University of Vermont
RockSat-C
Integrated Subsystem Testing Review

4/3/2020
Presentation Outline

• Section 1: Mission Overview
• Section 2: Subsystem Status
• Section 3: Integrated Subsystem Testing Status
• Section 4: Full System Integration Plan
• Section 5: Project Management Update
1.0 Mission Overview

Name of Presenter
The goal of the UVM RockSat C payload is to study the effects of microgravity on off axis laminar mixing in micropropulsion systems, and additionally track the spin rate of the rocket during flight using an IMU.
Mission Overview- Mission Objectives

• Fluid stream mixing experiment requirements/objectives
  – Pressurization of two fluid lines (ethanol and water)
  – Solenoid valves to precisely inject fluid streams into mixing chamber
  – Clear mixing chamber (simulated combustion chamber) to observe the mixing of the fluid streams
  – LED lighting to illuminate experiment
  – Photo burst of mixing chamber for analysis
  – Video recording of experiment for affirmation of proper mixing
  – Secondary containment of all fluids
Mission Overview - Mission Objectives

• Data analysis of fluid stream mixing images captured
  – Side view of mixing chamber allows for viewing development of mixing profile
  – Viewing of both the raw images, video and algorithmically processed plots will be utilized
Mission Overview - Mission Objectives

• Inertial Measurement Unit data collection requirements/objectives
  – Inertial measurement unit
    • Z axis accelerometer primary requirement, X and Y axis secondary requirement
  – Power to IMU for duration of flight
  – Data collection and storage

• No overall changes in instrumentation that will alter our goals
  ○ Some changes to the design on the power system, but overall system is the same
    • Will be explained in further detail
Concept of Operations

- Based on science objectives, describe what the payload will be doing during flight and highlights areas of interest (timeline)
- Include all significant events during your flight
- RS-C 2013 Mission timeline included on following slide for guidance
- You may used the example on following slide updated with your mission events
- We will touch quickly on this, but not go into depth unless needed
## Mission Overview: Concept of Operations (area of interest)

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>Time (s)</th>
<th>Event</th>
<th>MCU</th>
<th>Power Bus</th>
<th>Solenoid Valves</th>
<th>RunCam</th>
<th>ArduCam</th>
<th>LED</th>
<th>IMU Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>122.2</td>
<td>39.5</td>
<td>Runcam and LED Activation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>41.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124.4</td>
<td>54.5</td>
<td>Begin Flow Sequence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124.5</td>
<td>56.5</td>
<td>Deactivate Flow Sequence - Continue RunCam Capture for 30s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>56.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>58.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>62.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>71.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124.6</td>
<td>86.5</td>
<td>Deactivate RunCam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>87.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>88.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>89.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>90.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>91.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>92.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>approx-600</td>
<td>No Status Change until Splashdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mission Overview: Concept of Operations

- G switch triggered
- Timer activated
- Begin IMU data collection

$t = 0 \text{ s}$
Alt: 122 km
RunCam and LED activated

$t = 39 \text{ s}$
Alt: 122 km
RunCam and LED activated

$t = 39 \text{ s}$
Alt: 122 km
RunCam and LED activated

$t = 54.5 \text{ s}$
Alt: 124 km
Begin Flow Seq.

$t = 56.5 \text{ s}$
Alt: 124.5 km
Deactivate Flow Seq.

$t = 69.5 \text{ s}$
Apogee

$t = 86.5 \text{ s}$
Alt: 124.56 km
Deactivate RunCam

$t \approx 600 \text{ s}$
Splash Down
Battery Discharges
Changes since STR

• Mechanical
  – Material type for diaphragm
    • Due to plastics lack of resistance to ethanol, the final material for the diaphragm tanks will be metal
    • Plastic tanks will be used for testing purposes
  – No other major changes to the system
• Electrical
  – New and improved power distribution board
• Software
  – no changes
• Numerous small dimensional tweaks for strength/fit
2.0 Subsystem Status
Subsystem Overview

- Subsystems
  - Mechanical
    - FMS
    - FPS
  - Structural
    - STR
  - Electrical
    - PDS
    - DAS
  - Software
    - SW
Software Testing Status

