

LaACES Program

Flight Readiness Review Document

for the

DemonSats-1

Experiment

by

Team

DemonSats-1 from Northwestern State University

Prepared by:

<u>Holden Rivers</u>	Jun 19, 2020
Holden Rivers - Team Captain, NSU	Date
<u>Jesse Coats</u>	Jun 18, 2020
Jesse Coats, NSU	Date
<u>Savon Gipson</u>	Jun 18, 2020
Savon Gipson, NSU	Date
<u>Harley Godwin</u>	Jun 18, 2020
Harley Godwin, NSU	Date
<u>Carlee Lake</u>	Jun 17, 2020
Carlee Lake, NSU	Date
<u>Abigail Poe</u>	Jun 19, 2020
Abigail Poe, NSU	Date
<u>Charles Beam</u>	Jun 18, 2020
Charles Beam, LSMSA	Date

Submitted:

<u>Abigail Poe</u>	Jun 19, 2020
--------------------	--------------

Reviewed/Revised:

<u>Holden Rivers</u>	Jun 19, 2020
----------------------	--------------

Approved:

<u>Anna Dugas</u>	Jun 19, 2020
Faculty Mentor: Anna Dugas	Date
<u>Christopher Lyles</u>	Jun 19, 2020
Department Head: Christopher Lyles	Date

LaACES Signoff Date

Change Information Page

Title: FRR Document for DemonSats-1 Experiment

Date: 06/19/2020

Status of TBDs

TBD Number	Section	Description	Date Created	Date Resolved
1	3.3.2	Add to technical objectives about conforming with LaACES requirements	4/14/2020	6/8/2020
2	3.3.2	Address sampling rate	4/14/2020	6/8/2020
3	3.4.1	Add information regarding flight characteristics and LaACES requirements for payload	4/14/2020	6/8/2020
4	3.4.2	Clarify science and technical requirements	4/14/2020	6/8/2020
5	3.4.2	Add LaACES technical requirements	4/14/2020	6/8/2020
6	4.2.3	Rework scientific and technical objective traceability charts	4/14/2020	6/10/2020
7	4.2.3	Address reasoning for using NOAA soundings	4/14/2020	6/10/2020
8	4.2.3	Address reasoning for collecting latitude, longitude, gyroscope, and accelerometer data	4/14/2020	6/10/2020
9	4.4.1	Address sampling rate frequency	4/14/2020	6/10/2020
10	4.4.2	Create flow charts for software including setup, loop, and all subroutines	4/14/2020	6/13/2020
11	4.4.2	Add further information about ground software including flow charts	4/14/2020	6/13/2020
12	4.4.2	Remove NOAA soundings in analysis software including a flow chart	4/14/2020	6/10/2020
13	4.6.1	Discuss mechanical design of payload box	4/14/2020	6/8/2020
14	4.6.2	Address weight budget issues of source, uncertainty, and contingency	4/14/2020	6/10/2020
15	5.0, 6.1, 6.2, and 6.3	Address how team completed payload during COVID-19 crisis	4/14/2020	6/13/2020
16	7.3	Elaborate on ground software design including calibrations, calculations, and data analysis in Excel and create flow charts	4/14/2020	6/18/2020
17	8.2.2	Add COVID-19 accommodations	6/2/2020	6/10/2020

TABLE OF CONTENTS

Cover.....	i
Change Information Page	ii
Status of TBDs	iii
Table of Contents.....	iv-v
List of Figure.....	vi-vii
List of Tables	viii
1.0 Document Purpose	1
1.1 Document Scope	1
1.2 Change Control and Update Procedures	1
2.0 Reference Documents	2
3.0 Goals, Objectives, Requirements	3
3.1 Mission Goal.....	3
3.2 Objectives	3
3.3 Science Background and Requirements	4-7
3.4 Technical Background, Requirements, and Goals.....	7-9
4.0 Payload Design	9
4.1 Principle of Operation.....	9
4.2 System Design	10-14
4.3 Electrical Design.....	14-29
4.4 Software Design.....	30-33
4.5 Thermal Design.....	34
4.6 Mechanical Design.....	34-36
5.0 Payload Development Plan	37
5.1 Electrical Design Development	37
5.2 Software Design Development	37
5.3 Mechanical Design Development	38
5.4 Mission Design Development.....	38-39
6.0 Payload Construction Plan.....	39
6.1 Hardware Fabrication and Testing.....	39-49
6.2 Integration Plan.....	49-50
6.3 Software Implementation and Verification.....	50
6.4 Flight Certification Testing.....	50-56
7.0 Mission Operations	57
7.1 Pre-Launch Requirements and Operations	57-62
7.2 Flight Requirements, Operations and Recovery	62-63
7.3 Data Acquisition and Analysis Plan	63-65
8.0 Project Management	66

8.1 Organization and Responsibilities	66
8.2 Configuration Management Plan	66-70
8.3 Interface Control	70
9.0 Master Schedule.....	70
9.1 Work Breakdown Structure (WBS)	70
9.2 Staffing Plan.....	71
9.3 Timeline and Milestones.....	71-72
10.0 Master Budget.....	73
10.1 Expenditure Plan.....	73
10.2 Material Acquisition Plan	74
11.0 Risk Management and Contingency	74-75
12.0 Glossary	76

LIST OF FIGURES

1. Graph of saturation vapor pressure as a function temperature.....	4
2. Figure of atmospheric temperatures (a) and pressures (b) for altitudes above the earth's sea level....	5
3. Graph of the expected saturation vapor pressure over flight altitudes at 30°C.....	6
4. High-level schematic of MegaSat system design.....	10
5. Schematic of MegaSat group interfaces.....	11
6. Schematic of RTC circuit of MegaSat.....	15
7. Schematic of 1N457 diode temperature sensor circuit of MegaSat.....	16
8. Schematic of pressure sensor circuit of MegaSat.....	18
9. Graph of expected output voltage vs % relative humidity.....	19
10. Schematic of humidity sensor circuit of MegaSat.....	20
11. Pinout of ultimate GPS module to interface with MegaSat.....	21
12. Schematic of MPU-6050 module interface with MegaSat.....	23
13. Image of MPU6050 with AD0 pin solder to high.....	24
14. Mid-level block diagram of the MegaSat.....	26
15. Diagram of pathway from sensor input to data storage.....	26
16. Pinout of Arduino Mega 2560 interface with MegaSat.....	27
17. Image of battery pack.....	28
18. Schematic of power supply circuit MegaSat.....	28
19. Drawing of flight string channels for DemonSats-1 enclosure.....	35
20. Images of payload box.....	35
21. Images of external sensor placement on box.....	36
22. Image of completed payload on scale.....	36
23. Graph of voltage output over temperature for red-coated external temperature sensor.....	40
24. Images of temperature calibration experimental setup.....	41
25. Image of pressure calibration experimental setup.....	41
26. Initial results of MegaSat pressure sensing system output using vacuum chamber.....	42
27. Example of pressure sensing system response compared to a timer.....	42
28. Effect of gain potentiometer adjustments on the %RH range.....	44
29. Image of humidity calibration setup example.....	45
30. Comparison of MegaSat humidity sensing system and ADC sensor outputs.....	45
31. Road trip data comparing Adafruit GPS module's altitude measurement with Coordinates iPhone GPS tracker for elevation data.....	46
32. Road trip data comparing Adafruit GPS module, Sparkfun BME280 sensor, and Coordinates iPhone GPS tracker for elevation data.....	47
33. Map of converted data from Adafruit GPS module collected during road trip.....	48
34. Map showing accuracy of GPS coordinates using Adafruit GPS module.....	48
35. Image of fridge system test example.....	51
36. Temperature trends of payload sensors for refrigerator system test compared to type-K thermocouples.....	53
37. Temperature trends of payload sensors for freezer system test compared to type-K thermocouples.....	53
38. Percent relative humidity trends of different humidity sensors in fridge system test at 4°C.....	54

39. Percent relative humidity trends of different humidity sensors in fridge system test at -15°C.....	54
40. Percent relative humidity trends of different humidity sensors during vacuum test.....	56
41. Calibration results for external temperature sensor compared with type-K thermocouples.....	58
42. Calibration results for internal temperature sensor compared with type-K thermocouples.....	59
43. Calibration results for MegaSat humidity sensor in saturated salt solutions.....	60
44. Calibration results for additional humidity sensor in saturated salt solutions.....	61
45. Calibration results for MegaSat pressure sensor compared to oil-filled Bourdon tube gauge.....	62
46. Flowchart of post-flight data analysis plan.....	63
47. Flowchart of post-flight software plan for DemonSats-1 payload.....	64
48. Organization chart.....	66
49. Map of route to deliver payload to LSU.....	68
50. Image of approximate start and end times and total duration of trip to deliver payload.....	68
51. Work breakdown structure for DemonSats-1 project.....	70

LIST OF TABLES

1. Coefficients for vapor pressure of pure water as a function of temperature.....	5
2. Traceability of science objectives.....	12
3. Traceability of technical objectives.....	13-14
4. Traceability of technical goals.....	14
5. List of sensors used in the MegaSat.....	14-15
6. List of saturated salts to be used in humidity sensor calibration.....	21
7. I2C register map for data used from MPU-6050.....	23
8. I2C register map for data used from BME 280.....	25
9. Recommended settings of BME 280 for highest accuracy in altitude.....	25
10. Power budget for DemonSats-1 Payload.....	29
11. Output data characters to be stored to microSD card.....	30
12. Flight software plan for DemonSats-1 payload.....	31
13. Flight software main loop plan for DemonsSats-1 payload.....	32
14. Post-flight software main loop plan for DemonSats-1 payload.....	33
15. Mass budget for DemonSats-1 payload.....	36
16. Foam board template for final DemonSats-1 payload enclosure.....	38
17. Electronic checkpoints for MegaSat Shield.....	39
18. List of pressure conversions used for pressure calibration data.....	43
19. Saturated salts used in initial humidity sensor calibration.....	43
20. List of offset values for MPU6050 data for DemonSats-1.....	49
21. Pressure results from vacuum test.....	56
22. List of designated team representatives for limited attendance launch.....	69
23. Staffing Plan for DemonSats-1 Project.....	71
24. Major milestones.....	71
25. Gantt chart for DemonSats-1 Project.....	72
26. Expenditures for payload.....	73
27. Locations of materials acquisition.....	74
28. Risk event analysis.....	74-75
29. Risk contingency plan.....	75

1.0 Document Purpose

This document describes the design for the DemonSats-1 experiment by Team DemonSats from Northwestern State University for the LaACES Program. It fulfills part of the LaACES Project requirements as the Flight Readiness Review (FRR) and will be submitted by June 19, 2020.

1.1 Document Scope

This FRR document specifies the scientific and technical purpose and requirements for the DemonSats-1 experiment and provides a guideline for the development, operation, and cost of this payload under the LaACES Project. The document includes details of the payload design, fabrication, integration, testing, flight operation, and data analysis. In addition, project management, timelines, work breakdown, expenditures, and risk management are discussed.

1.2 Change Control and Update Procedures

Changes to this FRR document shall only be made after approval by designated representatives from Team DemonSats and the LaACES Institution Representative Asst. Professor Anna Dugas. Document change requests should be sent to the LaACES Institution Representative at dugasa@nsula.edu.

2.0 Reference Documents

1. Granger, D. and J. Collins, *MegaSat Assembly Manual*, Louisiana State University, Baton Rouge, LA, July 2019.
2. Buck, A.L., "New Equations for Computing Vapor Pressure and Enhancement Factor," *Journal of Applied Meteorology*, Volume 20, December 1981, pgs. 1527–1532.
3. Rauber R., J. Walsh, Charlevoix, *Severe and Hazardous Weather: An Introduction to High Impact Meteorology*, 3rd ed., 2009
4. Kay, L., S. Palen, B. Smith, and G. Blumenthal, *Twenty-first Century Astronomy*, 4th. ed., New York, NY, 2013.
5. Nullet, D., "The Natural Environment," *Geography 101*, Kapiolani Community College [Online]. Available: https://laulima.hawaii.edu/access/content/group/2c084cc1-8f08-442b-80e8-ed89faa22c33/book/chapter_5/humidity.htm
6. Engineering ToolBox, (2004). *Relative Humidity in Air*. [Online] Available at: https://www.engineeringtoolbox.com/relative-humidity-air-d_687.html
7. ACES-62 Flight Info. Available: <https://laspace.lsu.edu/aces/flightinfo/?fname=ACES-62&fname=ACES-62>
8. Adams, C. "The MegaSat Project: High-Level/ Mid-Level Overview" presented on 03/21/2019.
9. "DS3231: Extremely Accurate I2C-Integrated RTC/TCXO/Crystal" *Maxim Integrated*. Datasheet 19-5170, Rev. 10, 03/2015.
10. "1N457/A" *Fairchild Semiconductor*. Datasheet 1N457/A, Rev. A, 2002.
11. Adams, C. and J.C. Collins. "MegaSat_Full_Schematic_PDF" Revision 2.3. 02/18/2019
12. "1230: UltraStable Specifications." *TE connectivity*. Datasheet 09/2017.
13. "HIH-4000 Series: Humidity Sensors." *Honeywell*. Datasheet 00917-5-EN. 2010.
14. Greenspan L., "Humidity Fixed Points of Binary Saturated Aqueous Solutions." *Journal of Research of the National Bureau of Standards – A. Physics and Chemistry*, Volume 81A, No. 1, January–February 1977.
15. Delano. "FGPMMOPA6H GPS Standalone Module Data Sheet" *Global Top Technology, Inc.* Rev. V0A. 01/31/2012.
16. Gakstatter, E. What Exactly is GPS NMEA Data? <https://www.gpsworld.com/what-exactly-is-gps-nmea-data/>
17. "MPU-6000 and MPU-6050 Product Specification" InvenSense. Rev. 3.4 08/19/2013. Available: <http://43zrtwysvxb2gf29r5o0athu.wpengine.netdna-cdn.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
18. "Sparkfun Atmospheric Sensor Breakout – BME280." *SparkFun Electronics*. Product page. <https://www.sparkfun.com/products/13676>
19. Foamular PROPINK IS Extruded Polystyrene (XPS) Rigid Foam Insulation. *Owens Corning*. Product Data Sheet. Pub. No. 57862-G. July 2018.
20. "Constructing a Standard LaACES Payload Box" *LaACES Student Ballooning Course*. LSU 01/15/2020.
21. Morris A., R. Langari. *Measurement & Instrumentation: Theory & Application*. Elsevier. 2nd ed. 2016, pgs. 463-491.
22. Engineering ToolBox, (2003). *Vacuum Pressure - Units Converter*. [online] Available at: https://www.engineeringtoolbox.com/vacuum-converter-d_460.html.

(Shown as [#] in text)

3.0 Goals, Objectives, Requirements

3.1 Mission Goal

For this first LaACES project of NSULA/LSMSA collaboration, the team plans to construct a payload to be launched by atmospheric sounding balloon to 100,000 ft (30,480 m) while measuring atmospheric temperature, pressure, % relative humidity, and altitude. The students will use this data to explore the psychrometric principle of saturated vapor pressure as a function of altitude. Prior to this experiment, the team will perform calibration testing on each sensor, with focus on reading accuracy and sampling frequency.

3.2 Objectives

The team will construct and calibrate the LaACES MegaSat payload ^[1] to be attached to the LaACES balloon system to be launched in southern central Louisiana this summer. The payload will conform to all requirements designated by the LaACES program.

3.2.1 Science Objectives

- Determine the profiles of temperature, atmospheric pressure, and % relative humidity with altitude.
- Use the external temperature and pressure data from the payload flight and the Arden Buck Method ^[2] to calculate the saturated (equilibrium) vapor pressure of moist air with altitude.
- Compare % relative humidity readings and the equilibrium vapor pressure with altitude to determine the vapor pressure (absolute humidity) during flight.

3.2.2 Technical Objectives

- Construct the MegaSat payload, recently designed at LSU A & M, using an Arduino Mega control system with GPS, motion, pressure, humidity, and temperature sensing and a real-time clock (RTC).
- Ensure that the payload falls within the LaACES constraints for weight and shape and meets LaACES design requirements to allow flight vehicle interface.
- Calibrate each analog sensor for accuracy in measurement.
- Verify the survivability of the payload and functionality of sensors under expected flight conditions through thermal and vacuum tests.

3.3 Science Background and Requirements

3.3.1 Science Background

Vapor pressure, also known as absolute humidity, is a measurement of the amount of moisture in the air. This quantity is difficult to measure directly with instrumentation, so the humidity of an environment is often measured as a ratio called relative humidity. Relative humidity compares the actual vapor pressure to the saturation, or equilibrium, vapor pressure at a given temperature (See Equation 1). The saturation pressure is the state in which pure water vapor is in stable thermodynamic equilibrium with a planar surface of pure water or ice.^[2]

$$\%RH = \frac{\text{vapor pressure}}{\text{equilibrium vapor pressure}} \quad (\text{Eqn. 1})$$

Frequently, the saturation or equilibrium vapor pressure is determined as a function of temperature alone (See Figure 1). However, according to Buck in 1981,^[2] calculating the equilibrium vapor pressure in meteorology, where the temperatures (-80 to 50 °C) and pressures (10 to 1015 millibars) vary drastically, is more accurate when determining it as a function of both temperature and pressure (See Figures 2a,2b). Therefore, the DemonSats group has tasked themselves to learn and utilize the Arden Buck Method^[2] for calculating the equilibrium vapor pressure during their LaACES experiment.

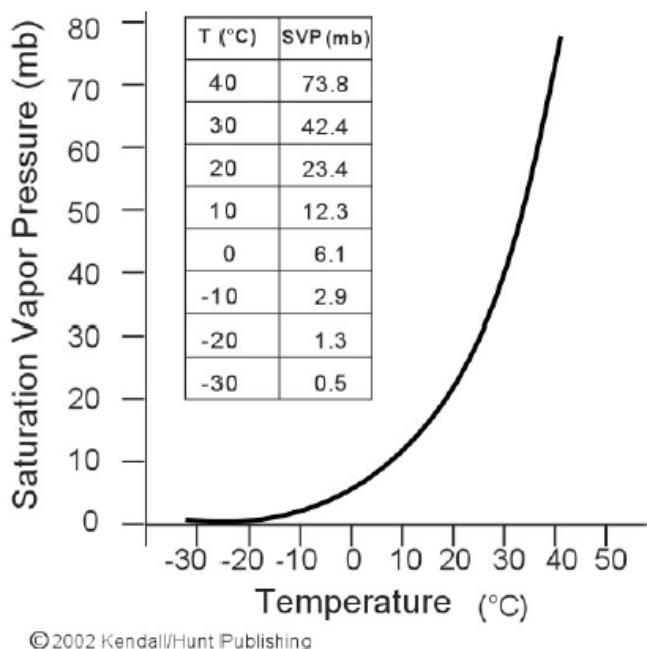


Figure 1: Saturation Vapor Pressure as a Function of Temperature^[3]

In the Buck paper, several equation coefficients were generated to ascertain the best models over the meteorologically appropriate temperatures and pressures. The DemonSats team has reviewed the graphs of these models from the paper and determined that the saturation vapor

According to the Buck paper "New Equations for Computing Vapor Pressure and Enhancement Factor," saturation pressure can be calculated as:

$$e'_w = f_w * e_w \quad \text{for water} \quad (\text{Eqn. 2})$$

and

$$e'_i = f_i * e_i \quad \text{for ice } < 0 \text{ } ^\circ\text{C} \quad (\text{Eqn. 3})$$

where f_w or f_i are enhancement factors created for the Arden Buck method and e_w or e_i are saturation pressures determined by temperature only. The enhancement factors (f) incorporate pressure data into the saturation vapor pressure calculations.

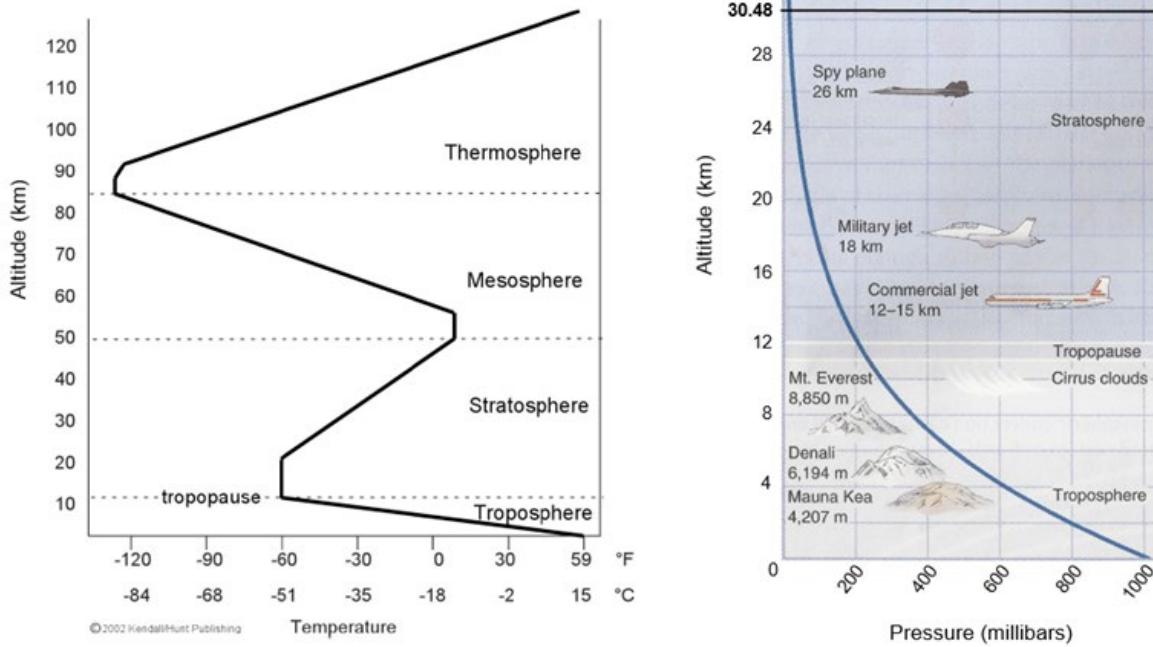


Figure 2: Atmospheric Temperatures (a) and Pressures (b) for Altitudes above the Earth's Sea Level. [3,4]

pressure equations which will be used for data analysis post flight will be from the models:

$e'_w = f_{w3} * e_{w2}$ for the temperature interval of 0 to 50 °C and $e'_i = f_{i3} * e_{i1}$ for the temperature interval of -80 to 0 °C. Table 1 shows the values of the coefficients for each model. These models were selected by the team based on their behavior within the specified temperature ranges given above when compared to values from the Wexler's equations (1976-7) and pressure ranges when compared to Hyland's values (1975).^[2] The maximum relative errors of the e_w models when compared to Wexler's values are included on Table 1 and show that the largest errors generated in these models occur on the outer edges of the temperature ranges. These temperature ranges should be outside the temperatures in which our experiment will occur.

Table 1: Coefficients for Vapor Pressure (in mb) of Pure Water as a Function of Temperature in °C [2]

Curve	a	b	c	Temperature Interval (°C)	Max. Rel. Error and Location (% and °C)
e_{w2}	6.1121	17.368	238.88	0 to 50	0.05(50)
e_{i1}	6.1115	22.542	273.48	-80 to 0	0.14(-80)
f_{w3}	$7 \cdot 10^{-4}$	$3.46 \cdot 10^{-6}$	--	0 to 50	--
f_{i3}	$3 \cdot 10^{-4}$	$4.18 \cdot 10^{-6}$	--	-80 to 0	--

Therefore, the equations which the team will use to calculate the saturated vapor pressure for our experiment will be:

$$e'_w = \left[1.0007 + \left((3.46 \cdot 10^{-6}) * P \right) \right] * 6.1121 * e^{\frac{17.368*T}{238.88+T}} \quad \text{for } 0 \text{ to } 50 \text{ °C} \quad (\text{Eqn. 4})$$

$$e'_i = \left[1.0003 + \left((4.18 \cdot 10^{-6}) * P \right) \right] * 6.1115 * e^{\frac{22.542*T}{273.48+T}} \quad \text{for } -80 \text{ to } 0 \text{ °C} \quad (\text{Eqn. 5})$$

where, T = temperature in degrees C and P = atmospheric pressure in millibars

The saturation vapor pressure in our experiment will be compared to the altitude effects on temperature and pressure. Then the saturation vapor pressure can be compared to the relative humidity (%RH) readings from the humidity sensor to determine the actual vapor pressure over altitude during flight (See Eqn. 1).

To verify the calculations of this experiment, there will be two indicators that the team will be looking for when reviewing the results of the aforesaid calculations using the in-flight data: the humidity change due to clouds (if available) and the tropopause transition, where the relative humidity significantly decreases after rising above the troposphere. If the sounding balloon goes through clouds, we should see an increase in %RH readings and the resulting absolute vapor pressure. By the tropopause transition, the saturation vapor pressure should be minimal, and the relative humidity should be very low as well. The saturation vapor pressure should steadily decrease from around 40 millibars (if ground temperature is 30 °C) until around 6000 m, where it will remain at less than 1 millibar. In Figure 3, an initial plot has been made of the expected saturation vapor pressure with respect to altitude. The trend and magnitude of the saturated vapor pressure in this plot are similar to other references. [5, 6] Since the relative humidity can fluctuate with weather conditions, the team has not developed a preliminary plot for the absolute humidity. Instead, the relative humidity will be compared to relative humidity data from the closest National Weather Service station to the location from which the launch occurs during the time of flight.

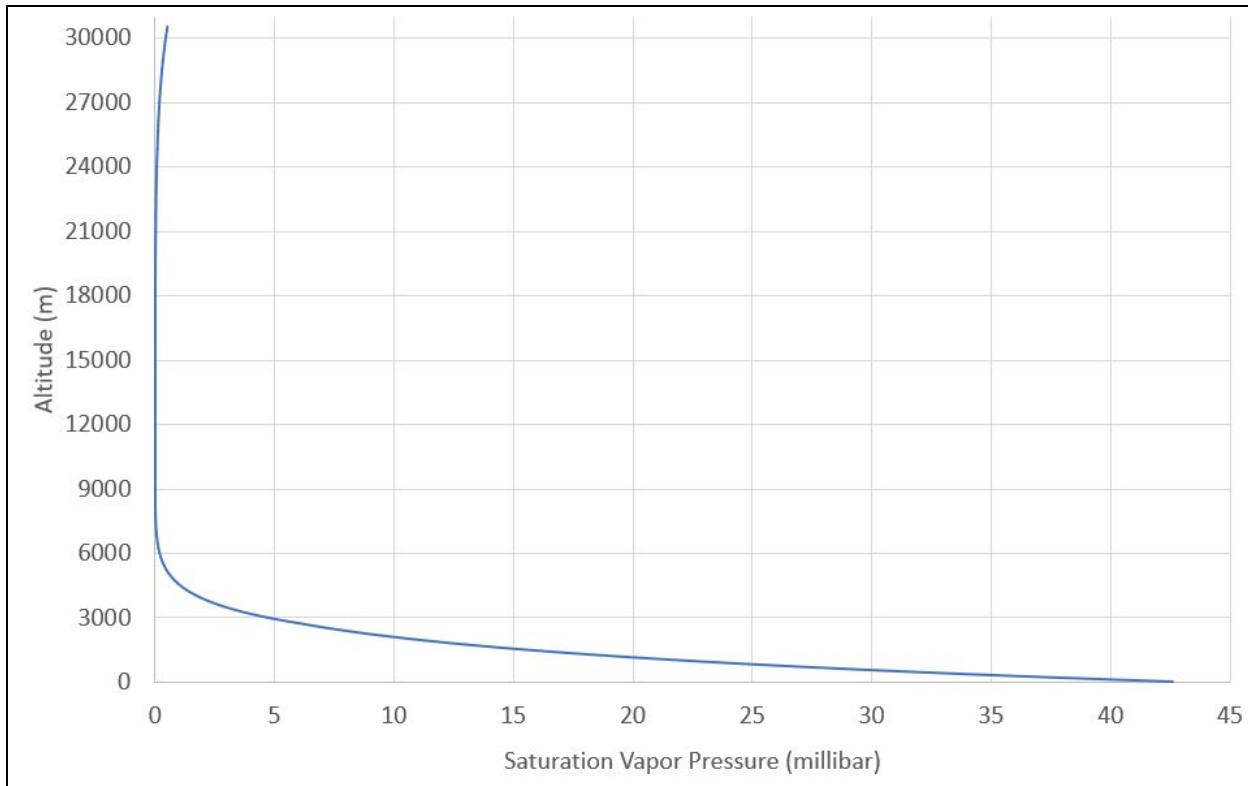


Figure 3: The Expected Saturation Vapor Pressure (in millibars) over the Flight Altitudes Assuming Ground Temperature at 30 °C

The team will aim to have data values collected at a 1/3 Hz sampling rate or data every 3 seconds. This rate should be fast enough to obtain data to achieve the scientific objectives. Based off data from the LaACES launch in May 2019, the average ascent rate at 60,000 ft was 1316 ft/min (6.7 m/s) and the overall target ascent rate was 1000 ft/min (5.1 m/s). The estimated descent rate was reported at 1400 ft/min (7.1 m/s).^[16] However, based on the time of fall (2092 seconds) from 100,000 ft (30,480 m), the average rate of descent was calculated to be 2868 ft/min (14.6 m/s). Therefore, the altitude readings at a 1/3 Hz sampling rate should vary between 15 m and 45 m; however, the initial drop after the balloon burst might exceed the 45-m altitude change between readings for a few seconds. The team is willing to sacrifice these initial descent readings to reduce the amount of recorded data. This data reduction will conserve memory, reduce the excess time spent in post-flight data analysis, and most importantly, decrease the processing demand on the computer used.

