L-Band Satellite Tracking and Characterization System

Detailed Design Document

Authors:

- Charles Kente Van Horn Jr. (Team Manager) - RF and Microwave Engineering (EE)
- Samuel Walker Taliaferro VIII Velasquez - Machine Learning (CPE)
- Justin Daigle- Communications (EE)
- Ethan Christopher Duval - Networking and Cyber Security (CPE)
- Jared Lane Jamison - Controls Robotics and Autonomy (EE)

Date: April 28, 2023

Abstract

Satellite Tracking and Characterization System. The system aims to provide efficient and accurate tracking and data collection of L-Band satellites using a Raspberry Pi 4 as the main processing platform. The design consists of several interconnected subsystems, including the computer system, operations software, mount subsystem, enclosure, receiver, power system, and antenna. The document provides a comprehensive overview of each subsystem, its specifications, connections, and implementation. Additionally, the document includes an acceptance test plan to evaluate the system's performance and ensure it meets the design requirements.

Table of contents:

1.1 Computer Introduction 3
1.2 Computer System Specifications 3
1.3 Component Connections 3
1.4 Computer Implementation 4
2.1 Operations Software Introduction 4
2.2.1 User Interface (GUI) 6
2.2.2 Client-Server Communication (CSComm) 6
2.2.3 Server-Client Communication (SCComm) 7
2.2.4 DataManager 8
2.2.5 DataProcessor 9
2.2.6 TLEParser 9
2.2.7 Scheduler 9

2.2.8 RSPDuo Interface 10
2.2.9 Yaesu Rotator Interface 11
2.3.0 Operations Software Implementation 13
2.3.1 Required Libraries 14
2.3.2 Executable Creation 14
2.4.1 Raspberry Pi Linux Service for Bluetooth Discoverability and Slave Side Operations Software 15
2.4.2 Service Configuration 16
2.4.3 Service Python Script 16
3.1 Mount Subsystem 16
3.2.1 Mount System Requirements 17
3.2.2 Mount System Derived Requirements 17
3.3 Mount Schematic 18
3.4 Rotator Design 19
3.5 Additional Mount Components 21
3.5.1 The Computer Interface 21
3.5.2 The Tripod 21
3.6 Enclosure 21
3.6.1 Enclosure Size 21
4.1 Receiver Subsystem 22
4.1.1 Filtering Stage 23
4.1.2 SDR Stage 24
5.1 Power System Introduction 25
5.2 Power system specifications 25
5.2.1 Battery Voltage 26
5.2.2 DC-DC Converters for Voltage Regulation 26
5.2.2.1 Computer power 26
5.2.2.2 Mount Power 27
5.3 Safety 27
5.3.1 Fuses 28
5.3.2 Emergency Cutoff Switch 28
5.4 Wire Gauge 28
6.1 Antenna Subsystem 30
7 Acceptance Test Plan 31
8 Actions Table 32

1.1 Computer Introduction

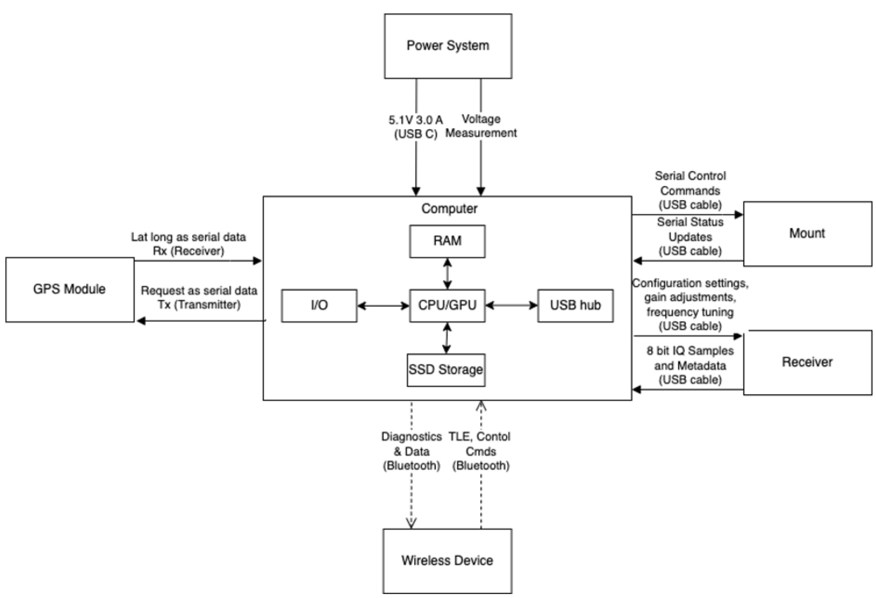
The computer system for the L-Band Satellite Tracking and Characterization System is responsible for providing the necessary processing power, storage, and connectivity to support the operations software and manage the system's hardware components. This document provides a detailed design explanation for the computer system, including the choice of platform, specifications, and connections.

1.2 Computer System Specifications

The chosen computer platform is the Raspberry Pi 4 due to its processing power, built-in connectivity options, and support for multiple programming languages. The specifications for the computer system are as follows:

- Platform: Raspberry Pi 4
- RAM: 4GB
- CPU: 1.5GHz
- USB Ports: 2 USB 2.0 ports (for ADC connectivity and mount control)
- Storage: 128GB Micro SD card
- Wireless Communication: Built-in Bluetooth 5.0
- Additional Features: SD card port, GPS connection, magnetometer, and sufficient memory to record 10 minutes of I/Q data sampled at 10MHz
- Total Cost: Raspberry Pi 4 with GPS module estimated at $105

1.3 Component Connections



- GPS module: Connected to the Raspberry Pi via TX and RX pins, providing time and position information.
- Magnetometer module: Connected to the Raspberry Pi via I2C (SDA and SCL pins), providing magnetic field measurements to determine the heading and orientation of the rotor
- Power: Raspberry Pi is powered via a USB-C connection, supplying 5.1V at 3.0A.
- Mount Control: Raspberry Pi is connected to the antenna rotator via the GS-232B interface, enabling USB control of Yaesu antenna rotators.
- ADC Connectivity: Raspberry Pi is connected to the RSPduo via a USB B to USB A cable for data acquisition and processing.

1.4 Computer Implementation

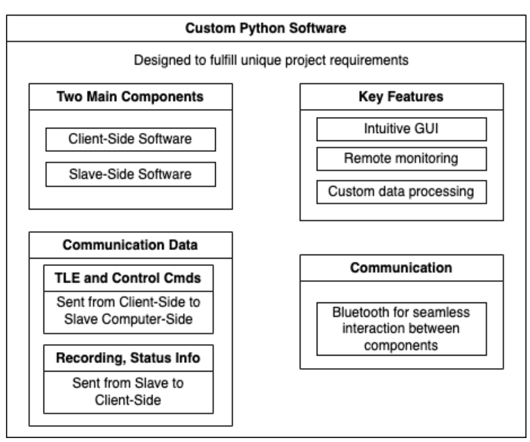
The computer system will run the operations software, which is implemented using Python and PyQt for the GUI. The Raspberry Pi's extensive online resources and ease of use with Python will facilitate the development and integration of the operations software. The GPIO pins on the Raspberry Pi will be used to interface with the GPS module as well as the magnetometer, while USB ports will handle connections to the RSPduo and the antenna rotator.

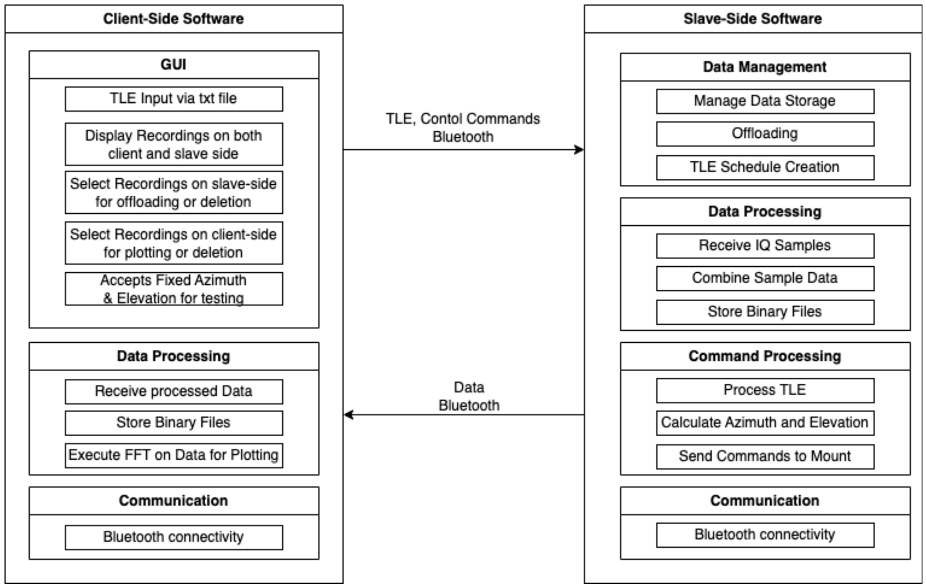
2.1 Operations Software Introduction

The operations software for the L-Band Satellite Tracking and Characterization System is responsible for providing an interface between the user and the system's hardware components. It allows the user to input commands, receive system status updates, and manage recorded data.

The software consists of several modules that communicate with each other to provide a seamless experience for the user.

