LaACES Program
Flight Readiness Review Document
for the
<u>DemonSats-2</u>
Experiment
by

Team **<u>DemonSats, Northwestern State University & LSMSA</u>**

Prepared by:

Carlee Lake	5-09-2021
Carlee Lake (NSULA, Team Captain)	Date
Lacee Savage	5/08/2021
Lacee Savage (NSULA)	Date
Savon Gipson	5/05/2021
Savon Gipson (NSULA)	Date
Louis Kater West	5/05/2021
Louis West (NSULA)	Date
Charles Beam	5-05-2021
Charles Beam (LSMSA)	Date

Submitted: Carlee Lake

Reviewed & Revised: Charles Beam

Approved:

Anna C. Dugas Digitally signed by Anna C. Dugas Date: 2021.05.10 18:48:35 -05'00'	5-10-2021
Anna Dugas (Team Faculty Mentor)	Date
Christopher Lyles (BPS School Director)	Date
LA SPACE Signoff	Date

Change Information Page

Title: FRR Document for DemonSats-2 Experiment
Date: 05/03/2021

List of Affected Pages		
Page Number	**Issue**	**Date**
3	3.1 Fixed mission goal	3/12/2021
6	3.3.1 Buck's Equations into text with variables	3/19/2021
8	3.3.1 Justify 10 sec readings and 50-71 m distance	3/19/2021
9	3.4 Expand Technical background	3/29/2021
31	4.3.5 Add Power Budget Discussion (Important!!)	3/27/2021
31	4.3.5 Include Uncertainties in the Power Budget	3/27/2021
34	4.4.2 Fix loop software flowchart to be functional	3/28/2021

Status of TBDs

TBD Number	Section	Description	Date Created	Date Resolved
1	3.1	Be more specific with mission goal	3/02/2021	3/12/2021
2	3.3.1	Buck's Equations into text with variables	3/02/2021	3/19/2021
3	3.3.1	Justify 10 sec readings and 50-71 m distance	3/02/2021	3/19/2021
4	3.4	Expand Technical background	3/02/2021	3/29/2021
5	4.3.1.4	Fix Humidity Sensor Filter and Report	3/03/2021	3/27/2021
6	4.3.5	Add Power Budget Discussion (Important!!)	3/02/2021	3/27/2021
7	4.3.5	Include Uncertainties in the Power Budget	3/02/2021	3/27/2021
8	4.4.2	Fix loop software flowchart to be functional	3/02/2021	3/28/2021
9	4.6.2	Address weight budget issues of source, uncertainty, and contingency	4/08/2021	4/27/2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 Document Purpose

This document describes the critical design for the DemonSats-2 experiment by Team DemonSats from Northwestern State University (NSULA) and the Louisiana School for Math, Science, and the Arts (LSMSA) for the LAACES Program. It fulfills part of the LAACES Project requirements for the Critical Design Review (CDR) to be submitted by March 31, 2021.

1.1 Document Scope

This CDR document specifies the scientific and technical purpose, requirements, and payload design plans for the DemonSats-2 experiment. This document specifically 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 CDR document shall only be made after approval by designated representatives from Team DemonSats and the LAACES Institution Representative, Assistant 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, Rev/20AUG2020.
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,* 4 . ed., New York, NY, 2013.
5. Dugas A.and A. Poe, “ACES-65 DemonSat-1 Flight Data”, August 2020.
6. 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
7. Engineering ToolBox, (2004). *Relative Humidity in Air*. [Online] Available at: https://www.engineeringtoolbox.com/relative-humidity-air-d_687.html
8. Rivers, H., Poe, A., Coates, J., Gipson, S., Godwin, H., Lake, C., Beam, C., Dugas, A. “LaACES Program Flight Readiness Review Document for the DemonSats-1 Experiment.” 6/19/2020.
9. “Past weather in Lafayette (and Lake Charles).” August 3, 2020. Available: https://www.timeanddate.com/weather/usa/lafayette/historic https://www.timeanddate.com/weather/usa/lake-charles/historic
10. ACES-65 Flight Information. Available: https://laspace.lsu.edu/laaces/flightinfo/?fname=ACES-65
11. LaACES Student Ballooning Course Information. Available: https://laspace.lsu.edu/laaces/student-balloon-course/
12. Adams, C. “The MegaSat Project: High-Level/ Mid-Level Overview” presented on 03/21/2019.
13. “BU-502: Discharging at High and Low Temperatures.” Available: https://batteryuniversity.com/learn/article/discharging_at_high_and_low_temperatures
14. “DS3231: Extremely Accurate I2C-Integrated RTC/TCXO/Crystal” *Maxim Integrated.* Datasheet 19-5170, Rev. 10, 03/2015.
15. “1N457/A” *Fairchild Semiconductor.* Datasheet 1N457/A, Rev. A, 2002.
16. “1230: UltraStable Specifications.” *TE connectivity.* Datasheet 09/2017.
17. “HIH-4000 Series: Humidity Sensors.” *Honeywell.* Datasheet 00917-5-EN. 2010.
18. 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.
19. Delano. “FGPMMOPA6H GPS Standalone Module Data Sheet” *GlobalTop Technology, Inc.* Rev. V0A. 01/31/2012.
20. “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
21. Adams, C. and J.C. Collins. “MegaSat_Full_Schematic_PDF” Revision 2.3. 02/18/2019
22. Lady Ada, “Adafruit Ultimate GPS Logger Shield,” *Adafruit Industries,* last updated 10-20-2016.
23. Energizer, “Lithium (2CR5) Product Datasheet,” *Energizer Holding, Inc.* form# EBC-8821EU-B.
24. Foamular PROPINK IS Extruded Polystyrene (XPS) Rigid Foam Insulation. *Owens Corning.* Product Data Sheet. Pub. No. 57862-G. July 2018.
25. “Constructing a Standard LaACES Payload Box” *LaACES Student Ballooning Course.* L29.04, LSU 08/20/2020.
26. 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 second LaACES project of NSULA/LSMSA collaboration, Team DemonSats plans to launch a payload by atmospheric sounding balloon to 100,000 ft (30,480 m) to explore saturated vapor pressure as a function of altitude and by constructing a MegaSat payload to measure atmospheric temperature, pressure, % relative humidity, and altitude.

3.2 Objectives

The team shall construct and calibrate the DemonSats-2 payload modeled from the LaACES MegaSat payload. [1] The completed payload shall be configured to be attached to the LaACES balloon system and launched in May 2021. The payload shall 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 DemonSats-2 payload modeled from the LSU MegaSat payload, which uses 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 for 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, or absolute humidity, is a measurement of the amount of moisture in the air. Quantifying the amount of moisture in the air is very difficult to measure directly with instrumentation, so the humidity of an environment is often measured as 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]

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

The equilibrium vapor pressure is typically 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, [3,4] is more accurate when determining it as a function of both temperature and pressure (See Figures 2a,2b).

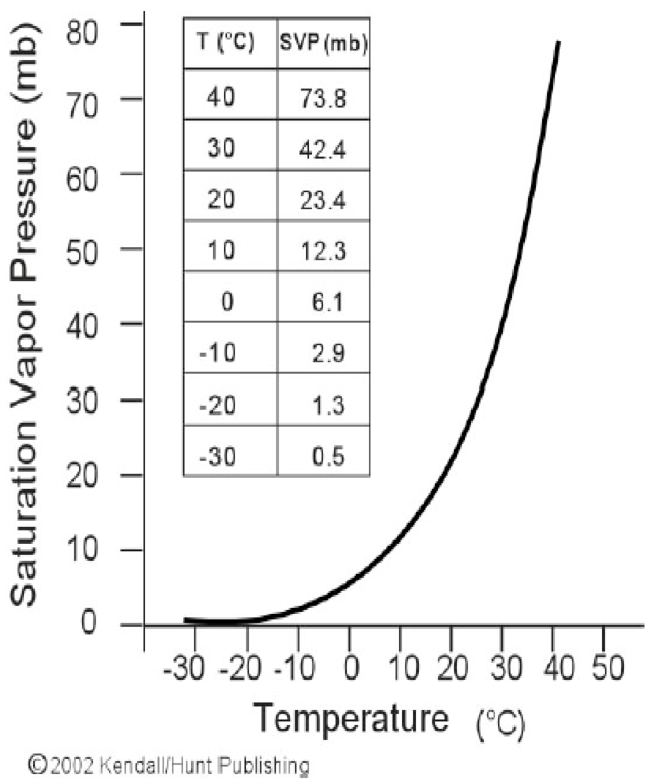


Figure 1: Saturation Vapor Pressure as a Function of Temperature [3]

Last year, the DemonSats group tasked themselves to learn and utilize the Arden Buck Method [2] for calculating the equilibrium vapor pressure during their LaACES experiment. The DemonSats-1 payload was launched on August 3rd, 2020 in southern Louisiana. Temperature and pressure data was successfully obtained (See Figure 3a, 3b). The determined vapor pressure using Buck's Method is shown in Figure 5. However, the humidity data received from the DemonSats-1 was compromised by condensation for the sensors that were accurate on the ground level, but there was an expected variability in the HIH-4000-003 sensors during the ACES-65 flight especially when the payload passed the tropopause. (See Figure 6). Therefore, this year's team plans on revisiting this experiment to verify that the relative humidity could compared to saturation vapor pressure determined by Buck's method to find the actual vapor pressure during flight versus altitude.

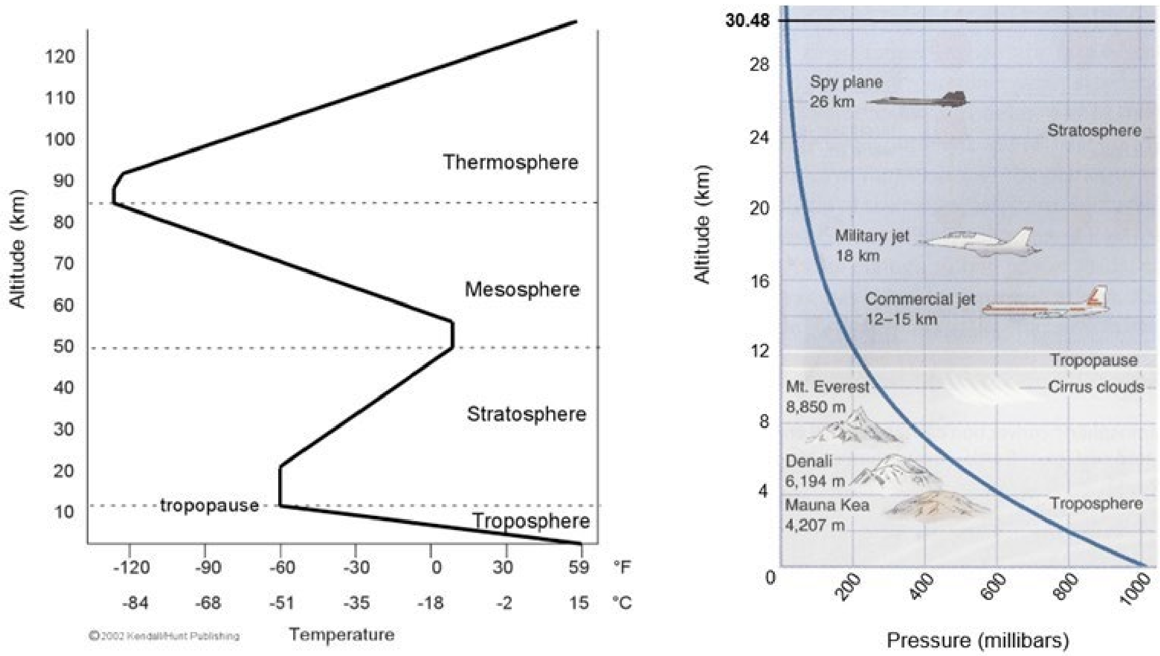


Figure 2: Atmospheric Temperatures (a) and Pressures (b) for Altitudes above the Earth's Sea Level. [3,4]

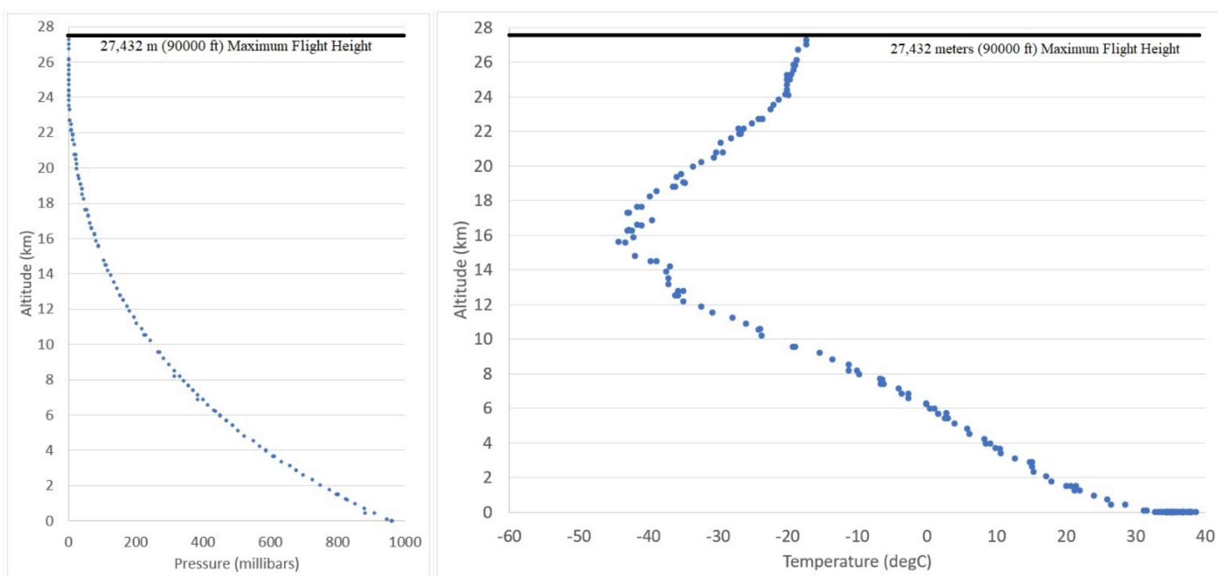


Figure 3: (a) DemonSats-1 Flight Temperature Data (b) DemonSats-1 Flight Pressure Data From the ACES-65 Flight Which Occurred on August 3, 2020.[5]

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

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

and

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

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.

Several equation coefficients are generated from the Buck et al. paper to determine the best fit saturated vapor pressure model to be used over the expected temperature and pressure ranges. The saturation vapor 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. The saturation vapor pressure equations then can be expanded to the following form given the model selections:

$$e'_w = \left[1 * f_{w3}(a) + \left((f_{w3}(b)) * P\right)\right] * e_{w2}(a) * e^{\left[\frac{e_{w2}(b)*T}{e_{w2}(c)+T}\right]} \quad \text{for 0 to 50°C} \quad (Eqn.\ 4)$$

$$e'_i = \left[1 * f_{i3}(a) + \left((f_{i3}(b)) * P\right)\right] * e_{i1}(a) * e^{\left[\frac{e_{i1}(b)*T}{e_{i1}(c)+T}\right]} \quad \text{for -80 to 0 °C} \quad (Eqn.\ 5)$$ [2]

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

Table 1 shows the values of the coefficients in the above equation 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 range errors mainly occur outside the temperatures in which our experiment shall be exposed to.

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}	$7*10^{-4}$	$3.46*10^{-6}$	--	0 to 50	--
f_{i3}	$3*10^{-4}$	$4.18*10^{-6}$	--	-80 to 0	--

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

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

$$e'_i = \left[1.0003 + \left((4.18 * 10^{-6}) * P\right)\right] * 6.1115 * e^{\left[\frac{22.542*T}{273.48+T}\right]} \quad \text{for -80 to 0°C} \quad (Eqn.\ 7)$$

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. Figure 4 shows that expected saturated vapor pressure trend for the flight using expected temperatures and pressures in the above equations with respect to altitude. The Figure 4 trend and magnitude of the saturated vapor pressure are similar to other references. [6, 7] Figure 5 shows the calculated saturated vapor pressure from the flight data obtained on August 3rd, 2020 with DemonSats-1. The trends are similar, and the team is interested to see if there will be a difference in the trends for a springtime launch.

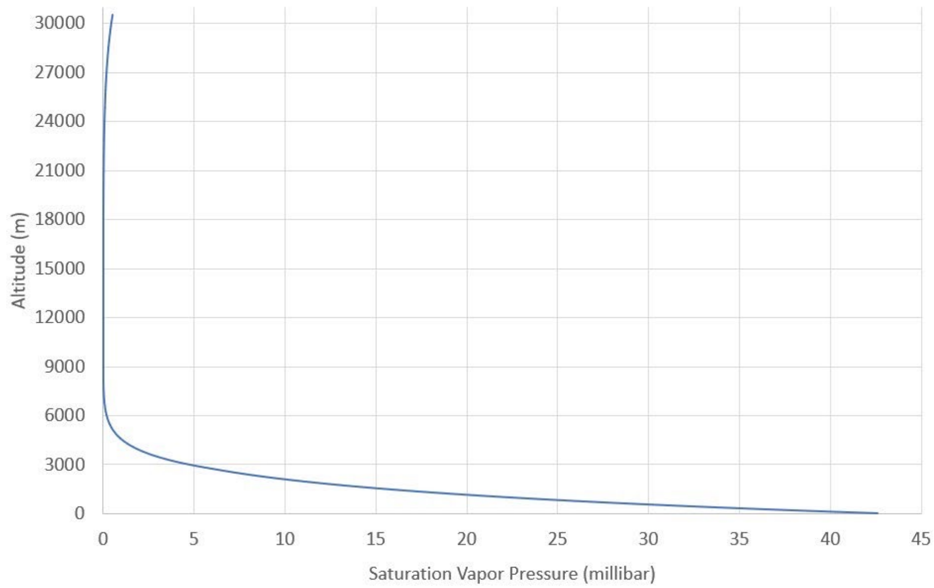


Figure 4: The Expected Saturation Vapor Pressure (in millibars) over the Flight Altitudes Assuming Ground Temperature at 30 °C [8]

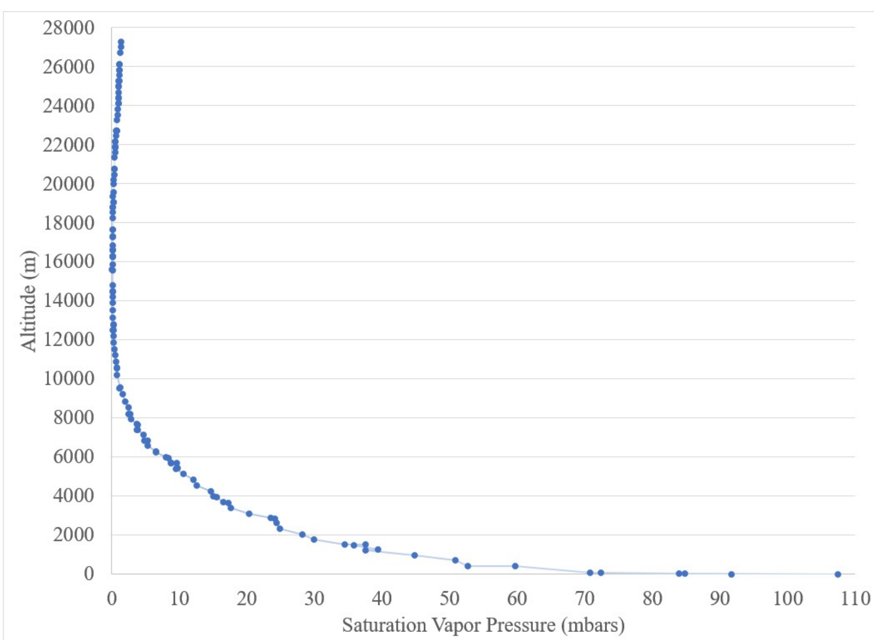


Figure 5: The Saturation Vapor Pressure (in millibars) Calculated from the DemonSats-1 Flight Data and Compared with Altitude (ACES-65, August 3, 2020) [5]

In addition to comparing the saturated vapor pressure with attitude, the team plans to compare the saturated vapor pressure to the relative humidity (%RH) readings to be obtained from the humidity sensor to determine the actual vapor pressure over altitude during flight (See Eqn. 1). For DemonSats-1, this comparison was not done since the humidity data received from flight appears to be compromised by varying levels of condensation for the HIH-4000-003 sensors. The HIH-4000-003 sensor data was highly variable (±27%) near the ground level, and the expected drop in humidity with altitude increase past the tropopause was not detected. The commercial BME-280 sensor data was much more precise, and it did detect the drop in humidity part the tropopause, however the ground level readings were not accurate,[9] likely due to the sensor being housed inside the box on the controller board, where the air was warmed by circuit heating. For DemonSats-2 team will need to explore the precision of the HIH-400-003. The team will also need to find an optimal position for the sensor on the payload to reduce condensation, yet to avoid readings of drier air affected by the circuit heating.

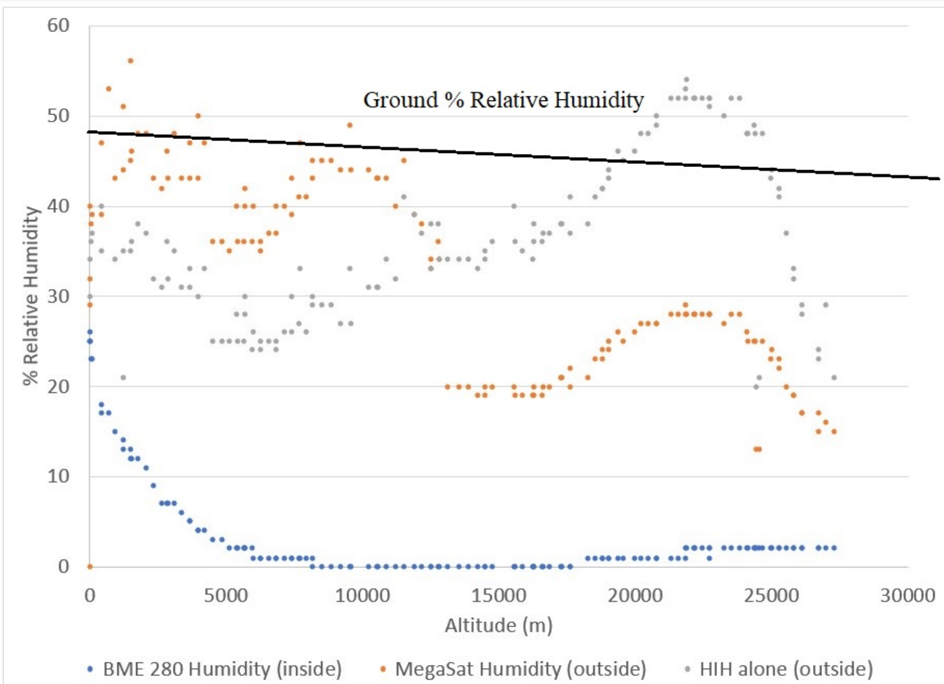


Figure 6: The Comparison of the Humidity Sensors Used in the DemonSats-1 Over Flight Altitudes (ACES-65, August 3, 2020). Ground %RH obtained from weather report.[5,9]

The team will aim to have data values collected at a 1/10 Hz sampling rate or data every 10 seconds. This rate is fast enough to obtain data to achieve the scientific objectives. From the LaACES ACES-65 flight report, the average ascent rate was 1000 ft/min (5.1 m/s) and the average rate of descent was reported to be 1400 ft/min (7.1 m/s). [10] Therefore, the altitude readings at a 1/10 Hz sampling rate should vary between 50 m and 71 m. Data taken from the DemonSats-1 payload was taken at 1/3 Hz, however the altitude data was only recorded when the GPS module received a GGA signal, which occurs sometimes at a 1/50 Hz rate. Since the trends obtained from this data were still informative, less frequent readings than 1/3 Hz and more frequent than 1/50 Hz were determined to be a basis for a requirement. The team decided to select the 1/10 Hz frequency rate, since the team wants to observe the humidity sensitivity changes through the possible clouds during flight.

3.3.2 Science Requirements

Below is the scientific requirements and constraints that must be met by the team for the payload to be able to calculate the saturated vapor and the actual vapor pressure:

- External dry bulb temperature reading shall be obtained (accurate within 5℃, in a range of -70℃ and 30℃). A sampling rate of 10 seconds is designed to collect data for every 50-71 m of change in the altitude of the payload (Refer to the end of Section 3.3.1 for more information of the determination to use a 10 second sampling rate).
- Atmospheric pressure reading shall be obtained (accurate within 5% full scale in a range of 0 absolute mbar to atmosphere), with a sampling rate of 10 seconds designed to collect data for every 50-71 m of change in the altitude of the payload.
- Percent relative humidity reading shall be obtained (accurate within 10%, in a range of 0% - 100% RH), with a sampling rate of 10 seconds designed to collect data for every 50-71 m of change in the altitude of the payload.
- Altitude readings shall be obtained using the GPS module, GGA NMEA sentences for an interval over a range of 0 km to 30.5 km. A sampling rate of 10 seconds shall allow for the collection of data for every 50-71 m of change in the altitude of the payload.

