Sensor Module Network for Monitoring Trace Gases in the International Space Station

Aaron Beck  
*Northern Illinois University*, z1886710@students.niu.edu

Drake Provost  
*Northern Illinois University*, Z1872355@students.niu.edu

Christopher English  
*Northern Illinois University*, Z1908199@students.niu.edu

Kamrin Gustave  
*Northern Illinois University*, Z189061@students.niu.edu

Follow this and additional works at: https://huskiecommons.lib.niu.edu/studentengagement-honorscapstones

Part of the Digital Communications and Networking Commons, Risk Analysis Commons, and the Space Habitation and Life Support Commons

**Recommended Citation**


This Student Project is brought to you for free and open access by the Undergraduate Research & Artistry at Huskie Commons. It has been accepted for inclusion in Honors Capstones by an authorized administrator of Huskie Commons. For more information, please contact jschumacher@niu.edu.
FINAL REPORT

Sensor Module Network for Monitoring Trace Gases in the International Space Station

TEAM 2

Aaron Beck, Chris English, Kamrin Gustave, Drake Provost

Faculty Mentor: Dr. Benedito Fonseca, ELE

Project for: Jet Propulsion Laboratory (National Laboratory)

Client Contacts:

Dr. Lance Christensen, JPL, lance.e.christensen@jpl.nasa.gov
Alexander Hart, JPL, alexander.j.hart@jpl.nasa.gov, (630) 767-9635
# TABLE OF CONTENTS

**ABSTRACT** .................................................................................................................. 4

1 INTRODUCTION ........................................................................................................... 5

1.1 Background ................................................................................................................. 5

1.2 Purpose of the Project ............................................................................................... 6

1.3 Previous Work Done by Others ............................................................................... 7

   1.3.1 Existing Products .............................................................................................. 8

   1.3.2 Patent Search Results ...................................................................................... 8

1.4 Brief Overview of the Report ................................................................................... 11

2 PROJECT DESIGN ....................................................................................................... 13

2.1 Optimal Design .......................................................................................................... 13

   2.1.1 Objective .......................................................................................................... 13

   2.1.2 Subunits ........................................................................................................... 14

   2.1.2.1 Graphical User Interface ............................................................................. 14

   2.1.2.2 Mote Firmware ........................................................................................... 23

   2.1.2.3 Sensor Mounting System ............................................................................ 29

   2.1.2.4 Automated Breathing Simulator ................................................................. 33

   2.1.2.5 Printed Circuit Board .................................................................................. 38

2.2 Prototype ................................................................................................................... 42

   2.2.1 Graphical User Interface ................................................................................. 42

   2.2.2 Mote Firmware ................................................................................................ 47

   2.2.3 Sensor Mounting System ................................................................................. 52

   2.2.4 Automated Breathing Simulator ....................................................................... 53

   2.2.5 Mote Electrical Components .......................................................................... 55

3 REALISTIC CONSTRAINTS ......................................................................................... 58

3.1 Engineering Standards ............................................................................................. 58

3.2 Economic Constraints ............................................................................................... 58

3.3 Environmental Constraints ....................................................................................... 58

3.4 Sustainability Constraints ......................................................................................... 59

3.5 Manufacturability Constraints .................................................................................. 60

3.6 Ethical Considerations and Constraints .................................................................... 60
3.7 Health and Safety Constraints ................................................................. 61
3.8 Societal Constraints ............................................................................. 62
3.9 Political Constraints ............................................................................ 62
4 SAFETY ISSUES ....................................................................................... 64
5 IMPACT OF ENGINEERING SOLUTIONS .................................................. 64
6 LIFE-LONG LEARNING ........................................................................... 65
7 BUDGET AND TIMELINE ........................................................................ 66
    7.1 Budget .............................................................................................. 66
    7.2 Timeline ........................................................................................... 68
8 TEAM MEMBERS CONTRIBUTIONS TO THE PROJECT ................................ 71
    8.1 Aaron Beck ....................................................................................... 71
    8.2 Chris English .................................................................................... 72
    8.3 Kamrin Gustave ................................................................................ 73
    8.4 Drake Provost .................................................................................. 74
9 CONCLUSION .......................................................................................... 75
10 REFERENCES ............................................................................................ 76
11 ACKNOWLEDGEMENTS ....................................................................... 78
12 APPENDIX ............................................................................................... 79
    12.1 Updated Specifications ...................................................................... 79
    12.2 Purchase Requisitions and Price Quotes ........................................... 80
ABSTRACT

The Jet Propulsion Laboratory (JPL) of the National Aeronautics and Space Administration (NASA) aims to develop a sensor network for the International Space Station (ISS) to ensure a comprehensive understanding of air quality within the station. The accumulation of carbon dioxide (CO$_2$) can lead to cognitive impairment, headaches, and potentially dangerous situations at high concentrations. Monitoring air content at the ISS is critical to maintaining a healthy environment for crew onboard. Exposure to harmful gases causes negative side effects that make crew sick, which may interfere with their responsibilities. CO$_2$ is a gas that should be monitored because of the side effects caused by prolonged exposure, such as nausea, vomiting, and headaches. Monitoring CO$_2$ content would help avoid negative side effects of CO$_2$ exposure and improve ISS crewmembers’ quality of life.

This project builds on previous senior design projects by enhancing the firmware and hardware of the sensor modules to support the collection and transmission of real-world data. The graphical user interface (GUI) functionality was improved to support new sensors and a network of more than one sensor module. The design of housing and mounts enables the replication of the ISS’s layout in NIU’s facilities, and the creation of an automated breathing simulator supports test automation.

No hard budget cap was specified by JPL, but the previous year's senior design project had a budget of $17,721.99. The expended budget for this year's project, excluding JPL's in-house CO$_2$ sensors, was $612.39, with most expenses allocated to the automated breathing simulator’s high-pressure valve.

Future work on the breathing simulator could further enhance the simulation of multiple human breathing patterns and enable the network to operate with a wider range of sensors.

If air content and flow data is collected at different locations in the ISS, crew can use this information to better understand trends in the air content which would help them maintain healthy air quality. Air CO$_2$ content can be monitored by collecting data with a wireless CO$_2$ sensor network and displaying this data in a digestible form at a dedicated base station computer. This is the fourth iteration of a proof-of-concept wireless CO$_2$ sensor network system proposed by NASA JPL, now with added support for airflow measurements. The current system features a protocol for framing data for wireless transmissions, fully programmed microcontrollers to get data from sensors to the base station, and a GUI to display CO$_2$ data and airflow data from multiple sensor modules.
1 INTRODUCTION

1.1 Background

Without air flow data, individual sensor measurements on the ISS lack spatial information. Only the pockets of air closest to the sensors are measured and thus the system has no way of tracking where the air is coming from or moving towards. To model the quality of air throughout all the ISS’s modules, localized sensor measurements must be complemented with anemometers to understand the flow of the measured air properties throughout the enclosed space. This is useful to further enhance the system and allow for identification of areas or crew on the spacecraft that may routinely produce larger quantities of CO₂.

The outcome of this project will be to incorporate these anemometers into the existing sensor array system. This will include but is not limited to interfacing the anemometers with the electrical system and data communication network, upgrading these systems as needed, updating software to drive the anemometers, adding functionality to store and send this sensor data, designing a housing and mount for the electronics, and updating the base station’s GUI to support multiple sensors and data visualization. Specifically, the GUI upgrades requested are added visualization support for multiple Media Access Controller (MAC) addresses, wind velocity (m/s), and wind direction (degrees). The Printed Circuit Board (PCB) used in the project should support collecting data from the anemometer sensor, storing the data in a logical manner, and transmitting requested data to a base station.

Currently, the team at JPL are lacking a comprehensive ability to determine the levels of gas build-up, notably CO₂, throughout the ISS. Previous CO₂ detectors on the ISS have failed to show significant build-up of the gases, despite crewmembers reporting headaches which were likely caused by heightened levels of CO₂. This new system should be able to help determine the reason behind the build-up of gases by tracking air quality and flow in a more comprehensive manner. This is needed as exposure to high levels of CO₂ can cause health problems including, but not limited to, impaired cognitive ability.

A rudimentary system exists that can generate, transmit, and plot pseudo data of CO₂, water vapor, temperature, and pressure. This system, while useful, lacks a spatial overview of air quality. This project aims to integrate air flow sensors (anemometers) into the existing sensor network to provide data on air movement, which can be used to infer spatial airflow information. This project will also work to collect and display all the disparate information of the sensors into one centralized location. JPL’s existing GUI will need to
be updated and improved to visualize the data recorded by the anemometers and be able to visualize data from more than one node at a time. The sensors will need to be able to transmit all data to a base station using commercial off-the-shelf (COTS) transceivers with predefined Zigbee protocols. The team also wants to quantify key network performance metrics as well as the drift in the various sensors. To enable automation and consistency in system testing, a robotic "breathing machine" will be designed, which emits CO₂ to emulate gas production by humans. The following figure illustrates how all components of the project interact to create a cohesive system.

![Figure 1: A block diagram of the entire system](image)

**1.2 Purpose of the Project**

Without air flow data, individual sensor measurements on the ISS lack spatial information. Only the pockets of air closest to the sensors are measured and thus the system has no way of tracking where the air is coming from or moving towards. To model the quality of air throughout all the ISS’s modules, localized
sensor measurements must be complemented with anemometers to understand the flow of the measured air properties throughout the enclosed space. This is useful to further enhance the system and allow for identification of areas or crew on the spacecraft that may routinely produce larger quantities of CO$_2$.

The outcome of this project will be to incorporate these anemometers into the existing sensor array system. This will include, but is not limited to, interfacing the anemometers with the electrical system and data communication network, upgrading these systems as needed, updating software to drive the anemometers, adding functionality to store and send this sensor data, and updating the base station’s GUI to support multiple sensors and data visualization. Specifically, the GUI upgrades requested are added visualization support for multiple MAC addresses, wind velocity (m/s), and wind direction (degrees). The PCB used in the project should support collecting data from the anemometer sensor, storing the data in a logical manner, and transmitting requested data to a base station. In addition, a device will be designed which mimics the emission of CO$_2$ in human breath, which will facilitate automated testing of the sensor system.

1.3 Previous Work Done by Others

Wireless sensor networks (WSN) have been around since the twentieth century. Though originally developed for military applications, the study and use of WSNs has become widespread [19]. Current applications include military, environmental, agricultural, and health, among others. Modern research in WSNs includes optimization and new applications.

NASA JPL-Caltech originally proposed this wireless sensor project in 2019 with Dr. Lance Christensen, a lab manager at JPL, offering his expertise in the field to assist in the project. Work has been done by Joseph Gehant, Susanna Eschbach, and Zahra Abbas, the 2020-2021 senior design team. They designed and built a ZigBee mesh network and Java GUI to display information about CO$_2$ sensor data and log network health metrics. Specifically, work was done to ensure minimal communication interference within an environment with preexisting Wi-Fi and Bluetooth networks.

The following year, Milan Patel, Pette J. Ramos, Alexander Hart, and Arman Eshtiaghi, the 2021-2022 senior design team, improved the robustness of the system by adding a backup system coordinator to remove a single point of failure from the system. This team also ported the GUI to the Qt framework, which allowed for
a more user-friendly interaction and additional navigation features to improve user-interaction with displayed data.

