A Backup Inertial Positioning System

NASA Colorado Space Grant Consortium DemoSat Summer 2022

From left to right: Graham Avers, Elliot Matlack, Alex Eriksson

Graham Avers: grahamavers@gmail.com
Elliot Matlack: emat7@rams.colostate.edu
Alex Eriksson: eriksson@colostate.edu

Colorado State University

Team B.I.P.S.

8/12/22

Graduate Advisors: Jacob Gottfried, Anthony Puga

CSU NASA Space Grant Program Director: Azer Yalin

Abstract

When traveling medium to long range distances, Global Positioning Systems are a tool very commonly used to determine the path and position of a vehicle or module. On most locations on Earth, GPS satellites are readily available to be used for this purpose. However, on other planets, or in places where GPS signals can not be accessed, alternative methods of position tracking are required. The way that we have attempted to solve this issue is through the use of an Inertial Measurement Unit (I.M.U.), a set of altimeters, and a camera. In previous applications, using solely an IMU in order to track position proved to be relatively ineffective as the accumulated error of such a system proved to be too great. By implementing altimeters and a camera to check position estimates against, we were able to remove some of that error and have shorter lengths of time to perform solely dead reckoning calculations. Through the use of 3 different methods of determining position, the device was able to achieve a final position accuracy of .7218 kilometers offset to that of a GPS. The 3 different methods used were: an Inertial Measurement Unit (IMU), pictures taken on descent, and the final landing elevation compared to topographical data of the surrounding area. The IMU measures accelerations, rotations, and magnetic fields with an accelerometer, gyroscope and magnetometer. Using rotations and magnetic field measurements, a global axis system can be established for the accelerometer readings. An ENU (East = +X, North = +Y, Up = +Z) axis system was used for orienting the accelerometer measurements. These measurements were then double integrated with rectangular integration and offered a final position. The final elevation was recorded by a high accuracy onboard altimeter. This elevation was compared with points from elevation data pulled from the google elevation API. Elevation data in the region of the launch that was at the final elevation was then plotted on a high resolution satellite map for comparison with the descent photos. On descent, pictures were taken every 60 seconds. These descent photos were to be used to narrow down the final landing spot. Although these three measurements alone are inaccurate, together they offered a comprehensive and accurate final position.

Table of Contents:

Mission Overview

Introduction:

Global Position Systems (GPS) offer high precision and accurate navigation and positioning support on Earth. However, as the human race continues its exploration beyond, GPS tracking will become insubstantial on newer, underdeveloped / out of range planets and areas. More localized tracking methods could offer higher accuracy, and more reliability. This is where the idea for a backup inertial positioning system was born. With the use of a 9 axis Inertial Measurement Unit (IMU), acceleration, rotation, and magnetic fields can be measured. With the use of a rotation matrix and basic numerical integration, position can be calculated. In theory, this alleviates the need for any external assistance for navigation and position determination. However, as with everything, there are errors. The biggest problem with this method of position determination is accumulated error. If the IMU measures an acceleration at .001 m/s^2 when in reality it is .00095 m/s^2, and continues to have an error of this amount, error will accumulate, and will be accentuated through double integration. This is known as "drift." To counter this, multiple methods of determining position were implemented. On the descent of the package, pictures were taken of the surrounding area that were then compared to a high resolution satellite map to help with an estimate of the final position. Along with this, an onboard high accuracy altimeter recorded the final elevation of the package. This final elevation was then contrasted with topographical data, which offered final coordinate positions of the final elevation. With these 3 sets of data, the team was able to determine the final position of the payload with an accuracy of .7218 kilometers offset from a standard GPS.

Design

Structure:

The main structure of our device was made out of a 6" diameter, 6" tall carbon fiber cylinder. Acrylic supports were added to act as mounting surfaces for electronics as well as provide a path for the flight tube to pass through. The carbon fiber cylinder was recycled from a past project but was perfectly usable. The acrylic was provided by the CSU Powerhouse laser lab.

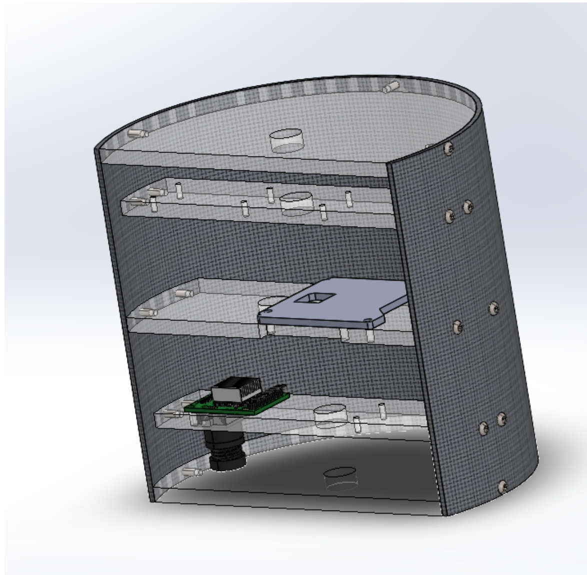


Figure 1: Cut View of the Structure

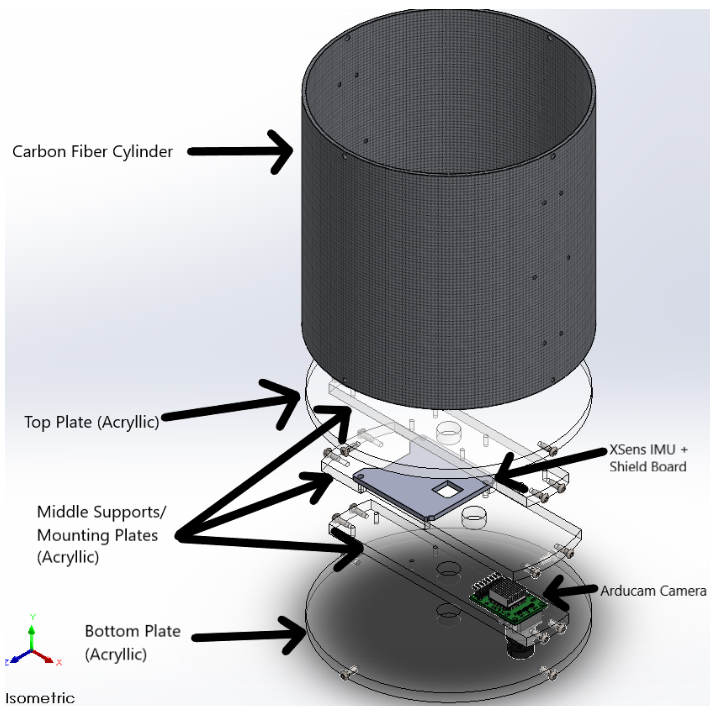


Figure 2: Exploded View of Structure

<u>Insulation and Padding:</u>

Double layered reflective insulation was added to the top and bottom circular acrylic pieces and also to the inside between the mounting holes for the acrylic supports. For added insulation and padding for the hard landing, thick black foam was added in sections around the outsides of the cylinder, making sure to leave room for the mounting hardware.