3.4 Technical Background and Requirements

The LaACES MegaSat payload, originally designed at LSU A & M, is the base unit for the DemonSat-2 payload. 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 the DemonSats-2 project will be to build and test the MegaSat payload for the purpose of reliably measuring temperature, pressure, humidity, and altitude. The DemonSats team will start the payload design using the circuit supplies, example IDE code, and the assembly manuals provided by the Physics & Astronomy Department at LSU.[11]

To measure temperature, the LSU MegaSat kit includes two 1N457 temperature diodes to measure internal and external temperature and calibrated in degrees Celsius. The external temperature sensor and signal conditioning circuitry will be used to monitor temperature that will be compared to altitude and used for calculation of the saturation vapor pressure.

To measure pressure for the Buck's Method of calculating saturation vapor pressure, the LSU MegaSat kit includes a 1230-015A-3L piezoresistive diaphragm gauge. The calibrated pressure changes in millibars detected by this sensor will be compared to altitude as well, like the data shown in Figure 3b.

To measure humidity, two HIH-4000-003 relative humidity sensors will be used. One of these sensors is included in the LSU MegaSat kit with signal conditioning circuitry, the other analog sensor will be added to ensure that a full range of relative humidity (0-100%) is received. In the DemonSat-1 experiment, the team discovered that the provided signal conditioning circuitry only gave the sensor output a range of 80% (see Figure 14 on page 21). The team has redesigned the signal conditioning circuit by changing the gain potentiometer from a 10 kOhm to a 100 kOhm. This redesign experiment is explained in Section 4.3.1.4.

Besides the humidity sensor signal conditioning redesign, the team plans to modify the payload container to optimize sensor sensitivity to the outside environment, yet to reduce condensation development on the sensors, especially the humidity sensor. Further details on the different sensors described above, as well as the detailed description of the control system design, and the payload container will be discussed in Section 4.0 Payload Design.

3.4.2 Technical Requirements

The following is a list of requirements that are needed for the payload to collect data successfully:

- Dry bulb temperature readings shall be calibrated to an accuracy within 5°C and a range between -70 to 30 C.
- Calibration of atmospheric pressure readings shall be accurate within 5% f. s. and a range of 0 to 1015 absolute millibars.
- Relative humidity readings shall be calibrated to 10% accuracy at a range of 0-100%.
- Time-stamped altitude, pressure, temperature, and humidity data shall be recorded by including real-time clock (RTC) data.
- Data collected and processed by Arduino Mega shall be organized and stored onto a microSD card for post-flight retrieval.
- Payload shall be enclosed in a box made of ¾" polystyrene and be below the LaACES constraint of 500 g weight.
- Payload shall conform to LaACES constraint for the flight vehicle interface allowing the payload to attach to the flight string by having two holes placed 17 cm apart diagonally on the payload box and lid.
- Payload survivability shall be verified through thermal and vacuum tests confirming that the sensors are able to perform under the expected flight conditions and that the payload box is able to protect the contents under the same conditions.
- The payload shall have a lithium battery pack to keep it functional throughout its entire flight.
- The payload shall be launched by the LSU team with simple instructions.
- The payload shall be recovered by the LSU team with data transmitted using the GPS beacon attached to the balloon vehicle.
- The data analysis plan shall be organized so that the science results can be presented 48-hours after flight.

4.0 Payload Design

4.1 Principle of Operation

The LaACES MegaSat payload will serve as the base unit for the DemonSat-2 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

The data will be stored on a microSD card for post-flight retrieval. The prototype will be powered by a 12-V lithium battery pack. The prototype will not have radio or wireless communication capabilities. The prototype will also be protected during flight by a vented Styrofoam box designed to handle falling from high altitudes.

4.2 System Design

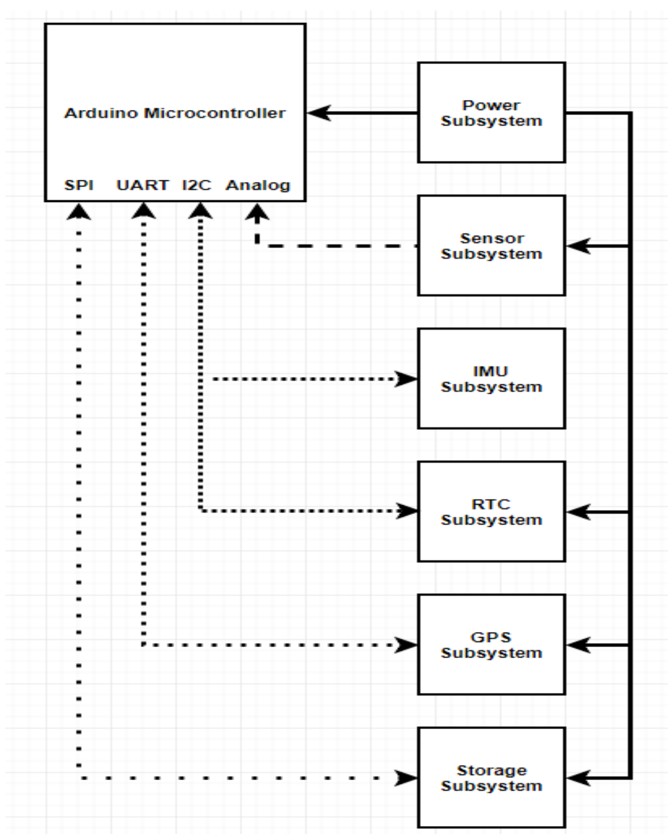


Figure 7: High-level system design of the MegaSat [12]

4.2.1 Functional Groups

The system design can be divided into six major functional groups: the power subsystem, the sensor subsystem, the IMU subsystem, the RTC subsystem, the GPS subsystem, and the storage subsystem (see Figure 7). The controller for the system will be the Arduino Mega 2560, which will control the interactions of all the subsystems. Power will be supplied by lithium batteries since these lightweight batteries perform well in cold temperatures and low pressures.[13] The sensor subsystem will consist of an Arduino Mega shield designed and printed by the LSU LaACES team, which will house the circuitry for the two thermometers, pressure and humidity sensors, and the acceleration/gyroscope module (IMU 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

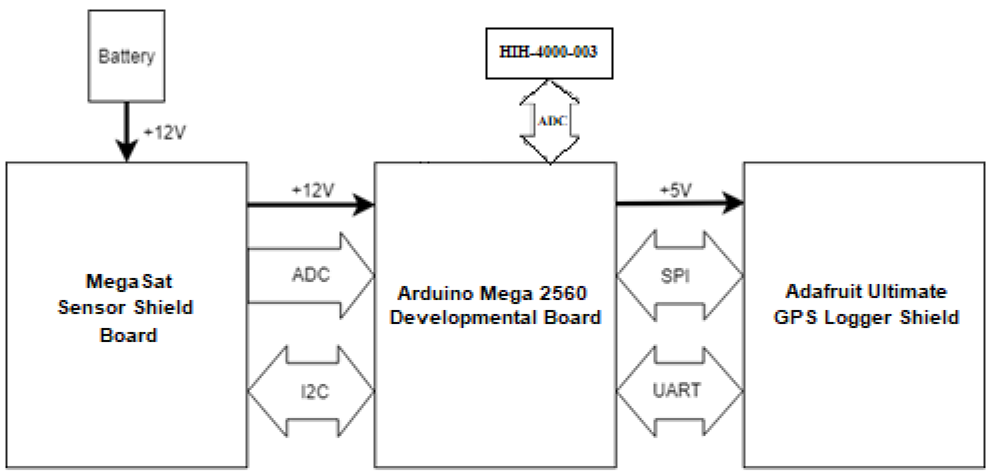


Figure 8: Group Interfaces of the MegaSat [1]

The system can be divided into five major group interfaces: power supply, system control (Arduino Mega), sensor shield board, GPS logger shield, and a backup HIH-400-003 humidity sensor (see Figure 8). Power will be supplied by a 12V lithium battery pack made from two 2CR5 batteries connected in series. The controller for the system will be the Arduino Mega 2560, which will control the interactions of all other functional groups. The Sensor shield 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 circuitry, a real-time clock (RTC), and the Sparkfun MPU-6050 Triple Axis Accelerometer/Gyroscope. The final group interface is the Adafruit Ultimate GPS logger shield, which contains the GPS transceiver and microSD module.

The group interfaces will be stacked as follows: the Arduino Mega 2560 on bottom, then the LSU MegaSat shield board, then the Adafruit Ultimate GPS logger shield, and the battery pack will be placed along the side the boards, but within the enclosure. The battery will be accessible prior to launch and operated with an I/O rocker switch.

Table 2: Traceability of Science Objectives

Objectives	**Requirements**	**Functional Components**
Determine the readings of temperature in relation to the altitude.	External dry bulb temperature reading shall be obtained and accurate within 5°C, in a range of -70°C and 30°C.	Calibration of the temperature sensors in the subsystem should be within 5°C for a range of -70°C to 30°C and will have a sampling rate of 10 seconds between cycles.
Determine the readings of atmospheric pressure in relation to the altitude.	Atmospheric pressure reading shall be obtained and accurate within 5% full scale in a range of 0 mbar abs. to atmosphere.	Calibration of the pressure sensor in the subsystem should be within a 5% f. s. for a range of 0 to 1015 abs. mbar and will have a sampling rate of 10 seconds between cycles.
Determine the readings of % relative humidity in relation to the altitude.	Percent relative humidity reading shall be obtained and accurate within 10%, in a range of 0% - 100% RH.	Calibration of the % RH sensor in the subsystem should be within a 10% RH for a range of 0% to 100% and will have a sampling rate of 10 seconds between cycles.
Determine the calculations of saturated vapor pressure using external temperature and pressure readings in relation to the altitude.	Temperature, pressure, and shall be accurate and combined using Buck's equations.[2] Altitude readings shall be obtained using the GPS module, GGA NMEA sentences for an interval over a range of 0 km to 30.5 km.	The data from the calibrated external temperature and pressure sensors in the subsystem should give the measurements needed to use Buck's equations. Altitude will be reported by the GPSGGA NMEA sentences.
Compare %RH readings and the saturated vapor pressure with altitude to determine the vapor pressure (absolute humidity) during flight.	External temperature, pressure, and % relative humidity readings shall be obtained, Buck's equations shall be implemented, and humidity and vapor pressure data shall be compared.	The data from the calibrated external temperature and pressure sensors in the subsystem should give the measurements needed to use Buck's equations. Altitude will be reported by the GPSGGA NMEA sentences.

Table 3: Traceability of Technical Objectives

Objectives	**Requirements**	**Functional Components**
Create a payload using Arduino Mega control system with temperature, pressure, humidity, altitude, and GPS sensors with a real-time clock.	Time-stamped data shall be recorded throughout the flight.	Proper installation and electronic verification testing
Stay within constraints set by LaACES for weight and mechanical design requirements.	The payload will remain below 500 g.	Weight budget shall be monitored.
	Holes spaced 17 cm apart diagonally on the lid allowing flight string attachments.	Proper design and implementation of payload box
	Payload shall use a lithium battery pack.	The battery shall be verified to last the entire duration of the flight.
	Payload shall be recovered by LaACES team.	The GPS beacon shall be transmitted location data to the balloon vehicle.
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.	Calibrate 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.	Collection and observation of GGA NMEA sentences from GPS subsystem
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.

4.3 Electrical Design

4.3.1 Sensors

Table 4: List of Sensors Used in the MegaSat

Sensor	Part Number	Signal Interface	Power Supply	Operating Temperature (°C)
Real-time Clock [14]	DS3231 RTC	I2C	3.3 V	-40 to 85
Thermometer [15]	1N457	Analog	5 V	-223 to 126
Atmospheric [16] Pressure	1230-015A-3L	Analog	5 V	-40 to 125
% Relative [17] Humidity	HIH-4000-003	Analog	5 V	-40 to 85
GPS [19]	FGPMMOPA6H	UART	3.3 V	-40 to 85
Acceleration [20]	MPU-6050	I2C	3.3 V	-40 to 85
Gyroscope [20]	MPU-6050	I2C	3.3 V	-40 to 85

[#] References

4.3.1.1 Real-Time Clock (RTC) – The DS3231

The DS3231 is an accurate, I2C integrated real-time clock (RTC) that manages all time keeping information, seconds, minutes hours, day, date, month, and year, along with an automatic compensation for leap year. A 32kHz temperature-compensated crystal oscillator is integrated to maintain the time elapsed. This crystal oscillator is used due to its ability to avoid external temperature changes that can affect the oscillation frequency, thus affecting the accuracy of the DS3231. The RTC is equipped with a battery backup, to ensure data is still calculated in the event of a possible power failure.

The DS3231 datasheet states the accuracy of the RTC is ±2 minutes per year in temperatures between -40°C to +85°C. [14] Given that the DS3231 is interfaced via I2C, the required voltage supply must be 3.3V and less than 200 mA current. The datasheet also states, to avoid damaging the module, low temperature (260 °C) and quick soldering is needed.

Figure 9 displays the connection of the RTC (slave) to the Arduino Mega 2560 controller (master) through inter-integrated circuit (I2C) communication. The design of the I2C serial protocol allows for 10 sensors to be connected to the Arduino Mega by way of two “open-drain” lines called the Serial Data (SDA) and Serial Clock (SCL). The SCL line acts as the clock signal that is responsible for adjusting and transferring data between devices on the I2C bus and the SDA line carries the data. In the MegaSat, there are two sensors attached to this I2C bus, the MPU-6050 accelerometer/gyroscope module and the RTC.

Mega that uses 5V logic. This means that the level shifter allows the external master and slaves to communicate with each other by changing the voltage amplitude of the signal bidirectionally.

The Arduino programming assigns each senor and its respective sensor components with a hexadecimal address. This address allows for a sensor signal to be called into program writing or

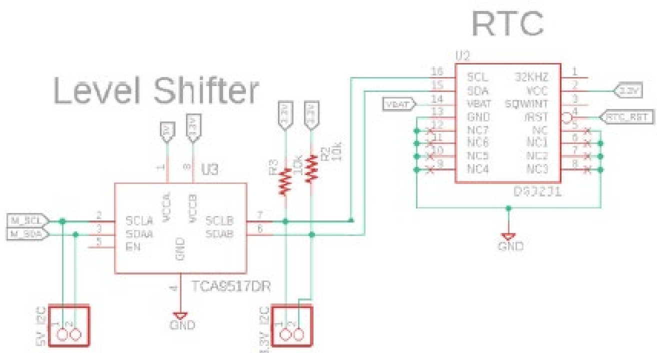


Figure 9: Schematic of RTC Circuit of MegaSat [21]

The circuit above shows that the SDA and SCA lines are pulled up with 10 kΩ resistors connected to the 3.3 V supply. Since the two lines are "open-drain", pull up resistors are required to be attached to them. These pull-up resistors allow for the I2C logic to be transferred bidirectionally to the master and slave(s). The level shifter(U3) incorporated into the circuit design, allows communication between the I2C sensors that use 3.3V logic and the Arduino Mega that uses 5V logic. This means that the level shifter allows the external master and slaves to communicate with each other by changing the voltage amplitude of the signal bidirectionally.

The Arduino programming assigns each senor and its respective sensor components with a hexadecimal address. This address allows for a sensor signal to be called into program writing or reading by listing this specific 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 internal and external temperature sensors used in the MegaSat, supply temperature readings over a broad scale of temperatures (-70 to 30 ℃). The forward biased 1N457 can provide linear changes of temperature (50 to 400K) versus voltage. The temperature sensitivity of the 1N457 diode is approximately -2.5 mV per Kelvin or degrees Celsius.[15] The internal and external temperature sensor circuits consist of a 1N457 diode, an AD820 operation amplifier, LM234DT current source, AD820 operational amplifier, three 0.1µF ceramic capacitors, a 10 µF electrolytic capacitor, resistors, and potentiometers. (See Figure 10).[21]

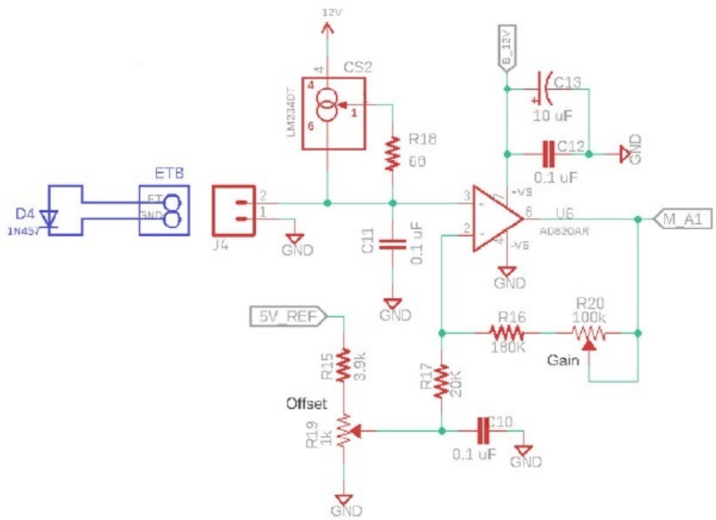


Figure 10: Schematic of 1N457 Diode Thermometer Circuit of MegaSat [21]

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 10, is used to provide a constant 1 mA current bias to the diode. This set current (I_{set}) can be calculated using the formula:

$$I_{set} = \frac{227\mu\frac{V}{K} * (300K)}{R12}, \text{ where } R_{12} = 68\Omega \qquad \textit{(Eqn. 6)}$$

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:

$$A_V = 1 + \frac{R16 + R20}{R17} \qquad \textit{(Eqn. 7)}$$

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 ADC input of the Arduino Mega. 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 sensors shall be tested at a variety of temperatures with room temperature air, freezer air, warm water, salted ice water, iced water, and in dry ice. The sensors will be compared to the Onset Hobo and InTemp dataloggers (MX2302A and CX603) or a Type-K Thermocouple with reader. Then an analog/digital calibration curve will be developed for later programming use.

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

The 1230 designed by TE Connectivity is a piezoresistive silicon pressure sensor packaged with a dual-in-line configuration. The pressure sensor uses a piezoresistive strain gauge in a Wheatstone bridge configuration, this produces a differential output voltage of 1 to 100 mV for a change in internal strain caused by differences in pressure. This sensor can measure 0 to 15 psia (0 – 1034 mbar).[16] The performance of this sensor is +-0.5%. The non-linearity is +-0.1%.

Integral temperature compensation is provided over a range of -20°C to +85°C using laser-trimmed resistors. A second laser-trimmed resistor is included to normalize pressure sensitivity variations by programming the gain of an external differential amplifier. 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.

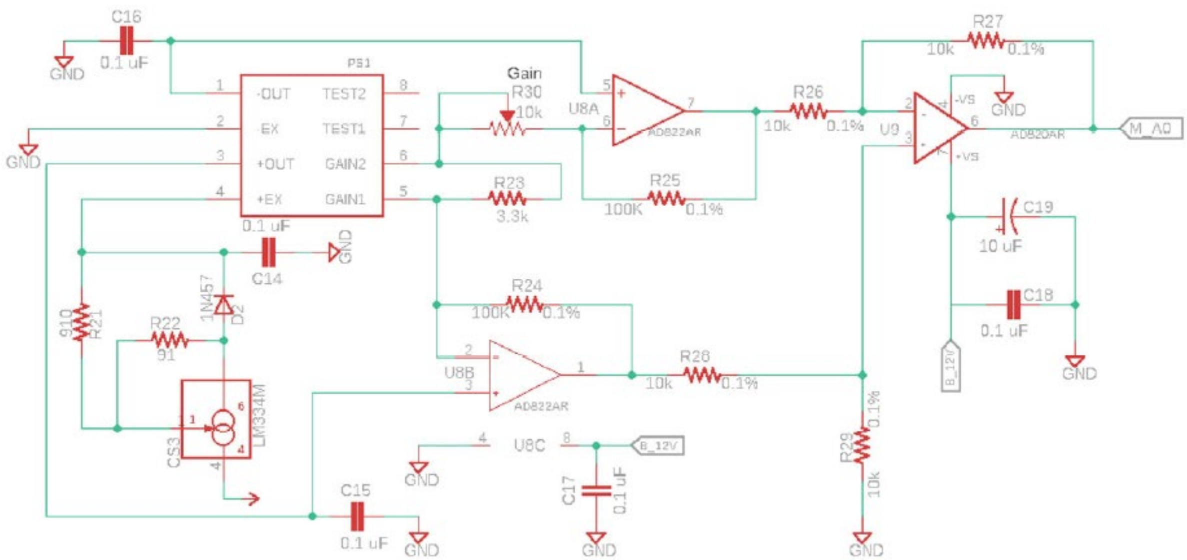


Figure 11: Schematic of Pressure Sensor Circuit of MegaSat [21]

In the pressure module circuit shown in Figure 11, a constant current supply is supplied at 1.5mA by CS3 in a zero-temperature coefficient current source configuration. Using Ohm's law:

$$R1 = \frac{0.134V}{1.5mA} = 89.3\Omega \qquad \textit{(Eqn. 10)}$$

R1 = 893Ω and R2 = 10 Ω, the resistors 91 Ω and 910 Ω 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 kΩ. R30 can be tuned to provide a more precise control of gain given by the formula:

$$R_{eff} = R30 + \frac{Rint*R23}{Rint+R23} \text{ and } Av = \frac{R24}{R_{eff}} \qquad \textit{(Eqn. 11)}$$

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 to condition the signal for the Arduino's analog input.

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 the amount of water vapor present in the air, to the maximum amount that would be present if the air were saturated. The HIH-4000-003 humidity sensor measures the partial pressure of water vapor as a ratio to the equilibration (saturation) vapor pressure. The Earth's atmosphere is made up of a variety of gasses such as Nitrogen, Carbon Dioxide, Argon, and Oxygen. Water vapor can also be found in trace amounts. The atmospheric pressure equates to the sum of all the partial pressures of each gas.

According to its datasheet,[17] 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 12):

$$V=(0.05*x) + (0.75) * \%RH \qquad \textit{(Eqn. 12)}$$

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.

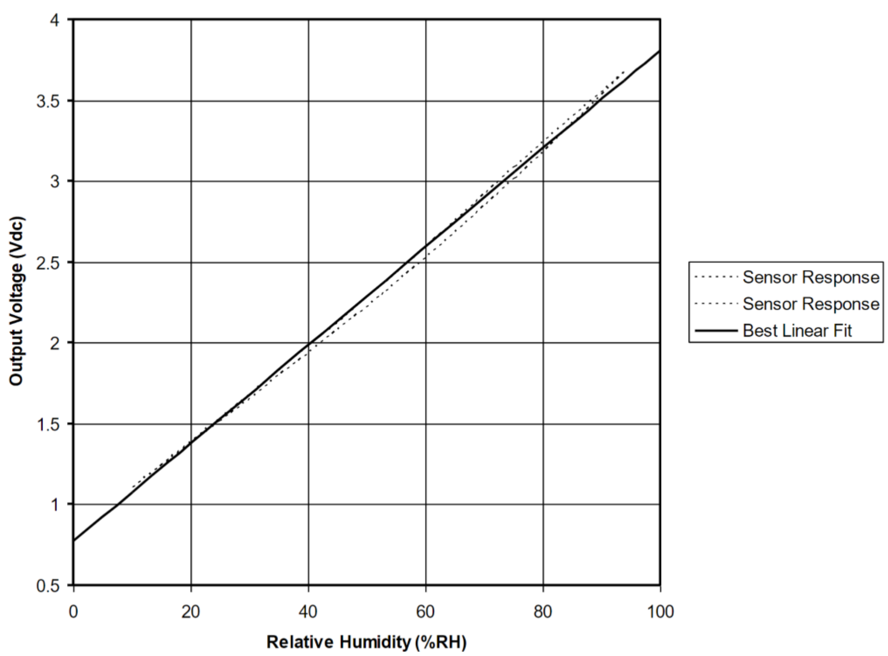


Figure 12: Expected Output voltage vs. %Relative Humidity [17]

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:

$$\text{True RH} = \frac{\text{(Sensor RH)}}{1.0546 - 0.00216(\text{T})} \quad \textit{(Eqn. 13)}$$

Measured values in %RH are expected to increase as the payload rises to the troposphere especially if clouds are present, this is because of 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 as a result in the decrease in air density. A graph of %RH can then be plotted as a function of time to determine the position of the payload relative to altitude.

Figure 13 displays the circuit design for the humidity sensor on the MegaSat shield. Various components that are included were used to optimize the humidity sensing module by reducing noise. A steady supply of 5V, is required for the sensor to work properly. This power supply 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.

