University of Colorado Boulder
Department of Aerospace Engineering Sciences
ASEN 4028

Project Final Report (PFR)
May 6, 2024

Controllable Open-source Water Sensing (COWS)

Information

Project Customer

Name:	Khosro Ghobadi-Far
Email:	khosro.ghobadifar@colorado.edu

Team Members

Name: Janelle Nedrow Position: Project Manager Email: sane5550@colorado.edu	Name: Ethan Stapp Position: Systems Engineer Email: etst4571@colorado.edu
Name: Micah Bergeron Position: Mechanical Lead Email: mibe5777@colorado.edu	Name: Sean Vestecka Position: Software Lead Email: seve8485@colorado.edu
Name: Gerrell Miller Position: Financial Lead Email: gerrell.miller@colorado.edu	Name: Blake Wilson Position: Electrical Lead Email: blwi7501@colorado.edu
Name: Hideyuki Nakanishi Position: Electrical and Software Email: hina5754@colorado.edu	Name: Toan Lu Position: Mechanical and Software Email: tolu9219@colorado.edu
Name: Kristyn Grell Position: Software Email: krgr9395@colorado.edu	

Contents

List of Figures

List of Tables

Nomenclature

A: Area of a Solar Panel

Comms: Communications

CONOPS: Concept of Operations

DOD: Depth of Discharge

DR: Derived Requirement

e: Elevation Angle

FR: Functional Requirement

G: Solar Irradiance

GLONASS: Глобальная навигационная спутниковая система (RU), Global Navigation Satellite System (EN)

GNSS: Global Navigation Satellite System

GPS: Global Positioning System

I: Current

I/O: Input/Output

IR: Interferometric Reflectometry

I^2C: Inter-Integrated Circuit

MicroSD: Micro Secure Digital

P: Power

PC: Power Capacity

PCB: Printed Circuit Board

RF: Radio Frequency

RSD: Remote Sensing Device

SLA: Sealed Lead Acid

SNR: Signal to Noise Ratio

UART: Universal Asynchronous Receiver/Transmitter

V: Voltage

1 Project Purpose and Design

Authors: Ethan Stapp, Sean Vestecka, Kristyn Grell, Janelle Nedrow, Blake Wilson, Toan Lu, Micah Bergeron

1.1 Purpose

With climate change and global warming becoming an ever-present issue across the world in recent years, the need for accurate water level monitoring systems is increasing yearly as sea levels rise and fresh water scarcity become more prevalent issues. Many remote sensing devices are already deployed across the world for monitoring weather. However, these devices are often too expensive for many communities across the world to afford, and they can contain expensive or proprietary software. With the current worldwide availability and coverage of GNSS satellites, and the knowledge of the GNSS-Interferometric Reflectometry (GNSS-IR) technique, this provides a unique opportunity to develop a low-cost and open source GNSS-IR water level monitoring system that can be deployed in remote locations across the world and accurately measure changes in water level in near real-time.

The mission objective for this project is to develop a low-cost, open source, near real-time water level monitoring system with high temporal resolution that can be deployed in remote locations across the world and be operable in varying weather conditions. The low-cost nature of this project is made achievable through the GNSS-IR technique. This technique allows for the use of low-precision GNSS receivers that have a lower cost. With this project being both low-cost (less than $800) and open-source, this project will be able to be built and used by anyone around the world for water level or snow level monitoring. The CONOPS for the GNSS-IR system is included below in Figure 1.0.1.

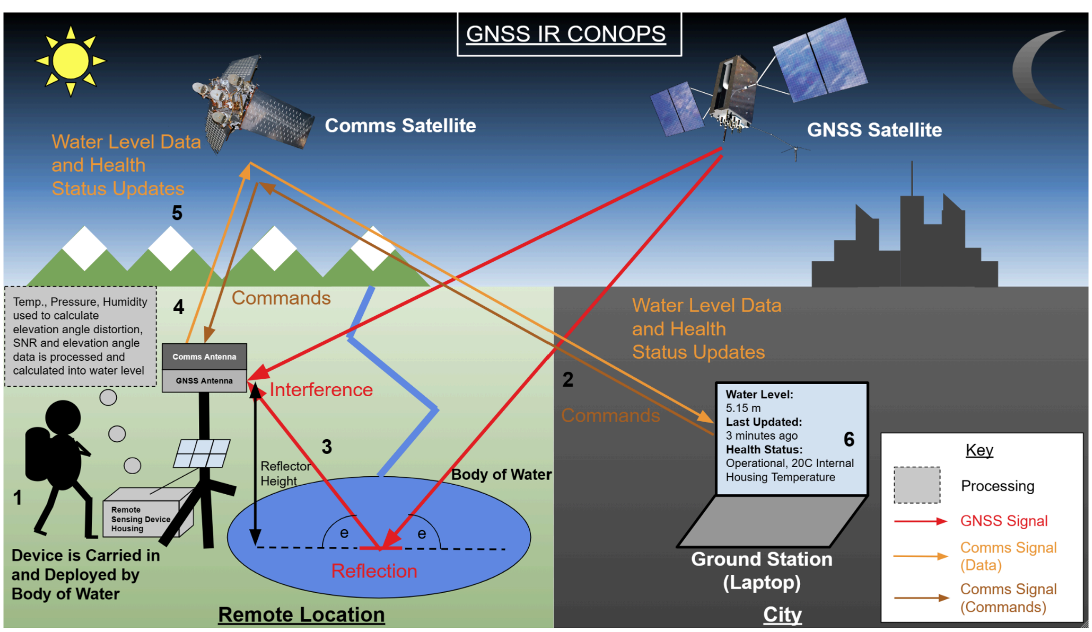


Figure 1.0.1 GNSS-IR CONOPS

The first step to setting up the remote sensing device is backpacking in the GNSS-IR device and deploying it next to a body of water. Second, system configuration commands will be sent from the ground control station to the remote sensing device to specify minimum and maximum azimuth angles, in addition to elevation angles. Third, the remote sensing device will receive the direct and reflected signals through the GNSS antenna and then process them into SNR and elevation angle for further processing. Fourth, the SNR and elevation angle data will be processed on-site through the microprocessor into water level data. After this, onboard temperature, pressure, and humidity sensors will be used to calculate tropospheric elevation angle distortion effects. Fifth, the water level data and health status updates (including internal housing temperature and the operational status of the device) will be sent through the comms module to the comms antenna, over the Iridium satellite network, to the ground control station. Sixth, the ground control station will display the water level data, time when that data was taken, and the health status of the device on a screen for the user. Moreover, data processing option commands (e.g. specifying average vs. instantaneous water height) and system reboot commands (resetting the GNSS receiver and rebooting the comms module) can also be sent by the user.

The GNSS-IR Technique involves using interferometric reflectometry on the signals produced by GNSS constellations in order to perform measurements. The basics of the technique are shown in the following figure. Signals reflected off of a surface intersect signals that directly reach an antenna. The interference between this reflected signal and the direct line-of-sight signal manifests in the SNR of the signal collected by the antenna. This is demonstrated below in Figure 1.0.2.