3.3.2 Science Requirements

Below is a list of science requirements that must be met for the payload to successfully calculate the saturated vapor pressure and actual vapor pressure:

- External dry bulb temperature readings shall be accurate based on the sensor design (within 5°C and range of -70 to 30°C) with a sampling rate of at least 3 seconds to obtain data for at least 50 m of altitude change. (See the end of Section 3.3.1 for more details on the sampling rate determination).
- Atmospheric pressure readings shall be accurate based on the sensor design (within 5% full scale) with at least a 3-second sampling rate to obtain data for at least 50 m of altitude change.
- Relative humidity readings shall be accurate based on the sensor design (within 10%) with at least a 3-second sampling rate and for a range near 0–100 % RH.
- Altitude measurements shall be accurate within 50 m to compare with the pressure, temperature, and humidity changes during the balloon flight.

3.4 Technical Background and Requirements

The MegaSat payload, originally designed at LSU A & M, will serve as the base design for the DemonSats-1 device. The MegaSat uses the Arduino Mega microcontroller system to interface and collect data from several environmental monitoring sensors and then stores the data on a microSD card for later retrieval. This prototype will not have radio or wireless communication capabilities.

3.4.1 Technical Background

The technical purpose for this project will be to reproduce and refine the MegaSat payload from the perspective of a new LaACES team. The DemonSats team will start the payload design using the circuit supplies, example IDE code, and assembly manuals provided by the Physics & Astronomy Department at LSU. The MegaSat payload system is scheduled to be

released as the updated base model for LaACES in Fall 2020, and the DemonSats team plans to assist LSU with this model's design by reviewing payload building procedures, electronic control testing checks, complete control system programming/debugging, and sensor calibration checks.

The DemonSats team will however attempt to program (in Arduino IDE) the system independently from LSU's example code as an attempt to show flexibility in the MegaSat's capability, especially with the GPS/MicroSD and Accelerometer/Gyroscope modules. The team also plans to modify the payload container to reduce condensation development on the internal sensors and to increase the sensitivity of the pressure, external temperature and humidity sensors by placing them on the outside of the box. Details on the different sensors, the control system design, and the payload container will be discussed in Section 4.0 "Payload Design."

3.4.2 Technical Requirements

Below is a list of technical requirements that must be met for the payload to successfully collect and store data:

- Internal and External dry bulb temperature readings shall be calibrated to be accurate within 5 °C and a range of -70 to 30 °C.
- Atmospheric pressure readings shall be calibrated to be accurate within 5% f. s. and a range of 0 to 1015 absolute millibars.
- Relative humidity readings shall be calibrated to be accurate within 10% for a range of 0-100%.
- Altitude measurements shall be tested to determine its accuracy.
- The payload shall record time-stamped altitude, pressure, temperature, and humidity data throughout flight.
- Payload shall remain below the LaACES constraint of 500 g weight.
- Payload shall conform to LaACES constraints for the flight vehicle interface to allow the payload to attach to the flight string by having two holes spaced 17 cm apart diagonally on the payload box and lid.
- Payload shall show survivability verified through thermal and vacuum tests to confirm that the sensors can perform under the expected flight conditions and that the payload box will protect the contents from the expected flight conditions.
- Payload shall use a lithium battery pack which will keep it operational for the entire flight.
- Payload shall be activated and launched by the LSU team by using easy and simple instructions.
- Payload shall be recovered by the LSU team using positioning data transmitted via the GPS beacon attached to the balloon vehicle.

3.4.3 Technical Goals in Addition to Requirements

Below is a list of technical goals that will be attempted in addition to the requirements to assist LSU in the testing of the MegaSat design:

- GPS latitude and longitude data shall be recorded, and the data analysis procedure and sensor accuracy shall be explored.
- Acceleration data shall be calibrated and recorded in X, Y, and Z directions (selecting ± 4 g), and the data analysis procedure and sensor accuracy shall be explored.
- Gyroscope data shall be recorded as rotation about X, Y, and Z without exceeding the program analog/digital conversion threshold (selecting ± 500 g), and the data analysis procedure and sensor accuracy will be explored.

4.0 Payload Design

4.1 Principle of Operation

The LaACES MegaSat payload serves as the base design for the DemonSats-1 device. The MegaSat uses the Arduino Mega 2560 development board to interface and collect the following data:

- RTC time logging
- internal and external temperature
- atmospheric pressure
- % relative humidity
- GPS – latitude, longitude and altitude
- acceleration – movement in X, Y, and Z
- gyroscope – rotation about X, Y, and Z

In addition to the MegaSat design, the DemonSats-1 device will also use an additional altitude and humidity sensor to ensure the collection of data to achieve the scientific objectives. The data will be stored on a microSD card for later retrieval. This prototype will not have radio or wireless communication capabilities. The prototype will be powered by a 12 V lithium battery pack and protected during flight by a Styrofoam box.

4.2 System Design

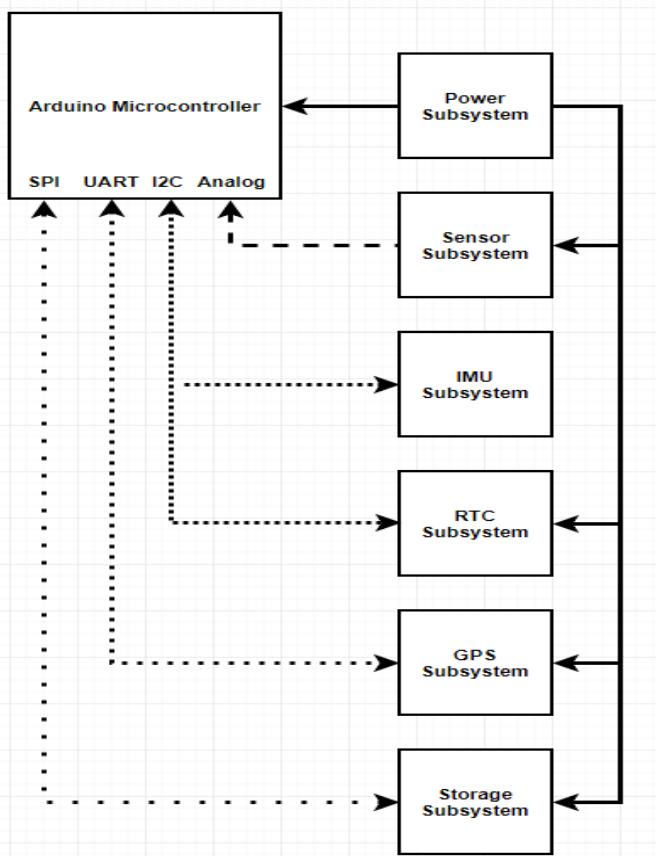


Figure 4: High-level Schematic of MegaSat System Design [8]

4.2.1 Functional Groups

The system design can be divided into 6 major functional groups: the power subsystem, the sensor subsystem, the IMU subsystem, the RTC subsystem, the GPS subsystem, and the storage subsystem (See Figure 4). The controller for the system is the Arduino Mega 2560, which controls the interactions of all the subsystems. Power is supplied by lithium batteries since these lightweight batteries perform well in cold temperatures and low pressures. The sensor subsystem mainly consists of an Arduino Mega shield designed and printed by the LSU LaACES team, which houses the circuitry for the two temperature sensors, pressure sensor, humidity sensor, and the acceleration/gyroscope module (IMU subsystem). To ensure that the scientific requirements will be met, an additional altitude sensor and humidity sensor have been added to the sensor subsystem (See section 4.3 for more details on sensors). The Mega shield board with the sensors also houses a real-time clock (RTC) for accurate datalogger timekeeping. The final two subsystems are both housed on the Adafruit Ultimate GPS logger shield. The GPS transceiver is located on this “off-the-shelf” module, along with a microSD reader/writer unit.

4.2.2 Group Interfaces

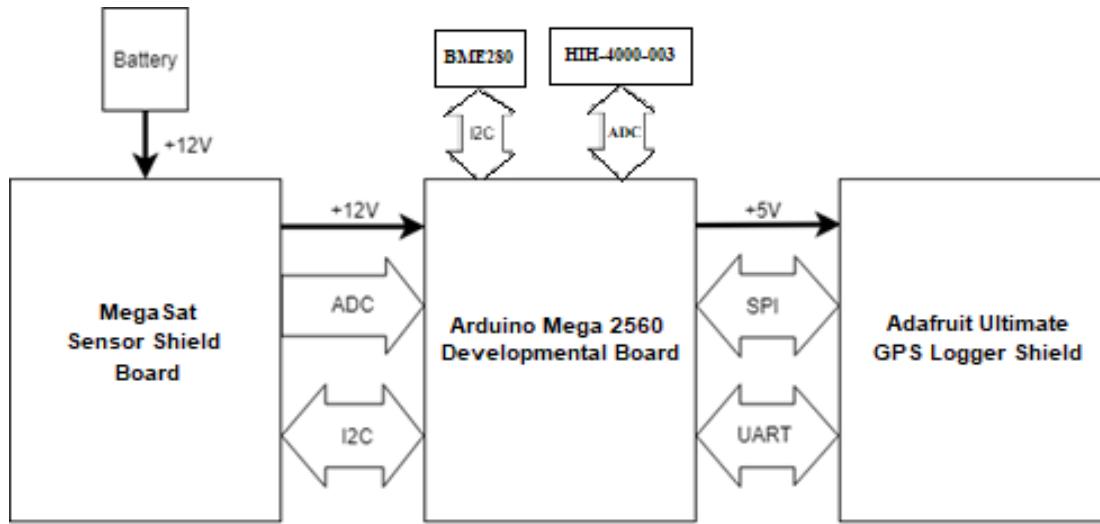


Figure 5: Group Interfaces of the MegaSat [1]

The system can be divided into 5 major group interfaces: power supply, system control (Arduino Mega), sensor shield board, GPS logger shield and added sensors (See Figure 5). Power will be supplied by a 12 V lithium battery pack made from two 2CR5 batteries connected in series. The controller for the system is the Arduino Mega 2560, which controls the interactions of all other functional groups. The Sensor shield board is an Arduino Mega shield designed and printed by the LSU LaACES team. This board houses the power management circuitry, the temperature sensors circuitry, the pressure sensor and its circuitry, a humidity sensor and its circuitry, a real-time clock (RTC), and the Sparkfun MPU-6050 Triple Axis Accelerometer/Gyroscope. The next group interface is the Adafruit Ultimate GPS logger shield, which contains the GPS transceiver and microSD module. The final group interface is the added sensors to assist in achieving the scientific requirements for humidity and altitude. The humidity sensor is an additional HIH-4000-003 connected directly to the analog pins of the Arduino Mega. The altitude sensor is the Sparkfun Atmospheric Sensor Breakout (BME280) which is connected to the Arduino Mega via I2C.

The group interfaces have been stacked as follows: the Arduino Mega 2560 on bottom, then the LSU MegaSat shield board, then the Adafruit Ultimate GPS logger shield, and then the additional sensors. The battery pack has been placed along the side of the electronics within the enclosure so it is accessible prior to launch.

4.2.3 Traceability

Table 2: Traceability of Science Objectives

Objectives	Requirements	Functional Components
Determine the profiles of <u>temperature</u> with altitude.	External dry bulb temperature readings shall be accurate based on sensor design.	Calibration of external temperature sensor in Sensor Subsystem to within 5°C, range of -70 to 30°C and a sampling rate of 3 seconds
Determine the profiles of <u>atmospheric pressure</u> with altitude.	Atmospheric pressure readings shall be accurate based on sensor design.	Calibration of pressure sensor in Sensor Subsystem to within 5% f. s., a range of 0 to 1015 abs. mbar and a sampling rate of 3 seconds
Determine the profiles of <u>% relative humidity</u> with altitude.	Relative humidity readings shall be accurate based on sensor design.	Calibration of humidity sensor in Sensor Subsystem to within 10%, a range of 0% to 100% and a sampling rate of 3 seconds
Determine saturated vapor pressure using external temperature and pressure data with altitude.	Temperature, pressure, and altitude sensors shall be accurate based on sensor design.	Calibration of external temperature and pressure sensor as mentioned above. Properly program GPS for altitude reporting.
Compare % RH readings and the equilibrium vapor pressure with altitude	Altitude measurements must be accurate.	Proper programming of GPS Subsystem and determine if altitude is accurate within 50 m for a range of 0 to 30.5 km

Table 3: Traceability of Technical Objectives

Objectives	Requirements	Functional Components
Construct payload using Arduino Mega control system with temperature, pressure, humidity, altitude, and GPS sensors and a real-time clock.	The payload shall record time-stamped data throughout flight.	Proper installation and electronic verification testing
Ensure payload conforms to LaACES constraints for weight and mechanical design requirements.	Payload shall remain below LaACES maximum weight of 500 g.	Weight budget shall be monitored.
	Payload box shall have two holes spaced 17 cm apart diagonally on the lid to allow attachment of flight strings.	Proper design and implementation of payload box
	Payload shall use a lithium battery pack.	Proper installation of battery pack on payload and verification that the battery shall last for duration of expected flight time
	Payload shall be recovered by LaACES team.	Positioning data transmitted using GPS beacon attached to balloon vehicle upon landing
Calibrate each analog sensor for accuracy in measurement.	Internal and external dry bulb temperature readings shall be calibrated to be accurate.	Calibration of temperature sensors in Sensor Subsystem within 5°C and a range of -70 to 30°C
	Atmospheric pressure readings shall be calibrated to be accurate.	Calibration of pressure sensor in Sensor Subsystem within 5% f. s. and a range of 0 to 1015 absolute millibars
	Relative humidity readings shall be calibrated to be accurate.	Calibrated of humidity sensor in Sensor Subsystem within 10% and a range of 0 to 100%
	Altitude measurements shall be tested to determine its accuracy.	Calibration of GPS subsystem within 50 m and a range of 0 to 30.5 km

Table 3: Traceability of Technical Objectives continued

Verify survivability of payload and functionality of sensors under expected flight conditions.	Sensors shall perform under expected flight conditions.	Verification through thermal and vacuum tests
	Payload box shall protect contents from expected flight conditions.	
	Payload shall withstand conditions in high altitude.	Enclosure box shall be installed.
	Payload shall have enough power supply.	Power budget shall be monitored.
	Payload shall be able to withstand ground impact.	Enclosure box shall be installed.

Table 4: Traceability of Technical Goals

Goals	Functional Components	
GPS latitude and longitude data analysis procedure and sensor accuracy shall be explored.	GPS data including latitude and longitude shall be recorded as clear data.	Translate NMEA codes into latitude and longitude data through proper programming of GPS subsystem
Accelerometer data analysis procedure and sensor accuracy shall be explored.	Acceleration data shall be calibrated in X, Y, and Z directions and recorded.	Proper programming of IMU
		Calibration of MPU-6050 in Sensor Subsystem
Gyroscope data analysis procedure and sensor accuracy shall be explored.	Gyroscope data shall be calibrated in X, Y, and Z directions and recorded.	Proper programming of IMU
		Calibration of MPU-6050 in Sensor Subsystem

4.3 Electrical Design

4.3.1 Sensors

Table 5: List of Sensors Used in the DemonSats-1 Payload

Sensor	Part Number	Signal Interface	Power Supply	Operating Temperature (°C)
Real-time Clock ^[9]	DS3231 RTC	I2C	3.3 V	-40 to 85
Temperature ^[10]	1N457	Analog	5 V	-223 to 126
Atmospheric Pressure ^[12]	1230-015A-3L	Analog	5 V	-40 to 125
% Relative Humidity ^[13]	HIH-4000-003	Analog	5 V	-40 to 85
GPS ^[15]	FGPMMOPA6H	UART	3.3 V	-40 to 85

Table 5: List of Sensors Used in the DemonSats-1 Payload cont...

Acceleration [16]	MPU-6050	I2C	3.3 V	-40 to 85
Gyroscope [16]	MPU-6050	I2C	3.3 V	-40 to 85
Atmospheric [18] Sensor Breakout	BME280	I2C	3.3 V	-40 to 85

[#] References

4.3.1.1 Real-Time Clock (RTC) – The DS3231

The DS3231 is touted as an extremely accurate, I2C-integrated real-time clock (RTC), which has an integrated temperature-compensated 32 kHz oscillator and crystal to keep the elapsed time. The RTC maintains seconds, minutes, hours, day, date, month, and year information, with leap-year compensation. Data from the module is sent via I2C to the Arduino Mega. The RTC subsystem also has a battery backup to extend the timekeeping accuracy as the main power supply potential decreases.

According to the DS3231 datasheet, the accuracy of the RTC is ± 2 minutes per year in temperatures between -40°C to +85°C.^[9] Since the sensor is interfaced via I2C, the supply voltage will need to be 3.3 V and less than 200 microAmps current. The datasheet also reminds the team that quick and low-temp (260 °C) soldering will be necessary in order to avoid damaging this semiconductor module. Preflight testing has been done to verify that the RTC module functions effectively as a timekeeper.

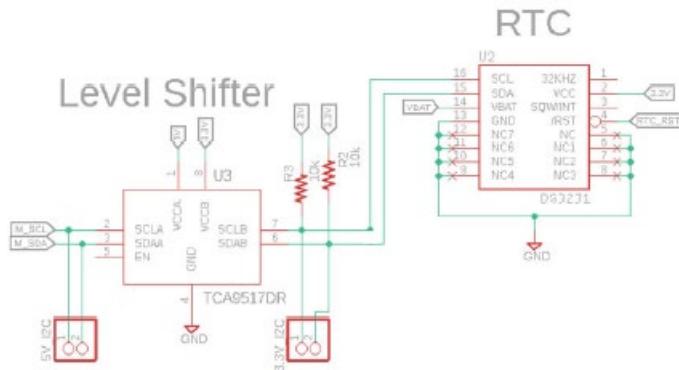


Figure 6: Schematic of RTC Circuit of MegaSat [11]

The RTC (slave) is connected to the Arduino Mega 2560 controller (master) through inter-integrated circuit (I2C) communication. The I2C serial protocol is designed to have up to 10 sensors connected to the Arduino Mega through two “open-drain” lines called the Serial Clock (SCL) and Serial Data (SDA). In the MegaSat, we have 2 sensors attached to this I2C bus, the RTC and the MPU-6050 accelerometer/gyroscope module. The circuit above shows that the SCL and SDA lines are pulled up with 10 kOhm resistors connected to the 3.3 V supply. These pull-up resistors allow for the I2C logic to be transferred bidirectionally to the master and slave(s). A level shifter (U3) has been incorporated into the circuit design to enable communication between the Arduino Mega that uses 5 V logic and the I2C sensors that use 3.3 V logic. In other words, the level shifter changes the voltage amplitude of the signal bidirectionally

so that the external master and slaves can “talk” to each other. In the Arduino programming, each sensor and its respective sensor components have a hexadecimal address. Therefore, a sensor signal can be called into program reading or writing by listing its address. The DS3231 sensor address is 0x68, the reading address is 0xD1, and the writing address is 0xD2.

4.3.1.2 Internal and External Temperatures – The 1N457 Diode

The function of the internal and external temperature sensors used in the MegaSat is to provide temperature readings over a wide range of temperatures (-70 to 30 °C). The forward biased 1N457 diode can provide linear changes of voltage versus temperature between the ranges of 50 to 400 K. For the 1N457 diode, the temperature sensitivity is approximately -2.5 mV per Kelvin or degrees Celsius.^[10]

The internal and external temperature sensor circuits are each composed of a LM234DT current source, AD820ARZ operational amplifier, a 3.9 kΩ resistor, a 20 kΩ resistor, a 180 kΩ resistor, a 68 Ω resistor, a 1 kΩ potentiometer, a 100 kΩ potentiometer, three 0.1 μF ceramic capacitors, a 10 μF electrolytic capacitor, and a 1N457 diode (See Figure 7).^[11]

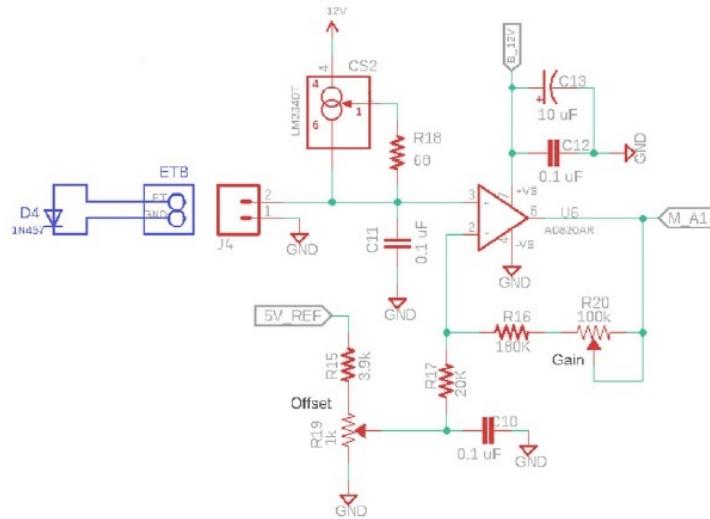


Figure 7: Schematic of 1N457 Diode Temperature Sensor Circuit of MegaSat^[11]

The temperature sensor circuit works by utilizing a FET-input operational amplifier, AD820 (U6), in a non-inverting configuration to sum the inputs of the forward voltage of the 1N457 diode with a DC offset voltage. The LM234DT IC chip, CS2, set to current source configuration, shown in Figure 7, is used to provide a constant 1 mA current bias to the diode. This set current (I_{set}) can be calculated using the formula:

$$I_{set} = \frac{227 \mu V_K * (300K)}{R_{12}}, \text{ where } R_{12} = 68\Omega \quad (\text{Eqn. 6})$$

At this current, the forward voltage of the diode changes by -2.5 mV per Kelvin and provides an input signal for the positive input of U6. The gain needed will be approximately 15 as the

temperature will range from -80 to 30°C during the flight. This corresponds to a forward voltage of 240 mV to 300 mV. The voltage gain can be calculated using the formula:

$$Av = 1 + \frac{R_{16} + R_{20}}{R_{17}} \quad (Eqn. 7)$$

where R_{20} can be adjusted for a gain of 9 to 14. The DC offset, provided to the negative input of U6, will be adjusted by tuning R_{19} so that the signal will be in the acceptable range (0 to 5 V) for the Arduino Mega's ADC input. RC lowpass filters are provided by RC networks C_{11} , R_{18} and C_{10} , R_{17} . The signal received at the Analog input of the Arduino Mega is then conditioned with a program to return a value for temperature with respect to a change in voltage.

The specifications for the LM234DT are as follows: the operating temperature range is -25 to 100 °C and the thermal effect on current will cause the current to change by 0.132% per degree Celsius increase in temperature. The specifications for the AD820ARZ operational amplifier are as follows: the operating temperature range is -40 to 85°C. The specifications for all chip resistors are as follows: the operating temperature range is -55 to 155 °C and the resistors operate for 1000 hours at 40°C/95% relative humidity. The specifications for 0.1 µF ceramic capacitors are as follows: the operating temperature range is -55 to 125°C, the capacitors can operate at 85°C/85% relative humidity, and capacitors can withstand 5Gs of force for 20 minutes. The 10 µF electrolytic capacitor specifications are as follows: the operating temperature range is -40 to 85°C. The effect of environmental factors such as temperature and humidity on potentiometers will change the resistance by ±1%/±2%. [10]

The temperature sensor circuits have been tested, and an analog/digital calibration curve has been developed for later programming use. The sensors have been tested at a variety of temperatures in dry ice, in salted ice water, in iced water, in a dry ice cooler, at room temperature, and in warm water. During the testing, the temperature sensor readings have been compared to a calibrated Type-K thermocouple.

4.3.1.3 Atmospheric Pressure – The 1230-015A-3L Piezoresistive Diaphragm Gauge

The 1230 pressure sensor designed by TE Connectivity is a piezoresistive silicon pressure sensor packaged with dual line configured wiring. The pressure sensor uses a piezoresistive strain gauge in a Wheatstone bridge configuration to produce a differential output voltage of 1 to 100 mV for a change in internal strain caused by differences in pressure. This sensor is an absolute pressure diaphragm transducer that measures in a range from 0 -15 psi (0 – 1034 mbar). [12]

Temperature compensation is provided between the range of -20 to +85 °C using laser trimmed resistors. An additional laser trimmed resistor is included to help normalize pressure sensitivity variation by controlling the gain of an external differential amplifier. The resulting sensor sensitivity is between ±1% of the pressure range (0.15 psi or 10.34 mbar). The effective operating temperature range is from -40 to +125 °C while the storage temperature is from -50 to +150°C. The supply current is 0.5 mA to 2.0 mA

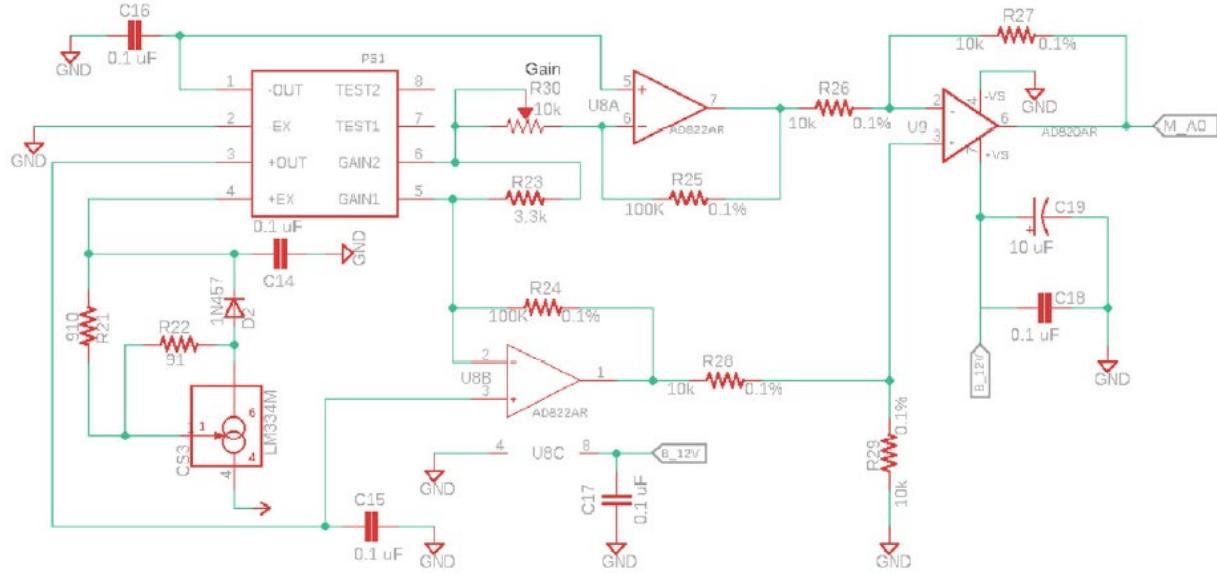


Figure 8: Schematic of Pressure Sensor Circuit of MegaSat [11]

In the pressure module circuit shown in Figure 8, a constant current supply is supplied at 1.5 mA by CS3 in a zero-temperature coefficient current source configuration. Using Ohm's law:

$$R1 = \frac{0.134V}{1.5mA} = 89.3\Omega \quad (\text{Eqn. 8})$$

and $R2 = 10R1 = 893\Omega$, the resistors 91Ω and 910Ω were chosen to provide a 1.5 mA current source. A dual FET-input op-amp (U8) forms an inverting difference amplifier with a gain of 30. During normal operation, an output voltage of 0 to 3.12 V over a pressure range 0 to 15 psi is provided when $R24$ and $R25$ are $100\text{k}\Omega$. $R30$ can be tuned in order to provide a more precise control of gain given by the formulae:

$$R_{eff} = R30 + \frac{R_{int}*R23}{R_{int}+R23} \text{ and } Av = \frac{R24}{R_{eff}} \quad (\text{Eqn. 9})$$

where $R24$ and $R25$ equal and R_{int} = internal resistance of pressure sensor. The final amplifier stage (U9) is another FET op-amp to provide an output voltage with reference to ground in order to condition the signal for the Arduino's analog input.