• Completed:
  - Formatting RaspberryPi
  - Installing all software tools and interfaces on RaspberryPi (Raspbian/Python3, Vim, MariaDB, MySQL)
  - Compare storage:
    - SD card: 128GB
    - previously used 34MB for RockOn
  - Determining how to turn the RaspberryPi on and off safely at the addition/removal of power
  - Creating a base image using the ArduCam
  - Enabling RaspberryPi Camera Module for more camera functionality
  - Replaced the ArduCam Lens
  - Enabled Burst Mode for ArduCam
  - Created code for IMU sensor to collect all variables
  - Create a database for the IMU in MariaDB/MySQL
• To do:
  - Fix small bugs in IMU sensor data collection
  - Test ArduCam and IMU sensor together
  - Integrate input/output elements into one script and run with time component
Software ArduCam Testing

Test: ArduCam can focus on an object within 1 inch (Dried Mango)

```python
from picamera import PiCamera
from time import sleep
from subprocess import call
import mysql.connector
import json
from datetime import datetime

camera = PiCamera()
camera.resolution = (1280, 720)
camera.framerate = 30
sleep(5)
camera_shutter_speed = camera.exposure_speed
camera.exposure_mode = 'off'
g = camera.awb_gains
camera.awb_mode = 'off'
camera.awb_gains = g
camera.capture_sequence(['image%d.jpg' % i for i in range(10)])

#camera.capture_continuous(['image%d.jpg' % i for i in range(10)])
camera = PiCamera()
camera.start_preview()
sleep(5)
for i, filename in enumerate(camera.capture_continuous('image%d.jpg')):
    if i == 10:
        break
camera.capture('/home/pi/test01.jpg')
camera.stop_preview()

call('sudo shutdown -h now', shell=True)

for i in range(5):
    now = datetime.now()
    timestamp = datetime.timestamp(now)
    date_time = now.strftime('%d/%m/%Y %H:%M:%S')
    print('date and time:', date_time)
```
import logging
import sys
import time
import mysql.connector
import json
import datetime

from Adafruit_BNO085 import BNO085

def main():
    import logging
    import sys
    import time
    import mysql.connector
    import json
    import datetime
    # Create and configure the BNO sensor connection. Make sure only ONE of the
    # below 'bno = ...' lines is uncommented:
    # Raspberry Pi configuration with serial UART and RST connected to GPIO 18:
    # bno = BNO085.BNO085(serial_port='/dev/serial0', rst=18)
    # BeagleBone Black configuration with default I2C connection (SCL=P9_19, SDA=P9_20)
    # and RST connected to P9_12:
    # bno = BNO085.BNO085(rst='P9_12')
    # bno = BNO085.BNO085(rst=18)

    # Enable verbose debug logging if -v is passed as a parameter.
    if len(sys.argv) == 2 and sys.argv[1].lower() == '-v':
        logging.basicConfig(level=logging.DEBUG)

    # Initialize the BNO085 and stop if something went wrong.
    if not bno.begin():
        raise RuntimeError('Failed to initialize BNO085! Is the sensor connected?')

    # Print system status and self test result.
    status, self_test, error = bno.get_system_status()
    while True:
        try:
            # Initialize the BNO085 and stop if something went wrong.
            if not bno.begin():
                raise RuntimeError('Failed to initialize BNO085! Is the sensor connected?')
            # Print system status and self test result.
            status, self_test, error = bno.get_system_status()
            break
        except Exception as e:
            print('Got error: {!r}'.format(e))

    print('Sleeping is before retrying')
    time.sleep()
Software IMU Sensor Testing

```python
def store_display_data(heading, roll, pitch, sys, gyro, accel, mag, x, y, z, temp_c, xm, ym, zm, xg, yg, zg, xa, ya, za, xla, yla, zla, xga, yga, zga):
    # Load database user credentials from JSON
    credentials = json.load(open("static/credentials.json", "r"))

    # Connect to database
    database = mysql.connector.connect(
            host=credentias["host"],
            user=credentias["user"],
            password=credentias["password"],
            database=credentials["database"]
    )

    # Create cursor object that executes database commands
    cursor = database.cursor()

    # SQL insert statement
    insert_sql = "INSERT INTO IMU_data (timeStamp, heading, roll, pitch, sys, gyro, accel, mag, x_quaternion, y_quaternion, z_quaternion, w_quaternion, temp, x_magnetometer, y_magnetometer, z_magnetometer, x_gyroscope, y_gyroscope, z_gyroscope, x_accelerometer, y_accelerometer, z_accelerometer, x_linear_accel, y_linear_accel, z_linear_accel, x_gravity, y_gravity, z_gravity) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,....
```