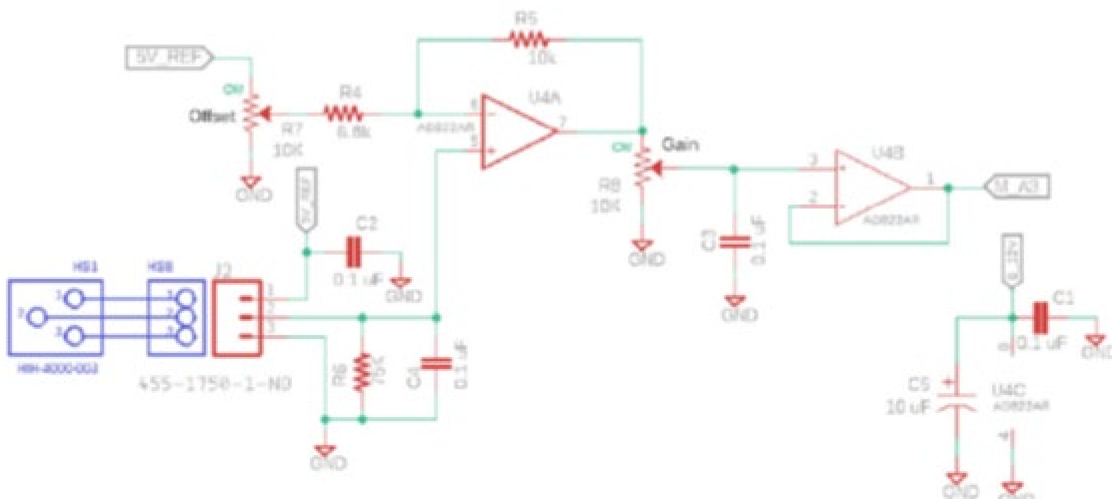


Figure 13: Schematic of Humidity Sensor Circuit of MegaSat [21]

During the DemonSats-1 calibration testing showed that the existing filter design for the LSU MegaSat resulted in a limited range of 80% (see Figure 14).[8] Therefore, a second HIH-4000-003 was included in the DemonSats-1 design to ensure that the team received data from the full range of 0-100% relative humidity. The second sensor was connected to the Arduino Mega 2560 directly through the analog A4 pin with no filter.

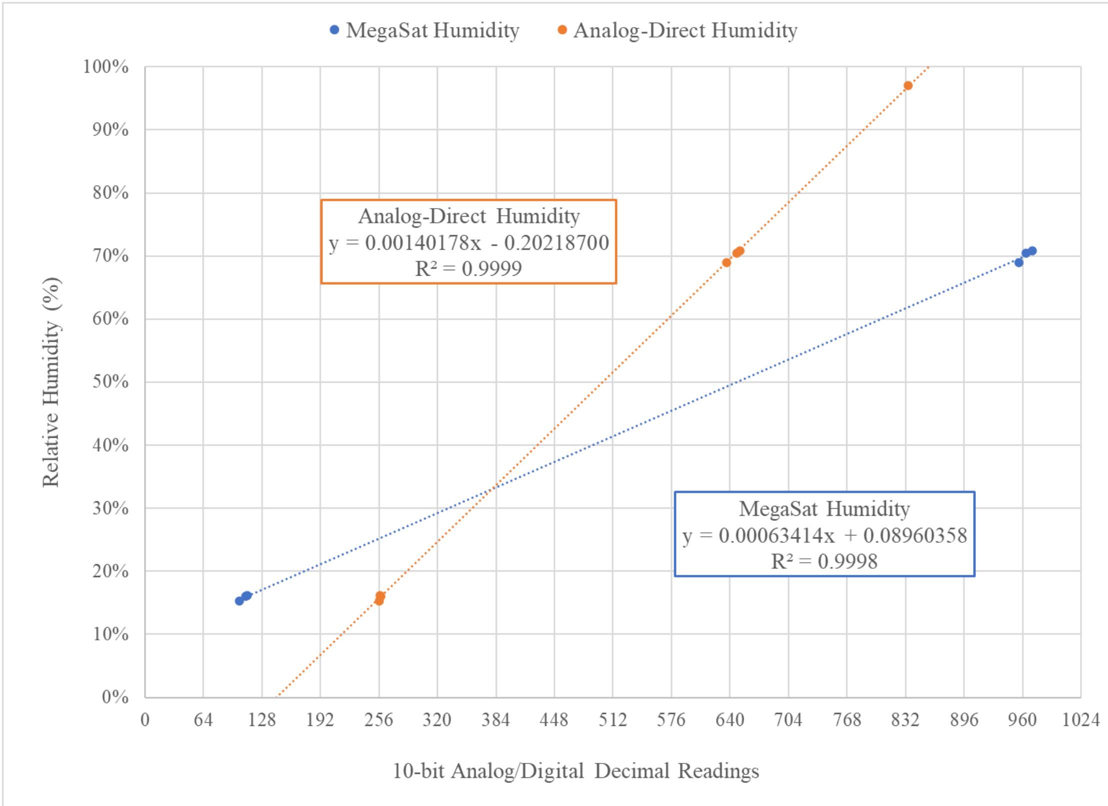


Figure 14: The Comparison of the MegaSat Humidity Sensing System and the Analog-Direct Humidity Sensor Outputs from DemonSats-1[8]

The DemonSats-2 teams has explored how to fix the MegaSat filter design for the HIH-400-003 to improve the sensitivity when compared to the analog-direct sensor and to achieve a reading range 0 – 100%. The schematic in Figure 13 was constructed using through hole components and a breadboard (see Figure 15).

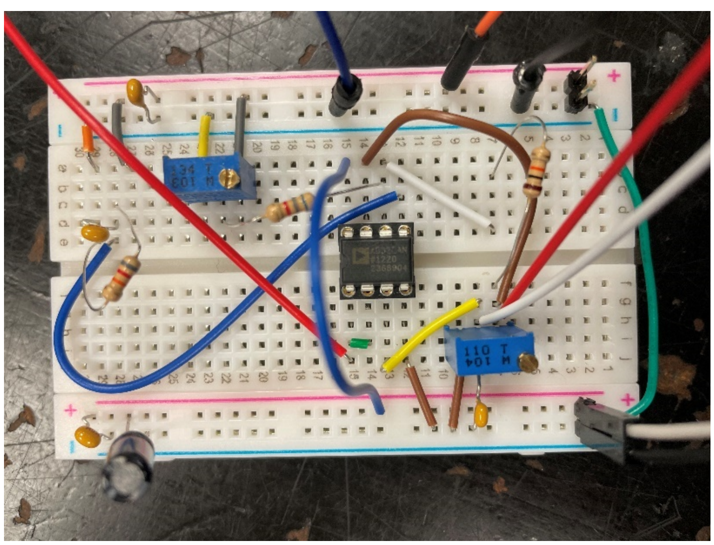


Figure 15: Redesigned Humidity Sensor Filter Circuit

Then a HIH-4000-003 sensor and the circuit voltage output was compared to the %RH readings from the HOBO External Temperature/RH Sensor Data Logger (MX 2302A) and the VLIKE MS6508 Digital Temperature Humidity Meter while all the probes were in either Lithium Chloride (~11% RH)[18] or in the cool mist of a reptile fogger (~100% RH). To have a filtering circuit with the best sensitivity and range, the voltage outputs of the sensor and circuit should yield 0 V for 0% RH and 5 V for 100% RH. The voltage output of the circuit used measured using a Vernier LabPro and Vernier Voltage Sensor.

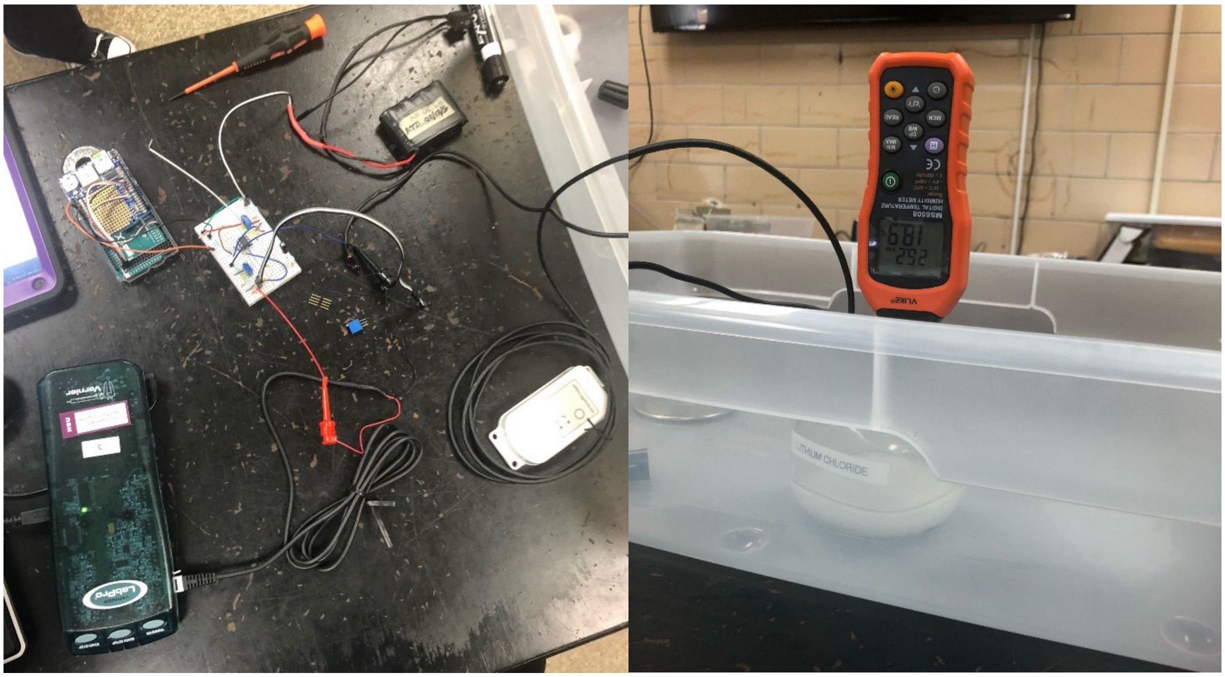


Figure 16: Testing the Humidity Sensor Filter Circuit Redesign

It was determined that the filter circuit which could provide the correct sensitivity and range would have the R8 potentiometer in Figure 13 switched from the 10 kOhm potentiometer to a 100 kOhm potentiometer. Through experimentation, the humidity readings were at the largest sensitivity within the 5 V output range when the R8 was set to 39-40 kOhm, while the offset potentiometer was set to 7.1 kOhm between the wiper and power (5V ref) pins.

Table 5: Readings from Humidity Sensor Circuit Redesign

Testing solution	**Expected %RH of air inside chamber** (at 25^{o}C)	**HOBO MX2302A Reading**	**VLIKE MS6508 Reading**	**R8 set to 39.4 kOhm (wiper to pin 7 side)**
Lithium Chloride (LiCl)	11.35%[18]	13.3%	15%	0.606 V
Cool Mist from Reptile Fogger	100 %	100%	100%	4.6 V

Based on the two measurements, the expected output for the redesigned humidity sensor filter circuit is shown in Figure 17. There is still 0.4 V available to add to the range and the 100 kOhm potentiometer allows for more adjustment when the circuit is calibrated on the MegaSat. Adding a 100 kOhm potentiometer instead of a 50 kOhm potentiometer to the design allows for the potentiometer to operate at its best mechanical setting, which is where the wiper is near the middle. The corrected schematic for the humidity sensor circuit is shown in Figure 18.

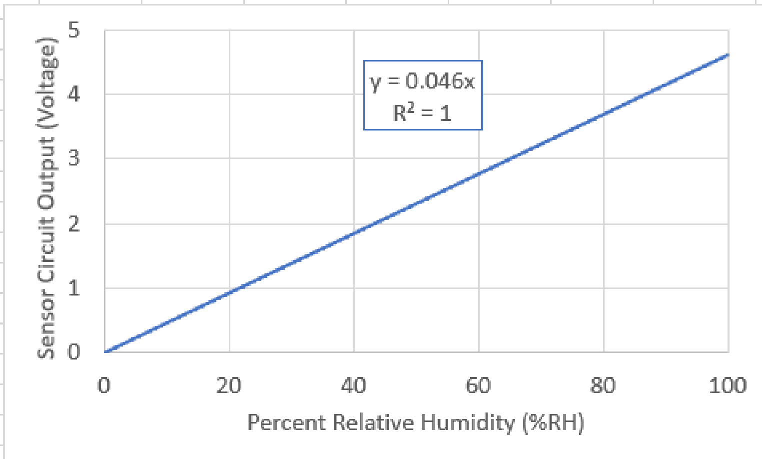


Figure 17: Expected Output of the Humidity Sensor Filter Circuit Redesign

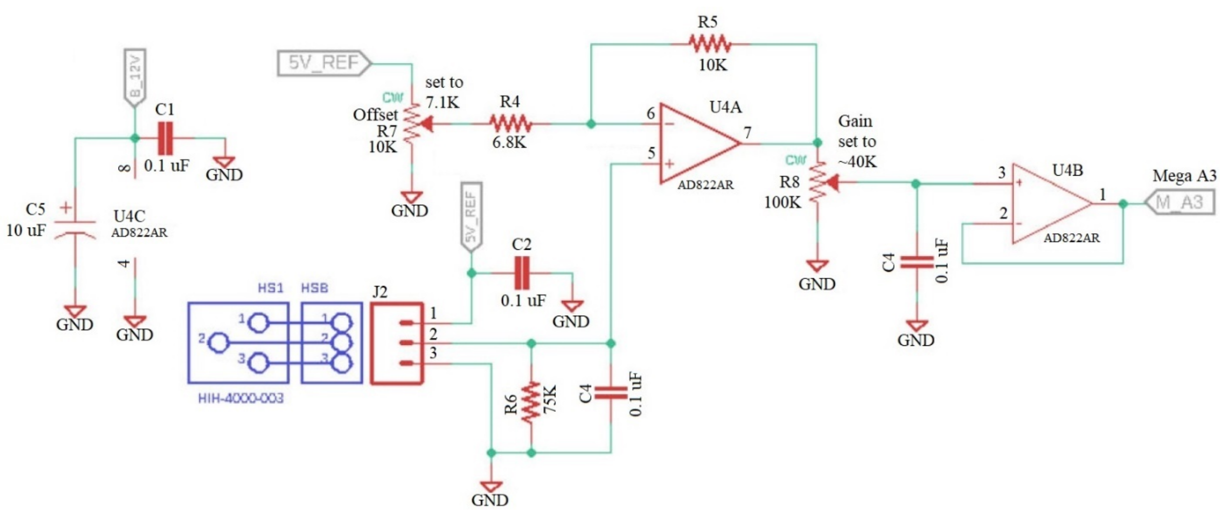


Figure 18: Schematic of the Humidity Sensor Filter Circuit Redesign

The team also plans on including an additional, unfiltered HIH-4000-003 sensor connected to an extra analog pin on the Arduino Mega for back-up data, since it is easy to add to the payload with a low amount of additional program data use.

The humidity sensors will be tested using known %RH as reference, and analog/digital calibration curves will be developed for programming use. Salt-saturated chambers of different %RH levels will be used for calibration set points, and a the VLIKE MS6508 Digital Temperature Humidity Meter will also be used in parallel during the testing.[18] The HOBO External Temperature/RH Sensor Data Logger (MX 2302A) will be used to compare the calibrated HIH-4000-003 sensor circuits in the fridge and freezer tests of the complete payload.

4.3.1.5 GPS - FGPMMOPA6H GPS Module

The FGPMMOPA6H GPS module is packaged on the Adafruit Ultimate GPS Logger Shield. On the GPS module is the GPS Chipset MT3339 by MediaTek. This chipset is made 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.” [19]

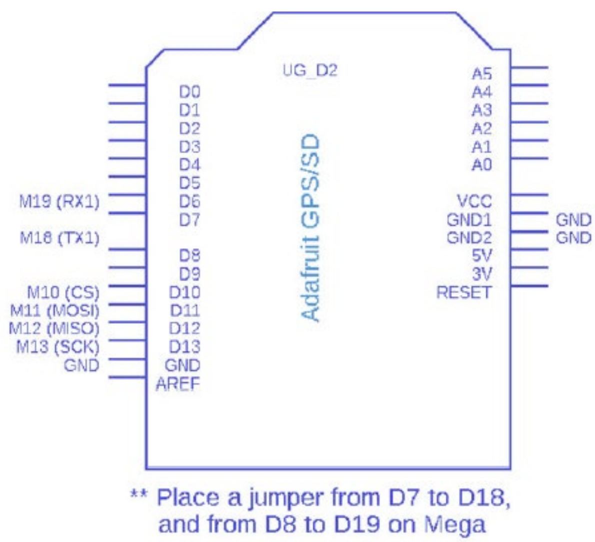


Figure 19: Pinout of Ultimate GPS Module to Interface with MegaSat [21]

To retrieve the GPS information, the Adafruit Ultimate GPS Logger Shield contains the aforesaid receiver which connects with the National Marine Electronics Association (NMEA) GPS tracking satellites. The data from the NMEA sentences received can be used to record where the payload travels during and after flight.   The data from this receiver will include the number of satellites, latitude, and longitude. GPRMC (Global Positioning Recommended Minimum Coordinates) is the default NMEA code type, but GPGGA (system fix data) type is needed for our team to collect altitude data. Therefore, both format types will be incorporated and collected as the RMCGGA option, which will be set in the Arduino IDE code. The GPGGA and GPRMC formats display the data in a different order. Using the Microsoft Excel data sorting function, the different NMEA sentences will be separated post flight. An example of the GPRMC and GPGGA 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 [19]

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 sentences 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 module will be checked for accuracy by reviewing data which will be obtained from automobile ground travel from Shreveport to Natchitoches. The GPS latitude and longitude data will be collected and stored onto the microSD card in the degree decimal format (DD.DDDD) so that the 3D Maps add-on function within Microsoft Excel can overlay the coordinates onto a Bing Map to verify that the travel coordinates are accurate. If the coordinates fail to be accurate, the Ultimate GPS module will be replaced one of the other GPS modules available. The code to parse the decimal degrees of the latitude and longitude is:

```
float latit = GPS.latitudeDegrees;
float longit = GPS.longitudeDegrees;
```

4.3.1.6 Sparkfun MPU-6050 Acceleration / Gyroscope

The MPU-6050 Triple Axis Accelerometer and Gyro Breakout by Sparkfun is combines a 3-axis gyroscope and 3-axis accelerometer into one module. The Sparkfun module communicates via I2C with the Arduino Mega 2560. The module uses an onboard Digital Motion Processor (DMP), that is capable of processing 9-axis “MotionFusion” algorithms.[20]

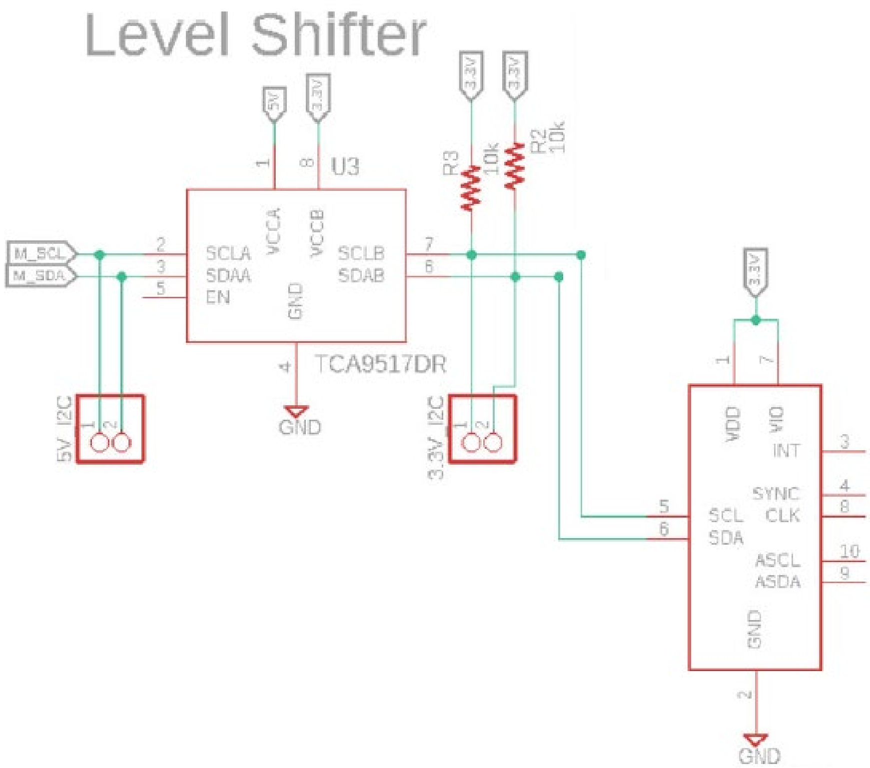


Figure 20: Schematic of MPU-6050 Module Interface with MegaSat [21]

The circuit above displays how the I2C components, including the MPU6050, are connected to the Arduino Mega. The MPU-6050 (slave) is connected to the Arduino Mega 2560 controller (master) through inter-integrated circuit (I2C) communication. The design of the I2C serial protocol allows for 10 sensors to be connected to the Arduino Mega by way of two “open-drain” lines called the Serial Data (SDA) and Serial Clock (SCL). In the DemonSats-2 payload, we have two sensors attached to this I2C bus, the MPU-6050 accelerometer/gyroscope module and the RTC. In Figure 15, the SCL and SDA lines are pulled up with 10 kΩ 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). The level shifter(U3) incorporated into the circuit design, allows communication between the I2C sensors that use 3.3V logic and the Arduino Mega that uses 5V logic. This means that the level shifter allows the external master and slaves to communicate with each other by changing the voltage amplitude of the signal bidirectionally. The Arduino programming assigns each sensor and its respective sensor components with a hexadecimal address. This address allows for a sensor signal to be called into program writing or reading by listing this specific address. The MPU-6050 module address is 0x69. Table 6 shows the list of the data addresses we will use in the Arduino IDE program.

Table 6: I2C Register Map for Data Used from MPU-6050[20]

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 21: MPU6050 with AD0 Pin Soldered to High

Since the 0x69 address is to be used for the MPU6050, the AD0 Jumper on the breakout board must be resoldered to set the AD0 pin high (+5V) instead of the default low (0V) for 0x68. Data from the Sparkfun MPU-6050 is not needed for the DemonSats-2 scientific objectives, however the team plans to incorporate this module from the MegaSat instructions for practice in I2C programming.