Hardware:

The majority of the hardware was ¼" 4-40 and 2-56 for some of the electronics mounting. All hardware was provided by CSU and was free to our group. To add additional damping, rubber bushings were added to all of the hardware that mounted the acrylic to the carbon fiber to prevent any cracking where the tapped holes were.

Electronics:

An extensive list of all of our electronic components can be found in Table 1. All electronics were bolted down to the acrylic supports, with the batteries and heaters zip tied to the side insulation and the middle acrylic support.

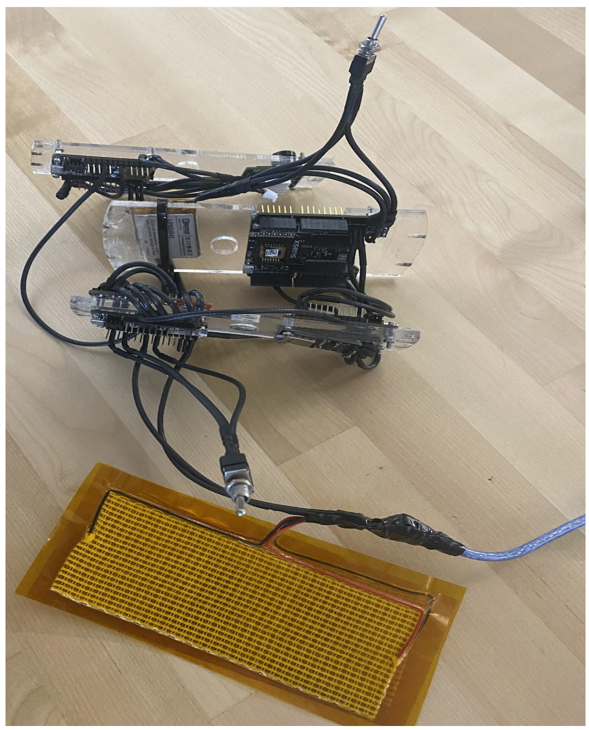


Figure 3: Assembled Electronics

Machining:

All of our acrylic pieces were laser cut by a colleague. We then drilled and tapped all of the mounting holes in the acrylic on the sides where the carbon fiber interfaces with the supports. In addition to machining the acrylic, holes had to be drilled in the carbon fiber which we did using a 3D printed drill guide. Through our research, we found that the best way to drill carbon fiber is with a brad point solid carbide bit, so we purchased the correct sized bit from Mcmaster.

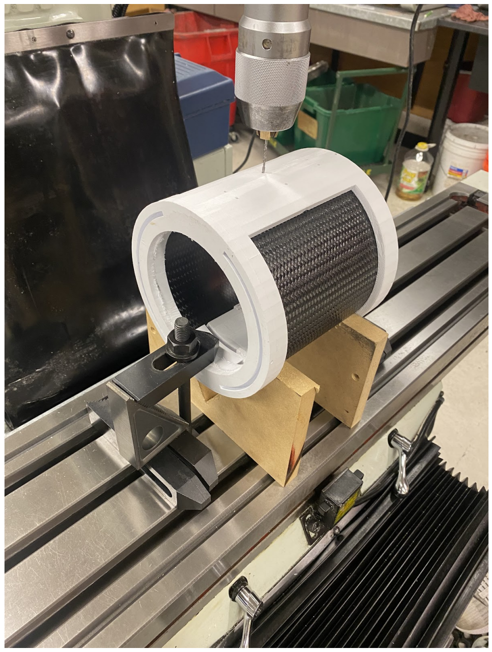


Figure 4: Machining the Carbon FIber

Figure 4 shows the machining setup we used to drill holes into the carbon fiber. The white plastic that is being clamped down on is our custom 3D printed drill guide to ensure that the holes we drilled were in the correct locations and had the correct spacing.

