and moved some electronics for ease of access	Minimized mass by rounding the corners and shaving off thickness from the skin.	Decreased mass by shrinking the divider and by decreasing the skin of the lid.

Current Print Progression

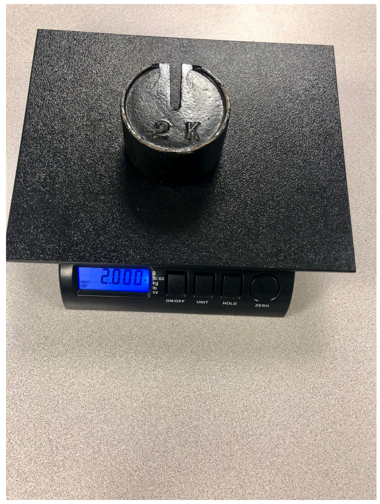
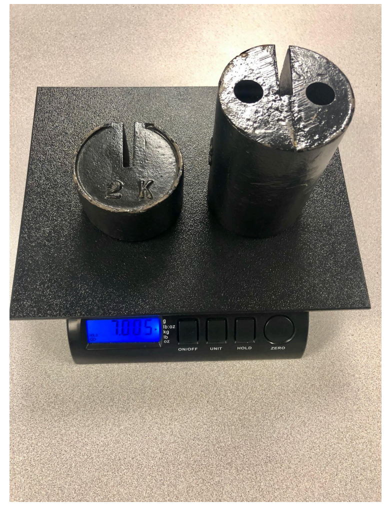
Print 1	Print 2	Print 3
		
Issues with material that it was printed with and that caused warps.	Issues with lid design and print thickness on walls.	Issues with the weight of the print.
(material error)	(printer error)	(design error)
Print 4	Print 5	Print 6
		
Printer x-axis motor failure during the print	Issues with Printer x-axis.	Issues with the total weight of the structure.
(printer error)	(printer error)	(software issue)

	Print 7	
		
	No Issues	
	Final Design	

Lid Version Progression

Lid Version 1	Lid Version 2	Lid Version 3
		

Stress Test Setup

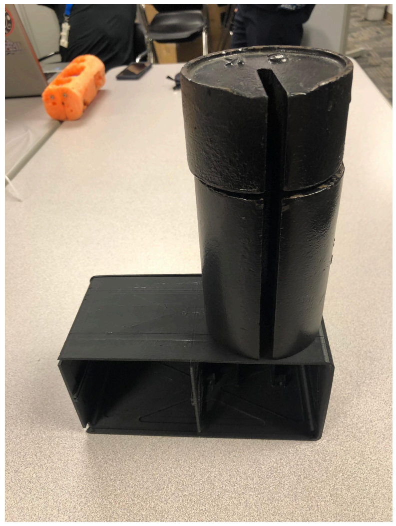
5 kg	2 kg	7 kg
		

Stress Test of Version 5 Print 6

Avionics Left (5 kg)	Avionics Center (5 kg)	Avionics Right (5 kg)
		

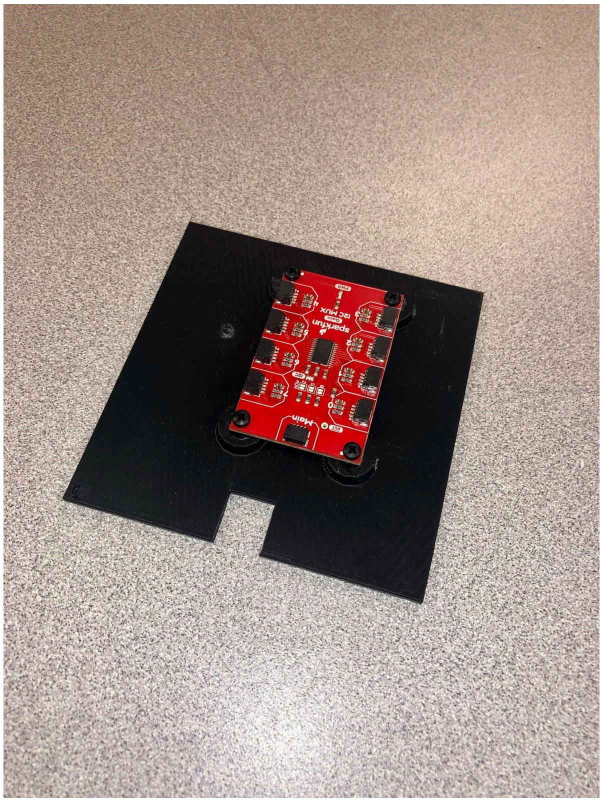
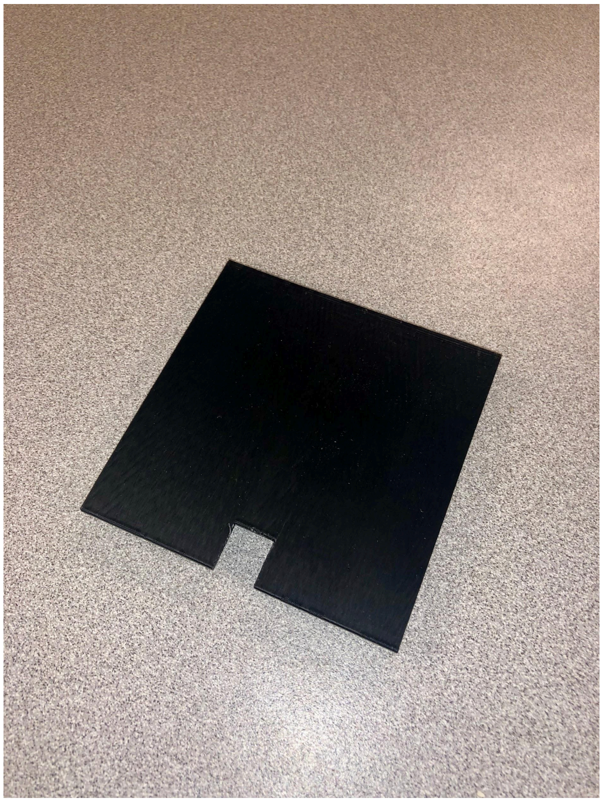
Base Left (5 kg)	Base Center (5 kg)	Base Right (5 kg)
		
Zebrafish Left (5 kg)	Zebrafish Center (5 kg)	Zebrafish Right (5 kg)
		

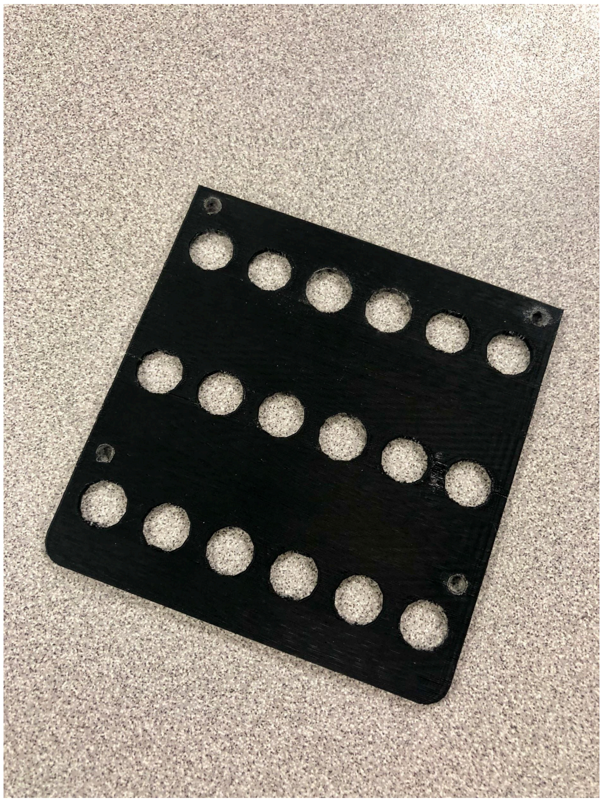
	Avionics/Zebrafish (7 kg)	
		
	Spirulina (7 kg)	
		

Avionics Left (7 kg)	Avionics Center (7 kg)	Avionics Right (7 kg)
		
Base Left (7 kg)	Base Center (7 kg)	Base Right (7 kg)
		

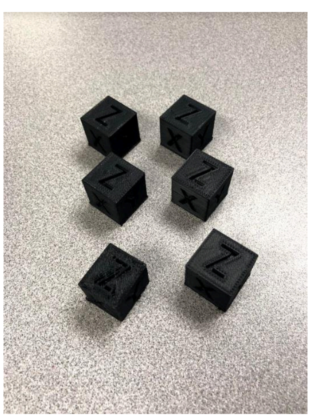
Zebrafish Left (7 kg)	Zebrafish Center (7 kg)	Zebrafish Right (7 kg)
		

Extra Parts

Divider Version 2 Avionics	Divider Version 2 Zebrafish
	

Spirulina Faceplate Version 1	
	

Extra Photos

Test Cube	Print History
	

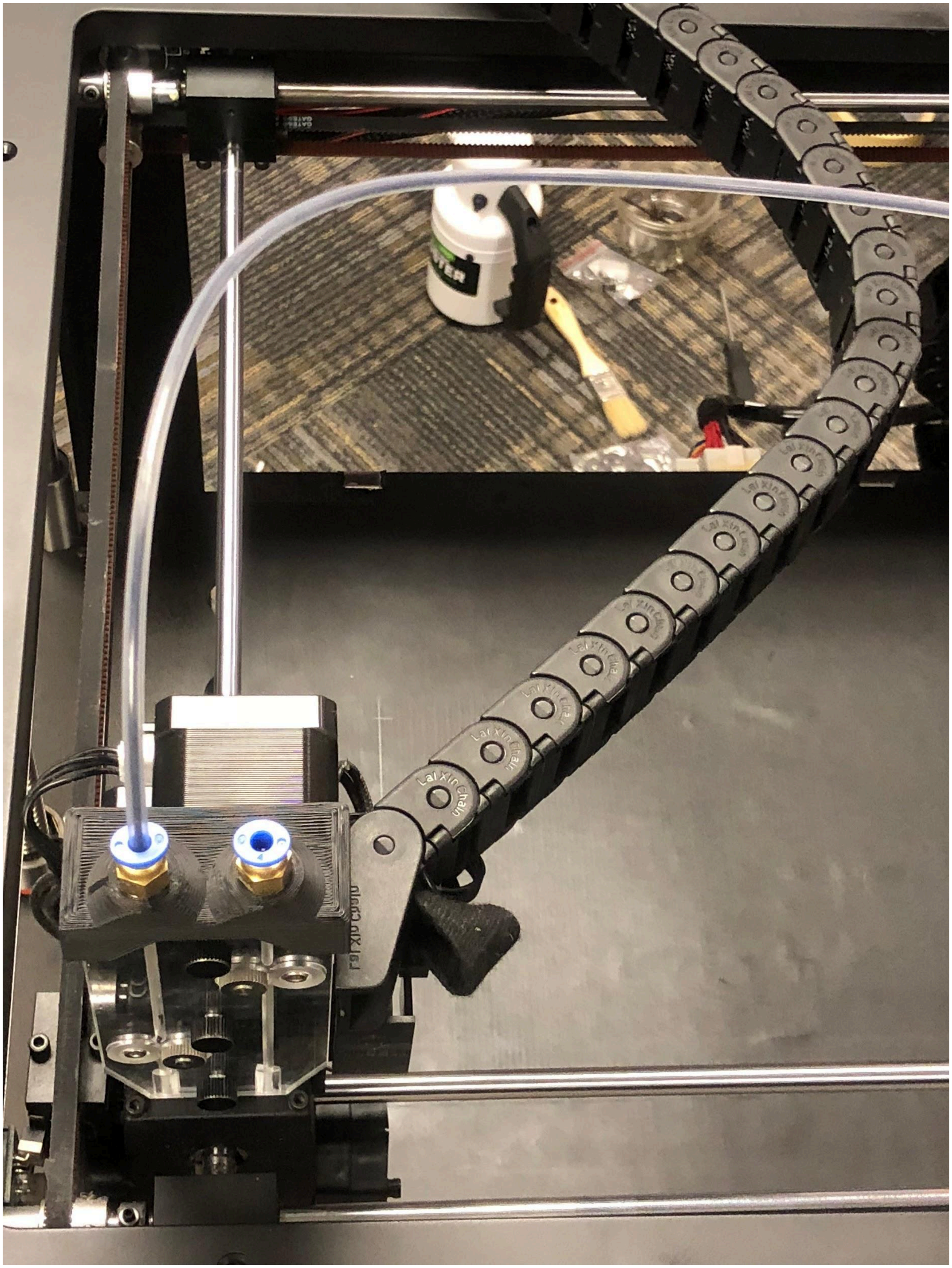
<u>Procedure to Repair 3D Printer:</u>

<u>Repair</u>

1. Removal
2. Lift the printer from the platform
    a. Remove the styrofoam casing from the base
3. Access the back and side panels.
    a. Remove the screws from the back and side panels (28 screws total)
    b. Remove back and side panels
4. Access the door and hinge
    a. Remove the screws from the door and hinge (10 long black screws)
5. Access the top panel of the printer
    a. Remove the screw from the top panel (6 medium black screws without the spacer) (2 medium black screws with the spacer) (7 short black screws) (4 nano black screws)
    b. Make sure to place the screws in separate containers.
6. Make repairs if necessary.
7. Place the printer from the platform
    a. Remove the styrofoam casing from the base

Spools of filament shown as the printer is being disassembled.