4.3.2 Sensor Interfacing

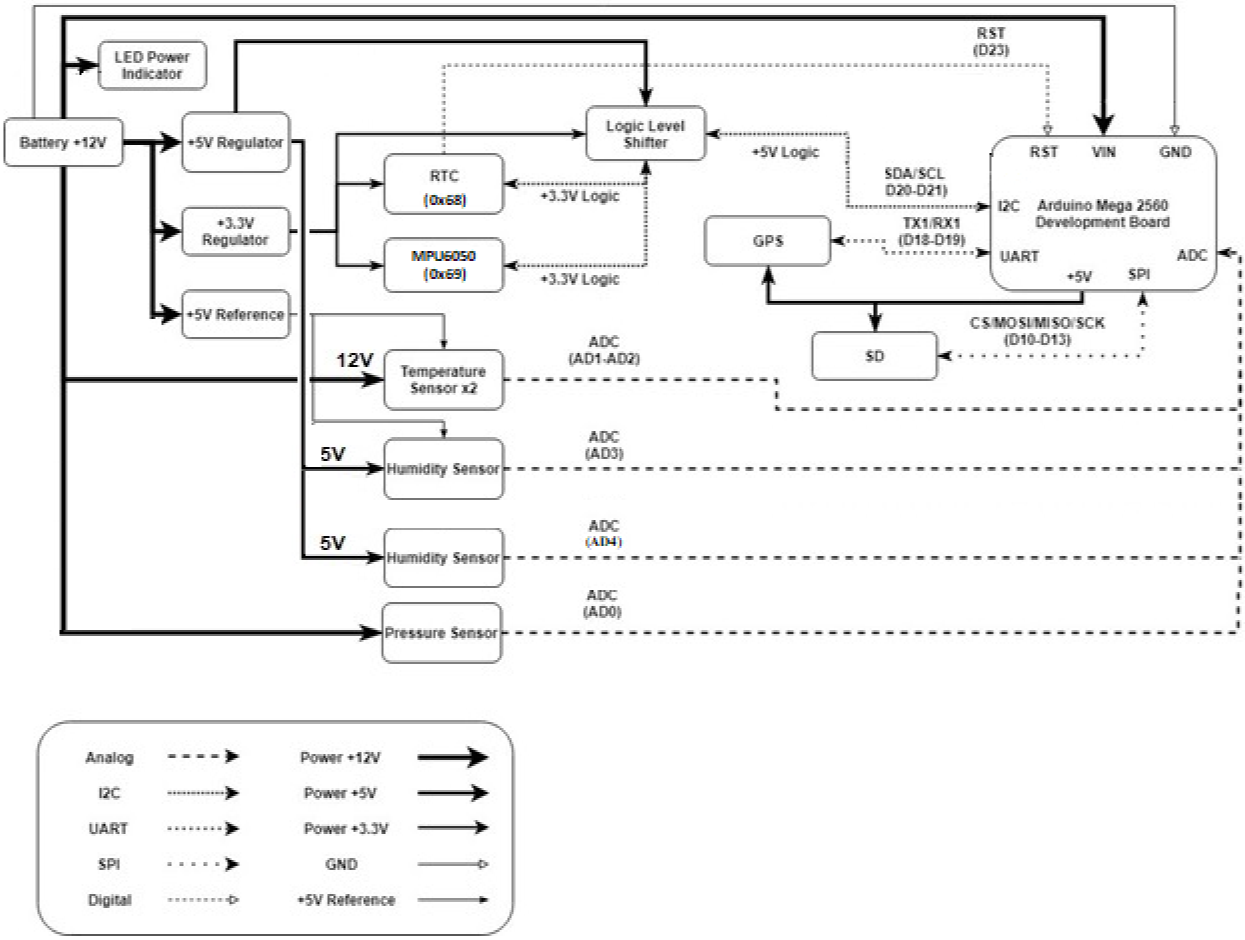


Figure 22: Mid-level block diagram of the DemonSats-2 MegaSat [1]

4.3.3 Control Electronics

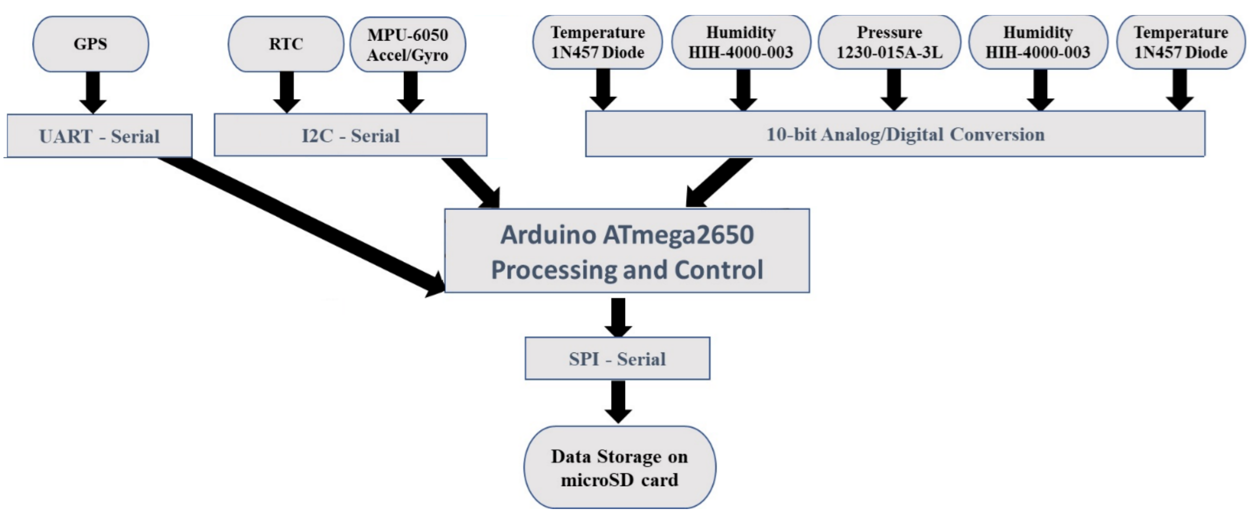


Figure 23: The Pathway from Sensor Input to Data Storage

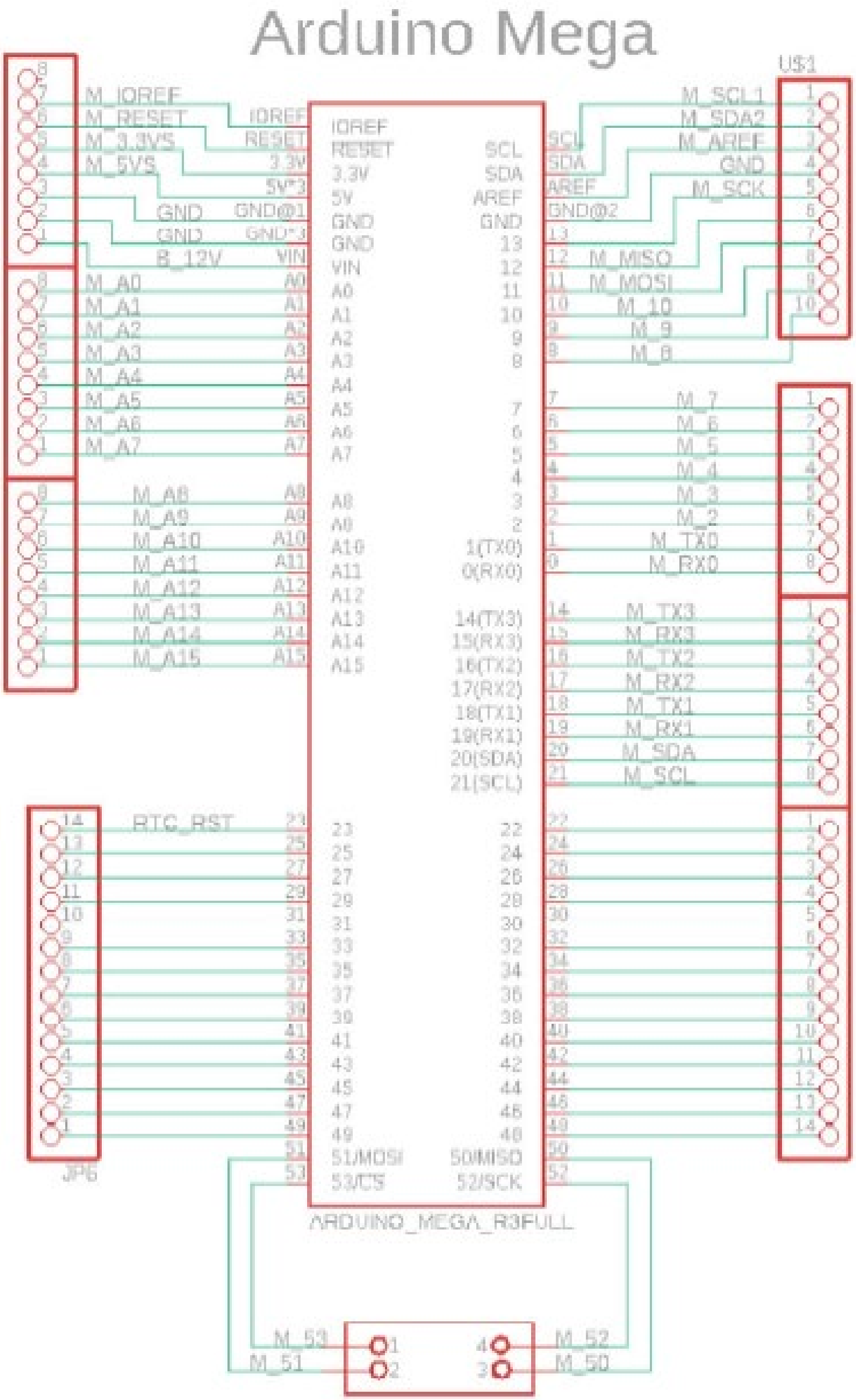


Figure 24: Pinout of Arduino Mega 2560 Interface with MegaSat [21]

4.3.4 Power Supply

The main power will be supplied by lithium batteries because they perform well in cold temperatures and low pressures and are lightweight.[13] A 12 V lithium battery pack will be made from two 2CR5 batteries connected in series (see Figure 25). Then, the battery pack will be connected to the MegaSat Sensor Shield. On the MegaSat shield, the 12 V main supply will be split into 5 V and 3.3 V using voltage regulators (see Figure 26).

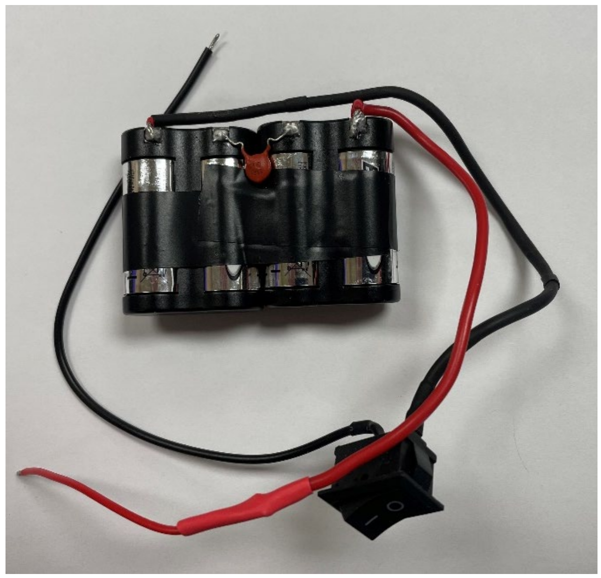


Figure 25: DemonSat-2 Battery Pack Design (two 2CR5 lithium batteries attached in series with a PTC thermistor)

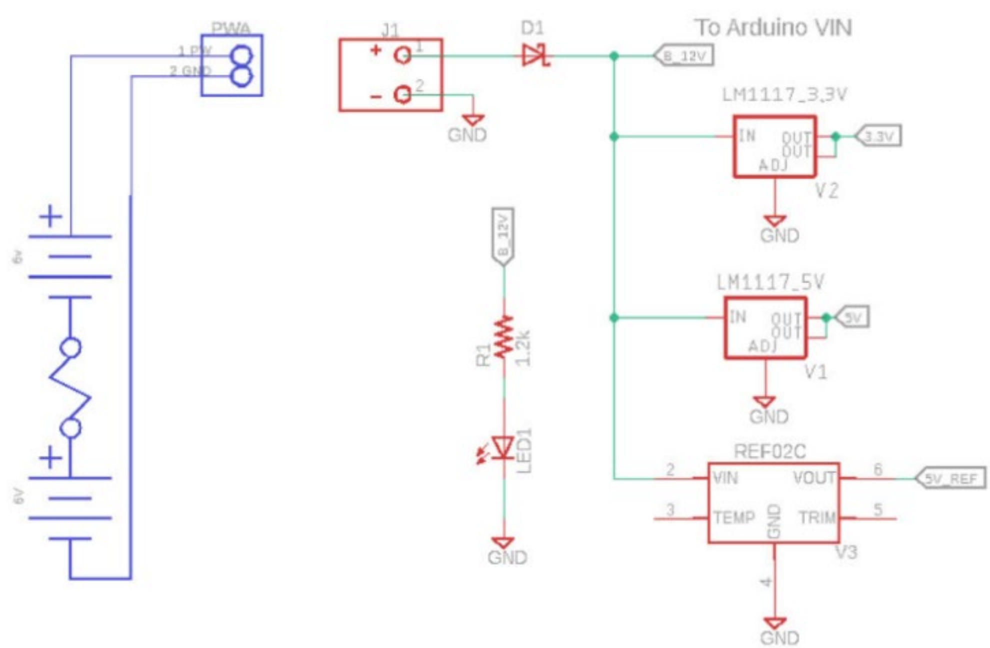


Figure 26: Schematic of Power Supply Circuit for MegaSat [21]

4.3.5 Power Budget

The amount of power needed for this payload has been calculated by adding the power requirements for the different components involved. The Arduino Mega 2560 controller will receive 12 V with an uncertainty of minus 2 V as the battery pack discharges. It has been observed by the team with the DemonSat-1 payload that full operation of the payload diminishes after the battery pack discharges below 10 V.[8] The current demand of the Arduino Mega has been measured with the DemonSat-1 payload. The rest of the voltage and uncertainties are based off the measured voltages from the current DemonSat-2 prototype. The current demands of the sensor components are obtained from their respective data sheets. [14-17, 20, 22] The calculated maximum operational power requirement for the payload is 3158 milliwatts.

The 12 V battery pack provides a typical capacity of 3000 mAh of current hours when rated at 200 ohms at 21℃.[23] Since the payload will be partially operated at subzero temperatures, the expected lifetime of the battery pack is assumed to be reduce to 30% of its typical capacity. Therefore, by comparing the power requirements and the power supply, the DemonSats-2 payload is expected to last approximately 4-6 hours. For the ACES-65 flight, the DemonSat-1 payload operated from approximately 3.5 hours.[5] Therefore, the battery pack for the DemonSat-2 payload should be sufficient designed for power supply.

Table 7: Power Budget for the DemonSats-2 Payload

Component	Voltage (V)	Current (mA)	Power (mW)
Arduino Mega 2560	12 (- 2)	200 (- 50)	1500 to 2400
GPS Logger Shield	5 (-0.2)	20[22]	96 to 100
MPU-6050 Accel/Gyro/DMP	3.3(-0.1)	3.9[20]	12.5 to12.9
Temperature Circuit (x 2)	12(- 2)	2[15]	20 to 24
Pressure Sensor	12(- 2)	1.5[16]	15 to 18
Humidity Sensor (x 2)	5(-0.2) * 2	0.2[17]	1.9 to 2.0
RTC module	3.3(-0.1)	0.2[14]	0.6 to 0.7
		Total Power Consumption (mW):	1646 to 2558
Battery Pack (2CR5 x 2)	Voltage (V)	Ideal Charge (mAh)	Energy (mWh)
	12	3000[23]	36,000
Expected Lifetime at Room Temperature (hours)			14 to 22
Expected Lifetime – De-rated for Subzero Temperatures (hours)			4.2 to 6.6 hours (30% effective)

4.4 Software Design

The Arduino Mega microcontroller in the MegaSat payload system will be programmed using Arduino IDE. The data will be analyzed by importing the comma-separated values (*.csv) file into Microsoft Excel.

4.4.1 Data Format & Storage

The team aims to have data values collected at a 1/10 Hz sampling rate. This data rate is fast enough to obtain data to achieve the scientific objectives. From the LaACES ACES-65 flight report, the average ascent rate was 1000 ft/min (5.1 m/s) and the average rate of descent was reported to be 1400 ft/min (7.1 m/s). [10] Therefore, the altitude readings at a 1/10 Hz sampling rate should vary between 50 m and 71 m. Data taken from the DemonSats-1 payload was taken at 1/3 Hz, however the altitude data was only recorded when the GPS module received a GGA signal, which occurs sometimes at a 1/50 Hz rate. Since the trends obtained from this data were still informative, 1/10 Hz is an effective in between reading which will allow the team to collect and work with less overall data compared to the 1/3 Hz.

The data is organized as a comma-delimited text string which is stored on the microSD card. Each string is expected to have 186 or 189 characters. At a 1/10 Hz sampling rate and estimating the payload flight to last approximately 2 hours with another 2 hours of ground time data collection in pre- and post-launch, the microSD card will be expected to store 1440 lines of data. Therefore, the data should be about 0.3 MB, which is far below the 32 GB storage capacity of the microSD card.

Table 8: Description of Output Data Characters to be Stored to MicroSD Card

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

Humid1	Humid2	RTC Time	Accel X	Accel Y	Accel Z	Gyro X	Gyro Y
DDD.DD,	DDD.D,	HH:MM :SS,	DD.DD,	DD.DD,	DD.DD,	DDD D,	DDD D,
7	6	9	6	6	6	5	5

Gyro Z	Pressure decimal	Ext Temp decimal	Int Temp decimal	Humid1 decimal	Humid2 decimal	GPS NMEA string
DDDD,	DDD,	DDD,	DDD,	DDD,	DDD,	GGA or RMC
5	4	4	4	4	4	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. 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 for the vapor saturation pressure vs. altitude using equations 4 and 5 of this report. Also, if the relative humidity data is accurate, 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

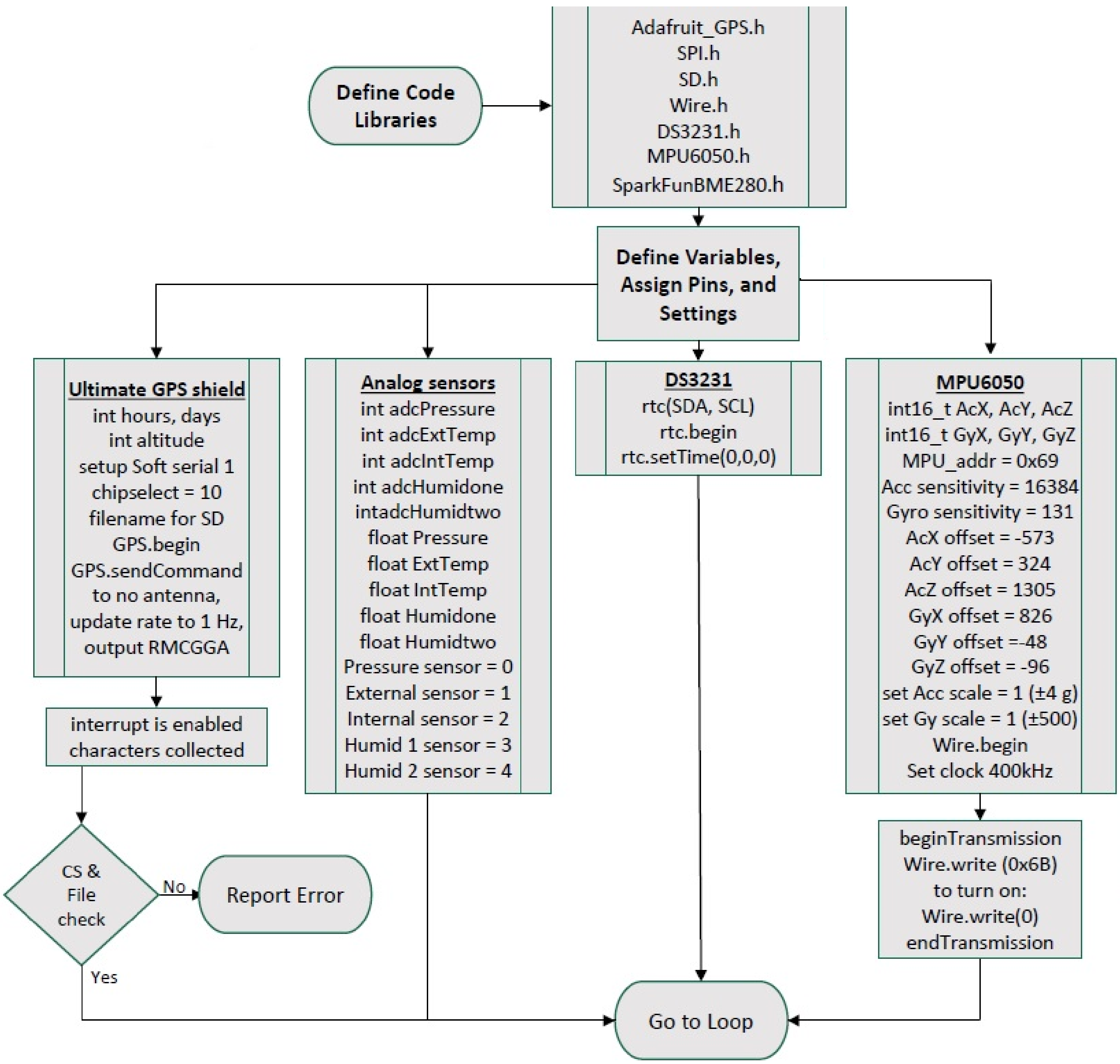


Figure 27: Flight Arduino IDE Software "void setup()" Plan for DemonSats-2 Payload [8]

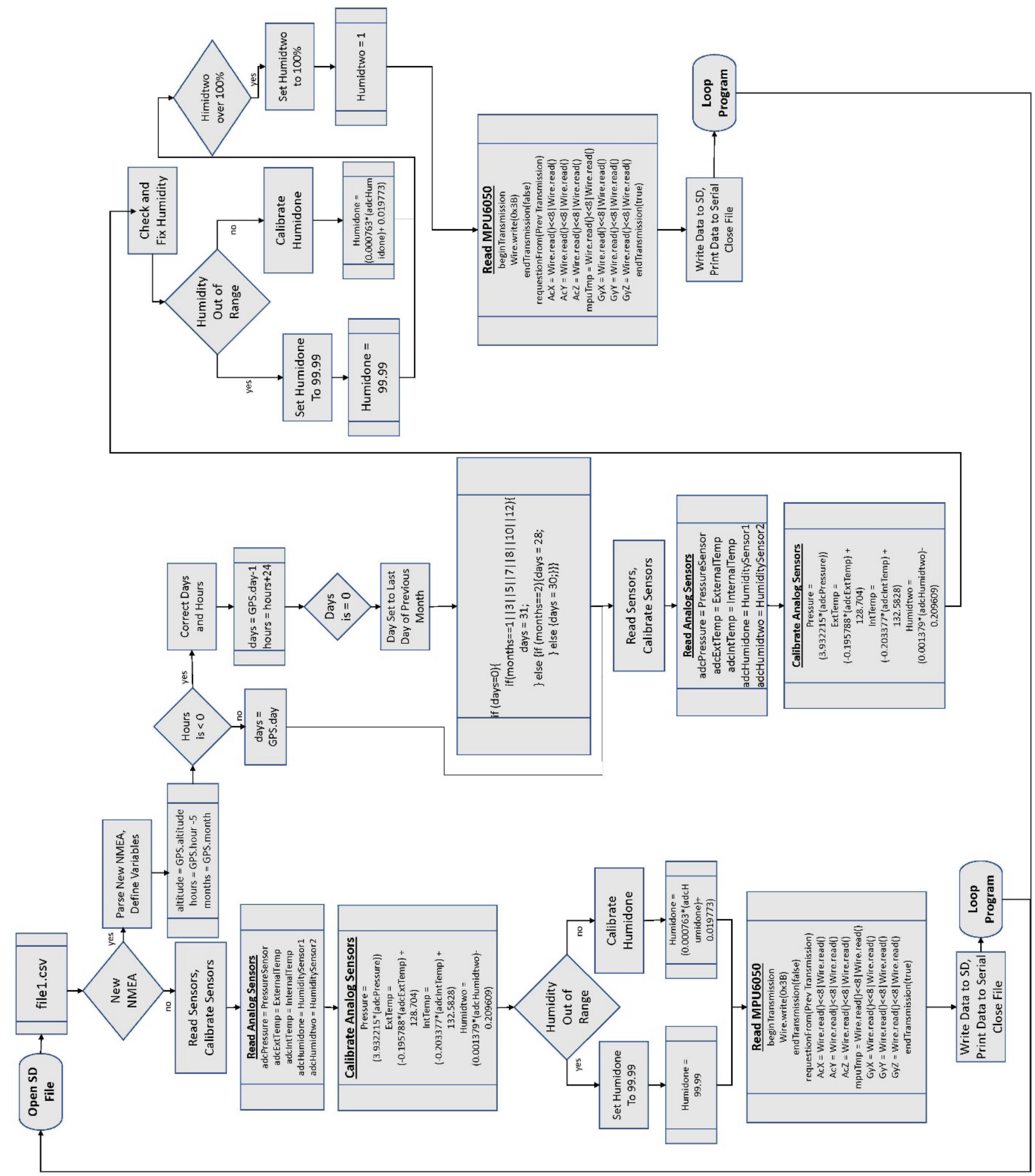


Figure 28: Flight Software Main Loop Plan for DemonSats-2 Payload

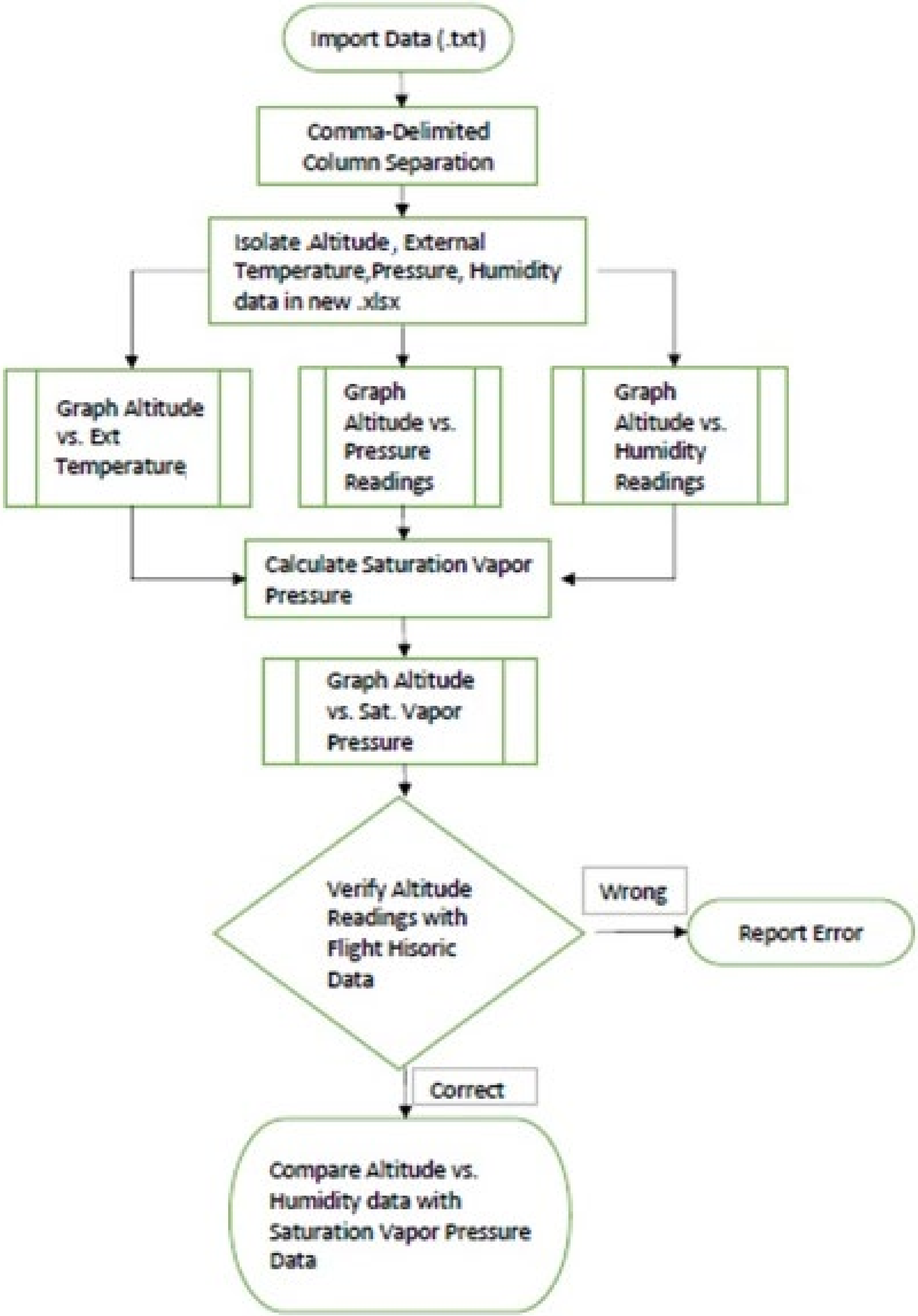


Figure 29: Post-Flight Software Plan for DemonSats-2 Payload [8]

4.5 Thermal Design

The payload is expected to experience a temperature range between -70 to 30 ℃. The payload electronics will be insulated inside a 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 ℃ temperatures. Since there are no high voltage circuits planned for the design, acrylic conformal coating will not be necessary to prevent arcing.