1.3.1 Existing Products

The company Paragon currently uses a Human Breathing Simulation (HBS) device in their test bed. This HBS accurately recreates the inhalation and exhalation cycle at user-defined breathing rates. Gas temperature, humidity, oxygen (O₂), and CO₂ content are controllable and recorded during testing. Paragon’s HBS uses at least one solenoid and a large gas tank to store the CO₂. Their system differs from this project in that it includes inhalation, humidity, and temperature controls. While Paragon’s HBS might be similar to this project’s breathing device, their HBS is not on the market so it would not be accessible for the tests needed by this project. [12]

Michigan Instruments has created a breath simulation machine. However, it is targeted at cardiopulmonary resuscitation (CPR) training. Because of this, Michigan Instruments’ device differs from the design objectives of this project. The main reason their breath simulation module would be insufficient for this project is that theirs is not designed for long continuous runtimes, which is what is needed, but rather is designed for short bursts for training purposes. [11]

1.3.2 Patent Search Results

The following patents were identified which relate to WSNs of air quality sensors:

- **Name:** Portable System For Measuring And Analyzing Indoor Air Quality And Method Thereof
  - **Document ID:** US 20220268748 A1
  - **Date Published:** 2022-08-25
  - **Inventor Name:** Manasa Hara Bhimaraju
  - **Abstract:** “The present invention discloses a portable system for measuring and analyzing indoor air quality and method thereof the system comprises an Internet Of Things (IOT) enabled sensor device (101) for collecting a user's geospatial data, wherein the sensor device (101) comprises a plurality of sensors for collecting indoor air quality data at pre-defined intervals of time. Further, a data analysis and storage module (102) are employed for receiving
the indoor air quality data collected by the sensor device (101), wherein the data analysis and
storage module (102) automatically analyses the data which is subsequently stored on a remote
server (102a) for future requirements. Additionally, a User Interface (UI) module (103) is
employed for displaying the analyzed data obtained from the remote server (102a) through a
web-based application installed on a user interface device.” [8]

• Name: System and Method For Controlling Indoor Air Quality
  ▪ Document ID: US 20220154954 A1
  ▪ Date Published: 2022-05-19
  ▪ Inventor Name: Richard R. Sinur, Kyle Anderson, Jeremy Yingst, Eric Theriault, Loic Ares, Jason
    Asmus, Seddik Rougab
  ▪ Abstract: “A system and method for obtaining environmental data—namely air quality
    information—from various devices contained within a structure is disclosed herein. The various
    devices contain sensors that can obtain environmental data, which is then analyzed by the
    system to determine if any level of a component within the data is outside of a predefined
    threshold range. If the system determines that the level of the component is outside of the
    predefined threshold range for that given component, the system will carry out certain steps in
    order to bring the level within the predetermined threshold range. These steps include
    selecting the appropriate appliance and the proper operating conditions to most efficiently
    bring the level back within the predetermined threshold range. Once the system has
determined that the level is back within the predetermined threshold range, the system will
instruct the selected appliance to turn OFF.” [9]

• Name: Integrated Air Quality Control System
  ▪ Document ID: US 10982873 B1
  ▪ Date Published: 2021-04-20
  ▪ Inventor Name: Colby Kevin Clark
  ▪ Abstract: “Methods, systems, and apparatus, including computer programs encoded on a
computer storage medium, for monitoring air quality with multiple sensing devices. The methods, systems, and apparatus include actions of: obtaining first sensor data from a sensor that is not an air quality sensor, determining that the first sensor data satisfy a first air quality criterion, in response to determining that the first sensor data satisfy the first air quality criterion, obtaining second sensor data from an air quality sensor, determining that the second sensor data satisfy a second air quality criterion, and based on a determination that the second sensor data satisfy the second air quality criterion, activating an air quality device.” [10]

• The following patents were identified which relate to automated breathing simulator devices:
  
  • Name: Metabolic Simulator Having A Catalytic Engine
    
    ▪ Document ID: US 20150076409 A1
    ▪ Date Published: 2015-03-19
    ▪ Inventor Name: Stefan Frembgen
    ▪ Abstract: “A respiratory metabolic simulator is disclosed. The respiratory metabolic simulator includes a catalytic carbon dioxide generator having a first inlet adapted for receiving a fuel and a second inlet adapted for receiving a gas; a breathing simulator; and a controller; wherein an exhaust of the catalytic carbon dioxide generator combines with an exhaust of the breathing simulator; and wherein the controller is configured to vary at least one of the fuel and the gas provided to the catalytic carbon dioxide generator such that the combined exhausts emulate human exhalation.” [5]

  • Name: Metabolic Simulator
    
    ▪ Document ID: US 20120060933 A1
    ▪ Date Published: 2012-03-15
    ▪ Inventor Name: Stefan Frembgen
    ▪ Abstract: “A respiratory metabolic simulator includes a cell that produces CO.sub.2 and depletes O.sub.2, e.g., a direct methanol fuel cell having an external electrical circuit. An electric load is applied to the external electrical circuit of the direct methanol fuel cell to vary
the electrical load applied to the external electrical circuit of the direct methanol fuel cell to produce carbon dioxide. The carbon dioxide generated by the direct methanol fuel cell is supplied to respiration gases produced by the respiratory metabolic simulator. The direct methanol fuel cell is also used to remove oxygen from the respiration gases prior to mixing the respiration gases and the carbon dioxide.” [6]

- Name: Breathing Simulator For Evaluation Test Of Respirator
  - Document ID: US 20070259322 A1
  - Date Published: 2007-11-08
  - Inventor Name: Hisashi Yuasa, Takeshi Honda, Kohsuke Nozaki, Kazushi Kimura
  - Abstract: “A breathing simulator for use in an evaluation test of respirators is provided, the breathing simulator being able to simulate larger respirations, undergoing a change neither in cycle nor in the amount of ventilation even in a test of respirators having a high pressure drop, able to easily generate not only such regular waveforms as a sine wave, a rectangular wave and a triangular wave but also respiration waveforms of workers and arbitrarily created waveforms as air waveforms, being reduced in size and power consumption, and easy to maintenance. The breathing simulator comprises plural air cylinders (1) for generating an air waveform, a single electric cylinder (2) for actuating the air cylinders, a servo controller (3) for controlling the operation of the electric cylinder, an input/output unit (4) for recording an analog input from the exterior and reproducing the recorded data at an arbitrary magnification, a PC (5) for creating an arbitrary waveform, and a waveform generator 6 for outputting the created waveform.” [7]

1.4 Brief Overview of the Report

This report explains in depth the work planned to be done by the current senior design team to build from previous teams’ work. Section 2 focuses on the project design, outlining the alternative designs and how this team selected the optimal design from those. Section 3 then discusses the standards and constraints imposed on the project in order to operate safely, effectively, and ethically. Following this, section 4 details the safety concerns associated with this project, and how it is planned to mitigate these risks. Section 5 describes the
impact of the previous engineering solutions in a global context. The proceeding section 6 addresses the educational value of this senior design project. After this, section 7 displays the project’s planned timeline and budget. Section 8 touches on the contributions of each group member, and section 9 summarizes the paper. The paper concludes with sections 10, 11, and 12 which comprise the references, acknowledgements, and appendix respectively.
2 PROJECT DESIGN

2.1 Optimal Design

Section 2.1 outlines the optimal design as determined by the senior design team in the fall 2022 semester. Section 2.2 will discuss the progress this team achieved towards this optimal design during the spring 2023 semester, as well as areas for further improvements.

2.1.1 Objective

This project aims to expand on previous work done by NIU senior design projects for JPL. This multi-year endeavor seeks to develop a sensor network which is capable of measuring CO₂ and airflow data on the ISS to gain a better understanding of air quality and how it affects crew. Previous projects designed and produced a network of Zigbee transceiver modules to transmit CO₂ data, and a GUI to display this data.

The previous sensor modules had firmware which could only support JPL’s custom portable tunable laser spectrometer (PTLS), which was used to measure CO₂. This project will refactor and version out much of this old firmware in order to be more adaptable to a variety of sensors. Instead of only supporting the PTLS, the sensor modules will be able to externally interface with both the PTLS and TriSonica Mini anemometer sensors, with planned flexibility to support other sensors of the same data protocol. Adding this support will require updating the electronics of the sensor modules via the design and manufacturing of a PCB.

Data captured by the sensor modules is transmitted to a base station, where it is logged and visualized in a GUI. Currently, this GUI only supports displaying data for one sensor at a time; it is planned to revise this to support displaying data from multiple sensors simultaneously. The GUI also lacks support for visualizing airflow data from the new anemometers, which will need to be integrated. Changes to the base station software and sensor modules will be designed with network performance analysis capabilities in mind.

These sensor modules will need housing to contain the electronics, and a mounting system to enable better replication of sensor placement on the ISS within the facilities of NIU. These housing and mounting assemblies will be drafted in computer-aided design (CAD) software, and then 3-D printed out of a material such as polylactic acid (PLA) filament.
Previously, these sensor modules were tested by having a human participant sit in a room with the sensor and breathe for potentially hours at a time. To automate testing without needing human participation, this project plans to create an automated breathing simulator (ABS) device. This will emit CO$_2$-enriched air in a quantity and cadence which mimics human breath. This will also potentially enable more reproducible system test results, as humans have varying breathing speed and CO$_2$ content depending on their health and level of activity. Initially this device will only emit gases, but this can be improved to better replicate human breath and its influence on spatial airflow by adding factors such as humidity, temperature, inhalation flow, and O$_2$ consumption.

2.1.2 Subunits

2.1.2.1 Graphical User Interface

In previous years, senior design teams working on this project developed a GUI in C++ using the Qt framework. This GUI was split into 3 tabs: an ISS map that displayed CO$_2$ levels, CO$_2$ concentration graphs as measured by the project’s sensors, and network health metrics graphs. This GUI, while sufficient at the time, does not fully meet the needs of the project as it now stands. Originally, only CO$_2$ sensors were part of the project, and it was planned to have one sensor per ISS module. As a result, the GUI displayed ISS data on a 2-D map without spatial information.
This year, the senior design team has planned to not only incorporate CO₂ sensors, but also anemometers to measure airflow and facilitate a better understanding of CO₂ distribution. This introduces new requirements that the existing GUI cannot meet. First, the GUI does not have a way to display anemometer data. The anemometer collects wind component data in three dimensions, meaning the current top-down view of the ISS is insufficient as it cannot visualize one of the airflow dimensions. In order to visualize the 3-D data of the anemometer, it would be desirable to move to a 3-D ISS Map. This introduces the design challenge of how to visualize a 3-D map in a 2-D GUI. A top-down approach like in the original ISS map is called an orthographic projection, which prevents one axis from being seen. A common solution to this is to use either isometric projections or perspective projections. These display all three axes at once, allowing for some representation of 3-D axes without the need to be dynamically interactable. Isometric projections have all three axes equally visible, whereas perspective projections have all three visible but with some more front facing than others. This explanation of isometric projection is given by [1]:

![GUI made by the previous team featuring a 2D "overhead" map of the ISS](image)

Figure 2: the GUI made by the previous team featuring a 2D "overhead" map of the ISS
In this projection perspective, it is possible to represent airflow in a 3-D space using a display of arrows which map to the vector components of the airflow, like so:

Each arrow corresponds to a different vector component, which allows anemometer data to now be visualized via the length and direction of each arrow. The second need unmet by the existing GUI is the ability to correlate sensor measurements with their location inside an ISS module, and to allow for more than one sensor per ISS module. This can be addressed by defining a specific location for each sensor, and then
visualizing sensor data from that location. Together, both this need and the need for anemometer 3-D data visualization can be addressed with an isometric projection ISS map, like so (map image from [2]):

![Isometric map of the ISS with mock sensor readings overlaid](image)

*Figure 5: Isometric map of the ISS with mock sensor readings overlaid*

Seen within the map are several circles representing sensor modules. The yellow circles represent anemometers, and the salmon-colored circles represent CO₂ sensors. All sensors visualize their data in the location it was measured, which allows for better contextualization of data and the ability to add multiple sensors to each ISS module. Anemometers update their wind vector arrows to reflect the measured airflow, and CO₂ sensors display their measured concentrations in bubbles above their location which change color in a way similar to a heatmap when CO₂ concentrations get too high (not shown). This new style of ISS map addresses the shortcomings of the original 2-D map. The first step of GUI development for this senior design project planned to implement this ISS map into the existing GUI, replacing the 2-D ISS map.