Side Paneling of 3D printer disassembly



Overhead view of the 3-D without the protective casing

Assembly

1. Access the top panel of the printer
   a. Place the screws from the top panel (6 medium black screws without the spacer) (2 medium black screws with the spacer) (7 short black screws) (4 nano black screws)
2. Access the door and hinge
   a. Place the screws from the door and hinge (10 long black screws)
3. Access the back and side panels.
   a. Insert back and side panels
   b. Place the screws from the back and side panels (28 screws total)

Procedure to Clean the 3D Printer:

Cleaning

Removing "Old" Grease

- Apply cloth to lead screw (start from the top then go to the bottom making sure each grove is cleaned)
- Apply cloth to metal rods and remove the grease from the rods

Applying "New" Grease

- Wear latex free gloves
- Using "Lude" branded grease apply a quarter size serving on your index finger and thumb
- Moving your index finger and thumb up and down the lead screw making sure each groove is properly lubricated

Freddy cleaning and applying “Lude” on one of the rods in the 3D printer

<u>Electronics Team with Documentation</u>

<u>Electronics Contingency Plans:</u>

Parts, Code, Sensors and Tools

Purpose: A set number of solutions were developed to be used in case of any issue with the sensors or any physical component of the electronics/sensors bay occurred. A plan to have the code of the payload backed up on one or more flash drives which would be used in any case there was an issue with the code already installed. All tools, sensors, cables, accessories and any instructions needed to accomplish any of these contingency plans set forth depending on what is wrong with the payload have been listed in the Contingency Plans Report for Electronics/Sensors where everything else listed in this section can be found. <u>The report along with each contingency plan currently listed may change until the payload is complete, contingency plans may also be added to this document if deem necessary by the team and may also be completely remove from the document.</u>

Components of the Payload under Electronics/Sensors that will be able to be replaced in case of any issue.

1. Arduino MKR ZERO
2. Qciic Mux Breakout (8 inputs)
3. MPU-9250 Acceleration, Gyro Sensor
4. TMP102 Temperature Sensor
5. APDS-9301 Luminosity Sensor
6. LilyPad lights, 3 colors

a. Battery

7. Arduino Code

Contingency: Solutions along with any material associated with that repair will be listed below under the section of the physical components that is thought to be having issues.

1. Arduino MKR ZERO
    a. If the team decides that the Arduino is the source of the problem 2 possible solution have been developed.
        i. Solution 1: Have an Arduino with a clean memory and upload the code straight from the USB flash drive that has the final code for the Payload.
        ii. Solution 2: Have one or more Arduinos as back up with the code already preinstalled, ready to be installed in the payload.

    iii. Solution 3: Replacing the Arduino as a whole should be one of the last options seeing that it would involve opening up the payload to installed the replacement part.

2. Qwiic Mux Breakout
    a. 3 solutions were developed so far if a problem with the Qwiic Mux breakout unit were to occur (Always check for the green light to make sure its receiving power)
        i. Solution 1: A faulty wire connecting the sensors to the Mux might be the issue and can be solved by replacing that wires connected to the Mux.
        ii. Solution 2: The mux Itself might be having issues with replaying all that information to the Arduino in which the Mux can be completely replaced.
        iii. Solution 3: If damage of any kind were to noticed on the Mux while in transportation the team may replace the Mux completely or see if it can be replaced if time is of the essence.

3. MPU-950 Accelerometer, Gyro Sensor
    a. Only one Accelerometer sensor will be placed on the payload and must be operating at 100 percent. Solutions for any issues associated with this sensor will be listed below.
        i. Solution 1: Troubleshoot the sensor by making sure all the connection leading from the sensor to the Mux and finally to the Arduino are all well connected.
        ii. Solution 2: If the team decides that no other solution can be used or problem identified a replacement accelerometer sensor may be installed into the payload. After a systems check making sure signal and data from all sensors is being recorded this solution can be considered complete.

4. Si7021 Temperature Sensor
    a. Four Temperature sensors will be used on the payload, recording temperature of the different algae test tubes during the ascend, zero g and descend phase.
        i. Solution 1: Just like the accelerometer connections from the sensors leading up all the way to the Arduino must be checked to make sure a secure connection is achieved by each cable. Problem might just be a loos wire.
        ii. Solution 2: A problem with the code might be the issue if the code was altered in any way while leading up to the launch, if so, a replacement Arduino with the preinstalled code or a blank one may be installed on the payload and upload the code to the blank Arduino right after.

   iii. Solution 3: If the sensor itself or any of the other 3 temperature sensors have damage of any kind the team may replace one or multiple sensors if necessary.

5. APDS-9301 Luminosity Sensor
   a. Three luminosity sensors will record the intensity of the LED light that will be used to grow the algae. Three different light will be used in three different compartment to grow algae in different lighting condition. Lights Red, Green and Blue will be used. Seeing that only one sensor per lighting condition will be utilized, these sensors have to be working at 100 percent.
      i. Solution 1: Check all the connections from the sensor itself leading all the way to the Arduino must be check to see that all wire is secured. If a faulty wire is found a cannot be repaired and new one may be installed.
      ii. Solution 2: A coding error might be the reason the sensors seem to not be working, If that is decided to be the problem and replacement Arduino may be installed with having the code already installed or have it transferred from a flash drive.
      iii. Solution 3: If the Sensor itself is damaged in any way it may be replaced with a new one making sure it is installed exactly like the original one.

6. LilyPad LED (Red, White, Blue)
   a. These LED light strips are going to be the source of light and nutrients for the algae to grow. Three different LED strip will be used in each of the 3 different testing conditions along with other sensors. Any problem with the LED lights were to happen, the algae might die and no DATA will be able to be collected.
      i. Solution 1: Make sure all connections to power are established with the battery and that the battery is charged. If the battery is not charged a replacement one may be installed if not the one currently installed on the payload may be charged.
      ii. Solution 2: If any of the LED light is not working and power is not the issue a set of replacement LED may be installed but must be aware of the color being installed as well that all connections to power are secure. LED lights will not be connected to the Arduino if not connected directly with power until the mission is over and we're are able to retrieve out payload.

7. Arduino Code
   a. The code used for the Arduino is what is receiving and storing all the data coming from all the sensors and it is vital for the team to see this data to know what happened during the time the payload was in space.

i. Solution 1: Have the code backed up onto one of more flash drives. This code must be the final version and should be an identical copy to the one installed on the payload. This code can be used to transfer onto a blank Arduino if any issue with the Arduino already installed on the payload were to have any issues.
ii. Solution 2: Have 2 Arduinos with the code pre-installed that way they are ready go and don't need a computer to upload anything to it. The Arduino would just have to be installed and have all connections secured.

Progress and Procedures

- Removing unnecessary pins on electronics parts **(Figure S2)**

- Making working table mockup: After unnecessary pins were removed on the electronics, they were connected to each other. The MUX breakout is connected via four wires to each sensor and the same four hwires types are used to connect the Arduino to the MUX breakout. The four wire types are: power (3.3V), ground (GRD) and data ports: SDA (Serial Data) and SCL (Serial Clock). Both SDA and SCL are used to send data from sensors to the MUX and Arduino. For the table mockup some Qwiic wire connectors were changed for Dupont pins. In the payload Qwiic connectors will be used on the MUX and all sensors. The ease of use of the Qwiic connectors are preferred for connecting sensors in narrow spaces in the payload. Only the Arduino there will still be connected with Dupont pins. FInally, SD card simply slots into the Arduino. **Figure S3** depicts the process of adding Qwiic adapters to the sensor to be compatible with the Qwiic connectors. First Pins are soldered to the board, then the adapter to pins.

- Rewriting code to read sensor data & write to SD card: Arduino computer software and the needed libraries for the sensors were installed. Additionally, example codes for the used sensors were downloaded. These were used as a base to write the code to read the sensors. The difficulty in this is combining the code for different sensors and reading and writing the data correctly. This was done to gain a working understanding on how to code in the Arduino software. A beginning was made on modifying the original code written by Vijay. This code only enabled sensors but the part that actually reads and writes data still needed to be added.

- Projected current draw for 9 LED lighting configuration is 270 mAh at 3.0 volts. Two 1.5 volt AA Energizer lithium batteries will be used and will store 3300 mAh at 3.0 volts. Battery holders have been received, will begin testing and integration with structures.

Each battery weighs 15 grams. Runtime with battery system is projected to be 12.22 hours (3300/270=12.22). LED configuration for test launch can be seen in (**FIGURE S5)**

- Integration in the payload structure: The structures team expects the electronics team to provide locations for each part of the electronics and they will produce mounting solutions. No real decisions were made on this yet because not all parts are decided yet. In particular, the LEDs and batteries are not decided yet. This makes it difficult to decide on the placement. These placements will affect the options where other sensors can be placed, making the integration of electronics one of the final steps of the electronics group.

- Communication with the structures team has yielded several structure changes to incorporate sensor updates and location optimization. These changes included modifying the location for securing nut location, and allowing for wiring to be run in the nanolab. There was no significant effect to the center of gravity.

- In order to test the flight worthiness of the avionics, a test set up was prepared. The test set up included all sensors to be launched, as well as an LED compartment in which the light sensors could record the combined light of one of each of the different color LEDs (**Figure S5)**
- The level 1 rocket that was going to be used for the test launch was cleaned of any extra material that might have been left over from the previous rocket. Like rope and parachute components attached to the inside as well as glue and different kinds of hooks.
    - This was done to weight the rocket as well as to find the center of gravity for the rocket.
    - The payload and rocket motor weight were both simulated down to the gram and placed in the section of the rocket that it would be during launch. We did this to simulated the actual weight on the rocket to find the center of gravity and the center of pressure.
- Measurements for the rocket needed by the Apogee rocket packet were recorded to be passed down and used later on in the simulation of the rocket where we could see it trajectory all the way to apogee.