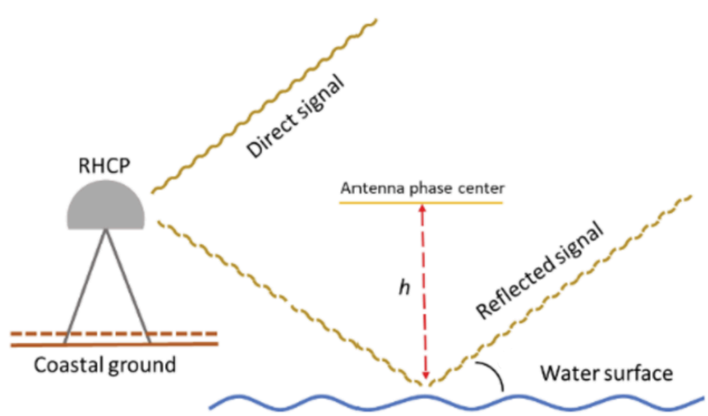


Figure 1.0.2 GNSS-IR Reflection Demonstration Image[3]

Looking at the equation for the sinusoid signal and the equation for the signal-to-noise ratio present in the GNSS signal collected by the antenna:

$$x(t) = A\cos(2\pi f t + \theta)$$

$$\delta SNR(t) = A\cos(\frac{4\pi h}{\lambda} \sin[e(t)] + \theta)$$

These equations have a relation in their frequency component that can be used to calculate the height of the reflected signal. By setting them equal to each other, this can solved down to get the following relation for reflector height based on the frequency of the SNR signal:

$$2\pi f = \frac{4\pi h}{\lambda}$$

$$f = \frac{2h}{\lambda}$$

$$h = \frac{\lambda}{2} f$$

The derivation of this relationship is from Larson et al. (2013)[4]. With this relationship, knowing the frequency of the SNR signal and the wavelength of the GNSS carrier waves will return the height of the reflected signal. For the purposes of this project, this reflector height can be used to measure the surface height of bodies of water.

1.2 Detailed Design

1.2.1 High Level Functional Requirements

Included below in Table 1.2.1 is all of the functional requirements for the GNSS-IR system. All of these requirements flow down into the system level, subsystem level, and component level requirements which are not included here.

FR 1.0	The system shall cost less than $800 per unit, excluding monthly communication subscription costs.
FR 2.0	The system shall operate independently in remote environments in a temperature range of -30 to 55 C (-22 to 131 F).
FR 3.0	The system shall be able to receive signals from multiple GNSS constellations (GPS, GLONASS, Galileo, and BeiDou) as well as various frequencies (e.g., L1 and L5 for GPS).
FR 4.0	The system shall be able to receive and respond to commands from a ground control station regarding system configuration, system reboot, etc.
FR 5.0	The system shall be able to receive and respond to commands from a ground control station regarding data processing options (e.g., compute daily average of estimated water level data).
FR 6.0	The system shall send health status updates to the ground station.
FR 7.0	The system shall retrieve sub-daily water level with a precision better than 5 cm.
FR 8.0	The system shall be able to estimate water level variations within 2 hours latency.
FR 9.0	The system shall be able to transmit the SNR data and/or final results (water level variation) to a ground control station (e.g., a laptop) to be displayed live on a screen.

Table 1.2.1. Functional Requirements

1.2.2 Functional Block Diagram

A detailed functional block diagram (FBD) of the final GNSS-IR system is included below in Figure 1.2.2. The FBD includes each of the major components of the system arranged into their respective subsystems and locations. Components are highlighted in blue, subsystems are outlined in dashed gray lines, and locations are outlined in black dashed lines. The FBD also shows the connections between components to the pin level and highlights the signals sent between components, with data signals in black, 3.3V power in light green, 5-18V power in dark green, ground in pink, comms signals in orange, and GNSS signals in red. The system's functionality and subsystems are explained below the figure.

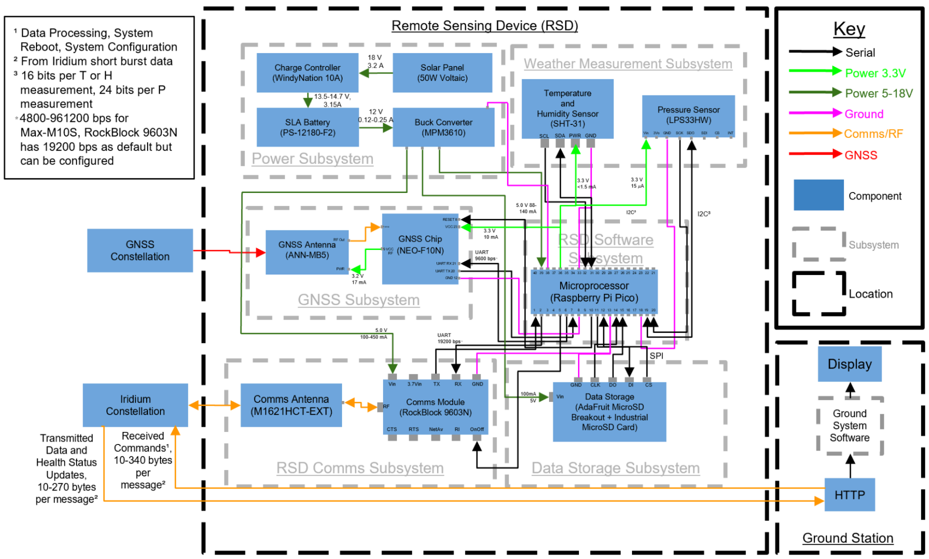


Figure 1.2.2 System Functional Block Diagram

Note: Unless otherwise mentioned, all components are rated within the -30 °C to 55 °C temperature range

The Power Subsystem consists of a solar panel, charge controller, battery, and buck converter. As shown in the FBD, the 50W Voltaic Systems solar panel supplies 18V and 3.2A to the WindyNation 10A charge controller. This charge controller then takes the power supplied by the solar panels and uses it to charge the sealed lead-acid (SLA) PS12100-F2 battery between 13.5-14.7V and 3.15A. The battery then supplies 12V and 0.12-0.25A to the MPM3610 buck converter, which converts the 12V from the battery to 5V for the RockBlock 9603N comms module, the Raspberry Pi Pico microcontroller, and the AdaFruit MicroSD Breakout data storage.

The GNSS Subsystem interfaces with GNSS constellations to read their signals and deliver the raw signal data to the microprocessor. Its two components are the GNSS Chip and the GNSS Antenna. The selected GNSS chip is the ublox NEO-F10N. Its purpose is to read the incoming RF signals from the antenna and parse GNSS signals, delivering the result of its parsing in the form of NMEA messages. The chip communicates with a bi-directional UART serial connection with a baud rate of 9600 bps with the microcontroller. The chip delivers the latest NMEA messages to the microprocessor over this connection. It also receives 3.3V power from the microprocessor's 3.3V rails. Additionally, the chip delivers the required 3.2V power to the antenna for its internal amplification. The selected GNSS antenna is the ublox ANN-MB5. This antenna receives GNSS signals and sends them to the GNSS chip.

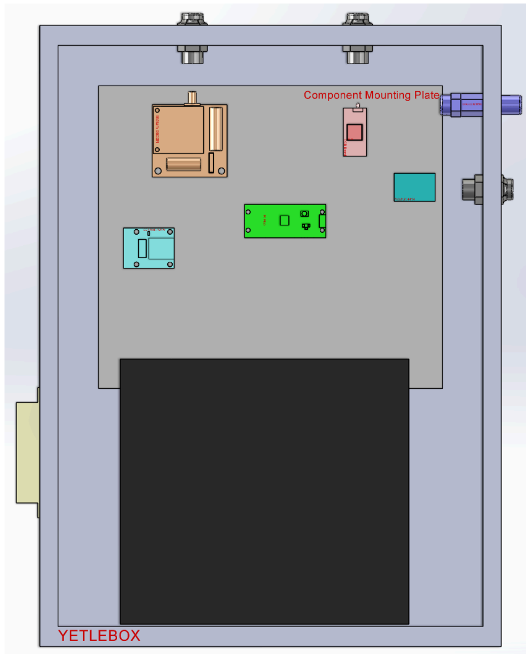
The Comms Subsystem interfaces with the Iridium satellite communications network. As shown in the FBD, its primary components are the communications antenna and the communications module. The selected communications module, the RockBlock 9603N, is based on an Iridium 9603 module. It acts as receiver and transmitter for messages to and from the network, sending messages from the microprocessor and receiving messages from the ground station. It has a bi-directional UART serial connection with a baud rate of 19200 bps with the microcontroller. It transmits and receives messages over its RF port, which connects to the communications antenna. It also takes in 5V power directly from the buck converter. The selected communications antenna is the Maxtena M1621GCT-EXT. This helical antenna is designed to interact with the Iridium satellite network. It connects to the communications module, delivering it the signals it receives.