Originally, sensor data could be seen graphed against time in a separate “Concentrations” tab.
This was limited in that it only supported CO$_2$ concentration data and was hard-coded to display exactly six sensors’ information. This meant it was not expandable to include additional sensors or new sensor types, like the anemometers. For this reason, one alternative design proposed a revised version of the sensor data tab (renamed from “Concentrations”) that could address these shortcomings:
This version allowed for dynamic expansion of the types and amounts of sensors within the network. After discussion, the senior design team felt that this version was cluttered and unappealing. A new way of displaying information was proposed instead. Rather than keeping sensor and network data in separate tabs from the ISS map, this version has all information combined into one screen. This is accomplished using sidebars.
As can be seen in the figure above, the information previously shown in the “Network Health Metrics” tab has been relocated into a sidebar on the left. The ISS map is shown in the middle, with CO₂ sensors and anemometers as previously described. Note that the anemometer (in yellow) of the JPM module is outlined in cyan, indicating it has been selected by the user. This brings up additional information in the right sidebar for that sensor. The name and location of the sensor is shown at the top, an enlarged display of the 3-D axis arrows is shown below, and graphs of data against time are shown below that. Note that it is common in meteorological settings to refer to airflow vector components as U, V, and W wind components corresponding to X, Y, and Z components respectively [3]. By only displaying graphs for one sensor at a time, the GUI allows all sensor data graphs to be accessible while avoiding visual clutter. This design was favored by the senior design team, their faculty mentor, and the clients, and will therefore be pursued.

In sharing this design with JPL, they noted that in previous years, senior design teams separated the data into different tabs because they thought some end users might not care about certain data points. For example, a user may want to see sensor data, but not want to see network health metrics data. In response to this, the possibility of these sidebar tabs being minimizable was suggested, which was satisfactory to the client. Given
this, the GUI will be designed such that these sidebars are not fixed in place, but rather can be removed and re-added as needed.

Static ISS maps are very capable tools and computationally cheap (they are nothing more than an image with perspective tricks), but they do have limitations. A primary challenge of static isometric maps is that depth along the line of sight will always be obscured. This means that while all 3 cartesian dimensions can be visualized, there will always be some ambiguity as to where a sensor is located. Is it on the wall closest to the observer, or the wall furthest from the observer? To illustrate this point, consider this figure:

![Figure 9: illustration of ambiguity of static isometric images](image)

On the left half of the image, a cube is present with a yellow circle that represents some sensor. When this cube is rotated 90 degrees clockwise (the red and blue help visualize this and serve no other purpose), the issue becomes apparent. Along the original line of sight, was the sensor at the center of the close face, or was it at the middle of the far edge? Worse, any position along the line of sight is possible, not just on exterior walls. Static maps do not account for this depth along the observer’s line of sight, leaving it ambiguous. This poses a challenge for position-sensitive data, like air currents.

Initially, a dynamic 3-D ISS map as a solution was proposed as an alternative design. This would be an interactable 3-D model of the ISS that can be manipulated at will. This is reminiscent of 3-D models in CAD software programs. In this scheme, the line of sight of the observer is not fixed, meaning that the ambiguity in
the location of a sensor can be remedied by simply rotating the ISS model. However, the drawbacks of increased complexity, computation cost, and development time would likely outweigh the benefits of a dynamic ISS map, and so this option was decided to not be pursued.

During discussion of this challenge with JPL, two static ISS maps were proposed: an isometric projection view and a top view. Because these have different lines of sight, they can resolve each other’s ambiguity, allowing for precise understanding of the sensor location in all dimensions.

![Figure 10: isometric drawing complemented with a top-down view](image)

As can be seen in the figure, a combination of the isometric projection (left) and a top view can make clear whether the sensor is in the center of the close wall (middle), or at the middle of the far wall’s lower edge (right). This principle can make clear the location of sensors at any position along the line of sight as well, not just at exterior walls.

This strategy of having two static ISS maps is a valuable one, though the team has determined that it is a lower priority than other previously mentioned upgrades. For this reason, the GUI will initially have only the isometric projection view. If extra development time remains after implementing higher priority features, the team will consider implementing a top view ISS map which can be switched between by the user. In the meantime, the senior design team will investigate solutions which offer some level of location disambiguation.
while having a lower development time, such as adding a hatching pattern for sensors on far walls and no pattern for sensors on close walls.

### 2.1.2.2 Mote Firmware

The firmware for this project has undergone numerous changes and revisions over the years since the project’s inception. Currently, the existing firmware provided by JPL will be used as a baseline to develop features and functionality of the system. The existing firmware’s main loop creates mock sensor data, packages it into a ZigBee frame, and transmits that ZigBee frame.

![Diagram 1: Firmware logic block diagram](image)

As seen above, the system prior to any additions continues to send the last pending ZigBee frame that has not been confirmed to be delivered. In this way, the system ensures that all the collected data will reach the base station GUI.

The current system only supports mock CO\textsubscript{2} data, and no network health metrics. The optimal design chosen for this project remedies this situation by drastically expanding the capabilities of the system to integrate CO\textsubscript{2} sensors, anemometers, network health data, and configuration support. This design is primarily based off the first of the firmware alternative designs, and it focuses primarily on adding new functionality to the system. Ultimately this was the deciding factor in choosing this firmware design because the client expressed a desire
for functionality over refactoring of the existing codebase. This design choice will allow the system to retrieve and transmit sensor data from both the TriSonica Mini and the Telaire-T6615, support configuration of these sensors dynamically, and support configuration of XBee ZigBee network parameters and retrieval of network health metrics. Ultimately the team decided that additional features were much more worthwhile than refactoring the existing codebase to use a queue. The time spent refactoring can instead be spent developing new features which will be the highlight of this optimal design.

The firmware of this design can be broken down into two main categories: network configuration, and sensor communication. Each mote will have a radio-frequency (RF) transmitter and a sensor module. The microcontroller in the system will need software to read data from a sensor module, format that data into a ZigBee API frame, and send the frame to the RF transmitter. It will also need to be able to process custom frames developed for this project so that each sensor can be configured remotely from the base station. The two sensors that will be integrated into this system are the TriSonica Mini Wind & Weather (TMW) sensor, and the Telaire-T6615 Non-dispersive infrared (NDIR) CO\textsubscript{2} sensor. Both sensors output data in their own unique way, and the documentation provided for each outline custom protocols for configuring various data collection parameters in these sensors. The firmware on the SAM S70 microcontroller will support decoding of custom ZigBee frames that include this sensor parameter data, that will then be streamed directly to the sensors through their respective connections to the microcontroller.

<table>
<thead>
<tr>
<th>Sensor Type Struct</th>
<th>Trisonica Mini</th>
<th>Telaire-T6615</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 = Trisonica Mini</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x02 = Telaire-T6615</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xFF = Super_Sensor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Wind Speed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Wind Direction (s, n, w)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Humidity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Temperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Carbon Dioxide PPM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Example sensor firmware structure

In order to keep track of what commands to send and which port to send them to, a table of sensor types will be stored on the SAM S70 and on the base station. Each mote will also contain a section of non-volatile memory reserved for the sensor type that is currently connected to it. When the system is initialized, each
mote will check that spot of reserved memory to determine if it “knows” what sensor was last recorded to be connected. If the mote reads a value that does not exist in the table of sensor types, it will send a custom payload within a ZigBee Transmit Request frame to the base station. This custom payload will notify the GUI to prompt the user to select from a drop-down menu or various interface whichever sensor is currently hooked up to the system. It will be of utmost importance that this selection is made correctly, otherwise the system will have no way of knowing which sensor it is reading from and/or various configuration commands will not work. The reason that this input needs to take place is because the SAM S70 firmware has no other way of identifying the type of sensor. If the sensor type is attempted to be deduced from the input signal, the firmware first needs to know which peripheral to read from. This information will be linked with the sensor type so that the firmware can successfully read and write to the right peripheral on the SAM chip. In the case that the firmware guesses correctly or attempts to read from each peripheral attached to the board, each sensor will output its data at a different rate, size, and format. Without knowing which sensor is plugged into the board, attempting to identify the sensor by parsing input data and comparing it to sample sensor data is a monumental task. Parsing the input data also does not handle the situation of duplicate sensor data formats for different sensors. The solution to this problem is to tell the SAM S70 what sensor is connected to it through the user interface. This selection should only need to be completed once, because the SAM S70 will store sensor type in non-volatile memory. In the case that a different sensor is swapped onto a mote, the user would simply change the sensor type in the GUI for a particular mote and a ZigBee frame would be sent to the mote to update its sensor identity.
When the type of sensor has been established or the system is being powered on, there will be custom status information sent to the GUI from each mote if the sensor connected to it supports metadata queries. This information, such as device serial number or sensor status will be initially sent to the base station. This data will be provided so that the user will be able to utilize as much data as possible about each sensor in the network. It will most likely be displayed on the screen when a specific mote is selected in the GUI. Within this sensor information pane will be other configuration controls that are specific to each sensor such as a calibration button or a temperature reading of the sensor. Tools like this are useful for sensor maintenance and care over a long period of system runtime. Next to the sensor metadata current network data for the mote will be displayed. This will include information about current packet route, last received signal strength, and temperature of XBee module.

Diagram 2: Sensor transmission sequence block diagram
All the data that is planned to be transmitted to the GU and to the sensors will be packed into a ZigBee frame. A sample packet is shown below.
This is structured in this manner to provide a simple and reproducible method of communication between the base station and the various sensors that are attached to the mote. The sensor type field within this packet provides the ability of error checking as well. If the sensor type from the packet does not match the sensor type that the mote has stored in memory, an appropriate error message will be sent, and the system will avoid sending the wrong configuration command or request to a sensor.

![Diagram 5: Optimal design program loop block diagram](image)

This is the current optimal design program loop. It is greatly simplified here for the sake of readability and flexibility. During the formatting of the ZigBee frame and the parsing of sensor commands, extra work will be done behind the scenes to decode the ZigBee escape characters that are present. This is necessary because sensor data or commands might contain special bytes that are reserved for ZigBee delimiters. The escape character mode on the XBee provides a method to encode these characters in order to prevent any parsing
errors from occurring. Because the current firmware and GUI do not use this escape character API mode, additional work is planned to enable it. This means that functions will be written to encode and decode the escaped characters on the transmission and receiving end of the ZigBee frames. Additional network health data will be requested from the base station through ZigBee AT commands. These will be transmitted within the ZigBee Remote AT Command API frame and should deliver some network metrics back to the base station. This design is based in-large off the two prior teams’ work on this project.

2.1.2.3 Sensor Mounting System

To ensure that the system closely replicates the environment of the ISS, the team will need to replicate both the positioning of the nodes, as well as the air flow they experience. To do this, three different mounting designs have been proposed, coupled with three different housing designs. For the sake of brevity, these designs are shown together. However, it is important to know that each housing design could work with each mounting design, except for the restraining housing system which is incompatible with the table mounting system. Each mounting system would be spaced to best mimic the layout of the ISS.

The table mount is what it sounds like, marking spots to place the motes with a piece of tape to verify the separation is correct on a table, then placing the sensor system on that piece of tape. This is the simplest mote placement system as well as the cheapest, but it is more likely to be accidentally moved and is more prone to incorrect airflow. This placement system is paired with the configurable mounting system seen to the side. Using the configurable housing system will enable the ability to set how restrictive the airflow is by choosing whether to put the walls up or not. This mounting system would be 3-D printed in multiple pieces.

Figure 11: a CAD model of the configurable housing design
Some benefits of this housing system are it works well to house any needed electronics. By being able to remove the walls, the team can quickly test various computer connectivity scenarios enabling a better understanding of how the wireless connectivity works through various barriers. This housing system also offers easy access to any part of the entire mote system allowing better management of any problems with individual motes when they arise.

Some drawbacks to this system are that it would require the most amount of manufacturing work to be in usable condition. Though it is important to mention that this extra work will not be greater than a couple of hours. This is due to being able to set the 3D printer printing while working on other sections of the system. Assembly is anticipated to be quick and simple as well as this design was made with assembly in mind. This system of housing does need to have at least one of its sides relatively parallel with the ground to keep it all together and be fully configurable.

The next placement system would add complexity as it would now include a stand for the housing as shown to the side. This platform would allow for the mote to be attached at a higher height so that it would be less likely to be interfered with.