The greatest risk to the payload succumbing to the low temperatures would be the integrity of the power supply. Lithium batteries will be used since they perform better than other batteries in cold environments.[13] The batteries will be placed next to 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 ½-inch, moist-resistant insulation foam (Owens Corning Foamular PROPINK Insulating Sheathing). This foam is made of extruded polystyrene (XPS) with a reinforced skin laminated on both sides for extra damage and moisture control.[24] The ½-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 and Internal Structure

The DemonSats-2 payload will be subject to mechanical stresses during flight. These stresses 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 components of the payload acts against itself upon landing.

To mitigate the mechanical strain caused by these forces, a double-walled foam box will be used to house and to protect the payload electronics. Additional inserts will also be added within the box to secure components and the power supply. The following components will be 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. The design of the enclosure is based on specifications provided through the LaACES student ballooning course materials. [25] The foam box has 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 30). In addition to the box seen in the drawing, inserts will be placed at the bottom of the box to support the Arduino Mega, small inserts will be added to secure the battery, and an insert will sit on top of the payload electronics to secure its vertical motion within the box.

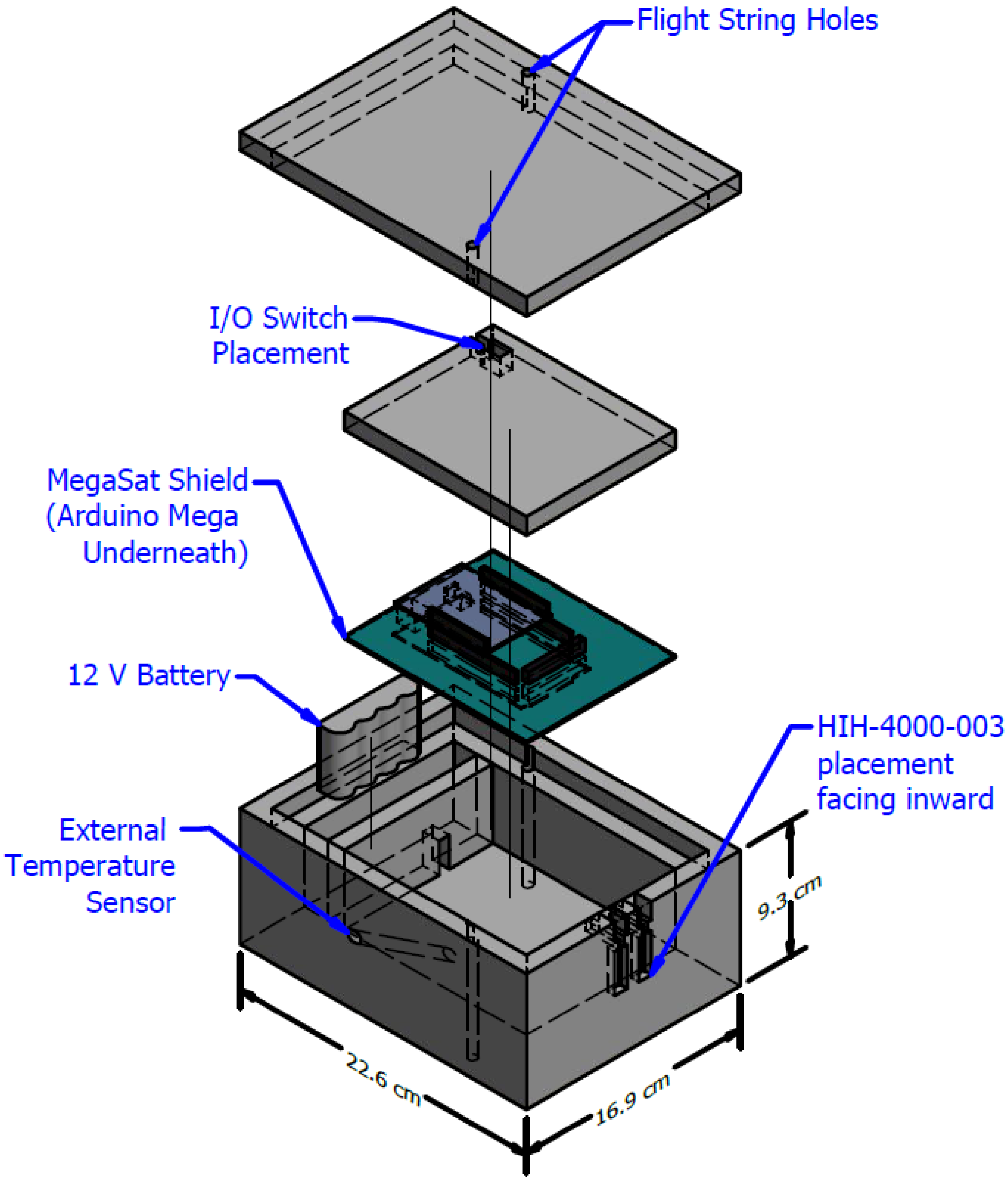


Figure 30: Mechanical Drawing of DemonSats-2 Enclosure

For better sensor accuracy and separation of the external temperature from the heated electronics, the external 1N457/A will be placed on the outside of the payload (see Figure 30). This outside placement was determined during the DemonSat-1 experiment. [5,8] The humidity sensors will also be placed on the outside of the insulated box, however they will be moved to the shorter edge opposite of the battery and near the humidity sensor circuitry (see Figure 30). This position is a move from their location in the DemonSat-1 experiment, since one of the sensors was taped over before the ACES-65 flight. The humidity sensors will also be loosely covered with heatsink and turned inward, toward the box to reduce the icing and condensation that occurred in the DemonSat-1 flight (see Figure 6).

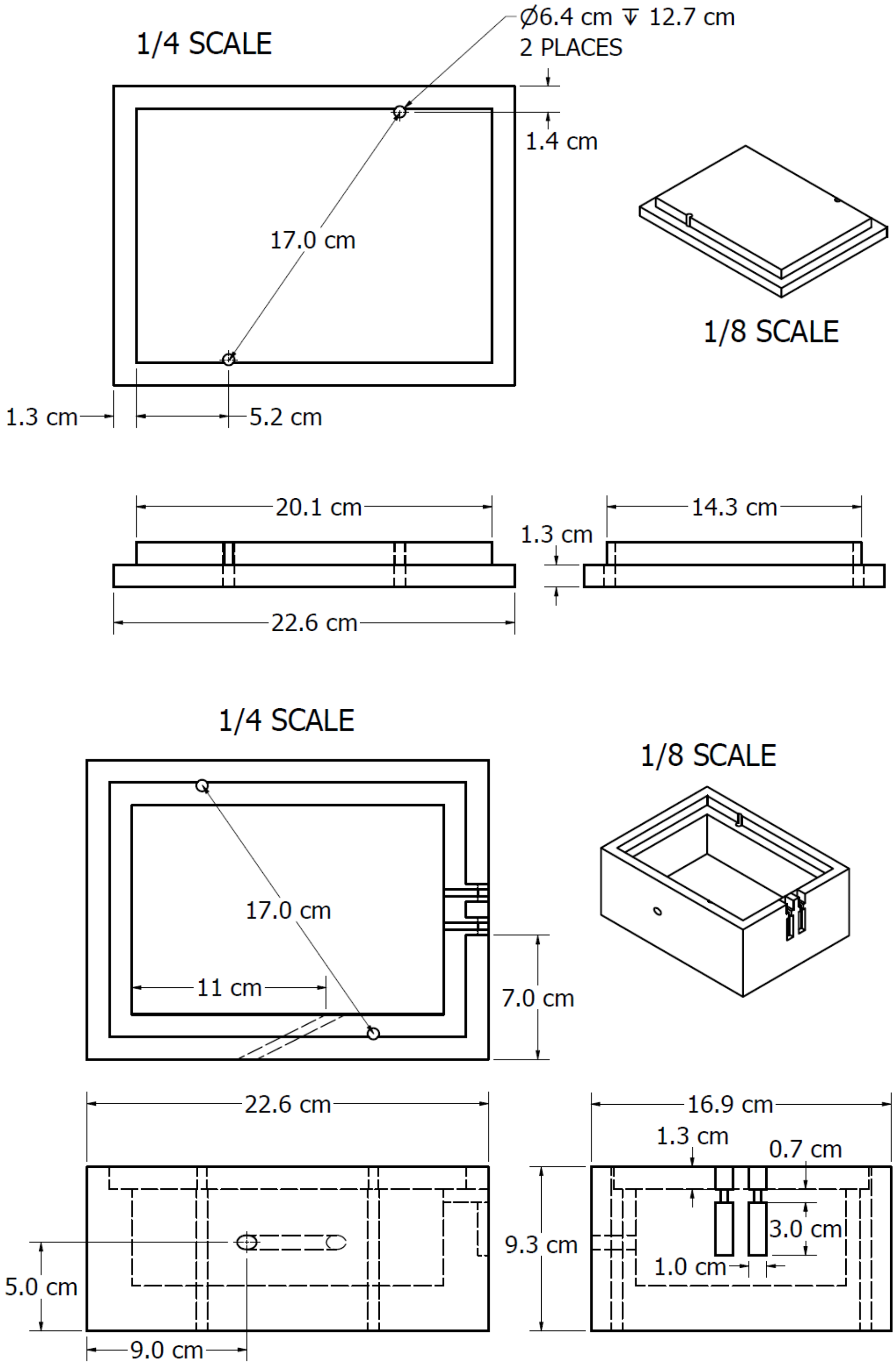


Figure 31: Orthographic Drawings of DemonSats-2 Outer Enclosure

4.6.2 Weight Budget

Table 9: The Weight Budget for DemonSats-2 Payload

Component	Weight (g)	Uncertainty (+/- g)
Arduino Mega 2560	37.0	0.5
GPS Logger Shield with microSD	24.0	0.5
MegaSat Sensor Shield	85.5	0.5
Battery Pack (2CR5 x 2)	92.8	0.5
Foam Box Enclosure	182.3	0.5
Miscellaneous (i.e. wires, hose)	24.2	1.0
TOTAL	**445.7 g**	**±1 g**

The mass budget of the payload above shows the distribution of mass for the main components of the payload. This mass is under the 500 g limit requirement for a LaACES flight.

Figure 32: The Components and Completed Payload on the Scale (total = 0.4457 kg)

5.0 Payload Development Plan

5.1 Electrical Design Development

Since the main objective of the electrical design is to build the LSU MegaSat payload, most of the development has been done. The DemonSats-2 team will follow the assembly manual provided by the LSU team to construct the payload. To ensure that each sensor is working efficiently, each will be tested under the operating conditions. The tests that will be performed will determine whether a sensor needs to be modified in design or replaced.

The only electrical design changes that have occurred between the LSU MegaSat and the DemonSat-2 electronics is the swapping of the R8 - 10 kOhm potentiometer used to adjust the gain in the humidity sensor circuit to a 100 kohm potentiometer so that the sensitivity and range can be set to the appropriate level around 40 kOhms. (See Section 4.3.1.4 for more details about this design change.)

5.2 Software Design Development

The DemonSats-2 team will program the MegaSat payload using Arduino IDE. Example code from the Sparkfun and Adafruit will be used to start the coding for the MPU-6050 module and Ultimate GPS logger shield, respectfully. The main goal of the Arduino programming will be to send a string of characters that list the sensors values to the microSD card at a 1/10 Hz sampling rate, where the data will be stored.

The additional programming of manually calling I2C registry commands will also be added to the example coding so that the DS3231 RTC, and MPU-6050 accelerometer will both communicate with the Arduino Mega.

The comma-separated-values (*.csv) files stored onto the microSD card will be transferred after the flight to the team's laptop and Microsoft Excel will be used to analysis the data. Visual Basics Macro within Microsoft Excel will also be utilized by the team to speed up the repetitive data analysis operations. With the teams limited experiences of Microsoft Excel, this will also pose as an opportunity for them to learn an advance function of the program, along with basic programming using Visual Basics.

5.3 Mechanical Design Development

Based off the DemonSats-1 enclosure, Table 11 below details the foam part dimensions to be cut for the DemonSats-2 enclosure. The side and edge walls will be constructed with Gorilla glue and a butt joint, since it was determined that the Gorilla glue was strong enough to keep this joint intact [5] and the butt joint is easier to cut out than a mitered butt (bevel) joint.

Table 10: The Foam Board Template for the Payload Enclosure [8]

	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

In addition to the based box design, the DemonSat-2 team have developed isometric (see Figure 30) and orthographic drawings (see Figure 31) of the final mechanical design using Autodesk Inventor. The 3D-CAD design allowed the team to explore possible sensor positioning within the enclosure. Several team members have learned how to use Autodesk Inventor through coursework at NSU. To ensure a proper enclosure is built, protypes will be built for practice. After each enclosure prototype is made, the box will be dropped (from 10 feet) with an equal-weighted and sized substitute for the payload housed inside to determine box stability and fabrication quality.

The assembly of the foam will be done as described in the LaACES student ballooning course materials, "Constructing a Standard LaACES Payload Box." [25] Besides the foam, 2 straws will be used to line and protect the through holes that are 17.0 cm apart from the friction forces caused by the flight string. Also, plastic grommets will be place on the ends of the holes to assist with this protection.

5.4 Mission Development

Throughout the develop of the DemonSats-2 payload, four NSU undergraduates and an LSMSA high school student will be incorporated into various parts of the project. By the end of the first week of April, the team will be finishing the MegaSat sensor shield, completing the sensor cabling, developing automated data analysis in Microsoft Excel, and starting to calibrate each sensor using environmental chambers and working standards/equipment. Programming the Arduino Mega, developing technical drawings using Autodesk Inventor have largely been completed. Maintaining team project management and report writing is happening continuously toward the development of the DemonSat-2 mission.

In lieu of attending the thermal-vacuum testing at LSU in mid-April, the faculty advisor decided that the team was to participate with the other groups via Zoom and they performed the system tests on the DemonSat-2 payload at NSU. The thermal portion of the test was done by placing the complete payload into warm air (~30°C), a conditioned room (~20°C), a fridge (~0°C). standard freezer (-20°C), and ultra-subzero freezer (-80°C). The vacuum portion of the test was performed in an ABLAZE stainless steel 5-gallon vacuum chamber and the pressure was tested between 0 – 1014 absolute millibars. More detail on the system testing procedures has been included in Section 6.4.

6.0 Payload Construction Plan

6.1 Hardware Fabrication and Testing

Fabrication of the LSU MegaSat shield hardware was completed by the end of the first week of April. The basic electrical checkpoints which were tested on the MegaSat shield are included in Table 11. An effective, visual instruction manual has been provided with the MegaSat kit and has assisted the team in the assembly. The power, temperature, humidity, pressure, RTC, and MPU-6050 modules checkpoints were verified and reported in Table 11.

Table 11: Summary of Electronic Checkpoints for MegaSat Shield

ID	Name	Measured Value
+12 V	Input Power	+11.9 V
+5 V	5 V Linear Regulator	+4.9 V
+3.3 V	3.3 V Linear Regulator	+3.2 V
+5 V REF	5 V Reference	+4.9 V
CS1 – Pin 4	Voltage Check (Int Temp)	+11.8 V
J3	Sensor Supply Current	1 mA
CS2 – Pin 4	Voltage Check (Ext Temp)	+11.8 V
J4	Sensor Supply Current	1 mA
J2	Voltage Check (Humidity)	+4.8 V
CS3 – Pin 4	Voltage Check (Pressure)	+11.8 V
U2 – Pin 2, 13	Voltage Check (RTC)	+3.1 V
U1 – Pin 1, 2	Voltage Check (MPU-6050)	+3.1 V
AD0	MPU6050 Jumper on Board	+3.1 V

6.1.1 Temperature Sensing Calibration Testing

The external and internal temperature 1N457/A sensor outputs have been tested and calibrated for incorporation into the Arduino IDE code. This testing was done by measuring the decimal output from the 10-bit analog/digital (A/D) conversion done by the Arduino Mega using a simple Arduino IDE analog read, then logging to a microSD program. The decimals were compared to temperature measurements (°C) by a calibrated thermocouple module. The purpose of this experiment was to establish the calibration equations to be used for the temperature sensors within the Arduino IDE flight coding to convert from the decimal readings into readings in degrees Celsius. These calibration equations were determined by using an inverse linear trendline function on Microsoft Excel charts feature, where y = temperature and x = decimal number. See Figures 45 and 46 in Section 7 for the calibration graphs for the external and internal temperature sensors.

The calibration procedure involved the placing of 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 while comparing the sensors outputs to Type-K thermocouples which were read by a Kamtop Digital Thermometer. Circuit filter offset and gain potentiometers were be adjusted initially to improve the sensitivity for a range of -80°C to 50°C.

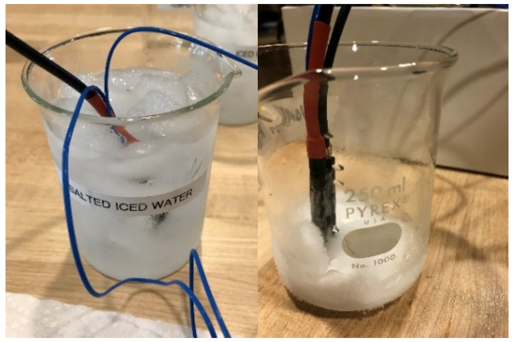


Figure 33: Temperature Sensor Calibration Setup

6.1.2 Pressure Sensing Calibration Testing

The pressure sensor module was tested by measuring the decimal output from the 10-bit A/D conversion done by the Arduino Mega using a simple Arduino IDE analog read, then logging to a microSD program while the payload was placed into a vacuum chamber with an attached pressure transducer. During a test, vacuum was applied using a single state vacuum pump and adjusted by 5 in Hg incremental stages between -5 and -30 in Hg to sync the sensor output with the pressure gauge. The resulting comparison was then graphed and a trendline made to be used in the Arduino IDE program.

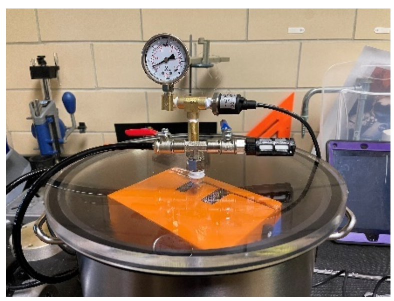


Figure 34: Pressure Calibration Setup

To calibrate the pressure sensing system to record pressure in absolute millibars, the comparison data from the Pace Scientific P1600-VAC-3000 pressure gauge was converted from inches of mercury vacuum gauge readings to absolute pressure millibar readings. Table 12 shows the conversion equivalents between the gauge and absolute pressure readings for the expected range of vacuum pressure testing.[26]

Table 12: List of Pressure Conversions Used for Pressure Calibration Data [26]

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 Calibration Testing

Since an accurate measurement of humidity was essential to the scientific requirements, a redundant sensor with no signal output conditioning has been added to the DemonSats-2 payload design. Therefore, there were 2 humidity sensors to calibrate and test. The purpose of the initial testing was to adjust the signal conditioning circuit of the MegaSat humidity sensor to achieve the best sensitivity within the 0 – 100% required range. The procedure used to do this testing was the same as described in Section 4.3.1.4. To describe briefly, the HIH-4000-003 sensor and the circuit voltage output was compared to the %RH readings from the HOBO External Temperature/RH Sensor Data Logger (MX 2302A) and the VLIKE MS6508 Digital Temperature Humidity Meter while all the probes were in either Lithium Chloride (~11% RH) or in the cool mist of a reptile fogger (~100% RH). The filtering circuit offset and gain potentiometers were adjusted to the best sensitivity and range, where the voltage outputs of the sensor and circuit should yield 0 V for 0% RH and 4.5 V for 100% RH. The voltage output of the circuit was measured using a Vernier LabPro and Vernier Voltage Sensor.

Figure 35: Humidity Calibration Setup

Once the MegaSat humidity sensor signal conditioning was set, both humidity sensors on the DemonSats-2 payload were then placed into the air above saturated salts (see Table 13) together with the HOBO External Temperature/RH Sensor Data Logger (MX 2302A) and the VLIKE MS6508 Digital Temperature Humidity Meter. Two calibrated devices were used since the salt chamber lids were built to house both the devices and the payload sensors, so they both needed to be used to maintain an airtight fit during testing. For the lithium chloride chamber, the sensors were placed into the jar for at least an hour, and for the higher %RH chambers for at least 30 minutes since equilibrium was reached faster.

Table 13: List of Saturated Salts Used in Humidity Sensor Calibration [18]

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 Chloride ($NaCl$)	75.09%± 0.11
Potassium Sulfate (K_2SO_4)	97.30% ± 0.45

During calibration testing, measurements were made of the decimal output from the 10-bit analog/digital (A/D) conversion done by the Arduino Mega using a simple Arduino IDE analog read, then logged to a microSD program. The decimals were compared to %RH readings of the calibrated devices. The purpose of this experiment was to establish the calibration equations used for the humidity sensors within the Arduino IDE flight coding to convert from the decimal readings into readings in %RH. These calibration equations were determined by using linear trendline function on Microsoft Excel charts feature, where y = %RH and x = decimal number. The calibration results have been included in Section 7.

6.1.4 Altitude Sensing System Testing

Altitude on the MegaSat design was determined from the three-dimensional GPS data (GPGGA) obtained from the Adafruit Ultimate GPS module. Because of the nature of its determination, the altitude was only observed and reported by the team. Altitude data was obtained from automobile ground travel from Natchitoches to Shreveport on a sunny day and compared to GPS tracker elevations from an iPhone app (GPS Tracks Pro, Version 3.6.6 by DM Software Solutions, LLC). The altitude is expected to vary from 25 m to 100 m above sea level for the trip taken.

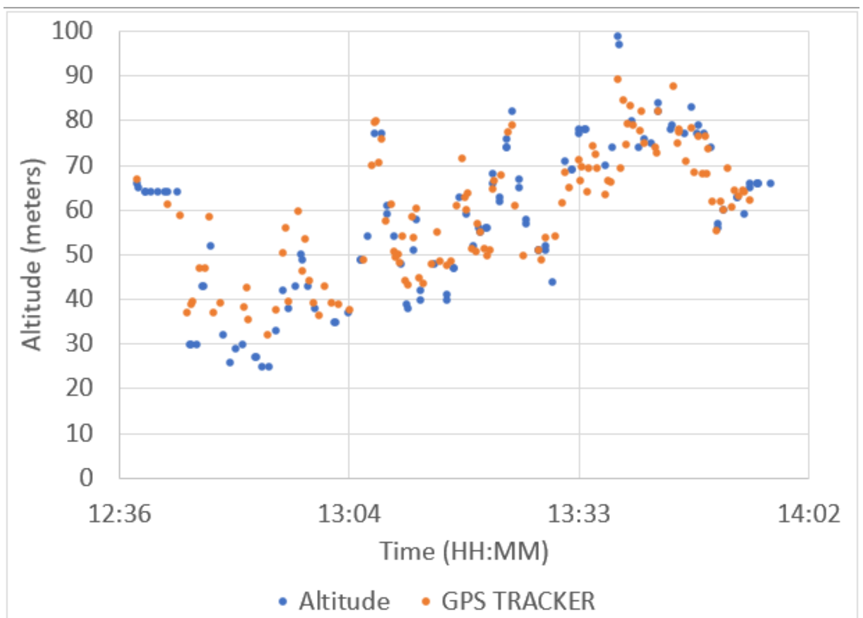


Figure 36: The Road Trip Data Comparing the Adafruit Ultimate GPS Module's Altitude Measurement with the Coordinates iPhone GPS Tracker Elevation data

During the road trip, the altitude varied from 25 m to 98 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 8.58 m on the time-synced data points. 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. However, experience with the DemonSat-1 payload results showed that this data collection irregularly does not cause a problem in the final data analysis, so the team determined that the altitude testing worked within the requirements.

6.1.5 GPS Latitude/Longitude Sensing System Testing

Latitude and longitude were determined from the two-dimensional GPS data obtained using the Adafruit Ultimate GPS module. Because of the nature of its determination, these directional coordinates were only observed and reported by the team. Unlike the altitude data, the latitude and longitude data was reported by the GPRMC and the GPGGA NMEA sentences. Therefore, the team expected to receive data with a 10-second resolution during testing.