The sensor output has been observed to be linear, and a calibration curve (See Section 7) has been developed to convert the sensor's analog signal into pressure readings in millibars which shall be stored on the microSD card.

4.3.1.4 Humidity Sensor – The HIH-4000-003 Relative Humidity Sensor

The HIH-4000-003 humidity sensor by Honeywell measures the percentage of ambient relative humidity by producing an output voltage that has a linear relationship to the relative humidity. Relative humidity is a measurable value that describes the amount of water vapor in the air. Several gases make up the Earth's atmosphere such as Nitrogen, Oxygen, Argon, and Carbon Dioxide. Water vapor can also be found in trace amounts. The sum of all the partial pressures of each gas makes up the atmospheric pressure. The HIH-4000-003 humidity sensor measures the partial pressure of water vapor as a ratio to the equilibration (saturation) vapor pressure.

According to its datasheet,^[13] the sensor will output a voltage that will change linearly as a function of % Relative Humidity according to the following equation when measured at 25°C and supplied with 5 Volts (See Figure 9):

$$V=0.05*x+(0.75) \text{ *%RH} \quad (\text{Eqn. 6})$$

Specifications for the HIH-4000-003 humidity sensor include operating temperatures ranging from -40 to 85°C and an operating humidity ranging from 0% to 100% RH. The power supply recommended should be between 4 to 5.8 Volts and 200 to 500 microAmps.

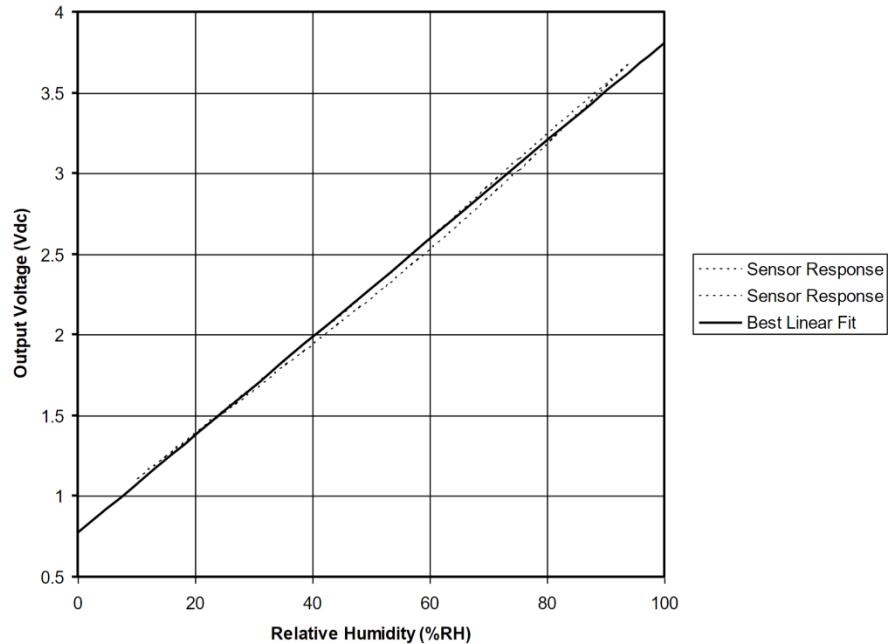


Figure 9: Expected Output voltage vs. %Relative Humidity ^[13]

This sensor is best calibrated at 5 VDC and at 25°C. Therefore, there is a temperature (in degrees Celsius) compensation equation for sensor readings:

$$\text{True RH} = \frac{(\text{Sensor RH})}{1.0546 - 0.00216(T)} \quad (\text{Eqn. 10})$$

Measured values in %RH are expected to increase as the payload rises to the troposphere especially if clouds are present. This increase is due to the water vapor content in this level of the atmosphere. Once the payload rises out of the troposphere, a decrease in %RH is expected to occur. This decrease is due to the decrease in air density. If water vapor is present in the stratosphere, it is likely to be dispersed and not as frequently detected. A graph of %RH can then be plotted as a function of time to determine the position of the payload relative to altitude.

The circuit design for the humidity sensor on the MegaSat shield, as shown in Figure 10, includes various components used to optimize the humidity sensing module by reducing noise. The sensor requires a steady supply of 5 volts to function accurately. This is provided by V3. Both components labeled U4 represent the AD822 op amp. This type of op amp is used to provide a low offset and low noise. Resistors R5, R6, R7, and R8 are used to determine the voltage gain of this stage. The two variable resistors within the system, R7 (offset) and R8 (gain), can be fine-tuned for precision. The output of U4 is connected to the analog channel A3 of the Arduino Mega microcontroller.

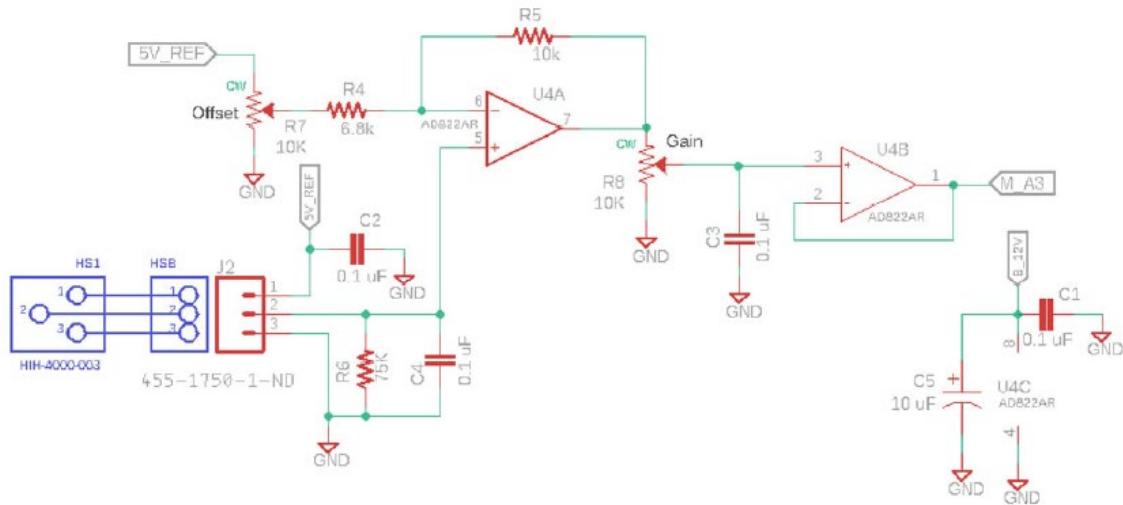


Figure 10: Schematic of Humidity Sensor Circuit of MegaSat [11]

The humidity sensor can also be directly wired to the analog pin. A second HIH-4000-003 has been connected to the Arduino Mega 2560 directly through the analog A4 pin. This second humidity sensor has been included in the DemonSats-1 design to ensure that the team will receive data from the full range of 0-100% relative humidity. Preliminary testing showed that the incorporation of the active filter (shown in Figure 10) on the MegaSat shield causes the full range of humidity reading to be limited to 80% within the 0-100% range when set to the highest gain setting.

The humidity sensors have been tested, and analog/digital calibration curves have been developed for programming use. Salt-saturated chambers of different %RH levels were used as calibration set points, and a factory calibrated %RH sensor was also used in parallel during the testing (See Table 6).

Table 6: List of Saturated Salts Used in Humidity Sensor Calibration [14]

Saturated-salt solution chemical	Expected %RH of air inside chamber (at 30°C)
Lithium Chloride (LiCl)	11.30% ± 0.35
Magnesium Chloride (MgCl ₂)	32.44% ± 0.14
Sodium Nitrite (NaNO ₂)	73.14% ± 0.31
Sodium Chloride (NaCl)	75.09% ± 0.11
Potassium Sulfate (K ₂ SO ₄)	97.30% ± 0.45

4.3.1.5 GPS

The FGPMMPA6H GPS module is packaged on the Adafruit Ultimate GPS Logger Shield. On the GPS module is the GPS Chipset MT3339 by MediaTek. This chipset or transceiver is touted to achieve “the industry’s highest level of sensitivity (-165 dBm) and instant Time-to-First Fix (TTFF) with the lowest power consumption (66 to 82 mW) for precise GPS signal processing.”^[16] This chipset is also expected to work well in low receptive, high velocity conditions.

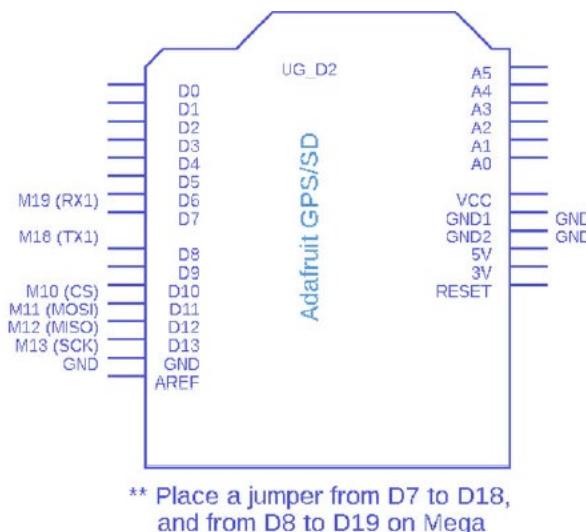


Figure 11: Pinout of Ultimate GPS Module to Interface with MegaSat [11]

During operation, the GPS module sends out National Marine Electronics Association (NMEA) data. Parsing out the NMEA codes provided from this GPS module has been explored during the programming of the payload. NMEA codes contain a multitude of data points. The data from these NMEA codes can be used to find where the payload was during the flight and where the payload landed after the flight. The default NMEA code type is GPRMC (Global Positioning Recommended Minimum Coordinates), but the GPGGA (system fix data) type is needed for our team to collect altitude data. So both format types will be collected as the RMCGGA to ensure that the GPS time stamp and 2D location coordinates (latitude and longitude) are included throughout flight. GPRMC and GPGGA formats display the data in different order, so the different NMEA sentences will be separated post flight using the Microsoft Excel data sorting function. An example of the RMC and GGA formats are as follows:

\$GPRMC, 123519, A, 4807.038, N, 01131.000, E, 022.4,084.4, 230394,003.1, W*6A
where,

123519 = the time stamp (HH:MM:SS)

A = active status (or V = void)

4807.038, N = latitude (DDMM.MMMMM)

01131.000, E = longitude (DDMM.MMMMM)

022.4 = Speed over ground (knots)

084.4 = Track angle

230394 = Date (DD/MM/YY)

003.1, W = Magnetic variation (degrees)

*6A = Checksum data

\$GPGGA, 200642, 3145.0878, N, 09305.8043, W, 1, 04, 1.85, 60.5, M, -25.7, M, *6A
where,

200642.000 = the time stamp (HH:MM:SS)

3145.0878, N = latitude (DDMM.MMMMM)

09305.8043, W = longitude (DDMM.MMMMM)

1 = Quality Indicator (1-uncorrected to 5-decimeter precision)

04 = number of satellites used

1.85 = horizontal dilution of precision

60.5, M = altitude in meters

-25.7, M = geoidal separation in meters

*6A = checksum [16]

The team will need to report the altitude of the payload from the GPS measurement to meet its scientific objectives. However, the team plans to have the payload collect the full NMEA sentence so that other information (e.g. the number of satellites in contact with the transceiver, latitude and longitude) are available for further observation and analysis. The GPS transceiver has been tested prior to flight by monitoring automobile ground travel from Natchitoches to Shreveport.

4.3.1.6 Sparkfun MPU-6050 Acceleration / Gyroscope

The MPU-6050 Triple Axis Accelerometer and Gyro Breakout by Sparkfun is a 3-axis accelerometer and 3-axis gyroscope combined into one module. The module has an onboard Digital Motion Processor (DMP) that is capable of processing 9-axis “MotionFusion” algorithms.^[17] The Sparkfun module communicates via I2C with the Arduino Mega 2560.

Level Shifter

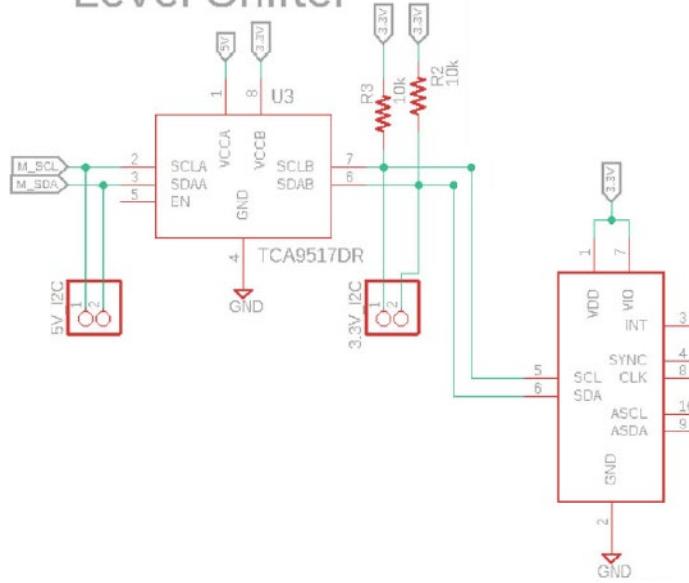


Figure 12: Schematic of MPU-6050 Module Interface with MegaSat [11]

The MPU-6050 (slave) is connected to the Arduino Mega 2560 controller (master) through inter-integrated circuit (I2C) communication. The I2C serial protocol is designed to have up to 10 sensors connected to the Arduino Mega through two “open-drain” lines called the Serial Clock (SCL) and Serial Data (SDA). In the DemonSats-1 payload, we have 3 sensors attached to this I2C bus, the RTC, the MPU-6050 accelerometer/gyroscope module and the BME280 Atmospheric Breakout. The circuit above shows how the MegaSat I2C components, including the MPU6050, are connected to the Arduino Mega. In Figure 12, the SCL and SDA lines are pulled up with 10 kOhm resistors connected to the 3.3 V supply. These pull-up resistors allow for the I2C logic to be transferred bidirectionally to the master and slave(s). A level shifter (U3) has been incorporated into the circuit design to enable communication between the Arduino Mega, which uses 5 V logic, and the I2C sensors that use 3.3 V logic. In other words, the level shifter changes the voltage amplitude of the signal bidirectionally so that the external master and slaves can “talk” to each other. In the Arduino programming, each sensor and its respective sensor components have a hexadecimal address. Therefore, a sensor signal can be called into a program read or write by listing its address. The MPU-6050 module address is 0x69 and there are many addresses for the different data requests, which can be found in the MPU6050.h library. Table 7 shows the list of the data addresses we have used in the programming.

Table 7: I2C Register Map for Data Used from MPU-6050

Data to Be Requested from MPU-6050	I2C Address
X acceleration	0x3B and 0x3C
Y acceleration	0x3D and 0x3E
Z acceleration	0x3F and 0x40
Gyroscope about X	0x43 and 0x44
Gyroscope about X	0x45 and 0x46
Gyroscope about X	0x47 and 0x48
Chip Temperature	0x41 and 0x42

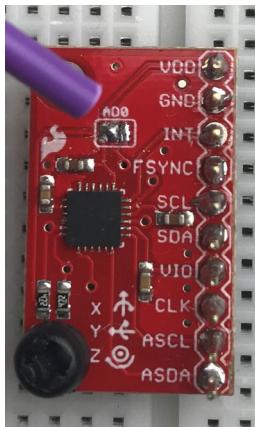


Figure 13: MPU6050 with AD0 pin solder to high

It is important to note that since the 0x69 address is to be used for the MPU6050, the solder on the AD0 Jumper on the breakout board must be changed to set the AD0 pin high (+5V) instead of the default low (0V) for 0x68. The team had some difficulty discovering this solution to the payload's I2C communication issues until they finally realized that the instructions to do this were found in the MegaSat Assembly Manual, Appendix E.

The Sparkfun MPU-6050 has been tested before flight for acceleration reaction rates in the X, Y and Z while resting on a level desk aligned with the three axial directions so that only one of the directions was subjected to 1 g at a time.

4.3.1.7 Sparkfun Atmospheric Sensor Breakout – BME280

The BME280 Atmospheric Sensor Breakout board from Sparkfun uses the Bosch BME280 sensor, which is a small profile barometric pressure, humidity, and temperature sensor. The BME280 sensor can be used to determine altitude between 0 and 9200 m with a relative 2-m accuracy by comparing the measured pressure to the expected pressure values at different elevations.^[18] This sensor was added to the DemonSats-1 to report altitude data within 9200 m at times when the GPS is unable to obtain the 3D measurements from satellite fixes and the team does not expect the sensor to report altitude correctly as the barometric pressure nears zero.

The BME280 sensor data for temperature and humidity are recorded on the DemonSats-1 payload to serve as internal sensors inside the payload housing. The humidity sensor is reported to have an extremely fast response time (1 s) and high overall accuracy $\pm 3\%$ over the wide temperature range. The temperature sensor is included in the BME280 for temperature compensation for the humidity and pressure sensors and is reported to be optimized for low noise with high resolution with -40 deg C to +85 deg C.^[18]

The breakout board is designed to communicate via SPI or I2C. In the DemonSats-1 payload, I2C communication (with an address of 0x77) has been utilized to interface with this sensor since I2C handles multiple sensors better than SPI and the microSD module is already utilizing the SPI interface. The following table is the registry map for I2C communication, and the data to be requested from BME280 are located from 0xFE to 0xF7:

Table 8: I2C Register Map for Data Used from BME 280 [18]

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state										
hum_lsb	0xFE	hum_lsb<7:0>																		
hum_msb	0xFD	hum_msb<7:0>																		
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00										
temp_lsb	0xFB	temp_lsb<7:0>								0x00										
temp_msb	0xFA	temp_msb<7:0>																		
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00										
press_lsb	0xF8	press_lsb<7:0>																		
press_msb	0xF7	press_msb<7:0>																		
config	0xF5	t_sb[2:0]		filter[2:0]		spi3w_en[0]														
ctrl_meas	0xF4	osrs_l[2:0]		osrs_p[2:0]		mode[1:0]														
status	0xF3					measuring[0]		im_update[0]												
ctrl_hum	0xF2																			
calib26..calib41	0xE1...0xF0	calibration data																		
reset	0xE0	reset[7:0]																		
id	0xD0	chip_id[7:0]																		
calib00..calib25	0x88...0xA1	calibration data																		

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:	do not change	read only	read / write	read only	read only	read only	write only

The BME280 Atmospheric Sensor Breakout board also includes some adjustable sampling modes and filters to tailor its use in different environments. Since the team plans to utilize the BME280 primarily for altitude, the following settings have been selected and included in the Arduino programming. Using the following settings, the expected performance is also included in Table 9. This additional current consumption (0.633 mA) should not hinder the overall power budget of the DemonSats-1 payload.

Table 9: Recommended Settings of BME280 for Highest Accuracy in Altitude [18]

Sensor mode	normal mode, $t_{\text{standby}} = 0.5 \text{ ms}$
Oversampling settings	pressure $\times 16$, temperature $\times 2$, humidity $\times 1$
IIR filter settings	filter coefficient 16
Performance for suggested settings	
Current consumption	633 μA
RMS Noise	0.2 Pa / 1.7 cm
Data output rate	25Hz
Filter bandwidth	0.53 Hz
Response time (75%)	0.9 s

4.3.2 Sensor Interfacing

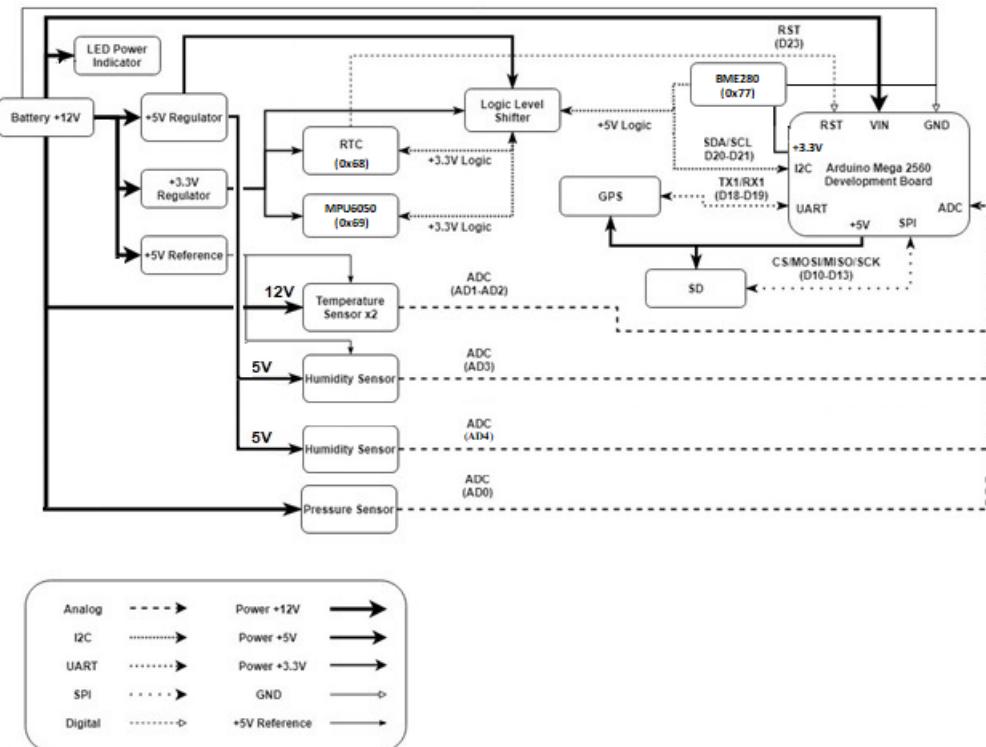


Figure 14: Mid-level block diagram of the DemonSats-1 MegaSat [1]

4.3.3 Control Electronics

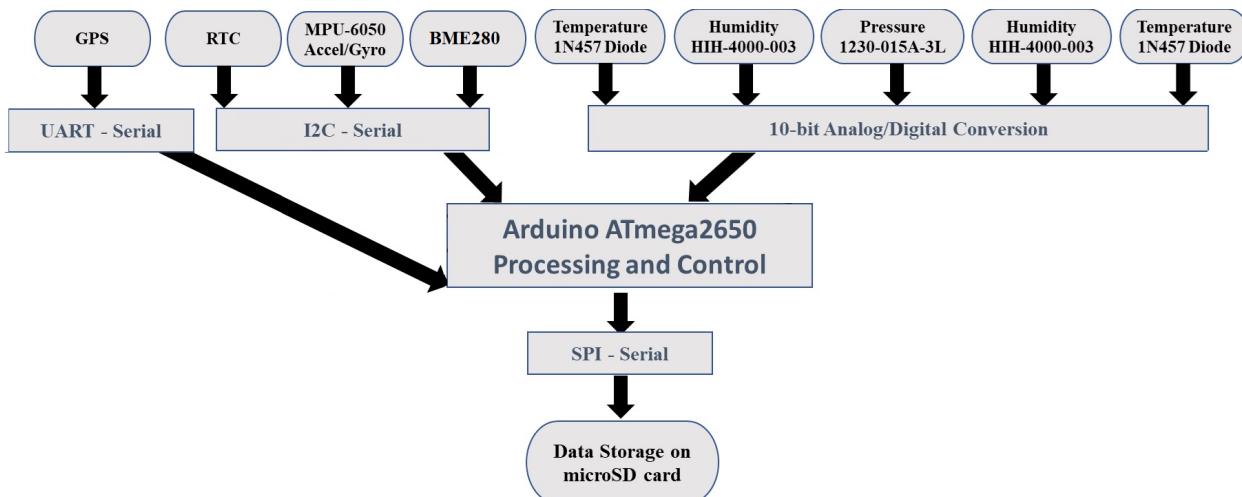


Figure 15: The Pathway from Sensor Input to Data Storage

Arduino Mega

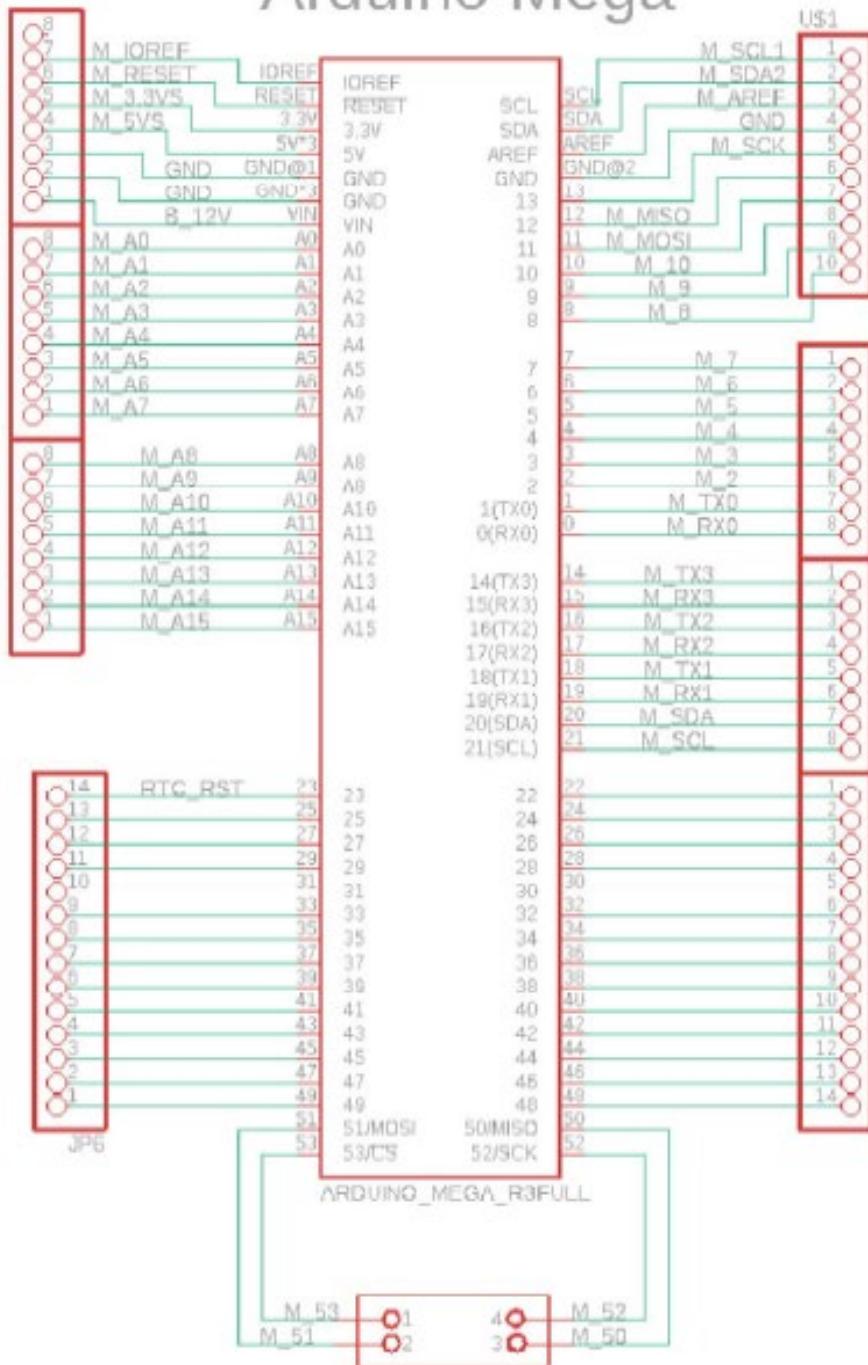


Figure 16: Pinout of Arduino Mega 2560 Interface with MegaSat [11]

4.3.4 Power Supply

The main power of the payload is supplied by lithium batteries because they perform well in cold temperatures and low pressures, some conditions to be expected during flight, and are lightweight. A 12 V lithium battery pack has been made from two 2CR5 batteries connected in series with overheating protection by a Positive Temperature Coefficient (PTC) thermistor (See Figure 17). The battery pack is connected directly to the MegaSat Sensor Shield. On the MegaSat shield, the 12 V main supply is split into 5 V and 3.3 V using voltage regulators (See Figure 18).