Modifications

- The temperature sensors (quantity = 4) have been changed. The originally selected temperature sensor was the SparkFun TMP102. These are replaced by SparkFun SI7021 combined humidity and temperature sensor. This change was made to add the capability to measure the humidity The form factor of both sensors are similar. As a result this should not increase the difficulty of fitting the sensors in the payload. The wiring procedure of the new sensors is essentially the same. The weight of the old temperature senor and the new temperature and humidity sensor are both 0.9 grams.

- The accelerometer (quantity = 1) is changed as well. The original MPU 9250 sensor was replaced by the same MPU 9250 sensor with Qwiic connectors built in. These connectors are easier to use inside the payload. The new type weighs 2.5 grams vs 1.3 grams for the old one.

- It was decided that Qwiic adapters would be used so that Qwiic cables and connectors could be used. Qwiic products are much more fail proof than dupont connectors, and they have a much more secure attachment.

Photos

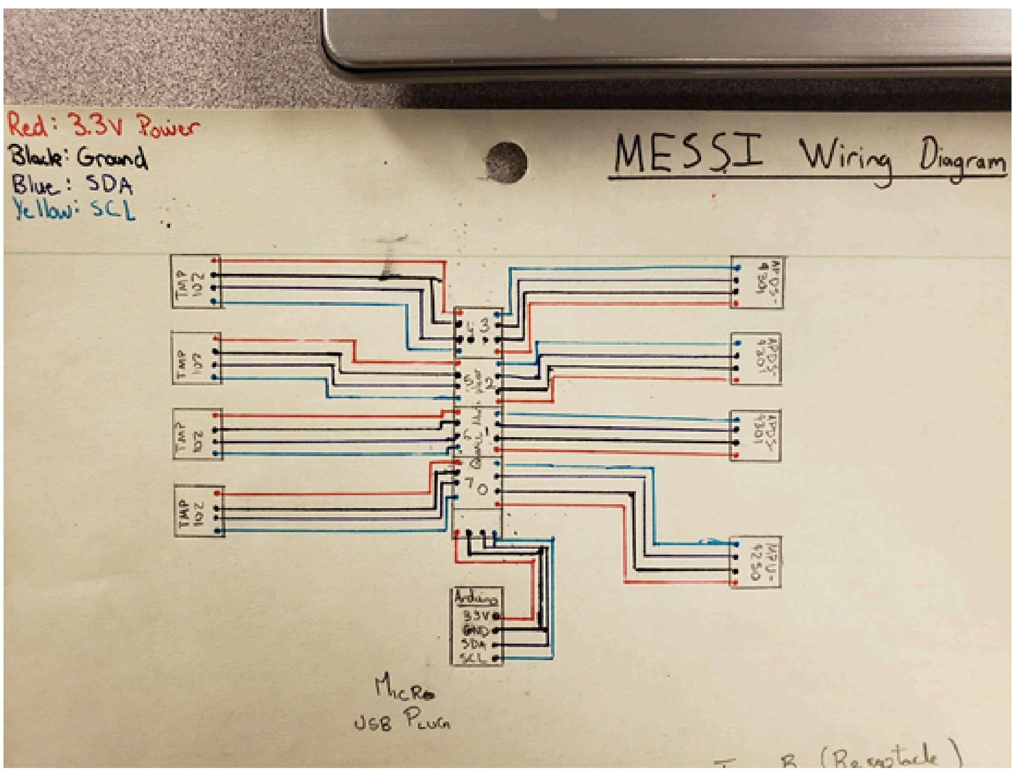


Figure S1 : Messi Avionics Wiring Diagram

Figure S2: Removing Factory Pins from Arduino MKR ZERO

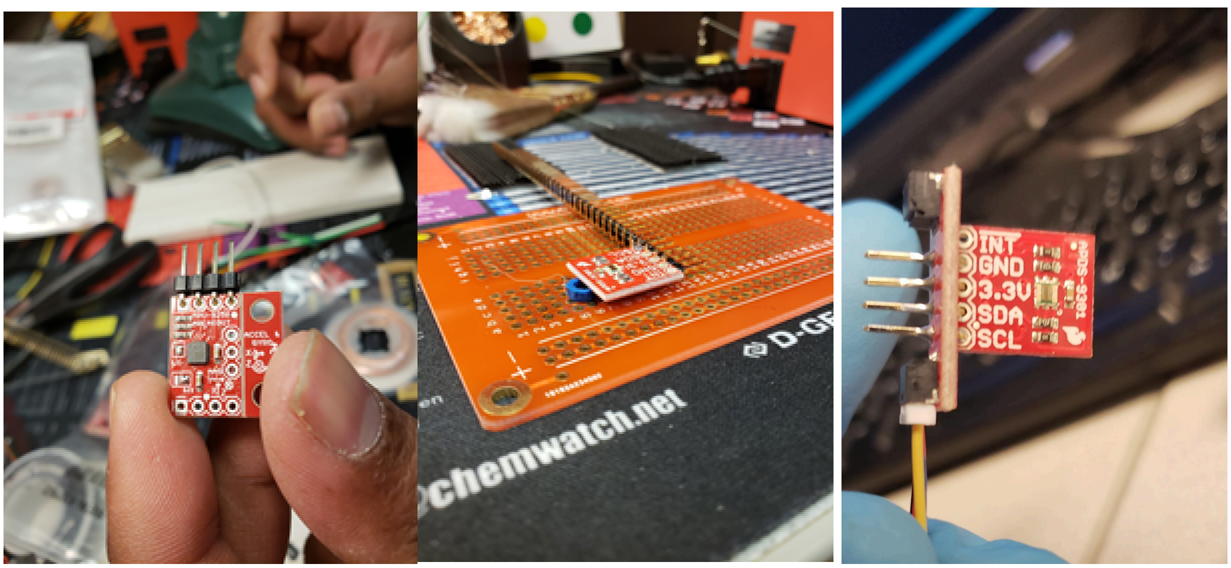


Figure S3: Adding Quiic Adapters

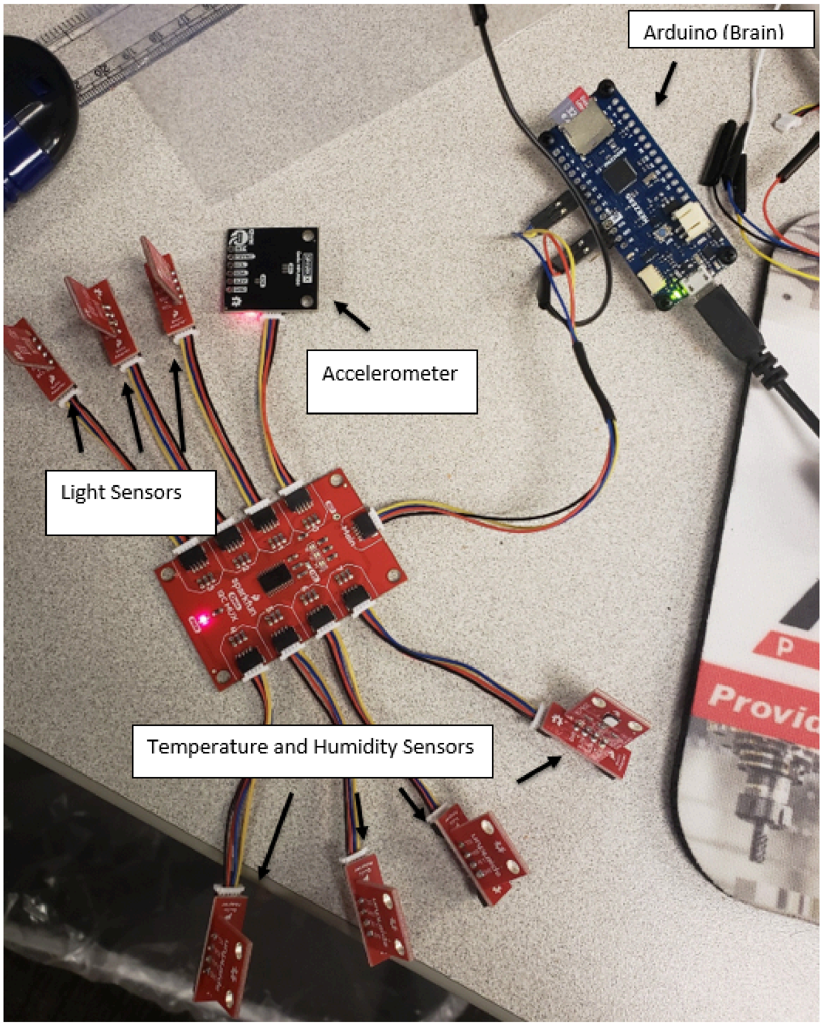


Figure S4: On table mockup with Quiic components installed

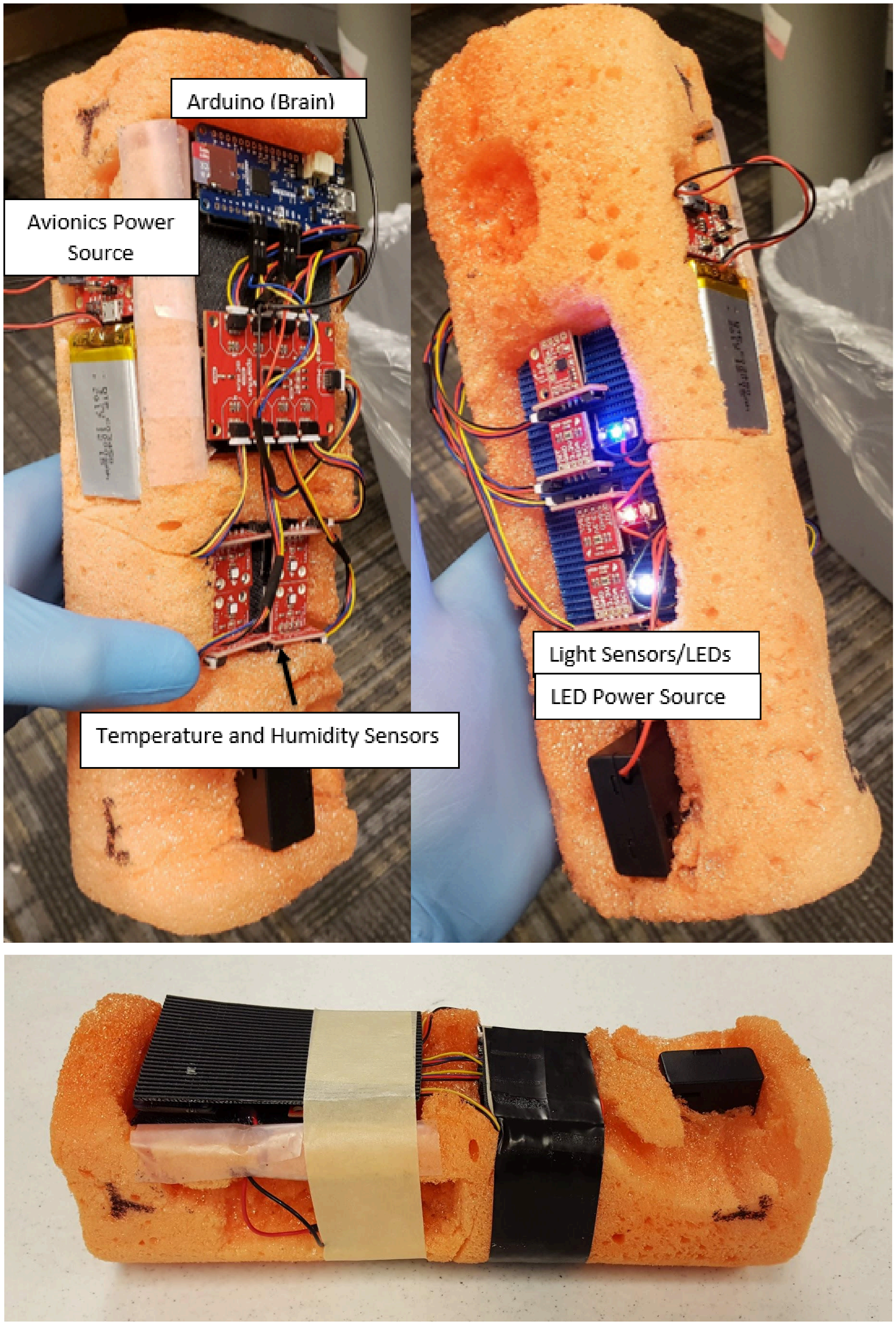


Figure S5: Avionics Test Launch Configuration

Software Design:

To begin software development, we were given a set of sensors to be used, and a base functionality program was written by Vijay that was simply the bare-bones of how the sensors must be called. Taking this starting point, we were to integrate it with the Nanoracks provided simulation program, to ensure that our program would be capable of logging flight data received by the Nanorack Feather Frame (NFF) module. To begin development, it was necessary to first solidify the base functionality of the program, and the sensor polling and data-logging functionality were made more compact, and put into functions so that they could more simply be called by the program later. The NFF code was built to be user friendly, and provided a region to paste your own code within the NFF code, which would then execute the user's code when it reached that phase. This, however, would not work for us, as it would cause our code to be executed less often than we would like, and additionally entirely skip operations of our code if data was not available from the NFF. The entire NFF program was converted into a function that we could call every 20ms to query the serial port for new data. The next issue to face was that we couldn't gather sensor data while simultaneously gathering serial port data. This was solved by implementing multithreading functionality, and making our sensors one thread, with the NFF program a second thread. This allows for concurrent operation without issue. Because operation of the program is dependent on a serial connection, some minor alterations needed to be made to allow us to test the program on a small test launch. Simply, references to Serial.read and Serial.write were removed, and datalogging script was made to ignore whether a serial connection is established.

<table>
<tr><td>

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <SparkFunMPU9250-DMP.h>
#include <Sparkfun_APDS9301_Library.h>
#include <SparkFun_Si7021_Breakout_Library.h>
#include <Thread.h>
#include <StaticThreadController.h>
```

</td><td>The code begins with declaration of header files to be used and variable declaration.</td></tr>
<tr><td>

```
Thread NFF_THREAD = Thread();
Thread MESSI_THREAD = Thread();
StaticThreadController<2> controller
(&NFF_THREAD, &MESSI_THREAD);
```

</td><td>This section of code defines controller objects to be used for triggering various parts of the code at set delays.</td></tr>
<tr><td>

```
APDS9301 apds;
MPU9250_DMP imu;
Weather weather;
```

</td><td>These are the initialization of sensor objects.</td></tr>
<tr><td>

```
typedef struct NRdata
{
  char flight_state;
```

</td><td>This is the data structure used by Nanoracks to receive and store flight information.</td></tr>
</table>

```
  double exptime;
  double altitude;
  double velocity[3];
  double acceleration[3];
  double attitude[3];
  double angular_velocity[3];
  bool liftoff_warn;
  bool rcs_warn;
  bool escape_warn;
  bool chute_warn;
  bool landing_warn;
  bool fault_warn;
} NRdata;

NRdata* flight_info;
```

```
void LogData(float data[]);

int parse_serial_packet(const char* buf, NRdata*
flight_data);
```

Here we declare a couple functions that need to operate cross-scope.

```
unsigned int count = 1;
unsigned int elapsedtime = millis();

char Fl_state;
float data[40];
```

A couple time-keeping variables, followed by the Flight State converter variable, and the Data array that we use to store all data before outputting to file.

```
void sensors(){
  float light[3] = {0};
  float humidity[4] = {0};
  float tmp[4] = {0};
  float mpu[3][3] = {0};
```

Here we define the Sensor function. This is where the main functionality of the payload is executed.
It begins with declaration of local variables to temporarily store sensor values.

```
   for (byte x = 0 ; x < 8 ; x++)
  {
   enableMuxPort(x);
   if(x==0)
   {
    imu.update(UPDATE_ACCEL | UPDATE_GYRO |
UPDATE_COMPASS);
```

A 'for' loop is initiated to iteratively sequence through each of the 8 mux ports, each with different instruction defined below.
For Mux port 0, we activate the IMU and initialize three of its functionalities.

```
    mpu[0][0] = imu.calcAccel(imu.ax);
    mpu[0][1] = imu.calcAccel(imu.ay);
    mpu[0][2] = imu.calcAccel(imu.az);
    mpu[1][0] = imu.calcGyro(imu.gx);
    mpu[1][1] = imu.calcGyro(imu.gy);
    mpu[1][2] = imu.calcGyro(imu.gz);
    mpu[2][0] = imu.calcMag(imu.mx);
```

Here we set the nine mpu array adresses we initialized earlier, setting them in order to the Accelerometer x, y, z, Gyrometer x, y, z, and then Magnetometer x, y, z.

mpu[2][1] = imu.calcMag(imu.my);
mpu[2][2] = imu.calcMag(imu.mz);
}
if(x==1)
{
apds.begin(0x39);
light[0] = apds.readCH0Level();
}
if(x==2)
{
apds.begin(0x39);
light[1] = apds.readCH0Level();
}
if(x==3)
{
apds.begin(0x39);
light[2] = apds.readCH0Level();
}
if(x==4)
{
tmp[0] = weather.readTemp();
humidity[0] = weather.getRH();
 	
}
if(x==5)
{
tmp[1] = weather.readTemp();
humidity[1] = weather.getRH();
}
if(x==6)
{
tmp[2] = weather.readTemp();
humidity[2] = weather.getRH();
}
if(x==7)
{
tmp[3] = weather.readTemp();
humidity[3] = weather.getRH();
}
disableMuxPort(x); //Tell mux to disconnect from this port
}
data[1] = mpu[0][0];
data[2] = mpu[0][1];
data[3] = mpu[0][2];
data[4] = mpu[1][0];
data[5] = mpu[1][1];
	On Mux ports 1 through 3, we are addressing Light sensors, with each we initialize the sensor and set it's read value to the local light variable.
	
	For Mux ports 4 through 8, we will be addressing the Temperature/Humidity sensors. With each, the sensor doesn't need refreshing so we just immediately grab their values and set them to local variables.
	
	After each iteration of the for loop, it closes the mux port to ready for next connection.
	
	After for loop, we set each of the local variables to our main Data array.

```
  data[6] = mpu[1][2];
  data[7] = mpu[2][0];
  data[8] = mpu[2][1];
  data[9] = mpu[2][2];
  data[10] = light[0];
  data[11] = light[1];
  data[12] = light[2];
  data[13] = tmp[0];
  data[14] = tmp[1];
  data[15] = tmp[2];
  data[16] = tmp[3];
  data[17] = humidity[0];
  data[18] = humidity[1];
  data[19] = humidity[2];
  data[20] = humidity[3];
  Serial.println(F("Sensor Funct ran."));
```

This is an output for debugging purposes.

```
  DebugOutput(data);
  LogData(data);
  return;
}
```

At end of Sensor Function, it runs the DebugOutput and LogData functions, then returns to main.

```
void LogData(float data[]){
  File Datalog = SD.open("datalog.txt", FILE_WRITE);
```

The Log Data function is used to, as the name implies, save data to a text file. It first declares a file pointer object to create a file for writing, or append data to a file that already exists.

```
  if(Datalog)
  {
```

This If is a conditional, if the file can be accessed.

```
   if(count==1)
   {
    Datalog.print(F("Sample Number,Flight State,Time (s),X Acceleration (G),Y Acceleration (G),Z Acceleration (G),X Gyration (deg/s),Y Gyration (deg/s),Z Gyration (deg/s),X Magnetic field (mu*T),Y Magnetic field (mu*T),Z Magnetic field (mu*T),A_Light Intensity (lux),B_Light Intensity (lux),C_Light Intensity (lux),A_Temperature (C),B_Temperature (C),C_Temperature (C),D_Temperature (C),A_Humidity (%),B_Humidity (%),C_Humidity (%),D_Humidity (%),Flight Altitude (ft),FV1,FV2,FV3,Facc1,Facc2,Facc3,Fatt1,Fatt2,Fatt3,Fw1,Fw2,Fw3,Floff,Frcs,Fesc,Fchu,Flan,Ffau"));
    Datalog.println();
   }
```

On the first iteration of the LogData function, it prints a File header to the text file.

```
   Datalog.print(count);
   Datalog.write(',');
   Datalog.print(Fl_state);
   for (int i = 0; i < 40; i++)
    {
     Datalog.write(',');
```

On every iteration, function saves Count, FL_State, and all information saved to Data array to file.

```
      Datalog.print(data[i]);
    }
    Datalog.println();
    Datalog.close();
  }
```

Then closes file.

```
else
  {
    Serial.println(F("error opening datalog.txt"));
  }
```

Else, the Datalog file couldn't be created or accessed.

Prints error to Serial Monitor.

```
  Serial.println(F("Data logger ran."));
  count++;
  return;
}
```

Increment Count variable, and return to main.

The DebugOutput function, simply prints all information saved to data array to Serial monitor, for debugging purposes.

```
void DebugOutput(float data[]){

  int i;
  for(i = 0;i < 40; i++){
    Serial.print(data[i]);
    Serial.print(F("\t"));
  }
  Serial.println();
  return;
}
```

The setup function runs the first time arduino is ran after a reboot.

It initilizes the Arduinos connections and sets the serial port Baud rate.
If the Serial Connection is unavailable, it waits eternally for serial connection. This has to be removed for battery power operation.

Enable mux port 0 and initialize IMU

Initialize IMU mode

Disable Mux port.

Lots of Serial Prints for debugging here.

```
void setup() {
delay(10);
  Wire.begin();
  Serial.begin(115200);
  while (!Serial) {  }
  Serial.println(F("Initializing IMU..."));
  enableMuxPort(0);
  Serial.println(F("enabled mux port"));
  imu.begin();
  Serial.println(F("IMU began"));
  imu.setSensors(INV_XYZ_GYRO |
INV_XYZ_ACCEL | INV_XYZ_COMPASS);
  Serial.println(F("IMU sensors initialized"));
  disableMuxPort(0);

  Serial.println(F("IMU initialized."));
```

If SD card cannot be initialized, program enters a closed loop, no point to continue if sensor data cannot be saved.

```
  Serial.println(F("Initializing SD card..."));
   if (!SD.begin()) {
    Serial.println(F("Card failed, or not present"));
    while (1);
    }
```

Otherwise success is sent to serial monitor.

```
  Serial.println(F("SD card initialized."));

  weather.begin();
  Serial.setTimeout(20);
```

Initialize Temp/Humidity sensors
Set serial timeout for NFF functionality.

```
  NFF_THREAD.onRun(NFFFunct);
  MESSI_THREAD.onRun(sensors);
  controller[0].setInterval(20);
  controller[1].setInterval(100);
  Serial.println(F("Threads initialized."));
}
```

Here we declare that the two threads initialized earlier run the NFF and sensors functions, respectively. Then we tell the controller the intervals they should run; 20ms and 100ms.

```
void loop() {
  Serial.println(F("Initializing Controller."));
  while(1){
```

This is the actual portion of the code that runs on loop. There is an output for debugging, then enters infinite loop.

