L-Band Satellite Tracking and Characterization System

User Manual

Authors:

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

Date: Nov 15, 2023

[**Installation Guide: Initial Check-in 1**](#_heading=h.wolebk4g9sbw)

[**Installation Guide for System Hardware (Rotator side) 2**](#_heading=h.t5k6jlq3sn5f)

[**USB Usage Safety 7**](#_heading=h.9ogtkpkfkjcs)

[**Cabling and Wiring 7**](#_heading=h.qnj0hdk7wobu)

[**Receiver System Connection Guide 8**](#_heading=h.waq63wyz84w9)

[**System Requirements Before Installation 8**](#_heading=h.76pmboyl4p15)

[**Terminal App Installation Guide 9**](#_heading=h.gmbxip66imia)

[**Connecting to the system 10**](#_heading=h.djhye7ytbnr1)

[**Terminal App User Manual 10**](#_heading=h.qf1cm7s9m0q7)

[**GUI App Installation Guide 12**](#_heading=h.1zwuf4fk7gcx)

[**GUI App User manual 12**](#_heading=h.3usaztwukjs4)

[**Raspberry Pi Installation and Setup (Optional) 17**](#_heading=h.vs64v9kymxt4)

## Installation Guide: Initial Check-in

The first step of the installation guide is to determine that all components necessary for functionality are present, and correctly placed.

For internal system enclosure, ensure the presence of:

1. Battery
2. Rotator Control Box (Silver box placed next to battery)
3. Power Inverter
4. Raspberry Pi (onboard computer)
5. Rotator Control Box Interface Circuit Board
6. Raspberry Pi GPS module
7. RSPduo SDR
8. Antenna Filters

Additionally, ensure proper cabling for all devices. In order, these wires should be:

* Power cables for battery to inverter / fuse box / EPO
* Wire from Raspberry Pi to EPO / GPS module
* 7-Pin wire from rotator control box to Interface circuit board
* Micro-USB from Raspberry Pi to Interface circuit board (Feather M0)
* 2x Micro-USB from Raspberry Pi to LNAs
* Micro-USB from Raspberry Pi to SDR

The External wires for the enclosure should be:

* Two 8-Pin wires from enclosure system rotator control box to rotator motors (one Azimuth, one Elevation)
* Coaxial cables from receiver system to antennas.
* External GPS module connecting to Raspberry Pi.

## Installation Guide for System Hardware (Rotator side)

Step 1: Rotator Mounting Guide

1. Ensure all components shown below are gathered:



1x Yaesu G-5500DC dual rotator

2x Yaesu Masting Clamp

4x Yaesu rotator to clamp bolts + washers

2x Masting clamp security bolts + washers + nuts (larger)

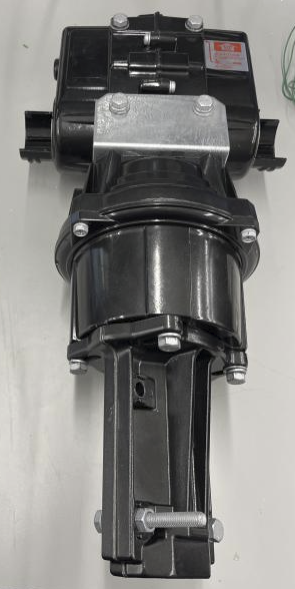
2x U-bolt clamps

1x Tripod

1. Using the two security bolts, connect the two masting clamps together using the hole at the bottom of the clamp.



1. Using the four provided bolts, connect the masting clamp to the rotator system:



1. After this, position the rotator on the top of the tripod, and tighten the bolts on the bottom of the rotator to secure the mount:



Step 2: Antenna Mounting Guide

1. Ensure all components shown below are gathered:



1x Antenna

2x PVC antenna mounting pipe w/ attached bracket

2x Antenna mounting plastic bolts

2x U-bolt clamps

NOTE: Before mounting the antenna system to the rotator, ensure that the rotator is currently positioned at 0, 0. This can be determined by the small notches around the rotation points. This will ensure that the antenna is positioned correctly when fastening the brackets and bolts, as well as ensuring correct calibration when testing / recording. Also, ensure that the antenna is mounted pointing as perfectly close to 0 degrees azimuth and elevation to maximize pointing accuracy.