Figure 17: Battery pack made of two 2CR5 lithium batteries attached in series with a PTC thermistor (Battery pack is wrapped vertically with electric tape to secure solder joints and wiring.)

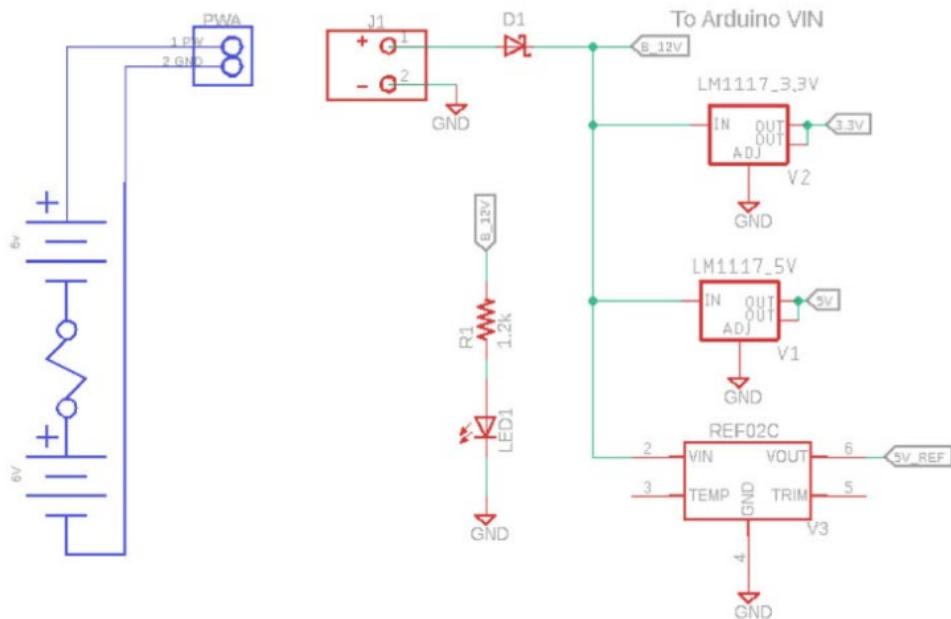


Figure 18: Schematic of Power Supply Circuit for MegaSat [11]

4.3.5 Power Budget

Table 10: The Power Budget for DemonSats-1 Payload

Component	Voltage (V)	Current (mA)	Power (mW)
Arduino Mega 2560	12	200	2400
GPS Logger Shield	5	20	100
MPU-6050 Accel/Gyro/DMP	3.3	3.9	12.9
Temperature Circuit (x 2)	12	2	24
Pressure Sensor	12	1.5	18
Humidity Sensor (x 2)	10	0.2	2
RTC module	3.3	0.2	0.7
BME280 Atmospheric Sensor	3.3	0.633	2
Total Power Consumption (mW):			2559.6
Battery Pack (2CR5 x 2)	Voltage (V)	Charge (mAh)	Energy (mWh)
	12	3000	36,000
Expected Lifetime at Room Temperature (hours)			14 hours
Expected Lifetime – De-rated for Subzero Temperatures (hours)			4.2 hours (30%)

During testing, the team discovered that the lithium battery packs only last about 4 - 5 hours. The MegaSat shield analog sensors do not operate effectively once the battery pack reaches 10 V. Therefore, the lithium battery only effectively delivers about 12 Watt-hours to the payload, which equates to about 4.7 hours. In future DemonSats payloads, operating the analog sensors with filters on a 9 V range would improve the power usage from the batteries.

The team also discovered that the power regulator chip directly on the Arduino Mega 2560 gets very warm (70-80°C). Since the Arduino Mega should be able to receive 6-20V and the recommended range from Arduino is 7 to 12V, the excessive heating was a concern for the team. In future DemonSats payloads, the current supply delivered from the MegaSat shield to the Arduino may need to be determined and reduced.

4.4 Software Design

The MegaSat payload system has been programmed using Arduino IDE.

4.4.1 Data Format & Storage

The team aims to have data values collected at a 1/3 Hz sampling rate. This data rate should be fast enough to obtain data to achieve the scientific objectives. Based off data from the LaACES launch in May 2019, the average ascent rate at 60,000 ft was 1316 ft/min (6.7 m/s) and the overall target ascent rate was 1000 ft/min (5.1 m/s). The estimated descent rate was reported at 1400 ft/min (7.1 m/s).^[16] However, based on the time of fall (2092 seconds) from 100,000 ft (30,480 m), the average rate of descent was calculated to be 2868 ft/min (14.6 m/s). Therefore, the altitude readings at a 1/3 Hz sampling rate should vary between 15 m and 45 m; however, the initial drop after the balloon burst might exceed the 45-m altitude change between readings for a few seconds. The team is willing to sacrifice these initial descent readings to have a consistent sampling rate of recorded data for programming ease. The 3-second per sample rate will conserve memory, reduce the excess time spent in post-flight data analysis, and most importantly, decrease the processing demand on the computer used.

The data is organized as a comma-delimited text string which is stored on the microSD card. Each text string is expected to have 216 or 219 characters. At a 1/3 Hz sampling rate and estimating the payload flight to last approximately 2 hours, the microSD card will be expected to store 7200 lines of data. Therefore, the data should be about 1.6 MB, which is far below the 32 GB storage capacity of the microSD card.

Table 11: Description of Output Data Characters to be Stored to MicroSD Card

Data Type	GPS Date	GPS Time	Altitude from GPS	Altitude from BME	Pressure from MegaSat	Pressure from BME	Ext Temp	Int Temp
Output Format	DD/MM/YYYY, ,	HH:MM : SS, ,	DDDD D, ,	DDDD D.D, ,	DDDD. DD, ,	DDDD. DD, ,	DDD. D, ,	DDD. D, ,
# of char	11	9	6	8	8	8	6	6

BME Temp	Humid1	Humid 2	Humid BME	RTC Time	Accel X	Accel Y	Accel Z	Gyro X
DDD.D ,	DDD.DD ,	DDD.D ,	DDD, ,	HH:MM : SS, ,	DD.DD ,	DD.DD ,	DD.DD ,	DDDD, ,
6	7	6	4	9	6	6	6	5

Gyro Y	Gyro Z	Pressure decimal	Ext Temp decimal	Int Temp decimal	Humid1 decimal	Humid2 decimal	GPS NMEA string
DDDD, 5	DDDD, 5	DDD, 4	DDD, 4	DDD, 4	DDD, 4	DDD, 4	GGA or RMC 72 or 69

After the payload's flight and retrieval, the team will copy the data stored on the microSD card onto a laptop with Microsoft Excel. The data will then be separated into individual columns in Microsoft Excel with the comma as the delimiter. This data separation will yield 36 columns of data in the worksheet. The data will then be further parsed into different worksheets, and separate graphs of external temperature, pressure, and humidity vs. altitude will be made. Calculations and a graph will be made after determining the vapor saturation pressure vs. altitude using equations 4 and 5 of this report. Also, if the relative humidity data is accurate, as verified through comparison to the National Weather Service data as discussed in section 3.3.1, a curve of actual vapor pressure will be reported. The team plans to utilize Microsoft VB macros in Excel to speed up the data analysis process.

4.4.2 Flight Software

Table 12: Flight Software Setup Plan for DemonSats-1 Payload

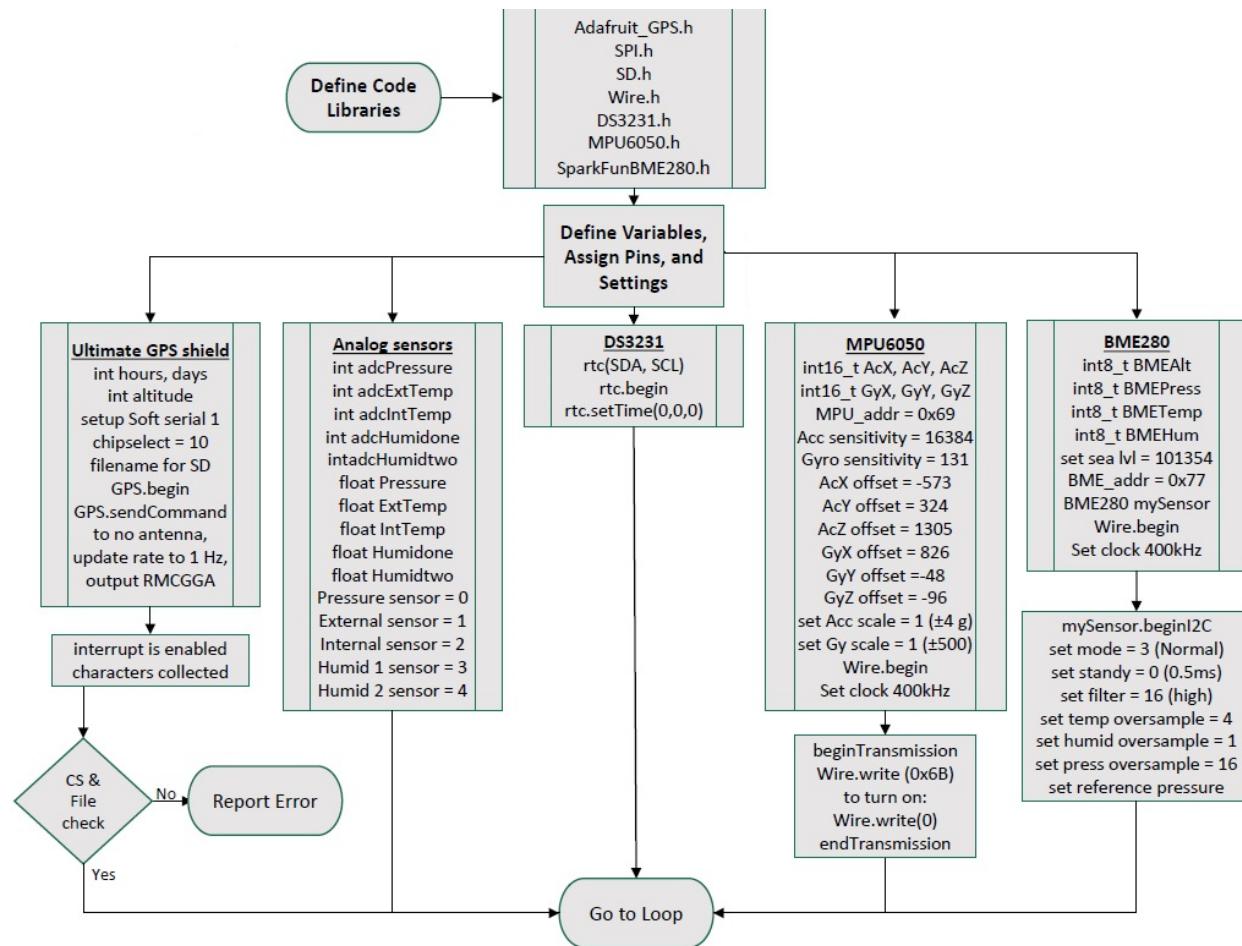


Table 13: Flight Software Main Loop Plan for DemonSats-1 Payload

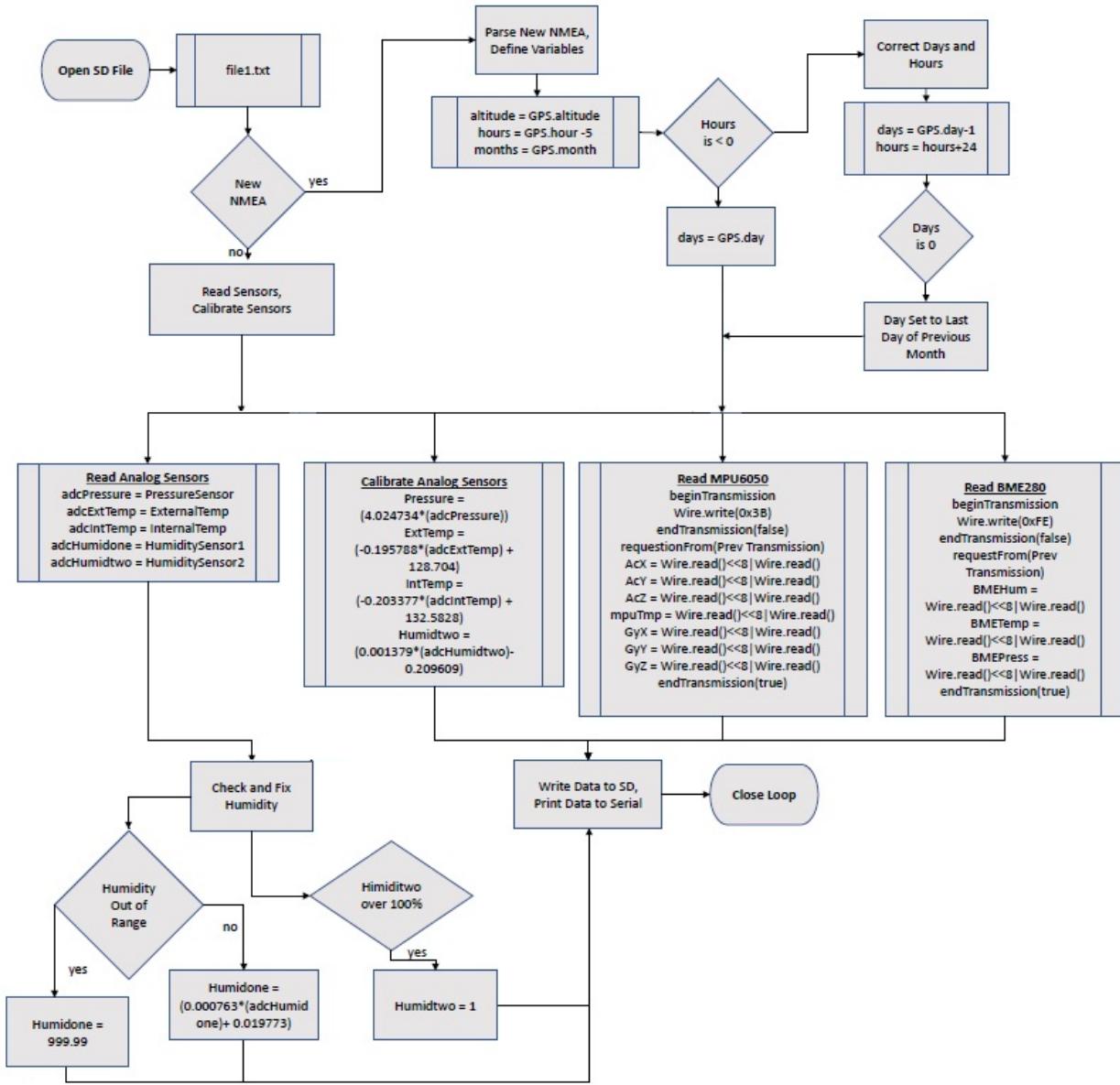
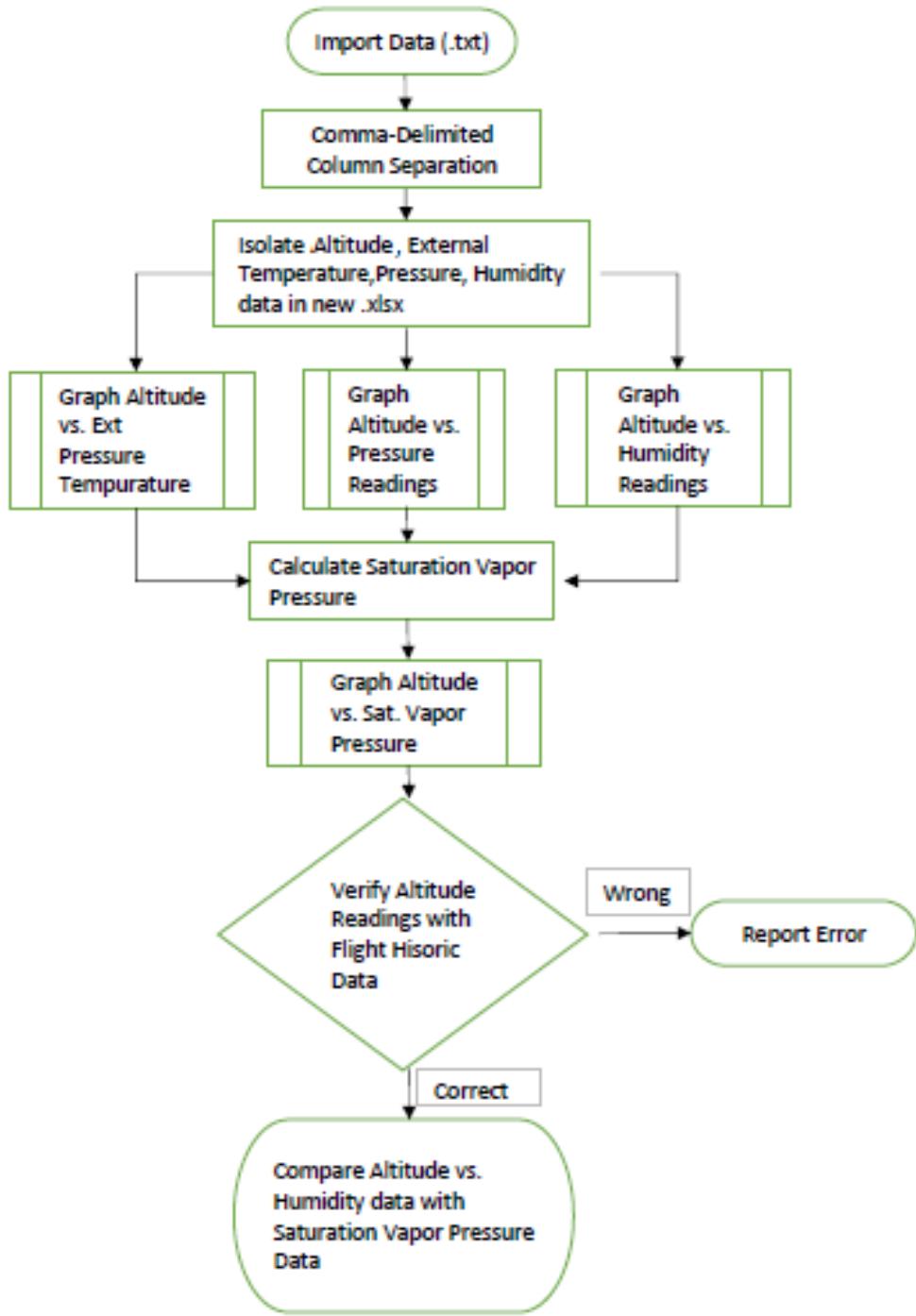


Table 14: Post-Flight Software Plan for DemonSats-1 Payload



4.5 Thermal Design

The thermal environment that the payload will be subjected to during flight has been reviewed in Figure 2 in section 3.3.1. In general, the payload is expected to experience a temperature range between -70 to 30 °C. Since no high voltage circuits are included in the design, no acrylic conformal coating is required to prevent arcing. The main boards of the payload have been slightly insulated inside the vented foam enclosure and the heat dispersion of the circuitry during operation should keep the internal part of the payload slightly warmer than the outside temperature. No heater will be needed because all the components in the MegaSat design can withstand the short time that they will be exposed to below -40 °C temperatures. The lithium battery pack will be placed on the side of the MegaSat shield with additional foam surrounding it to offer more insulation and protection from the cold temperatures during flight to conserve battery life.

The box is made from two layers of $\frac{1}{2}$ -inch, moist-resistant insulation foam (Owens Corning Foamular PROPINK Insulating Sheathing). This foam is an extruded polystyrene (XPS) insulation board with a reinforced skin laminated on both sides for extra damage and moisture control.^[19] The $\frac{1}{2}$ -inch board has a R-3.0 value which will slightly limit the conductive heat loss of the payload during flight.

4.6 Mechanical Design

4.6.1 External Structure

During the flight path of the DemonSats-1, the payload will be subject to mechanical stress. This stress will come in the form of tension when the payload is acted upon by gravity and the acceleration upon ascent; compression when the payload is acted on by G-force; torsion as the payload resists rotational motion; shear as individual parts of the payload attempt to slide past one another; and bending as the unequal weight distribution of the payload acts against itself in flight and upon landing.

To mitigate mechanical strain caused by these forces, a double-walled foam box is used to house the payload. Additionally, inserts have been designed to protect fragile components and secure the power supply. The following components have been used to construct the enclosure: 1.3 cm thick polystyrene foam sheets, polyurethane glue, two plastic straws, four plastic grommets, carpet tape, and a roll of Econokote.^[20] The design of the enclosure is based on specifications provided through the LaACES student ballooning course materials. The foam box has the dimensions in the figure below and a lid that can be removed to access the internal payload. There are two holes placed 17 cm apart throughout the box and lid to allow for attaching the flight strings connected to the balloon vehicle (See Figure 19). There are inserts adhered to the bottom of the box to support the Arduino Mega, an insert used to separate the battery and the payload electronics, a small insert to secure the side of the battery, and an insert which sits on top of the payload electronics to secure its vertical motion within the box (See Figure 20).

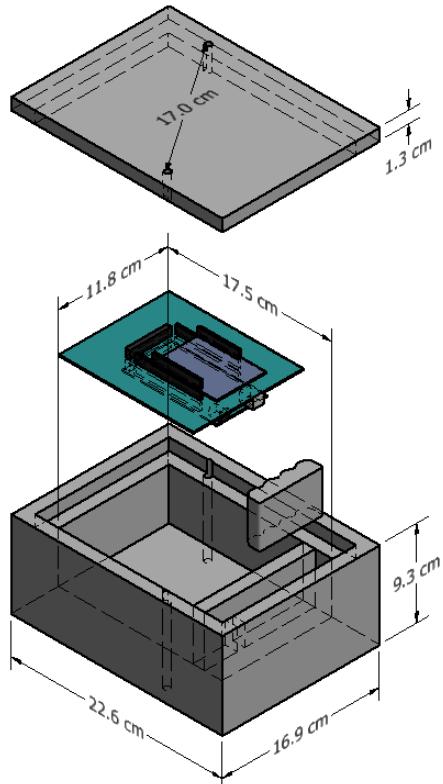


Figure 19: Drawing of Flight String Channels for DemonSats-1 Enclosure

There have been 2 grooves cut into the box to house the HIH-4000-003 humidity sensors. To secure the sensors in place, a window screen has been attached to the box prior to wrapping it with Econokote. Also, diagonal holes have been made to hold the external temperature sensor and the atmospheric pressure tubing in place while being exposed to the outside environment of the box. (See Figure 21). The holes were made diagonally to increase the surface area of the foam which will hold the sensor and tubing in place without compromising the mechanical strength of the box.



Figure 20: Images of empty box (left) and filled box with lid on side (right)

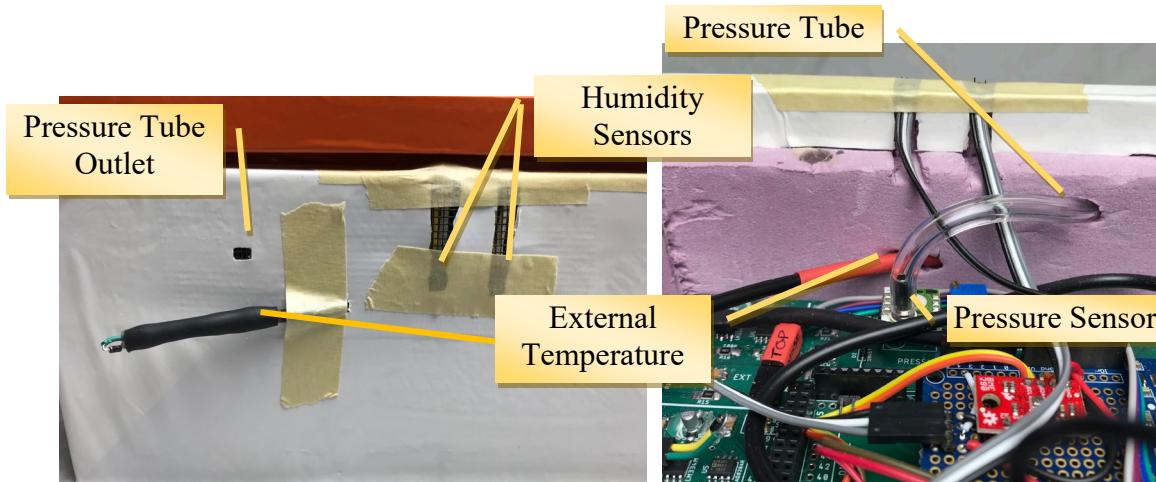


Figure 21: Images of the external sensor placement on box (left) and view inside the box (right)

4.6.2 Mass Budget

Table 15: The Mass Budget for DemonSats-1 Payload

Component	Mass (g)	Uncertainty (+/- g)
Arduino Mega 2560	37.0	1.0
GPS Logger Shield with microSD	24.0	1.0
MegaSat Sensor Shield	85.5	0.5
Battery Pack (2CR5 x 2)	88.0	1.0
Foam Box Enclosure	155.0	1.0
Miscellaneous (i.e. wiring, switch)	50.7	3.0
TOTAL	440.2 grams	1.3

The mass budget of the payload above shows the distribution of mass for the main components of the payload. The highest mass of the DemonSats-1 comes from the foam box enclosure including all of inserts, followed by the 12 V lithium battery pack and the completed MegaSat sensor shield. The mass of each main component listed have been determined using a gram balance. Replicate components not used in the payload were measured to determine uncertainties when possible, or datasheets were used to get the values. The completed payload mass has been determined to be 0.44 kg, or a weight of 4.3 N (see Figure 22).



Figure 22: Completed payload on the scale (0.44 kg)

5.0 Payload Development Plan

5.1 Electrical Design Development

Since the primary electrical design objective has been to build the LSU MegaSat payload, most of the electrical design development has been straight forward. The DemonSats team followed the assembly manual made by the LSU team to fabricate the payload. This assembly has been completed, and the shields have been tested for electronic functionality using a multimeter in the lab prior to the COVID-19 shutdown (See Section 6.1). Then the team and the faculty advisor tested the operation each of the sensors under expected operating conditions. When a problem occurred with a sensor as a result of the design, the team has attempted to test other sensors that could assist in the data collection for the project.

5.2 Software Design Development

The DemonSats team has programmed the MegaSat payload independently from the LSU team. Example code from Adafruit and Sparkfun have been utilized when effective for the GPS logger shield, MPU-6050, and BME280 modules. The overall control system interface has been programmed in Arduino IDE. The goal of this Arduino programming has been to send a text string of characters listing the sensor values to the microSD card at a 1/3 Hz sampling rate. For more details on the list of data collected and software flowcharts, see Section 4.4.

Following the COVID-19 shutdown of campus, the Arduino IDE programming subgroup members have been sent home with an Arduino Mega and GPS/microSD module. They have been working remotely to finalize GPS parsing and data storage programming. Meetings over Zoom with the programmers and faculty advisor have taken place to test the flight software functionality on the MegaSat payload. The faculty advisor has taken the responsibility in programming the I2C communication syncing between the DS3231 RTC, MPU-6050 accelerometer, and BME280 atmospheric sensor. The team has been instructed on the programming technique used for the I2C portion of the program since even the faculty advisor had not used a manual method of calling registry commands to communicate with different types of sensors prior to this project.

The DemonSats also have been setting up a Visual Basics Macro within Microsoft Excel to perform the data analysis operations described in Section 4.4. From the microSD card, a comma-delimited text string is transferred to a Microsoft Excel workbook. Then the next macro operations are to parse the data needed for the scientific objectives into different worksheets to create the graphs and perform the calculations. With the team's limited experience using Microsoft Excel, this macro development has been an opportunity for them to learn an advanced function of the program and to practice basic programming using Visual Basics. Since the closure of campus, the macro development has been done at home under the direction of the data analysis manager and faculty advisor.

5.3 Mechanical Design Development

The team has utilized the instructions provided by LSU to build the payload enclosure. During the development of the first box, the team discovered that the dimensions in the LSU instructions did not work for the DemonSats-1 payload completely. The team suspects that the length dimensions of the main boxes pieces were measured off by one thickness of the foam sheet when the instructions were produced. Table 16 below lists the individual foam piece dimensions for the final payload enclosure.