```
   controller.run();
   if(NFF_THREAD.shouldRun()){
    LogData(data);
    Serial.println(F("NFF Thread Triggered."));
   }
   if(MESSI_THREAD.shouldRun()){
    Serial.println(F("MESSI Thread Triggered."));
   }
  }
}
```

It initiates the controller, then waits until thread gets initialised, each triggering one of the If statements here, the If statements simply run the LogData function and output to serial monitor for debugging.

And this is the end of the program.

Example of Data Received by NFF Simulator:

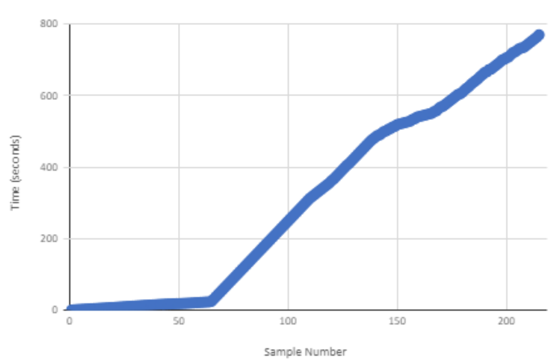


Figure S6: This graph represents the time of samples sent from NFF simulator in seconds on the y axis, compared to sample number on the x axis. This shows that receiving data from NFF happens on a non-linear timeline. This is why multithreading was a necessity.

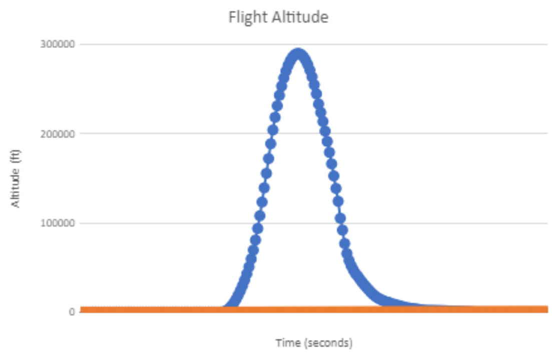


Figure S7: Expected flight altitude over time, in feet.

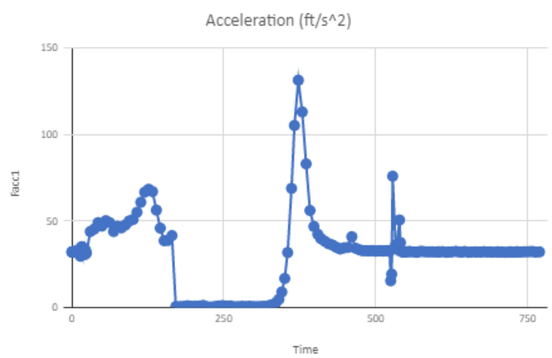


Figure S8: Acceleration over time, in ft/s^2

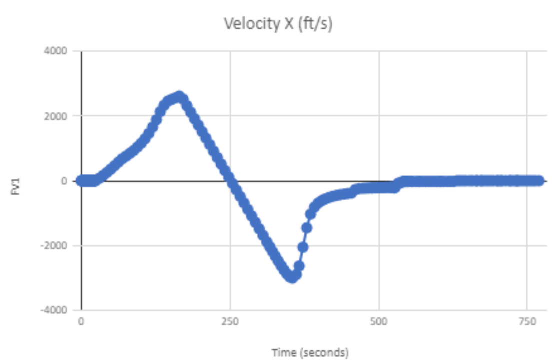


Figure S9: Vertical Velocity over time, in ft/s

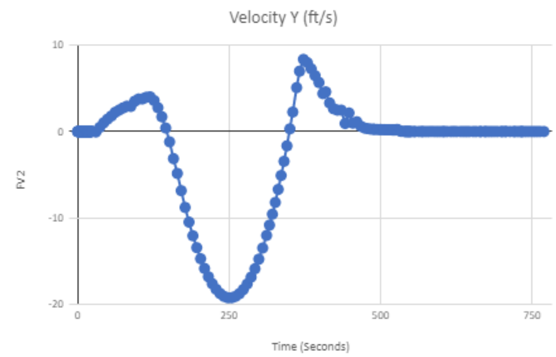


Figure S10: Horizontal Velocity in East/West direction from Launch site, in ft/s.

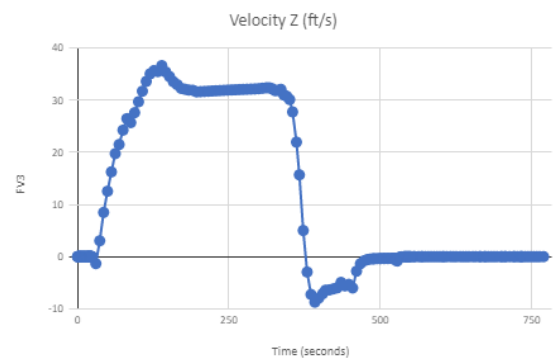


Figure S11: Horizontal Velocity in the North/South direction from Launch site, in ft/s.

Bio/Spirulina Team with Documentation

Progress and Procedures

- **Pasteurization of Seawater**
  To be completed before arrival of Spirulina (Completed 9/18/18)
  - Start by first sterilizing a suitable sized vessel. A large pot or glass bowl would be of a good size. You will need at least two liters
  - Sterilize the container with a solution of 50% bleach and 50% water to prevent any previous growth or contamination.
  - Vessel needs to be rinsed multiple times to prevent any bleach residue which could hinder growth.
  - Once vessel has dried in the sun, store in a clean container for transport to beach. Do not dry with towel which can leave dust residue
  - Transport the vessel to the beach and collect 2 liters of sea water directly from the ocean.
    - Our seawater was collected from Daytona Beach
  - This vessel needs to be safely transported in a clean box to a suitable heat source.
  - Sea water will then need to start pasteurization process by boiling at over 150 degree fahrenheit for a constant fifteen minutes .
  - Once time is completed, this container needs to sit for twenty-four hours to let the sea water cool and settle.
  - The next day, the process needs to be repeated by boiling again at over 150 degrees for fifteen minutes.
  - Let the container cool overnight and the sea water is now ready for usage.

- **Establishing Master Colony**
  Completed by five days after Spirulina are affected by sunlight (Completed 10/3/18)
  - In preparation of new colony, a sterilized container must be obtained.
  - Clean and prepare this container by using a mixture of 50% bleach to water ratio and make sure container is completely rinsed out with water to avoid contamination or killing of the algae (Figure 12)
  - The container should be left to dry in the sun to avoid dust or tiny tissue particles from a paper or cloth towel.
  - When the samples of spirulina arrives, pour entire container of spirulina and traveling liquids into the container. This should be about 10 ml. (Figure 11)
  - Add 200 ml of growth medium for every 10 ml of spirulina and traveling liquid. (Figure 13)

- o This mixture will then need to be allowed to sit and grow with little interruption for a period of two to three weeks. (Figure 1) (Figure 2)
- o The colony needs to be kept in a room around 32 degree celsius and under a white-light bulb.
- o The bulb needs to provide 100 foot-candle of light and kept eighteen to twenty-four inches away.
- o Daily agitation is important to allow oxygen into the medium. A slow continuous agitation would provide better results.

- **<u>Procedure to set up new tank</u>**
  <u>Complete after 4 weeks of initial growth (Completed 11/1/18)</u>
  a. Equipment needed: 500ml graduated cylinder, 30ml beaker, 70% isopropyl alcohol, measuring cup, distilled water, and latex gloves
  b. Clean out new growth tank with distilled water with 2 ml alcohol
  c. Rinse with distilled water three times
     i. Make sure no alcohol smell remains, may affect growth
  d. Rinse out graduated cylinder and beaker with distilled water twice
  e. Carefully transfer spirulina growth into beaker (Figure 6)
     i. More aseptic technique is to scoop spirulina with beaker so some medium transfers for moisture
  f. Measure out 200 ml of medium from original growth chamber and add to beaker
  g. Now agitate beaker to break up spirulina mass (Figure 7)
     i. 80 ml of medium still remained in the bowl
  h. Activate bubbler as you start to add in new water (Figure 5)
  i. Slowly add 3 liters of pasteurized seawater into new growth tank
     i. Seawater was obtained from reserve recently pasteurized and stored in lab
     ii. For each liter, add 20 ml of seawater concentrate medium
     iii. This is store bought and comes in a tube
  j. Move powerstrip and lamp while adding seawater
  k. In the end we had 3.06 liters total medium in growth tank
  l. Plug in lamp and reset to proper location for light-candles
     i. For us will be about 10 inches
  m. Filter spirulina to remove some of the old medium
     i. Don't let dry out, some medium can be transferred