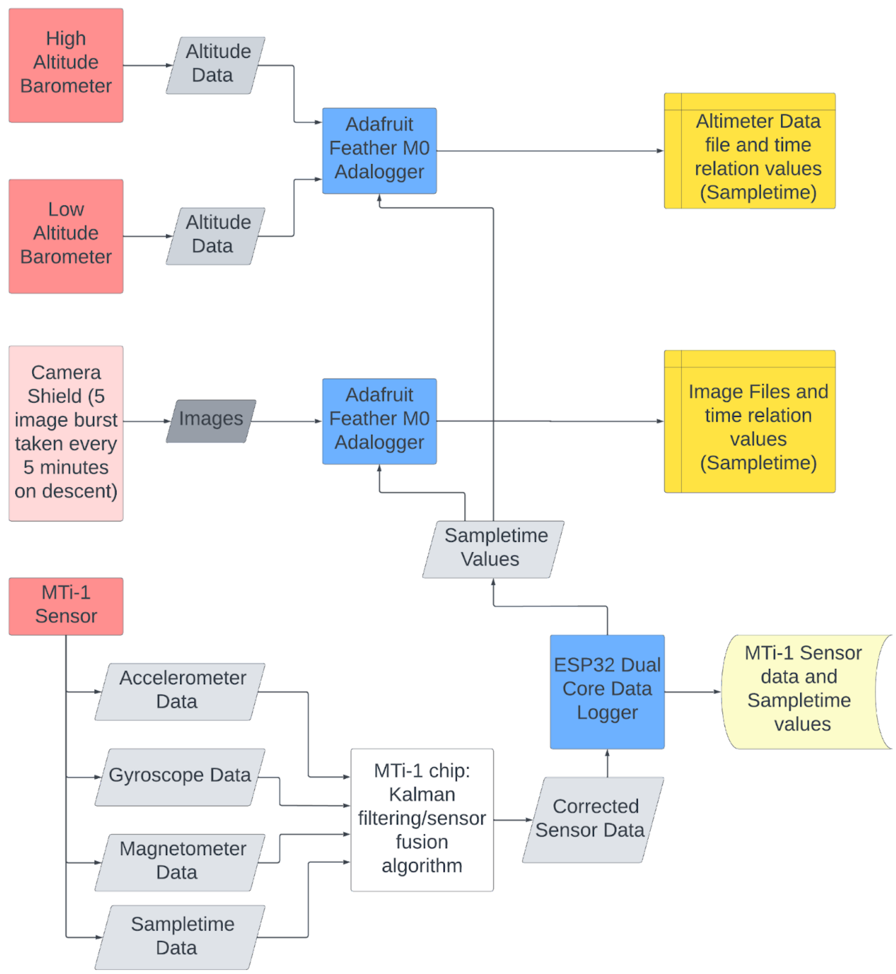
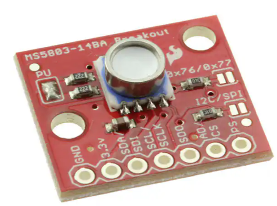
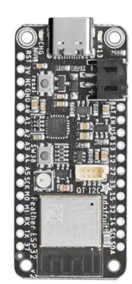
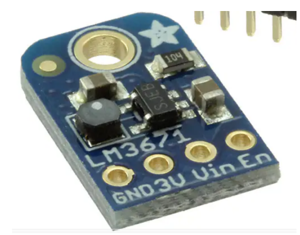
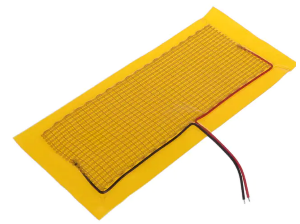
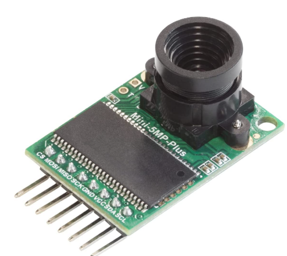


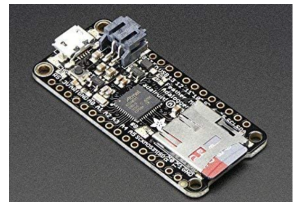
Figure 5 depicts a functional block diagram for the electronics of the payload.

Parts:

Table 1: List of all parts, cost, description, and picture

Name	Cost	Description	Picture
XSens MTI-3-01 DK	$449	Xsens Inertial Measurement Unit, Main sensor package for payload	
BME 688	$19.95	Altimeter, reads temperature, gas, humidity, and pressure. Not very accurate at high altitude, however, offers high accuracy at low altitude.	
Sen-12909	$64.50	High altitude barometer, will be able to determine altitudes upward of 100,000 feet.	
ESP32 Adafruit Feather	$19.95	Dualcore ESP32 chip mounted on a adafruit feather board. Allows for easy integration of code and mounting. As well as battery plug + built in charger.	

Voltage Regulator	$4.95	Steps down voltage output from batteries and (3.7-4.2v) to 3.3v for Xsens VDDA Pin.	
Heaters	$5.95	Basic Heaters for preserving battery efficiency at higher altitude and subzero temperatures.	
Micro SD Card	$7.99	Data storage	
2000 mAh Lipo battery	$15.19	Battery for powering dual core esp32 microcontroller	
820 mAh lipo battery	$13.09	Batteries responsible for powering adaloggers and all sensors	
Arducam Mini OV5642	$39.99	5 Megapixel Camera, responsible for taking pictures at peak and on descent.	

Adafruit Feather M0 Adalogger x2	$25.99	Single Core processing unit, also has built in mini sd reader/writer and battery charger	
#43 Brad Point Carbide Drill Bit	$17.92	Drill bit made specifically for drilling into carbon fiber.	

Challenges:

Processing Bottleneck:

One important test performed was for determining the maximum write speed the adalogger could handle. The Xsens IMU is able to output free acceleration and other measurements necessary for position calculations at 100 Hz. The high frequency data from the IMU proved to be too much due to the single core architecture of the Adalogger. Because of this, the maximum frequency that could be recorded onto an SD card was 20 Hz. However, with the implementation of a dual core ESP32, receiving the high frequency data from the IMU and writing it to the SD card in parallel was now achievable and allowed data writing speeds greater than 100 Hz.

Carbon Fiber Cylinder Machining:

For the main body of the payload, a 6” diameter carbon fiber tube was used. To mount the acrylic plates, holes had to be precisely drilled and had to line up with the holes exactly on the other side of the cylinder. Unfortunately, a 6” long, .089” diameter carbide drill bit that can spin at 4000 rpm while drilling through carbon fiber is not a product that currently exists. To solve this problem, the team created a 3d printed drill guide. The carbon fiber tube would slide into the drill guide, and the holes on the 3d printed plastic would line up perfectly on both sides. See figure 6 for a visual on the 3d printed drill guide.

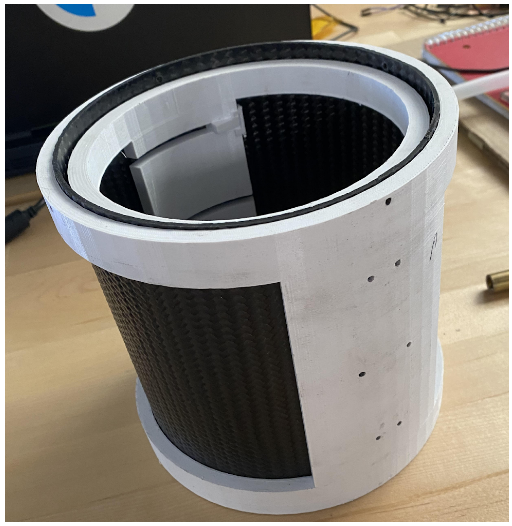


Figure 6 shows the carbon fiber cylinder inserted into the drill guide, notice the spaced guide holes in the plastic.