The GPS latitude and longitude data was collected from automobile ground travel from Natchitoches to Shreveport on a sunny day. The data was stored onto the microSD card in the degree decimals (DD.DDDD) format. The latitude and longitude coordinates for the GPGGA data were mapped using the Microsoft Excel add-on function "3D Maps." The 3D Maps add-on overlaid the coordinates onto a Bing map, and the team used the smaller GPGGA data set to verify the accuracy of the map path made during the testing to determine the effectiveness of the Adafruit Ultimate GPS module selected. If the measurements reported by the GPS module were off by more a mile consistently, then other available module boards could be used to replace it.

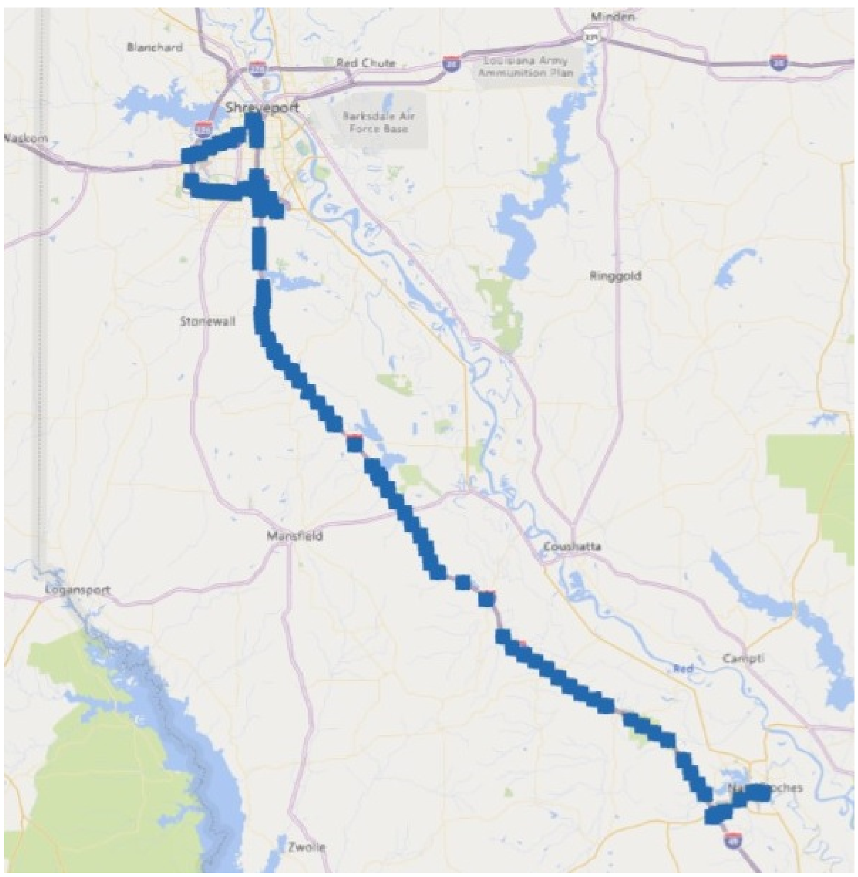


Figure 37: Data from the Adafruit Ultimate GPS Module Collected on a Road Trip from Natchitoches to Shreveport

The GPGGA data were very accurate (within 10 m) when compared to the iPhone app GPS Tracks Pro (version 3.6.6 by DM Software Solutions, LLC). The average reported accuracy of the iPhone GPS app was ±5 m.

6.1.6 MPU 6050 Accelerometer/Gyroscope Sensing Calibration Testing

Since the accelerometer and gyroscope data was not needed to meet the scientific objectives and requirements for the DemonSats-2 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.

To calibrate still movement to zero, the 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 then Ax, Ay, Az = 0 g, 0 g, 1 g respectively for this orientation. Then the box was moved to have the Y-axis pointed upward to get Ay = 1 g, and after 15 minutes, and then the box was moved so that the X-axis was pointed upward to get Ax = 1 g for 15 minutes. Then offsets were applied in the Arduino code so that the average decimal output for each Ax, Ay, Az at vertical would equal 1g. The gyroscope data from the same experiment was used to determine the offset for the gyroscope sensor, so that the Gx, Gy, Gz would equal zero when the payload was at rest.

Table 14: List of Offset Values for the MPU6050 Data for the DemonSats-2

Sensor Orientation	Offset Value
Accelerometer X-axis (Ax)	-573
Accelerometer Y-axis (Ay)	324
Accelerometer Z-axis (Az)	1305
Gyroscope X-axis (Ax)	824
Gyroscope Y-axis (Ay)	-47
Gyroscope Z-axis (Az)	-96

6.2 Integration Plan

The power, sensor, IMU, RTC, GPS, and storage subsystems were all designed to communicate via the Arduino IDE and operated by the Arduino Mega 2560 microcontroller. A flight code was created with analog sensor decimal-to-measurand conversion, as well as serial GPS, and I2C RTC and MPU-6050 data collection and the data was stored to a microSD card. Initially each of the subsystem programing was developed and tested for operation. The GPS Soft Serial code from last fall's capstone was used as the main architecture for the flight code. This capstone project code also included the microSD logging programing, as well as a temperature and pressure analog read code. The rest of the analog sensors required basic analog read programming to be incorporated, along with the decimal-to-measurand conversion math coding. The data conversions were in the following units: pressure (millibar), temperature (°C), and humidity (%RH). The flight IDE program also included coding to convert the GPS reported date to Central Daylight Time (CDT). The GPS data was reported by default in Greenwich Mean Time (GMT). Finally, the I2C programming was incorporated to communicate with the DS3231 RTC and MPU6050. Most of the coding done beyond the capstone project was done by the programming manager.

The team could not travel to LSU's Baton Rouge campus for the thermal and vacuum testing due to the ongoing COVID-19 pandemic. Instead, the team watched the LSU Thermal Vac Test via Zoom, while the team performed independent thermal and vacuum tests at the Northwestern State University, Biological & Electrical Applied Physics (BEAP) lab under the director of Prof. Dugas. The thermal portion of the test was done by placing the complete payload into warm air (~30°C), a conditioned room (~20°C), a fridge (~4°C). standard freezer (-20°C), and ultra-subzero freezer (-80°C). The vacuum portion of the test was performed in an ABLAZE stainless steel 5-gallon vacuum chamber and the pressure was tested between 0 – 1014 absolute millibars. The effects of the thermal and vacuum tests on the humidity sensors were also observed. More detail on the system testing procedures is included in Section 6.4.

6.3 Flight Software Implementation and Verification

The integrated flight code has been designed to collect the data from all the sensors into a single line of comma-delimited string for each time point, which was appended to a text (txt) file and stored on the microSD card. The data strings were reported onto the Arduino IDE Serial Monitor when the payload was linked to a PC to get a real-time verification of data during initial testing. Most of the details for implementation of the flight software on the Arduino Mega 2560 have been explained in Section 6.2. The system tests described in Section 6.4 provided the sufficient opportunity for software verification prior to flight.

6.4 Flight Certification Testing

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

Temperature fluctuations that might damage the electronics has been tempered by an insulated foam box with extra insulation around the battery pack. The payload has been placed in several low temperatures via -80°C and -20°C freezers and a refrigerator for 1/4-hour increments. The data obtained on the microSD during these tests was analyzed to determine proper functioning of the payload and battery pack.

Pressure fluctuations has been simulated in a small, portable vacuum chamber. To test the vacuum stability, the DemonSats-2 payload has been exposed to vacuums of near 0 to 1014 abs. millibars. The data recorded on the microSD card was analyzed to ensure proper functioning of the payload.

The measurement of humidity fluctuations have been tested by exposing the payload to changing humidity conditions. Lower humidity has been generated and measurement observed during the air evacuation in the vacuum pressure testing. The fluctuating of humidity has also been observed during the cycling of the refrigerator and standard freezer during their operational cycles.

The windy and rotational fluctuations have been 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 performed with measurements. Instead, a vigorous shaking in the 3-axial directions and a 10-ft drop of the payload was performed to ensure that it would still make measurements afterwards and that the payload remained in its designed position within the box.

6.4.1 System Testing Procedures

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 and 4.6 for more details).
- The box with the payload was placed alongside either the Onset HOBO and InTemp dataloggers (MX2302A and CX603) while in different temperature environments: warm air (~30°C), a conditioned room (~20°C), a fridge (~0°C). standard freezer (-20°C), and ultra-subzero freezer (-80°C) for 15 mins each
- The payload collected and recorded data to the microSD card, which was retrieved from the microSD after testing was complete and imported into Microsoft Excel.
- The data logged during the experiments was analyzed to ensure that the payload withstood the thermal conditions that it would likely experience in flight without affecting its integrity or functionality.
- The internal and external temperature data were reported and compared to temperature readings from either the Onset HOBO and InTemp dataloggers (MX2302A and CX603) which was placed alongside the payload during the system test.
- Humidity data was also reported and compared to the %RH readings of the Onset HOBO MX2302A for the data from the standard freezer and warmer system tests.

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 and 4.6 for more details).
- The box with the payload was placed into a small, stainless steel vacuum chamber with the Pace Scientific P1600-VAC-30 pressure transducer attached and the Onset HOBO MX2302A datalogger was set inside with the payload.
- The vacuum pump attached to the chamber was turned on in a place with adequate ventilation, and pressure was vacated at 5-in Hg increments until full vacuum (-30 in Hg).

Then the vacuum was released through a valve slowly or in increments. Each increment change was separated by 2 minutes to ensure sensor stability at constant pressures.

- The payload collected and recorded data to the microSD card, which was retrieved from the microSD after testing was complete and imported into Microsoft Excel.
- The pressure and humidity data logged during the system test was analyzed to verify that the payload can withstand the low pressure and humidity conditions which it may experience in flight without affecting its integrity or functionality.
- The pressure data was collected and compared to the inches Hg readings on the vacuum chamber's Bourdon tube and the readings from the Pace Scientific P1600-Vac-30 pressure transducer.
- Humidity data was collected and compared to the %RH readings of the Onset HOBO MX2302A which was placed alongside the payload during the system test.

6.4.1.3 Shock Test Procedure

- The payload was placed inside an insulated foam box with extra insulation around the battery pack (See Section 4.5 and 4.6 for more details).
- The lid of the enclosure was taped to the enclosure bottom with painter's tape to secure it.
- With the payload operational, the box was shaken back and forth vigorously by hand in the 3 axial directions, X, Y and Z.
- The lid was removed, and observations were made on the positioning of the electronics within the box after the shake test.
- The microSD card was retrieved after testing to see if data had been stored.
- Then lid of the enclosure was taped to the enclosure bottom with painter's tape to secure it for a drop test.
- The operational payload was dropped from 10 feet onto the concrete lab floor.
- The lid was removed, and observations were made on the positioning of the electronics within the box after the drop test.
- The microSD card was retrieved after testing to see if data has been stored.

Figure 38: The Payload Drop Test Setup

6.4.2 System Testing Results

6.4.2.1 Thermal Test Results

The thermal testing consisted of placing the DemonSats-2 payload into different controlled temperature environments for 20 mins. Figure 39 showed the full system testing data which was collected during the thermal test done on April 16, 2021. The payload's external sensor was directly compared to a commercial Onset datalogger (InTemp Dry Ice for -80 freezer, HOBO MX2302A for rest) during the thermal test. The team determined that the temperature reported by the external temperature sensor was near or over the 5 ^{o}C error in the upper and lower regions of the expected temperature range. Therefore, the team recalibrated the temperature sensors and repeated the system test on April 27, 2021 (see Figure 40).

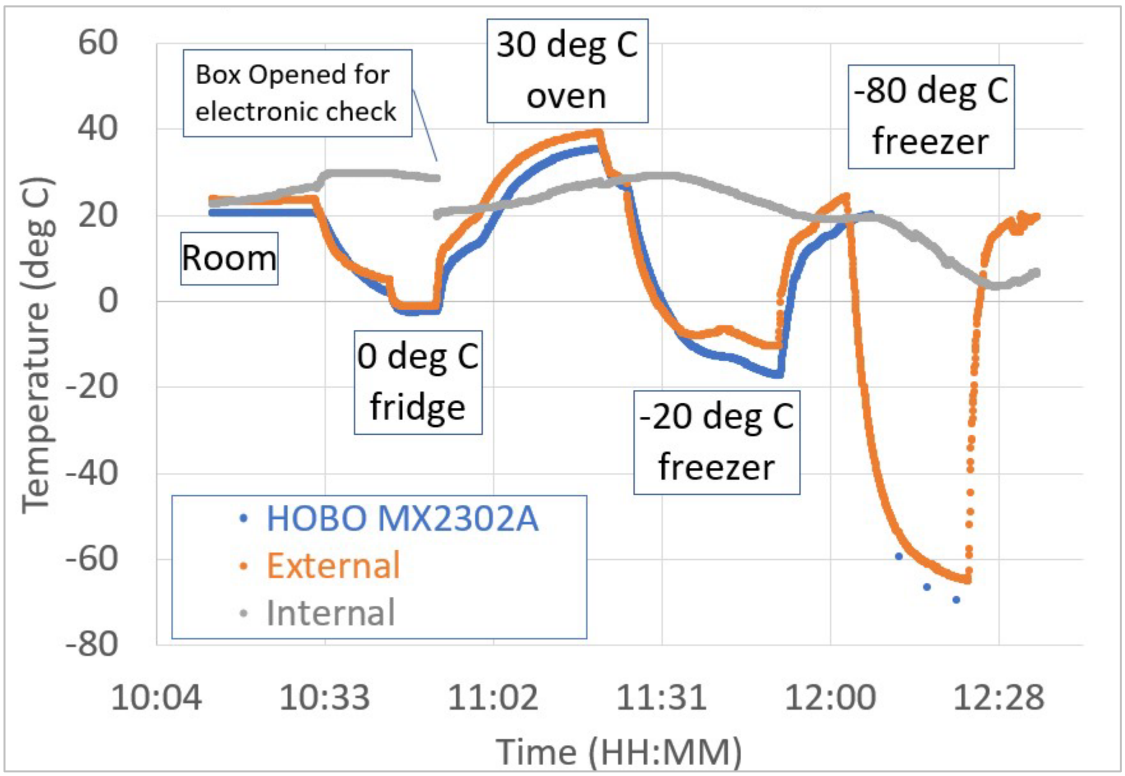


Figure 39: The Temperature Sensor Output for the Thermal System Test (completed on April 16)

After the recalibration, the external temperature sensor reported data to be within ±2 ^{o}C error once the sensor and the Onset datalogger reached thermal equilibrium. The larger difference between the sensors were a result of the Onset HOBO datalogger having a slower response time during large drops in temperature. The internal temperature was independently compared with the HOBO (data not shown) and was also found to be within the 5 ^{o}C error requirement. During the system test, the internal temperature successfully monitored the increase of temperature in the box due to the electronics radiating heat energy. During the thermal system test, the internal temperature sensor detected that the box's internal temperature ranged from 3 ^{o}C to 38 ^{o}C.

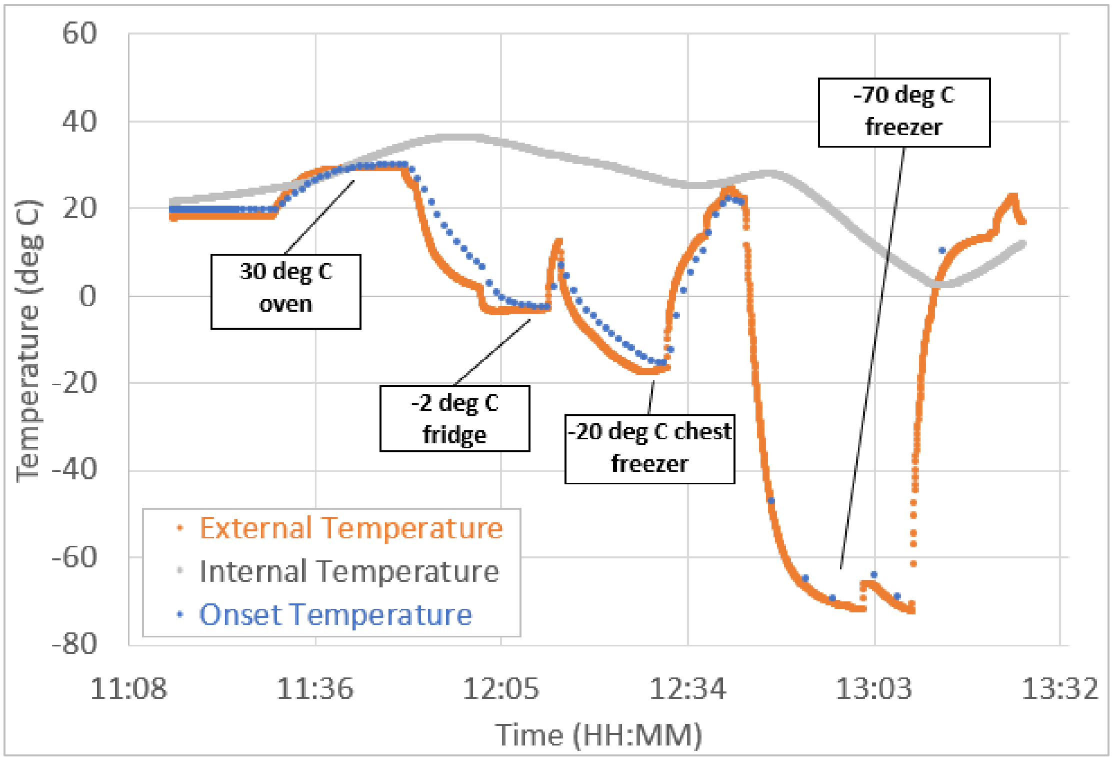


Figure 40: The Temperature Sensor Output for the Final Thermal System Test

The thermal system testing results were also reviewed for the humidity sensors. Moving the payload between the dry climates of the oven and freezers to the more humid climates of outside, a room and the fridge allowed the team to observe the sensitivity of the humidity sensors when compared to the Onset HOBO MX2302A. The percent relative humidity for all the sensors where within 10% RH except for at the end of the fridge test where a condensation drop was found on the HOBO probe, making its %RH readings go up until it dried.

The payload humidity sensors were placed to face inward toward the box foam inside screened-in windows to reduce condensation. The humidity data from the thermal test showed that the sensors maintained their sensitivity and ability to read from low %RH to 100% RH, while they were able to recover from the initial condensation which occurs on the box when it was removed from cold environments and brought back into the humid lab room. On the right side of the data on Figure 41 was where the longest delay for the condensation to occur, which was 10 mins. The box was observed to be wet for that 10-minute period as well. Once the box dried, it was observed that the humidity sensors reported a drop in percent relative humidity as well.

Coatings for the humidity sensor were explored, but found that all coating tried (loosely-wrapped heat shrink, masking tape, cloth sock) either stopped the HIH-4000-003 sensor from working or only make the condensation delay in sensing actually air %RH worse. Therefore, in spite of the box condensation causing the delay in the HIH sensor's ability to sensor the actual air %RH, placing the sensors uncovered and facing toward the box foam was the best alternative found.

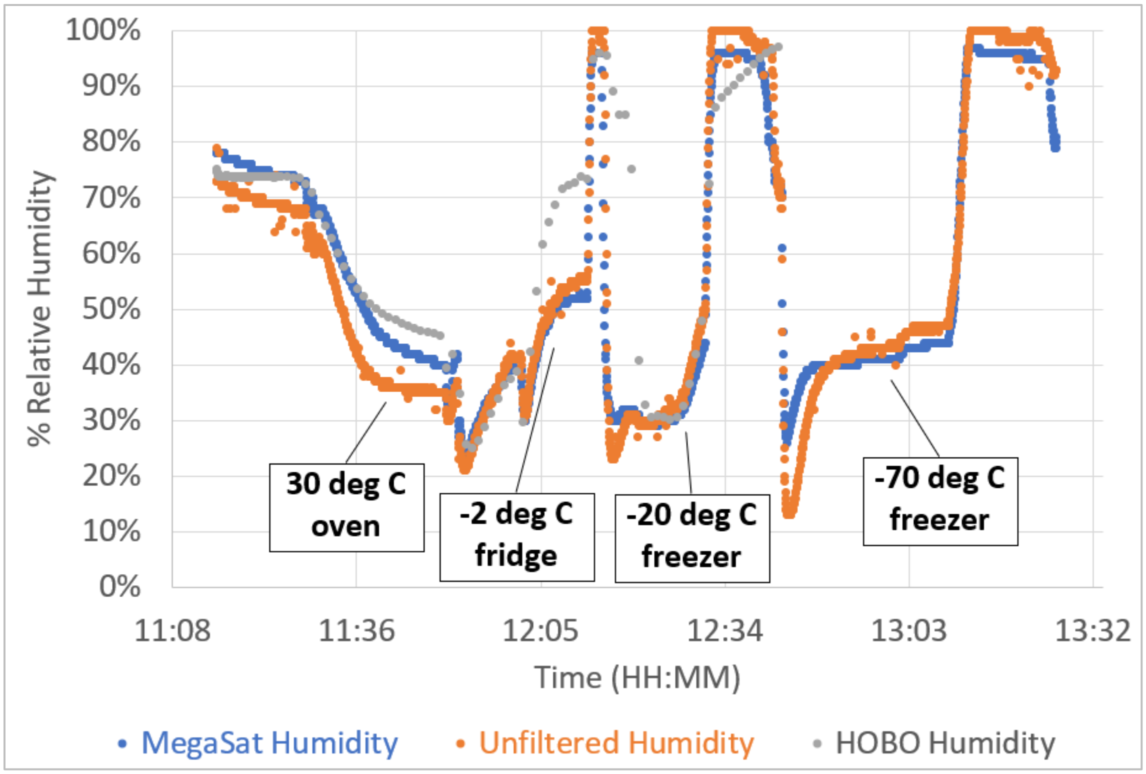


Figure 41: The Humidity Sensor Output for the Final Thermal System Test

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 NSU. The pressure sensing system response during testing was compared to a Pace Scientific P1600-VAC-30 pressure transducer as described in 6.4.1.2 and displayed in Table 12. The pressure measurement was within 1% for the MegaSat pressure sensing system during the system testing, which met the 5% science requirement for pressure (see Figures 42 and 43).

The detection of humidity fluctuations by the DemonSats-2 payload during the vacuum system testing was also explored. Figure 44 showed that all the humidity sensing systems were able to detect changes in %RH as the air was gradually evacuated from the test chamber. During the vacuum test, the %RH was also measured by the Onset HOBO MX2302A datalogger (see Figure 44). The external humidity sensors of the payload trended very closely (~3%) with each other and were mainly less than 10% RH from the HOBO datalogger. The response of the HOBO while at low pressures was suspected to be off since the complete datalogger was exposed to low pressures and not just the sensing probe. An Onset technical support representative concurred with the team's suspicion and expects the datalogger to have greater error (-10% RH error) when in low pressure environments.