Database changed

| ID | Timestamp | heading | roll | pitch | sys | gyro | accel | mag | x_quaternion | y_quaternion | z_quaternion | w_quaternion | temp | x_magnetometer | y_magnetometer | z_magnetometer | x_gyroscope | y_gyroscope | z_gyroscope | x_accelerometer | y_accelerometer | z_accelerometer | x_linear_accel | y_linear_accel | z_linear_accel | x_gravity | y_gravity | z_gravity |
|----|-----------|---------|------|-------|-----|------|-------|-----|             |             |             |             |      |               |               |               |             |             |             |             |               |               |               |               |               |               |         |         |         |
| 1  | 2020-03-10 23:16:16.784692 | 3.3     |      |      |     | 0.3  | 0.32  | 0.03 | 0.18        | 0.18         | 0.18         | 0.18         | 0.18 | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18         | 0.18         | 0.18           | 0.18           | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18 |
| 2  | 2020-03-10 23:16:16.784692 | 3.3     |      |      |     | 0.3  | 0.32  | 0.03 | 0.18        | 0.18         | 0.18         | 0.18         | 0.18 | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18         | 0.18         | 0.18           | 0.18           | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18 |
| 3  | 2020-03-10 23:16:16.784692 | 3.3     |      |      |     | 0.3  | 0.32  | 0.03 | 0.18        | 0.18         | 0.18         | 0.18         | 0.18 | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18         | 0.18         | 0.18           | 0.18           | 0.18           | 0.18           | 0.18           | 0.18         | 0.18         | 0.18 |
Subsystem Testing Status - Mechanical

- Fluid Mixing System
  - Completed
    - Waste tank assembled
      - Gasket added to reduce leaking
    - Mixing chamber cut and tested
      - For final integration we will use Sam’s chamber/header
  - To do
    - Re-print headers?
    - Test Valves using the PDS
    - Test system with valves and fluid actuation
      - Valves arrive mid April

- Fluid Pressurization System
  - Completed
    - System is completely assembled
    - Successful pressure tests completed
Subsystem Testing Status - Mechanical

- Fluid Pressurization System cont.
  - To do
    - Integrate system with the FMS, electrical system, and software
    - Start testing with fluids and valves
      - Valves arrive mid April
    - Machine diaphragm tanks

- Mechanical System completion
Subsystem Testing Status - Structural

Structural Subsystem

- Structure has been assembled
  - In the process of testing different printer settings with prusa PLA, 10% infill
  - Making full structural prototypes and abuse testing them
  - Canister drop tested

Slight .05” gap that needs to be addressed in final
Subsystem Testing Status - Electrical

Quick Status

- Original power distribution board had unresolved issues (12V system inconsistent)
- New board/components designed and ordered
- RPI breakout board/LED testing complete
Power Distribution Board V2

• Original board suspect to noise issues resulting from the 3 high frequency switching circuits in close proximity

• New board simplified to only include 5V and 12V circuits (3.3V components run entirely off Raspberry Pi)

• Top/bottom ground planes provide low impedance connections, EMI shielding and heat sinks for regulators

• High frequency component separation

• Looks cooler

• Board and components ordered (~1 week until ready to test)
Power Distribution Board V2 Schematic
Power Distribution Board V2

Lee Driver integration points on board
Battery
3.7V 2500mAh

Power Distribution
12V
5V

G-Switch
RBF

Valve Control
Lee Valve

MCU
IMU
Arducam

Run Cam

Switch Circuit

3.3V

Power
Data in
Control
12V Circuit

- **MCU**
  - 0V - Off
  - 3.3V - On

- **Valve Control**
  - 12V

- **Lee Valve**
  - 12V

- **12V Regulator**
  - 12V

- **Power**: Red lines indicating power flow
- **Data in**: Blue lines indicating data input
- **Control**: Pink lines indicating control signals
5V Circuit

- **Power**
- **Data in**
- **Control**

- **0V - Off**
- **3.3V - On**

- **5V Regulator**
  - 5V
  - 5V
  - 5V

- **MCU**
- **Run Cam**
- **IMU**
- **Arducam**

- **Switching Circuit**
  - 3.3V
  - 5V

- **LED**

From top to bottom:
- The 5V Regulator outputs 5V to the MCU and Run Cam.
- The MCU and Run Cam are powered by 5V.
- The IMU and Arducam are connected to a 3.3V - 0V power switching circuit.
- The Switching Circuit provides 3.3V to the LED and controls the power flow to and from the 5V Regulator.
Backup Option

• In the event that the new power board is not functional, aftermarket backup options have been found to ensure mission success is not risked.