Magnetic Field and Current Disturbances:

For the Xsens Magnetometer to be accurate, it must be in a calibrated environment. Normally this calibrated environment is just on Earth. However, with the introduction of many wires closely packed together, all with power flowing through them, they introduced additional magnetic fields. These magnetic fields offered disturbance to the XSens and resulted in inaccurate measurement. To solve this, the wires were shortened in length, routed away from the sensors, and the wire itself was changed from plain copper to copper wire shielded with aluminum. This decreased the amount of magnetic field disturbance in the system. Another issue with the XSens was that it had two required power inputs, however, the power for both of those inputs came from one source. When one input pulled more current, the other input would receive less. This change in current created a magnetic disturbance which also made the data inaccurate. To solve this issue, a separate power source was made for each power input.

The software that came along with the Xsens MTi-3 also included a local magnetic field mapper. This was utilized to calibrate the onboard magnetometer to account for local disturbances (any electrical signals the shielded wire could not protect from) while the rest of the sensors are running and any ferro-magnetic materials inside of our payload. This was accomplished by running the mapping software while the rest of the sensors were running and spinning the payload through every orientation it would experience during flight. The picture shown below are the results of our calibration.

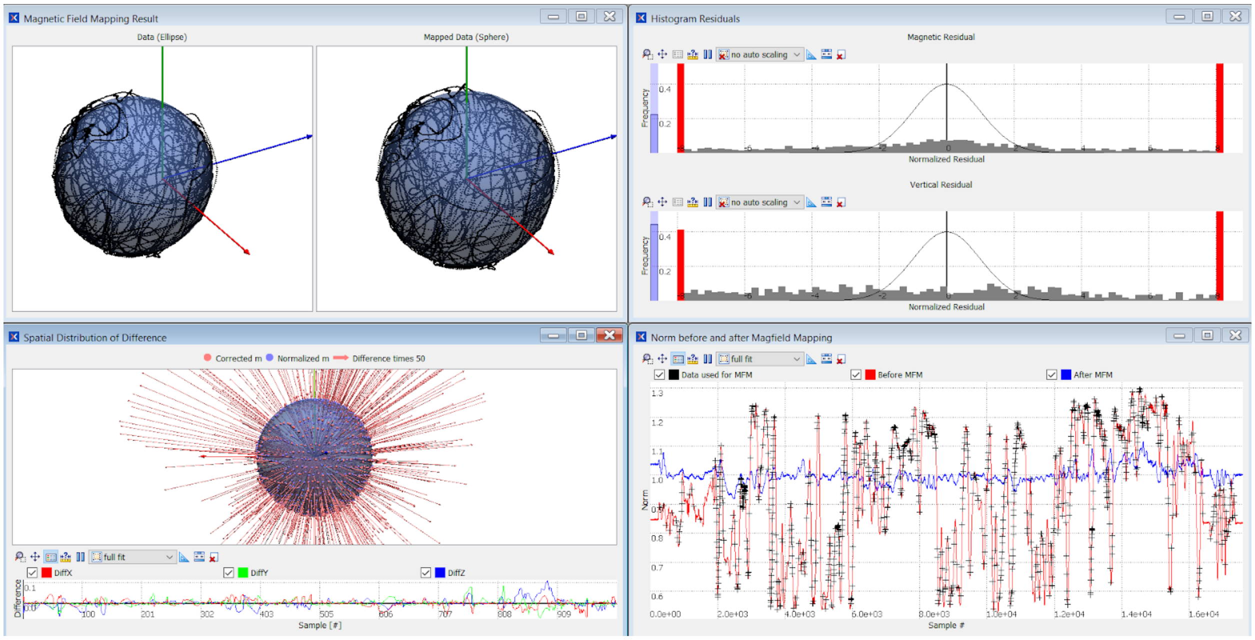


Figure 7: Magnetic Field Mapping Results

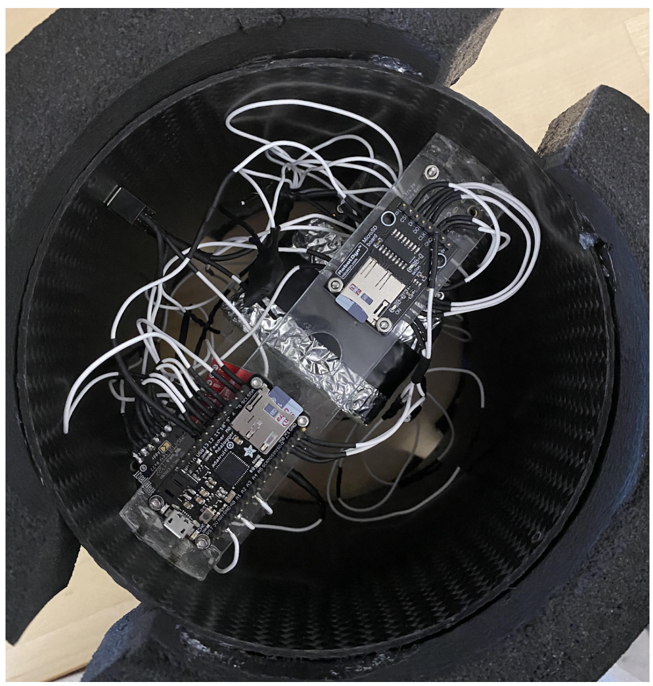


Figure 8 shows second iteration of completed wiring setup without shielded wiring

Figure 9 shows final iteration of wiring setup with Shielded Wiring

Euler Angle Limitations in 3D space (Gimbal Lock)

Rotation in 3D space is described and visualized with independent axis'. However, when converting these axis from the sensor package to a more global axis system, these axis are not always independent. This is a major flaw with Euler angles. When 2 axis align, it is impossible to determine which is which afterwards without the use of external assistance. This results in the loss of a degree of freedom. To solve this problem, quaternion angles were implemented instead of euler angles. Because quaternion angles have 4 components instead of 3, it is impossible to lose the 3rd degree of freedom. See figure 9 for a visual representation of the Euler issue.