![Figure 12: a CAD model of the pole and platform](image)

Some benefits of the pole and platform placement are that the mote would be able to set whatever height is needed to better model the position onboard the ISS. Having this stand-based system means that the sensor can be placed and will not move for the duration of the test. By having it be a narrow pole it would likely take up less area than the mote alone reducing its interference with other groups.
Some drawbacks to this placement system are that it would be unwieldy to move after being placed down. With the mote on top of the platform this method becomes slightly top heavy needing more support or some other way to combat this. This system would also have a base of some sort which could be a tripping hazard. With the mote being at the top of the platform any necessary maintenance becomes difficult.

The housing system proposed for this alternative design features a specifically designed 3-D printed shell that would be made for each mote based on specifications of the actual mounting system. A sample of what the design would resemble is shown below.

![Figure 13: an example of the molded housing](image)

Some of the benefits of this system are that due to designing the housing for each mote specifically this is the housing that would fit best for each mote. Fully surrounding the mote limits possible dust, debris, or water from tampering with collected test data. The team would also be able to determine the effects of walls resembling the ultimate housing better as that can be included in the design of the housing.

Some drawbacks of this design are that each mote will not take time to design a housing for instead of a general flexible design. With the housing nearly fully covering the mote it becomes much harder to access for maintenance. It also leaves the sensor less exposed. If the ultimate housing is changed, the team would also have to redesign the housing for each mote - making the testing process longer.

The next placement system would add complexity as it is largely based upon the pole and platform placement, but this design adds telescoping. This platform would allow for the mote to be attached at a higher height so that it would be less likely to be interfered with. This height could be changed both to be a more accurate mimicry and to also store more compactly.
Figure 14: a CAD model of the telescoping pole placement system

Some benefits of the telescoping feature are stated above. Specifically, disassembly is possible and can decrease space needed for storage. The telescoping system also benefits from the pole and platform placement system. By setting the height in the design stage this mounting system can better mimic the position onboard the ISS. Having this stand-based system means that the sensor can be placed once and remain steady for the duration of the test. By having it be a narrow pole it would likely take up less area than the mote alone reducing its interference with other groups.

Some drawbacks to this placement system are that, due to the mote on top of the platform, this method becomes slightly top-heavy which necessitates more support or some other way to combat this. The top heaviness is of great concern as the pole will be narrower than that of the pole and platform to allow it to telescope. This system would also have a base of some sort which could be a tripping hazard.

The restraints housing system is the least complicated as it is just some flexible material wrapped around the motes and tightened against the pole so that the pressure and friction holds up the system. An example of this can be seen to the side where a flash drive is secured to a pole by two pieces of rope so that it stays upright.

Some benefits of this design are that it would be the easiest to make. This housing system also allows for the easiest access for maintenance and other needs including increasing the amount of air flow the sensors would see. The team would also be able to get this housing system working the most quickly.
Some drawbacks to the restraint housing setup are that the restraints would introduce extra pressure on the components to hold it to the pole. If the team chooses this design, they will need to decide where the non-sensor components would go as the more mass the greater the pressure needed. With the sensor being rectangular the stress from the restraints would concentrate on the corners and increase the chance of breaking the sensor which is not ideal.

The ultimate design for the mounting system was chosen to be a combination of the configurable mounting system in conjunction with the telescoping platform. The reason the configurable mounting system was chosen was because it gave the most flexibility with design while not being needlessly unique and this method of mounting did not put unnecessary stress on part of the sensor. The telescoping mount was chosen because it allowed for the best isolation of airflow while still being able to be stored easily. Both designs will allow the team to create and maneuver various mounts for the sensors so that the sensors might successfully simulate their position on the ISS.

2.1.2.4 Automated Breathing Simulator

One of the most crucial design ideas to decide on is how to store the CO₂; this is because at standard atmospheric pressure, a minimum of 237.6 liters of CO₂ gas per test is required. The team would like to be able to perform more than one test. Luckily most gases, CO₂ included, decrease in volume when they are put under pressure. This can drastically reduce the volumetric amount of storage needed. This ultimately leads to two clear alternatives, a lower volume tank that would be replaced with each test or a high-capacity tank that would house all the gas needed. To streamline the development process, the tank selection will be limited to those already on the market.

*Figure 15: Tank Selection Determination Graph*
The graph was made using Equation 1 which was based on constraints from [7], [8], and [9].

\[ \text{Weight}_{\text{Canister}} = \frac{\text{Density}_{\text{CO}_2} \cdot \text{rate}_{\text{breathing}} \cdot \text{Volume}_{\text{breath}} \cdot \text{Concentration}_{\text{breath CO}_2} \cdot \text{Time}_{\text{tests}} \cdot \text{Pressure}_{\text{atmospheric}} \cdot \text{Gravity}_{\text{Earth}}}{\text{Pressure}_{\text{container}} \cdot \text{Concentration}_{\text{CO}_2}} \]

\textbf{Equation 1: equation to find the weight of a pressurized canister}

One of the first questions to be asked when selecting a replaceable tank is what concentration of CO$_2$ to use since that will dramatically affect the volume of the canister. By restricting to only available canisters to those on the market it was found that there are not good options for a CO$_2$ concentration around 100% as there are no manufacturers who produce less than 1-pound canisters meant for CO$_2$ storage for reasonable prices. This suggests choosing a concentration of CO$_2$ of 5% as going lower would not be cost-effective, going higher quickly leads to unmanufactured canister sizes, and by choosing a concentration of 5%, which mimics the CO$_2$ content of human exhalation, no extra complexity is needed to achieve the desired concentration of CO$_2$ as it is already achieved.

After searching for low-capacity replaceable tanks, the canisters that best fit the design requirements for the replaceable tank would be the Tippmann 48/3000 Aluminum Compressed Air tank shown to the side. The cost of a single tank is currently around $50-$75 making getting multiple (2-3) tanks have a resulting cost of $150-$225. The reason behind utilizing multiple tanks is to run tests back-to-back without having to wait for a refill. By purchasing 2-3 tanks instead of just one, redundancy is also built into the system meaning that if one tank breaks, testing will still move forward.
Figure 17: Example Gas Canister

Some benefits of using the low-capacity replaceable tank are that the initial investment will be cheaper, even when considering multiple small tanks compared to one high-capacity tank. The smaller tanks, due to their size, will also be easier to store both while in use and while not in use. Due to the minimum weight of the low-capacity replaceable tank, the ABS would be more easily maneuverable. In choosing to use the low-capacity replaceable tank, the team would be better able to make more than one ABS due to factors stated above as well as lower safety concerns.

Some drawbacks to using the low-capacity replaceable tank would be that they have an ongoing cost, both in time and money regarding refilling the tanks. The team also needs to find a location that would refill the tanks because NIU does not have the resources to perform this procedure on campus. Each test could not be automated with an ABS using the low-capacity replaceable tank as the tank would need to be replaced before each test could begin. The final drawback would be that the design of the ABS would need to have the ability to remove the canister, which would not be needed on the high-capacity tank.
One of the first questions to be asked when selecting a replaceable tank is what concentration of CO\(_2\) to use since that will dramatically affect the volume that is needed. From the graph above it can be shown that expecting to use a concentration of 5% is not feasible as there is very little chance of getting a 700 lb. high-pressure gas canister approved due to safety concerns as well as space and budget concerns. This suggests choosing a higher concentration would be the most beneficial. The team chose 100% (realistically ~99.9%) concentration of CO\(_2\) because this is the upper limit of concentration and any concentration above 5% will require an additional system to create the 5% CO\(_2\) concentration needed for these tests to be as close as possible to human CO\(_2\) exhalation contents. An N of 50 tests was chosen arbitrarily but it is not foreseen that the ABS will be used for more than 10 straight weeks of testing.
After searching for canisters that best fit the design requirements for a high-capacity tank, the suggested option would be the 35 lb. CO₂ Aluminum Cylinder from Beverage Elements shown to the side. 35 lb. gas cylinders rated for 3300 psi CO₂ storage cost around $320-$400 with this model at $315.95. It is important to note that for both this cylinder and the low-capacity replaceable tanks, the costs stated are just for the tanks and do not include shipping, nor any connectors needed.

Some of the benefits of choosing the high-capacity tank is that once filled for the first time, the high-capacity tank should never need to be filled again, cutting time and money that would otherwise be spent on those refills. Another benefit of choosing the high-capacity tank is that once it is installed, the high-capacity tank would not need to be interacted with.

Some drawbacks of choosing the high-capacity tank would be that this tank is larger, bringing an inherently greater level of safety risk. The high-capacity tank, even when unfilled, would not be light and with the added weight of the CO₂ the canister would grow to be quite heavy—perhaps 50 lb. or heavier. If the high-capacity tank were chosen, that would mean that due to the weight and safety concerns the ABS would most likely not be mobile. The sheer size of the high-capacity tank would also make it difficult to work around. With the high-capacity tank having a CO₂ concentration of not 5%, the team might have to create a system to mix the CO₂ with the atmospheric air to achieve the desired 5% concentration.

The 2.5 lbs. canister containing 5% CO₂ was ultimately chosen as it allows for the system to be more portable as well as allowing for the system to be safer and cheaper.

The ABS system will be tested at various stages throughout the development cycle. One of the first tests will be ensuring that the system can handle pressurized gases exhaled through a nozzle. It will be tested to handle CO₂ gas chemically, but this would be mostly an abundance of caution as care to select non-reactive materials would be used. After ensuring the frame is chemically sound as well as structurally sound, the gas canister will be installed, and some low-pressure tests will be recorded until it performs well enough to test with full pressure. Once the system has proven reliable, a 1-hour test will be conducted, if that results positively, an initial overnight test will be performed. After this test the frame will be thoroughly inspected as well as the canister to make sure no complications arose over the night. If the ABS passes that inspection, it would be approved to begin use in experimentation of the sensors.
2.1.2.5 Printed Circuit Board

Previously, a JPL laser driver PCB integrated with the SAM-S70 chip has been the hardware for the mote. Using the general design requirements outlined by the client, as well as the laser driver PCB as a reference, the new design will take what is deemed required or generally helpful and create a PCB containing a lower range of functionality that prioritizes the primary design objectives. This is important as the laser driver boards are much more expensive than what is necessary and are underutilized by the scope of this design project. The optimal design for this aspect of the project should be designed with the following list of requirements. Firstly, the PCB must have pin connections for a power source, XBee RF antenna module, and off-board peripheral devices. When considering which level of functionality is necessary for progressing successfully with this project, additional peripheral devices and their connections were evaluated based on necessity and improvement to the overall device. The optimal design should be capable of retrieving data from and communicating with the proposed sensors which include the TMW and NDIR sensors. Also, the client expressed interest in additional peripheral setup which—while not required—certainly influenced the overall final decision on the optimal design.

The base functionality prioritizes the necessary functionality defined by the client and does not risk overcomplicating or delaying the implementation of the mote for testing. Given that the base design elements are included in the more complex designs, this design does not provide many advantages outside of reducing time or monetary costs. Other designs which derive themselves from this contain more explicit advantages and disadvantages and go beyond the client’s completion criteria.
MicroSD (Secure Digital) ports such as the one above is one of the additional peripheral alternatives considered by the team when designing the PCB. The addition of a removeable and accessible memory card which can contain a myriad of things from firmware updates to storage for unsent data recorded during base station communication outages. These aspects improve the overall design and provide the PCB with a more comprehensive way to record data for longer periods of time. In contrast to the improvements provided via the microSD, the disadvantages of this additional peripheral are primarily concerns that the integration of the port would be well beyond the technical knowledge of the team. Given the large amount of research required to fully understand more complex peripherals such as this, the addition of a microSD may be too complicated for the scope of this project. Despite these concerns, the team believes that the addition of this port is justified and will create more cohesive data transmission worth the potential risks.
The circuit above shows the general mock design for the microSD port and includes connections to the High-Speed Multimedia Card Interface (HSMCI) peripheral pins located on the SAM-S70 chip. The DAT(0-3) pins are responsible for transmission of data to the microSD card, with a configurable option to detect the insertion of a card via the CD/DAT3 pin connection. All data pins are bussed to their respective I/O (Input/Output) pins on the SAM-S70, as well as the CLK and CMD pins. The CLK pin provides a method of timing the transfer of data to the card, while the CMD pin allows for bidirectional commands to be sent between the microcontroller and the SD port.
Another variation on the initial design would include a battery or wall outlet power supply which would replace the current design’s off-board power supply connected via wires. Instead, this design would utilize a Universal Serial Bus (USB) power circuit similar to the one shown in the figure above to provide power to the board with a much more user-friendly option. One such advantage of this addition is that the user would have a much simpler way of connecting power to the board. Use of this additional functionality is relatively inexpensive and provides a connection to the board which is generally easier to access than the base power alternative. Despite the relatively positive addition this alternative would provide, the USB circuit is relatively unnecessary as the JPL laboratory has already provided the team with a voltage booster/bumper circuit that provides an accurate conversion of 5V to the board meaning any alternative power source would have to exclude this converter.