Software Architecture

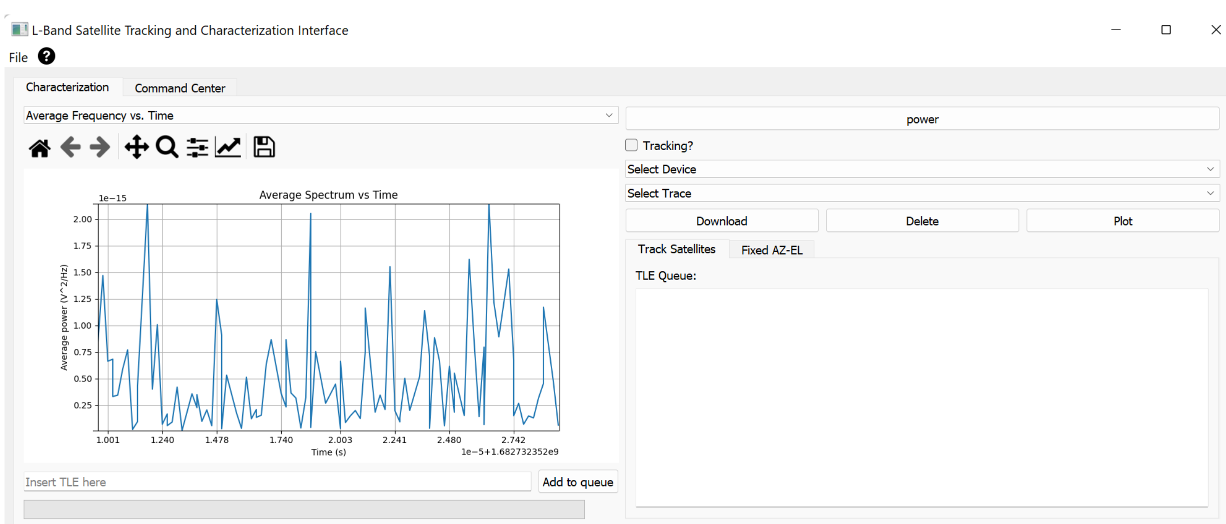




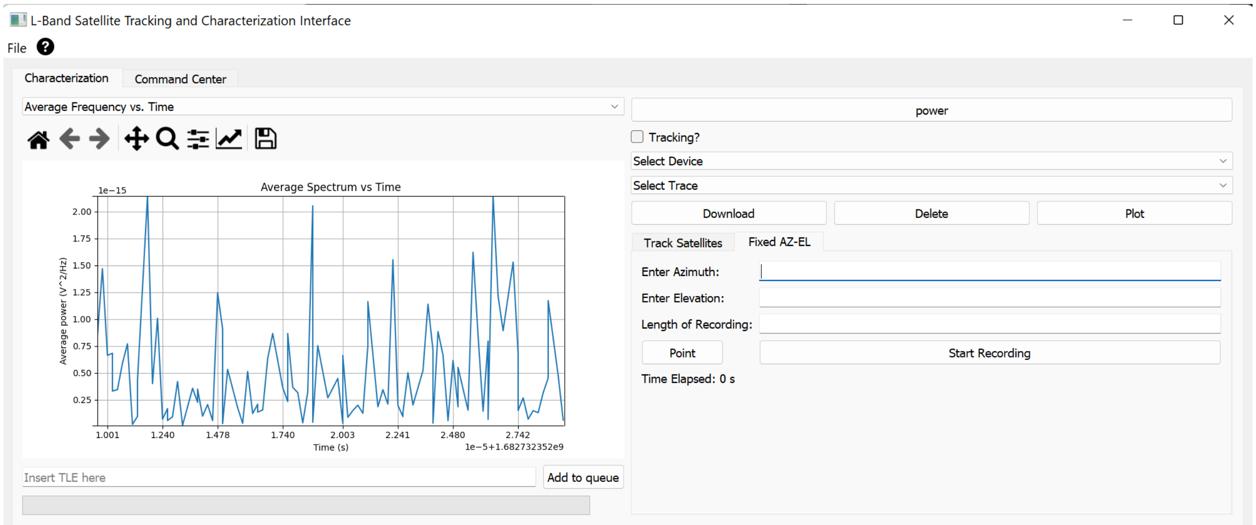
The software architecture consists of the following main components:

User Interface (GUI)
Client-Server Communication (CSComm)
Server-Client Communication (SCComm)
DataManager
DataProcessor
TLEParser
Scheduler
RSPDuo Interface
YaesuRotator Interface

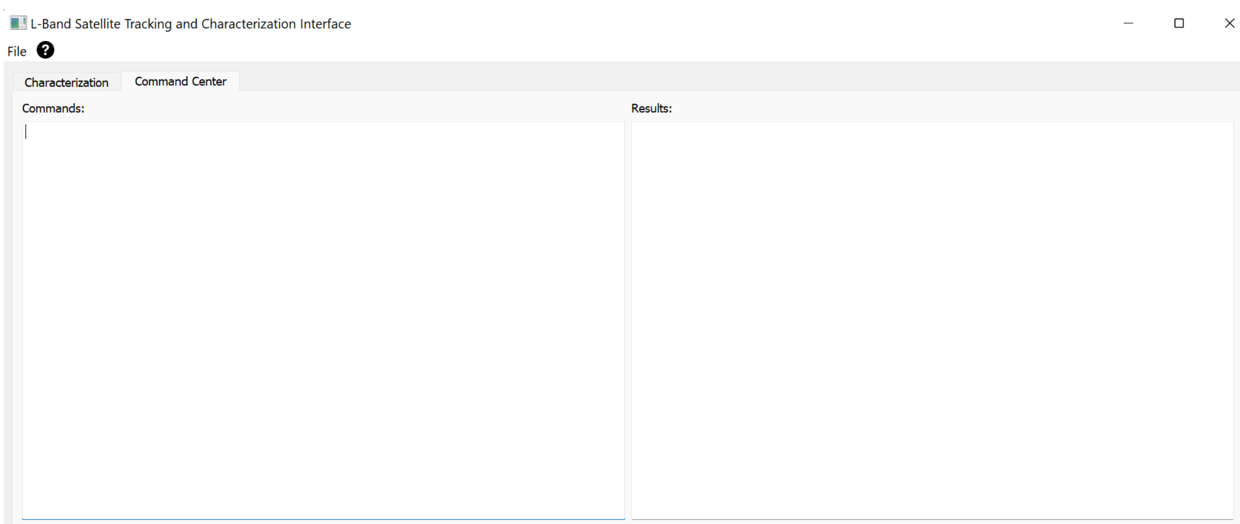
2.2.1 User Interface (GUI)



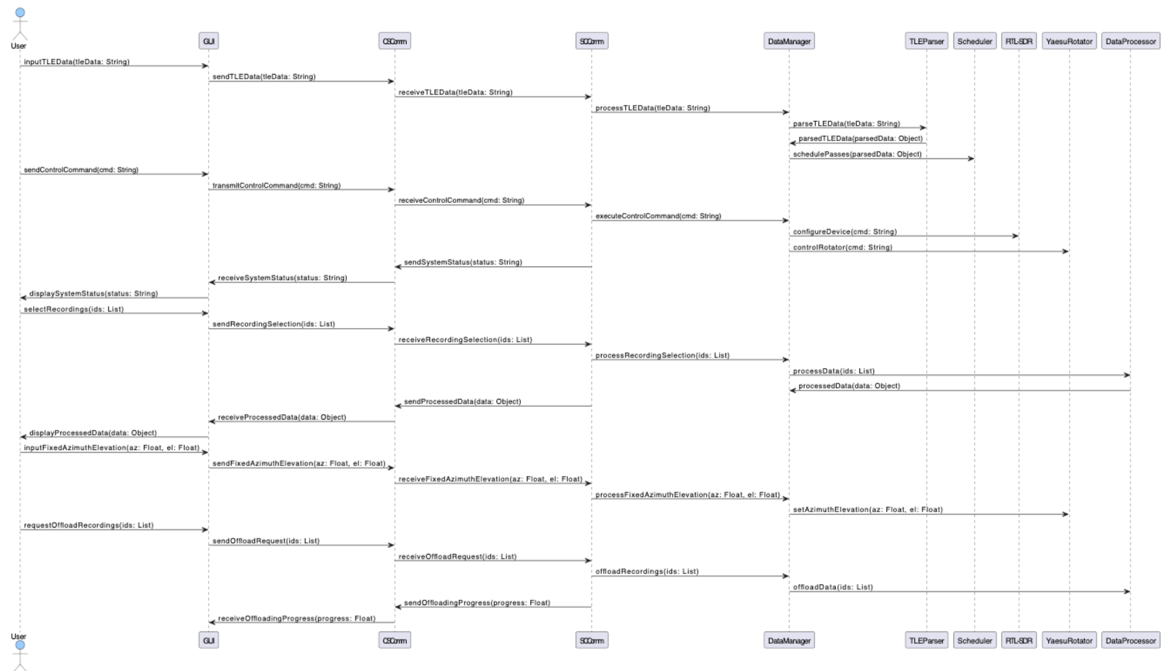
The GUI provides an intuitive interface for users to interact with the system. Users can input TLE data, send control commands, select recordings, request an offload of recordings, input fixed azimuth and elevation for stationary recording, as well as put the system to sleep.



The GUI has multiple states, depending on what the user intends to do with the system. The images provided above show the two primary states that the user will interact with. Another state allows the user to interface directly with the computer, sending commands as strings instead of using buttons and other interfaces within the GUI. This is shown below.



2.2.2 Client-Server Communication (CSComm)



CSComm is responsible for facilitating communication between the GUI and SCComm. It transmits TLE data, control commands, recording selections, and offload requests from the GUI to SCComm.

The algorithm consists of the following steps:

1. Initialize Bluetooth connection:
   a. Search for available Bluetooth devices.
   b. Display the list of devices to the user.
   c. Connect to the selected device (SCComm) using its address.

2. Listen for user input in the GUI:
   a. If the user inputs TLE data, send the data to SCComm using a formatted command like "TLE:<tle_data>".
   b. If the user sends a control command, send the command to SCComm using a formatted command like "CMD:<control_command>".
   c. If the user selects recordings, send the selection to SCComm using a formatted command like "RECSEL:<recording_ids>".
   d. If the user requests offloading recordings, send the offload request to SCComm using a formatted command like "OFFREQ:<recording_ids>".
   e. If the user inputs fixed azimuth and elevation, send the values to SCComm using a formatted command like "FIXAZEL:<azimuth>,<elevation>".

3. Monitor the Bluetooth connection:
    a. If the connection is lost or disconnected, display an error message and attempt to reconnect.
4. Terminate the connection:
    a. When the user exits the application or shuts down the system, close the Bluetooth socket and release any resources.

2.2.3 Server-Client Communication (SCComm)

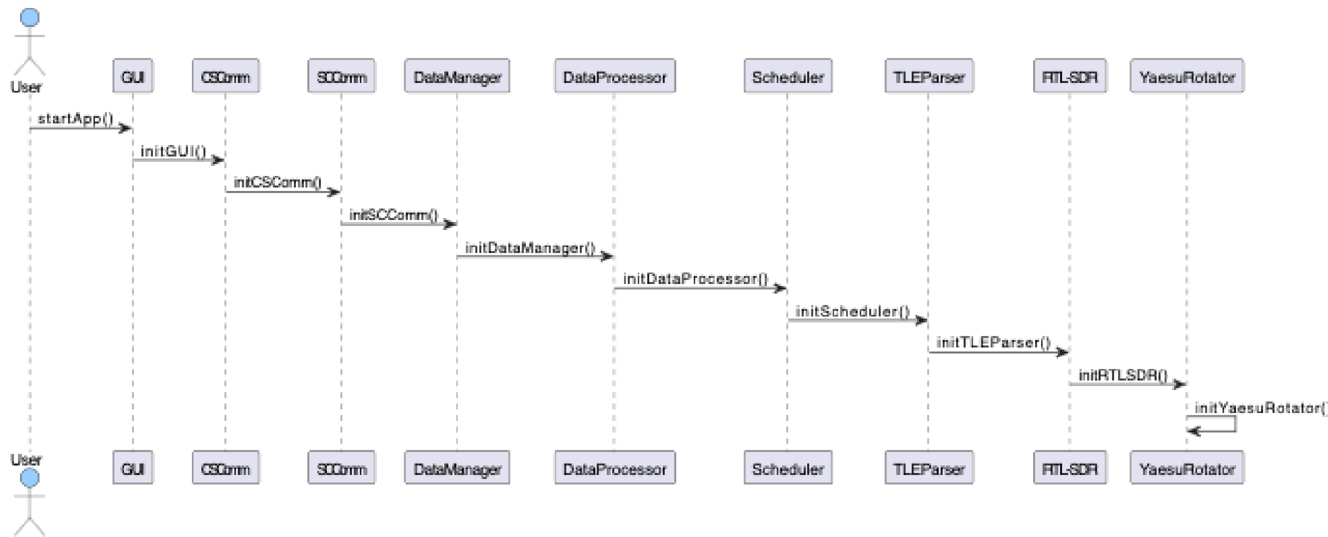
SCComm receives data and commands from CSComm, processes the received data, and executes the commands. It communicates with DataManager and DataProcessor to manage and process data.