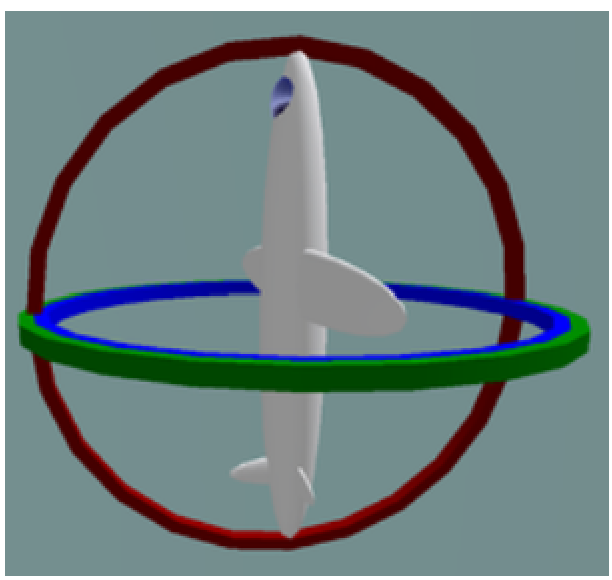


Figure 10 depicts a possible "Gimbal lock" that can occur when using Euler angles.

Post Processing:

Multiple methods of Post Processing were implemented after the launch. First and foremost being the calculation of the payload's final position from IMU measurements. In order to calculate final position, Axis must be determined. An ENU (East = +X, North = +Y, Up = +Z) axis system was used for this project.

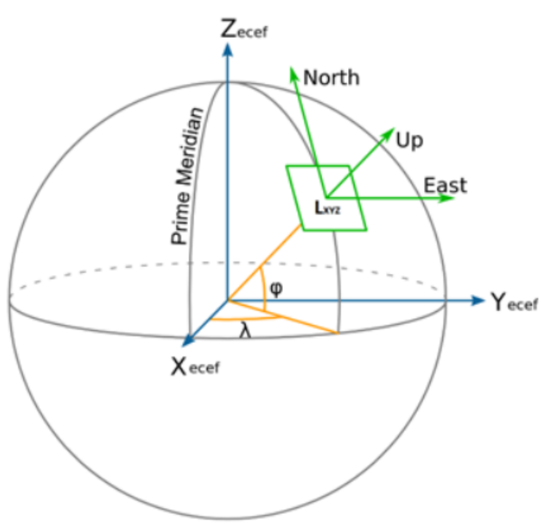


Figure 11: Representation of the ENU (East North Up) axis system.

As the accelerometer reads accelerations on its local X, Y, and Z axis, the magnetometer and gyroscope can be used to calculate the relation to the ENU axis system. Using a rotation matrix, the local accelerations on the accelerometer can be converted to the ENU axis system. See Figure 12 below.

$$\begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$

Figure 12 shows the rotation matrix R_{LS}

Elevation Final Position Estimation:

Another method that was used to determine the final position was by measuring the final elevation using an onboard altimeter. Then the final elevation was compared to a topographical map. Any value not within 3 meters of the final elevation was filtered out. These coordinates were then plotted on a high resolution satellite map which could be compared to the final position calculated by the IMU and could be compared to pictures taken on descent. To get this topographical data, the google Elevation API was implemented using a helpful matlab script developed by github user Jose, pinxau1000, Rosa. The matlab Mapping Toolbox was used for plotting the possible elevation points, and for accessing the high resolution satellite map. See Figure 13 for a hypothetical final landing elevation of 1600 m in the designated landing area of Genoa, CO.

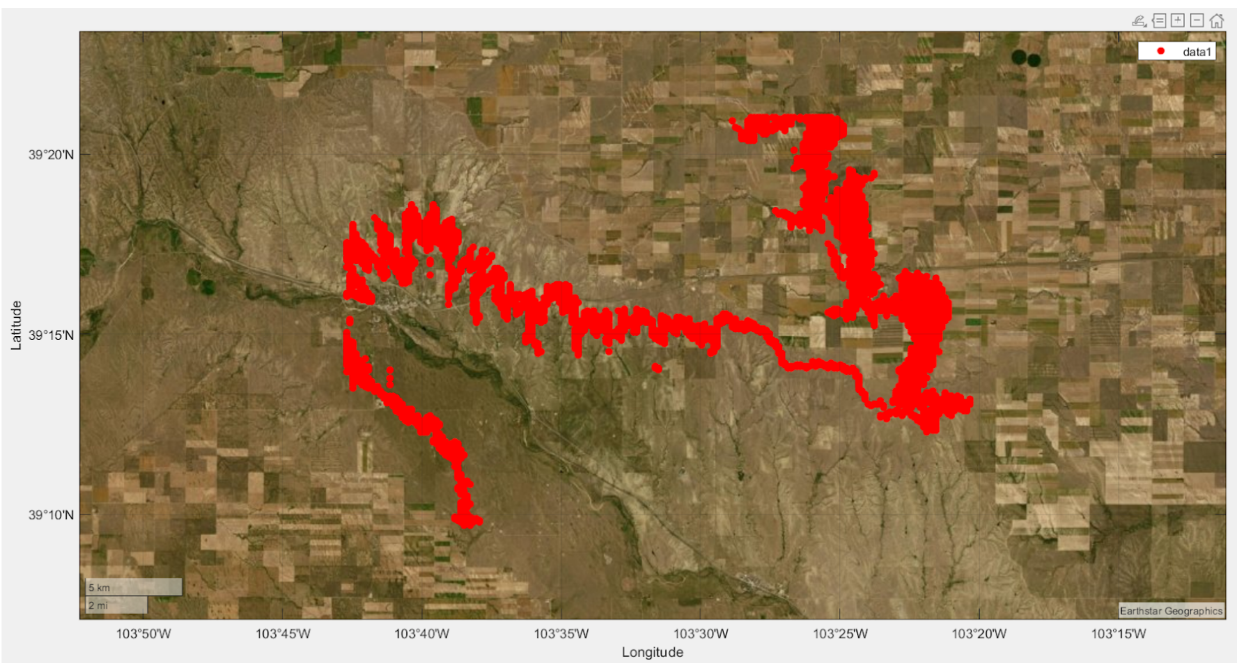


Figure 13, Simulated Matlab plot

The red dots show possible coordinate positions that relate to the hypothetical landing elevation of the payload. These can be used to help provide a more accurate final position.