Given the timeframe of this project and the overall importance of the alternative peripherals listed, the design team believes that the inclusion of the microSD card interface would be a viable addition to the PCB. With the emphasis on real-time data collection expressed by the client, the addition of this storage method mitigates loss of packets. Additionally, firmware may need to be updated as time goes on which would no longer require an onsite Atmel In-circuit Emulation (ICE) with the inclusion of the microSD addition. Conversely, the strict power constraint set by the client’s current provided technology makes the addition of a USB power supply difficult to implement. In order to do this the power constraints and technology would require a redo beyond
the foreseeable scope of this project. Given the analysis made on both peripheral alternatives, the team has decided the optimal design would include the microSD alongside the base design features.

2.2 Prototype

2.2.1 Graphical User Interface

The GUI received at the start of this academic year had undergone major revisions by Alexander Hart, an intern at JPL, during the summer between last year’s work and this year. This updated version removed the partially functioning 2D ISS map and significantly refactored the behind-the-scenes data processing in the GUI to improve its flow. At the time, it was only capable of plotting data from the first mote in the network, and was designed to plot pseudo data for CO$_2$, humidity (“water”), temperature, and pressure. Each type of data had its own set of functions to process and plot it, resulting in many functions having four nearly identical duplicates within the codebase. This was the version of the GUI received by this year’s senior design team:

![GUI Image](image)

*Figure 21: The GUI received by this year’s senior design team. From top to bottom, it has plots for pseudo data of CO$_2$ concentration, humidity (“water”), temperature, and pressure.*

Before making any steps towards an ISS map, it was determined that firstly the GUI should be able to handle multiple motes and incorporate data from the anemometers. To achieve this, many of the aforementioned duplicate functions were refactored to have a single flexible version which could support any type of data,
rather than creating even more duplicates for the new anemometer data. Drop-down menus were then created to allow selection of motes by their MAC address, and the appropriate number and type of plots were added for the new sensor data. Plots were segregated by whether they were CO₂ data or anemometer data, allowing the user to have different motes for each (though one mote may serve both roles if it has both sensors). In addition, the data payload structure has been redefined, which required updates to the GUI’s payload parsing code. At the time of writing, the GUI has successfully plotted pseudo data generated and sent by one mote in the network for all new data types:

![Atmospheric Telemetry Viewer](image)

**Figure 22: Plotting pseudo data for all data types. This pseudo data was generated by one mote, packetized, sent through the network, parsed by the GUI, and plotted.**

Issues with the anemometer sensor prevented the team from pulling real data from it. However, the team was able to successfully interface with a Coz-IR CO₂ sensor, which provided CO₂, temperature, and relative humidity data. The following figure shows a spike in CO₂ concentration and relative humidity following a breath on the sensor, as well as measurements of ambient temperature. Also showcased are auto-scaling Y axes, which resize to fit the data. No pseudo data was used at the time for the other plots.
Figure 23: Plotting real data for CO2, temperature, and humidity with auto-scaling Y axes. The rise in data is from measurements of a human exhalation. No pseudo data was used for other plots.

This auto-scaling works, though it has room for improvements. A “cushion” should be applied above and below the line, which both prevents undesirable line contact with the edges of the plot and prevents unreasonably narrow Y ranges, as can be seen in the temperature and humidity plots in the above figure. The auto-scaling also currently uses all data, rather than only the data visible. This means that it can only increase in range, and does not decrease in range when the extra space is no longer needed.

One substantial limitation of the GUI prototype at the time of writing is that it acquires a “data lag” over time, with a root cause currently not known. What this data lag means is that when the GUI is initially run, the sensor data is displayed in real-time. As the GUI runs for longer, however, an increasingly long delay develops between the time of data sampling and the time of plotting that data. It has been observed that the number of packet read attempts from serial grows faster than the number of packets processed, which is suspected but not confirmed to be related to the issue:
Table 2: A sample showing a comparison of packets attempted to be read from serial versus packets processed. Observe that the difference grows over time.

<table>
<thead>
<tr>
<th>Time Elapsed Since First Read (mm:ss)</th>
<th>01:00</th>
<th>02:00</th>
<th>03:00</th>
<th>04:00</th>
<th>05:00</th>
<th>06:00</th>
<th>07:00</th>
<th>08:00</th>
<th>09:00</th>
<th>10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Serial Read Attempts</td>
<td>123</td>
<td>235</td>
<td>344</td>
<td>460</td>
<td>576</td>
<td>680</td>
<td>771</td>
<td>866</td>
<td>991</td>
<td>1,106</td>
</tr>
<tr>
<td>Packets Processed</td>
<td>102</td>
<td>203</td>
<td>304</td>
<td>405</td>
<td>505</td>
<td>596</td>
<td>673</td>
<td>761</td>
<td>861</td>
<td>961</td>
</tr>
<tr>
<td>Difference</td>
<td>21</td>
<td>32</td>
<td>40</td>
<td>55</td>
<td>71</td>
<td>84</td>
<td>98</td>
<td>105</td>
<td>130</td>
<td>145</td>
</tr>
</tbody>
</table>

When investigating how this occurs, it was noticed that each time “Packet Serial Read Attempts” increments while “Packets Processed” doesn’t, it is because of a “truncated” packet read. Interestingly, if a packet isn’t read in full, it is retried from the beginning rather than being skipped or messing up the read alignment with
the start of a packet. This example output below highlights this. In this version of the code, the “timestamps” in each subpayload are actually an index, so they should increment with each subpayload. Using this fact, one can see that the truncated packet containing index 000017b6 is retried from the beginning. This packet took 2 reads but 1 processing, which explains the above trend. It does not, however, explain why some packets fail to read in full the first time, nor does it make clear why a data lag develops over time. Further investigation is needed to identify the root cause for the data lag.

<table>
<thead>
<tr>
<th>Annotations</th>
<th>Example Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example of a truncated packet</td>
<td>MainWindow::readData() called (elapsed time: 1524 ms) MyFrameParser::serialRead() numRead = 4 (should be numSubpayloadsTotal/numSubpayloadsPerPacket) MyFrameParser::FetchFrame() called MyFrameParser::FetchFrame() receiveBuffer: 0000: 37 65 30 30 36 34 39 31 30 30 31 33 61 32 30 30</td>
</tr>
<tr>
<td>Note the index</td>
<td>0010: 34 31 63 33 34 64 35 33 35 31 37 33 65 38 65 38</td>
</tr>
<tr>
<td></td>
<td>0020: 30 30 31 31 63 31 30 35 30 31 61 66 30 35 63 36</td>
</tr>
<tr>
<td></td>
<td>0030: 30 30 30 30 30 31 37 62 36 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0040: 62 65 36 36 36 36 34 31</td>
</tr>
<tr>
<td></td>
<td>MyFrameParser::FetchFrame() Error: receiveBufferSize (36) is &lt; length (104), returning out of FetchFrame().</td>
</tr>
<tr>
<td>Example of a complete packet</td>
<td>MainWindow::readData() called (elapsed time: 1544 ms) MyFrameParser::serialRead() numRead = 5 (should be numSubpayloadsTotal/numSubpayloadsPerPacket) MyFrameParser::FetchFrame() called MyFrameParser::FetchFrame() receiveBuffer: 0000: 37 65 30 30 36 34 39 31 30 30 31 33 61 32 30 30</td>
</tr>
<tr>
<td>Index is repeated, indicating these are the same packet</td>
<td>0010: 34 31 63 33 34 64 35 33 35 31 37 33 65 38 65 38</td>
</tr>
<tr>
<td></td>
<td>0020: 30 30 31 31 63 31 30 35 30 31 61 66 30 35 63 36</td>
</tr>
<tr>
<td></td>
<td>0030: 30 30 30 30 30 31 37 62 36 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0040: 62 65 36 36 36 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0050: 30 30 30 30 30 31 37 62 37 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0060: 62 65 36 36 36 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0070: 30 30 30 30 30 31 37 62 38 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0080: 62 65 36 36 36 36 34 31</td>
</tr>
<tr>
<td></td>
<td>0090: 30 30 30 30 30 31 37 62 39 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>00a0: 62 65 36 36 36 36 34 31</td>
</tr>
<tr>
<td></td>
<td>00b0: 30 30 30 30 30 31 37 62 61 30 32 30 36 34 31</td>
</tr>
<tr>
<td></td>
<td>00c0: 62 65 36 36 36 36 34 31</td>
</tr>
</tbody>
</table>

Figure 24: Example output demonstrating a truncated and then retried packet.

Other aspects of the GUI needing improvement include testing the GUI with multiple motes in the network (there was only one functioning at the time), testing with real anemometer data, re-implementing the 2D map of the ISS per the request of the client, and implementing a 3D map of the ISS.
2.2.2 Mote Firmware

The existing mote firmware went through a major overhaul from the version received towards the end of the Fall semester. The main goal of the changes was to improve the code’s readability and allow the project to be more maintainable and developer-friendly for the future. Some of the issues that were addressed in this endeavor include removing as much duplicate code as possible, removing as many redundant global variables as possible, replacing unnecessary register calls with Atmel Software Framework library calls when applicable, and fixing errors such as a lack of cache coherency.

Another reason for the vast amount of change was because the majority of the existing codebase contained files that were not being used by the current senior design project. This bloat appeared to be inherited code from another project entirely. All this wasted clutter can be incredibly distracting to a new developer, and this slowed down development drastically at the start of the Spring semester.

Currently, the prototype firmware is built for a completely different target than the previous firmware. This is due to the lack of documentation and support for the custom board that was previously used for this project. An evaluation board called the SAME70 Xplained was provided to the team, which had much more documentation and access to invaluable reset pins in the case of a firmware emergency. For the prototype to be built on this new target, rather than refactor the prior bloated codebase, the decision was made to rebuild from scratch. The new firmware architecture can be seen in the following figure.
This decision in rebuilding the project allowed for several architectural design choices to be made regarding the file structure and build system. The prototype uses GNU make for building, and OpenOCD for flashing to the evaluation board. This code was written with modularity in mind, so new device drivers can be implemented as quickly as possible for future developers. Previously all drivers were placed in a single directory mixed with unneeded peripheral drivers. The peripheral drivers are described as unneeded because the ASF (Atmel Software Framework) already contained peripheral drivers for the code.
One example of a needed development feature that was implemented in the prototype is the logger utility. Previously, the firmware used functions with duplicate code and blocking loops that output a variable one byte at a time. This data was sent to a designated debug UART port which required multiple wires to be soldered onto the custom PTLS board before it could be connected to a serial terminal. Using the new evaluation board in conjunction with the ASF serial I/O library, a single micro-USB cable provides serial debugging info to a terminal which can be called via one of three logger macros. These were designed with developer usability in mind, as they allow for robust input and versatile output.
Figure 28: Example of early prototype log output
As one can see, the logger allows for much more rapid development by enabling quick prototypes of code with practically instantaneous verification or failure. This aided immensely in adding all the features from the previous codebase, especially when working with the DMA (Direct Memory Access) controller.
**Figure 30**: The logger utility shows data from the CoZ-IR sensor in the form of CO\textsubscript{2} ppm, temperature, and humidity

Ultimately, the only new *visible* feature from the prototype is data collection from the CoZ-IR sensor and the colorful logging utility. Unfortunately, the team encountered issues with the anemometer sensor and was unable to get any data output from it. However, implementing an anemometer driver with the new firmware project structure should take much less time with the new build environment than with the old one, and this rapid increase in development time will allow for further refinement of the system in the future. Therefore, the main takeaways of the new firmware are not necessarily improvements of the firmware features, but the process of developing the firmware and the environment in which one does so.