Board comes in 12V and 5V output variations using an input voltage as low as 2.9V (battery is ~3.7V output).
LED testing

Testing Conditions:
- f/2.4
- 1/1000 shutter speed
- ISO 100
~1 inch from spaghetti

Testing Conditions:
- f/1.8
- 1/1000 shutter speed
- ISO 100
~1 inch from spaghetti
LED testing results

- These results provide a rough idea as to what our mixing chamber will look like (mission conditions should be more favorable as the LED will backlight the mixing chamber)
RPI Breakout testing

- Multimeter used to ensure pins are all connected as designed (MM displays voltage drop between probes so 0V indicates short circuit, pins are connected properly)
RunCam testing

- RunCam tested using monitor to ensure it works

- Further testing will be performed after payload construction (during mission simulations)
3.0 Integrated Subsystem Testing Status
List of Subsystem integration

Subsystems will be integrated in this order (this is subject to change)

- Fluid mixing system and fluid pressurization systems are almost fully built
  - FMS and FPS will be integrated together first
- The electrical system (PDS) and software components will then be integrated simultaneously with the mechanical components
Integrated Subsystem Testing Status

- **Mechanical Systems**
  - The FPS will be integrated with the FMS upon the arrival of the valves and the machining of the diaphragm tanks
    - This will verify and allow for many tests
      - Verifies all unions and tubing fit
      - Verifies the pressure in the system remains at 5 psi when fully integrated
      - Verifies that fluid flows through the valves and mixes
      - Allows for leak tests
      - Allows for RunCam and Arducam tests
        - Will be able to process the images and test our analysis methods

- **Mechanical Integration with Electrical and Software Systems**
  - The power distribution board will be integrated with all systems to open up the ability for more mission similar testing (full payload)
Mechanical Integration with FPS, FMS, DAS
Integrated Subsystem Testing Status

- Mechanical integration with Electrical and Software Systems cont.
  - The power distribution system drives the whole experiment
    - Upon integration with the FMS, FPS and SW we will be able to
      - Test the Raspberry Pi’s system turn on/power off at the addition of power
      - Test the software input elements (ArduCam and IMU)
        - Test whether ArduCam takes an appropriate amount of quality images for necessary data collection
      - Use images produced by the ArduCam to run image processing in OpenCV
      - Test software output elements (LED and Solenoid Valves)
      - Test system run time and adjust timing of input/output components
Integrated Subsystem Testing Status

● Mechanical integration with Electrical and Software Systems cont.
  ○ The power distribution system drives the whole experiment
    ■ Upon integration with the FMS, FPS and SW we will be able to
      ● Run mission simulated trials to ensure battery is able to provide enough power
      ● Run payload in repeated mission similar conditions until battery fails
      ● Run payload in full mission conditions then let payload sit idle to simulate splashdown (wait for battery to die to ensure no data is corrupted on loss of power)
4.0 Plan for Full System Integration
Mechanical Verification

• Pressure tests
  – Verify that the pressure in the system holds at 5 psi
  – Initial test has already been done and system held at 5 psi.
    • Additional pressure tests will be done upon full integration of the FMS and FPS
    • This will occur when valves arrive and diaphragm tanks are in hand (≈ April 14)

• Connectivity and Spatial Tests
  – Verify that all tubing and unions fit with no leakage
    • FPS has been built, all connections fit as expected
  – Verify that both the FMS and FPS fit in the housing
  – These tests will be performed when valves arrive and diaphragm tanks are machined allowing for full integration between both mechanical subsystems

• Leak Tests
  – Testing fluid actuation and if any leaking occurs in fluid lines, mix, chamber, or waste tank
    • Will need almost all systems integrated to perform full leak test
Mechanical Verification

• Overview
  – The three listed tests are the main indications of a working mechanical system
  – Many additional tests will be performed with full system integration to fine tune and calibrate all instruments
  – These tests will be performed as soon as valves are in hand (≈ April 14)
Integration Procedure

- General Outline (full system integration)
  - FMS and FPS built and integrated into housing
  - Power distribution board built and integrated into housing along with software components
    - RunCam
    - ArduCam
    - Raspberry Pi
    - IMU
Vibe Check

Backup parts will be brought

- Backup structure
- Backup wiring
- Only one perfect mixing chamber
  - Backup composite
  - Test final flight hardware or test composite?
Electrical Verification/testing