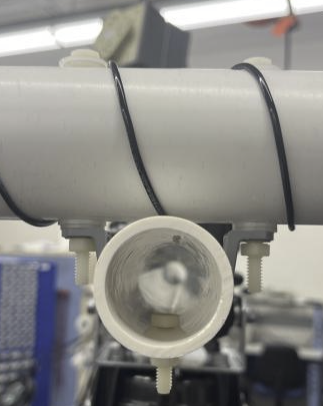
1. Slide the provided PVC pipe through the holes in the elevation rotator, positioning the clamp in the middle of the rotator (using the labeled markers as a guide):



1. Slide the U bolts over the PVC pipe until they are on the mounting brackets. Once in the grooves, tighten until the PVC pipe no longer swivels freely.



1. Verify that the PVC pipe is securely attached to the rotator, and ensure that the tripod has been extended and is stable.
2. Using the plastic bolts, washers and nuts, bolt the antenna to the associated clamp. This can be done by sliding the plastic bolt through the width of the antenna, and fastening it on the bottom of the clamp:



1. Tighten all bolts to secure the antenna to the rotator system, and verify all bolts are tight before testing.

## USB Usage Safety

• The provided USB drive is specifically formatted for the tracking system. Do not

replace it with another drive.

• Remove the USB drive only when the Raspberry Pi is powered off to prevent data

corruption.

• Always reinsert the USB drive into the same port it was removed from, which should be the top right usb port based on the way we set the raspberry pi. This could change if the user decides to set up the raspberry pi manually.

• For data offloading, safely eject the USB from your computer before reconnecting it to the Raspberry Pi.

The default viewing window for tracking is set to 8 hours but can be manually adjusted as needed with the setViewingWindow command.

## Cabling and Wiring

1. Take the bottom cable(labeled at the exit of the box), and connect it to the bottommost plug of the rotator.
2. Take the top cable, also labeled at the exit of the box, and plug it into the topmost port of the rotator.
3. Connect the coaxial cables to the SDR located in the black box. This box is inside of the larger box that contained the rotator.
4. Check the fuse located in the box and ensure that it is not burned. If it is, replace it.
5. Rotate the emergency stop switch to activate power to the system. It should be up, and able to be depressed again.
6. Ensure that the power to the rotator control box is on. This can be determined by checking if the lights to the two dials are lit. If they are not lit, there are three things that could be wrong.

* The power switch on the rotator box may not be flipped down. This is the red switch located to the top of the box. Flip this switch to the down position. If the lights are now on, the system is powered and you may skip the next step.
* The fuse in the control box could be burned. This is located in the control box, in the top right corner. If this fuse is burned, replace it.
* The rotator may have been knocked loose from the inverter. If this is the case, simply plug the box back in.

1. Ensure that everything is plugged into the inverter. USB cables should be plugged into the USB ports, and the rotator box should be plugged into the AC wall plug.
2. Place the top back on the box that contains the battery and connect to the system.

## Receiver System Connection Guide

The coaxial cables of the receiver subsystem should be connected as shown in the above diagram to ensure proper signal flowpath.

## System Requirements Before Installation

To ensure the optimal performance and compatibility of the Operations Software, your system should meet the following requirements:

• Operating System: Ubuntu 22.04.

• Python Version: Python 3.10. The preinstalled Python version on Ubuntu 22.04 is compatible; however, ensure it matches or exceeds the version mentioned.

Preliminary Setup Steps:

1. Update System: Ensure that your Ubuntu system is updated to the latest packages and security patches.

sudo apt update && sudo apt upgrade

2. Python Check: Verify the installed Python version using:

python3 --version

If the version is lower than the required, follow the instructions to install the appropriate Python version.

## Terminal App Installation Guide

Installation Guide for client.py (Client-Side Code)

This guide assumes you are using a system with Python installed. The following steps will guide you through setting up the Client.py script on your machine.

Step 1: Download the Client Script

• Navigate to the GitHub repository at https://github.com/samueltv250/MDE.

• Locate the Client.py script.

• Download the script directly or clone the repository using git:

git clone https://github.com/samueltv250/MDE.git

Step 2: Running the Client Script

• Ensure you are still in the directory containing client.py and your Python environment is activated.