Table 16: The Foam Board Template for the Final DemonSats-1 Payload Enclosure

	Cut lengths (cm)			Final Lengths (cm)			Quantity needed
	Length	Width	Depth	Length	Width	Depth	
Top and Bottom Outer Piece:	24	17.7	1.4	23.8	17.5	1.4	2
Top and Bottom Inner Piece:	21	14.8	1.4	20.8	14.6	1.4	2
Outer Side Walls:	22.6	8.9	1.4	22.4	8.7	1.4	2
Outer End Walls:	16.2	8.9	1.4	16	8.7	1.4	2
Inner Side Walls:	19.4	5.9	1.4	19.2	5.7	1.4	2
Inner End Walls:	13.2	5.9	1.4	13	5.7	1.4	2
Above MegaSat Filler board:	14.7	11.7	1.4	14.5	11.5	1.4	1
Battery Separator Wall:	11.9	5.9	1.4	11.7	5.7	1.4	1
Filler by Battery	5.9	2.2	1.4	5.7	2	1.4	2

The faculty advisor with virtual assistance from mechanical design team manager completed the double-walled foam box enclosure. Thermal and vacuum tests have been performed, and the data has been described in Section 6.4. After observing the robustness of the completed box, having a limited number of supplies available, and due to the limited access to impulse testing and video equipment, the faculty advisor has advised the team to not perform and report data on the shock testing which was originally planned and described in the CDR. The team has decided to trust that this borrowed design from LSU will be effective enough to keep the microSD card protected during the impact from the fall. Future DemonSats payloads can be tested if the lab remains open for the full academic year.

5.4 Mission Development

Throughout the development and finalization of the DemonSats-1 payload, eight NSU undergraduates and LSMSA high school students have been involved in various parts of the project. The details of each team member's roll have been described in Section 9.2. Following the COVID-19 shutdown of campus on March 13th, the team has continued to meet through WebEx and Zoom at least once a week, with additional meetings for subcommittees as needed.

Since the CDR, the team has been actively involved in learning project management topics, how to program the Arduino Mega, and how to calibrate temperature, humidity and pressure sensors through virtual meetings and videos made by the faculty advisor. Since this project has been extended into the summer, one of the team members has left the team due to his graduation and new obligations. Three other team members have also graduated and have chosen to remain active members of the team. Three team members have also been working as

full-time essential employees and have had to forcibly decrease their amount of time to contribute to this project. So due to the campus shutdown and summer extension, the team's mission has developed into sheer determination to produce a finished, working payload and to complete the required reports and presentations.

6.0 Payload Construction Plan

6.1 Hardware Fabrication and Testing

Since the MegaSat comes as a kit from LSU, the hardware fabrication was completed early in March, and the basic electrical checkpoints have been tested. The MegaSat shield has been put together using the MegaSat Assembly Manual^[1] provided with the kit. The power, temperature, humidity, pressure, RTC, and MPU-6050 modules checkpoints have been verified and are reported in the table below. For the current measurement on the temperature modules, the voltage was measured across a 1 kOhm resistor, and Ohm's law was then used to determine the current since the lab equipment available was not sensitive enough to detect 1 mA effectively.

Table 17: Summary of Electronic Checkpoints for MegaSat Shield

ID	Name	Expected Value	Measured Value
+12 V	Input Power	+12 V	+11.98 V
+5 V	5 V Linear Regulator	+5 V	+4.9 V
+3.3 V	3.3 V Linear Regulator	+3.3 V	+3.2 V
+5 V REF	5 V Reference	+5 V	+4.9 V
CS1 – Pin 4	Voltage Check (Int Temp)	+12 V	+11.6 V
J3	Sensor Supply Current	1 mA	0.96 mA
CS2 – Pin 4	Voltage Check (Ext Temp)	+12 V	+11.6 V
J4	Sensor Supply Current	1 mA	0.95 mA
J2	Voltage Check (Humidity)	+5 V	+4.9 V
CS3 – Pin 4	Voltage Check (Pressure)	+12 V	+11.6 V
U2 – Pin 2, 13	Voltage Check (RTC)	+3.3 V	+3.1 V
U1 – Pin 1, 2	Voltage Check (MPU-6050)	+3.3 V	+3.1 V
AD0	MPU6050 Jumper on Board	+3.3 V	+3.1 V

The team has verified that all the analog sensors respond to their respective measurand with voltage changes which are converted to decimal numbers by the 10-bit analog/digital conversion of the Arduino Mega 2560. See Section 7.1.1.2 for the final calibration results of 10-bit decimal readings to measurand for each 1N457 temperature sensor, HIH-4000-003 humidity sensor, and the 1230-015A-3L pressure sensor.

6.1.1 Temperature Sensing System Testing

In Figure 23, an inverse relationship was observed between the voltage output from the external temperature sensor module and the temperature. For this reproducible ($n=3$, 2 different person) experiment, the temperature sensor was compared to calibrated Type-K and Type-T thermocouples while being placed on dry ice, in an salted ice bath, in an ice bath, in a warm water bath, and at room temperature. The voltage measurements were taken with a Vernier voltage sensor attached to a Vernier LabPro DAQ. LoggerPro 3 software was used to collect the voltage data from the MegaSat external temperature sensor by attaching the Vernier voltage sensor to the Analog 1 (A1) header pin and a ground (GND) header pin of the MegaSat shield. The results (See Figure 23) showed an inverse linear trend of $\text{Voltage} = -0.0154 * (\text{temperature in } ^\circ\text{C}) + 3.0208$. Since the detected voltages in the experiment were between 0 and 5 V, the sensor's circuit effectively converted the current change in the 1N457 diode into the voltage change that the Arduino Mega can detect, so no alterations were made to the sensor hardware.

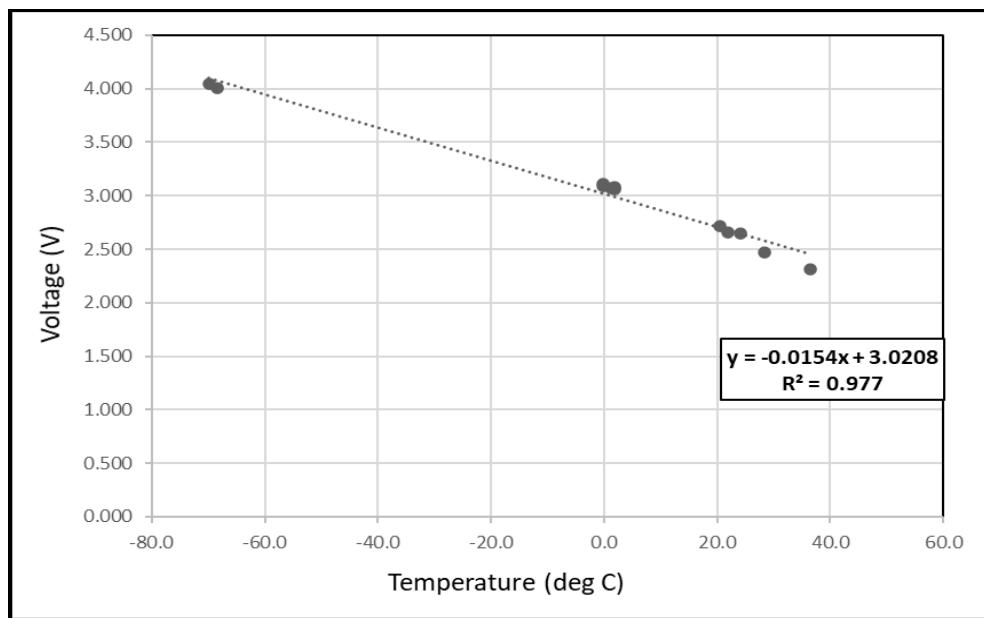


Figure 23: The Comparison of Voltage Output to Temperature for the Red-coated External Temperature Sensor ($n=4$)

Then external and internal temperature sensor modules were characterized by measuring the decimal numbers from the 10-bit analog/digital conversion by the Arduino Mega using a simple Arduino IDE analog read to microSD log program. The purpose of these experiments was to establish the calibration equations to be used in the Arduino IDE flight code program to convert the decimal readings into degrees Celsius. The procedure consisted of placing the external and internal temperature sensors into beakers of warm water, fridge-cooled water, iced water, iced salt water, dry ice cooled water, and dry ice alone (See picture of setup example in Figure 24.). The sensors were compared to Type-K thermocouples which were read by a Kamtop Digital Thermometer. The results of these experiments are in Section 7.1.1.2.



Figure 24: Temperature Calibration Experimental Setup

6.1.2 Pressure Sensing System Testing

The pressure sensor module has been characterized by measuring the decimal numbers from the 10-bit analog/digital conversion by the Arduino Mega using a simple Arduino IDE analog read to microSD log program while the payload is placed into a vacuum chamber with attached pressure transducer. See Figure 25 for a picture of the setup. During the test, vacuum was applied using a single state vacuum pump and adjusted by 5 in Hg incremental stages between -10 and -30 in Hg to sync the sensor output with the pressure gauge. The results of this experimental method yielded a stepwise data response from the 1230-015A-3L pressure sensing system as seen in Figure 26. The data shows steady readings at steady pressures inside the vessel and a quick response time during transitional pressures. To explore pressure system's response time further, a test of constant pressure increase was used along with a timer to determine the difference between the gauge readings on the vessel and the MegaSat pressure measurement. Figure 27 shows no difference between the timer and the elapsed time recorded from the pressure sensing system. Therefore, once the pressure sensing system is calibrated to record pressure in absolute millibars, the system should be accurate and responsive enough to satisfy the scientific and technical requirements for pressure measurement. The sensor has been compared to a Bourdon-tube gauge which can have an uncertainty of $\pm 5\%$ f. s.^[21] Therefore, the calibration of the pressure sensing system can be within 50 millibars since the gauge used in calibration could be off by this value.



Figure 25: Pressure Calibration Setup

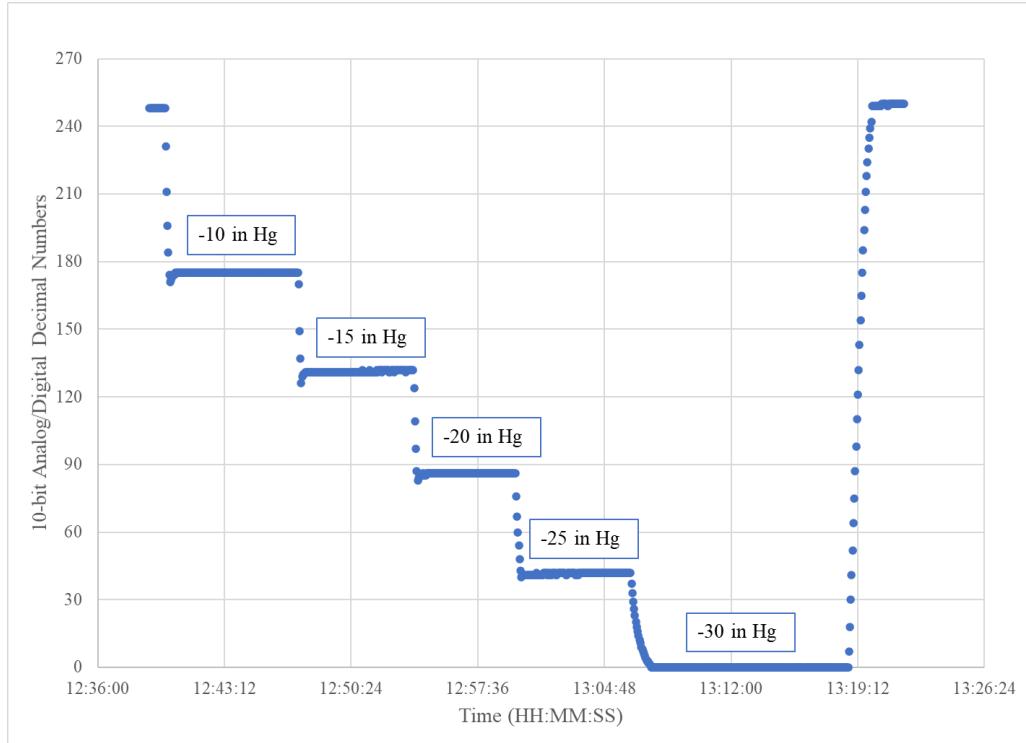


Figure 26: Initial Results of the MegaSat Pressure Sensing System Output Obtained Using Vacuum Chamber with Bourdon-style Oil-fill Pressure Gauge for Comparison



Figure 27: Example of Pressure Sensing System Response Between -30 to 0 in Hg (Decimal Equivalents) When Compared to a Timer

To calibrate the pressure sensing system to record pressure in absolute millibars, the comparison data from the Bourdon tube pressure gauge had to be converted from in Hg vacuum gauge readings to absolute pressure millibar readings. Table 18 shows the conversion equivalents between the gauge and absolute pressure readings used to compare to the 10-bit A/D decimal numbers recorded from the DemonSats-1. The calibration results are in Section 7.1.1.2. On a side note, the gain potentiometer was left at 5 kOhm and the difference in recorded data and gain adjustment was not determined for the DemonSats-1 payload. The reason this experiment was not done was due to sensing system's sensitivity already satisfying the technical requirements and the team's limited capability to experiment on supplementary questions.

Table 18: List of Pressure Conversions Used for Pressure Calibration Data [22]

Inches of Mercury (in Hg) gauge	Millibars (absolute)
0	1014
-5	843.88
-10	674.45
-15	505.68
-20	333.83
-25	167.80
-29.92	0

6.1.3 Humidity Sensing System Testing

Initial calibrations of the HIH-4000-003 humidity sensor on the MegaSat provided evidence that the active filter may not be correctly designed. The Arduino's decimal conversion values of the sensor's voltage showed a linear trend when compared to the % relative humidity measurand; however the sensing system exhibited a limited range near 80% of the 0 to 100% RH range that is possible and needed to fulfill our science requirements (See Figure 28). The experimental procedure to explore the humidity detection used the following saturated salt solutions:

Table 19: List of Saturated Salts Used in Initial Humidity Sensor Calibration [14]

Saturated-salt solution chemical	Expected %RH of air inside chamber (at 30°C)
Lithium Chloride (LiCl)	11.30% ± 0.35
Sodium Chloride (NaCl)	75.09% ± 0.11

When comparing the payload's humidity sensor decimal conversion readings to the VLIKE MS6508 Digital Temperature Humidity Meter inside the saturated salts solutions, it was observed that the decimals were near the maximum number of 1023 (2^{10}) when in the 75% RH chamber. The team then decided to explore with the active filter by adjusting the 10 kOhm gain potentiometer. Figure 28 shows that the results of the gain adjustment did not effectively change the %RH range measured by the sensor. Since this discovery happened during COVID-19 "stay-at-home" restrictions, the team decided to not pursue redesigning the filter since the equipment to do so was on campus. Instead, the team decided that the simpler solution would be to adjust

the offset potentiometer of the existing MegaSat humidity sensing system to start near 0% RH for 0 decimals and allow it to measure values from 0 to 80%, then to add a directly wired (no filter) HIH-4000-003 sensor to another analog pin (A4) to get the remaining 80 to 100% RH with less sensitivity. The faculty advisor and team leader knew that this solution was possible since the same humidity sensor was wired this way on the RockSat payload which they built at the RockOn! Workshop the previous summer.

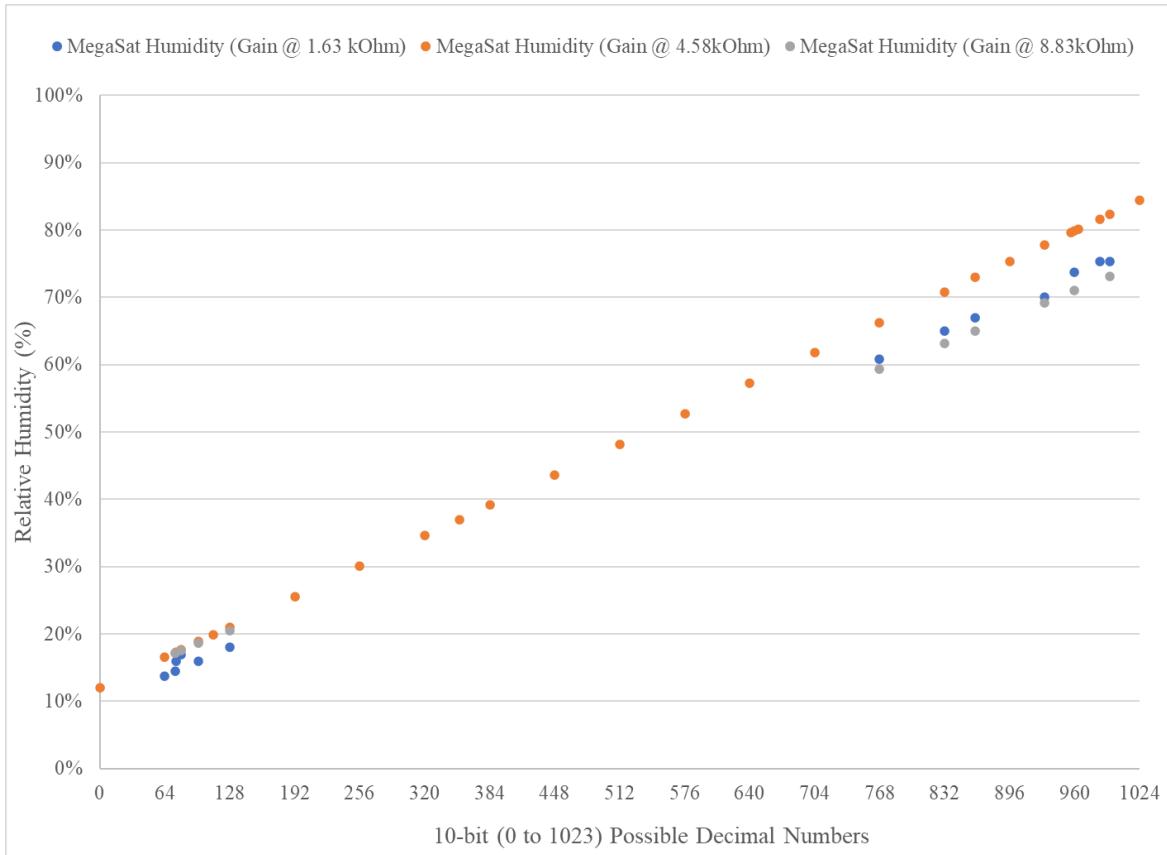


Figure 28: The Effect of Gain Potentiometer Adjustments on the %RH Range of the MegaSat Humidity Sensing System (Gain @ 4.58 kOhm was selected as the final setting)

Based on the preliminary gain adjustment results, the gain was set to 4.58 kOhm (schematic # R8), and the offset was shifted to 2.48 kOhm (schematics #R7) so that 0% RH yielded a near 0 decimal reading. Once these settings were determined, the sensor calibration equation used in Arduino programming was determined using five different saturated salt solutions. (See Table 6 on page 21 for the list of salts used.) For the calibration experiments, the salts were wetted in glass jars, and once in thermal equilibrium, the humidity sensors were placed in the air above the saturated salt for several minutes until the VLIKE MS 6508 placed beside them had stable readings of the contained air. For lithium chloride, the sensors were placed in the jars for at least an hour, and for the higher %RH chambers, the monitoring meter reached equilibrium within 30 minutes. (See Figure 29 for a picture of the setup.)

Both humidity sensors used on the DemonSats-1 payload were taped together with the sensor plates exposed and placed into the salt chambers together. The readings from the analog-direct HIH-400-003 humidity sensor showed that the full range of %RH could be detected from this sensing system in the decimal range of 144 to 857. However, the sensitivity of measurement was less than the MegaSat humidity sensing system. The sensitivity of the MegaSat humidity system was determined to be 15.8 decimals per 1% RH whereas the Analog-Direct humidity system sensitivity was 7.1 decimals per 1%RH. (See Figure 30 for the comparison between the two sensing systems.) From this observation and knowing that low humidity detection will be more important to achieve the scientific requirements than high humidity detection, the team decided to use both sensors to obtain data for the full range but specifically use the more sensitive MegaSat humidity system for the low side of %RH range to get the best humidity data the DemonSats-1 payload can achieve in its current setup.



Figure 29: Humidity Calibration Setup Example

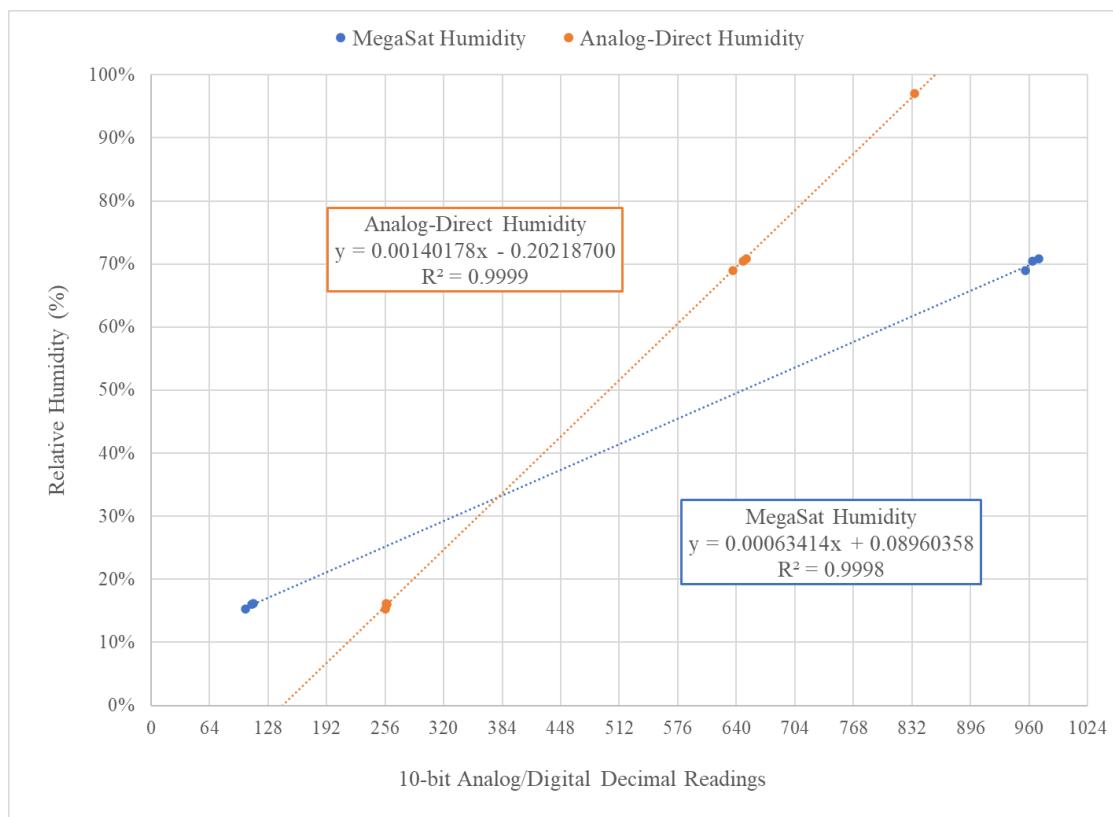


Figure 30: The Comparison of the MegaSat Humidity Sensing System and the Analog-Direct Humidity Sensor Outputs

6.1.4 Altitude Sensing System Testing

Altitude on the MegaSat design was determined from the three-dimensional GPS data obtained using the Adafruit Ultimate GPS module. Because of the nature of its determination, the altitude was only be observed and reported by the team. Altitude data was obtained from automobile ground travel from Natchitoches to Shreveport and compared to GPS tracker elevations from an iPhone app (Coordinates-GPS Formatter version 6.4 by Mapnitude).

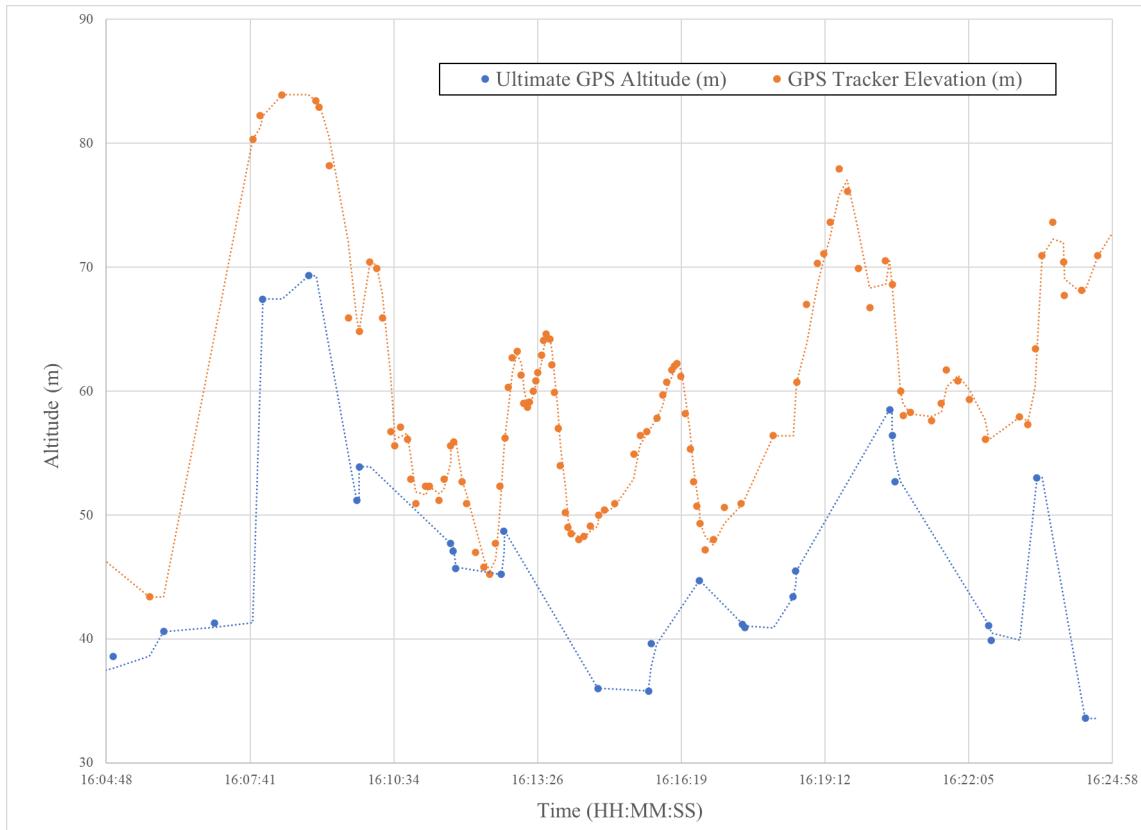


Figure 31: A Portion of the Road Trip Data Comparing the Adafruit Ultimate GPS Module’s Altitude Measurement with the Coordinates iPhone GPS Tracker Elevation data

During the road trip, the altitude varied from 35 m to 84 m above sea level. Based on the GPS Tracker Data, the altitude measured by the payload was within 50 m the entire trip with the biggest difference being 42.4 m (on right in Figure 31). The same graph showed that the GPGGA data or fixed data was collected irregularly and sometimes sparingly (~2 minutes between readings) despite the payload being on an open road with few obstructions to the satellites. Since the Adafruit Ultimate GPS was considered by the team to not be reliable enough to provide consistent altitude data, the Sparkfun BME280 Atmospheric Sensor Breakout board was installed on the DemonSats-1 payload. In Figure 32, the data collected by the Ultimate GPS module and the BME 280 module was compared to the iPhone GPS Tracker app data. There was a clear correlation between the GPS Tracker data and the BME280 in terms of accuracy and resolution. Therefore, the team has been convinced that adding the BME280 module will improve the scientific data results, at least until its elevation limits has been reached.