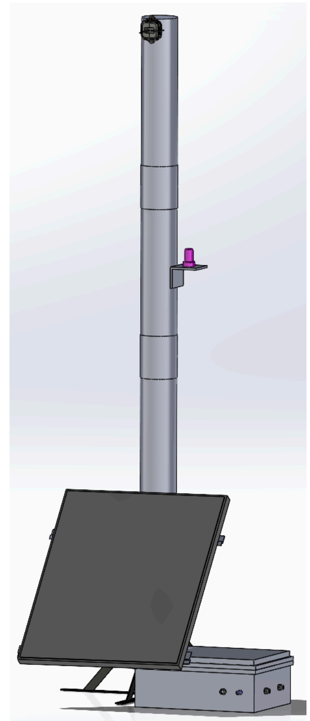
The Data Storage Subsystem interfaces with the Remote Sensing Device Software Subsystem, where the Data Storage Subsystem stores the calculated water level height in case the Remote Sensing Device Software Subsystem is unable to communicate with the Ground Station Software Subsystem. The selected Micro SD card is the Kingston Industrial Micro SD card. Its purpose is to store the calculated water level data and interface with the selected Micro SD card reader, Adafruit MicroSD Card Breakout. The Micro SD card reader interfaces with the microprocessor through I^2C.

The Remote Sensing Device Software Subsystem consists of the Raspberry Pi Pico microcontroller, as shown in the FBD. This subsystem is responsible for processing SNR, elevation angle, and weather data (air pressure, air temperature, and relative humidity) into water level and sending water level data to the Comms Subsystem for transmission. This subsystem is also responsible for processing system reboot, system configuration, and data processing commands. Because of this required functionality, the subsystem interfaces with the GNSS Subsystem through UART to receive SNR and elevation angle data, interfaces with the Comms Subsystem through UART for sending data and receiving commands, interfaces with the Weather Measurement Subsystem through I^2C for receiving temperature, pressure, and humidity data for tropospheric distortion effects, and interfaces with the power subsystem which powers the microcontroller through the buck converter.

Finally, the Ground Station Software Subsystem consists of the website (HTTP), the ground station software, and the display. The Ground Station Software Subsystem interfaces with the Comms Subsystem through the Iridium constellation to transmit system reboot, system configuration, and data processing option commands to the remote sensing device. It also receives water level data and health status updates from the remote sensing device. The Ground

Station Software Subsystem provides a user interface for the purpose of sending commands to the remote sensing device and viewing near real-time water level data and health status updates.





(a) (b)

Figure 1.2.3: (a) Final design of the component box, contains a layout of components within the component box. (b) Final design of the antenna pole containing both the modular design of the antenna pole, placements of antennas, solar panels, and location of the component box.

Figure 1.2.3 depicts the final design of the component box and the modular design of the antenna pole. The overall dimensions of the component housing box is 14.6” by 10.63” by 5.9” while the dimensions of the antenna pole was 78.7” by 23.72” by 20.28”. The component box is waterproof for the safety and integrity of the components inside from various weather conditions. The component box contains cable glands that allow the cables for the communication antenna, GNSS antenna, and silicon tube for the pressure sensor while making it waterproof.

The antenna pole was made of PVC poles that were 0.5 m increments that allowed the antenna to be modular for the purpose of one design for any body of water size. The PVC poles are interconnected via PVC couplings to ensure the modular design, while ensuring secure structural integrity. The solar panels are attached to the bottom of the antenna pole which is secured by using hose clamps. The GNSS antenna was placed at the top of the antenna pole to ensure the optimal coverage for any body of water. The communication antenna was placed at a max height of 1.25 m due to the length of the cable, but it was still able to communicate to and from the Iridium satellites due to PVC being transparent to RF. Finally, the component box is placed under the solar panel to regulate the temperature to ensure that the temperature inside the component box does not exceed the battery charging and discharging ranges.

2 Testing, Verification, and Validation

Authors: Hideyuki Nakanishi, Ethan Stapp, Janelle Nedrow, Sean Vestecka, Kristyn Grell, Gerrell Miller, Blake Wilson, Toan Lu

Included below in Table 2.0.1 is all models and prototype tests completed. The Power Model is described in more detail below in Section 2.1.

Model or Prototype Test	Description	Requirements Verified
Power Model	Using MATLAB, the model looked into power consumption, solar panel generation, battery life, and battery recharge time.	FR 2.0, DR 2.1, DR 2.3
Wind Beam Deflection Model	Using MATLAB, the model calculates the deflection of the tip of the antenna pole due to forces from and the mounted solar panel for a variety of PVC pipe sizes.	FR 3.0, DR 2.1, DR 7.2.1
Thermal Model	Using MATLAB, the model calculates the radiation produced by the components within the housing. The model was also used to verify that the temperature within the housing is still within the temperature requirements.	DR 2.1, DR 7.2.1
GNSS-IR Prototype	Using a SparkFun breakout board for the NEO-F10N and the Raspberry Pi 4, test data was taken to verify the system would meet the 5 cm accuracy requirement without use of an array configuration. The data taken had an error of 1.5 cm.	FR 2.0, DR 2.1, FR 7.0, DR 7.2

Table 2.0.1 Models and Prototype Tests Completed

Key tests that were performed from Table 2.0.2 are T1, T3, T4, T7, and T8. Test 1 focused on evaluating the Raspberry Pi Pico's performance in extreme cold temperatures of -30 °C (-22 °F). It required a cooler, thermocouple, and dry ice to create the necessary temperature conditions. The goal was to verify the system's functionality under varying thermal stresses, addressing requirement FR 2.0 which states that “The system shall operate independently in remote environments in a temperature range of -30 to 55 °C (-22 to 131 °F)” . A significant safety consideration in this test was the handling of dry ice and the management of extremely low temperatures. Proper safety gear, like insulated gloves and temperature-controlled environments, was essential to mitigate these risks.

Test 3 consisted of the Raspberry Pi Pico, along with a RockBlock 9603N, M1621-hct, and a laptop, which was used to assess the system's responsiveness to commands from a ground control station. This includes testing for system configuration, data processing options, and executing reboot commands. This test is critical for ensuring reliable communication and command execution under operational conditions, fulfilling requirements FR 4.0 which states that “The system shall be able to receive and respond to commands from a ground control station regarding system configuration, system reboot, etc.” and FR 5.0 which states that “The system shall be able to receive and respond to commands from a ground control station regarding data processing options (e.g., compute daily average of estimated water level data)”.

The objective of test 4 was to test the system's ability to transmit SNR data and water level variation results live to a ground control station display. It involved the use of a comprehensive set of equipment including the Raspberry Pi Pico, RockBlock 9603N, M1621-hct, and a laptop. This test is crucial for verifying real-time data transmission capabilities, addressing requirement FR 9.0 which states that “The system shall be able to transmit the SNR data and/or final results (water level variation) to a ground control station (e.g., a laptop) to be displayed live on a screen”.

Test 7 involved measuring battery life and solar charge time of the battery to verify the system’s power system is sufficient for the RSD to remain operational off its own power, per requirement FR 2.0. This test consisted of the RSD (including the battery, solar panel, and charge controller) and a timer to monitor battery discharge time and solar charge time.