     ii. small amount returned to bowl for contingency growth
        1. contingency growth will be added to main tank setup after new culture has survived for 14 days without incident. (Figure 10)
  n. Add spirulina to new growth tank (Figure 8)
  o. Over all now 3.31 liters of liquid now in tank with spirulina
  p. Leave bubbler to run twelve hours on, twelve hours off
  q. Put on the lid to prevent evaporation (Figure 9)

- **<u>Creation of Subculture Colony</u>**
  <u>Can be completed after ten weeks of growth (To be completed)</u>
  - o Three tanks will be set up just like the orginal growth tank
    - ▪ Clean out new growth tank with distilled water with 2 ml alcohol
    - ▪ Rinse with distilled water three times
      - Make sure no alcohol smell remains, may affect growth
    - ▪ Rinse out graduated cylinder and beaker with distilled water twice
    - ▪ Carefully transfer spirulina growth into beaker
      - More aseptic technique is to scoop spirulina with beaker so some medium transfers for moisture
    - ▪ Measure out 200 ml of medium from original growth chamber and add to beaker
    - ▪ Now agitate beaker to break up spirulina mass
      - 80 ml of medium still remained in the bowl
    - ▪ Activate bubbler as you start to add in new water
    - ▪ Slowly add 3 liters of pasteurized seawater into new growth tank
      - For each liter, add 20 ml of seawater concentrate medium
      - This is store bought and comes in a tube
  - o The original control colony growth will have a predetermined amount of spirulina transferred aseptically from the control tank into the new tanks
  - o Each new tank will be lined up along a table with a solid partition separating each tank to prevent light bleeding over
  - o Each tank will then have a row of colored LEDs attached to the lid
    - ▪ These LEDs will produce less light, due to size, then a standard LED light bulb, so will be placed closer to the growth medium
    - ▪ 5 Red, 3 White, 11 Blue
  - o Each tank will now have continuous growth until for the next few months

<u>Preparation of Experiment</u>

1. Confirm that the payloads are fully prepared with structure and electronics configured in their final form.
2. Confirm that the mothership, red, green, and blue cultures of spirulina are grown and healthy.
3. Clean and sanitize in boiling water enough test tubes for 4 payloads worth of spirulina.
4. Prepare led lights and test tube holding system to confirm they are ready to begin the experiment.
5. Measure and confirm the uniform mass of the empty and dried test tubes.
6. Fill each test tube with (1-3 mL dependent on tube size) and secure and seal each tube as per flight operations procedure for sealing.
7. Measure the “wet mass” of the test tube, water, and spirulina.
8. Record each individual mass of each labeled and marked tube.

Methodology- Flight Payload and Ground Control Payload

1. Take individual (75% test tube size) samples from mother colony directly to fill each test tube.
2. As the samples are taken, insert them directly into the holding mechanism within the payload.
3. As soon as the payloads are loaded, close and secure them with their respective colored lights on.
4. Assure twice a day, every day from time of sampling until hand off to launch officials, that the spirulina is still being lit by the lights, and that no containment has been breached. Complete this assurance under dark lighting condition so as to not introduce light pollution.
5. Follow flight operation procedures for payload preparation for handoff to flight officials.

Modifications

- The experimental parameters have been changed for the study during the launch
  - The Initial plans had three sets of LED lights (Red, Green, and Blue) to be emitting against the spirulina while the rocket achieved suborbital flight.
  - Further research into growth capabilities has found that the green chlorophyll color of the algae will lead to the green light emitting bulb to have no effect since the light will be reflected from the cells.
  - The procedure has now changed so that the green light bulb shall be replaced by a White light frequency emitting bulb that simulates a continuing growth procedure as we have now.
  - Thus, the final product will compare colored emitting of red, blue, and white lights
- When the initial procedure was drawn up, the team wanted the subculture growth tanks to use color emitting light bulbs
  - After researching colored light bulbs, it was found that many light bulbs were just painted glass and did not emit the proper light frequency
  - Discussion with electronics team found that the LEDs to be used during the experiment could be rigged up for the culture tanks
  - A second set of LEDs was purchased for the tank
  - Since these are the same LEDs that will be used in the experimental flight, this will actually keep variance down between the tank and payload.
  - The lids, which are already purchased, will be fitted with proper holding foam to house the LEDS and kept at the proper 100 light-candles position

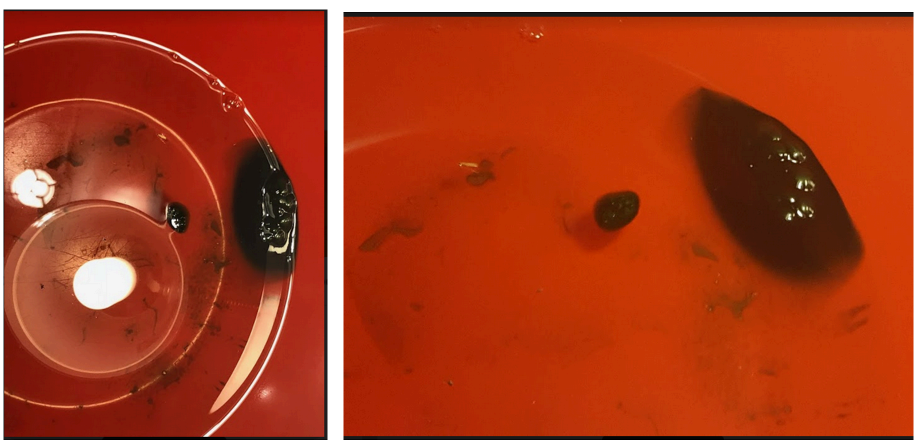
- Initial discussion with project leader decided separate tank has been set up for controlled growth and contingency growth
    - When the new growth tank arrived and the spirulina was being transferred, a decision was made to leave some in the old growth container as a contingency plan if the new growth tanks killed the spirulina from too much aeration or light
    - Contingency growth medium container has a few grams of Spirulina that is now growing in a mass like shape in the medium (Figure 3)(Figure 4).