- The main test for the PDS will be running the payload in mission simulated conditions. This will be performed as soon as the payload is finished being constructed (≈ April 14)
- Secondary tests include running all electrical components to see how long the battery will last under full load as well as running the components under no load to find quiescent power draw
- Ensuring payload is electrically isolated from rocket body through continuity tests with multimeter
Overview of Full System Tests

- Main test will be to run full mission like trials of fluid actuation via the Raspberry Pi
- Have not performed this high level test yet
  - Target for these is end of April
5.0 Project Management Update

Name of Presenter
# Updated Schedule

<table>
<thead>
<tr>
<th>UVM RSC 2020 Schedule</th>
<th>Start Date</th>
<th>End Date</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mech Order</td>
<td>Jan-20</td>
<td>Feb-15</td>
<td></td>
</tr>
<tr>
<td>Lee Order</td>
<td>Feb-10</td>
<td>April-10</td>
<td></td>
</tr>
<tr>
<td>Electronics Order</td>
<td>Jan-20</td>
<td>Feb-15</td>
<td></td>
</tr>
<tr>
<td>Prototype Enclosure</td>
<td>Feb-18</td>
<td>Feb-20</td>
<td></td>
</tr>
<tr>
<td>FPS Assembly</td>
<td>Feb-10</td>
<td>Feb-29</td>
<td></td>
</tr>
<tr>
<td>FMS Assembly</td>
<td>Feb-24</td>
<td>Feb-29</td>
<td></td>
</tr>
<tr>
<td>Pi Testing</td>
<td>Feb-8</td>
<td>April-30</td>
<td></td>
</tr>
<tr>
<td>RunCam Testing</td>
<td>Feb-8</td>
<td>April-30</td>
<td></td>
</tr>
<tr>
<td>LED Testing</td>
<td>Feb-10</td>
<td>Feb-13</td>
<td></td>
</tr>
<tr>
<td>PCB Building</td>
<td>Feb-18</td>
<td>April-7</td>
<td></td>
</tr>
<tr>
<td>PCB Testing</td>
<td>February-2</td>
<td>April-30</td>
<td></td>
</tr>
<tr>
<td>Subsystem Level Integration</td>
<td>March-1</td>
<td>April-30</td>
<td></td>
</tr>
<tr>
<td>ISTR</td>
<td>April-3</td>
<td>April-3</td>
<td></td>
</tr>
<tr>
<td>Final PCB</td>
<td>April-7</td>
<td>April-8</td>
<td></td>
</tr>
<tr>
<td>IMU tests</td>
<td>April-6</td>
<td>April-13</td>
<td></td>
</tr>
<tr>
<td>ArduCam and IMU tested together</td>
<td>April-6</td>
<td>April-13</td>
<td></td>
</tr>
<tr>
<td>Software component tests with &quot;run time&quot; tests</td>
<td>April-6</td>
<td>April-13</td>
<td></td>
</tr>
<tr>
<td>Valve testing</td>
<td>April-13</td>
<td>April-30</td>
<td></td>
</tr>
<tr>
<td>Integrated FMS, FPS, PDS tests</td>
<td>April-13</td>
<td>May-4</td>
<td></td>
</tr>
<tr>
<td>FMSTR</td>
<td>April-27</td>
<td>April-27</td>
<td></td>
</tr>
<tr>
<td>Full System tests</td>
<td>May-4</td>
<td>May-30</td>
<td></td>
</tr>
</tbody>
</table>
Schedule Notes

• Slightly behind schedule due to malfunction in our PDS forcing a redesign of the system
  – New board has been designed/ordered and backup plan in place
• Also due to COVID-19 some of the initial plans for manufacturing parts have fallen through
  – 3D printing headers for the mixing chamber
• Overall no huge concerns, have not fallen too far behind schedule
Summary of Progress

• Behind schedule in terms of subsystem integration
  – Each individual subsystem is progressing but need to start testing subsystems together
  – USPS used for subsystem integration transportation
  – New PCB built and ordered
    • Will allow us to get back on track with multi-system tests before the Mission Simulation report
  – Solenoid valves arriving soon will also open up the gate for significant multi-system tests and allow us to get back on schedule
  – Software is ready for system level testing, soon to be ready for subsystem integration with electrical elements
• The plan is to keep moving forward with individual subsystems, when PCB, valves, and tanks come in begin significant system level testing and integration
• Currently learning OpenCV in preparation to process test images
### User Guide Compliance