• Run the script using Python:

python3 client.py

• If there are no errors, the client-side script should now be running, and you can begin using the commands outlined in the user manual to control the satellite tracking system.

## Connecting to the system

When the system is activated and you've downloaded the necessary Python scripts for control, search for a Wi-Fi network named ‘SatelliteTrackingSystem’ and connect to it. This network is specifically designated for system connections. Make sure to maintain a distance of at least 15 feet from the system during the connection process.

## Terminal App User Manual

Command Descriptions:

1. calibrate\_date\_time

• Synchronizes the Raspberry Pi clock with the client’s system.

• Response: Confirmation of time synchronization.

2. calibrate

• Calibrates the rotator to the correct starting position.

• Response: Confirmation of rotator calibration.

3. set\_single\_tuner or set\_dual\_tuner

• Selects the recording tuner mode. The default is single tuner at 10MHz.

• Response: Acknowledgment of the selected tuner mode.

4. setViewingWindow <start\_time> <end\_time>

• Optionally sets a specific time range for satellite tracking.

• Response: Confirmation of the viewing window setting.

5. setCord <latitude> <longitude>

• Updates the system with the new geographical coordinates.

• Response: Confirmation of updated coordinates.

6. add\_to\_queue

• Initiates schedule creation based on TLE data.

• Response: Update confirmation and the new schedule.

7. calibrate

• Initiates a 5-minute calibration routine for the rotator, setting its maximum and minimum positions for accurate satellite tracking.

• Response: Confirmation of calibration completion.

• Description: This command is essential for the initial setup and any time the system is moved to a new location. During calibration, the rotator will determine its full range of motion, which is critical for precise tracking. It is advised not to interrupt the rotator during this process.

8. stop\_tracking

• Stops the ongoing satellite tracking.

• Response: Confirmation that tracking has stopped.

9. getMeta

• Retrieves system metadata, including used space, tracking status, recording status and directory contents.

• Response: Metadata information.

10. get <filename>

• Downloads a specific recording from the server.

• Response: File transfer or a “File not found” message.

11. shutdown or reboot

• Shuts down or reboots the Raspberry Pi.

• Response: Acknowledgment of command execution.

12. move <azimuth> <elevation>

• Commands the rotator to move to specific coordinates.

• Response: Confirmation of movement execution.

13. clear\_schedule

• Clears the existing satellite tracking schedule.

• Response: Confirmation that the schedule is cleared.

14. device\_get

• Provides information about connected SDR devices.

• Response: Details of connected devices.

How to Start Tracking:

1. Ensure that the Raspberry Pi’s time is accurate by entering calibrate\_date\_time.

2. Input calibrate to calibrate the rotator.

3. Select the tuner mode with set\_single\_tuner or set\_dual\_tuner.

4. (Optional) To change the default 8-hour viewing window, use setViewingWindow <start\_time> <end\_time>.

5. (Optional) If needed, update your location with setCord <latitude> <longitude>.

6. Input add\_to\_queue to create the tracking schedule. Ensure a TLE data file is present in the script’s directory.

7. Start the tracking process with start\_tracking.

8. record\_fixed <sat\_name> <total\_time\_in\_seconds\_to\_record> <frequency\_to\_record>

• Record without tracking for a fixed time and frequency.

• sat\_name is just used for the file name.

## GUI App Installation Guide

If you choose to use the user interface (Recommended):

* Download Application:
  + Navigate to the git repository below:
    - <https://github.com/Ethancd19/L-Band-satellite-tracking>
  + Locate the application file “L-Band Satellite Tracking and Characterization System”
  + Download file (It is recommended to download to Desktop for ease of use).

## GUI App User manual

Once you have installed the application, you can proceed with using the software.