The algorithm consists of the following steps:

1. Initialize Bluetooth server socket and await incoming connections.
2. Once connected, continuously listen for incoming data:
    a. If the received data is TLE data (starts with "TLE:"), extract the TLE data and forward it to DataManager for processing.
    b. If the received data is a control command (starts with "CMD:"), extract the control command and execute it using the DataProcessor.
    c. If the received data is a recording selection (starts with "RECSEL:"), extract the recording IDs and forward them to DataManager for processing.
    d. If the received data is an offload request (starts with "OFFREQ:"), extract the recording IDs and forward them to DataManager to initiate the offloading process.
    e. If the received data is fixed azimuth and elevation (starts with "FIXAZEL:"), extract the values and forward them to DataProcessor for processing.
3. Monitor the Bluetooth connection:
    a. If the connection is lost or disconnected, display an error message and wait for a new connection.
4. Terminate the connection:
    a. When the system shuts down, close the Bluetooth server socket, client socket, and release any resources.

2.2.4 DataManager

DataManager is responsible for scheduling satellite passes, managing recordings, and offloading recordings based on user input. It communicates with TLEParser, DataProcessor, RSPDuo Interface, and YaesuRotator Interface to perform these tasks.



2.2.5 DataProcessor

DataProcessor processes the TLE data received from DataManager and calculates satellite positions. It also processes recordings and provides processed data to the GUI.

2.2.6 TLEParser

TLEParser parses the TLE data received from DataProcessor and returns the parsed data to DataManager. It is responsible for converting raw TLE data into a format that can be used by other components.

2.2.7 Scheduler

The Scheduler component is responsible for scheduling satellite passes based on the provided TLE data. It calculates the future passes of the satellite over the observer's location and determines the start and end times of each pass. This information is used to automatically control the system to track and record satellite signals during these passes.

Scheduling Algorithm

The scheduling algorithm consists of the following steps:

1. Parse TLE data using the TLEParser module.
2. Calculate satellite positions and observer locations using the skyfield library.
3. Determine the time window for scheduling passes. This can be set to a specific duration, e.g., the next 24 hours or a fixed number of future passes.
4. Iterate through the time window and calculate the satellite's position at regular time intervals (e.g., every minute).
5. For each calculated position, determine if the satellite is above the observer's horizon. If it is, record the start time of the pass.
6. Continue iterating until the satellite is no longer above the horizon. Record the end time of the pass.
7. If more passes are required, continue iterating through the time window and repeat steps 5-6.
8. Compile a list of scheduled passes with their start and end times.

Scheduler Integration

The Scheduler component will be integrated into the DataManager module. Once the TLE data is received from the GUI and parsed by the TLEParser, the DataManager can call the Scheduler to calculate the satellite passes. The DataManager will then use this information to control the RSPDuo Interface and YaesuRotator Interface, initiating recordings and antenna position adjustments during the scheduled passes.

2.2.8 RSPDuo Interface

The RSPDuo Interface communicates with the RSPDuo hardware to configure the device, receive samples, and store the recorded data.

The RSPDuo Interface algorithm consists of the following steps:

1. Initialize the RSPDuo device using the rtlsdr library.
2. Set the required configuration parameters, such as sample rate, center frequency, and gain.
3. Start the data streaming process from the RSPDuo device.
4. Continuously read the I/Q data samples from the RSPDuo device.
5. Monitor the DataManager for commands to start or stop recording.
    a. Upon receiving a start recording command, create a new file and write the I/Q data samples to the file.

   b. Upon receiving a stop recording command, close the file and save the recorded data.
6. If the DataManager sends a command to shut down the system, stop the data streaming process, close the RSPDuo device, and clean up any remaining resources.

Interface Integration

The RSPDuo Interface component should be integrated into the DataManager module. The DataManager will be responsible for controlling the RSPDuo Interface by sending commands to start or stop recording, change device settings, or shut down the system. The DataManager will also coordinate with the Scheduler to ensure that recordings and antenna adjustments are made during the scheduled satellite passes.

2.2.9 Yaesu Rotator Interface

The YaesuRotator Interface is responsible for controlling the antenna rotator, adjusting the azimuth and elevation angles to accurately track the satellite as it passes overhead. This interface communicates with the rotator using a serial connection and the rotator's specific command protocol. The following is a detailed description of the algorithm used by the YaesuRotator Interface:

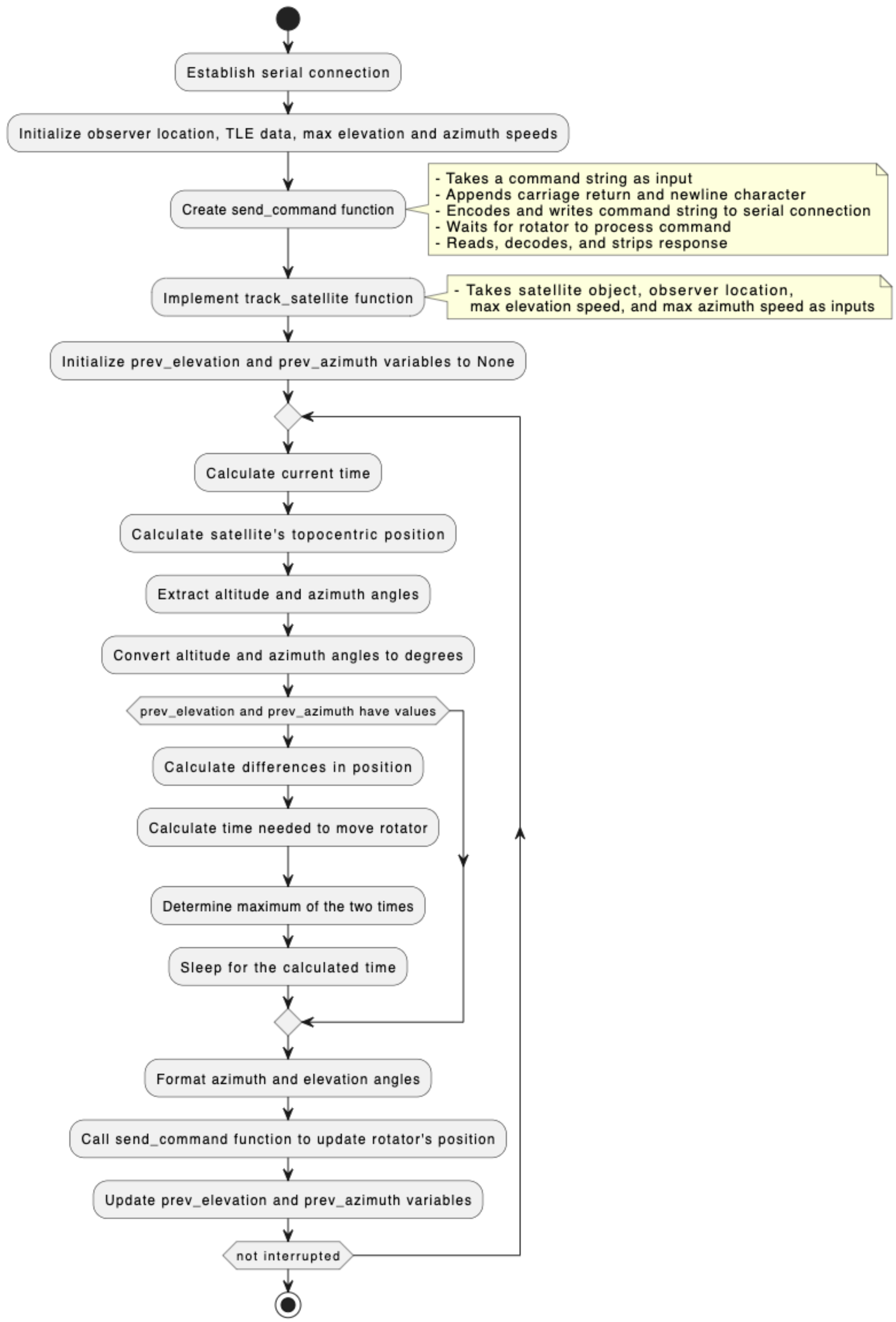
1. Initialization:
   a. Establish a serial connection to the rotator by specifying the correct serial port and baud rate.
   b. Initialize any necessary data structures or variables, such as observer location (latitude, longitude, elevation), the current satellite's TLE data, and maximum elevation and azimuth speeds.
2. Send command function:
   a. Create a send_command function that takes a command string as input.
   b. Append a carriage return and newline character to the command string, as required by the Yaesu rotator's protocol.
   c. Encode the command string into bytes and write it to the serial connection.
   d. Wait for a brief period (100 ms) to allow the rotator to process the command.
   e. Read the response from the rotator, decode it, and strip any leading or trailing whitespace.
3. Track satellite function:
   a. Implement a track_satellite function that takes a satellite object, observer location, maximum elevation speed, and maximum azimuth speed as inputs.
   b. Initialize prev_elevation and prev_azimuth variables to None.

c. Continuously loop through the following steps until interrupted (e.g., by a KeyboardInterrupt exception):
    i. Calculate the current time using a library like Skyfield's timescale.
    ii. Calculate the satellite's topocentric position relative to the observer's location at the current time.
    iii. Extract the altitude and azimuth angles from the topocentric position.
    iv. Convert the altitude and azimuth angles from radians to degrees.
    v. If prev_elevation and prev_azimuth have been assigned values (i.e., not None):
        1. Calculate the difference in position since the last update.
        2. Calculate the time needed to move the rotator based on elevation and azimuth differences and their respective maximum speeds.
        3. Determine the maximum of the two times to avoid overshooting.
        4. Sleep for the calculated time.
    vi. Format the azimuth and elevation angles as a command string according to the rotator's protocol (WXXX YYY, where XXX is the azimuth angle and YYY is the elevation angle, both in degrees).
    vii. Call the send_command function with the formatted command string to update the rotator's position.
    viii. Update the prev_elevation and prev_azimuth variables with the current elevation and azimuth.

Error handling and cleanup:

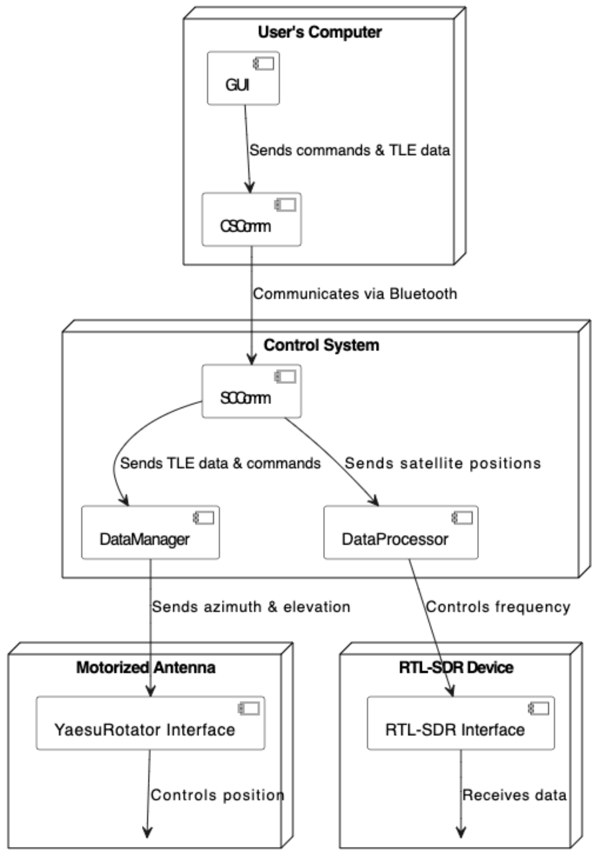
1. Implement error handling for exceptions that may occur during the execution of the tracking algorithm, such as serial communication errors or invalid TLE data.
2. In the case of an exception or interruption, exit the tracking loop and clean up any resources, such as closing the serial connection.