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Status/Reason (if needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center of gravity in 1&quot; mid-can?</td>
<td></td>
</tr>
<tr>
<td>Contained in can</td>
<td></td>
</tr>
<tr>
<td>Connected to can by 4/5 bulkheads on top and bottom only</td>
<td></td>
</tr>
<tr>
<td>Mass at 20±0.2lbs</td>
<td>Need to weigh and find mass</td>
</tr>
<tr>
<td>Shared canister clearance</td>
<td></td>
</tr>
<tr>
<td>No voltage on the can</td>
<td></td>
</tr>
<tr>
<td>Activation wires at least 4 ft</td>
<td></td>
</tr>
<tr>
<td>Activation wire at least 24 gauge</td>
<td>28 AWG multi strand teflon insulated</td>
</tr>
<tr>
<td>Early Activation: current &lt; 1 A</td>
<td>Not using</td>
</tr>
<tr>
<td>T-0 Activation: current &lt; .1 A</td>
<td>Using (no current until G-Switch activation)</td>
</tr>
<tr>
<td>Battery Type</td>
<td>Lithium Ion Polymer (will not charge at Wallops)</td>
</tr>
</tbody>
</table>
Shared Can Logistics

• Who are you sharing with?
  – We are sharing with a Rock-On team
  – Rock-On’s mission is to collect data with various sensors and a Geiger counter

• Plan for collaboration
  – The Rock-On team will not be making the payload until June
  – However we know exactly the size, weight, and fit of their payload

• Are you mounting to the top or the bottom? Are you using a mid mounting plate?
  – Bottom

• Who needs what port?
  – Neither team will require a port
## Budget - $2500

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Actual Part</th>
<th>Final Price (including s&amp;h)</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adafruit</td>
<td>IMU/Battery/Charger</td>
<td>101.64</td>
<td></td>
</tr>
<tr>
<td>Runcam</td>
<td>Runcam</td>
<td>69.99</td>
<td></td>
</tr>
<tr>
<td>Pishop</td>
<td>Raspberry Pi</td>
<td>62.95</td>
<td></td>
</tr>
<tr>
<td>Robotshop</td>
<td>ArduCam</td>
<td>81.54</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>Micro SD</td>
<td>26.16</td>
<td></td>
</tr>
<tr>
<td>Bh photo</td>
<td>Runcam lens</td>
<td>85.6</td>
<td></td>
</tr>
<tr>
<td>JLCPCB</td>
<td>PCB’s</td>
<td>23.99</td>
<td></td>
</tr>
<tr>
<td>MatterHacker</td>
<td>PLA for 3D printing</td>
<td>157.95</td>
<td></td>
</tr>
<tr>
<td>Eplasitcs</td>
<td>Epoxy</td>
<td>45.24</td>
<td></td>
</tr>
<tr>
<td>Beswick</td>
<td>All Beswick</td>
<td>338.3</td>
<td></td>
</tr>
<tr>
<td>Swagelok (order one)</td>
<td>Fittings</td>
<td>46.8</td>
<td></td>
</tr>
<tr>
<td>McMaster (order one)</td>
<td>Misc</td>
<td>112.09</td>
<td></td>
</tr>
<tr>
<td>Digikey</td>
<td>Misc electronics</td>
<td>44.83</td>
<td></td>
</tr>
<tr>
<td>Mouser</td>
<td>Misc electronics</td>
<td>23.2</td>
<td></td>
</tr>
<tr>
<td>Mouser</td>
<td>Misc electronics</td>
<td>12.46</td>
<td></td>
</tr>
<tr>
<td>Lee</td>
<td>Lee Valves</td>
<td>1020</td>
<td></td>
</tr>
<tr>
<td>Amazon (order two)</td>
<td>Acrylic tubing</td>
<td>19.25</td>
<td>-19.25</td>
</tr>
<tr>
<td>Swagelok (order two)</td>
<td>Fittings</td>
<td>46.8</td>
<td></td>
</tr>
<tr>
<td>Protolab</td>
<td>Mixing Header</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Clippard</td>
<td>avt 24-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remington</td>
<td>Hookup wires</td>
<td>28.4</td>
<td>-28.4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>2399.54</strong></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Concerns
  – Headers for mix chamber
    • Potentially buy them from Proto labs
  – Coronavirus
  – Power Board V2 doesn't work forcing us to use backup boards (slightly harder to integrate)

• Overall in good shape for meeting goals by the Mission Sim report