Step 1

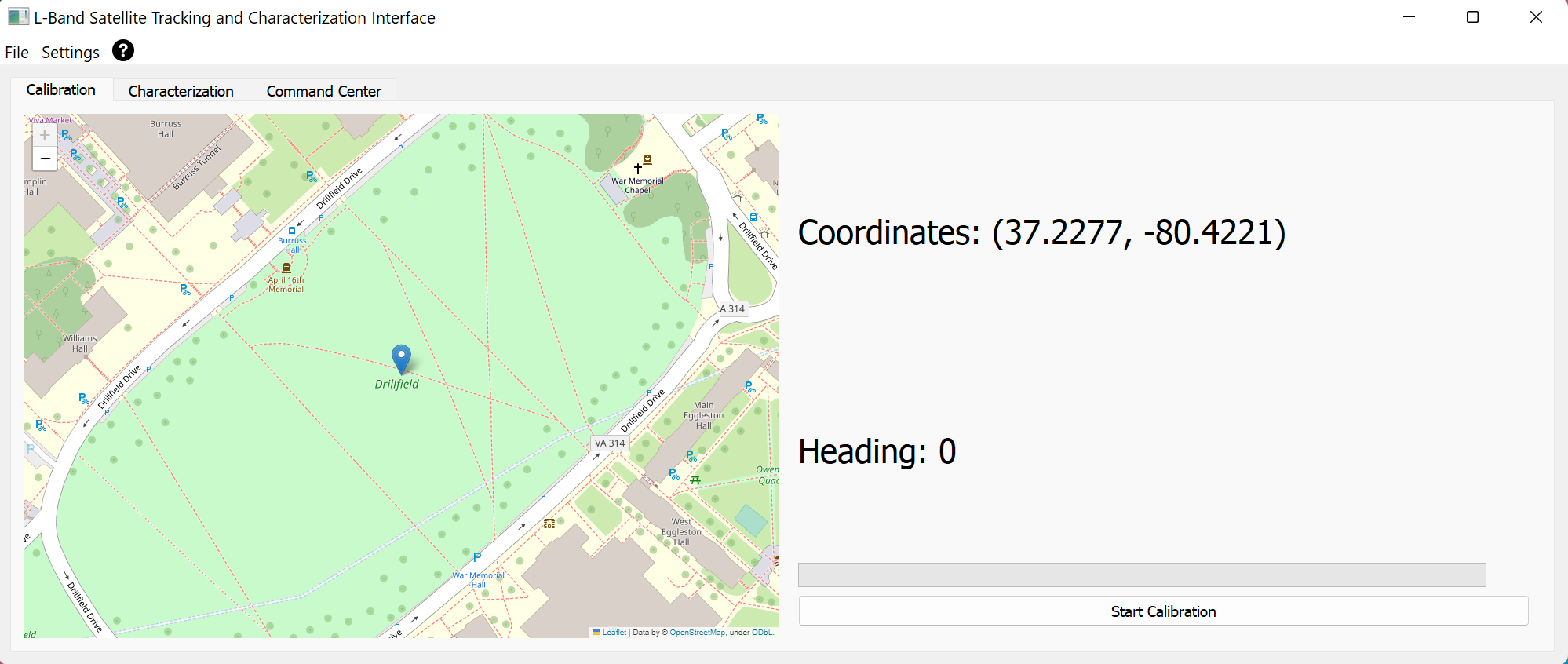
Open the application:

* Double click on the application Icon in the folder you downloaded it to.
* You don't need to worry about running and initializing Client.py as opening the application does this automatically