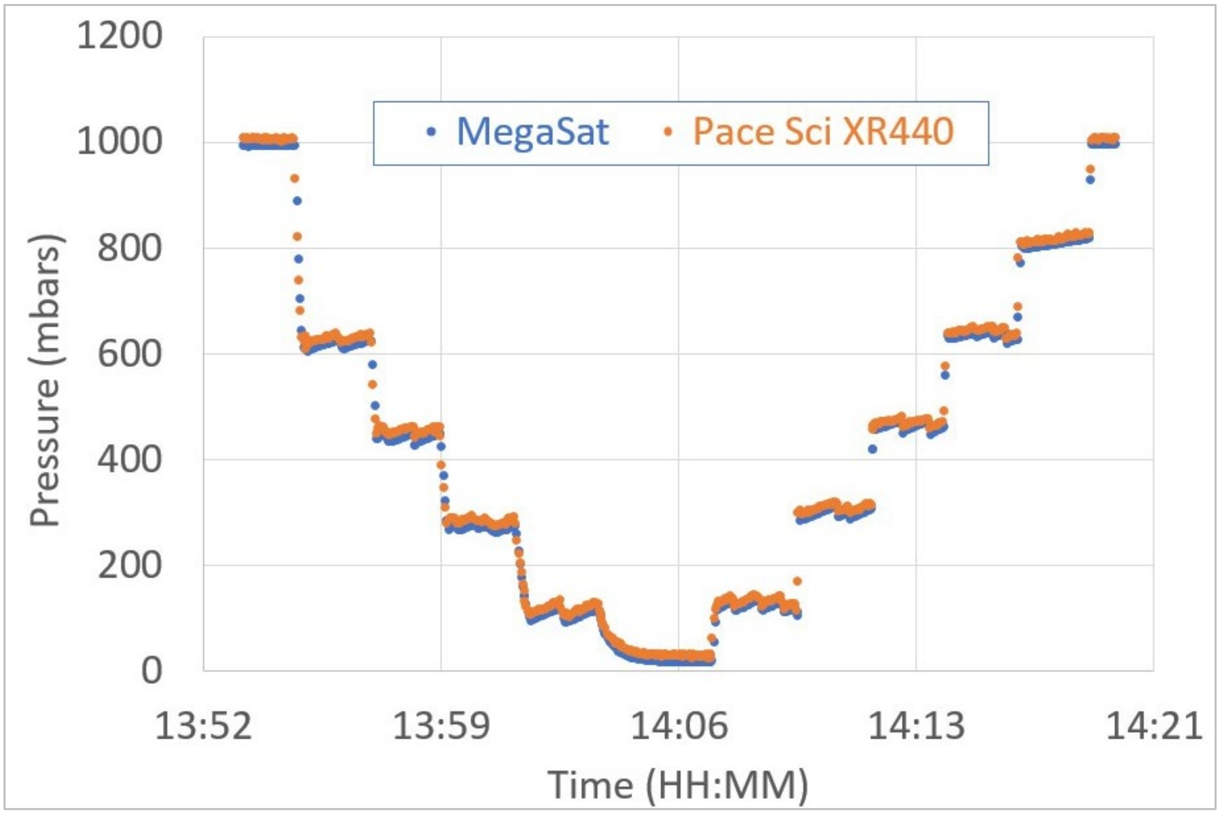


Figure 42: The Pressure Sensor Output for the Vacuum System Test (completed on April 16)

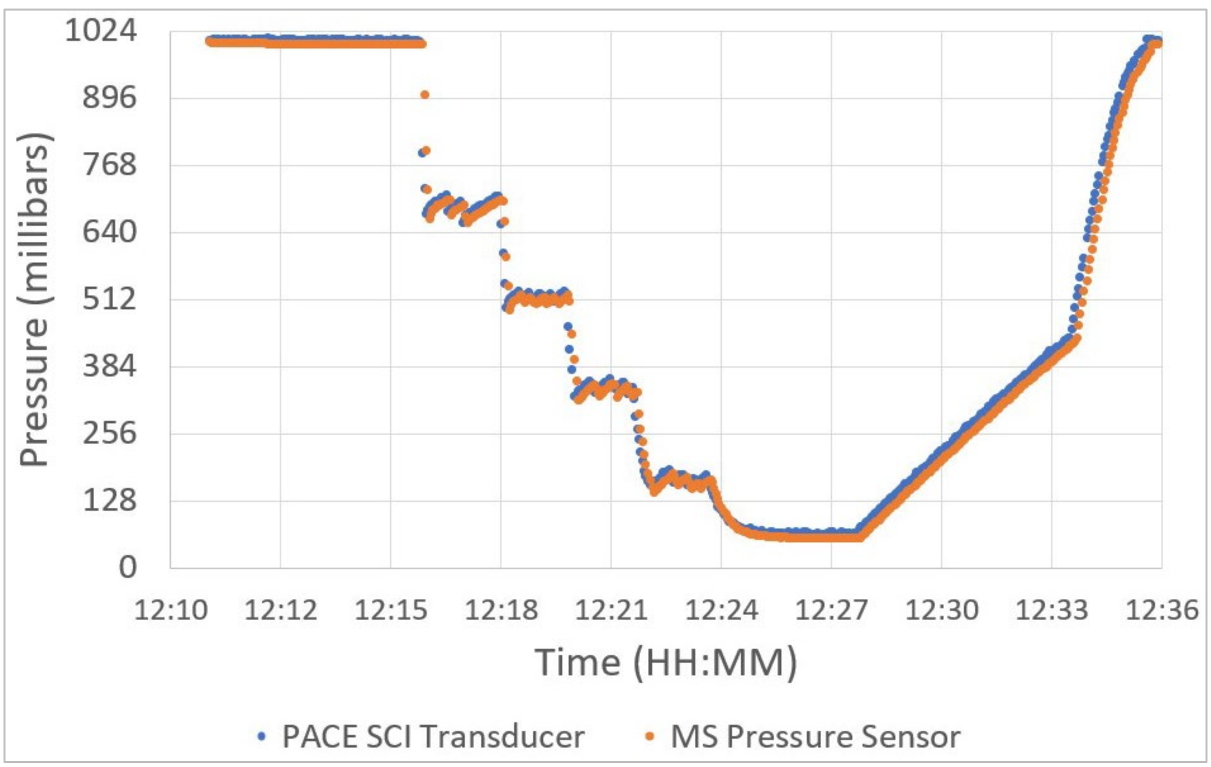


Figure 43: The Pressure Sensor Output for the Final Vacuum System Test

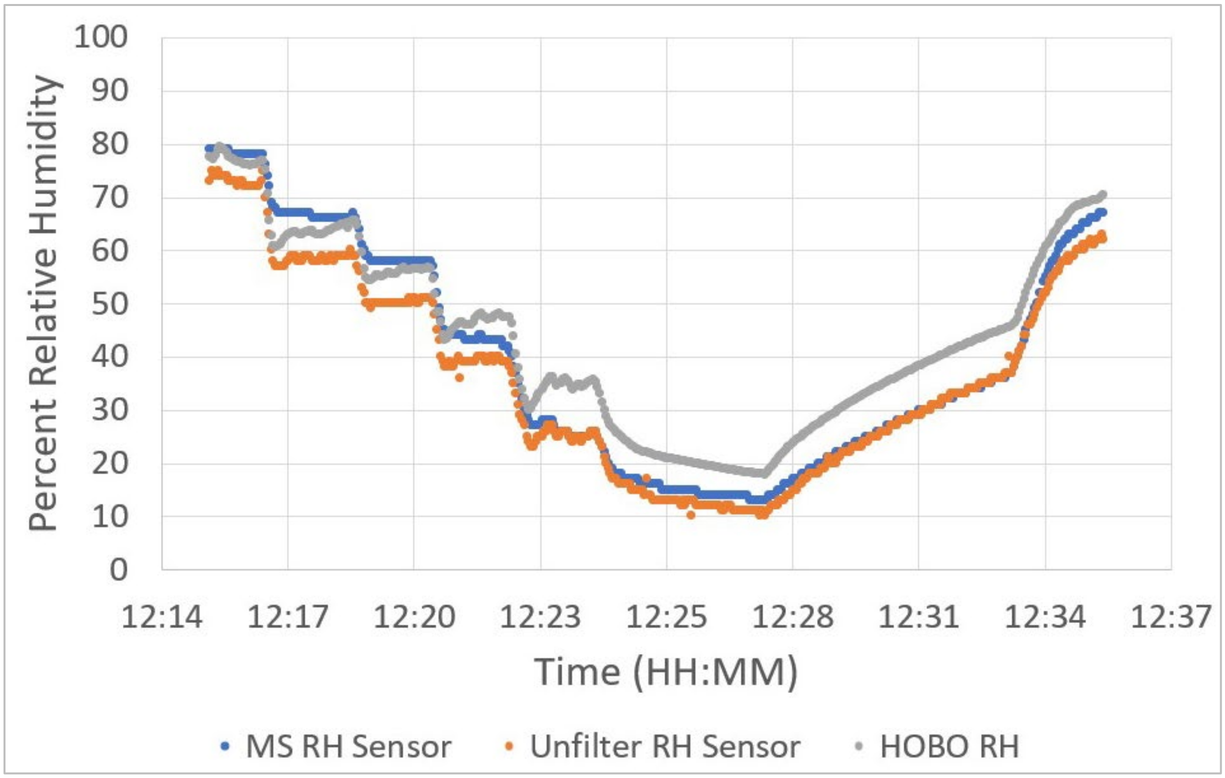


Figure 44: The Humidity Sensor Output for the Final Vacuum System Test

6.4.2.3 Shock Test Results

The box was shaken back and forth vigorously by hand in the 3 axial directions, X, Y and Z, while operational. Data was collected without issue. For the drop test, the enclosure was dropped 10 feet to the concrete lab floor to see if damage would occur. The enclosure did not sustain any damage.

7.0 Mission Operations

7.1 Pre-Launch Requirements and Operations

7.1.1 Calibrations

All the analog sensors (temperature, pressure, humidity) have been compared to calibrated devices. All the analog sensors had their 10-bit decimal outputs converted to meaningful measurements of their respective measurands. The GPS and RTC must be observed to be accurate, or the modules would need to be replaced. The MPU-6050 data must be adjusted to the payload's orientation and movement when compared to the coordinates provided on the sensor board. The steps to completing these calibrations were already detailed in Section 6.1.

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.
- ✓ Check if sensors and hose placement are correct.
- ✓ 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.
- ✓ Remove microSD card from payload.
- ✓ 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 45 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 0.9929. 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.

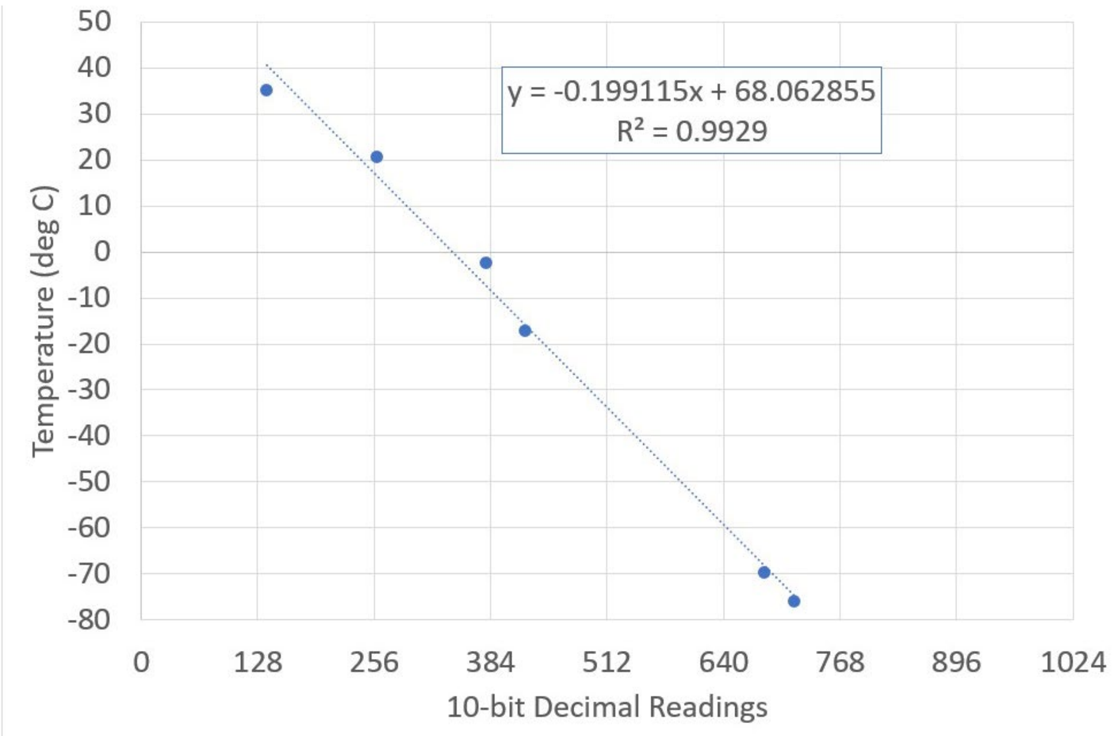


Figure 45: Calibration Results for External Temperature Sensor Compared With Onset HOBO MX2302A

7.1.1.2.2 Internal Temperature Sensor

Figure 46 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 an inverse linear regression and found to have an R^2 value of 0.99. 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.

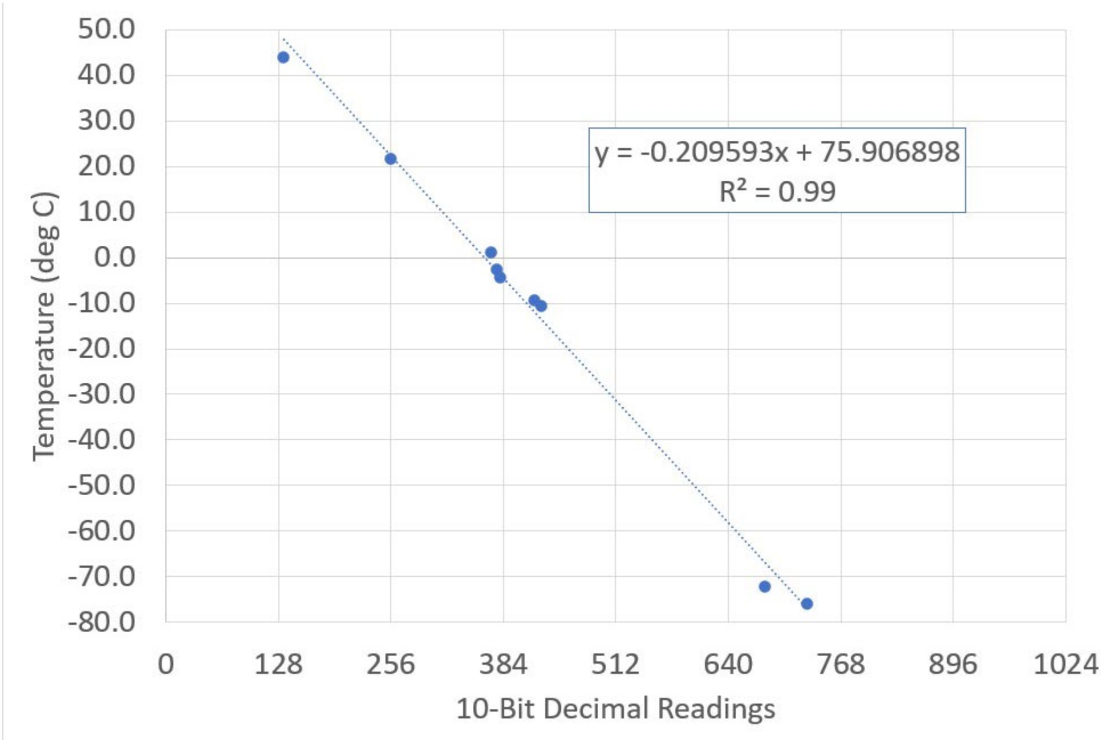


Figure 46. Calibration Results for Internal Temperature Sensor Compared With Onset HOBO MX2302A

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 the Onset HOBO MX2302A which was placed inside the jar.

The data was used in calibrating the humidity sensor to convert the 10-bit ADC readings into decimal relative humidity. The decimal relative humidity readings were plotted against the corresponding ADC values, with the graph shown in Figure 47. The correlation was calculated using a linear regression and found to have a R^2 value of 0.99. The sensor recorded values from 0-100% relative humidity (or 0 to 1 decimal), there by validating the filtering circuit redesign to meet our science requirements. The formula from the calibration curve in Figure 47 was inserted into the MegaSat programming. The sensitivity of the MegaSat humidity sensor was determined to be 0.001 or 0.1% RH.

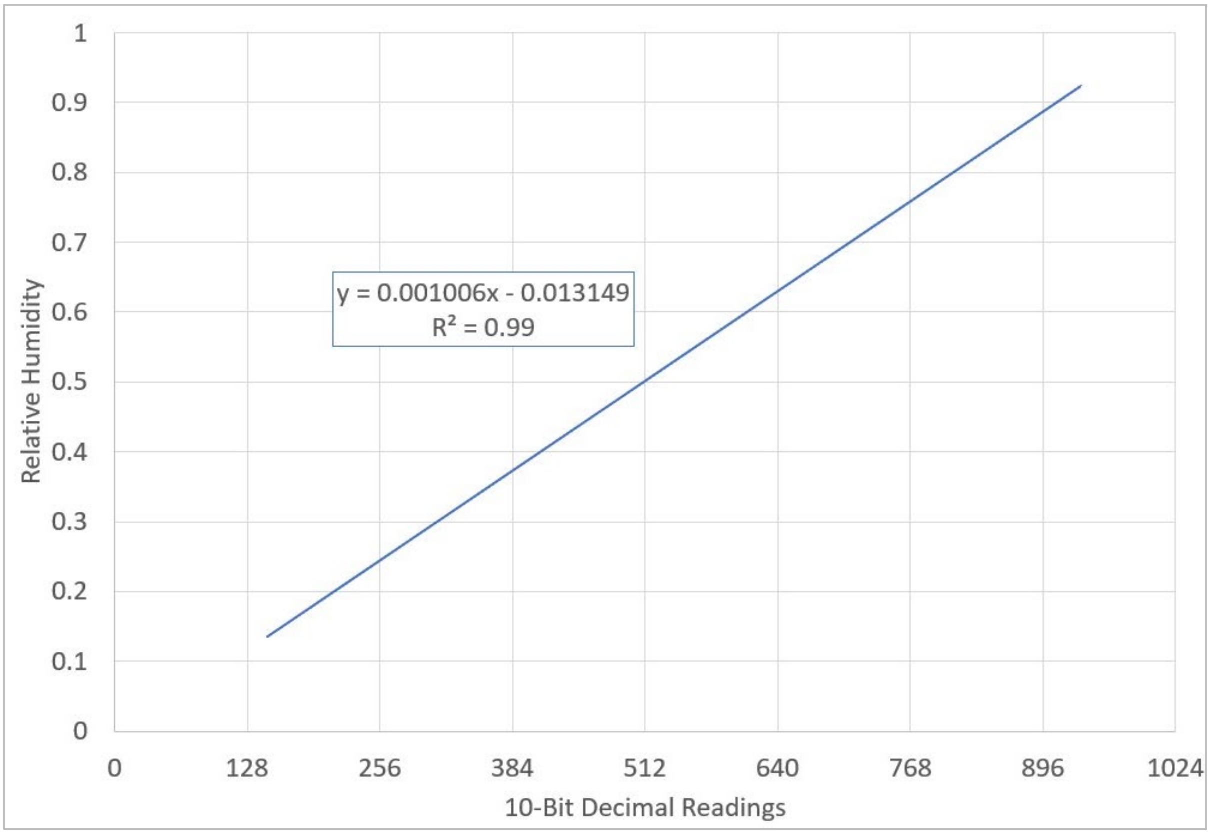


Figure 47. Calibration Results for MegaSat Humidity Sensor Compared With Onset HOBO MX2302A

7.1.1.2.4 Extra Humidity Sensor

Various saturated salt solutions were used in calibrating the unfiltered, backup 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 the Onset HOBO MX2302A which was placed inside the jar.

This data was used in calibrating the humidity sensor to convert the 10-bit ADC readings into decimal relative humidity. The decimal relative humidity readings were plotted against the corresponding ADC values, with the graph shown in Figure 48. The correlation was calculated using a linear regression and found to have a R^2 value of 0.999. This sensor showed less sensitivity (0.13%) than the MegaSat sensor, and 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 decimal relative humidity during flight.

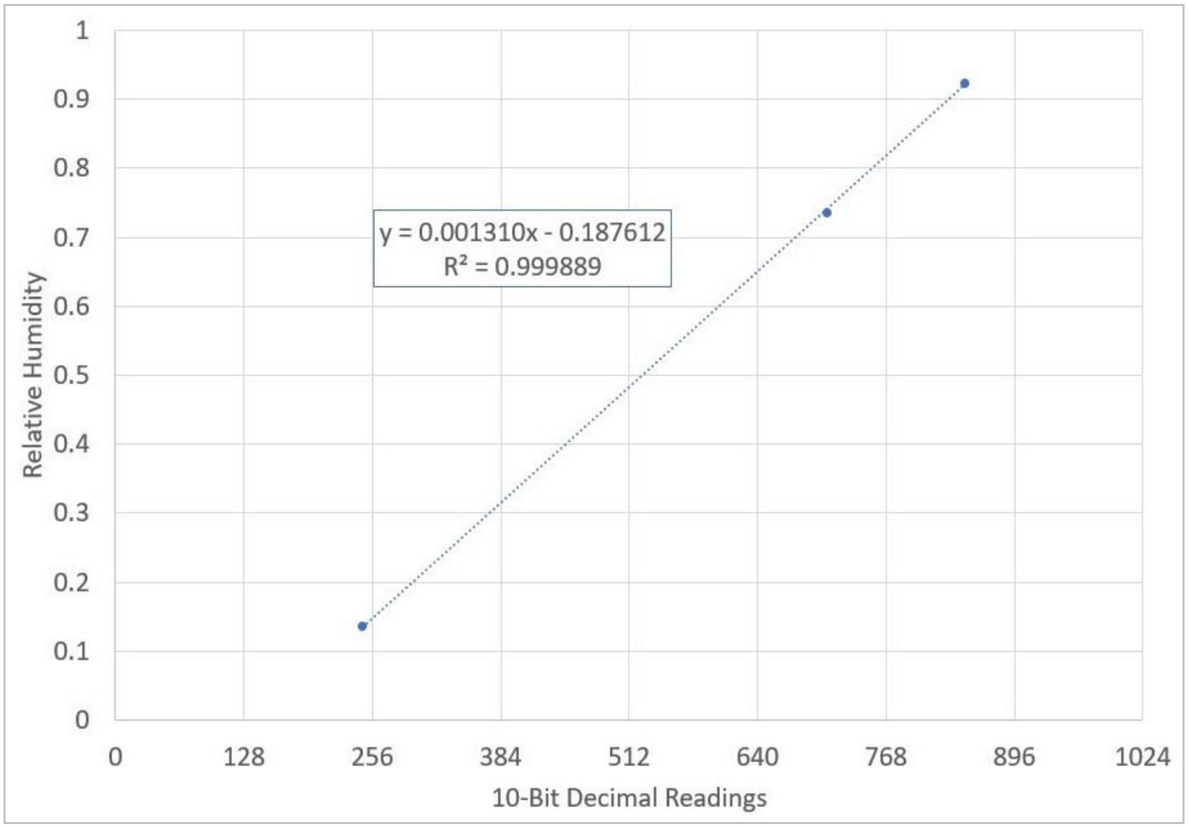


Figure 48. Calibration Results for Unfiltered Humidity Sensor Compared With Onset HOBO MX2302A

7.1.1.2.5 MegaSat Pressure Sensor

A vacuum chamber was used to calibrate the pressure sensor. The testing started at atmospheric pressure, and as the chamber was evacuated, the vacuum was increased in increments of 5 in Hg over the range of -5 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 and the data was compared to the Pace Scientific P1600-VAC-30 pressure transducer (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 pressure transducer was converted from inches Hg (gauge) to absolute millibars (see Table 12 for conversion factors), and the millibar values were plotted against the corresponding 10-bit ADC values. The resulting graph is shown in Figure 49. 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 0.999. Since the P1600 transducer has an uncertainty of ±2% f. s., the accuracy of the readings may vary by 20 millibars. The formula from the calibration curve was inserted into the MegaSat programming so that the 10-bit ADC values was converted to millibars during flight.

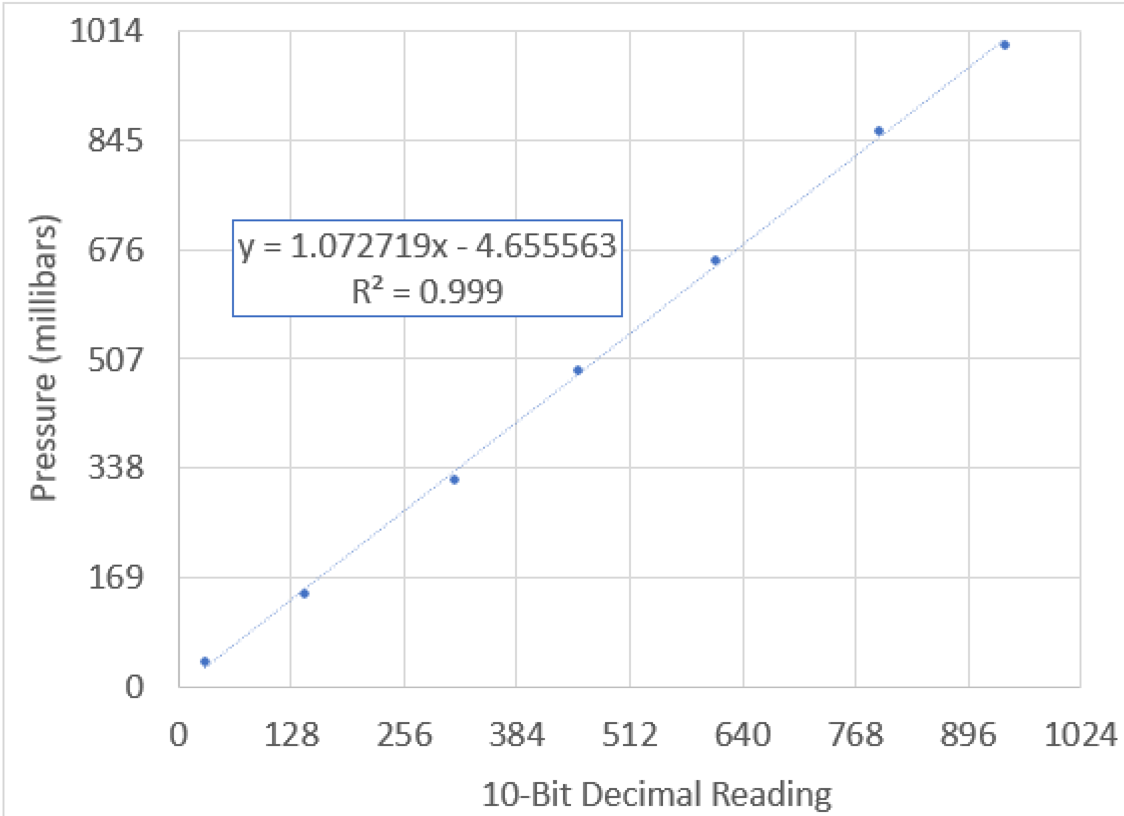


Figure 49. Calibration Results for MegaSat Pressure Sensor Compared to P1600-VAC-30

7.1.2 Pre-Launch Checklist

- ✓ Remove lid.
- ✓ Confirm microSD card in 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.
- ✓ Secure the lid with painter's tape.
- ✓ Follow LaACES team instructions for payload integration on to the flight string.
- ✓ Follow LaACES team instructions for balloon launch.