Test 8 involved deploying the system near the Boulder Reservoir to verify its operability over an extended period of two weeks. The test setup included a variety of equipment like the Raspberry Pi Pico, RockBlock 9603N, M1621-hct, YETLEBOX, PS-12180-F2 battery, Renogy 50W solar panel, and a laptop. It's designed to ensure system stability and operational reliability in a real-world environment, meeting requirement FR 2.0. Because this test was a day in the life test of the RSD and ground control station, this test also further verifies requirements FR 7.0, FR 9.0, and FR 8.0. The primary safety risks in this test include environmental exposure and equipment failure. To mitigate these, it was necessary to ensure waterproof housing for the equipment and have contingency plans for equipment malfunctions or data loss. Should the equipment malfunction or lose data, the regular health status updates will be able to notify the ground station and will be able to automatically shut off.

Test ID	**Description**	**Equipment/Facilities**	**Requirement ID**
T1	Conduct an operational test on the Raspberry Pi Pico in a controlled environment to ensure functionality at cold temperatures (-30 °C).	Raspberry Pi Pico, Cooler, Thermocouple, Dry ice	FR 2.0
T2	Conduct a Multi-Constellation Signal Report Test to verify the system’s capability to receive and process signals from GPS, GLONASS, Galileo, and BeiDou using L1 carrier frequencies.	NEO-F10N, ANN-MB5, Laptop	FR 3.0
T3	Execute a Commands Test to assess the system's ability to receive and execute instructions from a ground control station, including system configuration, data processing options, and reboot commands.	Raspberry Pi Pico, RockBlock 9603N, M1621-hct, Laptop	FR 4.0 & FR 5.0

T4	Execute a Real Time Transmission Test to verify the system's capability in sending consistent and accurate health status updates to the ground station. Verify the system's proficiency in estimating water level variations within two hours latency and displaying final results to a ground control station to be displayed live on a screen.	Raspberry Pi Pico, RockBlock 9603N, M1621-hct, NEO-F10N, ANN-MB5, Laptop	FR 6.0, FR 8.0, FR 9.0
T5	Execute an Antenna Height Test to confirm the system's ability to retrieve sub-daily water levels with a precision better than 5 cm.	NEO-F10N, ANN-MB5, Laptop	FR 7.0
T6	Conduct a Water Resistance Test to evaluate the systems durability when subjected to direct water exposure.	YETLEBOX, Shower	FR 2.0
T7	Conduct a Power Test that will test the charging and discharging of the battery to verify battery life and charge time.	WindyNation 10A Charge Controller, Renogy 50 W Solar Panel, PS-12180-F2 SLA Battery	FR 2.0
T8	Conduct a deployment test near a body of water to verify system operability for 2 weeks.	Boulder Reservoir, RSD, Laptop	FR 2.0

Table 2.0.2 Major Tests Completed

Our two most critical tests, the two week deployment test and power test, are described in detail below.

2.1 Power Test and CDR Power Model Verification

2.1.1 Battery Life

Battery life is another important factor when considering the size of a battery and evaluating whether the system is able to operate for 3 days without solar power generation, per requirement DR 2.3.1. With the RSD needing to be operable in varying weather conditions, a battery life of 3 days should provide flexibility for overcast days or storms where solar power generation is limited. The selected battery type is lead-acid as it has a large range of allowable charging temperatures. In order to calculate battery life, the ampere-hour of the battery given by the datasheet was divided by the current drawn from the battery, assuming all the components are turned on at all times.

The current drawn from the battery can be expressed as below:

$$\eta_{Buck\ Converter} = \frac{P_{out}}{P_{in}} \cdot 100 \quad (16)$$

Where $\eta_{buck\ converter}$ is the efficiency of a buck converter that is given by the datasheet.
P_{out} is the output power from a buck converter in Watts.
P_{in} is the input power to a buck converter, which is equivalent to the given output power of a battery in Watts.

$$\eta_{Buck\ Converter} = \frac{V_{out} \cdot I_{out}}{V_{Battery} \cdot I_{Battery}} \cdot 100 \quad (17)$$

Solving for unknown $I_{Battery}$,

$$I_{Battery} = \frac{\eta_{Buck\ Converter}}{100} \cdot \frac{V_{out} \cdot I_{out}}{V_{Battery}} \ \text{[A]} \quad (18)$$

Note V_{out}, I_{out}, and $V_{Battery}$ are given by the datasheets.

$$Battery\ Life = \frac{AmpHour_{Battery}}{I_{Battery}} \ \text{[hr]} \quad (19)$$

With the analysis of battery life, it is essential to analyze the worst and optimal scenarios. For this reason, the minimum and maximum current draw from the components were defined based on the datasheets, which determined the range of battery life without power generation as shown in the table below.

In the optimal condition, the battery is able to supply power for more than 3 days without generating power, whereas in the worst case scenario of using the maximum current draw at all times, the remote sensing device can function for 1.7 days. The worst case scenario was assumed not to happen since the Comms Subsystem, which draws the most current, does not transmit and receive data at all times. These results are summarized below in Table 2.1.1.

Battery Life [days]	1.7 ~ 3.7

Table 2.1.1 Battery Life without Power Generation

2.1.2 Battery Recharge Time

Because the remote sensing device needs to operate in remote environments at varying latitudes and weather conditions, per FR 2.0, another important consideration for the remote sensing device was battery recharge time. When calculating the battery recharge time for the remote sensing device, the final recharge time is reported in peak sun hours, which is defined as an hour of 1000 W/m^2 or more of solar irradiance. The equations for this model are included below. First, the discharged battery capacity was calculated in Whr, which is the amount of energy in Whr that needs to be recharged by the solar panels. To calculate this, the battery capacity of 216 Whr was multiplied by the depth of discharge (DOD) of 50%. A depth of discharge of 50% was used since the battery charge needs to be kept between 50% and 100% charged for preserving the battery health of the sealed lead acid (SLA) battery.

$$Discharged\ Battery\ Capacity\ =\ (Battery\ Capacity)\cdot DOD\ \ [Whr] \tag{20}$$

After calculating the discharged battery capacity, the discharged battery capacity was divided by the efficiency of lead acid batteries, 85%, to calculate the total required energy for recharging the battery from the solar panel in Whr.

$$Energy\ Required\ for\ Full\ Charge\ =\ \frac{Discharged\ Battery\ Capacity}{Lead\ Acid\ Efficiency}\ \ [Whr] \tag{21}$$

$$Adjusted\ Solar\ Output\ =\ W_{Solar}\cdot(PWM\ Efficiency)\ \ [W] \tag{22}$$

$$Charge\ Time\ =\ \frac{Energy\ Required\ for\ Full\ Charge}{Adjusted\ Solar\ Output}\ \ [hr] \tag{23}$$

The adjusted solar output was then calculated by taking the wattage of the solar panel and multiplying the wattage of the solar panel by the PWM efficiency of the charge controller of 85%. This gave an adjusted solar output from the solar panel in watts. To calculate charge time in peak sun hours, the energy required for full charge in Whr was divided by the adjusted solar output in watts. This gave a battery recharge time of 3.38 peak sun hours. This nearly meets the requirement for the solar panels recharging the batteries in no more than 3 peak sun hours, DR 2.3.2. This is a result of the increase in battery capacity made after TRR.

There is, of course, a tradeoff between battery capacity and charge time. In deployments near the poles, where sunlight is limited, clients would likely opt for the larger battery. However, here in Boulder the smaller 10.5 Ah battery would absolutely be sufficient. In this case, the battery can be charged from 50% DOD in 1.97 peak sun hours.

Throughout the power model analyses, there are several uncertainties that need to be considered. The first, and most important of which is varying current draw from all components over time. This difference caused the aforementioned power model to largely overshoot the actual system results for battery life. Furthermore, due to solar irradiance variation with altitude, the solar power can be increased at higher altitude. Varying temperature would affect the battery performance. For this reason, the battery life would be more uncertain. Lastly, there may be manufacturing imperfections; currents and voltages on the system may vary from values on datasheets.