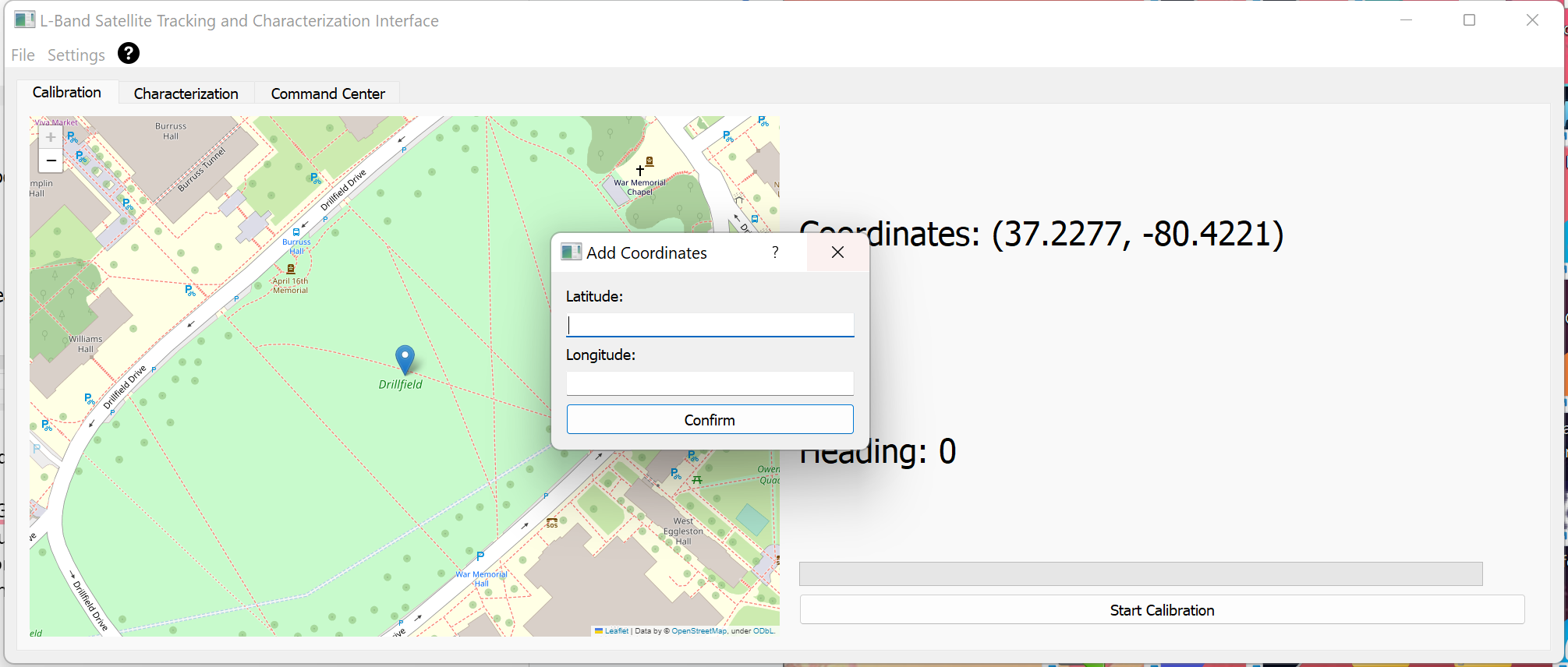
Step 2:

Calibrate the system:

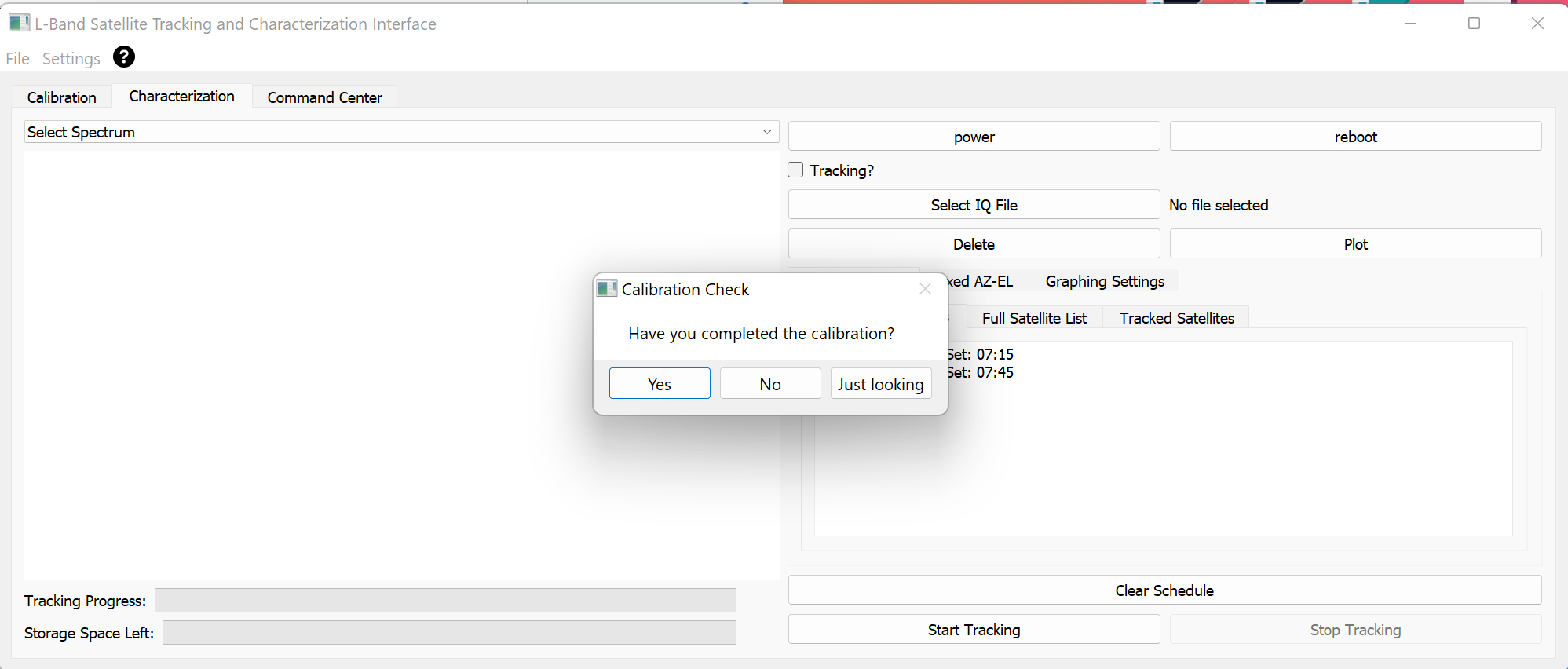
* Once you run the application it will take you to a landing page as shown below