The total time needed before flight string integration will be less than 5 minutes if everything goes smoothly. The only remediation the team could try to do at the launch site would be to replace the battery, which would not take more than 5 additional minutes.

7.2 Flight Requirements, Operations and Recovery

The payload flight should take approximately two hours to reach completion. It is expected that the payload will be active for about four hours from powering on until recovery and decommissioning. During the flight, all data is expected to be stored on the microSD. No telemetry or external data will be transmitted from the payload. The LaACES team will oversee the launch, tracking, and recovery of the payload. The data obtained (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 the DemonSats-2 payload during flight. The LaACES team will oversee the launch, tracking, and recovery of the payload likely with little involvement by the team, therefore the only ground software for the payload will be the data analysis completed in Microsoft Excel with the flight data stored on the microSD card.

Calibrations will be performed and verified for each sensor prior to flight, and the equation from the resulting calibration curves will be directly inserted into the Arduino IDE code to convert the decimal numbers from the sensors into the proper sensor readings that will be stored on the microSD card. Once the data from the microSD has been received, the data will be then imported into Microsoft Excel and analyzed. Figures 50 and 51 provide the details on the data analysis process in Excel.

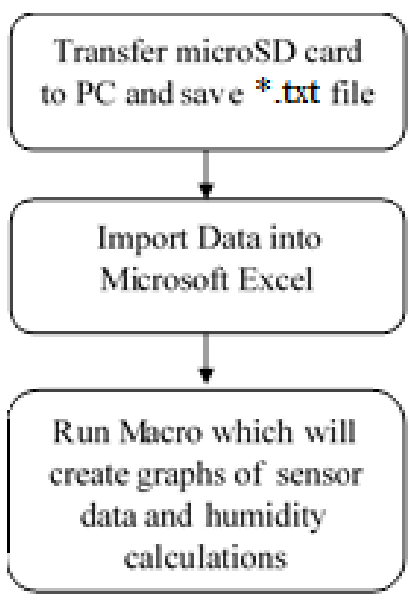


Figure 50: Post-Flight Data Analysis Flowchart

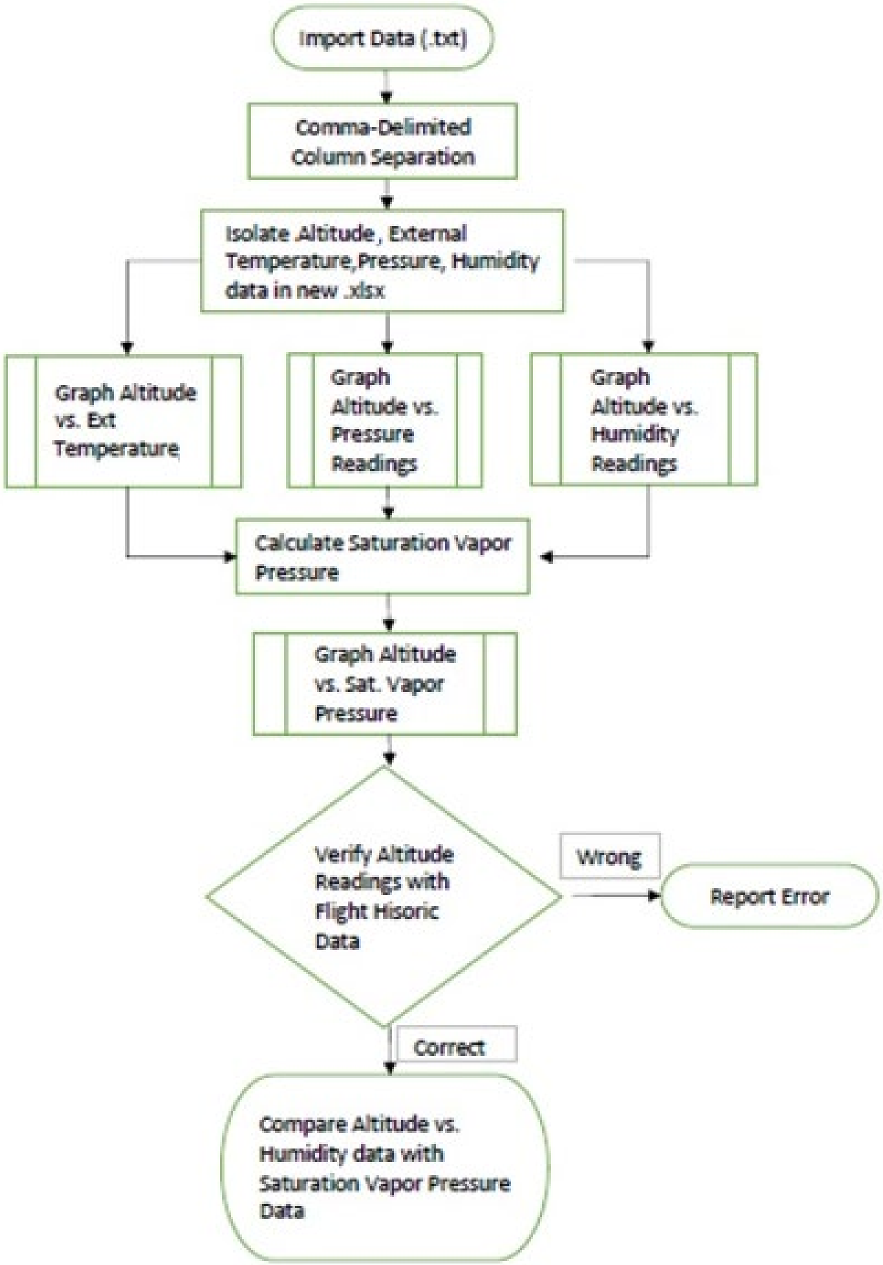


Figure 51: Post-flight Microsoft Excel Data Analysis Procedure

7.3.2 Ground Software Implementation and Verification

The only ground software for the DemonSats-2 payload is the data analysis completed in Microsoft Excel on flight data obtained from the microSD card. During implementation of the Microsoft Excel data analysis procedure (Figure 50), 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 flight data.

7.3.3 Data Analysis Plan

Once the data from the microSD has been received, the comma delimited data is imported into Microsoft Excel. The altitude, temperature, humidity, and pressure data will be placed on independent worksheets initially and then graphed versus time (from the GPS) to observe the flight trends, and to determine the data rows which occurred during flight. The flight only data with GPGGA NMEA sentences will be transferred to a new worksheet for analysis. Then the temperature, pressure, humidity will be graphed compared to altitude to observe any possible sensor issues detectable by the data output. Then the saturated vapor pressure will be calculated using Buck's Method by using the equations below (described in Section 3.3).

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

$$e'_i = \left[\mathbf{1.0003} + \left((\mathbf{4.18} * \mathbf{10^{-6}}) * \boldsymbol{P}\right)\right] * \mathbf{6.1115} * e^{\left[\frac{22.542 * T}{273.48 + T}\right]} \quad \text{for -80 to 0°C} \qquad \textit{(Eqn. 7)}$$

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

The saturation vapor pressure will then be graphed versus altitude. The humidity sensor readings will be compared to the saturation vapor pressure with respect to altitude to see if the actual vapor pressure could be determined using equation 1 from Section 3.3.

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

8.0 Project Management

The DemonSats-2 team will meet a minimum of once weekly on Wednesday and/or Friday afternoons for the team to participate in advancing the payload. The team will meet with the faculty advisor on Fridays at noon in-person and on Zoom for a project management update. Microsoft Teams is being utilized as the communication hub for sharing files such as the design documents. MS Teams is also used for direct messaging between team members and video calls. MS Project is being used to maintain efficient task completion and for establishing the WBS and Gantt chart.

8.1 Organization and Responsibilities

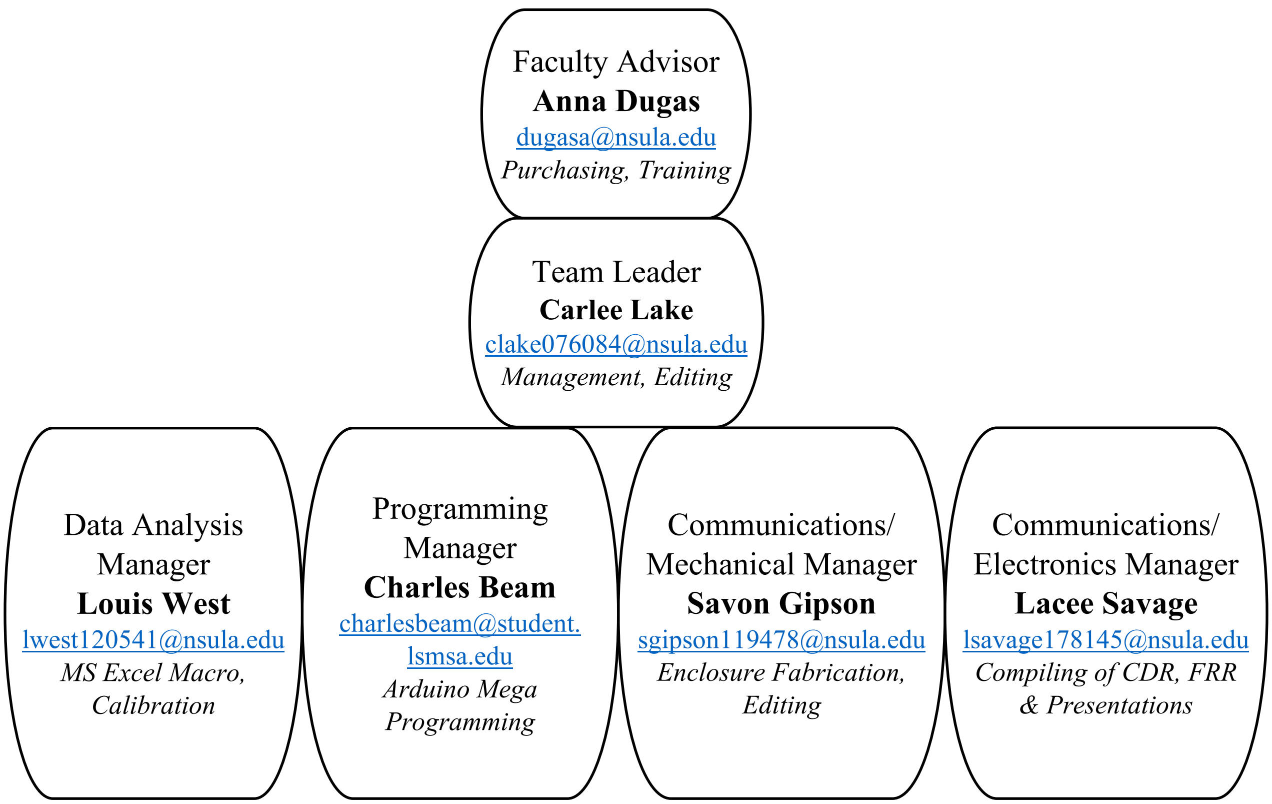


Figure 52: Team Organization Chart

8.2 Configuration Management Plan

All changes are first presented to the team leader before deciding to present the changes to the faculty advisor. Each member is responsible for initiating discussions about changes, issues encountered, and keeping notes about all sections in the MS Teams platform. The faculty advisor meets with the team at least once a week on Friday at noon for progress reports and updates on payload development.

8.2.2 COVID-19 Accommodations

8.2.2.1 FRR Presenting Authors

The DemonSats team will present their FRR defense via Zoom on May 17, 2021 between 10 am and noon. The presenting authors for the FRR will be all Carlee Lake and all other available team members.

8.2.2.2 Pre-Flight Procedures

- ✓ Remove lid
- ✓ Confirm microSD card in 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
- ✓ Check and humidity sensor windows, external temperature probe and pressure hose hole have not been blocked by tape

8.2.2.3 Payload Delivery

The payload will be hand-delivered by either Charles Beam or Anna Dugas in the week of May $17^{th} – 20^{th}$. Prior to delivery, the payload will be secured in a shipping box with extra protection around the payload and clearly labeled with the team's 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 person dropping off the payload will arrange a time make the drop off with a LaSPACE representative. When the drop-off person is approximately ten minutes from the LSU, he/she will call the number provided to obtain further instructions. When he/she arrives on campus, he/she will park in the LSU bookstore parking garage and will meet this representative to transfer the shipping box with the payload. Masks and guidelines for social distancing will be followed.

8.2.2.4 Flight Launch Attendee List

The team members can attend the launch in south Louisiana. The interested team members are shown in Table 23, along with their contact information and assigned roles.

Table 15. List of Team Interested in Attending the Launch

Priority Order	Team Member	Cell Phone Number	Proposed Flight Operations Role
1	Carlee Lake	318-206-9364	She serves as the team leader. She will assist with resolving any issues that may arise prior to the launch and post-flight data analysis
2	Savon Gipson	318-918-0781	He serves as the mechanical manager for this team. He will assist with resolving any issues that may arise prior to the launch with the box
3	Lacee Savage	832-221-8295	She serves as the communications manager for this team. She will assist in the post-flight data analysis and preparation for the final science report presentation.
4	Louis West	318-471-2281	He serves as the data analysis co-manager for this team. He will assist with final flight data analysis.
5	Charles Beam	225-916-4447	He serves as the data analysis co-manager for this team. He will assist with final flight data analysis.
6	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 in SD slot of payload through viewing window next to battery
- ✓ Verify that lights are still blinking
- ✓ 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
135 Sam Sibley Dr.
112 Bienvenu Hall
Natchitoches, LA 71497

8.2.2.8 Science Report Presenting Author

The presenting author for the FRR will be the available team who attended the launch. In the event of an emergency or other issue that prevents anybody from attending, the faculty advisor Anna Dugas has been designated as the alternate presenting author.

8.3 Interface Control

The Arduino IDE is used to interface with the Arduino Mega 2560 microcontroller, with the Mega interfacing with all sensors and data storing devices on the payload. The Arduino IDE has a serial monitor feature that is utilized to verify results of programming and calibrating the sensors. The serial monitor can be used to illustrate the character string that is sent to the microSD for real-time calibration results.

The Arduino Mega will be the sole controller of the payload interfaces during the time of flight. The flight balloon transceiver and the payload will have no direct communication while in-flight. Therefore, all the data collected during flight will be stored on the microSD card. After the payload is retrieved, the microSD card will be removed from the payload and the data on the card will be transferred to the team laptop for data analysis using Microsoft Excel.

9.0 Master Schedule

9.1 Work Breakdown Structure (WBS)

Table 16: Work Breakdown Structure for DemonSats-2 Project

1	**Project Management**
1.1	Team contract
1.2	*PDR due*
1.3	*CDR due*
1.4	*FRR due*
1.5	**Background**
1.5.1	**Scientific**
1.5.1.1	Temperature vs. Altitude
1.5.1.2	Pressure vs. Altitude
1.5.1.3	Sat. Vapor Pressure vs. Altitude
1.5.1.4	DemonSats-1 Data
1.5.2	**Technical**
1.5.2.1	Sensor Background
1.5.2.2	DemonSat-1 Problems
1.5.3	Objectives & Requirements
2	**Design Development**
2.1	**System Design**
2.1.1	Power Design
2.1.2	Sensor Design
2.1.3	RTC Design
2.1.4	GPS Design
2.1.5	IMU Design
2.1.6	Storage Design

2.2	**Electronics Design**
2.2.1	Real-Time Clock
2.2.2	Thermometer
2.2.3	Pressure
2.2.4	GPS
2.2.5	MPU-6050
2.2.6	Relative Humidity
2.3	**Flight Software Design**
2.3.1	Arduino IDE Programming
2.4	**Mechanical Design**
2.4.1	Enclosure Construction
2.4.2	Sensor Placement
3	**Device Prototyping**
3.1	Electronics Prototyping
3.2	Flight Software Prototyping
3.3	Data Processing & Analysis Plan
3.4	Mechanical Prototyping
4	**Flight Payload Fabrication**
4.1	Flight Payload Integration
4.2	Calibrations
4.3	System Testing
5	**Flight Operations**
5.1	*FRR Defense*
5.2	Launch
5.3	Data Processing & Analysis

9.2 Staffing Plan

Table 17: Staffing Plan for DemonSats-2 Project

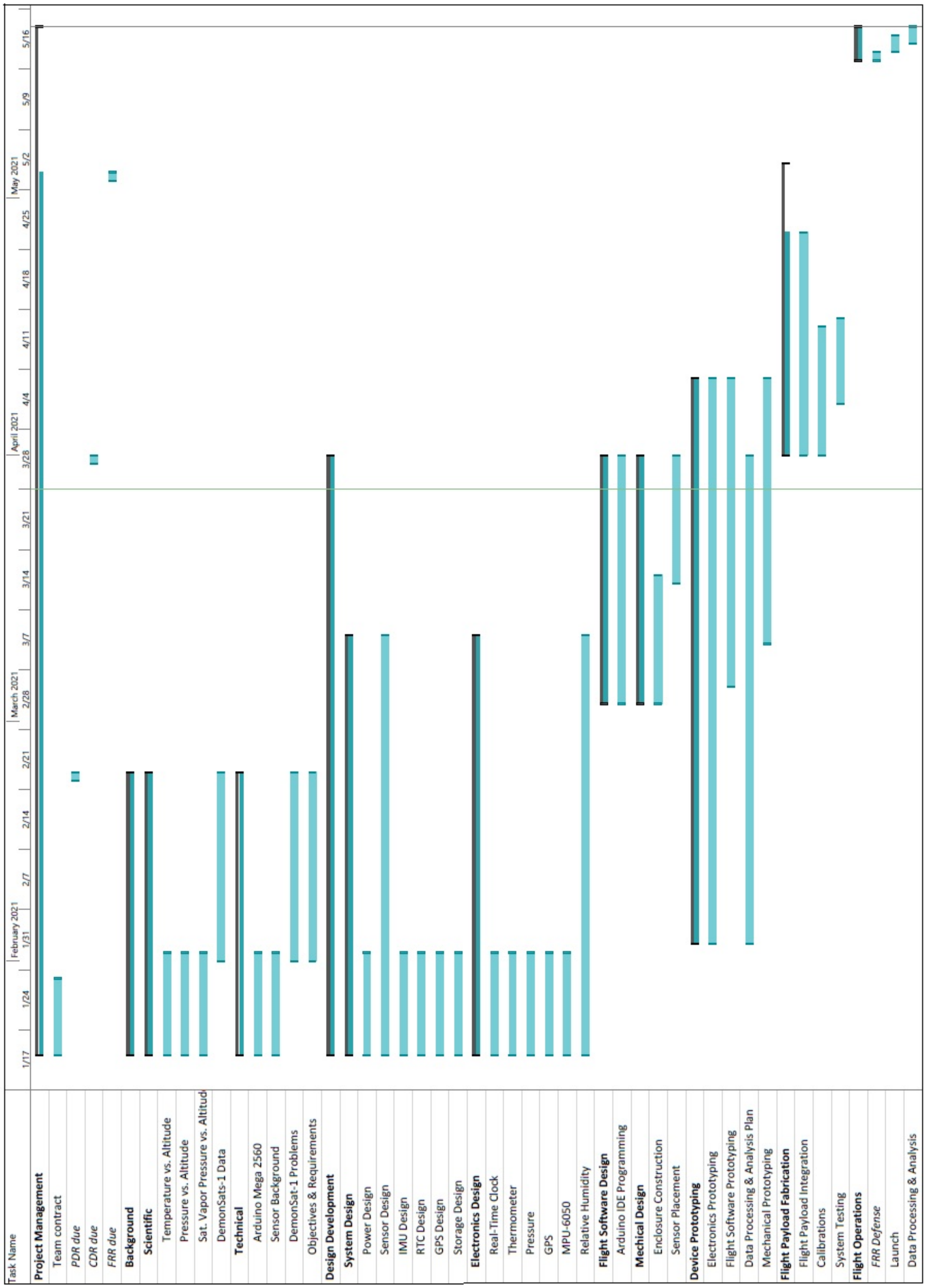
Team Member	Major	Previous Experience "Specialty"	Primary Interest in Project	Secondary Interest in Project	Assigned Role on Team
Carlee Lake	Biology	Writing, Presentations, Programming	Project Management	Communications	Team Leader
Charles Beam	HS	Programming	Arduino IDE Programming	Electronic Development	Programming Manager
Lacee Savage	EET	Electronic Troubleshooting, Circuit Building	Electronic Development	Communications	Communications/ Electric Development
Louis West	IET	Electronic Troubleshooting, Circuit Building	Data Analysis	Electronic Development	Data Analysis Manager
Savon Gipson	IET	CAD, Building Things	Communication	Electronic Development	Communications/ Mechanical Development

9.3 Timeline and Milestones

Table 18: Major Milestones

PDR	February 22, 2021
Pre-CDR	March 15, 2021
CDR	March 31, 2021
Thermal/Vacuum Testing at LSU	April 16, 2021
FRR	May 3, 2021 EXTENDED to 5/10
FRR Defense	May 17, 2021
Launch	TBD
Science Presentation	TBD

Table 19: Gantt Chart for DemonSats-2 Payload Project



10.0 Master Budget

10.1 Expenditure Plan

Table 20: List of Expenditures for the DemonSats-2 Payload

Description	#	Units	Unit Cost	Extended Cost
Lithium Batteries 6 V	2	6 pack	$39.98	$79.96
PTC Thermistors PTGL07AS5R6K4B51B0	1	10 pack	$0.59	$5.91
Toggle Power Switch	1	12 pack	$7.48	$7.48
16 AWG Red Wiring	1	25’ roll	$5.46	$5.46
16 AWG Black Wiring	1	25’ roll	$5.46	$5.46
Foamular XPS 4' x 8' 1/2" sheet	1	each	$12.98	$12.98
Roll EconoKote Mylar film, white, 6 ft roll	1	6' roll	$14.99	$14.99
Roll MonoKote Mylar film, orange, 6 ft roll	1	6' roll	$15.99	$15.99
Roll MonoKote Mylar film, purple, 6 ft roll	1	6' roll	$15.99	$15.99
Glue gun sticks	1	12 pack	$6.97	$6.97
Bottle Gorilla Glue	1	8 oz bottle	$10.97	$10.97
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				$7.99
SUBTOTAL				$232.49
CONTINGENCY	10%			$23.25
TOTAL				**$255.74**

Since the electronic components of the MegaSat shield have been provided through the LaACES team at LSU, the expenditure plan consists of parts needed for the power supply, the enclosure box, and basic prototyping materials. Six battery packs will be made to make sure that fresh batteries are used for testing and for flight. The gloves are to be used to assist in the balloon launch. The total sum including a 10% contingency is $255.75, which well below the allotted budget of $500 for this project. All of the testing equipment is already available in the faculty advisor’s research lab.

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 21: Locations of Material Acquisition

Description	Obtained from
Lithium Batteries 6 V	Amazon
PTC Thermistors PTGL07AS5R6K4B51B0	Mouser Electronics
Toggle power switch	Amazon
16 AWG red wiring	Home Depot
16 AWG black wiring	Home Depot
Foamular XPS 4' x 8' 1/2" sheet	Home Depot
Roll EconoKote Mylar film, white, 6 ft roll	Amazon
Roll MonoKote Mylar film, orange, 6 ft roll	Amazon
Roll MonoKote Mylar film, purple, 6 ft roll	Amazon
Heat sealing iron sock for EconoKote	Amazon
Bottle Gorilla Glue	Home Depot
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 22: Risk Event Analysis [Ranked 1 (low) – 5 (high)]

Risk Event	Likelihood	Impact	Detection Difficulty	When
Failed Component	3	3	3	Fabrication / Testing
Battery Pack Freezes	3	5	4	In-Flight
Mechanical Damage (Pre-Flight)	2	4	2	All Stages Before Flight
Mechanical Damage (Landing)	2	4	2	End of Flight
MicroSD Card Failure	2	5	4	In-Flight
Loss of Program File	2	3	1	Programming
Unexpected Weather	3	4	5	In-Flight
Payload Falling in Water	1	4	5	End of Flight

Table 23: 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	Louis West
Battery Pack Freezes	Reduce	Test insulation strategy and cold effect on battery prior to flight	Detection	Lacee Savage
Mechanical Damage (Pre-Flight)	Reduce	Test enclosure stability prior to installing payload circuitry	Detection	Savon Gipson
Mechanical Damage (Landing)	Reduce	Test enclosure stability for impulse reaction, strengthen weak stress points observed in drop testing	Detection	Savon Gipson
MicroSD Card Failure	Reduce	Consider installing additional microSD card writer into system. Verify card is in working order during pre-flight	Detection	Carlee Lake
Loss of Program File	Reduce	Save program with understandable title and backup file into MS Teams	Leaving for the day	Charles Beam
Unexpected Weather	Transfer	Accept that we cannot control the weather.	Forecast check	LaACES Management
Payload Falling in Water	Reduce	Consider installing conformal coating to circuitry or sealing box	Ask LaACES Mngt.	Carlee Lake

12.0 Glossary

ABS	Absolute (absolute pressure – starting at perfect vacuum)
A/D	analog to digital
CDR	Critical Design Review
CDT	Central Daylight Time
CSBF	Columbia Scientific Ballooning Facility
EET	Electrical Engineering Technology
FRR	Flight Readiness Review
GMT	Greenwich Mean Time
GPRMS	Global Positioning Recommended Minimum Coordinates
GPGGA	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
MS	Microsoft
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