2.1.3 Test Setup

To get the most accurate estimate of battery life, we performed the battery discharge test on the second RSD. During this test, the primary system was deployed at Boulder Reservoir for the two week deployment test (described below). Battery life was measured by monitoring the battery capacity LEDs on the solar charge controller of the second RSD. The second RSD was fully operational throughout the battery discharge test, computing reflector heights and transmitting data and health status updates. The battery used in this test was the PS-12180-F2 SLA battery, which has 18 Ah battery capacity and a voltage of 12V. The battery discharge test consisted of timing the time it took for the battery to discharge from 100% capacity to 50% capacity (i.e. from when the battery capacity LED on the charge controller went from solid green to solid yellow).

2.1.4 Power Test Results (With PS-12180-F2 Battery)

The battery capacity LED on the charge controller remained solid green for 5.5 days, after which we had to conclude the battery life test due to our Iridium subscription expiring on the RockBlock 9603N module on our second system. Due to us building two RSD's, we didn't have sufficient budget to renew our Iridium subscription for the second RSD used for the power test. Table 2.1.3 shows the predicted results versus the testing results for the power tests.

Parameter	Predicted Result	Result from Testing
Battery Life [days]	1.7 - 3.7	Greater Than 5.5 Days
Solar Recharge Time [hrs]	3.28	Not Tested

Table 2.1.3. Power System Testing Results

The battery discharge test proved that the battery capacity of the PS-12180-F2 SLA battery was greater than 5.5 days and more than sufficient to provide 3 days of battery life without power generation, meeting requirement DR 2.3.1. Because the battery didn't discharge past 50% depth of discharge in time during the battery discharge test, we were unable to complete the solar recharge test in time for this report, which would have consisted of timing how many peak sun hours were required for the battery to charge from 50% capacity to 100% capacity from the solar panel. Due to the overperformance of the PS-12180-F2 SLA battery, we are confident that a smaller battery capacity would be sufficient to meet the 3 day battery life requirement, such as the PS-12100-F2 SLA battery with a 10.5 Ah battery capacity and voltage of 12V (compared to the 18 Ah and 12V of the PS-12180-F2 SLA battery).

2.2 Two Week Deployment Test

The two week deployment test occurred in Boulder Reservoir from April 1st to 15th, in order to verify the whole RSD functions for a long period of time at a remote location without any software, electrical, and mechanical failures between the system and the ground control station, per requirement FR 2.0. This test was further extended to conclude on May 3rd. During this test, the RSD was placed at Boulder Reservoir for an entire two week period from April 1st to 15th, and remained functional until May 3rd when it was time to retrieve the system from Boulder Reservoir. This critical test also verified the accuracy of predicting change in water level within 5 cm, verified continuous data transmission, and verified the system’s latency in transmitting water level measurements. As such, this test further verifies requirements FR 7.0, FR 9.0, and FR 8.0 respectively.

2.2.1 Test Setup

The proper test setup is essential for making the RSD system work and collect proper data. Since the device processes reflected GNSS signals from the water surface, it is necessary to configure appropriate elevation and azimuth angles, which can be determined from Dr. Larson’s zone mapping website, as shown in Figure 2.2.1. From the zone mapping, the elevation and azimuth angle masks were set to be 5 - 25 and 67 - 235 degrees, respectively for this test.

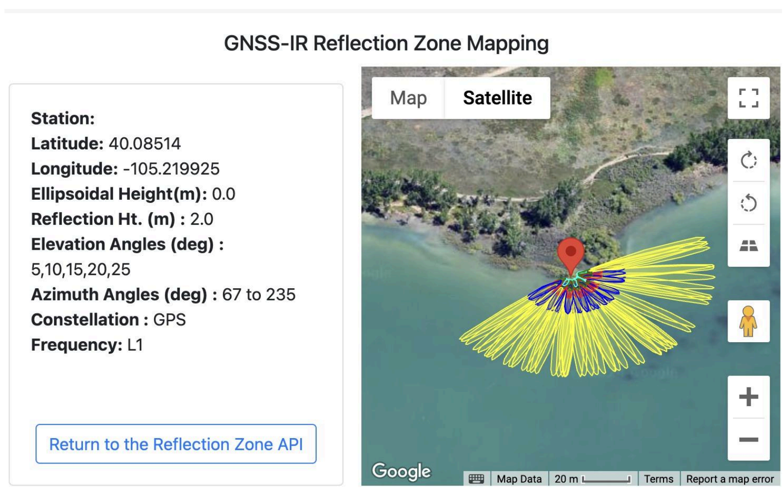


Figure 2.2.1 GNSS Reflection Zone Mapping

Along with configuring elevation and azimuth angles, the GNSS antenna was set perpendicular to the water surface and placed 2m high relative to the water surface to collect a large range of elevation angles up to 30 degrees. More importantly, the GNSS antenna was positioned south-facing towards Boulder Reservoir as most GNSS satellites come from the south in the northern hemisphere. Additionally, the solar panel was positioned South at an angle of 45 degrees relative to the ground, as shown in Figure 2.2.2. The required solar panel angle and solar panel direction would change depending on latitude; the angle and position of the solar panel is configurable for our system with the mounting brackets used.

Figure 2.2.2 RSD Setup on Site

The Iridium communications antenna was positioned on the antenna pole that had a clear line of sight to the sky for improved receiving and transmission of signals to and from Iridium satellites. Additionally, the temperature and humidity sensor and a pressure sensor were included with the system to correct for tropospheric distortion effects of the elevation angle of GNSS signals. These sensors were mounted such that they were not obstructed or covered so they could measure accurate data. Lastly, the electrical box that stored all the electrical components except for antennas and the solar panel was positioned beneath the solar panel in the shade to prevent overheating inside the electrical box from direct sunlight. The different sensors used to perform this testing consists of the onboard measurement sensors on the actual device, as they are all collectively required to collect, process, and transmit the data.

To measure the GNSS signals reflected off of the surface, the system used the NEO-F10N GNSS receiver, manufactured by Ublox. This was paired with an ANN-MB5 antenna, also manufactured by Ublox. To perform the aforementioned tropospheric distortion corrections, pressure, temperature, and humidity were measured by the device. The Sensirion SHT-31 is a temperature and humidity sensor, and the Adafruit LPS33HW is the pressure sensor. The RockBLOCK 9603N is the satellite communications module, and is a transceiver that was used to interface with the Iridium satellite communications network. This transceiver was paired with an external Maxtena M1621HCT-EXT Iridium antenna. The system was powered by a 50W Renogy solar panel paired with a PowerSonic PS-12180-F2 12V SLA battery. Due to manufacturer shortages, we were unable to use the planned 50W Voltaic Systems solar panel. However, the Renogy 50W solar panel was comparable in voltage and wattage. All of the components were mounted onto our structure or to the inside of the electrical box as needed. The GNSS configuration for this testing was an azimuth mask of 67-235 degrees, an elevation angle mask of 5-15 degrees, and an expected reflector height between 1 and 3 meters.

2.2.2 Real Time Transmission and Latency

One of the main objectives of this test was to verify the system's ability to transmit water level measurements within 2 hours latency, thus satisfying FR 8.0. In order to verify this, each water level measurement sent to the ground station also contained the time at which the data was collected. This time data was then compared to the time at which the message was received on the RockBlock server. Of all the data points taken, none had latencies longer than 2 hours. The testing results have been summarized below in Table 2.2.2.