Management

Budget

Table 2: Expenses and use of $1000 budget

Part Name	Quantity	Price
XSens MTI-3-01	1	$449.00
SEN-12909 High altitude altimeter	1	$64.50
BME688 barometer+temp sensor	1	$19.95
Dual Core adafruit	1	$19.95
Voltage Regulator	1	$4.95
Heater	2	$5.95
micro sd card	3	$7.99
2000mAh battery	1	$15.19
ArduCAM Mini OV5642	1	$39.99
adafruit data logger/microcontroller	2	$25.99
Battery 820 mAh	3	$13.09
Carbide drill bit	1	$17.92
Shielded Wiring	1	$10.00
12.5 lb Dry Ice	1	$10.50
Total	**16**	**$769.07**

Table 3: Total Weight of Payload

Part Description:	Weight:
Carbon Fiber Cylinder	209g
Plastic Top and bottom + middle supports	308g
Flight String Tube	10g
Batteries + electronics	118g
Insulation	150g
Misc Mounting + adhesives	~100g
Total	**895g**

Testing

During Launch, the payload endured temperatures as low as -80°C, experienced accelerations of 15 g's, survived near vacuum conditions, and high force impacts. To ensure the survivability of the payload, rigorous testing was required.

Cold Test:

The payload was placed inside of a cooler with dry ice. This simulated the extreme cold temperatures the package would experience. Temperatures reached -80°C during the test.

Whip Test:

The payload was attached to a rope and then spun around in circles, a rapid change in direction helped simulate the high acceleration changes the payload experienced.

Stair Test:

The payload was thrown down a set of stone stairs to simulate a worse case landing scenario wherein the payload's parachute would be caught by wind and would drag the payload across rocks and hard ground.

Code Resilience Testing:

Throughout the development of the code and payload, numerous tests were performed to determine the accuracy, efficiency, and reliability of the sensors.

Launch and Recovery

Below are pictures depicting launch and recovery of the payload. Both went smoothly and without any problems.

Figure 14 is a picture of the launch of the payload off of the balloon.

Figure 15 is a picture of the payload on its descent with the parachute deployed.

Figure 16 is a picture of the final landing position of the payload.

Results

Final Landing:

Observations

Upon landing, the module stayed together completely, suffering no structural damage due to the impact with the ground.

Payload Condition:

After landing, the payload was in very good condition as shown below.

Figure 17 depicts the payload's final landing position and condition.

Data:

The Maximum altitude for the payload was 88,000 feet. This is according to the launch provider Edge of Space Sciences (EOSS). The internal temperature of the payload did not go below 0 degrees Celsius. See Figure 18 for a graph of the temperature data.

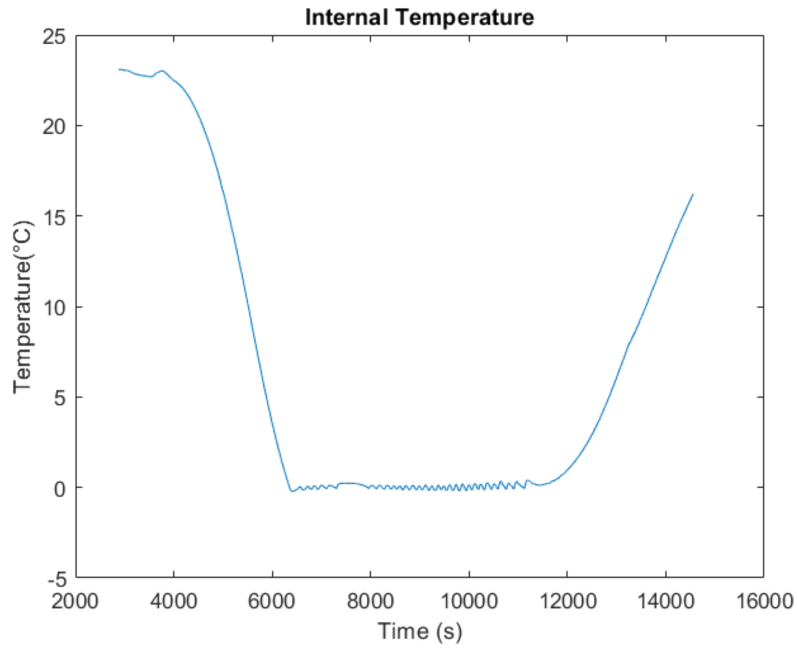


Figure 18 depicts the internal temperature of the payload for the duration of the flight

The altitude was a fairly easy bit of data. Below are two figures, each representing the data from the BME 680, and the Sparkfun Sen-12909. Notice how the BME 680 has a higher low altitude accuracy while the Sparkfun Sen-12909 has a higher high altitude accuracy.

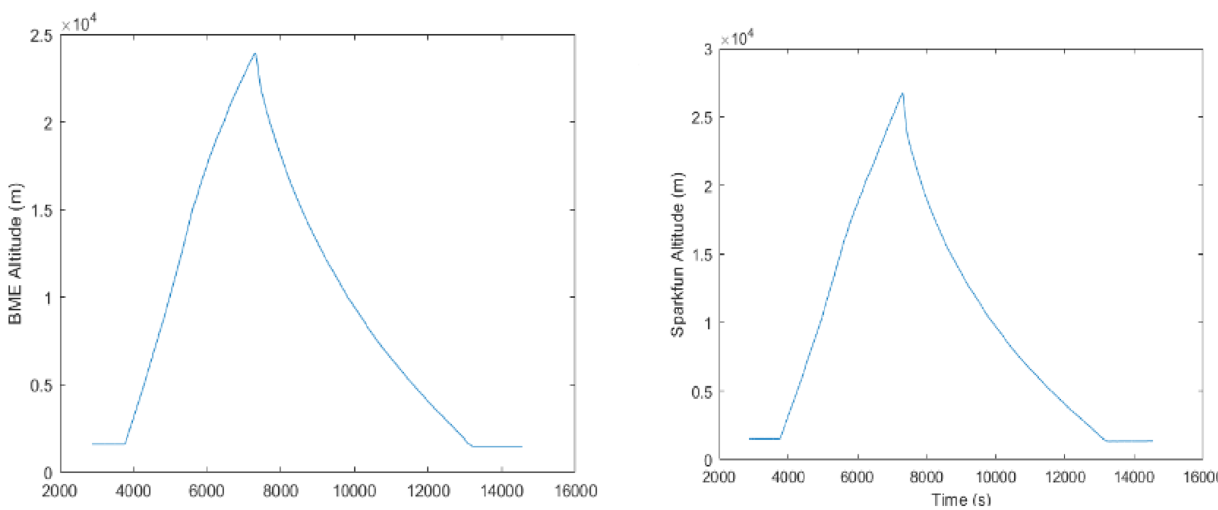


Figure 19 depicts both the BME altimeter's altitude measurements over the duration of the flight, as well as the Sparkfun Sen-12909's

Below in Figure 20 is a picture taken during the assent of the payload.