By implementing this algorithm, the YaesuRotator Interface can accurately control the antenna's position to track the satellite's path across the sky. This ensures optimal signal reception and data collection during satellite passes.



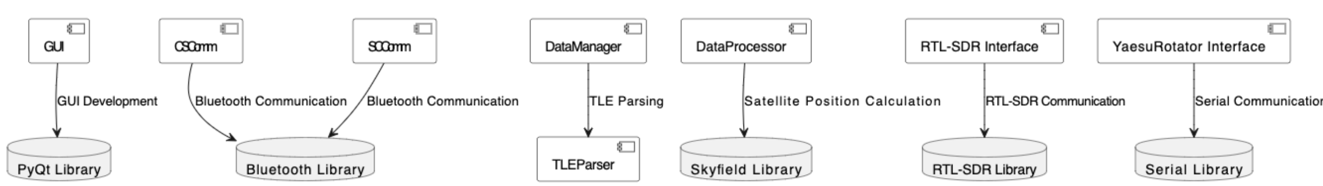
2.3.0 Operations Software Implementation

The operations software will be implemented using Python, as it provides extensive libraries for handling communication, data processing, and interfacing with hardware components. The GUI will be developed using PyQt, a Python library for creating desktop applications.



2.3.1 Required Libraries

bluetooth (for Bluetooth communication)
rtlsdr (for RSPDuo interface)
serial (for YaesuRotator interface)
skyfield (for TLE parsing and satellite position calculation)
PyQt (for creating the GUI)
Matplotlib (for plotting spectrums)



2.3.2 Executable Creation

Creating a standalone executable for the operations software allows for easy installation on Linux systems without requiring the client to have Python or other dependencies installed. To achieve this, the PyInstaller tool will be used to package the Python code and its dependencies into a single executable file.

The following steps outline the process of creating the executable:

1. Ensure that the operations software codebase is complete and functional.
2. Install PyInstaller if it is not already installed by running the following command:

   ```
   pip install pyinstaller
   ```

3. Open a terminal or command prompt and navigate to the directory containing the operations software code.
4. Run the PyInstaller command with the appropriate options:

   ```
   pyinstaller --onefile --windowed operations_software.py
   ```

   Replace `operations_software.py` with the main Python script file name of the operations software.
   a. - The `--onefile` option tells PyInstaller to create a single executable file instead of a collection of files.
   b. - The `--windowed` option specifies that the executable should run without showing a console window.
5. PyInstaller will analyze the code and its dependencies, and generate the executable in the `dist` directory within the operations software directory.
6. Test the generated executable on a Linux system to ensure its functionality and verify that all necessary dependencies are included.

Note that the size of the executable may be larger than the original Python script due to the inclusion of the Python interpreter and required libraries.

2.4.1 Raspberry Pi Linux Service for Bluetooth Discoverability and Slave Side Operations Software

The Linux service is responsible for running a Python script that makes the Raspberry Pi Bluetooth discoverable and starts the main script for the slave side operations software. The service ensures that the Raspberry Pi is ready for communication with the client side operations software and manages the startup and shutdown of the slave side script.

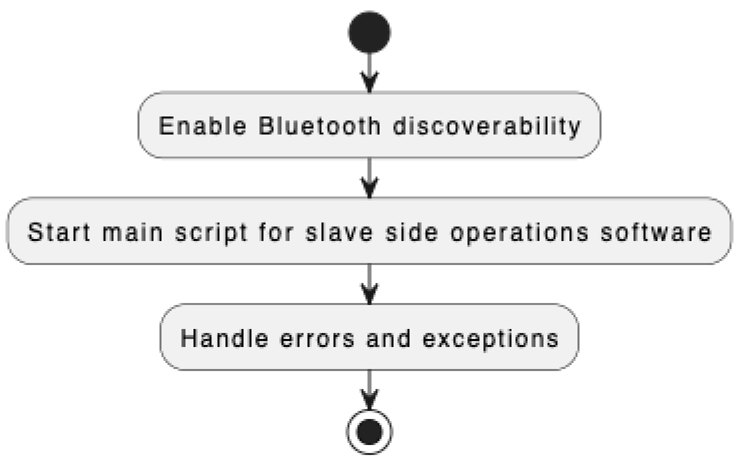
2.4.2 Service Configuration

The Linux service should be configured to start on boot, ensuring that the Raspberry Pi is discoverable and the slave side operations software is ready to run whenever the system is powered on. To achieve this, a system service file should be created with the appropriate settings and placed in the /etc/systemd/system directory. The service file should be configured to run the Python script as a user with the necessary permissions to access the Bluetooth device and other required resources.

2.4.3 Service Python Script

The Python script executed by the Linux service is responsible for the following tasks:

1. Enabling Bluetooth discoverability on the Raspberry Pi.
2. Starting the main script for the slave side operations software.
3. Managing any errors or exceptions that may occur during the execution of the main script.



3.1 Mount Subsystem

The mount subsystem encompasses the majority of physical hardware of the system, and is primarily tasked with mounting the antenna and receiver systems, stability of the system, azimuth and elevation rotation and environmental shielding. There are four separate subcomponents of the mount, those being the rotator system for azimuth and elevation control, the tripod for stabilizing and mounting purposes, the mount control box used to interface with the computer system, and the enclosure.

3.2.1 Mount System Requirements

The components chosen for the mount subsystem were picked primarily in accordance with the needs of the other subsystems, as the majority of the requirements for the mount were determined by the constraints of the other subsystems. The initial requirements for the mount subsystem are shown below:

- The system shall track a satellite until the signal from it disappears.
- The system shall observe a point of space given an azimuth and elevation measurement.

This is the primary subsystem requirement, as this is the main requirement for the rotator component, which is what the rest of the subsystem is designed around.

- The system shall operate within an ambient air temperature band of 14°F to 100°F (-10°C to 40°C).
- The system shall operate within the regular weather climate of Blacksburg (ex: light rain, snow).
- The system shall withstand lightning level current.

These requirements are the environmental concerns associated with the mount, specifically the tripod and controller portions of the mount.

- The system shall be transportable a minimum of 100 yards by two average adults.
- The system shall be shut down, disassembled, and packed up in less than ten minutes.
- The system shall be assemblable in less than five minutes.
- The system shall fit into two medium sized hard shelled containers

These requirements are the size concerns associated with the mount, to ensure that the mount design is sized appropriately to the desired task.

3.2.2 Mount System Derived Requirements

Using the above design considerations as a basis, the derived requirements for this project are as follows:

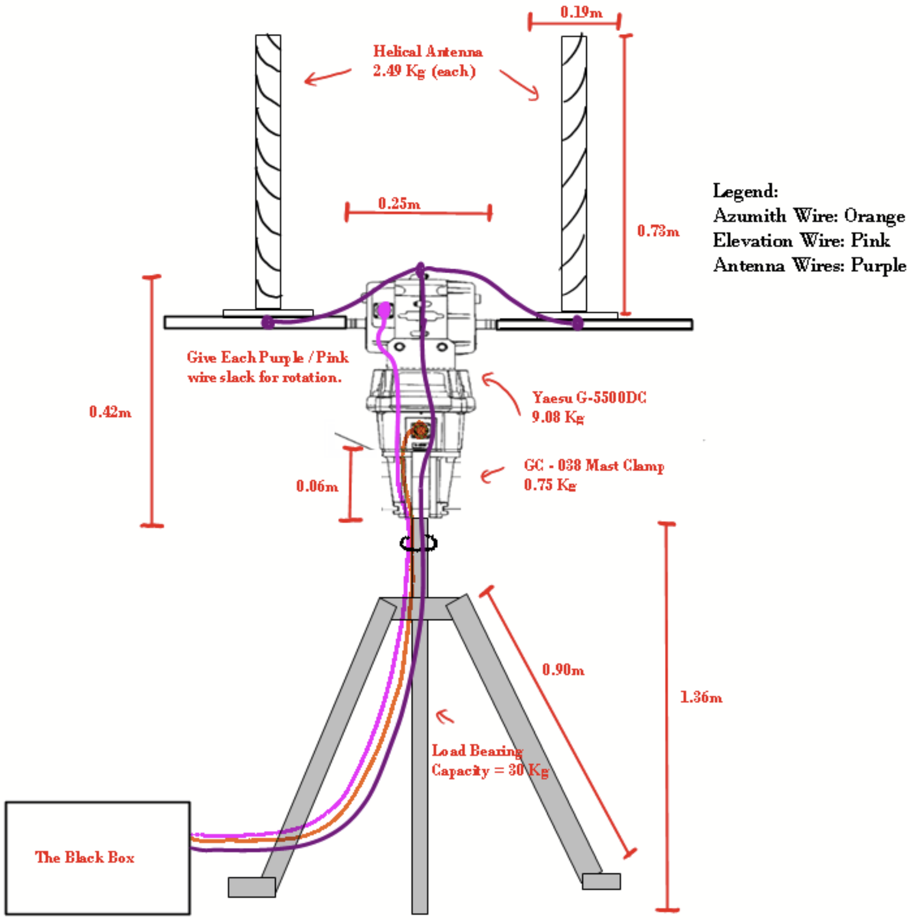
The mount shall:

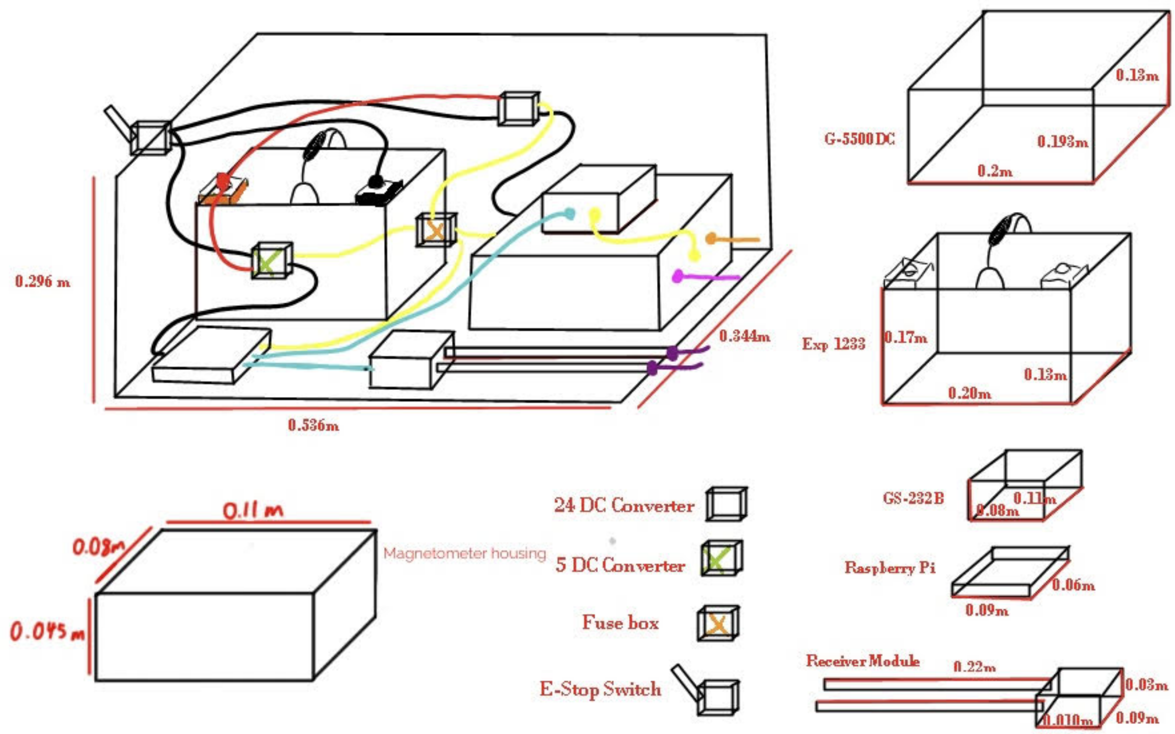
- Meet the rotational torque requirement to support our antenna at an Azimuth angle of 0° - 360° and Elevation angle of 0° - 180°.
- Be weather resistant to snow and light rain, as well as being capable of functioning in these environments for the allotted time.
- Have the rotational speed and continuous operational time necessary to track satellites during their orbit.