    - There is a seperate White emitting frequency light bulb set up creating a completely different growth area
    - The two tanks are separated by a partition to prevent any light bleeding which could affect the growth
- After weeks of continual health and growth of the base colony, the contingency growth was added back into the base growth.
    - There was concern that contingency growth chamber had continued evaporation due to the lack of a proper sized lid.
    - Base colony will become contingency colony after subcultures are created in the coming weeks.
    - Growth in bowl was transferred into cleaned test tube and 2 ml seawater was added
    - Tube was agitated to break up mass of Spirulina for 5 minutes and then added slowly to growth tank

Photos

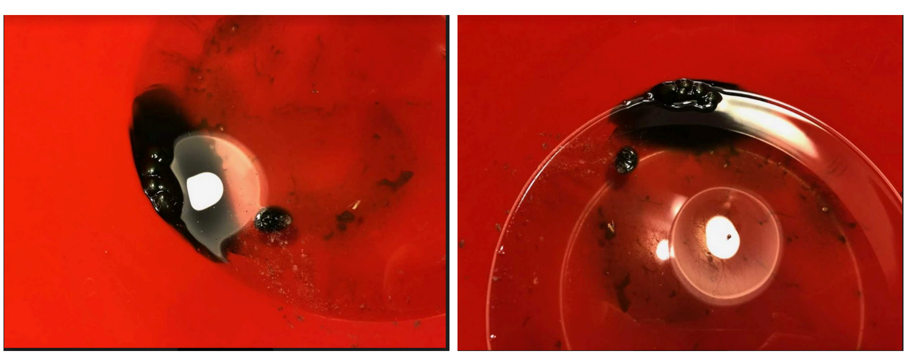
Growth Photos

Oct 10th, 1inch x 1 inch

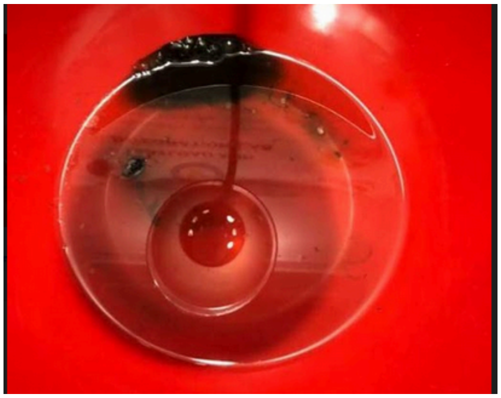
Oct 15th, 3 inches x 1 inch



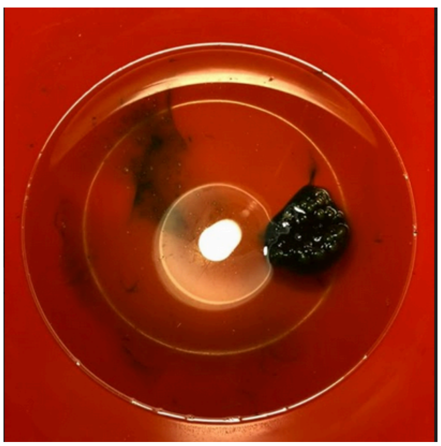
Oct 17th, Large growth -2 by ¾ inches eclipse sized. Small mass-½ by ½ circular sized



Oct 19th, Center 3 inches by 1.5 inches in size, Second growth is still about the same



Oct 23rd, 2 inches long and 1.5 inches across



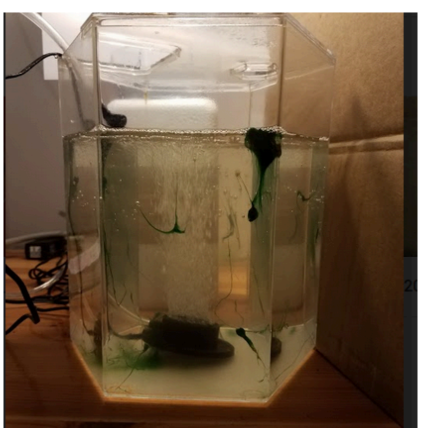
Oct 25th, 2 inches in length and 1.5 inches wide

Oct 29th, 2.5 inches dia. Tendrils 1 inch long

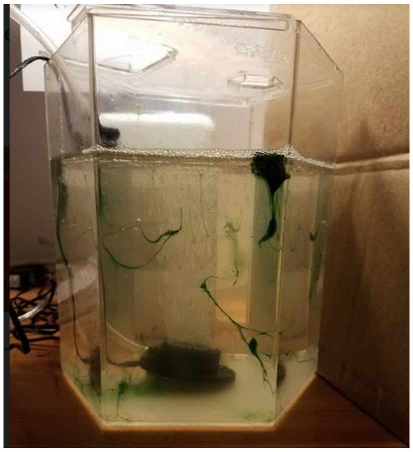
Oct 30th, 2.5 inches diameter

Nov 2nd, Left- base colony tank, Right-contingency

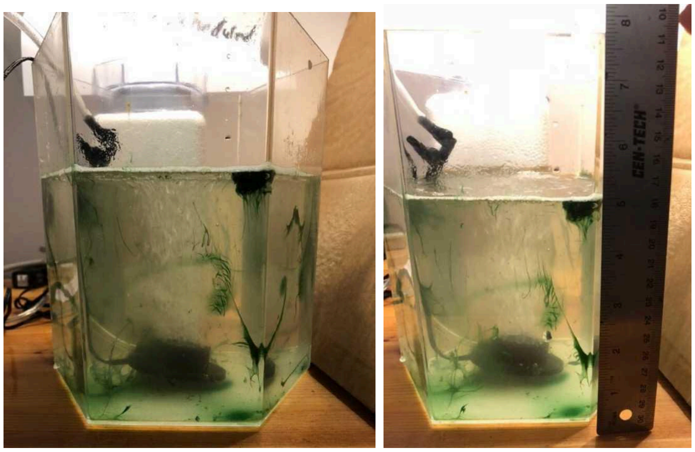
Nov 6th, Left- base colony tank, Right-contingency (1.5 inches by 3 inches with tendril)




Nov 8th, Left- base colony tank, Right-contingency

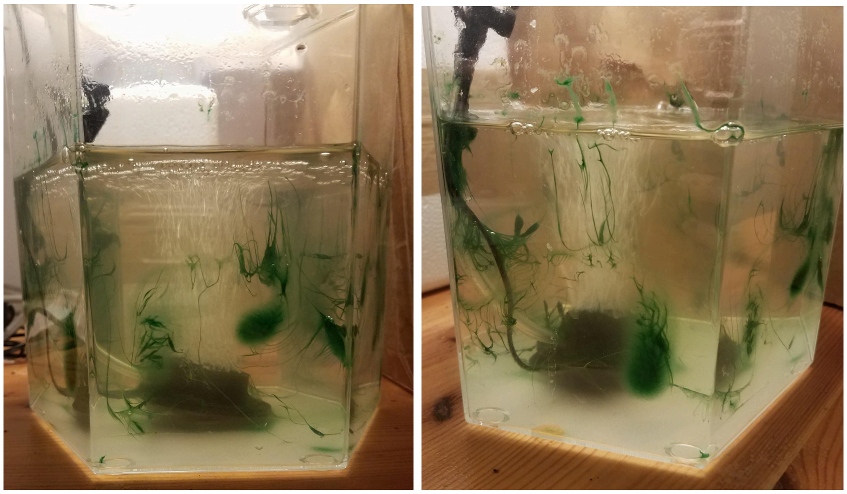

Nov 10, Left- base colony tank, Right-contingency


Nov 13, Left- base colony tank, Right-contingency (growth 2 inches by 1.5 inches)



November 16th - Master colony

November 20th - Master Colony

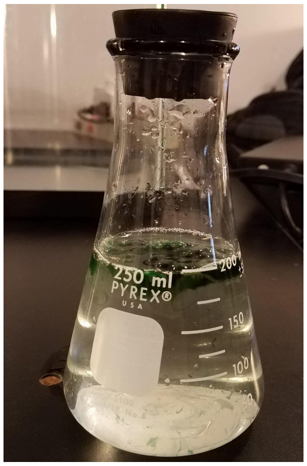


November 27th - Master Colony

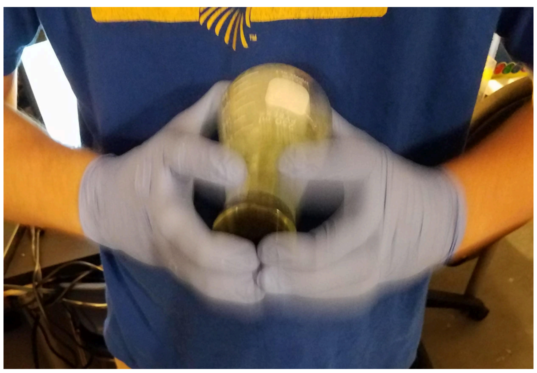
<u>Tank Transfer Photos - November 1st, 2018</u>

Culture before tank transfer process began

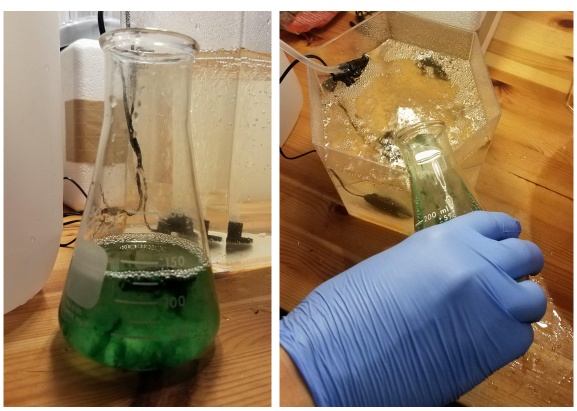
New, clean tank and light set-up before transfer



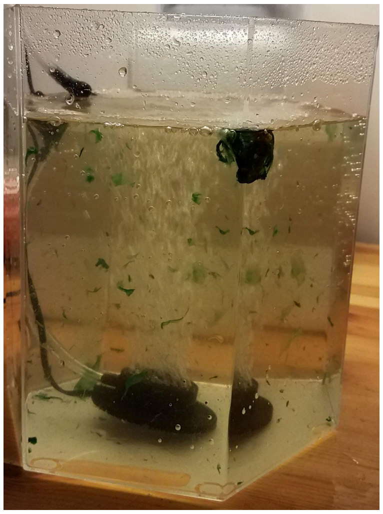
Spirulina culture in plugged 250 ml beaker



Agitation of culture in beaker before transfer to new tank



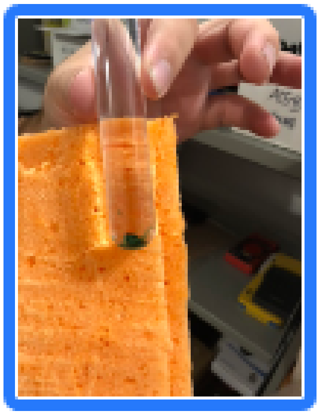
Agitated spirulina, then poured into new tank set-up



New mother culture tank after spirulina transfer

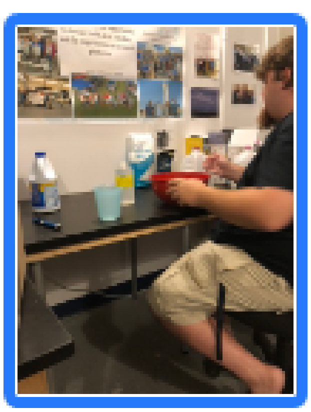
Contingency culture after transfer

Establishing Master Colony Photos



Spirulina Colony in tube

Sanitizing Growth Container



Measuring medium into container

Newly established colony

Flight Ops Team with Documentation

Flight Test Checklists, Procedures, Profiles

RockSim Procedures:

- **Begin by taking measurements of the rocket dimensions**
- **Fill out rocket data sheet with corresponding measurements and type of rocket; file for ezi-65 rocket could be found on apogee rockets(https://www.apogeerockets.com/Rocket_Kits/Skill_Level_3_Kits/EZI-65?zenid=kjop0uqs2h6hkv9q8pfu5t9le7#rocksim)**
- **Continue by selecting the engine, Cesaroni i175, through (http://www.thrustcurve.org/simfilesearch.jsp?id=1883); to include file into rocksim database, select engine reload data under the file tab. Then under launch tab, selection of motor can be made in which the engine file should be found.**
- **Include length, diameter, number of fins, empty mass, drag coefficient, nose shape, center of gravity, number of stages, number of motors, fin shape and area**
- **Identify type of recovery system to be used (primary and secondary)**
- **Utilize RockSim program and enter rocket design components: input nose cone, payload tube (tube coupler, bulkhead, eye-bolt washer, 2 nuts), body tube measurements (motor mount, ring, ring, launch lug, launch lug, parachute, custom fin set, epoxy/paint/misc.), payload mass (75g)**
- **Get theoretical results from launch profile, gather data from sensors during test flight, compare to RockSim data. Input of data and tools are vital for fidelity of simulation and flight profile.**
- **Check/input weather conditions, wind conditions (calm-damaging) for launch date at Bunnel, FL; input launch angle in order to compensate for crosswinds**

Checklist for Required Items (Before Leaving Daytona):

- ❏ Epoxy (for any and all repairs)
- ❏ Tool box
- ❏ Soldering station
- ❏ Extra batteries
- ❏ Power supply
- ❏ Drilling machine
- ❏ Dremel kit
- ❏ Spare sensors (if available)
- ❏ Foldable table
- ❏ Laptops (for documentation/analysis of data)
- ❏ Lubricants
- ❏ Nuts/Bolts
- ❏ Acetone
- ❏ Rain cover (in case)
- ❏ Payload
- ❏ Rocket
    - ❏ Rocket Stand

Pre-Flight Launch Facility Checklist:

- ❏ Inspect Launch Facility
    - ❏ Check for Foreign Object Debris around launch area
- ❏ Inspect Launch Pad
    - ❏ Check Base of Launch Pad
    - ❏ Check Launch Rod
- ❏ Obtain a weather report/briefing
    - ❏ Confirm that conditions meet specified requirements

Pre-Flight Checklist:

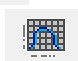
- ❏ Ensure that all sensors are connected properly to Arduino
  - ❏ Sensors include:
    - ❏ Temperature (1)
    - ❏ Light (1)
    - ❏ Accelerometer (1)
- ❏ Ensure battery is charged
- ❏ Connect battery power
- ❏ Turn on Arduino
- ❏ Inspect Arduino to ensure it is working properly
- ❏ Inspect payload to ensure structural integrity