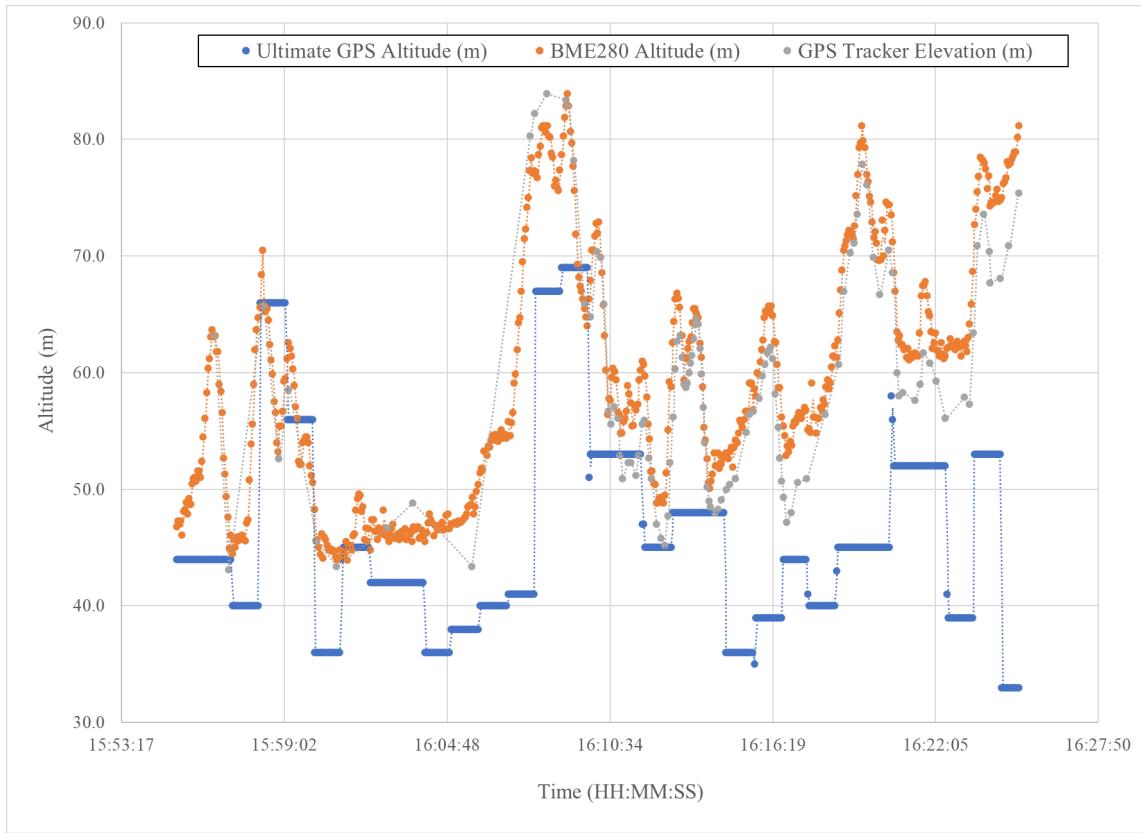


Figure 32: A Portion of the Road Trip Data Comparing the Adafruit Ultimate GPS Module and the Sparkfun BME280 Sensor Breakout in Altitude Measurement with the Coordinates iPhone GPS Tracker Elevation data

6.1.4 GPS Latitude/Longitude Sensing System Testing

Latitude and longitude are determined from the two-dimensional GPS data obtained using the Adafruit Ultimate GPS module. Because of the nature of its determination, these directional coordinates can only be observed and reported by the team. Unlike the altitude data, the latitude and longitude data can be reported by the GPRMC and the GPGGA-type NMEA sentences. Therefore, the team expected to receive data with a 3-second resolution during testing.

The GPS testing data was obtained from automobile ground travel from Shreveport to Natchitoches and back. The data was removed from the microSD card and then converted from degrees minutes (DDMM.MMMM) to degree decimals (DD.DDDD) in Microsoft Excel. Then the latitude and longitude coordinates were mapped using the nesting “3D Maps” add-on within Excel. The 3D Maps add-on placed the coordinates onto a Bing map, and the time which the data was collected could be added to the map for video verification of the map path. Figure 33 shows the results of the location tracking from the road trip.

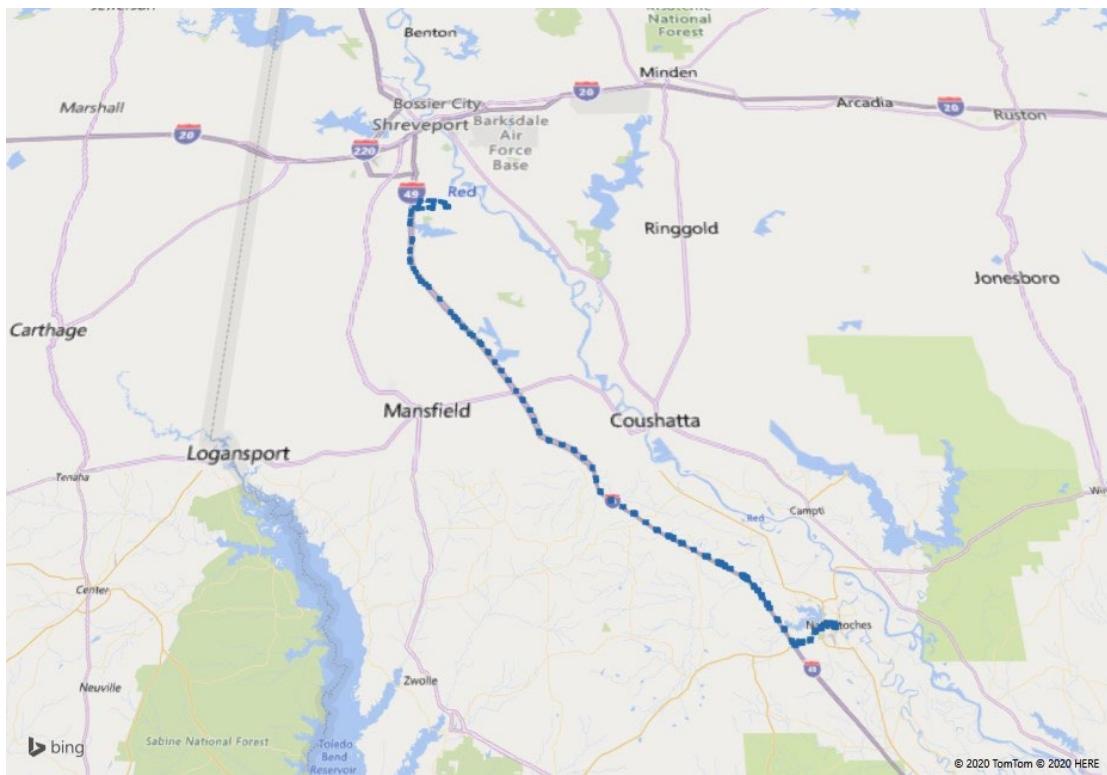
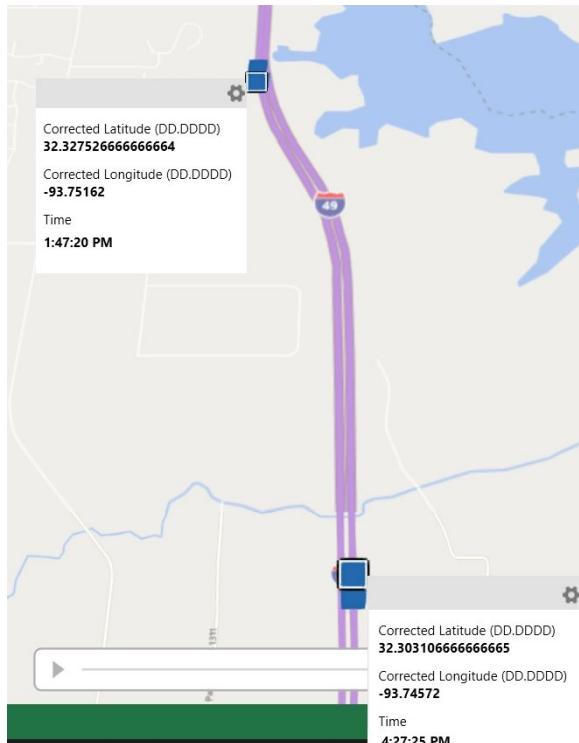


Figure 33: Data from the Adafruit Ultimate GPS Module Collected and Converted for on a Road Trip from Southeast Shreveport to Natchitoches and Back



The results were very accurate (within 10 m) when compared to the iPhone app Coordinates-GPS Formatter (version 6.4 by Mapnitude). The average reported accuracy of the iPhone GPS app was ± 6.7 m. In Figure 34 to the left, there is visual verification that the GPS coordinate data was accurate enough to detect which side of the interstate (I-49) the DemonSats-1 payload was on. Since the payload travelled between Shreveport to Natchitoches, and then back, the payload was on the Southbound lanes first, then it was on the Northbound lanes for the return trip.

Figure 34: Example for GPS Coordinates Accuracy using the Adafruit Ultimate GPS Module

6.1.5 MPU 6050 Accelerometer/Gyroscope Sensing System Testing

Since the accelerometer and gyroscope data were not needed to meet the scientific objectives and requirements for the DemonSats-1 payload, only basic functionality testing was performed. For the accelerometer data collected by the Sparkfun MPU6050, the numbers recorded were converted from the 14-bit decimal values provided by the module by dividing them by 16,384 (or 2^{14}). After the conversion, the accelerometer was then calibrated to zero in the X, Y, and Z axes marked on the MPU6050 module.

First the DemonSats-1 payload was placed on a level surface so that the Z-axis was facing upward, the Y-axis was pointing to magnetic North, and the X-axis was pointing due east and then left undisturbed for 15 mins. The expected result of the accelerometer was Ax, Ay, Az = (0 g, 0 g, 1 g) for this orientation. Then the box was arranged to have the Y-axis pointed upward to get Ay = 1 g, and after 15 minutes, the box was moved so that the X-axis was pointed upward to get Ax = 1 g for 15 minutes. The experimental results were then averaged, and offset values (see Table 20) were determined which were used to correct the Ax, Ay, and Az readings to received acceleration data with respect to gravity (g's). The gyroscope data from the same experiment was used to determine the offset for the gyroscope sensor (see Table 20).

Table 20: List of Offset Values for the MPU6050 Data for the DemonSats-1

Sensor Orientation	Offset Value
Accelerometer X-axis (Ax)	-573
Accelerometer Y-axis (Ay)	324
Accelerometer Z-axis (Az)	1305
Gyroscope X-axis (Ax)	826
Gyroscope Y-axis (Ay)	-48
Gyroscope Z-axis (Az)	-96

6.2 Integration Plan

The subsystems are to communicate through an Arduino IDE program and operated by the Arduino Mega 2560. Initially each of the subsystem programming was developed and tested for operation. The GPS Soft Serial Introduction example code from Adafruit was used as the main architecture for the flight code. Then microSD logging programming was incorporated to allow for the data logging and storage during further program testing. Then a basic analog read programming was incorporated to start to calibration of the 10-bit analog/digital decimal numbers to meaningful values of pressure (millibar), temperature (°C), and humidity (RH). Then general IDE programming language was used to implement the analog calibration equations and to fix the date and time stamp for the GPS data from GMT to CDT. Finally, the I2C programming was incorporated to communicate with the DS3231 RTC and MPU6050. Lastly, the BME280 I2C language was added to incorporate the sensor.

Much of the programming was done by the programming subcommittee and the faculty advisor through MS Teams chats after the COVID-19 campus shutdown. Three team members were sent an Arduino Mega, Ultimate GPS module, and soldering supplies to test their

programming at home. The faculty advisor had the MegaSat shield at her home with the available testing equipment, so the complete flight code integration and testing was finished through online collaborations.

6.3 Flight Software Implementation and Verification

The integrated flight code was designed to collect data from all the sensors into a single line of comma-delimited strings, which was then appended to a text file stored on the microSD card. The data rate was set to 1/3 Hz, or every three seconds, to adhere to the DemonSats-1 scientific requirements. Once the complete flight code was established, thermal and vacuum tests of the complete payload enclosed in a foam box were performed by the faculty advisor with online assistance from the team in experimental design and data analysis. The details of flight software verification have been placed in Section 6.1 and 6.4.

6.4 Flight Certification Testing

The payload may experience the following flight conditions: temperatures of -70 to 30°C, atmospheric to vacuum pressures, 0 to 100% relative humidity, calm to windy conditions, and rotation from 0 to 5000 deg/sec.

Temperature fluctuations which could harm the main electronics were tempered by an insulated foam box. To test the thermal stability of the payload, the DemonSats-1 payload was placed in the low temperature environments available (-12°C household freezer and 4°C household refrigerator) for a duration of 1 hour in each. The data collected on the microSD card during these tests was analyzed to ensure proper functioning of the payload, especially the external temperature sensor (See Section 6.4.1 for more details).

Pressure fluctuations were simulated in a small, portable vacuum chamber with attached Bourdon gauge. To test the vacuum stability of the payload, the DemonSats-1 payload was exposed to vacuums of near 0 to 1014 abs. millibars. The data recorded to the microSD card was analyzed to ensure proper functioning of the payload, especially the pressure sensing system (See Section 6.4.1 for more details).

The humidity fluctuations were observed by exposing the payload to changing humidity conditions within a freezer, fridge, and in a pressure vessel. Unfortunately, the team did not have a way to record comparison humidity data over time or in an opaque enclosed chamber, and faculty advisor accidentally broke the handheld humidity sensor she had before the humidity system tests were performed.

The windy and rotational fluctuations were accounted for in the design of the payload enclosure. The foam box was made to fit snuggly around the payload electronics to prohibit it from shaking loosely around inside the box during flight rotation. Rotational stability testing of the payload was not officially performed with measurements. Instead, a vigorous shaking of the

box in the 3-axial directions was performed to ensure that the payload would still make measurements afterwards and that the payload remained in its designed position within the box.

6.4.1 System Testing Procedures

After the campus shut down due to COVID-19, the faculty advisor took the payload and testing equipment to her house. The team had the unfortunate issue that many of the members live in different towns several hours apart and some team members moved to full-time employment as essential workers since the COVID-19 shutdown and now have full-time summer employment. Therefore, the faculty advisor had performed the following system tests and shared this data with the team through Microsoft Teams. Weekly meetings over Zoom, recorded videos and data placed on the Microsoft Teams platform were the methods in which the faculty advisor shared the data and results with the team members.

6.4.1.1 Thermal Test Procedure

- The payload was placed inside an insulated foam box with extra insulation around the battery pack (See Section 4.5 for more details).
- The box with the payload was placed in low temperature environments: -12°C freezer and 4°C refrigerator, with a dwell time of 1 hour (-80°C has been unavailable since campus shutdown).
- The sensors collected and recorded data to the microSD card, which was retrieved from the payload after testing was complete.
- The data logged during the experiments was analyzed to ensure that the payload could withstand the thermal conditions that it will experience in flight without affecting its integrity or functionality.
- The external temperature sensor should detect temperatures within 5°C of the Type-K thermocouples used for comparison.



Figure 35: Fridge System Test Example

6.4.1.2 Vacuum Test Procedure

- The payload was placed inside an insulated foam box with extra insulation around the battery pack (See Section 4.5 for more details).
- The box with the payload was placed into a small, stainless steel vacuum chamber.

- The vacuum pump attached to the chamber was turned on, and pressure was vacated at 5 in Hg increments until full vacuum. Then the vacuum was released through a valve in 5 in Hg increments. Each increment of time was approximately 5 minutes.
- The sensors collected and recorded data to the microSD card, which was retrieved from the payload after testing was complete.
- The data logged during the simulation was analyzed to verify that the payload can withstand the pressure conditions that it may experience in flight without affecting its integrity or functionality.
- The pressure sensing systems should detect pressures within 5% f. s., or 50 millibars, from the original calibration data in millibars.

6.4.2 System Testing Results

6.4.2.1 Thermal Test Results

The fridge system test consisted of placing the DemonSats-1 payload into a household refrigerator along with two Type-K thermocouples for 1 hour. The thermocouples were read by a digital thermometer located on the outside of the fridge and recorded manually every two minutes as a comparison. In Figure 36, the external temperature sensor data trended very well with the thermocouple readings. This result was expected since the thermocouples were taped right next to the external temperature sensor. Based on this data, the payload had satisfied the requirement to detect temperatures within $\pm 5^{\circ}\text{C}$ of the Type-K thermocouples used for comparison.

As for the results of the internal and BME280 on-board temperature sensors, the data agreed with what the team had expected in that both sensors showed that the payload electronics warmed up the interior of the enclosure. The difference (greater than 10°C) between the final readings of the internal and BME280 sensors was surprising to the team, though. The reasoning behind the temperature difference was likely due to the BME280's proximity to the Arduino's power module which was the main heating source when compared to the internal temperature sensor, which was on the opposite side of the payload. The BME280 also likely received heat energy conductively since it was housed on top of the GPS shield, whereas the internal sensor was suspended in the air of the enclosure and received mostly convective heat energy.

The freezer system test was similar in protocol to the fridge test except that the payload was placed in a household freezer. In Figure 37, the external temperature sensor data trended very well with the thermocouple readings, with a maximum difference of $\pm 3.23^{\circ}\text{C}$. This result was expected since the thermocouples were taped right next to the external temperature sensor. Based on this data, the payload had satisfied the requirement to detect temperatures within $\pm 5^{\circ}\text{C}$ of the Type-K thermocouples used for comparison. The internal temperature and BME280 also showed warming inside the box compared to the lower exterior temperature.

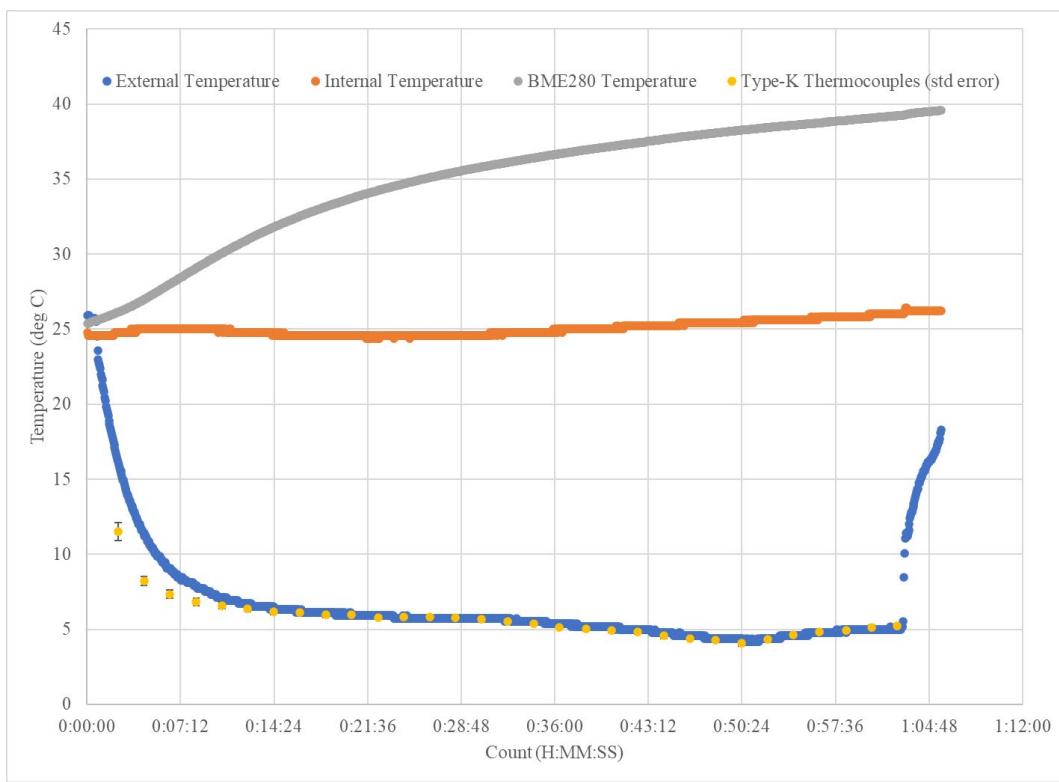


Figure 36: Temperature Trends of DemonSats-1 Payload Sensors for Refrigerator System Test Compared to Type-K Thermocouples (error bars are in standard error)

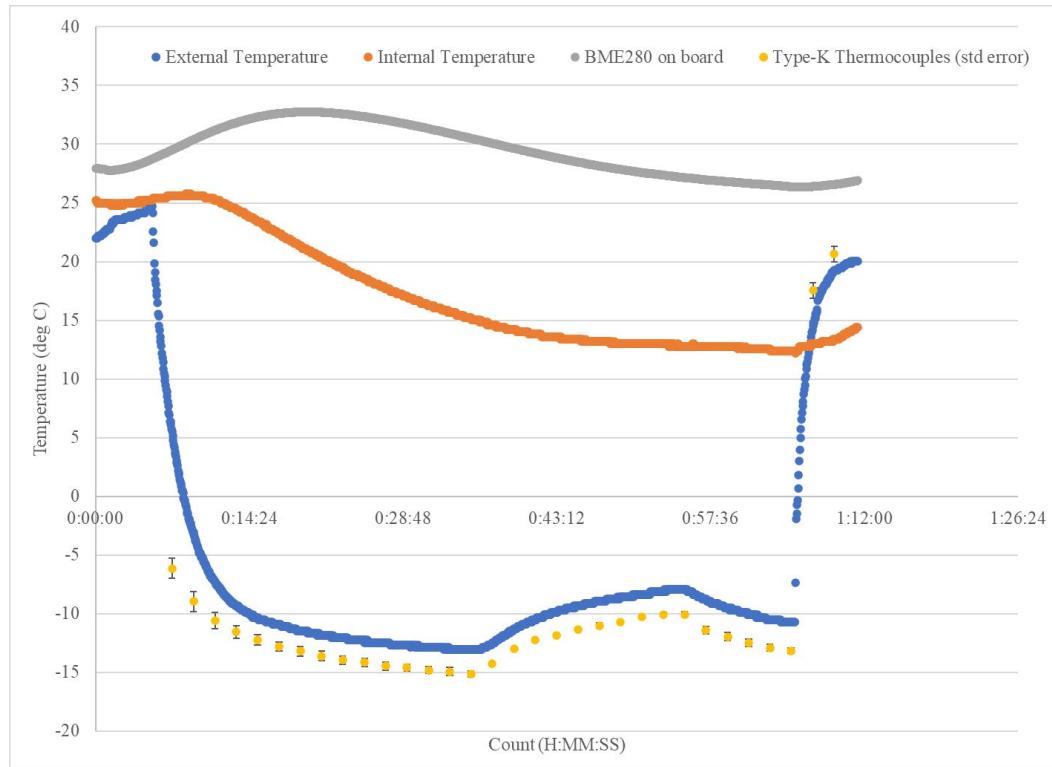


Figure 37: Temperature Trends of DemonSats-1 Payload Sensors for Freezer System Test Compared to Type-K Thermocouples (error bars are in standard error)

The detection of humidity fluctuations by the DemonSats-1 payload during the thermal system testing was also explored. Figures 38 and 39 showed that the placement of the HIH-400-003 humidity sensing systems on the outside of the payload enclosure allowed the payload to detect the humidity fluctuations which occur during the refrigeration cycle. The BME280 sensor module was inside the enclosure. The team suspected that the warming inside the enclosure and the insulating nature of the foam kept the interior humidity sensor from detecting the exterior humidity cycles. This evidence gave the team evidence to keep the HIH-4000-003 sensors on the outside of the box.

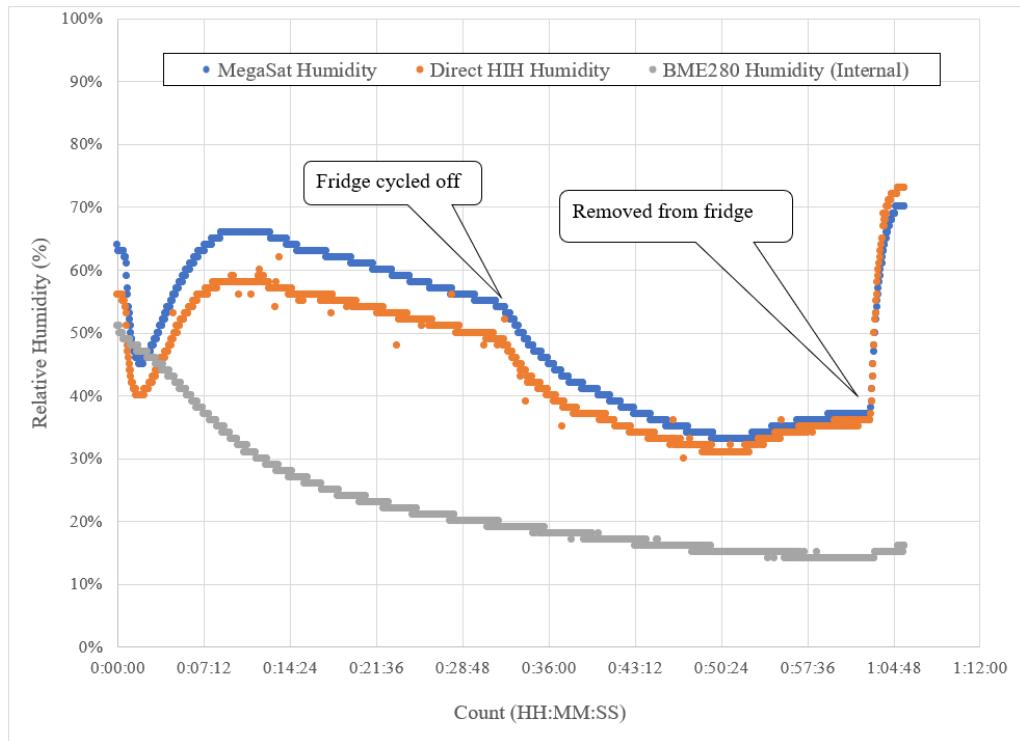


Figure 38: The % Relative Humidity Trends of the Different Humidity Sensors in a 4 deg C Fridge System Test with One Refrigerator Cycle

Since housing the HIH-400-003 humidity sensors on the outside of the box could make the sensors susceptible to condensation, the sensor plates were covered with masking tape during the design process to deter moisture from settling on them when exposed to excess water vapor. At the end of the thermal system test in the freezer, the payload collected some condensation on its exterior when it was removed from the freezer compartment. The humidity data in Figure 39 showed an increase to 100% when the payload collected condensation, but the humidity sensors returned to lower humidity even before the temperature of the payload returned back to room temperature and as the condensation evaporated away completely. Therefore, the thermal system tests also showed evidence that the humidity sensing systems show be responsive to humidity fluctuations.

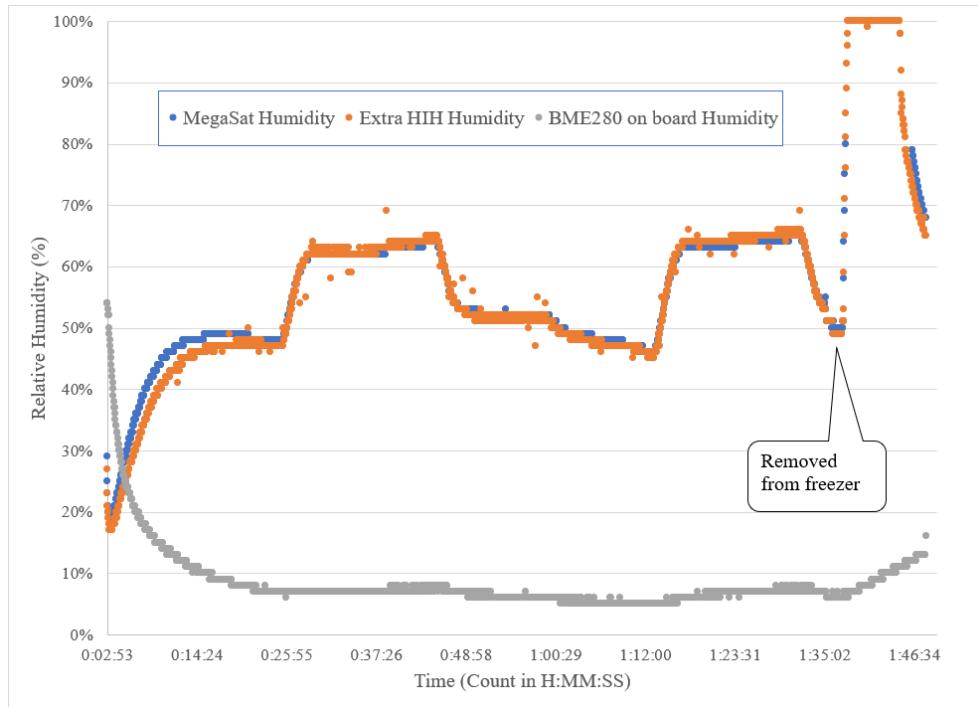


Figure 39: The % Relative Humidity Trends of the Different Humidity Sensors in a -15 deg C Freezer System Test with Two Refrigerator Cycles

6.4.2.2 Vacuum Test Results

Since the team was not able to travel to LSU-Baton Rouge for thermal/vacuum testing in April thanks to the COVID-19 pandemic, the vacuum payload testing was performed in the small vacuum vessel located at the faculty advisor's home. The pressure sensing system response during testing was compared to a Bourdon tube reading as described in 6.4.1.2 and displayed in Table 21. The pressure measurement was within 5% for the MegaSat pressure sensing system during the system testing, which met the science requirement for pressure. The BME280 pressure sensing system was not within the 5% requirement during the pressure increase part of the testing. The team suspects that the BME280 module's pressure data might be temperature sensitive since its temperature reached 59°C during the test.