* From here we can ensure that our GPS coordinates are correct, as well as if we are facing north.
  + If you have an issue with picking up GPS signal, and you know the coordinates of where you are you can manually input your coordinates by going to:
    - Settings -> Manually Add Coordinates
    - this will open a dialogue asking you to enter your latitude and longitude as shown below



* Once you have ensured that your location information is correct, you can start calibration. By pressing the “start Calibration” button.
  + This will initialize calibration and also visualize the progress of the calibration so you know when it is almost done
* When leaving the calibration tab, you will see a pop-up as shown below, ensuring that you have completed the calibration before you move onto anything else.



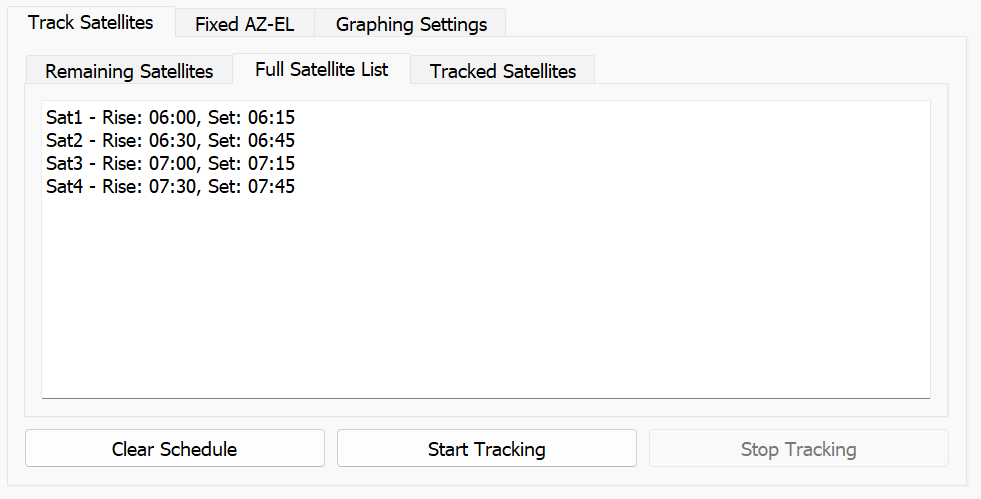
Step 3:

This section will be split up into two subsections: Scheduled Satellite Tracking and Fixed AZ-EL Recording.

Scheduled Tracking:

This mode is used if a user has a predetermined list of TLEs for satellites they would like to track throughout the day.