- ❏ Insert all electronics and sensors into payload
- ❏ Secure payload components within payload
- ❏ Inspect payload bay for any cracks
- ❏ Orient payload correctly when placing into rocket payload bay
- ❏ Secure payload within payload bay
- ❏ Inspect the rocket for deformities, etc.
- ❏ Ensure parachute is attached correctly and is void of rips/tears
- ❏ Fold the parachute inside the rocket
- ❏ Perform a final check to ensure all components are secured

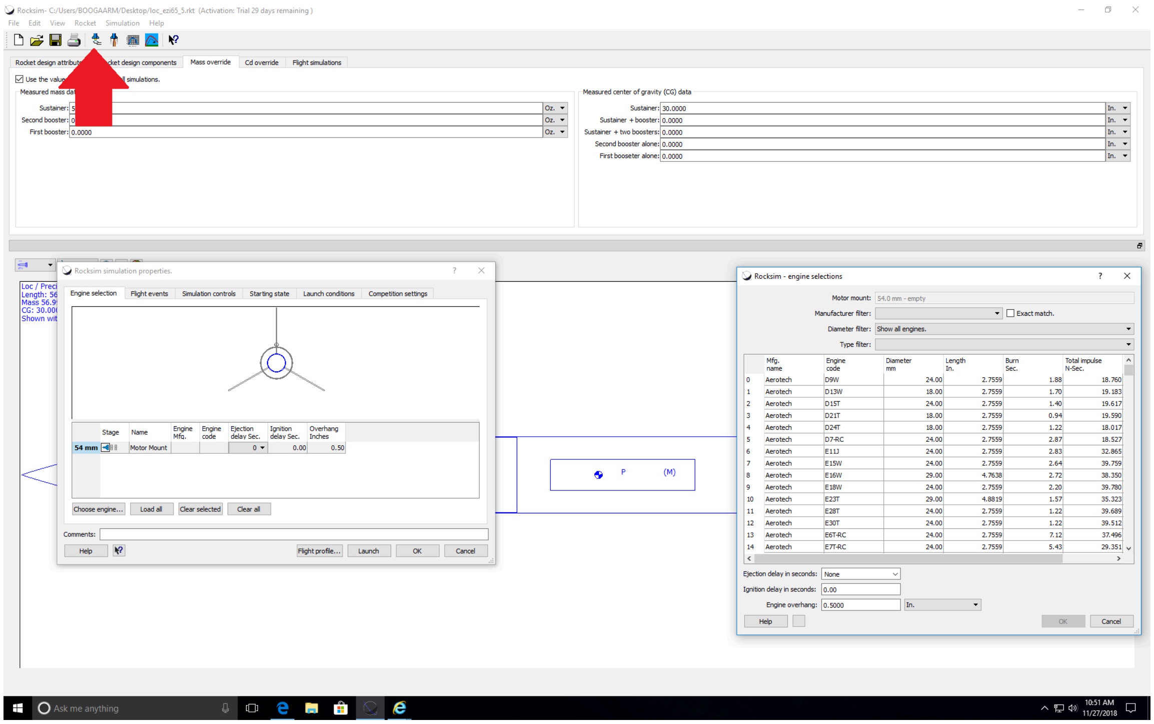
Post-Flight Checklist:

- ❏ Collect rocket
- ❏ Inspect for significant damage
  - ❏ Parachute significantly damaged
  - ❏ Structure significantly damaged
- ❏ Transport rocket and payload to computer for data download
- ❏ Collect data from Arduino; analyze

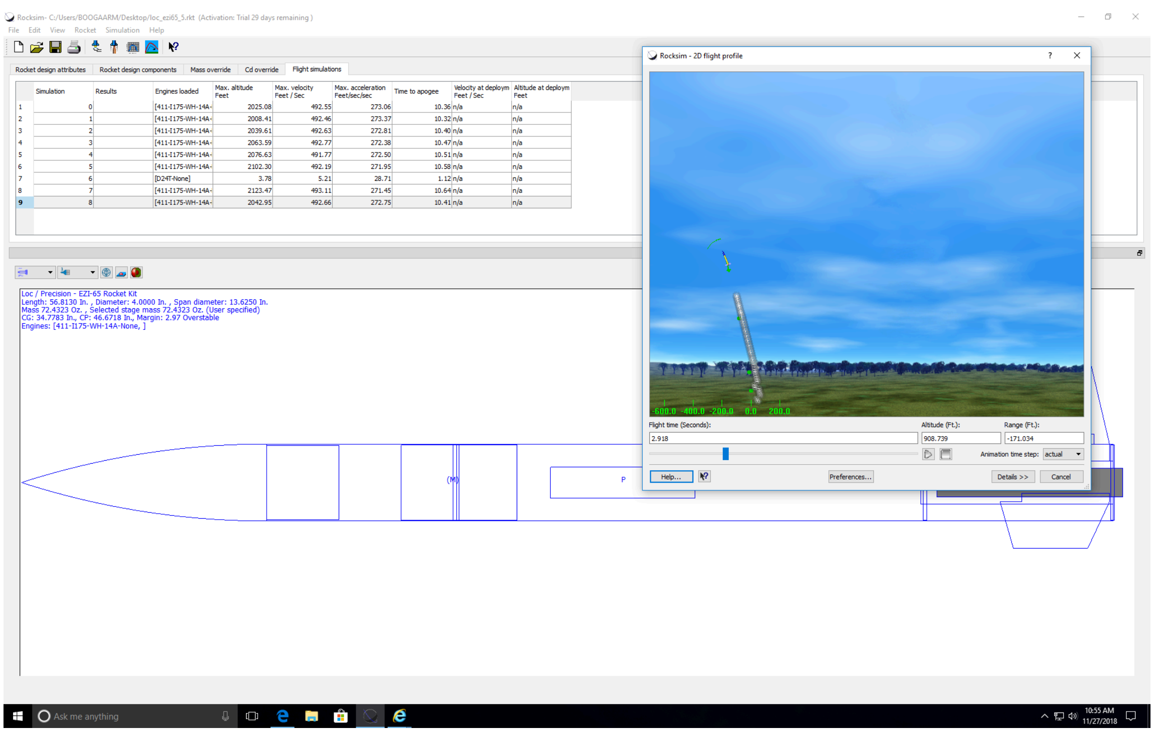
Flight Profiles:

Expected flight profiles for the level 1 test launch are needed to predict the flight path and landing spot of the rocket. After the flight, these flight profiles can also be compared to the retrieved sensor data. This is a way to verify the sensors on the test flight work correctly. Flight profiles for the test flight were generated by a computer simulation using Rocksim software. The following steps were taken to create the plots in Rocksim:

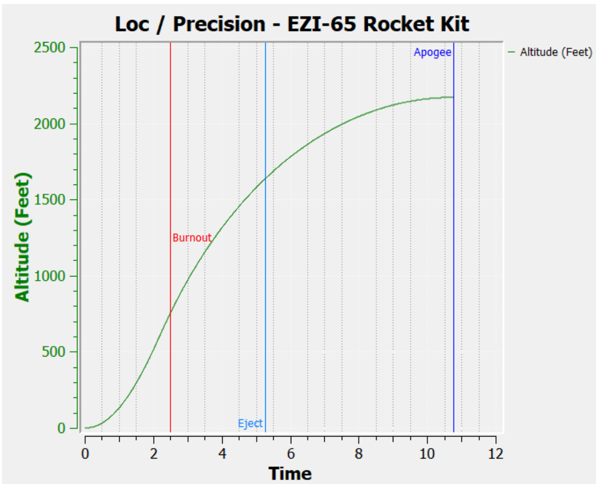
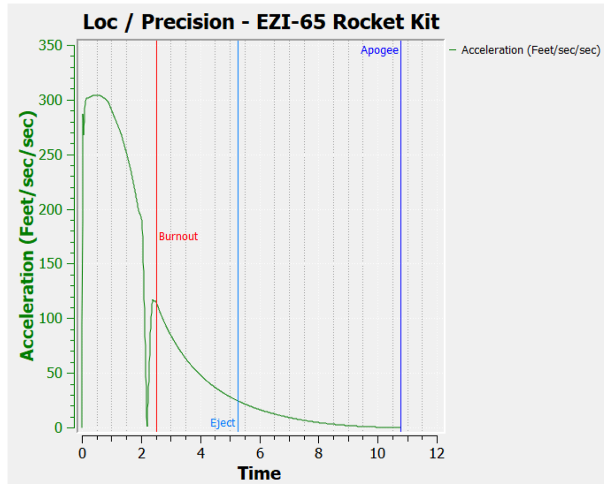
- Download EZI-65 rocket (.rkt) file from apogeerockets.com
- Download the Cesaroni I175 engine (.rse) file from http://www.thrustcurve.org/motorsearch.jsp?id=928
- Open Rocksim software
- File-> Open -> browse to EZI-65 .rkt file.
- FIle-> Reload engine data -> Import file -> browse to I175 .rse file -> Add to right column.
- Prepare for launch  -> Engine selection -> Choose engine -> Select I175 (-> check burn time = 2.5s/ specific impulse = 411.414 N*s)
- Prepare for launch -> Launch conditions -> Wind conditions -> Slightly breezy.
- Rocket design components -> Payload tube -> Mass override -> Fill in payload mass (~115 grams) in mass field & fill in measured CG location.
- Click Launch 
- Click Plot graph  -> Select needed metrics to graph.

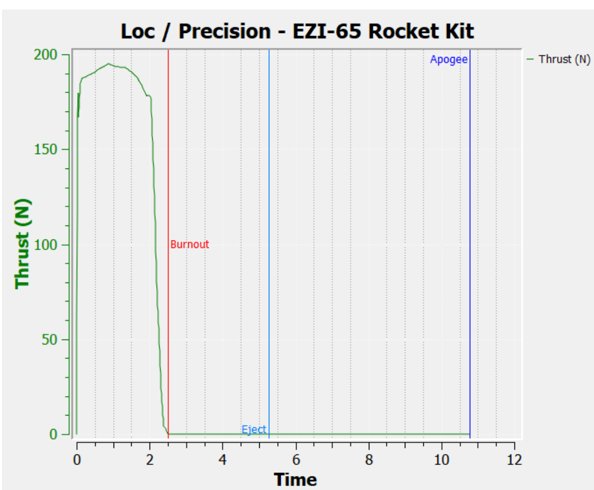
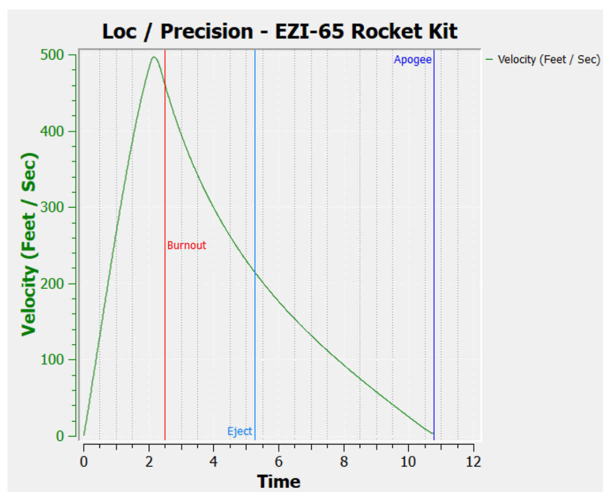


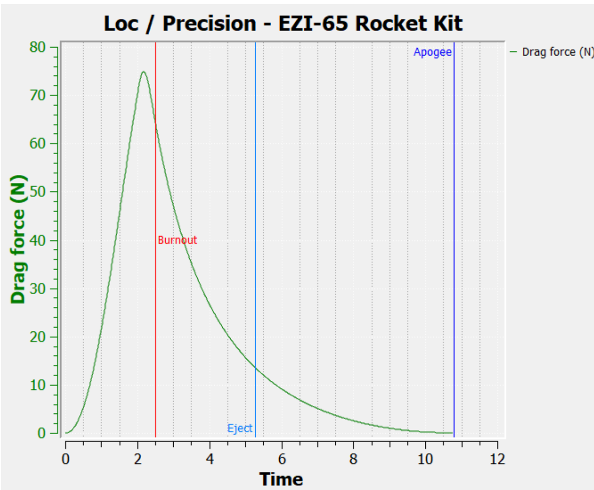
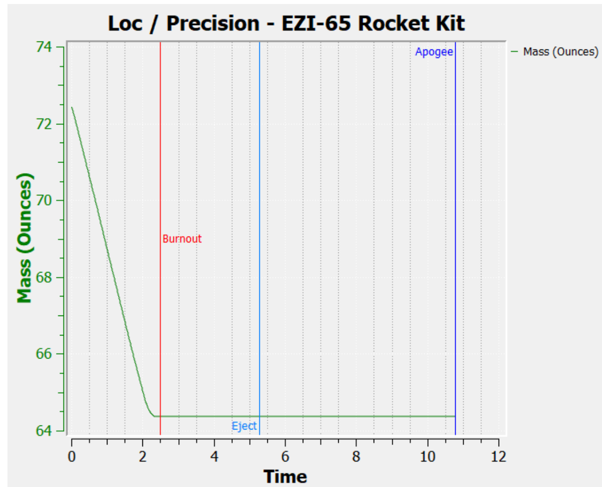
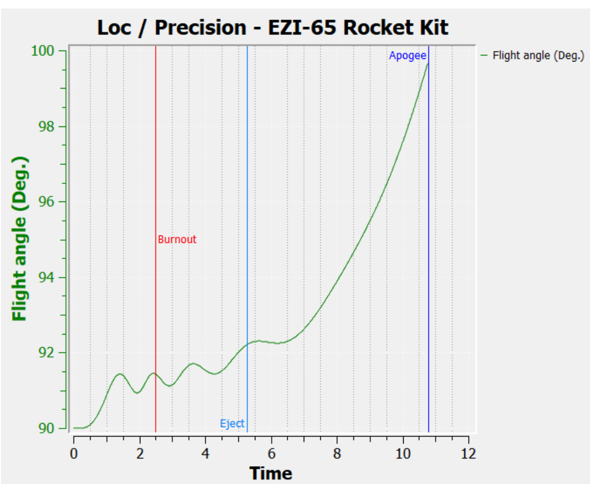
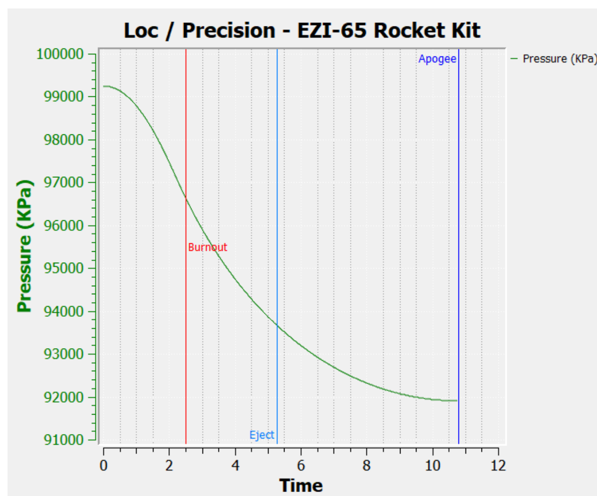
Prepare for launch, engine selection



Launch animation

Altitude	Acceleration
	

Thrust	Velocity
	

<table>
<tr><th>Drag Force</th><th>Mass</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th>Flight Angle</th><th>Pressure</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

(Rocksim)

Launch angle sensitivity studies:

The simulations were run to estimate the most important characteristics of the launch and how to optimize these results. The expected wind the simulations were performed at was slightly breazy (8-14 miles per hour). In order to adapt to these conditions the appropriate launch angle needs to be selected. See the table below for results of maximum range and altitude for different launch angles. Using this sensitivity studies the launch angle with the most favorable flight conditions can be selected. Note that there is a degree of randomness in the Rocksim results because the wind model ranges from 8 to 14 MPH. This is why some results may be considered outliers.

Launch angle (degrees)	Maximum altitude (feet)	Maximum range (feet)
-25	2016	702
-20	1894	954
-15	1895	953
-10	2017	701
-5	1957	841
0	2080	413
5	2065	386
10	**2115**	**104**
15	**2102**	**208**
20	2066	384
25	2118	63

The optimum launch will have a high maximum altitude with a low maximum range. The optimum launch angle for this wind condition based on this data seems to be 10-15 degrees. The maximum range reaches a minimum and the maximum range reaches a maximum around this angle.

References

Delrue, F., Alaux, E., Moudjaoui, L., Gaignard, C., Fleury, G., Perilhou, A., . . . Sassi, J. (2017). Optimization of Arthrospira platensis (Spirulina) Growth: From Laboratory Scale to Pilot Scale. Fermentation, 3(4), 59. doi:10.3390/fermentation3040059

How to Install the Thermal-Protector. (n.d). Retrieved October 29, 2018, from https://www.raise3d.com/pages/thermal-protectorr-intallation-instructions

James, Daniel E. Culturing Algae. 2nd Ed. Carolina Supply Book Company. 1978

Kumari, A., Pathak, A. K., & Guria, C. (2015). Effect of light emitting diodes on the cultivation of Spirulina platensis using NPK-10:26:26 complex fertilizer. Phycological Research, 63(4), 274-283. doi:10.1111/pre.12099

NanoRacks Feather Frame Simulator (Version 1.0) [Computer Software], Nanoracks LLC., Z. Porter, 5/15/17

Poulet, L., Fontaine, J., & Dussap, C. (2016). Plant's response to space environment: A comprehensive review including mechanistic modelling for future space gardeners. Botany Letters, 163(3), 337-347. doi:10.1080/23818107.2016.1194228

Rocksim (Version V9.0) [Computer software]. (n.d.). Retrieved October 29, 2018, from https://www.apogeerockets.com/RockSim/Rocksim_Trial

Wang, C., Fu, C., & Liu, Y. (2007). Effects of using light-emitting diodes on the cultivation of Spirulina platensis. Biochemical Engineering Journal, 37(1), 21-25. doi:10.1016/j.bej.2007.03.004

Yim, S., Ki, D., Doo, H., Kim, H., & Kwon, T. (2016). Internally illuminated photobioreactor using a novel type of light-emitting diode (LED) bar for cultivation of Arthrospira platensis. Biotechnology and Bioprocess Engineering,21(6), 767-776. doi:10.1007/s12257-016-0428-6