Water Level Message Time	Latency (hr:min:sec)
Minimum	00:30:08
Maximum	1:00:44
Average	00:44:40

Table 2.2.2 Data Transmission Latency

2.2.3 Water Level Accuracy

The primary objective of our 2 week deployment test was to verify that our system is able to measure the height between the GNSS antenna and a body of water with an accuracy of at least 5 cm, verifying FR 7.0. In order to verify this test we obtained truth data from Northern Water, which is reported to 1/100th of a foot, and compared their data with the data our system obtained. In Figure 2.2.3 below, collected data from April 14, 2024 to April 17, 2024 is plotted versus truth data, excluding any outliers that are greater than 40 cm. This figure uses the daily average of the raw data collected to account for any inaccuracies from real time GNSS-IR implementation. As shown in Figure 2.2.3, we were able to achieve a daily average within 5 cm accuracy.

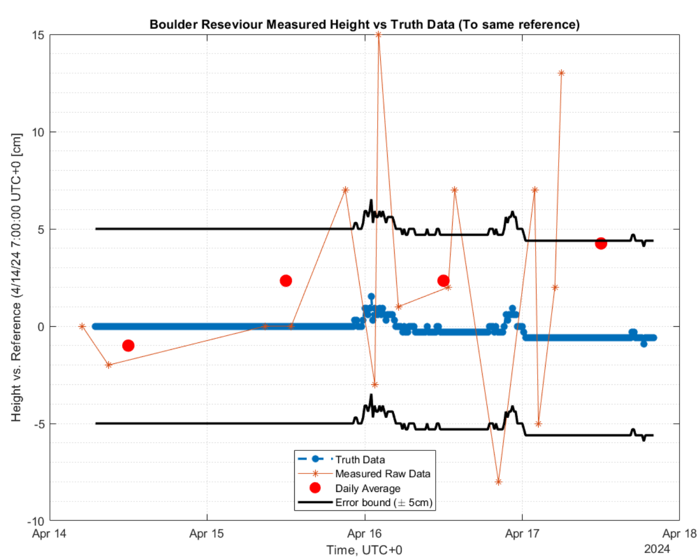


Figure 2.2.3 Water Level Accuracy Results

2.2.4 Functionality of Mechanical Subsystem

The main purpose of the mechanical subsystem functionality test was to confirm the survivability of the pole and the housing in any weather conditions such as wind, snow, and rainstorms. During the deployment test, snow and rainstorms hit Boulder Reservoir several times. However, the water did not soak into the system, and the device was working properly and sending consistent health status updates every hour without any issues. When the system was retrieved from deployment, it was confirmed that no water entered the component housing. This verified that the system was indeed waterproof. Moreover, windstorms of 35 - 40 m/s over two days on April 6th and 7th occurred during the deployment period. The deflection of the top of the pole stayed within a tolerable range, which we can tell because the pole did not break and we did not get wildly inaccurate water level measurements. This verified the beam deflection model as shown in Figure 2.2.4. The system also did not receive any damage as a result of the weather conditions and remained attached to the ground for the entirety of the deployment period. As a result, all the mechanical components met the functional requirements.

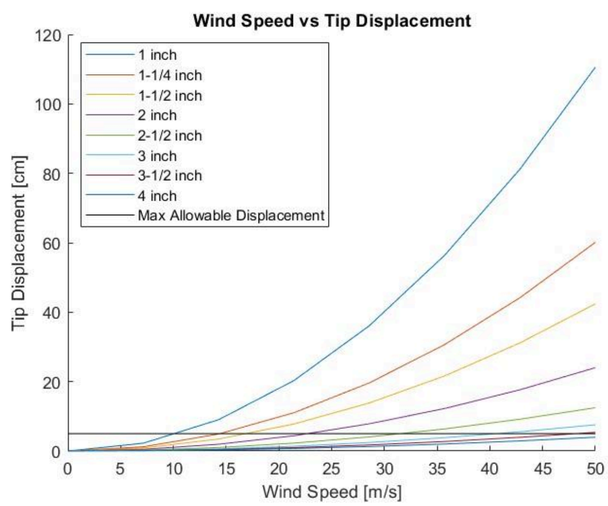


Figure 2.2.4 Pole Displacement due to Wind

3 Lessons Learned and Future Work

Authors: Janelle Nedrow, Gerrell Miller, Sean Vestecka, Toan Lu, Micah Bergeron

3.1 Key Lessons Learned

1. **Integration of Open-Source Components**: This project demonstrated the complexities involved in integrating multiple open-source hardware and software components. Handling compatibility and ensuring system stability with open-source elements are crucial for achieving functional coherence.
2. **Environmental Resilience**: The deployment emphasized the need for robust system designs that can withstand environmental stresses such as extreme weather conditions.

Designing for durability and reliability in harsh environments is essential for successful operation of remote sensing devices.

3. **Real-Time Data Transmission**: The project emphasized the importance of reliable and efficient data communication systems. Establishing a stable connection and ensuring timely data transmission from remote locations via satellite networks is critical for operational success and accuracy.
4. **Memory Allocation**: Working with embedded systems like the Raspberry Pi Pico revealed the challenges of managing limited memory resources effectively. Future projects need to prioritize efficient memory management by selecting a stronger micro-controller to optimize performance and reliability, particularly when processing and storing real-time environmental data.
5. **Component Compatibility**: Building a mechanical system with parts from a variety of suppliers and manufacturers highlighted the importance of attention to detail when purchasing components that must fit together.

3.2 Schedule

Below is the Gantt chart outlining the schedule to be followed in the spring. This includes all of the procurement, integration, manufacturing, testing and class deliverables to be completed.

The Critical Path is lined in maroon and outlines the important tasks that need to be done in order for the whole project to be successfully completed. The Critical Path starts with all of our procurement and then moves along the integration blocks, because we aren’t able to build our system and perform our tests without all of our components and without software written. It then moves to the Test Readiness Review, as this is required by the course for safety and competency reasons prior to testing. The path then follows all the tests and the last two deliverables, being the AES Symposium and the Spring Final report.

The margin for any given task was made based on the beginning date of whatever task is dependent on it. For example, both of our softwares should only take 26 days, but we don’t need them for another 19 days when we perform the multi-constellation test on March 4th.

The two pink stars are representative of the two assignments that are in the middle of testing and manufacturing. These are just to help visualize that the assignments lie in the midst of everything else whereas the other two assignments are at the end of everything and do not interrupt the flow of everything.