Future work on the firmware involves but is not limited to the following: updating utilities like the vector data structure to be memory safe and not allocate any memory; incorporating DMA channels for all input & output to get more CPU burst time; adding new sensor drivers, including one for the Trisonica Mini anemometer; and calibrating the RTC (Real-Time Clock) for accuracy. Additionally, the payload queue needs to be finished as there exists a bug where a recurring payload will get zeroed out when the queue capacity is greater than 15. The feature for flushing payloads needs to be reimplemented as well. Previous code was used as much as possible for reference.

### 2.2.3 Sensor Mounting System

Currently the sensor mount system consists of three telescoping legs that can be adjusted to improve stability as well as to the desired height. The PVC pipes are inserted into a baseplate designed to hold the electronic
components, specifically the TriSonica Wind and Weather sensor, the PCB, the XBee transceiver, and the CoZ-IR sensor. Currently stability is a concern as it is only a three-point system; however, especially with the regular sized model, the large area between the feet allows for a large spectrum of stability and since this is meant to be an inexpensive test stand that sits in a place that will not be disturbed, stability can be factored against cost, justifying the three-point stability system.

2.2.4 Automated Breathing Simulator

The current iteration of the automated breathing simulator (ABS) comprises the ABS frame and a gas canister, capable of pressurization up to 3,000 PSI. The design of the canister frame prioritizes safety by minimizing potential risks of damage. This is primarily achieved through the use of a rigid frame with a bottom-heavy design, which ensures stability and prevents unintentional tipping of the canister. The three bottom holders and end supports also serve to prevent tipping and resting on the side as the canister degasses, with the bottom holder ensuring that the canister rests on itself rather than on the more fragile end bolt.

At present, the prototype ABS has a rectangular prism frame, measuring 14 inches in height, 7 inches in width, and 7 inches in length. Testing at high pressures has yet to be conducted due to logistical delays in the arrival of gas valves, which would allow for safe pressurization to higher levels. Nonetheless, preliminary testing suggests that the ABS should be able to successfully withstand highly pressurized CO₂ for use in tests.
One of the most significant safety concerns associated with the high-pressure system is the potential for the gas canister to tip over, resulting in unwanted and unsafe gas release creating the possibility of extreme temperatures which could lead to damage of both the abs and the system as a whole. Therefore, the tipping angle of the canister was considered a crucial safety factor that needed to be minimized and documented. The current version of the ABS has a tipping angle ranging from 20 to 30 degrees, enabling it to tilt significantly and return to an upright position, which is the preferred configuration. This design choice is optimal because a horizontally positioned gas canister would pose numerous safety risks.
2.2.5 Mote Electrical Components

Since the initial concept phase of the project, the mote electronic hardware composition has undergone several revisions to accommodate the completion of the project. This year’s project initially focused on designing a new PCB that would integrate the SAM E70 microcontroller and a number of onboard peripherals. However, as the project continued, it became apparent that the design of such a large-scale requirement for the project would be too time consuming and would limit the overall progress of the final design. Because of this, the mote hardware focus would be shifted towards using already existing hardware alongside the sensors provided by the client to design a circuit for the mote to run the firmware from. The determination was that the mote hardware setup would need to include proper connections between three primary parts of the design. These three parts included: Sensor elements, transceiver, and a firmware base. Given the use of UART peripherals from both the provided XBee transceiver and all 3 sensors provided by the client, it was determined that any circuit board used to complete this design would use UART connections to both additional elements of the mote.
The primary sensor used for testing was the CoZ-IR CO₂ sensor, which includes measurement capabilities for CO₂ as well as temperature and humidity. This is particularly useful as this information can be used to
determine the concentration level of CO₂ within a certain location. The use of an anemometer, while originally planned, was unable to be utilized due to several hardware issues related to the TriSonica Mini provided by the client. The results of these issues were a TriSonica sensor which would not properly respond to commands and was unable to reach the bootloader based on testing done by the team.
3 REALISTIC CONSTRAINTS

3.1 Engineering Standards

The following engineering standards are relevant to the project:

- ZigBee Protocol (based on IEEE 802.15.4): This is the communication protocol used by the project’s XBee modules and therefore influenced the design and operation of the WSN.

- The FCC regulates RF devices contained in electronic-electrical products that are capable of emitting RF energy by radiation, conduction, or other means. These products have the potential to cause interference to radio services operating in the RF range of 9 kHz to 3000 GHz [21]. The Zigbee modules used in the project provide wireless connectivity to devices in the mesh network. The wireless Zigbee module operates within the Industrial, Scientific, and Medical (ISM) 2.4 GHz frequency band.

- When deciding on proper connectors for gas canisters, ASTM C1541 +Redline standards were followed. This standard focuses on connectors and fittings for shielded transition couplings which is what the system used.

3.2 Economic Constraints

- JPL did not establish a budget for this project. Last year the project had a total expenditure of $17,721.99, which was used as a tentative estimate for the anticipated size of this year’s budget. To stay within a reasonable budget, a self-imposed limit of $1,000 was set by this year’s team.

- The WSN uses multiple sensors and microcontrollers that JPL developed. Hence, the cost of sensors and microcontrollers did not affect this project. These modules were provided by JPL. However, the materials for the ABS needed to be purchased.

3.3 Environmental Constraints

-Launching items into space is very expensive, and increased mass of the system requires more fuel to reach orbit, increasing emissions of greenhouse gases. For this reason, in the future, materials and designs should be chosen with attention to minimizing mass.

- One of the environmental constraints for the WSN is RF spectrum congestion. At the ISS, crew use
various radio wave devices for their work. There is also a Wi-Fi network installed onboard the ISS, as well as Bluetooth [20]. Hence, the developed WSN must not interfere with any existing networks onboard the ISS.

- Power efficiency optimization is critical for the ISS. The ISS obtains its power by using large numbers of solar cells. Hence, energy and electrical power are scarce and should be used carefully for all the systems. It is essential to keep the power consumption of the WSN at a minimum threshold.

- The design of the WSN must be reliable to prevent any chance of failure. Even though the engineering design is reliable, prevention of failure is almost impossible. All electrical systems will fail sooner or later. Hence, the WSN must be easy to repair/troubleshoot in case of any failure.

- The most significant environmental constraint for the WSN is the materials used to construct the ISS. The WSN will need to have high reliability and validity when collecting data from any node. The ISS has a protective outer and inner layer primarily made of high-strength aluminum alloy. Aluminum metal has excellent RF shielding characteristics. Unfortunately, these characteristics of aluminum also have an impact on the functionality of the WSN. As a result, it can cause the signal to be absorbed or blocked if the doors are closed. Since the WSN uses the same RF spectrum as Wi-Fi, one can expect the network to have similar coverage performance as Wi-Fi devices.

3.4 Sustainability Constraints

- To be environmentally sustainable, manufacturing waste should be minimized where possible, and the operation of the system should be energy efficient. To do this, all parts must have analysis run to ensure that they will work and were not ordered until the parts have been approved by multiple members of the team.

- To be socially sustainable, the system should be reliable and safe, as failures and injuries negatively impact the public perception of an organization. To this end the team has ensured that various engineering and safety standards are followed.

- To be economically sustainable, the system should stay within budget and be compliant with regulations. By offering multiple choices for each design and deciding between them, the team has ensured that the economic sustainability of this project was considered.
3.5 Manufacturability Constraints

- When possible, designs should use the same components and fasteners as other systems on the ISS for easier installation and maintenance. Dr. Christensen has been in constant communication to ensure that ISS standard fasteners are chosen so that crewmembers aboard the ISS are able to easily do maintenance on the network.

- Designs should be easy to manufacture and reproduce, potentially leveraging additive manufacturing as applicable. Various 3D printing practices along with making sure all connectors are chosen from common types ensured that this project can be manufactured easily.

- The WSN project may in the future use custom sensors and microcontrollers that JPL develops. Custom-designed boards and sensors are not available in the market. However, all custom boards and sensors will be provided by JPL. Other hardware such as microcontrollers and XBee modules are widely available in the market.

- To make the ABS manufacturable, it was chosen to make the frame that will house it out of aluminum extrusion, as this material is inexpensive and easy to work with.

3.6 Ethical Considerations and Constraints

- Due to the risk to human life or property in the event of device failures, the system must be designed to minimize damage to the crew and ISS if any failures were to occur. For example, the system should be designed such that electrical failures do not cause an excessively high electrical load, short circuit, or fire.

- Inaccurate sensor measurements may lead to crew believing they are within safe levels of CO₂ when they are not. Care should be taken to validate the accuracy and performance of sensors, and the sensor modules should be capable of detecting and alerting the user to failures, as inaccurate data may be more harmful than no data at all. This year’s senior design team was unable to perform these precautions due to time constraints, so future teams should ensure these tests and features are implemented.

- Privacy concerns are always worth considering when monitoring is introduced into a system, and this
was something JPL requested to be investigated. Based on initial assessment, it is not expected that crewmember privacy will be compromised, as the data is anonymous and not associated to any one person within the ISS.

3.7 Health and Safety Constraints

− While external or noticeable injury may not be apparent due to concentrated buildup of trace CO₂, many additional risk factors such as poor decision-making and work performance are highly influenced by the quantity (ppm) of this gas in the air. Dangerous or fatal conditions caused by impairment of cognitive function—especially in high-risk circumstances—may arise in high levels of CO₂ and therefore must be prevented.

− Each part of this system will ideally be operating within the ISS in the future. For this reason, there are implicit safety conditions associated with the system. One safety concern is the durability and integrity of the individual components and of the entire system. This is a concern for the ISS because if parts of the system break apart, they could easily pose a hazard to more vital systems operating on the station.

− Another safety concern for this project is the reliability of data. If a sensor stops working and the crew of the station are not alerted, they might be at risk of exposure to high CO₂ levels. Therefore, warning lights or indicators of some sort should be developed such that the crew knows when a sensor needs replacing. Additionally, the GUI system should only visualize data from sensors that are operating without failure for the same purpose of providing factual data to its users and should display warnings if a failure is detected.

− Electromagnetic (EM) radiation is a crucial safety constraint for the project. Not all, but some EM radiation can be very harmful to humans. The WSN should not pose any health hazards to the crew working on the ISS. The ZigBee modules currently used in the project have been certified by Electro Magnetic Compatibility (EMC) testing, which provides this assurance.

− Electronic components and semiconductor chips are at the most significant risk of damage by Electrostatic Discharge (ESD). Even though ESD usually causes no permanent harm to the human body, it can give a sizeable and painful electric shock. The casing of the sensor network should protect components and crew from ESD.
− The system should not pose a fire hazard in any electrical failure modes.

− The device should not have sharp edges or other protrusions that can cause serious injury if collided with. To prevent sharp edges catching on people, the housing has been designed so that all sharp points of the sensor motes are contained within the housing. The housing itself will have rounded edges to lessen the chance of injury.

− Incomplete CO₂ detection allows for hazardous conditions on the ISS to go unnoticed; therefore, the system should alert its crew when sensors are malfunctioning, and potentially have airflow sensors positioned in a way that introduces redundancy into the system.

− A safety constraint for the wireless network is network security. Securing the wireless network from hackers and intruders is vital as their presence could pose a huge threat to the ISS. An unsecured network can result in data breaches and misleading information, and this could negatively impact the health of crew onboard the ISS. Fortunately, the ZigBee protocol offers security features which can help mitigate this risk.

− High pressure gasses will be used by the ABS. For this reason, safety protocols must be followed to avoid harm such as canister explosions, skin penetration by compressed air causing air embolism, and unexpected quickly accelerated objects (e.g., tubes whipping around, valves blown off).

3.8 Societal Constraints

− Failures have a significant impact on reputation in the public eye. For this reason, it is important for this project to be reliable to protect the public image of JPL and NASA. The integrity of these organizations is important due to the support they garner for further development and enhancement of space exploration.