Figure 20 is a picture taken during ascent.

With a final elevation of 1542 meters recorded. Byy using the matlab script along with knowledge of the final landing area of the package. The coordinates of 39.0554, -103.274 were acquired. This led to an error of .7218 kilometers from the GPS. See Figure 21 for a picture of the matlab generated figure. For a more accurate final position, a higher resolution topographical data set could be used. This topographical data came from the google elevation database by using a google elevation API key.

Figure 21 depicts a high resolution satellite map with possible final positions of the payload based off of the final altitude.

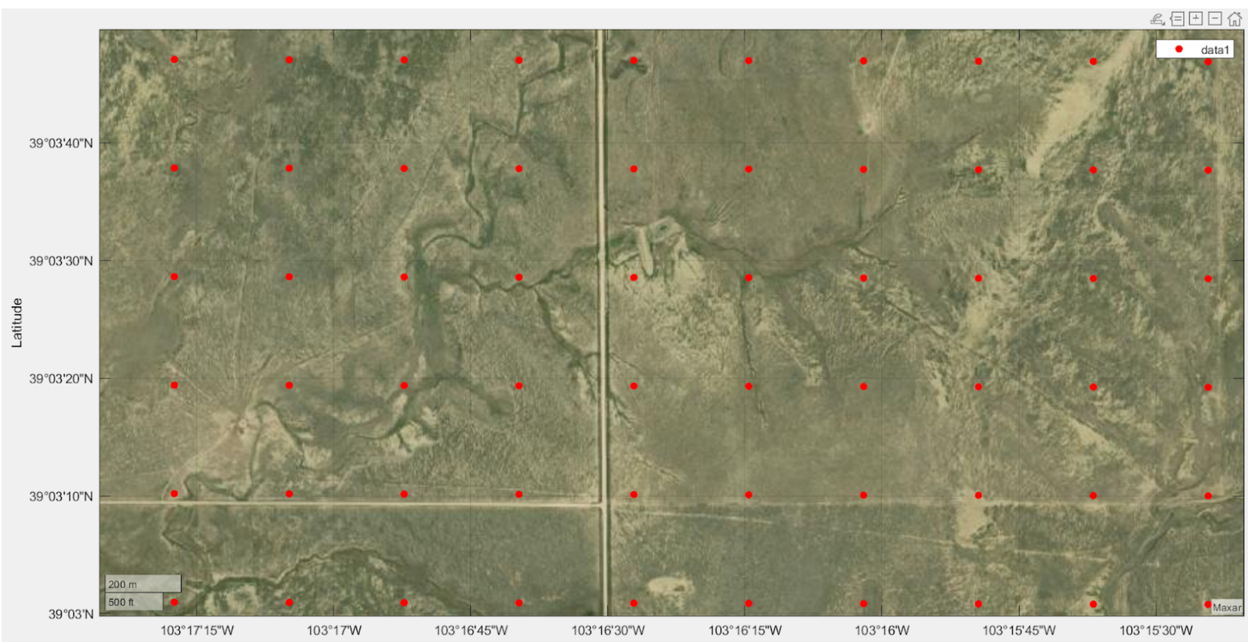
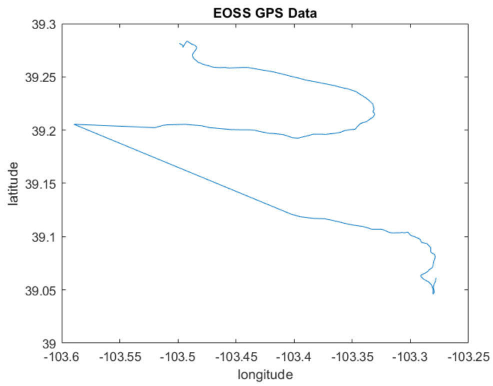


Figure 22 depicts the resolution of the data points available

Using this final position estimation, the group was able to filter the IMU data to get an accurate flight path. What is very interesting is that there was a fail in the GPS during descent, this resulted in a lack of data for this time period. However, using the IMU data, the flight path can be updated.



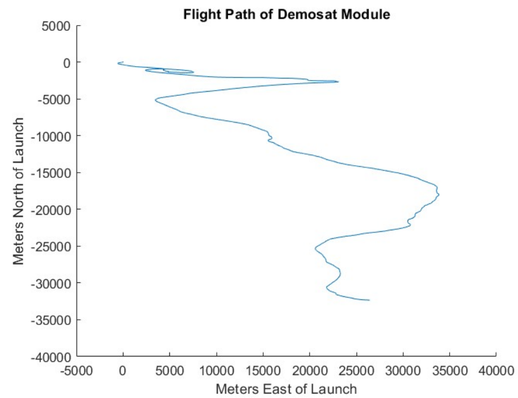


Figure 23 depicts the GPS data of the flight path (Left) and the filtered IMU data (Right). Notice the very straight line on the GPS data, this is due to a failure in the GPS.