The detection of humidity fluctuations by the DemonSats-1 payload during the vacuum system testing was also explored. Figure 40 showed that all of the humidity sensing systems were able to detect changes in %RH as the air was gradually evacuated from the test chamber. The HIH-4000-003 sensing systems were also able to detect the %RH change that occurred at the beginning of the test when the payload was taken from indoors to outside. During the vacuum test, the %RH was also measured by the VLIKE MS6508 digital humidity meter. The external humidity sensors of the payload trended very closely (~3%) with the comparison meter.

Table 21: Pressure Results from Vacuum Test

MegaSat Pressure System								
Bourdon Gauge Reading(in Hg)	Millibars (absolute)	Atmosphere to Vacuum	± std deviation	% f.s. difference between calibration	Vacuum to Atmosphere	± std deviation	% f.s. difference between calibration	% difference between directions
-5	843.88	889.53	±0	4.5%	893.55	±0	4.9%	0.45%
-10	674.45	712.42	±0	3.7%	712.42	±0	3.7%	0.00%
-15	505.68	543.38	±0	3.7%	555.45	±0	4.9%	2.22%
-20	333.83	366.27	±0	3.2%	382.21	±0.81	4.8%	4.35%
-25	167.8	186.7	±1.97	1.9%	203.68	±1.98	3.5%	9.09%
-30	0	6.8	±1.87	0.7%	6.8	±1.87	0.7%	0.00%

BME280 Pressure System								
Bourdon Gauge Reading(in Hg)	Millibars (absolute)	Atmosphere to Vacuum	± std deviation	% f.s. difference between calibration	Vacuum to Atmosphere	± std deviation	% f.s. difference between calibration	% difference between directions
-5	843.88	894.53	±1.70	5.0%	935.10	±1.38	9.0%	4.5%
-10	674.45	726.26	±1.35	5.1%	750.22	±0.18	7.5%	3.3%
-15	505.68	564.04	±0.92	5.8%	591.89	±0.11	8.5%	4.9%
-20	333.83	394.33	±1.31	6.0%	415.37	±0.25	8.0%	5.3%
-25	167.8	218.57	±1.51	5.0%	237.73	±0.34	6.9%	8.8%
-30	0	42.47	±1.89	4.2%	42.47	±1.89	4.2%	0.0%

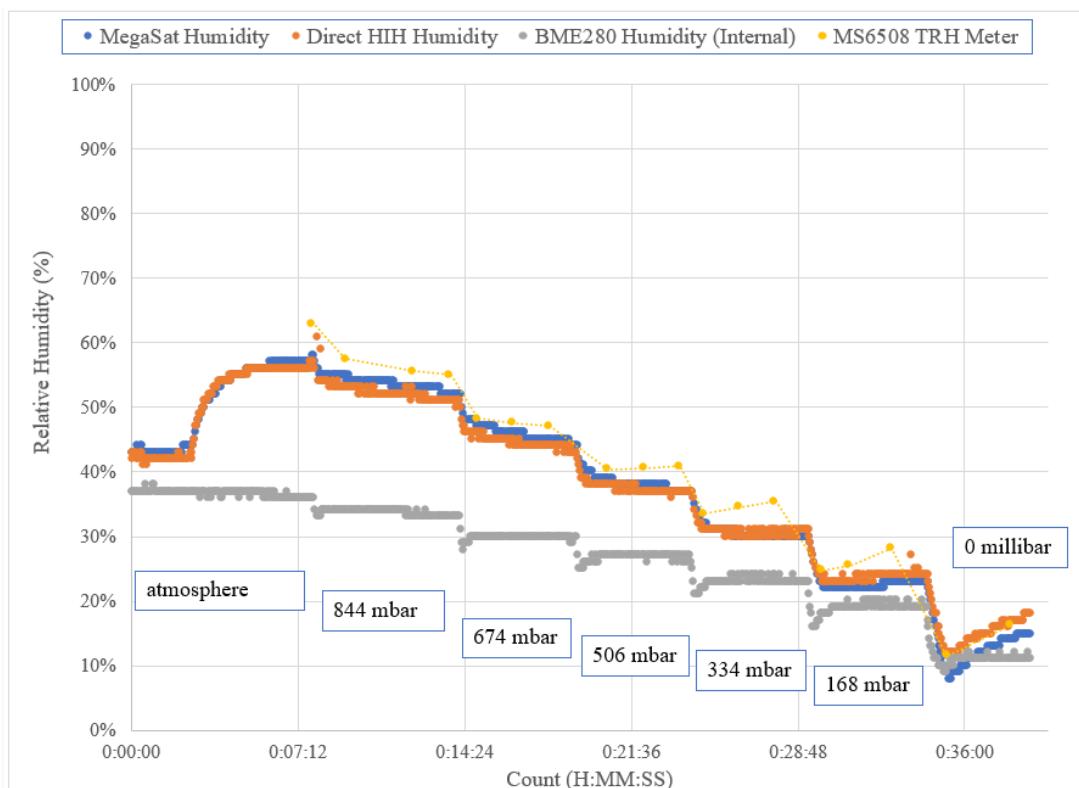


Figure 40: The % Relative Humidity Trends of the Different Humidity Sensors during the Vacuum System Testing

7.0 Mission Operations

7.1 Pre-Launch Requirements and Operations

7.1.1 Calibrations

Calibrations for the analog sensors have been completed in the weeks prior to launching. Due to COVID-19, some calibrations were completed in the lab, and the remaining calibrations were completed at the faculty advisor's house since she had the payload and supplies needed to perform these calibrations. The team had the unfortunate issue that many of the members live in different towns several hours apart and some team members moved to full-time employment as essential workers since the COVID-19 shutdown occurred. Therefore, the calibration procedures and data were discussed between the team members and the faculty advisor during weekly briefing via Zoom. Also, the data was shared on the Microsoft Teams platform so that the team could help analyze the data and integrate the calibration results into the Arduino IDE programming.

7.1.1.1 Calibration Procedures

Laboratory Calibration Procedure

- ✓ Plug in Arduino Mega to laptop with USB-A to USB-B
- ✓ Upload and verify Arduino Mega program for sensors being used
- ✓ Disconnect laptop
- ✓ Confirm empty microSD card is in SD slot of payload
- ✓ Place payload in foam box
- ✓ Check all electrical connections are secure
- ✓ Replace Battery Pack with new, fully charged pack
- ✓ Flip toggle switch to turn on payload
- ✓ Check power connections with multimeter
- ✓ Take 1 minute of data in steady environment where GPS fix can be made
- ✓ Perform calibration procedure
- ✓ Turn off payload by flipping toggle switch
- ✓ Disconnect battery pack
- ✓ Download data from microSD card onto laptop, then empty microSD card
- ✓ Insert empty microSD card in SD slot of payload
- ✓ Secure payload in foam box until flight integration

7.1.1.2 Calibration Results

7.1.1.2.1 External Temperature Sensor

Figure 41 was created using the data from the calibration experiment for the external temperature sensor (see Section 7.1.1). The correlation between the 10-bit ADC values and temperature was calculated using a linear regression and found to have an R^2 value of 0.9976. Moreover, the data was used to determine decimal to degrees Celsius sensitivity. This was calculated to be 5 decimals per 1°C , which means that the sensitivity is 1/5 of a degree Celsius (0.2°C). The formula from the calibration curve was inserted into the MegaSat programming and external temperature measurements were collected for the payload.

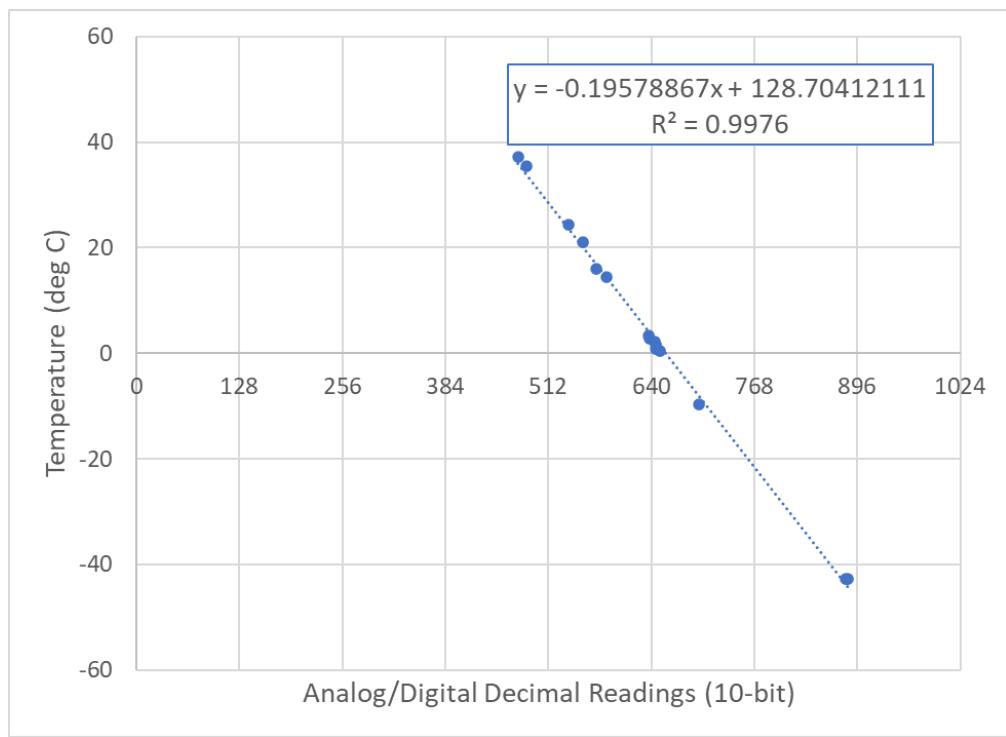


Figure 41. Calibration Results for External Temperature Sensor Compared With Type-K Thermocouples ($n=2$) R19: 0.6K (Offset) and R20: 10.95K (Gain)

7.1.1.2.2 Internal Temperature Sensor

Figure 42 was created using the data from the calibration experiment for the internal temperature sensor (see Section 7.1.1). The correlation between the 10-bit ADC values and temperature was calculated using a linear regression and found to have an R^2 value of 0.9997. Moreover, the data was used to determine decimal to degrees Celsius sensitivity. This was calculated to be 5 decimals per 1°C , which means that the sensitivity is 1/5 of a degree Celsius (0.2°C). The formula from the calibration curve was inserted into the MegaSat programming and internal temperature measurements were collected for the payload.

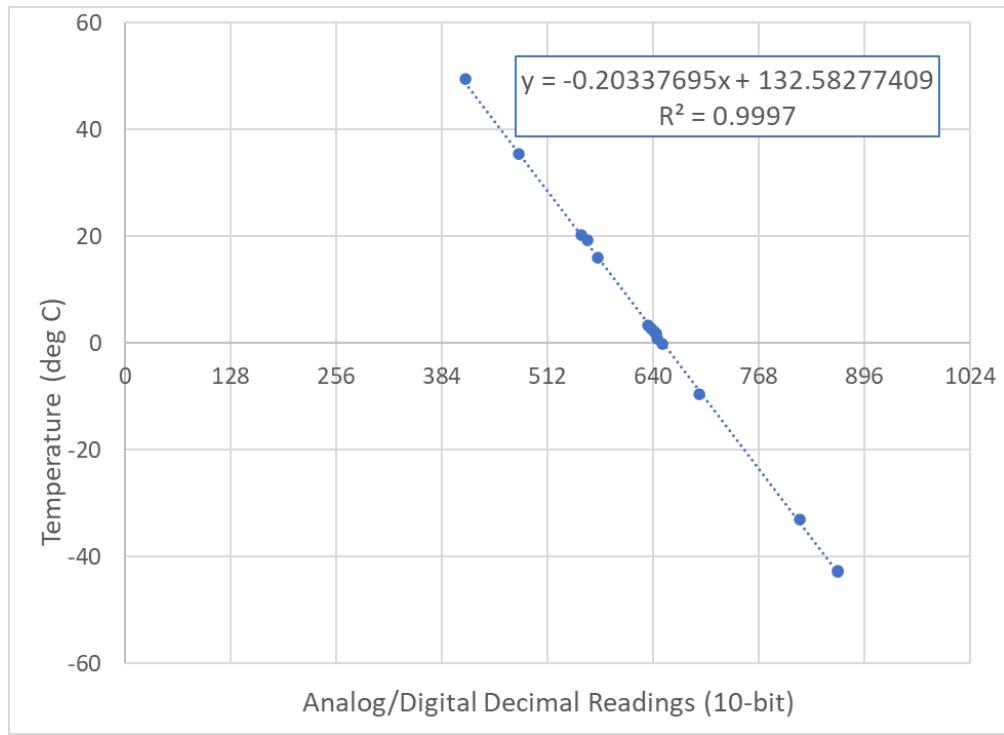


Figure 42. Calibration Results for Internal Temperature Sensor Compared with Type-K Thermocouples ($n=2$) where $R13: 0.49K$ (Offset) and $R14: 10.83K$ (Gain)

7.1.1.2.3 Humidity Sensor on MegaSat

Various saturated salt solutions were used in calibrating the humidity sensor (see section 6.1.3). The salts were wetted in glass jars and allowed to reach thermal equilibrium. The humidity sensor was then placed in the air above and remained until a stable reading was obtained. The sensor recorded data to the microSD card for the duration of this experiment, and humidity readings were taken with a VLIKE MS6508 Digital Temperature Humidity Meter placed inside the jar.

The data was used in calibrating the humidity sensor to convert the 10-bit ADC readings into % relative humidity. The % relative humidity readings were plotted against the corresponding ADC values, with the graph shown in Figure 43. The correlation was calculated using a linear regression and found to have a R^2 value of 0.9977. The sensor only recorded values from 0-80% relative humidity, inhibiting the ability to record the full range of 0-100% needed to meet our science requirements. The team determined that an additional sensor was needed to obtain the remaining values as discussed in section 6.1.3 and in the following section. The formula from the calibration curve was inserted into the MegaSat programming and decimal relative humidity measurements were collected for the payload. The code also reported “999.99” error if the RH measurement yields a greater decimal than 1024 (10-bit). This error meant that the humidity was higher than 80%.

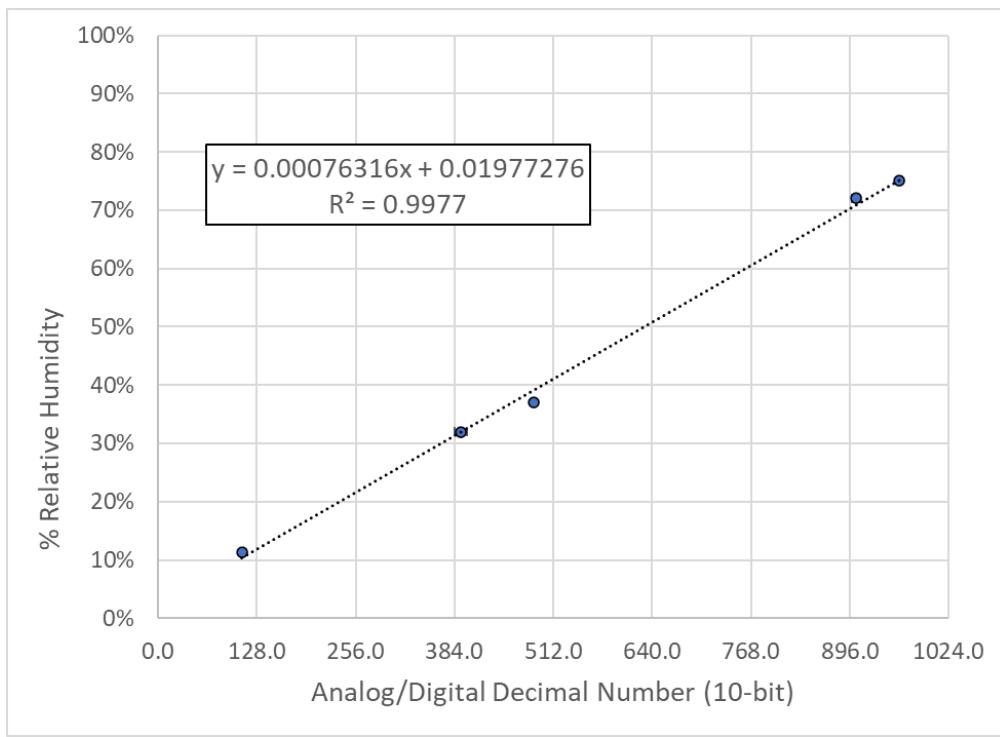


Figure 43. Calibration Results for MegaSat Humidity Sensor in Saturated Salt Solutions ($n=3$, error bars negligible)

7.1.1.2.4 Extra Humidity Sensor

Since the MegaSat humidity sensing system was unable to detect the full range of relative humidity, thus not meeting the DemonSats-1 science humidity requirement, an additional humidity sensor was added to the payload. It was tested with the other humidity sensor simultaneously.

Various saturated salt solutions were used in calibrating the humidity sensor (see Section 6.1.3). The salts were wetted in glass jars and allowed to reach thermal equilibrium. The humidity sensor was then placed in the air above and remained until a stable reading was obtained. The sensor recorded data to the microSD card for the duration of this experiment, and humidity readings were taken with a VLIKE MS6508 Digital Temperature Humidity Meter placed inside the jar.

This data was used in calibrating the humidity sensor to convert the 10-bit ADC readings into % relative humidity. The % relative humidity readings were plotted against the corresponding ADC values, with the graph shown in Figure 44. The correlation was calculated using a linear regression and found to have a R^2 value of 0.9909. Though this sensor showed less sensitivity than the initial sensor, the full span of 0-100% relative humidity was detected.

The formula from the calibration curve was inserted into the MegaSat programming so that the 10-bit ADC values will be converted to % relative humidity during flight. The data from the two sensors was combined during data analysis, with the initial sensor covering the 0-80% range and this sensor for 80-100% to provide the full span of values needed.

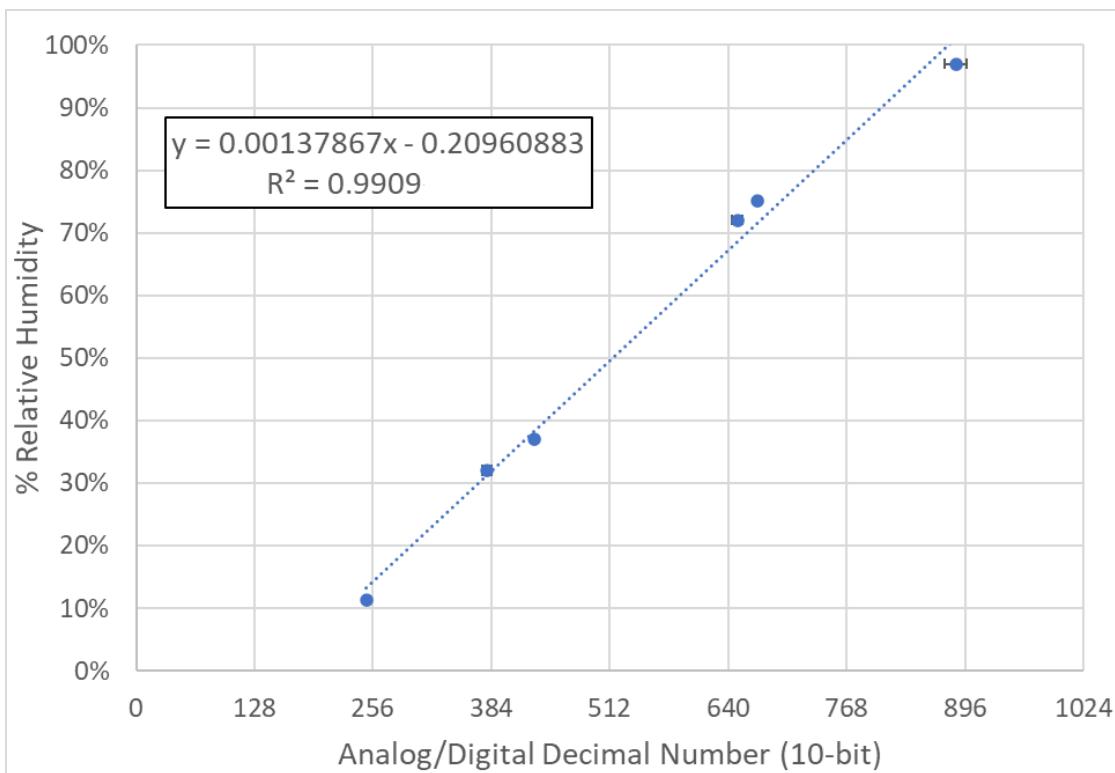


Figure 44. Calibration Results for Additional Humidity Sensor in Saturated Salt Solutions ($n=3$, error bars in standard deviation)

7.1.1.2.5 MegaSat Pressure Sensor

A vacuum chamber was used to calibrate pressure. The system test first measured atmospheric pressure. As the chamber was evacuated, the vacuum was increased in increments of 5 in Hg over the range of -10 to -30 in Hg. The pressure sensor data was recorded to the microSD during the test, and the pressure reading on the oil-filled Bourdon gauge atop the chamber was manually recorded at these intervals (See section 6.1.2 for more information).

This data was used to calibrate the pressure sensor so that the 10-bit ADC readings could be converted into the more meaningful millibar readings. The data from the Bourdon tube was converted from in Hg to millibars (See Table 18 for conversion factors), and the millibar values were plotted against the corresponding 10-bit ADC values. The resulting graph is shown in Figure 45. The correlation between the 10-bit ADC values and pressure was calculated using a linear regression and found to have an R^2 value of 0.9993. Since the Bourdon-tube gauge has an uncertainty of $\pm 5\%$ f. s., the accuracy of the readings may vary by 50 millibars. The formula from the calibration curve was inserted into the MegaSat programming so that the 10-bit ADC values were converted to millibars during flight.

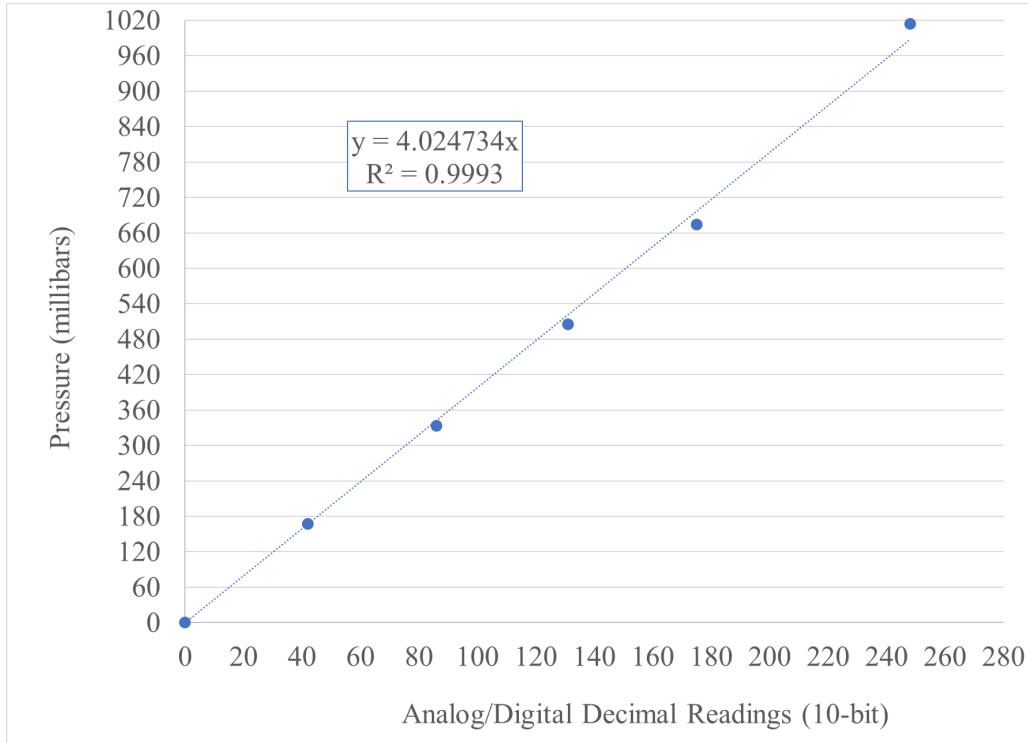


Figure 45. Calibration Results for MegaSat Pressure Sensor Compared to Oil-Filled Bourdon Tube Gauge ($n=2$, error bars negligible)

7.1.2 Pre-Launch Checklist

For LaACES Team Deployment

- ✓ Remove lid
- ✓ Confirm microSD card in SD slot of payload through viewing window next to battery
- ✓ Turn toggle switch to on position (I)
- ✓ Verify that lights are on and blinking through viewing window next to battery
- ✓ Replace and secure payload lid

7.2 Flight Requirements, Operations and Recovery

The payload flight should take approximately two hours. We expect that the payload will be active for about four hours from powering it up until recovery and decommissioning. During the flight, all data is expected to be stored on the microSD card. No telemetry or external data will be transmitted from our payload. The LaACES team will oversee the launch, tracking, and recovery of the payload. The data stored on the microSD card will be the most vital part of the payload to recover.

7.3 Data Acquisition and Analysis Plan

7.3.1 Ground Software

No telemetry or external data will be transmitted from our payload during flight. The LaACES team will oversee the launch, tracking, and recovery of the payload likely without the team, therefore the only ground software for the DemonSats-1 payload will be the data analysis completed in Microsoft Excel on data stored on the microSD card during the flight.

Calibrations have been performed and verified for each sensor prior to flight, and the equation from the resulting calibration curves have been input into the MegaSat to convert the decimal numbers into proper sensor readings that are stored on the microSD card (see Section 7.3.1 for more details). Once the data from the microSD has been received, the data will be then imported into Microsoft Excel into 37 columns and then analyzed. Figures 46 and 47 provide the details on the data analysis process in Excel.

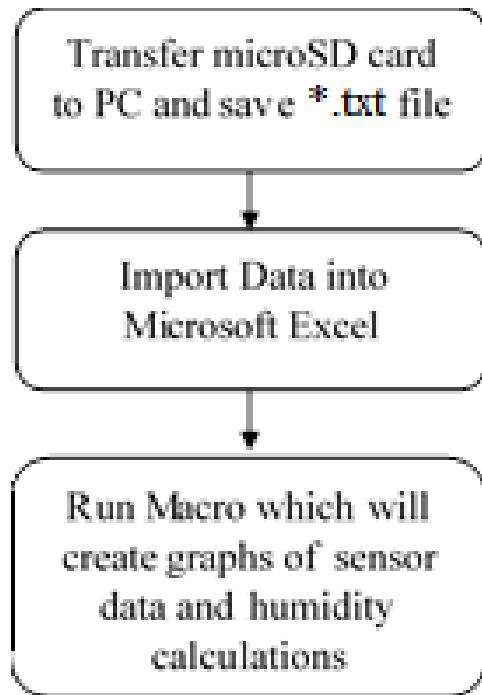


Figure 46: Post-Flight Data Analysis Flowchart

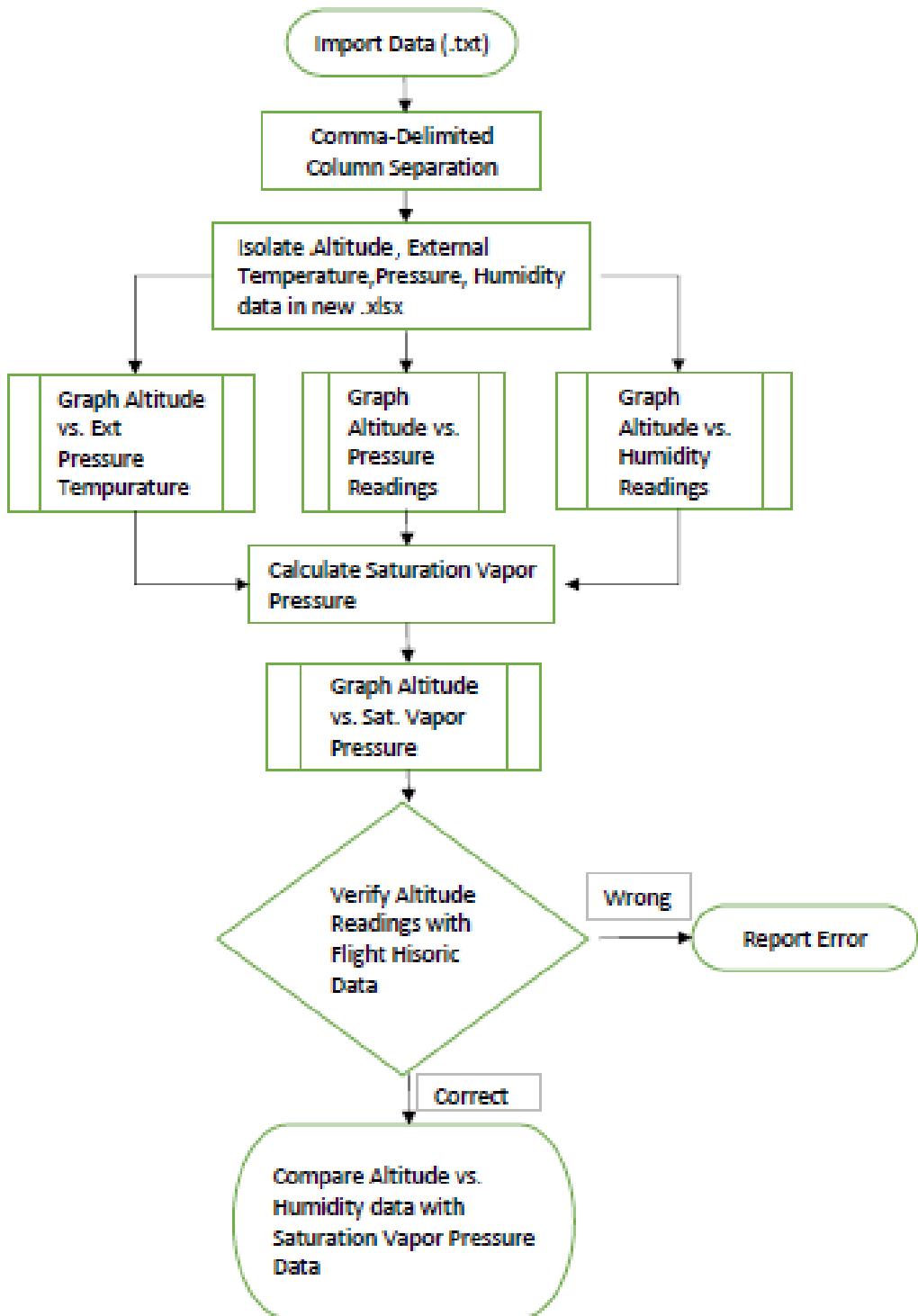


Figure 47: Post-flight Software Plan for DemonSats-1 Payload

7.3.2 Ground Software Implementation and Verification

The only ground software for the DemonSats-1 payload is the data analysis completed in Microsoft Excel on data stored on the microSD card during the flight. During implementation of the Microsoft Excel data analysis plan, the data analysis manager has worked with the faculty advisor to learn how to efficiently handle large datasets, advanced graphing features, and recording VB macros. The data analysis plan has been verified through practicing with example text files with the same format as the expected data from the flight.