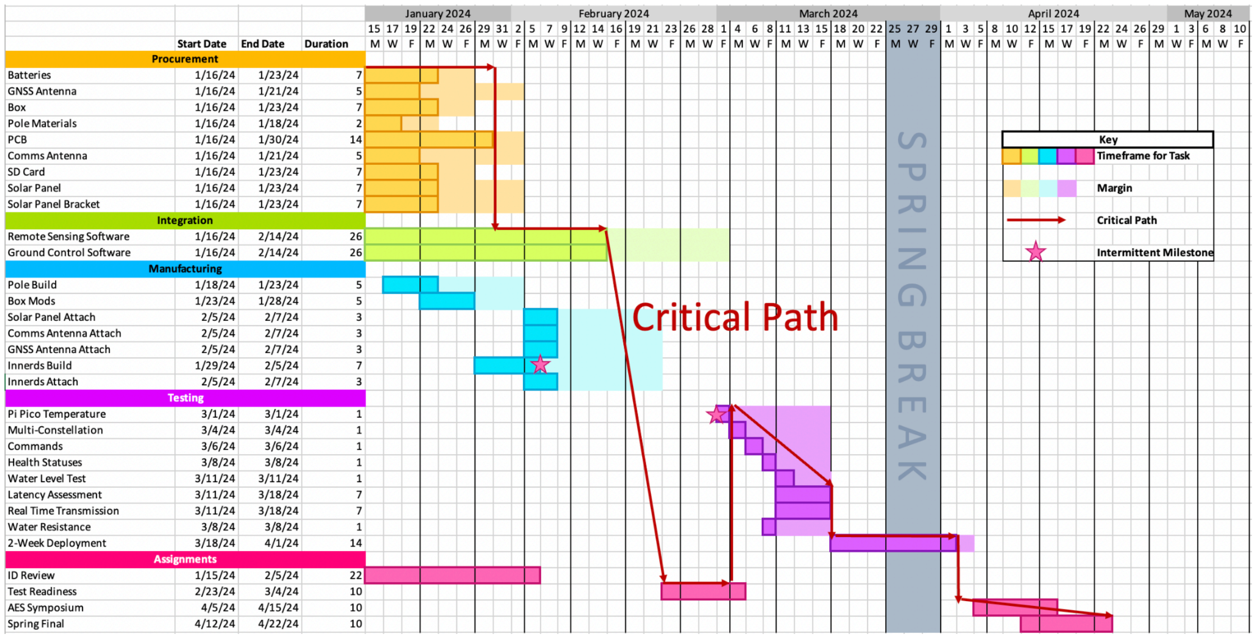


Figure 4.2.1 Predicted Gantt Chart

In reality, Figure 4.2.2 displays the actual timeline the team took to complete the project. As seen in Figure 4.2.2, there were major deviations from the predicted gantt chart in Figure 4.2.1. These deviations were due to unexpected difficulties with the software development for both the remote sensing device and ground station, the component manufacturing and attachment process, and finally testing. The deviations were mostly due to creating a remote sensing software that would work on a limited ram microcontroller like the pi pico. Due to this unforeseen difficulty, the software development took longer than expected. This caused testing to be delayed because most of the testing required the software to be functional for tests like the commands test, antenna height test, real time transmission, power test, and the two week deployment test. The other deviation was in attaching components to the inside of the electrical box, which was delayed due to unexpected variabilities in the delivery of some components.

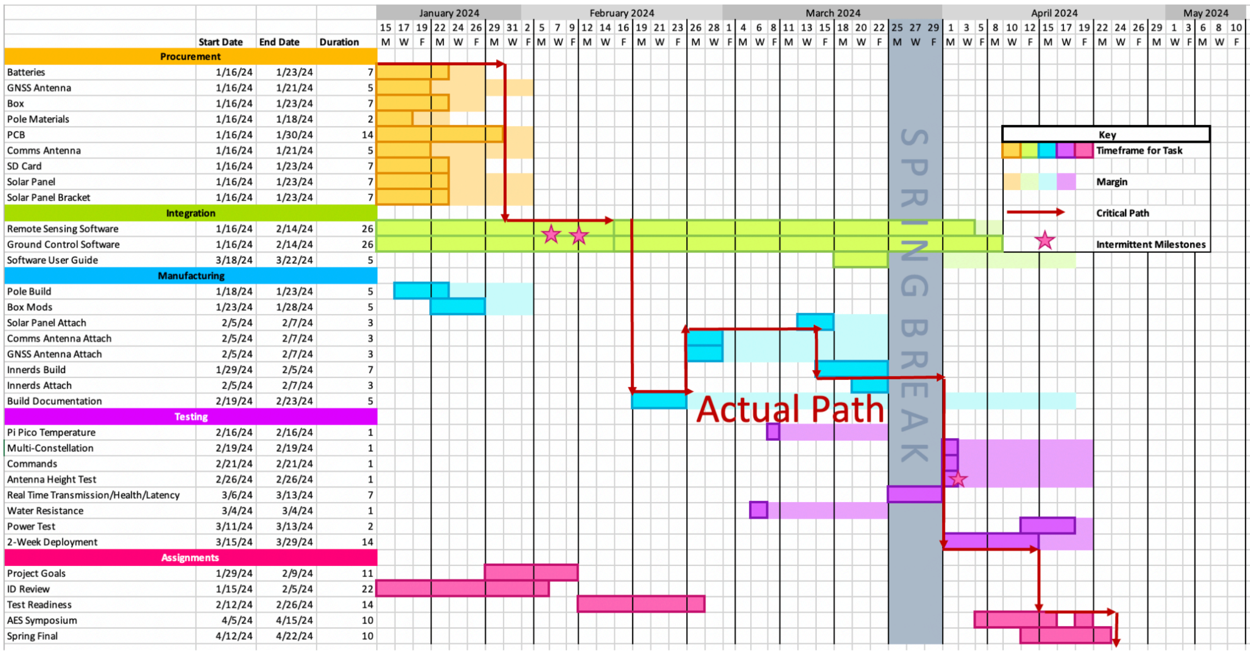


Figure 4.2.2 Actual Gantt Chart

3.3 Future Work

1. **Power System**: The final power system we used is oversized to a ridiculous degree. A solar panel with half the wattage and a battery with half the capacity would have been adequate. Our power system was considered for the worst-case scenario at a much higher latitude than our test with much less hours spent in the sun. We did perform our testing in April, with plenty of sunlight hours and the time of year being beyond the spring equinox, so the power system was given the best chance possible. Another test in a more harsh environment, like a winter at a greater latitude, could verify the power system's capability for a more extreme environment. We suspect that this testing would result in a variety of configurations for different environments, i.e. the smallest needed near the equator with larger sizing requirements in more extreme latitudes.
2. **Solar Charging Testing**: We are already aware from our power system testing that the power system, both for battery capacity and solar recharging, are oversized for our use. We were never able to properly discharge the battery to an adequate level to do a proper solar recharge test. We are aware that the current configuration is adequate, but as mentioned in the previous future work section, there is work to be done to make more sensible configurations that are properly sized for both budgetary and space limitations of altered configurations.
3. **Documentation**: One of the goals of our project was to make our system readily available to build and modify to the general public. An important thing left to do is produce better documentation for our component build and selection. A guide to the assembly, including the reasoning for certain mechanical and component decisions,

would greatly increase the accessibility of the project to less technical users still interested in using our work. This document would also be paired with the following other important section,

4. **Open-Sourcing**: One of the goals of our project was to make our system readily available to build and modify to the general public. The source code we created for the system, along with forks made of certain other open-source (Primarily Peter Hinch's micropython-async project, available on an MIT license), should be made easily accessible with proper documentation. We plan to work in our free time in the summer to create this documentation and eventually release it on an MIT license to allow any user to modify the project. A placeholder repository on Github has already been created at https://github.com/SeanVSquared/COWS, but the work to document and publish the work needs to be done. The code is also not 100% functional and more features and debugging will likely be needed.
5. **PCB Design & Printing**: A future version of this project that has less internal wiring requirements may be helpful. As of now, the project is entirely hand-wired together, with many different components placed on breakout boards for making a minimum viable product. Future iterations could be integrated into a single PCB, with all of the wiring handled in layers of fiberglass instead of bundles of wires. This may also reduce the price of the unit, as there is no need to buy breakout boards from various vendors when we create a monolithic PCB with the components more directly from a manufacturer. One exception to this may be the communications module, whose modem and satellite comms setup would be challenging to replicate for a team of our means, along with creating a PCB to handle RF.