In addition, the mount should:

- Be USB capable, and able to communicate with the onboard computer over a USB connection.
- Be assemblable in less than three minutes.
- Remain under a total weight of 30 Kg and cost less than $1100 for the rotator and all required additional systems.

3.3 Mount Schematic





3.4 Rotator Design

The rotator system is the main component of the mount, and is tasked with installing the antenna system and pointing it towards incoming satellites. This task requires that the mount is capable of supporting the chosen antenna at all required angles, as well as having the slew rate necessary to adjust the azimuth and elevation of the antenna at a high enough adjustment speed to track satellites. The rotator chosen for this is the Yaesu G-5500 DC, a dual azimuth / elevation rotator designed to support portable satellite communications installations and currently being sold for $759.99 USD. The device specifications for the Yaesu G-5500 DC are shown below:

SPECIFICATIONS

Voltage requirement:	110-120 or 200-240 VAC
Motor voltage:	22 VDC
Rotation time (approx.):	Elevation (180°): 65 sec. ± 20% Azimuth (360°): 60 sec. ± 20%
Maximum continuous operation:	3 minutes
Rotation torque:	Elevation: 12 kg-m (88 ft-lbs) Azimuth: 6 kg-m (44 ft-lbs)
Braking torque:	Elevation: 40 kg-m (289 ft-lbs) Azimuth: 40 kg-m (289 ft-lbs)
Vertical load:	200 kg (440 lbs)
Pointing accuracy:	±4 percent
Wind surface area:	1 m^2
Mast diameter:	38-63 mm (1-1/2 to 2-1/2 inches)
Boom diameter:	32-43mm (1-1/4 to 1-5/8 inches)
Weight (approx.):	Rotators: 8 kg (17.7 lbs) Controller: 3 kg (6.6 lbs)

The primary concern for the rotator is whether or not it is able to be mounted by the antenna for our project while still being able to track a satellite, which was the main driving force behind the choice of mount. The antenna system, due to it mainly being constructed out of PVC and copper wire, is estimated to weigh around 5.5 kg, which is able to be handled by our rotator system as the maximum rotation torque is 6 kg-m. The rotator is also capable of moving in a full azimuth angle of 360 degrees and an elevation angle of 180 degrees, giving the rotator the full capability of tracking satellites in their movement throughout the horizon. The two rotator components are also encased inside of a melamine coated aluminum casing to keep them weather resistant, although the controller components of the rotator are not weather resistant. As such, when designing the enclosure for the system, the components of the rotator that are not weather resistant (those being the rotator controller and the computer interface) are stored inside of a weatherproof container.

The full rotation time (for both azimuth and elevation) is estimated to be roughly 60 seconds for both planes, which meets our requirements as the requirement for this system is to be able to have the slew rate necessary to track the fastest data satellites (LEO satellites). As stated below, the visible time for an LEO satellite is 20 minutes for data transfer purposes, meaning that we should have more than enough time with the estimated 60s rotation time to make the necessary adjustments to track the satellite.

Satellite Lithium-Ion Batteries

Yannick Borthomieu, in Lithium-Ion Batteries, 2014

2.2 LEO Satellites

LEO satellites have a circular (or elliptical) orbit at a height of 250–2000 km from the Earth surface (Figure 14.3). The orbit period, mainly depending on the altitude, varies in the range 90–120 min. As the altitude of LEO satellites is low, their velocity is very high (>25,000 km/h) and they make 12–16 Earth turns per Earth day. It means that a LEO satellite experiences at least 12 to 16 sunlight and night periods in 24 h. Consequently, in LEO orbit, the maximum time during which a satellite is above the local horizon for an observer on the Earth is up to 20 min. This time is used to transfer data, images, and photographs to selected ground stations positioned in strategic locations.

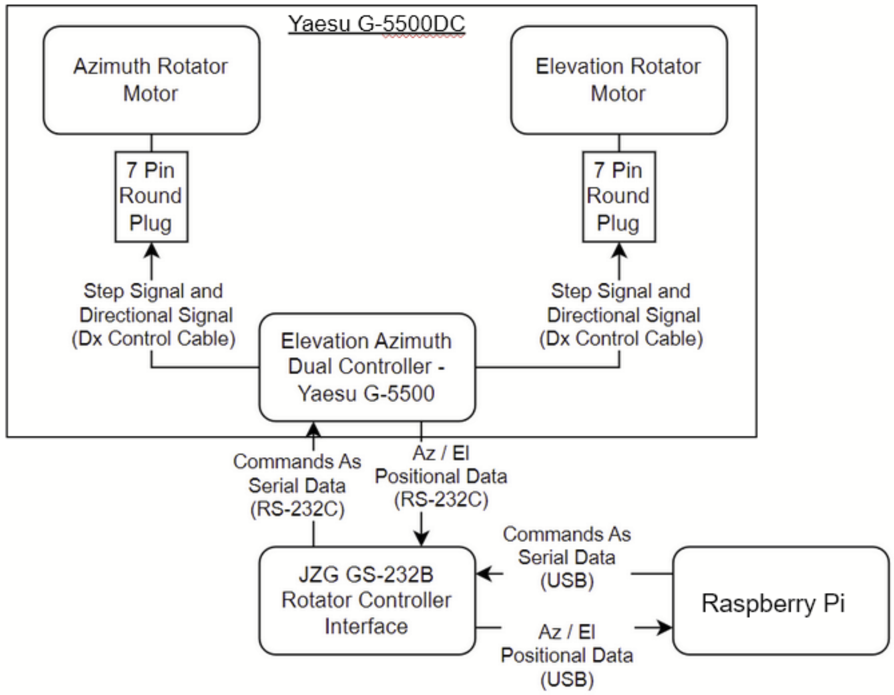
https://www.sciencedirect.com/topics/engineering/low-earth-orbit

The Yaesu G-5500 DC is delivered including the required mount controller unit and mast clamp, meaning that the necessary components for initial setup of the mount are provided within the mount package. The issues associated with this mount choice is that the provided mount controller component does not initially have USB capability, and in addition the rotator system needs a platform upon which to mount itself and additional wires to connect to the battery and computer. The additional components of the mount are aimed to solve these issues.

3.5 Additional Mount Components

3.5.1 The Computer Interface

The chosen component to establish USB capability between the mount and the computer system is the JZG GS-232B, which allows for digital control over a wide range of rotator designs, one of which being the G-5500 DC chosen for this project. The component establishes a connection to the computer using a USB connection, and to the rotator controller box using a 8-pin connection wire, a component provided within the JZG GS-323B interface kit. The wiring diagram for the system, as well as the commands sent from the separate components are shown below:

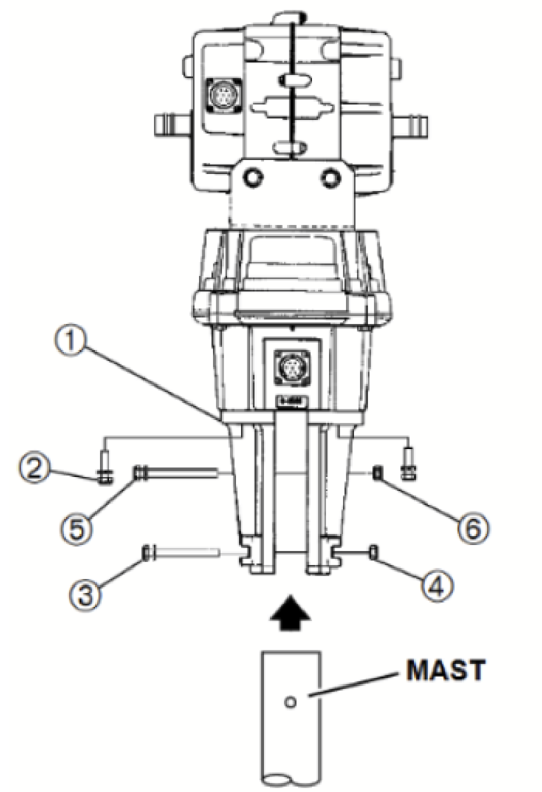


The software aspects of the interface between the mount and the computer are shown above, and the above picture depicts the wiring diagram and rough structure of the setup of the mount.

3.5.2 The Tripod

The last necessary component of the mount subsystem is the tripod component, which is used to stabilize the system and provide a base on which the rotator and antenna can be mounted. The chosen tripod for this project is the WEVZENEY Antenna Tripod, currently being sold for $59.99. The load-bearing capacity of the antenna is stated to be 30 kg, and as can be seen from the diagram above the antenna components in addition to the rotator will weigh roughly 15.5 - 16 kg. The torque requirement for the tripod, based on the torque requirement of the mount subsystem and the weight of the antenna, is roughly 25 kg, which approaches the maximum load of the tripod. As such, the tripod will have to be staked into the ground, which it is initially designed to do because of the added stakes in the tripod. Doing this will increase the load bearing capacity of the tripod, and return the tripod to normal operation levels.

The tripod is stated to be weather resistant and rust-proof due to the iron varnish on the device, and the reviews from previous users support these claims. The tripod mounting diagram is shown below:



3.6 Enclosure

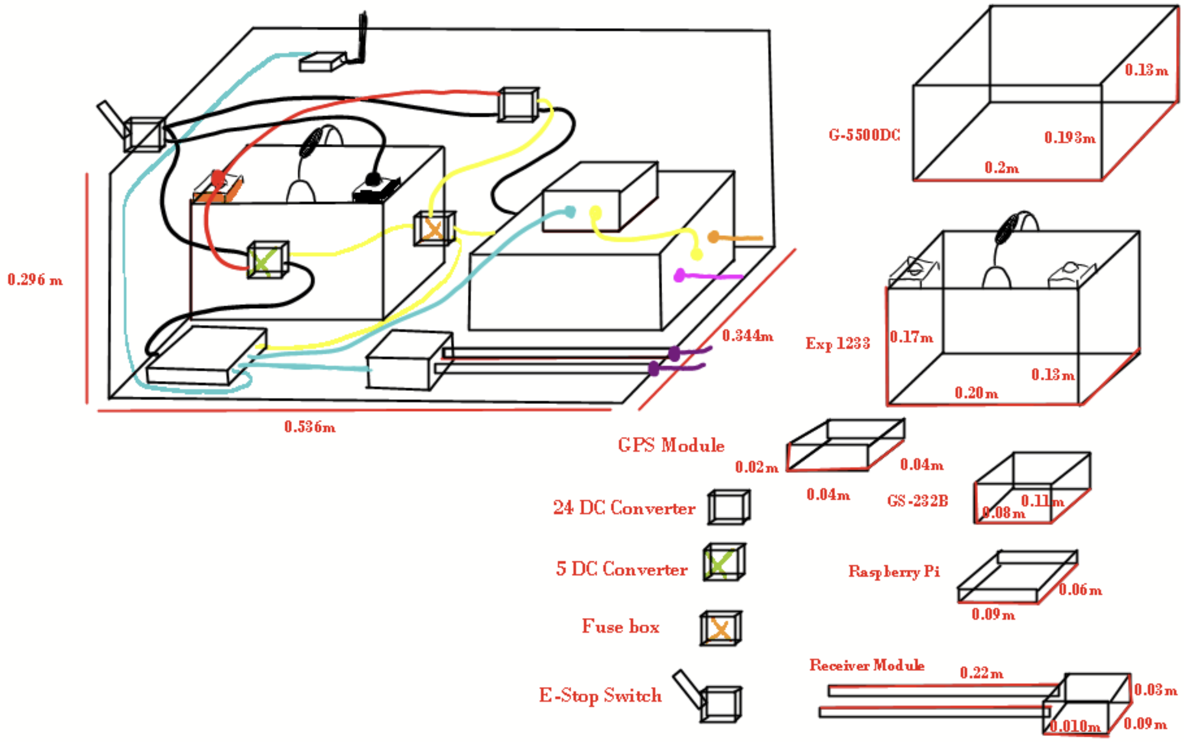
The main enclosure for the system will house the computer, battery, filters, receiver, and the mount controller. This is to ensure that all of these subsystems remain in a dry environment. To this end, the main enclosure will have to be fairly large.