3.9 Political Constraints

− The ISS has been in orbit for more than two decades. As a result, systems are failing as the craft becomes more worn over the years. NASA has reportedly been trying to deorbit the vessel since 2015, however the current deorbit plan will take place in 2031. Thus, if this project were to be deployed on the ISS, it might not last very long. For this reason, politicians might negate any new projects from
being launched onto the ISS. However, there also might be enough time before the deorbit to record enough meaningful data about the CO$_2$ airflow such that JPL or NASA deems it worthy of deployment. At this point in time, it is not known.

- It is important to note that the ISS is a facility representing multiple countries including the US. While the system is being developed by the US, it is to be used by all who are part of the ISS. Therefore, the system should not encroach on international security boundaries set by other countries as the system will be actively logging data on the ISS. The system should also fit within the values of the current NASA leadership.
4 SAFETY ISSUES

− The primary safety concern involved in this project is avoiding electrical shock. The bulk of the system is built of electrical components. A microcontroller attached to an end node is powered via power supply plugged into an AC wall outlet. This is true for other devices sourcing power from wall outlets, such as the base station and other routers. Any person dealing with such power should not touch the conducting leads of the power supply while plugging into an outlet. All internal electrical components should not be touched while turned on or plugged in. Ensure all active circuitry is grounded and properly wired to avoid injury.

− End devices in the network will be contained in housings. Each housing will be 3-D printed. To avoid injury, the team will be trained to use the 3-D printer safely.

− Sensor modules mounted to walls may fall if not mounted properly. If any module is mounted to a wall, it should be done securely.

− The ABS will periodically release gas from a high-pressure source. This has the potential to be a safety concern. To combat this, most of the tests will be run under direct supervision to ensure others do not get hit or run overnight when there is a lower chance of students being in the way. This pressure will also not be extremely great under normal operations as it is lessened through pressure regulators to better reflect the human breath.

− For testing, components will be elevated atop a highly elevated stand, this should be undisturbed and may cause damage to the components as well as those nearby if the stand is tipped over. Safety warnings should be provided on this stand to prevent those passing by from accidentally interacting with the stand.

5 IMPACT OF ENGINEERING SOLUTIONS

The impact of this project is fairly limited in its scope, as the system is primarily designed only for the ISS. It is not impossible, however, that the system could be used in other locations, both at JPL and the world at large. One such example is CO₂ concentration and distribution measurements within commercial airplane or train cabins, which impact the comfort of crew and passengers. Because it is planned for the sensor modules to be
able to accept any sensors that output serial data, the WSN can also be used in applications that need to monitor for other gases. This is especially important for odorless hazardous gases, as these can go unnoticed until it is too late. For example, this system can be implemented in manufacturing settings that produce carbon monoxide (CO) as a byproduct, which is an odorless and potentially fatal gas. This system may also be implemented in enclosed locations like underground cellars to detect low O₂ levels, which may go unnoticed if CO₂ levels are also low as the human body only detects poor breathing conditions via blood CO₂ levels. While not novel ([14], [16]), a system which can monitor air quality does pose value in many circumstances. Furthermore, the usage of this system may be expanded to any aerospace or nautical environment both commercial and private where buildup of CO₂ can occur which is particularly useful as different geometrically shaped areas can have different locations of trace gas buildups especially with variations in filtering quality. This reapplication also has usage for other more harmful gases if adjustments are made to the integrated sensors.

6 LIFE‐LONG LEARNING

This project is full of content which the team did not know before and either have learned or will learn during the project’s execution. For example, research has been conducted about the human body’s respiratory system and how it interacts with CO₂. The Zigbee communication protocol is currently being researched, specifically the types of supported frames, network setup and integration with XBee modules. The framework which the GUI is built from is new to us, and the team will be learning a lot about how to develop within it. There has been discussion on how to design the firmware in such a manner so that it can accept newly defined sensor types, and this has been pushing limits of how to design for future improvements.

A large portion of the firmware development involved learning embedded concepts such as direct memory access (DMA) and serial communication through UART, specifically their implementations on the SAME70 microcontroller. A tool for programming embedded devices called “openocd” was learned and implemented in the project for faster debugging and development.

Design processes, such as the design of the PCB, are introductions to the concept of applied circuit design and they will teach those who find the knowledge valuable a good amount about developing a digital circuit from scratch. Having novel experience with such a process means it is expected to take the team a large amount of
time to properly understand this process. Having a focus on integrating a microcontroller and peripherals of this device for this project means the process should be a valuable learning experience for this design skill.

Understanding of how to utilize fluid analysis software namely COMSOL was gained during this project as it was used to validate the test results of the ABS and sensor system. By working with multiple professors and working through multiple dead end a greater understanding of COMSOL and how to use it was found.

This project has many subunits, and this is a scale of project which each team member has seldom interacted with before, especially in an interdisciplinary setting. It will be a unique challenge to synchronize the work between the ABS, mote housing and mounts, mote hardware, mote firmware, and GUI. Even though each team member seems at first glance to have nearly disjoint tasks, all the systems are interconnected making it imperative to coordinate and collaborate on nearly all aspects of the project. This “silicon” of tasks is a hurdle the team has encountered, and steps were taken to help address it and work more collaboratively. By the end of this project, the team has grown in their abilities to work effectively in teams, even in environments where the project at hand may conceal the benefits of collaboration.

7 BUDGET AND TIMELINE

7.1 Budget

Much of the hardware which this project used is already owned by JPL. Because of this, this project had a limited number of expected purchases. No hard budget cap was specified by JPL. Previous years’ senior design projects spent a total of $17,721.99, which was used as a rough guideline for the funds available for this year’s project. Most purchases for this year’s project pertained to the ABS, specifically the high-pressure valve, with the expended budget for this year’s part of the project totaling $612.39. The justification for each purchase is as follows:

- **PCB:** This was needed to connect the SAME70 chip and XBee RF module to several sensors with input pins, and to a power source.
- **Electronic components:** This included various components which are not already owned and are needed to assemble the PCB, such as sensor input ports, resistors, solder, etc.
- **Aluminum sheet:** This was used to increase the stability of the ABS allowing the gas canister to sit on a solid surface.
− Various PVC piping: This was used for creation of the telescoping remote sensors.
− Various press fit framing: This was used to create the frame of the ABS such that it would be difficult to unintentionally damage or cause the ABS to be in a more precarious position.
− Three-way elbow connectors: This was used to connect the edges of the ABS frame.
− Cushioning foam: This was used to limit the stress that the gas canister tip was under.
− High pressure gas valve: This was used to safely control the gas outlet from the gas canister
− PLA filament: This was used to create part of the ABS and mote housing.

<table>
<thead>
<tr>
<th>Part</th>
<th>Part Description</th>
<th>Specifications</th>
<th>Vendor Option</th>
<th>QTY</th>
<th>Price per Unit</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing system</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>1</td>
<td>N/A</td>
<td>$17,721.99</td>
</tr>
<tr>
<td>First order form (frame/ABS materials)</td>
<td>Various parts needed to create the ABS</td>
<td>See order form 1</td>
<td>McMaster-Carr and JPL</td>
<td>N/A</td>
<td>N/A</td>
<td>$202.31</td>
</tr>
<tr>
<td>Second Order Form (Connectors and ABS Valve)</td>
<td>Various Connectors and ABS Valve</td>
<td>See order form 2</td>
<td>Digi Key, McMaster-Carr, Hatchbox</td>
<td>N/A</td>
<td>N/A</td>
<td>$455.81</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$18,334.38</td>
</tr>
</tbody>
</table>

*Table 4: Overall Budget*

See the appendix for these two order forms.
7.2 Timeline

**Chart 1: Fall Gantt Chart**
Chart 2: Spring GUI Gantt Chart

Chart 3: Spring Firmware Gantt Chart

Chart 4: Spring Mote Housing Gantt Chart
Chart 5: Spring ABS Gantt Chart

Chart 6: Spring Electrical Gantt Chart
8 TEAM MEMBERS CONTRIBUTIONS TO THE PROJECT

8.1 Aaron Beck

Aaron Beck contributed time, research, and effort in understanding the project’s goals and previous work accomplished up to this point. Significant time was spent researching the Trisonica Mini Wind & Weather Sensor, SAM-S70 microcontroller, Telaire-T6615 CO₂ sensor, the Digi XBee RF module, and the ZigBee protocol. Notes and ideas were recorded for the purpose of brainstorming potential firmware solutions to be implemented in the future. Results from this research provided direction for the development of documentation on firmware that became the basis for this report. Overall, these contributions resulted in the development of custom ZigBee Application Programming Interface (API) frame packet delivery units (PDU) that will become the basis for data transmission within the project. Exploring the ZigBee protocol deeper by reading parts of “ZigBee Wireless Networks and Transceivers” by Shahin Farahani provided more insight on the structure of the ZigBee protocol. This resulted in the discovery of methods to transmit additional data amongst the network by supporting custom sensor configuration commands. Aaron collaboratively worked with Drake Provost in brainstorming potential ideas for integration between the GUI and the firmware. He brainstormed hardware and firmware ideas with Kamrin Gustave to better understand the hardware of the system and its future integration with firmware. Aaron met with the team periodically to communicate research results and discuss firmware ideas with the group. He participated in several meetings with Alexander Hart to learn about the existing codebase, determine the build setup, and test existing mote network with the GUI. Aaron contributed to firmware and assigned sections of deliverable papers throughout the Fall semester.

During the Spring semester, Aaron encountered numerous problems with the current project setup and the existing codebase. These problems were severe enough to the point where a rewrite of the codebase was needed for clarity and future maintainability of the project. Additionally, the large IDEs previously used for the project’s development were cumbersome and unreliable as they failed to program the device on numerous occasions. One IDE – Microchip Studio – was not even available for the operating system on Aaron’s development machine because Microchip Studio only runs on Windows. If that wasn’t reason enough, the custom JPL board that had been used for development in the past did not have access to the reset pins. As a remedy, Aaron decided to pivot development to the SAME70 evaluation board that been sent to the team.
The firmware development started from turning on the board’s green LED to implementing all the functionality from previous years plus integration with a real CO$_2$ sensor, the CoZ-IR. In addition to programming the firmware’s features, Aaron devised a build environment for the codebase that can be setup on most common operating systems such as Windows, MacOS, or Linux. This build environment is simpler than installing bloated IDEs and allows for reliable flashing of the microcontroller using a program called “openocd.”

8.2 Chris English

Chris English was tasked with designing a mounting system for both the housing and placement of the sensors. To do this, Chris used his SolidWorks experience to create CAD models of several alternative designs. Ultimately, he provided three different options for a housing system and three different options for a placement system. After conferring with the rest of the team, an optimal design was chosen, after which he continued to finalize the designs to be manufacturable. To do this, Chris created a 3D printed part that allowed for telescoping poles. He also created a base plate in which the electronics could be inserted, holding them securely.

Chris was also in charge of gathering needed 3-D models. Specifically, he found 3-D models of the ISS to be used in the GUI, as well as to be used for determining the placement of the sensor motes in NIU’s lab space to accurately model the ISS on NIU’s campus. While doing this, Chris also discovered multiple papers focusing on the ISS. These papers mostly focused on airflow onboard the ISS, which aided the team’s understanding of how this project’s sensor network will be utilized within the ISS.

Chris was also in charge of designing the ABS, which included determining which kinds of CO$_2$ canisters would be used. To do this, Chris ran calculations determining the economic feasibility of various CO$_2$ concentrations, evaluating different numbers of tests to determine the size of the vessels. He also made sure that any gas canister selected would follow safety standards and be easy to fill and used by any member of the team.

As with the other members of the senior design team, Chris also worked with the team to create various papers and participated in various team meetings to determine the future of the project and provide continuous updates on his progress. Chris oversaw the creation of the Gantt chart and ensured that all weekly meeting presentations were run smoothly and effectively.
Chris successfully designed in manufactured the ABS enabling it to safely hold the JPL provided gas canister. In doing this Chris also modeled the fluid simulation of CO\textsubscript{2} being released from the ABS to validate the real life tests that the system would undergo.