4 Individual Report Contributions

Name	Contributions
Ethan Stapp	Section 1: Updated project background, described FBD, subsystems, and Conops. Section 2: Described major tests, worked on the power test section.
Janelle Nedrow	Section 1: Contributed to Purpose and background Section 2: Contributed to Power Section and 2-Week Deployment Test Section 3: Contributed to the Schedule Section and the Key Lessons Learned Report: Edited every section after they were written
Toan Lu	Section 1: Contributed in describing the final design of the system. Section 2: Contributed to power test section, and reviewed section. Section 3: Contributed to Schedule segment.
Hideyuki Nakanishi	Section 2: Contributed to describing the test set and test validation, and test results
Gerrell Miller	Contributed to preamble and overall formatting. Section 2: Contributed to key tests descriptions. Section 3: Contributed to key lessons learned.
Micah Bergeron	Section 2: Edited/contributed to mechanical subsystem functionality section. Section 3: Contributed to Key Lessons Learned segment
Kristyn Grell	Section 1: Contributed to description and requirements sections Section 2: Edited/contributed to power and 2 week deployment results
Sean Vestecka	Section 2: Added details and figures for major tests, 2 week test results Section 3: Future Work Segment
Blake Wilson	Section 1: Provided description for subsystems Section 2: Added details to major tests, two-week test results

5 Self-Assessment of Individual Project Goals

Team Name: Cows		
Student Name	**Team Role(s)**	**Goals (100 words or less)**
		Original Goals: I will continue to lead meetings, update schedules, pickup parts, interact with Boulder Reservoir for testing and overseeing all operations. I have been focusing my efforts on assisting Micah with modifying our mechanical subsystem components. I have been checking in with the other subteams to make sure that we are on track with our software development and

		parts testing. These goals will be considered successful when our box is assembled and waterproof and we are able to perform testing at Boulder Reservoir. These goals will be fulfilled when requirements DR 0.6.1, DR 0.1.1.2, DR 0.5.5.1, and DR 1.1 are met.
Janelle Nedrow	Project Manager	Self Assessment:I successfully led all of the meetings, made sure the group kept up with timing for all of the assignments and deadlines, and communicated with all necessary parties for testing. Our box and pole were built and withstood the harsh elements during our 2-week deployment. When we got behind on software, I helped our software lead make a task list and task schedule, so that other members could pick up tasks as others were finished. I joined the software team with no python experience and did everything I could to learn the language and code programs necessary for deployment.
		Original Goals: I will continue to support manufacturing, working closely with mechanical lead Micah Bergeron and Toan Lu. Additionally, I will be supporting Sean, our software lead, with software development when extra assistance is required. I will also support Blake, our Test Lead, with planning and executing our tests to verify our requirements. This also includes assisting with any smaller prerequisite tests that need to be conducted (i.e. doing individual component testing to verify component functionality).
Ethan Stapp	Systems Engineer	Self Assessment: I accomplished all of my goals for this semester. I worked closely with Blake on planning and executing all required tests and also assisted with all required prerequisite tests. Additionally, I helped when needed with software development and contributed to manufacturing by helping out Toan and Micah when required.
		Original Goals: I will oversee the manufacturing of our mounting pole and our box modifications. This includes personally attaching the base brackets to the bottom piece of the mounting pole as well as drilling the holes into the box for all necessary cable glands. I will also participate in mounting the solar panel to the bottom piece of the mounting pole. These goals will be considered successful when requirements DR 0.5.3.1, DR 0.5.3.3, DR 0.5.3.2, DR 0.5.3, and DR 0.6.1 are met. I will be working closely with Toan, Janelle, and Ethan.

Micah Bergeron	Mechanical Lead	Self Assessment: I was able to meet all of the goals by myself above. I was also able to complete the manufacturing process for our secondary system in a very short amount of time, which allowed us to perform concurrent tests and troubleshoot any issues that occurred with our deployed system in real time. Overall I was able to complete housing modifications for two component boxes, build two antenna poles, and attach two solar panels. In the future I will also be working on developing documentation for the build process so the system can be replicated. I was able to achieve the goals set above by working closely with Ethan, Janelle, and Toan.
Gerrell Miller	Financial Lead	Original Goals: As the financial lead in our senior project team, I will continue crafting a project-specific budget, conducting monthly financial reviews, and managing procurement to ensure cost-efficiency. I will assist with the mechanical team as well as the electrical team in the manufacturing and integration process. Success will be marked by adherence to our budget, demonstrating effective cost management, and securing resources within project specifications. I'll collaborate closely with Janelle and Ethan as well as our department leads, ensuring our financial strategies align with project milestones and quality standards.
		Self Assessment: As the financial lead, I've effectively managed our project's budget, conducting monthly financial reviews and overseeing procurement to ensure cost efficiency. My collaboration with the electrical team supported our manufacturing and integration processes. Through adherence to budget constraints and proactive resource management, I achieved our goals of maintaining cost effectiveness and aligning financial strategies with project milestones, working closely with Janelle, Ethan, and department leads.
		Original Goals: I will continue the development of the modular pole and box modifications. This includes fabricating the pole, attachment of the GNSS and Comms Antenna, as well as helping with any box modifications and component layout within the box. I will also participate in mounting the solar panels at an optimal height onto the modular pole. These goals will be accomplished if requirements DR 0.5.3.1, DR 0.5.3.3, DR 0.5.3.2, DR 0.5.3, and DR 0.6.1 are met. I will be closely working with Micah, Ethan, and Janelle.

Toan Lu	Mechanical/Manufacturing and Software Team Member	Self Assessment: I have achieved the goals that were set for this semester. I worked closely with Micah in order to fabricate the antenna pole. I also worked closely with Micah, Ethan, Sean, and Blake on complete integration of components within the component box.
Kristyn Grell	Software Team Member	Original Goals: I will be working with the software team to develop software for the ground station and remote sensing device systems. The main goals of this software are to take data from the GNSS chip and sensors, calculate water level, send this data to the ground station, and display it. There are many requirements for these subsystems, but I would argue the most important are FR 4.0, FR 9.0, DR 7.2.1, and DR 7.1.1. I will be working closely with Sean, Blake, and Hide. As software and electronics are interdependent, I will be available to assist with any electronics issues.
		Self Assessment: In the end, most of my software work was relegated to interfacing with the Iridium module and pressure sensor. Both of these components ended up working as expected. I also assisted Hide with debugging his code, and elevation angle correction. I will assist Sean in any way necessary for publishing an open source version of our code.
Blake Wilson	Electrical Lead/ Test Lead/ Software Team Member	Original Goals: As the electrical/test lead I will be working on integration between components and subsystems of the remote sensing device and planning and conducting tests on different subsystems and integration. Most, if not all, Component requirements are important for this to be successful. I will also be helping with software when needed. I will be working closely with Sean, Ethan, Hide, and Kristyn
		Self Assessment: The ground station was electronically integrated successfully and operated as expected working closely with Hideyuki and Toan. All tests were performed and executed successfully and were able to verify all requirements working closely with Ethan. Also succeeded in helping with software by writing code for the humidity/temp sensor and internal temp sensor. Thus I have completed my goals.

Sean Vestecka	Software Lead	Original Goals: I will continue work on the software for both the RSD and the ground station. I will work closely with the rest of the software team, including Kristyn, Blake, and Hide, along with our SE Ethan. I will oversee the multiple software unit tests that lead up to the larger tests in the test plan, ensuring each component of the software works together before integrating them. I will specifically learn lots about asynchronous execution for the RSD and bring my web dev skills for the ground station. These goals will be considered successful when DR 8.1, DR 9.1, and DR 7.1 are met.
		Self Assessment: The ground station was completed and is entirely functional. The RSD is mostly complete, with some work to be done on features and debugging. I did learn lots of new skills for the RSD primarily and the software team members successfully made many different functionalities with a reasonable level of oversight and review from my leadership role. The 3 requirements listed were all met through our two-week test, succeeding in my original goals.
Hideyuki Nakanishi	Electrical/ Software Team Member	Original Goals: I will be working on the software development that includes decoding NMEA messages and data processing, as well as working on the electrical integration of components and subsystems of the remote sensing device. Gaining a deep understanding of GNSS embedded systems, I will be developing reliable software with our lead Sean, Blake and Kristyn.
		Self Assessment: Data processing was completed with accuracy of better than 5cm. I contributed to the software subsystem and debugging the software, testing, analyzing the data for the semester, and maximizing the software system. I did learn a lot of new skills for the RSD and maximization process. The personal and project goals were all achieved.