The magnetometer will also require a separate enclosure though much smaller, as it has to be mounted to the top of the Yaesu G-5500 DC, to properly calculate the heading of the system, ensuring that we are pointing north during setup.

3.6.1 Enclosure Size

The size of the main enclosure will have to house the battery, which is 7.72"x 5.16"x 6.4", all of the filters in a solid line, which totals 9" when connected to the LNA, which also has to connect to the receiver. Together, this totals a length of nearly a foot. The main enclosure will also have to house the controller for the mount, which is 7.87"x7.48"x5.11". This, together with the converters and the GPS module that goes with the computer will take up most of the ground room in the storage container. The GPS module will have to be placed on top of the main enclosure, but due to the module and the associated wires having an IPX4 weather resistance rating in terms of snow and rain, this component can be exposed to the environment without concerns of weather-associated damages.

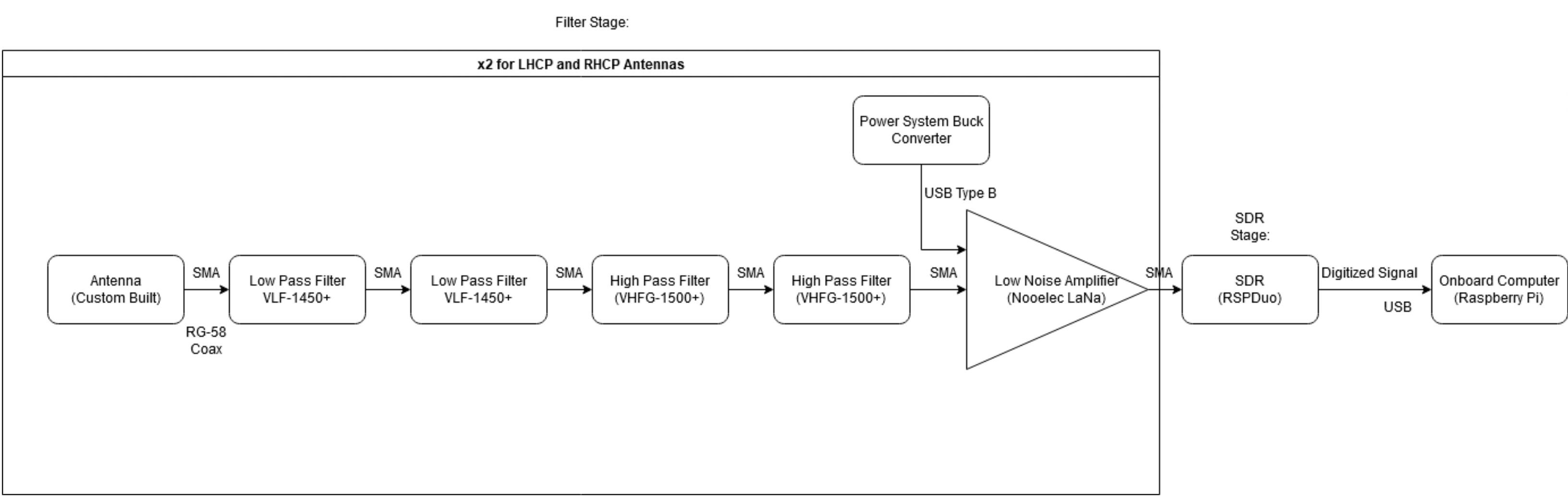
3.6.2 Main Enclosure Layout



The general layout of the main enclosure is shown above. The only cable exit will be the cables going to the mount, and the two antenna cables. The emergency switch will be built into one of the sides. any holes that are drilled into the main enclosure will be sealed with silicone to keep the box watertight. The fuse box will be centrally located in order to reduce hassle when replacing fuses.

4.1 Receiver Subsystem

The receiver subsystem consists of two stages, the filtering stage, and the SDR stage.



Frequency (MHz)	RspDuo Noise Figure (dB)	External LNA Noise Figure (dB)	External LNA Gain (dB)	Cable Loss ~10ft (dB)	Reference Noise Temperature (K)	Antenna Noise Temperature (Estimate) (K)	System Noise Temperature (K)
1400	4.66	1.2	26	2.06	300	35	129.989
1500	4.95	1.24	26	2.159	300	35	133.847
1600	5.27	1.3	26	2.256	300	35	139.592
1700	7.47	1.4	26	2.352	300	35	151.575
Average	5.5875	1.285	26	2.20675	300	35	138.75075

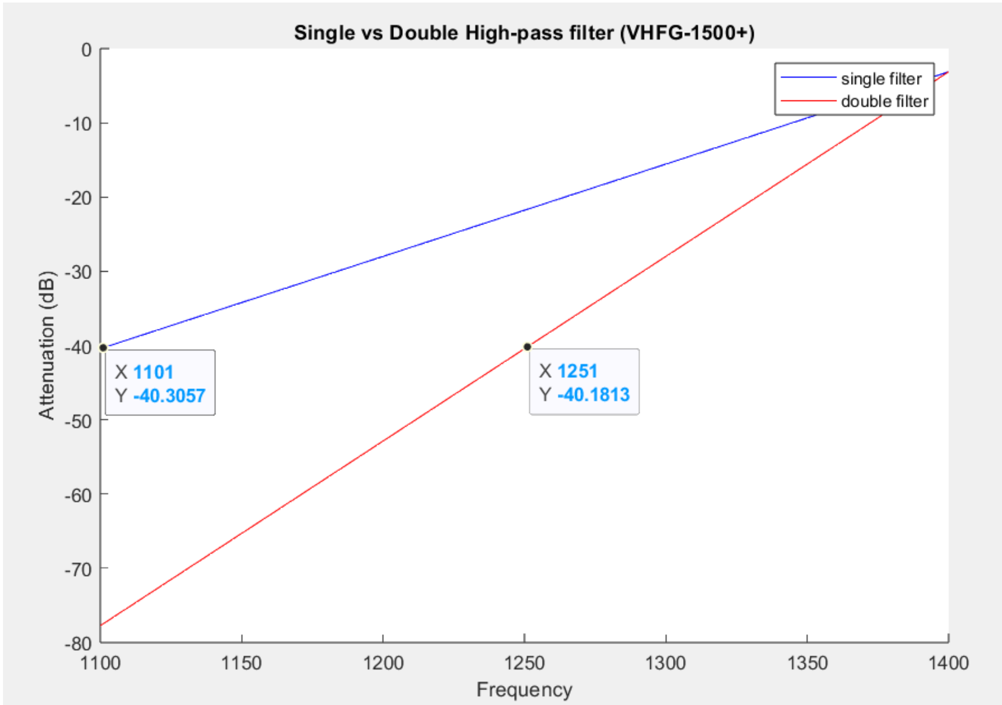
The receiver subsystem design is constrained by five system requirements:

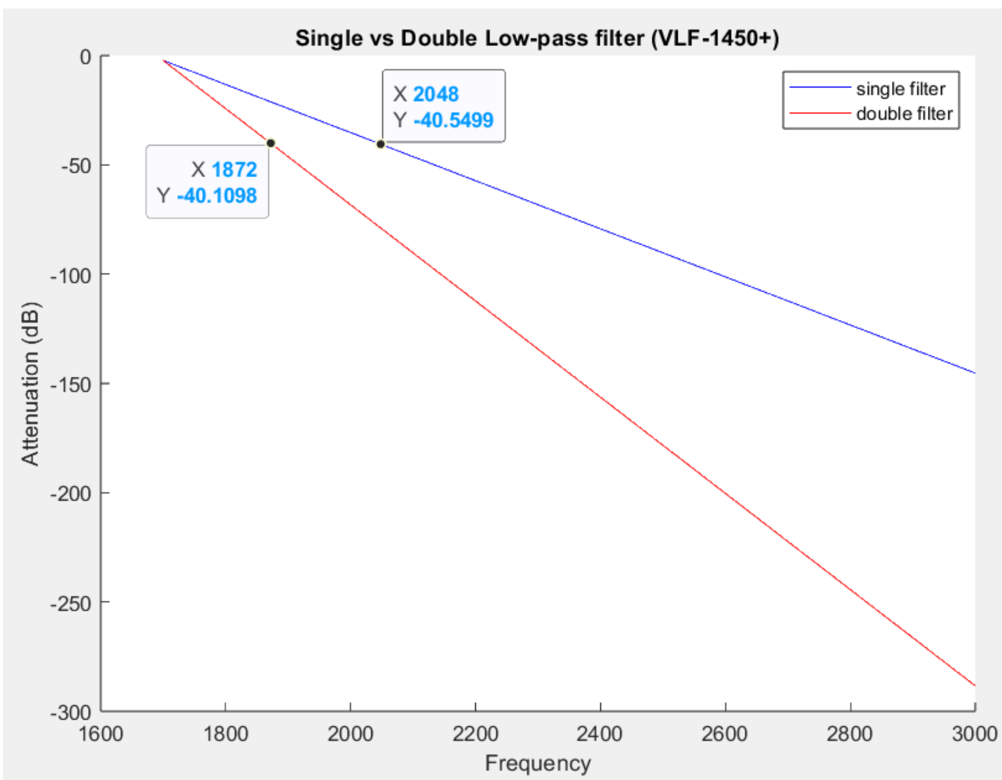
- FUN-2: The system shall have a receiver noise temperature less than 200K.
- FUN-4: The system shall have a receiver whose bandwidth is variable up to 2MHz minimum, with up to 10MHz desired.
- FUN-5: The system shall have >40 dB rejection of 0.5-1350 MHz and 1800-2700 MHz signals.
- PER-2: The system's receiver shall have a tunable range between 1550-1700MHz. 1400-1700MHz desired.
- COST-1: The system shall be replicable for < $2000 USD

FUN-2 is relevant to both stages of the receiver, while FUN-4 and PER-2 are relevant to the SDR stage, and FUN-5 is relevant to the filtering stage.