8.3 Kamrin Gustave

Kamrin Gustave made the following contributions throughout the fall semester of the design project. Kamrin researched PCB design standards including JTAG circuit integration, power integration, and microcontroller integration. A majority of the time spent by Kamrin during this semester was focused on developing an educated understanding of the SAM-S70 Microcontroller and the PCB design. Kamrin also spent significant time on researching the various electrical considerations for the Telaire-T6615, SAM-S70, and TriSonica microelectronics. The various topics of information on these electronic devices included power considerations, pin-out functions, data framing, signal frequency constraints, and peripheral integration for the microcontroller. Using this information, Kamrin developed several small-scale electronic diagrams to be integrated into the PCB’s final design including sub-circuits for the JTAG, Power supply, MicroSD card slot, and the XBee RF interfacing. A large portion of the material developed by Kamrin for the PCB was learned through the provided reference material via JPL, such as circuit diagrams for a JPL custom laser driver, as well as the SAM-S70's expansive reference documents. Furthermore, Kamrin also worked with Aaron to a slight degree in researching and beginning work on the firmware, primarily on the register-related aspect of the firmware. As a general expectation, Kamrin had also attended meetings and submitted written material throughout the semester which contained contributions from all four members of the design team. Overall, Kamrin’s contributions for the fall semester revolved primarily on research and development of the mote electronics and is likely to reflect similar areas of focus in the following semester.

During the spring, Kamrin Gustave primarily focused on improving or adjusting the hardware for the mote through various means. These methods include: board preparation, wiring diagrams, sensor testing, header soldering. Kamrin Also worked with Aaron Beck to integrate the firmware onto the hardware and often researched information on things such as Nectar and the Zigbee protocol. Much of the time for Kamrin was spent researching the TriSonica Mini Wind & Weather Sensor, SAM-E70 microcontroller, Telaire-T6615 CO\textsubscript{2} sensor, the Digi XBee RF module as well as various power supplies. He worked alongside Dr. Edward Miguel to
improve hardware setup and Dr. Benedito Fonseca for guidance. Kamrin consistently analyzed and applied changes to the hardware to create the most efficient system within the time to complete the project.

8.4 Drake Provost

Drake Provost has contributed to a variety of aspects of the project. During the fall semester, he studied the previous senior design project’s GUI to understand the areas in which it needed to be improved for this project. Examples of shortcomings included CO$_2$ sensor data not being localized within the ISS, there being a maximum of one sensor per module, no ability to support anemometer GUI visualization or logging within the base station’s memory, the sensor data tab having an inflexible number of sensors able to be visualized, and content being spread over 3 tabs which makes it harder to see all information at once. Following this, Drake designed ways to address these issues, such as by migrating the ISS map to a 3-D isometric display, using 3-axis arrow displays to visualize anemometer data, combining all tabs into one display while reducing clutter by utilizing minimizable sidebars, and having capabilities to add new sensors and sensor types to the network via the GUI. Drake made mockups of several different GUI iterations and assessed the advantages and drawbacks of each one, making graphics to exemplify shortcomings of different designs. He also identified areas of improvement that the optimal design does not yet address, such as how a user would specify the rotation of an anemometer, how exactly a user would initialize new sensors added to the network, and how one would define new sensor types. Beyond the GUI, Drake made substantial contributions to all write-ups in the semester and participated in many collaborative brainstorming sessions with team members to design solutions for project tasks, such as how the firmware would accommodate new sensor types.

During the spring semester, Drake made substantial progress on the GUI. He added support for selecting motes in the network, added plots for the anemometer data types, refactored many outdated coding design patterns, and successfully plotted sensor data sent from a mote. He also began implementing support for plot axis auto-scaling, and has made investigations and progress on many standing bugs, such as the GUI data lag.
9 CONCLUSION

It is important for the health of ISS crew members that air quality is understood throughout the station. Measuring both the levels of CO$_2$ and the distribution of this gas is essential to gain this holistic understanding. This project aimed to achieve this goal by expanding on previous years’ work. In no particular order, this first entailed redesigning the firmware and hardware of the sensor modules to support a variety of sensors. Previously, only the PTLS was supported in hardware and firmware. This project refactored the firmware to support a variety of data frames from different sensors. The electronics schematics were redesigned to accommodate this as well. Second, the existing GUI was updated to support custom data frames that store sensor data. Last year’s system lacked the capability to display real-world data from the anemometers or CO$_2$ sensors. Since it was also only able to display data from one sensor at a time, functionality was added to allow multiple sensor modules to be supported at once. Third, this project designed a housing and mount for each sensor module to facilitate better reproduction of the placement of modules in the ISS within NIU’s facilities. Fluid dynamics simulations were attempted to try to validate the test results received from the sensors and ABS, though issues with the simulations crashing prevented the team from obtaining results before the end of the project. Finally, this project began the creation of an automated breathing simulator device, which will facilitate test automation and more consistent tests for future teams once it is finished. Initial prototypes of the ABS were designed to only output CO$_2$-enriched air at a constant rate to simulate human exhalation, but future prototypes may increase the similarity to human breath by introducing controls such as variable cadence, volume, temperature, and humidity, as well as inhalation flow and O$_2$ consumption. No hard budget cap was specified by JPL. Last year’s senior design project spent a total of $17,721.99, and this year’s team spent a total of $612.39.
10 REFERENCES


11 ACKNOWLEDGEMENTS

For their funding and expertise NASA JPL should be acknowledged, especially Dr. Lance Christensen and Alexander Hart who worked closely with the senior design team.

Dr. Lance Christensen of JPL consistently offered expertise and knowledge which supported the team in achieving their goals for this project in addition to working with Northern Illinois University throughout the past few years to develop this project.

This team would also like to thank the faculty at Northern Illinois University for their help and support, especially Dr. Donald Peterson and the team’s faculty advisor Dr. Benedito Fonseca, and for facilitating the opportunity to work on this project.

This team would also like to thank Dr. Jifu Tan and Phillip Tomich for their help with working with Chris English to help him better understand how to use the COMSOL software to be able analyze the particle distribution of the CO2.
12 APPENDIX

12.1 Updated Specifications

XBEE S2C ROUTER

- Electrical
  - Supply Voltage: 2.1-3.3V
  - Transmit Current (3.3 V): 45 mA
  - Receive Current (3.3 V): 31 mA

- Wireless
  - Protocol: ZigBee, 802.15.4
  - Modulation: Orthogonal Quadrature Phase Shift Keying (OQPSK)
  - Configuration: Mesh, Point-to-Point, Point-to-multipoint, peer-to-peer
  - Operating band: ISM 2.4 -2.5 GHz
  - Indoor range: 60 m
  - Receiver sensitivity: -102 dBm
  - Transmit power: 6.3 mW

- Software
  - XCTU Digi software

- Protocol
  - UART

JPL TUNABLE LASER SPECTROMETER (TLS) Sensor

- Electrical
  - Sampling rate: 10 Hz

- Environmental
  - Measures: CO2, CH4, H2O
  - Range: 0 –10000 ppb

JPL TUNABLE LASER SPECTROMETER (TLS) MICROCONTROLLER

- Hardware
  - Base station
• **Software**
  - Language: C

• **Protocols**
  - UART

### BASE STATION

• **Software**
  - Interface: Qt

• **Protocols**: USB, UART

• **Language**: C++

### 12.2 Purchase Requisitions and Price Quotes

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part No.</th>
<th>Description of Item</th>
<th>Specifications (color, size, etc)</th>
<th>Qty</th>
<th>Unit $</th>
<th>Total $</th>
<th>Hyper Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>McMastercarr</td>
<td>8975K432</td>
<td>Absorption sheet for the gas canister to sit upon</td>
<td>1'x3&quot;x1/4&quot; 6061 AL</td>
<td>1</td>
<td>17.86</td>
<td>17.86</td>
<td><a href="https://www.mcmaster.com/8975K432-8975K434/">https://www.mcmaster.com/8975K432-8975K434/</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>48925K113</td>
<td>a pipe for motive placement</td>
<td>Standard-Wall Unraveled Rigid PVC Pipe for Water 1/8 Pipe Size, 10 Feet Long</td>
<td>1</td>
<td>15.20</td>
<td>15.20</td>
<td><a href="https://www.mcmaster.com/48925K113/">https://www.mcmaster.com/48925K113/</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>48925K42</td>
<td>a pipe for motive placement</td>
<td>Standard-Wall Unraveled Rigid PVC Pipe for Water 3/8 Pipe Size, 10 Feet Long</td>
<td>1</td>
<td>26.00</td>
<td>26.00</td>
<td><a href="https://www.mcmaster.com/48925K42/">https://www.mcmaster.com/48925K42/</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>47045T29</td>
<td>metal frame part with flanges to support metal plate</td>
<td>Press-Fit Framing Fitting, Standard Rail, Single 5</td>
<td>1</td>
<td>5.88</td>
<td>5.88</td>
<td><a href="https://www.mcmaster.com/47045T29-47045T1">https://www.mcmaster.com/47045T29-47045T1</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>47045T28</td>
<td>long part of metal frame</td>
<td>Press-Fit Framing Fitting, Standard Rail, Square, 1&quot; x 1&quot; Wide 1&quot; long</td>
<td>4</td>
<td>5.67</td>
<td>22.68</td>
<td><a href="https://www.mcmaster.com/47045T28-47045T5">https://www.mcmaster.com/47045T28-47045T5</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>47045T28</td>
<td>the base of the metal frame</td>
<td>Press-Fit Framing Fitting, Standard Rail, Square, 1&quot; x 1&quot; Wide 3&quot; long</td>
<td>1</td>
<td>14.00</td>
<td>14.00</td>
<td><a href="https://www.mcmaster.com/47045T28-47045T5">https://www.mcmaster.com/47045T28-47045T5</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>47045T82</td>
<td>the corners of the metal frame</td>
<td>Black Plastic Standard Three-Way Elbow Connector for Press-Fit Framing Fitting</td>
<td>8</td>
<td>6.87</td>
<td>54.96</td>
<td><a href="https://www.mcmaster.com/47045T82/">https://www.mcmaster.com/47045T82/</a></td>
</tr>
</tbody>
</table>

Please ENTER shipping cost on all items, if single vendor, in cell to the right. If multiple shipping costs from multiple vendors, please list each shipping charge on a separate line above.

Total cost of all items. Please confirm NO SALES TAX will be charged. Purchase total: 155.58

---

**Table 5: Detailed 1st Order From**

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part No.</th>
<th>Description of Item</th>
<th>Specifications (color, size, etc)</th>
<th>Qty</th>
<th>Unit $</th>
<th>Total $</th>
<th>Hyper Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>McMastercarr</td>
<td>BB8K2</td>
<td>Super-Cushioning Polyethylene Foam</td>
<td>3&quot; Diameter, 1/2&quot; Long</td>
<td>1</td>
<td>4.73</td>
<td>4.73</td>
<td><a href="https://www.mcmaster.com/BB8K2/">https://www.mcmaster.com/BB8K2/</a></td>
</tr>
<tr>
<td>McMastercarr</td>
<td>5811T1</td>
<td>Panel Mount High-Pressure Regulating</td>
<td>Air and inert Gas, 1/4 NPT Female</td>
<td>1</td>
<td>362.12</td>
<td>362.12</td>
<td><a href="https://www.mcmaster.com/5811T1/">https://www.mcmaster.com/5811T1/</a></td>
</tr>
<tr>
<td>Hutchins</td>
<td>895414047164</td>
<td>14g of PLA that the MakerBot wishes to use</td>
<td>Gray, PLA Filament, 1.75 MM, 1 KG Spool</td>
<td>1</td>
<td>24.99</td>
<td>24.99</td>
<td><a href="https://www.hatchbox.com/products/14-g-pla-gray-1-75-mm-1-kg-spool">https://www.hatchbox.com/products/14-g-pla-gray-1-75-mm-1-kg-spool</a></td>
</tr>
</tbody>
</table>

Please ENTER shipping cost on all items, if single vendor, in cell to the right. If multiple shipping costs from multiple vendors, please list each shipping charge on a separate line above.

Total cost of all items. Please confirm NO SALES TAX will be charged. Purchase total: 455.81

---

**Table 6: Detailed 2nd Order Form**

---

Page 80 of 80