7.3.3 Data Analysis Plan

The only ground software for the DemonSats-1 payload is the data analysis completed in Microsoft Excel on data stored on the microSD card during the flight. Once the data from the microSD has been received, the comma delimited data is imported into Microsoft Excel into 37 columns. The altitude, temperature, humidity, and pressure data from all of the payload sensors will be placed on independent worksheets initially and then graphed versus time to observe the flight trends and any possible sensor issues detectable by the data output. Then, the data will be analyzed following the flowchart in Figure 47.

8.0 Project Management

The DemonSats will meet at least once a week on Wednesday afternoons to collaborate between team members and the faculty advisor and to make progress on the payload development. Microsoft Teams is being used as the main communication (via blogs/email/video calls) and file sharing platform. MS Project is being utilized to schedule tasks and to develop a Gantt Chart (See Table 15 on page 41).

8.1 Organization and Responsibilities

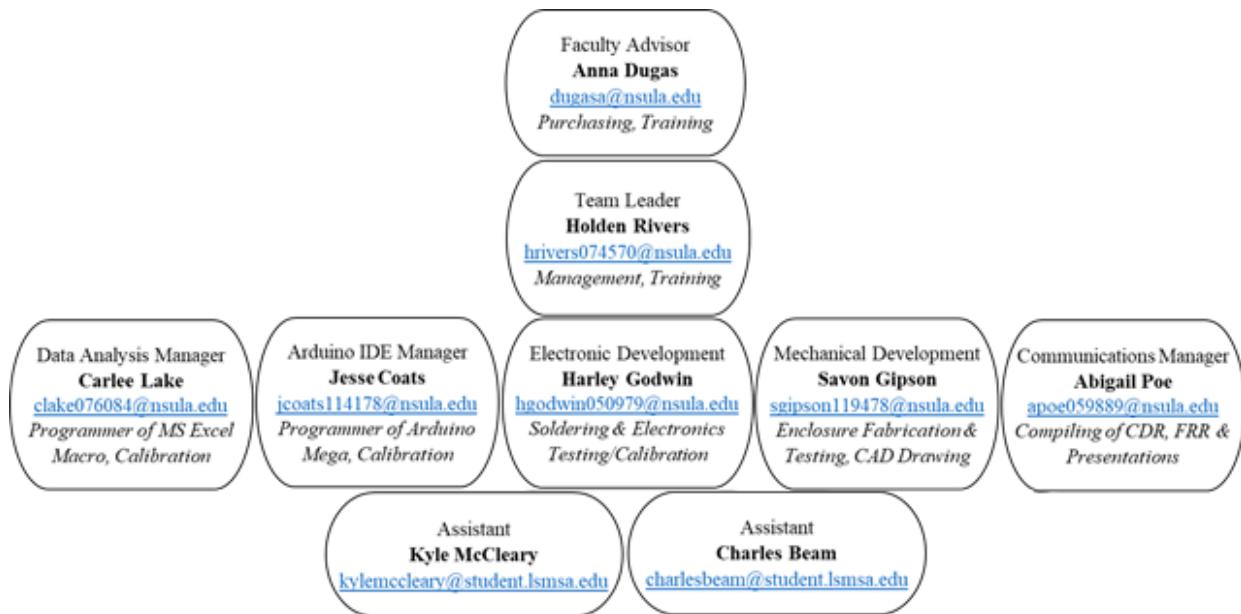


Figure 48: Organization Chart

8.2 Configuration Management Plan

Major changes were first proposed to the faculty advisor for consideration. Minor changes were first proposed to the team leader, who decided if the changes were to be further considered by the faculty advisor. Each team manager (See Figure 48) was responsible for logging changes, issues, and/or progress notes into their designated MS Team “Meeting Notes” section. The faculty advisor met with the team weekly on Wednesday afternoons for in-person progress reports until March 11, 2020 and then weekly Friday afternoons on Zoom until project termination.

8.2.1 Adjustments to Management Plan After COVID-19 Shutdown

After March 13, the NSU campus were closed for in-person classes and were transitioned to online courses for the rest of the spring term and the summer term. The DemonSats continued to meet through Zoom for weekly group meeting with the faculty advisor. Individual

subcommittees met with faculty advisor via Zoom and Teams on an as needed basis. Fortunately, the MegaSat hardware had been completed and tested for electronic functionality. Since the faculty advisor had more access to the testing equipment, she had the possession of the completed MegaSat hardware, and she assisted the students by collecting some of the testing/calibration data for the sensors at her home. She then sent the data to the data analysis manager to determine the calibration curves to be used in the Mega programming. The Arduino IDE programming subgroup was sent home with an Arduino Mega and GPS/microSD module to finalize the GPS parsing and data storage programming. The programmers compiled the main Arduino IDE flight software program at home. Then the programmers worked via Zoom with the faculty advisor to test the flight software functionality on the MegaSat payload. The faculty advisor also worked with the mechanical design team manager to complete a foam box.

8.2.2 COVID-19 Accommodations

8.2.2.1 FRR Presenting Author

The presenting author for the FRR will be Abigail Poe, the communications manager for this project. In the event of an emergency or other issue that prevents her from attending, the faculty advisor Anna Dugas has been designated as the alternate presenting author.

8.2.2.2 Pre-Flight Procedures

- ✓ Remove lid
- ✓ Confirm microSD card in SD slot of payload through viewing window next to battery
- ✓ Turn toggle switch to on position (I)
- ✓ Verify that lights are on and blinking through viewing window next to battery
- ✓ Replace and secure payload lid

8.2.2.3 Payload Delivery

The payload will be hand-delivered by Abigail Poe on June 25, 2020. Prior to delivery, the payload will be secured in a shipping box with extra protection around the payload and clearly labeled with the team name, university name, and contact information for the faculty advisor, Anna Dugas. The payload will have the batteries preinstalled and will be ready to launch following the instructions provided in the previous section.

The faculty advisor Anna Dugas will transfer the box to Abigail Poe at NSU campus on June 24, 2020. Abigail will then drive from Natchitoches, LA, to the LSU Visitor Center on June 25, 2020, using the route depicted in the map of Figure 49. The estimated total travel time is 3 hours, which is also shown in this map as well as Figure 50. When she is approximately ten minutes from the LSU Visitor Center, she will call the number provided to obtain further instructions from an LaACES representative. She will meet this representative to transfer the shipping box with the payload and will ensure that she wears a mask and follows guidelines for social distancing and any other specifications indicated when she speaks to the representative.

The contact information for Abigail Poe is 816-812-8730.

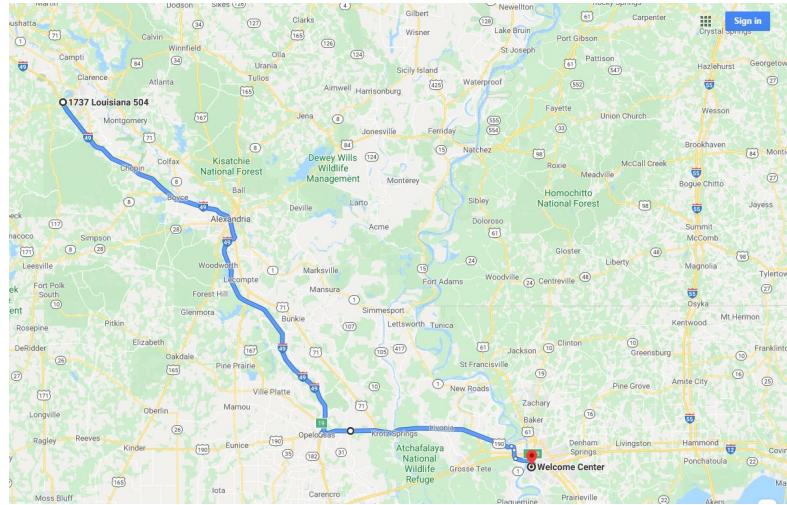


Figure 49. Google Map of the Route to be Taken during Payload Delivery with Approximate Start and End Times and Total Duration Shown

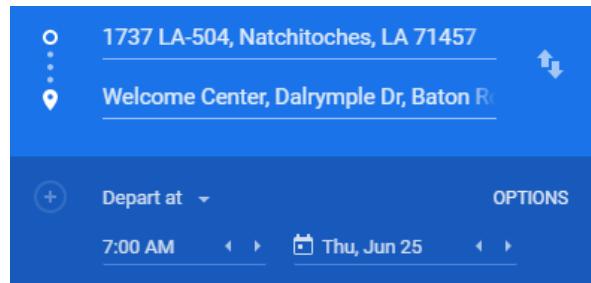


Figure 50. Close Up of the Approximate Start and End Times and Total Duration of the Trip

8.2.2.4 Flight Launch Attendee List

If team members will be allowed at the launch, the following members have been designated as representatives to attend it. These members, along with their contact information and assigned roles, are shown in Table 23.

Table 22. List of Designated Team Representative for Limited Attendance Launch

Priority Order	Team Member	Cell Phone Number	Proposed Flight Operations Role
1	Abigail Poe	816-812-8730	She serves as the communications manager for this team. Therefore, she will assist with documenting the launch, post-flight data analysis, and preparation and delivery of the science report presentation.
2	Holden Rivers	318-315-2266	He serves as the team leader. He will assist with resolving any issues that may arise prior to the launch and post-flight data analysis.
3	Anna Dugas	225-324-5705	She is the faculty advisor for this project. She will assist with resolving any issues that may arise prior to the launch and post-flight data analysis in preparation for the science report presentation.

8.2.2.5 Recovery Waiver Request

The team does not request a recovery waiver for this project.

8.2.2.6 Post Recovery Procedures

- ✓ Remove lid
- ✓ Confirm microSD card in SD slot of payload through viewing window next to battery
- ✓ Turn toggle switch to off position (o)
- ✓ Remove foam support on top of electronics, battery, and foam by battery to remove microSD
- ✓ Please email File 1.txt file to dugasa@nsula.edu
- ✓ Replace microSD card, foam and battery, and secure payload lid

8.2.2.7 Payload Return Plan

If the payload is recovered, the team asks that it be returned to them by shipping it to the address below. The box used to deliver the payload will be labeled with this address prior to delivery at LSU so that it can be used to mail the payload. The payload can be placed in the box and secured in the shipping material originally surrounding it. Afterwards, the box can be sealed and mailed.

The address to which this box shall be mailed is

Anna Dugas
119 Creston Ln
Shreveport, LA 71106

8.2.2.8 Science Report Presenting Author

The presenting author for the FRR will be Abigail Poe. In the event of an emergency or other issue that prevents her from attending, the faculty advisor Anna Dugas has been designated as the alternate presenting author.

8.3 Interface Control

The Arduino IDE will be used to interface with the Arduino Mega 2560 microcontroller, and the Mega will interface with all the sensors and data storage in the payload. The serial monitor feature in Arduino IDE will be used to verify programming and sensor calibration results. The serial monitor will also show the text string that will be sent to the microSD card.

During the time of flight, the payload interfaces will be controlled only by the Arduino Mega. There will not be direct transceiver communication with the payload while in-flight. Once the payload has landed, the microSD card will be removed and transferred to a PC for data analysis in Microsoft Excel.

9.0 Master Schedule

9.1 Work Breakdown Structure (WBS)

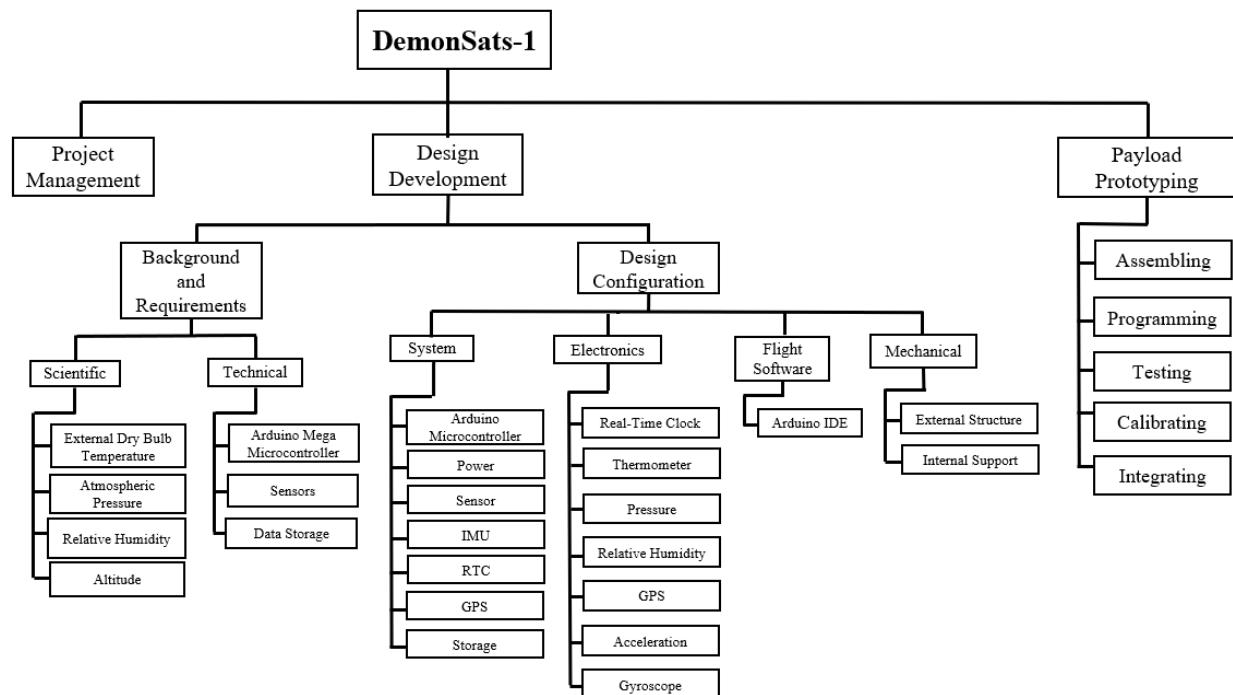


Figure 51: Work Breakdown Structure for the DemonSats-1 Project

9.2 Staffing Plan

Table 23: Staffing Plan for DemonSats-1 Project

Team Member	Major	Previous Experience “Specialty”	Primary Interest in Project	Secondary Interest in Project	Assigned Role on Team
Holden Rivers	IET	Problem Solving, Communication	Arduino IDE Programming	Electronic Development	Team Leader
Jesse Coats	LA /EET	Learning New Things	Arduino IDE Programming	Electronic Development	Arduino IDE Manager
Harley Godwin	EET	Electronic Troubleshooting, Circuit Building	Electronic Development	Arduino IDE Programming	Electronic Development
Carlee Lake	Biology	Writing, Presentations, Programming	Data Analysis	Communications	Data Analysis Manager
Abigail Poe	Biology	Writing, Presentations, Data Analysis, Experimental Design	Communications	Data Analysis	Communications Manager
Savon Gipson	EET	CAD, Building Things	Arduino IDE Programming	Electronic Development	Mechanical Development
Charles Beam	HS	Programming	Arduino IDE Programming	Electronic Development	Assistant – Programming and Electronics
Kyle McCleary	HS	Programming, Soldering, Wiring	Arduino IDE Programming	Electronic Development	Assistant – Programming and Electronics

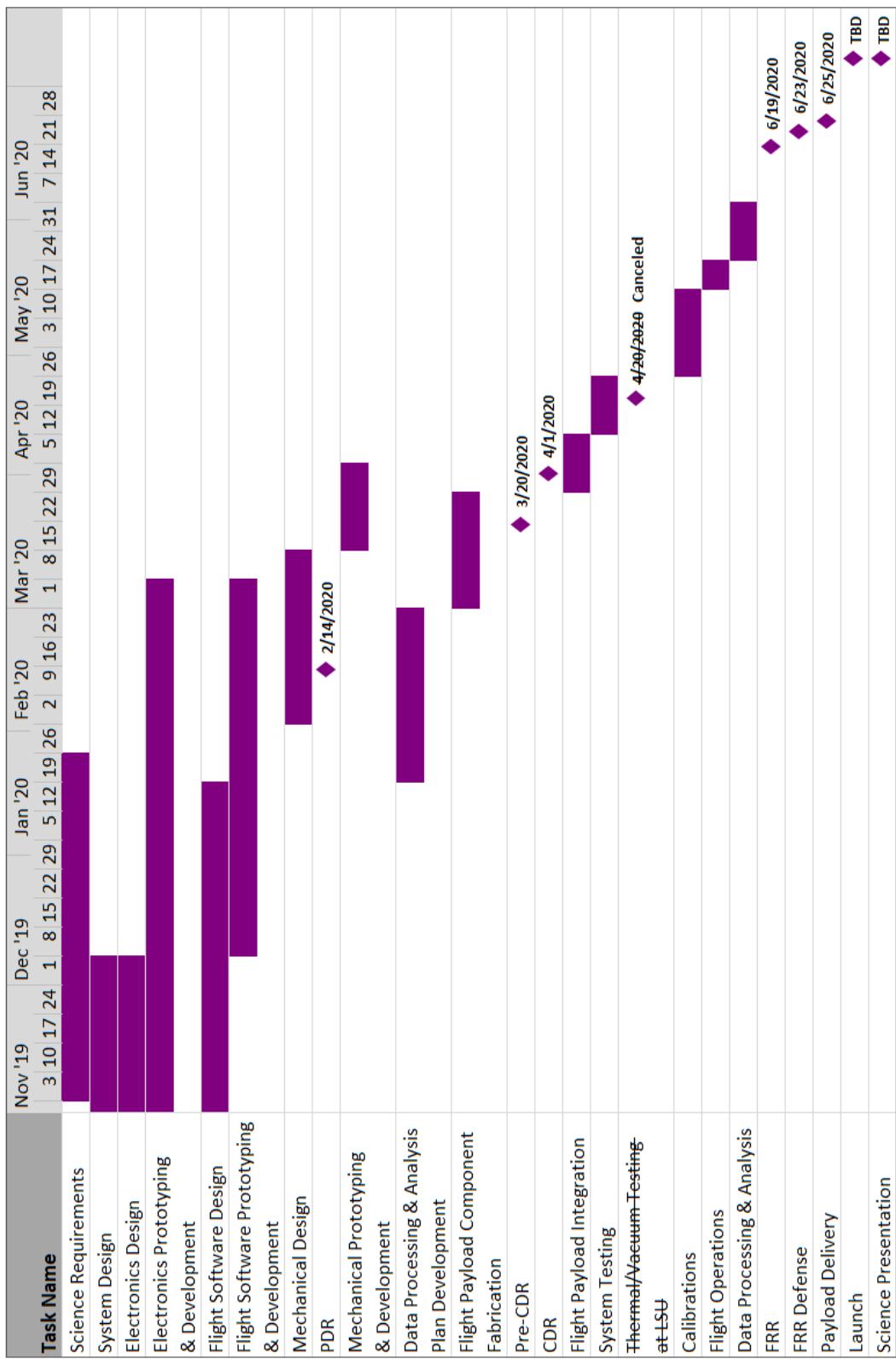
See Section 12.0 for abbreviations

9.3 Timeline and Milestones

Table 24: Major Milestones

PDR	February 14, 2020
Pre-CDR	March 20, 2020
CDR	April 1, 2020
Thermal/Vacuum Testing at LSU	April 19-20, 2020
FRR	June 19, 2020
FRR Defense	June 23, 2020
Launch	TBD
Science Presentation	TBD

Table 25: Gantt Chart for DemonSats-1 Payload Project



10.0 Master Budget

10.1 Expenditure Plan

Table 26: List of Expenditures for the DemonSats-1 Payload

Description	#	Units	Unit Cost	Extended Cost
Stainless Steel Vacuum Degassing Chamber & 3 CFM Pump	1	each	\$179.99	\$179.99
63-37 tin-lead solder, 0.020	1	1 lb.	\$28.80	\$28.80
SparkFun BME280 Atmospheric Sensor Breakout	1	each	\$22.95	\$22.95
PTC Thermistors PTGL07AS5R6K4B51B0	1	10 pack	\$0.59	\$5.91
Lithium Batteries 6 V	8	each	\$5.99	\$47.92
Toggle Power Switch	1	12 pack	\$7.48	\$7.48
16 AWG Red Wiring	1	25' roll	\$5.46	\$5.46
Speed Square Tool	1	each	\$4.03	\$4.03
Formular XPS 4' x 8' 1/2" sheet	1	each	\$12.98	\$12.98
Roll EconoKote Mylar film, white, 6 ft roll	1	6' roll	\$15.96	\$15.96
Roll EconoKote Mylar film, orange, 6 ft roll	1	6' roll	\$13.74	\$13.74
Heat sealing iron for EconoKote	1	each	\$22.99	\$22.99
Heat sealing iron sock for EconoKote	1	each	\$4.60	\$4.60
Bottle Gorilla Glue	1	8 oz bottle	\$10.99	\$10.99
Painter's tape, 1.41" wide	1	roll	\$5.99	\$5.99
Electrical Tape	1	roll	\$3.98	\$3.98
Carpet Tape	1	roll	\$6.99	\$6.99
Plastic shoulder washers (grommets), 1/4" ID	1	40 pack	\$5.40	\$5.40
100% cotton gloves for balloon handling	1	20 pack	\$1.00	\$19.98
SHIPPING				\$18.97
SUBTOTAL				\$445.11
CONTINGENCY	10%			\$44.51
TOTAL				\$489.62

The largest expense was the vacuum chamber and pump which are being used to calibrate the pressure module on the MegaSat shield. The SparkFun BME280 Atmospheric Sensor Breakout costs about \$23 and has been added to ensure low elevation data is collected when the GPS fails to get a fix. The next grouping of expenses pertains to the making the battery packs which will be used to power the payload. Four battery packs have been made, 3 for testing and 1 for the launch. The rest of the expenses listed in Table 16 come from the foam box construction and the gloves to be used to assist in the balloon launch. The total sum including a 10% contingency is \$489.62, which remains below the allotted budget of \$500 for this project.

10.2 Material Acquisition Plan

All the equipment from the expenditure list has been purchased and obtained prior to the CDR submission. Below is the list of locations from where each item has been obtained.

Table 27: Locations of Material Acquisition

Description	Obtained from
Stainless Steel Vacuum Degassing Chamber and 3 CFM Pump	Amazon
63-37 tin-lead solder, 0.020	Amazon
Sparkfun BME280 Atmospheric Sensor Breakout	SparkFun Electronics
PTC Thermistors PTGL07AS5R6K4B51B0	Mouser Electronics
Lithium Batteries 6 V	Amazon
Toggle power switch	Amazon
16 AWG red wiring	Home Depot
Speed Square Tool	Amazon
Formular XPS 4' x 8' 1/2" sheet	Home Depot
Roll EconoKote Mylar film, white, 6 ft roll	Amazon
Roll EconoKote Mylar film, orange, 6 ft roll	Amazon
Heat sealing iron for EconoKote	Amazon
Heat sealing iron sock for EconoKote	Amazon
Bottle Gorilla Glue	Stine's Home & Garden
Painter's tape, 1.41" wide	Stine's Home & Garden
Electrical Tape	Stine's Home & Garden
Carpet Tape	Stine's Home & Garden
Plastic shoulder washers (grommets), 1/4" ID	Grainger
100% cotton gloves for balloon handling	Amazon

11.0 Risk Management and Contingency

Table 28: Risk Event Analysis [Ranked 1 (low) – 5 (high)]

Risk Event	Likelihood	Impact	Detection Difficulty	When
Failed Component	3	4	2	Fabrication / Testing
Battery Pack Freezes	3	5	4	In-Flight
Battery Pack Overheats	3	5	4	Pre-Flight in Summer
Mechanical Damage (Pre-Flight)	2	4	1	All Stages Before Flight
Mechanical Damage (Landing)	2	4	1	End of Flight

Table 28: Risk Event Analysis [Ranked 1 (low) – 5 (high)] continued

MicroSD Card Failure	2	5	2	In-Flight
Loss of Program File	1	3	1	Programming
Unexpected Weather	3	4	1	In-Flight
Payload Falling in Water	1	4	1	End of Flight
A Virus Pandemic Closing Campus	5	5	1	Fabrication/Testing

Table 29: Risk Contingency Plan

Risk Event	Response	Contingency Plan	Trigger	Who is Responsible?
Failed Component	Reduce	Have spare parts & soldering tools available at all stages & on trip	Detection	Anna Dugas
Battery Pack Freezes	Reduce	Test insulation strategy and cold effect on battery prior to flight	Detection	Anna Dugas
Battery Pack Overheats	Reduce	PTC thermistor installed between batteries for stop	Detection	Anna Dugas since COVID
Mechanical Damage (Pre-Flight)	Reduce	Test enclosure stability prior to installing payload circuitry	Detection	Anna Dugas
Mechanical Damage (Landing)	Reduce	Test enclosure stability for impulse reaction, strengthen weak stress points observed in drop testing	Detection	Anna Dugas
MicroSD Card Failure	Reduce	Consider installing additional microSD card writer into system Verify card is in working order during pre-flight	Detection	Anna Dugas
Loss of Program File	Reduce	Save program with understandable title and backup file into MS Teams	Leaving for the day	Anna Dugas
Unexpected Weather	Transfer	Except that we can't control the weather.	Forecast check	LaACES Management
Payload Falling in Water	Reduce	Consider installing conformal coating to circuitry	Ask LaACES Mngt.	Anna Dugas
A Virus Pandemic Closing Campus	Response and Adapt	Host team meeting via online video conferencing, bring equipment home to build, program and test	Gov't decree	Anna Dugas

12.0 Glossary

ABS	Absolute (absolute pressure – starting at perfect vacuum)
CDR	Critical Design Review
CSBF	Columbia Scientific Ballooning Facility
EET	Electrical Engineering Technology
FS	Full Scale
FRR	Flight Readiness Review
GPRMS	Global Positioning Recommended Minimum Coordinates
PGPAA	Global Positioning system fix data
HS	High School
IDE	Integrated Development Environment
IET	Industrial Engineering Technology
LA	Liberal Arts
LaACES	Physics & Aerospace Catalyst Experiences in Research
PDR	Preliminary Design Review
PS	Physical Science
PTC	Positive Temperature Coefficient
RTC	Real-Time Clock
TBD	To Be Determined
TBS	To Be Supplied
VB	Visual Basics
WBS	Work Breakdown Structure