4.1.1 Filtering Stage

The filtering stage is designed to preselect signals by attenuating signals at frequencies below 1350Hz and above 1800Hz to >40dB. In order to achieve this goal, two pairs of low pass and high pass filters are used. The low pass filters are the Minicircuits VLF-1450+ filters, and the high-pass filters are VHFG-1500+ filters. By arranging these filters in pairs of two a steep enough attenuation curve is obtained to achieve *nearly* the system requirements:





As can be seen in the above graphs, by doubling the order of the low and high-pass filters, adequate preselection of signals below 1250Hz and above 1872Hz can be achieved. The ability to meet the preselection requirement, FUN-5, is limited by other requirements, namely, FUN-2 and COST-1. As more filters are added in series, the reflection between each SMA connection increases, and therefore the overall VSWR of the receiver subsystem. This increase in VSWR impacts our ability to meet the FUN-2 requirement, as a higher VSWR increases the noise figure of the receiver.

Increasing the number of filters will also increase our cost considerably, since the number of filters in the filtering stage is doubled, one for each antenna. Due to these considerations, this level of preselection has been discussed with the customer and has been determined to be adequate at this stage in the design; however, more research and development of the filtering system will be conducted in the future to potentially improve this performance.

At the end of the filtering stage is a low noise amplifier (LNA). This LNA enables us to increase the gain of our desired signals within the passband, as well as helps us to decrease our overall noise temperature of the system, allowing us to meet FUN-2.

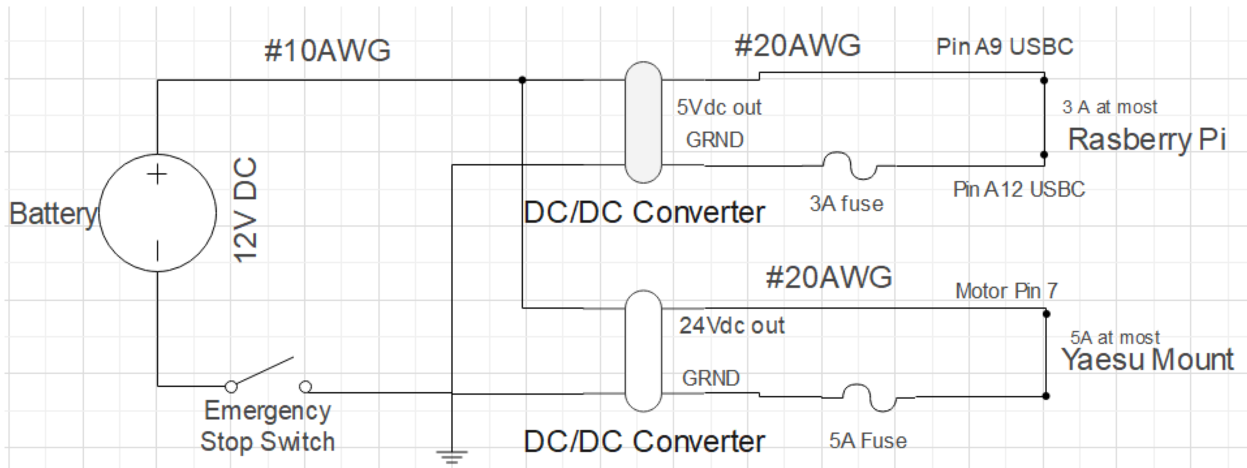
All of these connections between filters, the LNA, and the SDR are made using SMA-SMA connections.

4.1.2 SDR Stage

The SDR used is the SDRPlay RSP-Duo. The RSP-Duo is tunable to up to 2GHz, and has a tunable bandwidth of up to 2MHz when both SMA inputs are used, and up to 10MHz when one input is enabled at a time.

5.1 Power System Introduction

The power system is responsible for safely supplying a steady supply of power to all components of the system for a continuous period of 18 hours, without significant voltage drop, while ensuring that no components are harmed by current overdraw. The generalized circuit for the system is shown below. This includes all fuses and switches.



5.2 Power system specifications

To facilitate the power requirements of the systems the battery will have to power, a power budget was calculated using the current and voltage requirements of each system. The time used for each system varies, and each is shown below.

- The mount will have to be powered for a total of 1.5 hours.
- The computer will have to be powered for 16 hours at low power/idling.
- The computer will have to be powered at maximum current draw for 2 hours to facilitate the recording of signals and the processing of data.

The calculated power budget is shown below.

System	Voltage required	Amperage total	Time (hours)	Watt Hours	Amp Hours
Computer	5V	3A	2	30Wh	2.5Ah
Computer(low power)	5V	0.514A	16	41.12Wh	3.51Ah
Mount	24V	4.5	1.5	162Wh	13.5
Total	N/A	N/A	N/A	233.12Wh	19.51

5.2.1 Battery Voltage

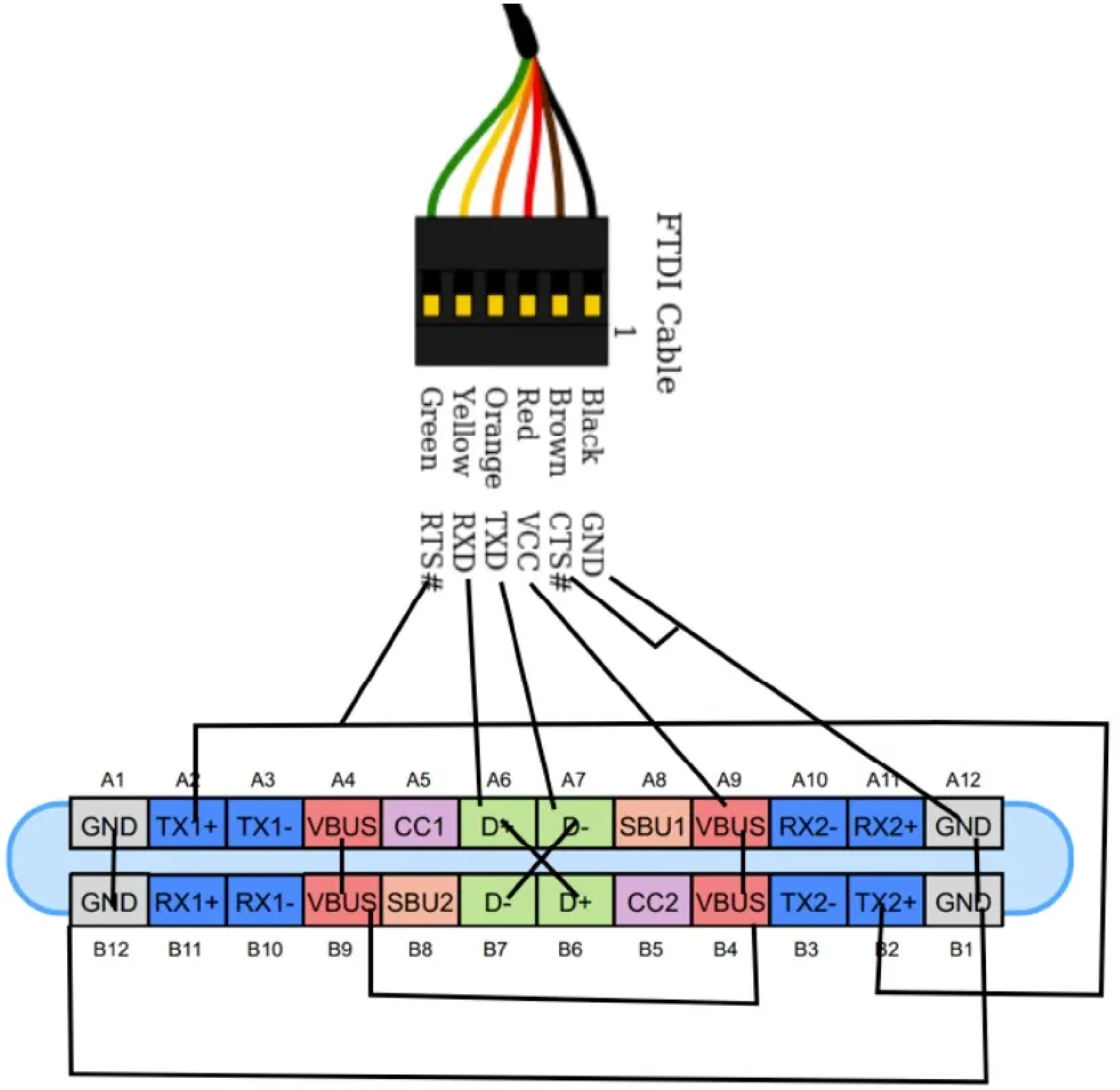
The chosen battery is a deep cycle lead acid battery, which has a capacity of 33 AH at 12V. A deep cycle battery was chosen over a standard car battery because of the superior discharge over time characteristics. Whereas a car battery would lose voltage very quickly over a smaller period of time, a deep cycle battery is more suited for steady current draw over a larger time window.

5.2.2 DC-DC Converters for Voltage Regulation

Since the various components of the other systems do not require 12 volts, this voltage will have to be stepped up or down in order to facilitate their voltage requirements, This will be done using off the shelf DC to DC converters.

5.2.2.1 Computer power

The computer's power will be supplied by the battery through a 12V-5V converter. This converter will have a secondary line going to the two LNAs in the receiver subsystem. Once the power has been stepped down, it will have to be supplied to the computer through a USBC cable. The pins for this are pins A9 and A12, shown below. Pin A9 should be connected to the power in, and A12 hould be connected to the ground of the system. In wires, this translates to red and black respectfully.



This cable will then plug directly into a separate power unit for the computer. This unit is to prevent damage to the computer in case of sudden power loss.

5.2.2.2 Mount Power

The mount will require a supply of 4.5 amps at a maximum, with a voltage of 22V. However, the tolerance for this voltage is within 24V. Since this is a much more common converter, this is the voltage that will be supplied. This converter will also be off the shelf to facilitate simplicity and ease of manufacture/maintenance.

5.3 Safety

There will be three main safety features on the power system.

- An emergency stop switch
- A fuse box

These two safety features will keep the system from damage in case of current overdraw, and will provide a quick and easy way to switch off the power in case of an emergency should the need arise.

5.3.1 Fuses

There will be two fuses in the system. One will be placed directly before the computer, but after the converter. This is because the inverter can supply more current than the computer can take, and so that should an overcurrent occur, the computer will not be damaged before the fuse breaks the circuit. The other fuse will be in a similar location, before the direct path to the mount but after the DC to DC converter. This should keep the mount safe from a dangerous overcurrent.

5.3.2 Emergency Cutoff Switch

The emergency cutoff switch is a manually operated single pole single throw switch connected to the negative terminal of the battery. It is intended to provide a manual way of safely cutting off all power to the system. It does this by eliminating the path to ground from the entire circuit.