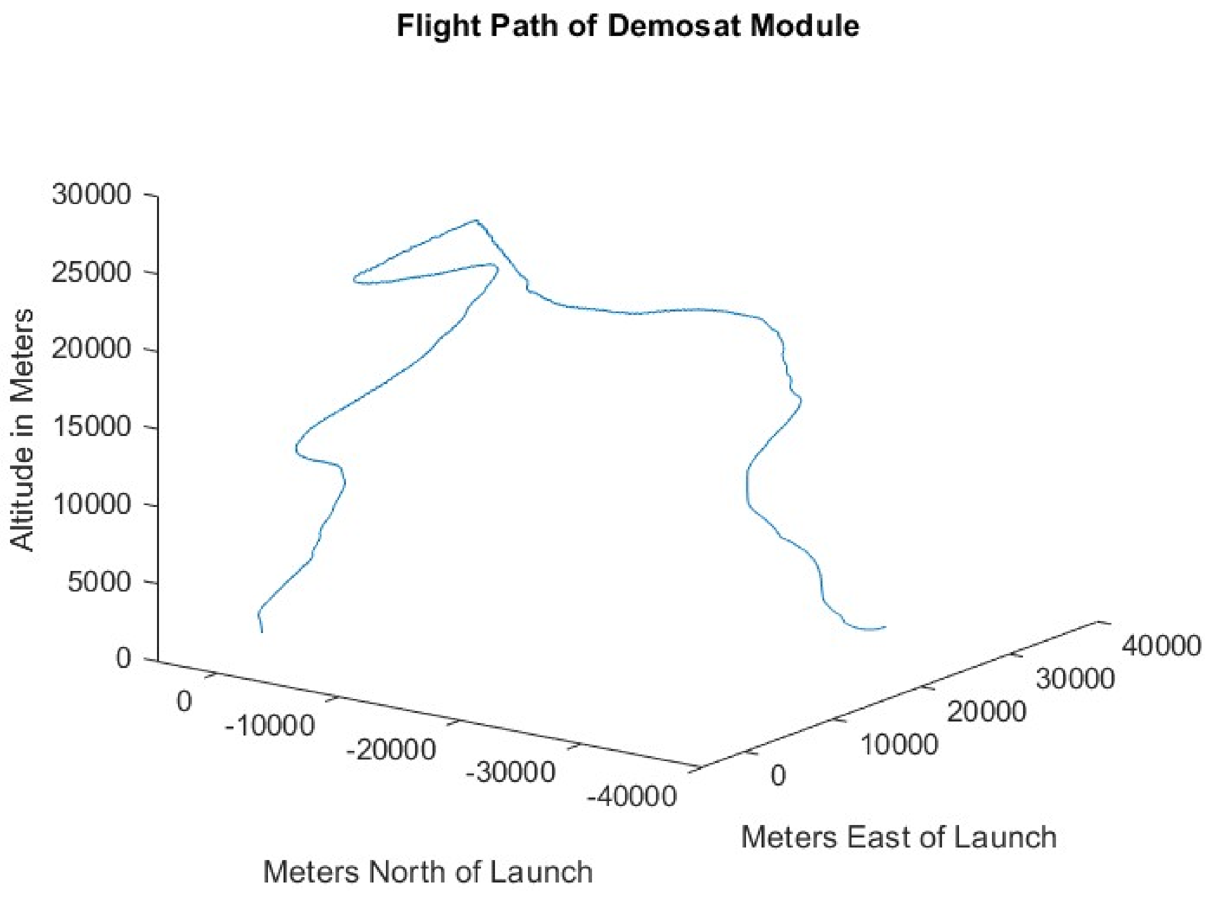


Figure 24 depicts the 3d flightpath of the payload according to the IMU and the altimeter data that was recorded.

Conclusion

Although this project was quite successful, more testing must be done to determine the efficacy of implementing this position determination method on a broader scale. It may be of value to look into more methods of determining position on a local level, to increase accuracy, and to increase the possibility of success during extraterrestrial applications.

Although the IMU was not as accurate as was initially expected, based on small scale tests, it performed well enough to consider continued applications on the large scale. Based on the offsets of the data for the entire flight, (the value of the mean x, y, and z accelerations) the group was able to perform a correction on the data acquired during the flight. This resulted in a flight path that is accurate enough to give a general idea of the direction and movement the payload experienced. This is visible when comparing it to the GPS data that was recorded.

Applications:

The development of this device was focused on being primarily an alternative backup positioning system which could be implemented during periods of time wherein there is a loss of communication, especially when traveling beyond the orbit of the Earth.

Future Projects:

One future project that came to mind was using an onboard Lidar Scanner to generate a point cloud of the surrounding landing area. Then comparing the slopes (change in elevation) to that of a global map to help offer a more accurate positioning. A very advanced version of this is used in Military applications such as the Tomahawk missile and other unmanned aircraft.

Github:

To view the code and libraries for this project on a more in depth level, visit the github page: https://github.com/GrahamAvers/NASA-Inertial-Positioning-System

Message to Next Year:

The biggest consideration when starting out on this project is having a grasp on a realistic timeline. Choosing a project is a difficult part of the first week or so of the internship, but is a crucial factor to whether you will succeed or not. Overall we were all very happy with the difficulty of the project that we chose and highly suggest spending a good portion of time and energy in this stage.

Acknowledgements:

Dr. Azer Yalin (Colorado State University) - Provided the Demosat team with his advice and support while overseeing the program.
Jacob Gottfried and Anthony Puga (Colorado State University) - For their oversight, help with sensor troubleshooting, and general engineering advice.
Bernadette Garcia (Colorado Space Grant Interim Director) - Gave us helpful feedback during our presentations and provided mentorship.
Jenna Hushka (Tech Support for XSens) - Helped us decide which sensor in the XSens lineup to choose for our application and assisted in troubleshooting to get our sensor recording.
Indraneel Bavkar (XSens Product Specialist) - Helped troubleshoot our XSens sensor when we were not getting North referenced data among other issues that arose.

References:

ArduCAM, "ArduCAM/arduino: This is arducam library for Arduino boards," *GitHub*. [Online]. Available: https://github.com/ArduCAM/Arduino. [Accessed:11-Aug-2022].

pinxau1000, "PINXAU1000/Matlab-Google-elevation-API: Matlab script to get a elevation profile along a path defined between two points with n resolution samples.," *GitHub*. [Online]. Available: https://github.com/pinxau1000/Matlab-Google-Elevation-API. [Accessed:11-Aug-2022].

"Don't get lost in Deep space: Understanding quaternions - technical articles," *All About Circuits*. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/dont-get-lost-in-deep-space-understanding-quaternions/. [Accessed:11-Aug-2022].

Papadimitriou, I. (2022, February 11). *Arduino I2C library (unfinished)*. Xsens knowledge base. https://base.xsens.com/s/question/0D509000016hfSECAY/arduino-i2c-library-unfinished?language=en_US [Accessed:11-July-2022].