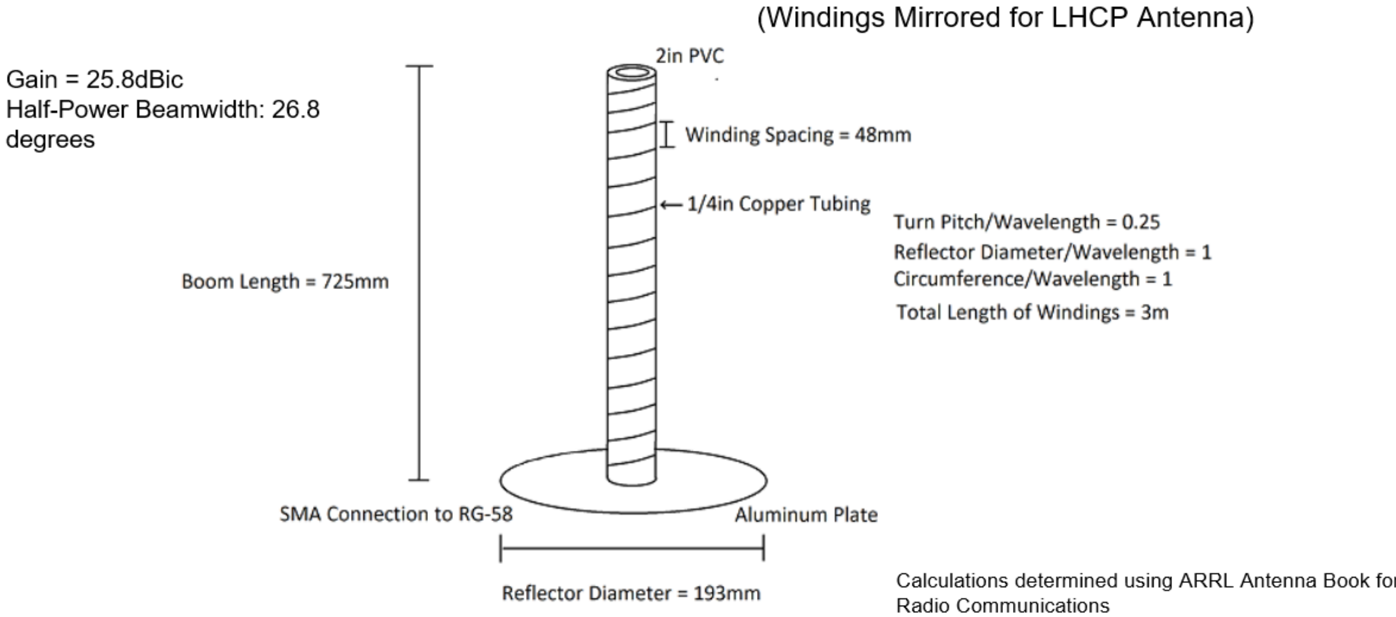
5.4 Wire Gauge

The wire gauge required for the various parts varies. To go from the positive terminal of the battery to the to DC/DC converters, #10 AWG is required. This is due to the current limitations imposed on wire gauges.For this project, a single core #10 AWG is used, which is rated for 52 amps. Another reason for choosing this wire is because the thicker the wire, the more temperature resistant it is. Since this system could easily be operating in cold weather as well as hot, the wire must be temperature resistant.

For the wire to the DC/DC converters, #18 AWG was chosen. This is because it is cheaper and easier to obtain. This wire will also serve as the main wire of the system after the converters. The only exception to this rule is the computer, which will require the wires to be connected to pins on a USBC cable.

AWG	Diameter *(mm)*	Diameter *(in)*	Square *(mm²)*	Resistance Copper *(ohm/1000m)* (ohm/1000ft)	Resistance Aluminum *(ohm/1000m)* (ohm/1000ft)	Typical Max. Current Load Ratings - Copper *(amps)*[1)]					
						Single Core	Multicore				
							up to 3 cores	4 - 6 cores	7 - 24 cores	25 - 42 cores	43 and above
40	0.08	.	0.0050	3448	5300						
39	0.09	.	0.0064	2693	4141						
38	0.10	0.0040	0.0078	2210	3397						
37	0.11	0.0045	0.0095	1810	2789						
36	0.13	0.0050	0.013	1326	2038						
35	0.14	0.0056	0.015	1120	1767						
34	0.16	0.0063	0.020	862	1325						
33	0.18	0.0071	0.026	663	1019						
32	0.20	0.0080	0.031	556	855						
30	0.25	0.010	0.049	352	541						
28	0.33	0.013	0.080	216	331						
27	0.36	0.014	0.096	180	276						
26	0.41	0.016	0.13	133	204						
25	0.45	0.018	0.16	108	166						
24	0.51	0.020	0.20	88	133	3.5	2	1.6	1.4	1.2	1.0
22	0.64	0.025	0.33	52	80	5.0	3	2.4	2.1	1.8	1.5
20	0.81	0.032	0.50	34	53	6.0	5	4.0	3.5	3.0	2.5
18	1.0	0.040	0.82	21	32	9.5	7	5.6	4.9	4.2	3.5
16	1.3	0.051	1.3	13	20	15	10	8.0	7.0	6.0	5.0
14	1.6	0.064	2.1	8.2	13	24	15	12	10	9.0	7.5
13	1.8	0.072	2.6	6.6	10						
12	2.1	0.081	3.3	5.2	8.0	34	20	16	14	12	10
10	2.6	0.10	5.3	3.3	5.0	52	30	24	21	18	15

6.1 Antenna Subsystem



Relevant Requirements:

- FUN-3: The system shall have an antenna that is RHCP polarized. LHCP Polarization is desired but not required.
- PER-2: The system's receiver shall have a tunable range between 1550-1700MHz. 1400-1700MHz desired.

The design of the antenna is relatively simple. Calculations are made based on the ARRL Antenna Book for Radio Communications. The dimensions are calculated with a center frequency of 1550MHz. Axial mode helical antennas have a fractional bandwidth of 56% of the center frequency. Therefore the antenna will have an operable bandwidth of 1,116MHz to 1,984MHz. This meets the PER-2 requirement. There will be two antennas. One wound in the right hand direction in one wound in the left, which achieves RHCP and LHCP polarization.

7 Acceptance Test Plan

Material to be delivered to customer NLT 2 weeks prior to date of in-person demo:

[ ] Electronic copy of the user manual (deliverable) explaining how to set up the user device, including software access if needed.
Material to be delivered to customer NLT 48 hours prior to the date of in-person demo:
[ ] Video (deliverable) documenting a successful acquisition and tracking of an Iridium satellite.
[ ] Electronic copy of the construction manual (deliverable) detailing assembly instructions, component list, and necessary tools.
[ ] Report documenting measurement of antenna beamwidth, ensuring it meets the requirement of having a bandwidth variable up to 2MHz minimum, with up to 10MHz desired (FUN-4).
[ ] Report documenting measurement of receiver frequency response, demonstrating preselection requirement of having >40 dB rejection of 0.5-1350 MHz and 1800-2700 MHz signals (FUN-5) is met.
[ ] Report documenting measurement of receiver temperature, demonstrating that the requirement of having a receiver noise temperature less than 200K (FUN-2) is met.
[ ] Complete, documented, and user-accessible GitHub repo (deliverable) containing all source code, schematics, and documentation.

In-person demo: Conducted indoors, location Whittemore landing/balcony. Customer brings a laptop running Linux Ubuntu, configured according to the user manual provided.
Demo script:

[ ] Demonstrate that pre-deployment equipment meets size (fits into two medium-sized hard-shelled containers - ME-4) and weight (transportable a minimum of 100 yards by two average adults - ME-1) requirements.
[ ] Demonstrate setup within 5 minutes (ME-3).
[ ] Demonstrate use of user device to command pointing to specified azimuth and elevation (I/O-4).
[ ] Demonstrate accessible azimuth and elevation ranges, confirming they meet the requirement of physically pointing the antenna at any satellite traveling across the sky from horizon to horizon (FUN-1).
[ ] Demonstrate slew rate, confirming required slew rate of motor subsystem by measuring the azimuth and elevation turning rates, and confirming that these values are adequate to keep up with LEO satellites
[ ] Set up a weak tone transmitter in the room. Transmitter turns on and off with 1 second period. Using a "fixed" mode acquisition task, starting with antenna pointing away from transmitter, demonstrate acquisition of this signal, including viewing of spectrum on the user device.

[ ] Demonstrate tuning range requirement of having a tunable range between 1550-1700MHz, with 1400-1700MHz desired (PER-2) is met.
[ ] Demonstrate ability to select a bandwidth between 2MHz and 10MHz (FUN-4). Confirm by observing the spectrum of the noise floor on the user device.
[ ] Self-interference test: Replace antenna with termination. Demonstrate absence of self-interference (designed such that self-generated radio interference is minimized and undetectable by the system - PER-4).
[ ] Demonstrate ability of user device to disconnect and reconnect without disrupting ongoing observing tasks (accessible by the wireless user at any time during operation - I/O-7).
[ ] Demonstrate teardown within 10 minutes (ME-2).
[ ] Demonstrate ability to offload acquisition data after teardown, including logs, averaged spectrum vs. time, signal strength vs. time, and relevant metadata (store data in non-volatile storage - I/O-3)

Customer will take possession of the instrument at the conclusion of the in-person demo. Within 1 week of the conclusion of the in-person demo:

[ ] Report documenting acceptance test and results (deliverable), including verification of:

8 Actions Table

Action #	Open/ Closed	Existing documentation	Plan to fix	Fixed
Make a physical drawing for this system so that we can flesh out the system.	Closed	**3.3 Mount Schematic**	Created a detailed design of the total system, including all measurements of components and wiring diagrams for subsystems.	Yes
Account for GPS antenna position so that it does not degrade the GPS signal strength	Closed	**3.3 Mount Schematic**	Included the position of the antenna on the exterior of the black box in the system design diagram, accounted for weather conditions.	Yes
Figure out how to tell which direction is North	Closed	**1.3 Component Connections**	Magnetometer module: Connected to the Raspberry Pi via I2C (SDA and SCL pins), providing magnetic field measurements to determine the heading and orientation of the rotor.	Yes

Solve issue with device used for execution of software (windows, linux or mac)	Closed	**2.3.2 Executable Creation**	To address the issue with the device used for the execution of the software, we will use Linux Ubuntu as the operating system. Ubuntu is free and open-source, which meets the requirement that all software used must be free and open-source.	Yes
Solve uml sequence on slide 16 of CDR	Closed	**2.2.2 Client-Server Communication (CSComm)**	Created a much more detailed sequence diagram.	Yes
Clear up the mount/computer controller on slide 17 of CDR	Closed	**2.2.9 Yaesu Rotator Interface**	Created a detailed sequence diagram of the algorithm.	Yes
Make mount / computer sequence diagram (create deeper level of UML)	Closed	**2.2.9 Yaesu Rotator Interface**	Created a detailed sequence diagram of the algorithm.	Yes
Clear up the receiver interface on slide 18 of CDR	Closed	**2.2.8 RSPDuo Interface**	Created a detailed sequence diagram of the algorithm/	Yes
Is the updating of the mount going to be continuous? Go more in depth with this. All of the commands for the slewing of the mount should be much more defined. Determine the accuracy of the slew rate of the mount.	Closed	**2.2.9 Yaesu Rotator Interface**	The continuous updating process of the mount is managed in the track_satellite function. It calculates the time needed to move the mount based on its max_elevation_speed and max_azimuth_speed, using this as a sleep time between commands. This ensures efficient position updates without overloading the mount with commands.	Yes
Make a diagram of what the system will look like once it is set up.	Closed	**3.3 Mount Schematic**	Created a detailed diagram with measurements of the system.	Yes

Figure out the torque that the tripod can support. (Physical testing)	Closed	**3.5.2 The Tripod**	Did research into issues with the mount subsystem tripod, developed solution to potential torque issue.	Yes
Make a list of tests to test the entire system(acceptance test plan). The flow diagram should reference that assignment. Put in a slide for each subsystem.	Open	**7 Acceptance Test Plan**	Made recommended changes after the first revision by Dr. Ellingson.	Yes
Ask Dr. Ellingson about power control from the computer.	Closed			
Talk to Mrs. Stover about the schedule.	Open			