* The user should import a list of TLEs using one of the following methods: File browser or dragging and dropping.
  + To use the Browser, navigate to file -> open -> open TLE and select the TLE file you would like to import
  + To use drag and drop, navigate to the location of your TLE file and simply drag the file into the application window
  + The file imported should be named “satellites.tle” otherwise it will cause an error in the system.
* once the file is imported there will be a momentary pause, while the back end creates the schedule based on the imported TLEs.
* To view the schedule that has been created for you can view the Track Satellites Tab on the right side of the application.
  + In this tab you will see three subtabs, Remaining Satellites, Full Satellite List and Tracked Satellites. This makes it easy for the user to keep track of the satellite tracking queue.

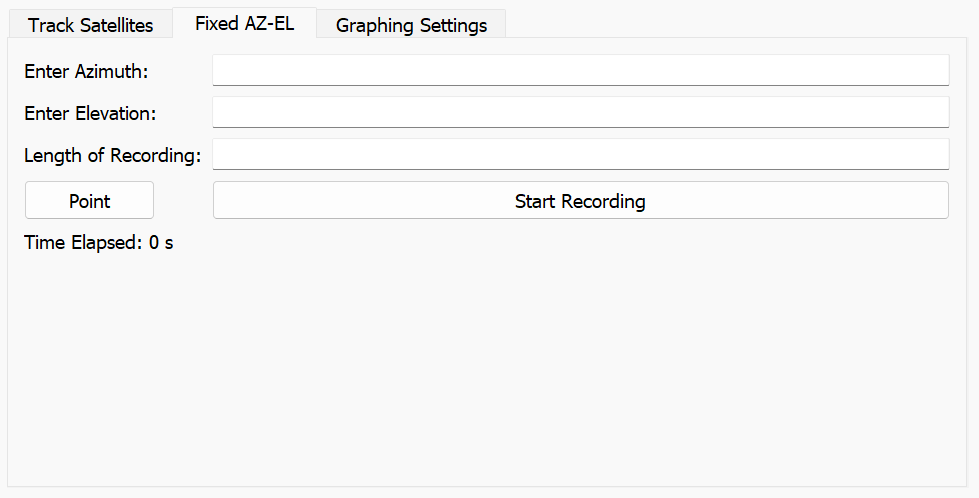


* Once the queue has been populated with the satellites from your initial List of TLEs, you can the tracking sequence by clicking on the “Start Tracking” button and if for any reason you need to stop tracking you can stop the sequence by clicking the “Stop Tracking” button.
* If for any reason you need to get rid of the schedule (I.E. You’ve imported the Wrong TLEs), you can click Clear schedule and this will not only clear the visual queue, but if connected, it will also clear the queue on the physical systems end.

Fixed AZ-EL (Azimuth-Elevation) Recording

This mode should be used if the user has a specific azimuth and elevation that they would like to point at statically and record data in that location

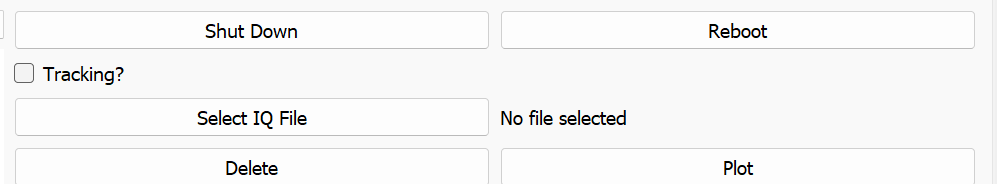
* Navigate to the Fixed AZ-EL tab, located next to the Track Satellites Tab.
* From here you can enter an Azimuth and Elevation, and optionally a length of recording.
  + Note: this can be used for testing as well by just inserting an azimuth and elevation and clicking point.
* Once the “Start Recording” button is clicked, a timer will start showing you how much time has elapsed since the start of this static recording.



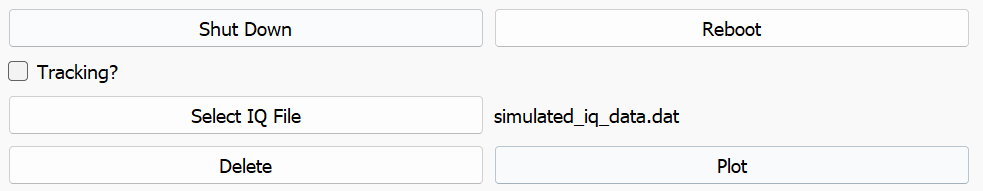
Step 4:

Graphing:

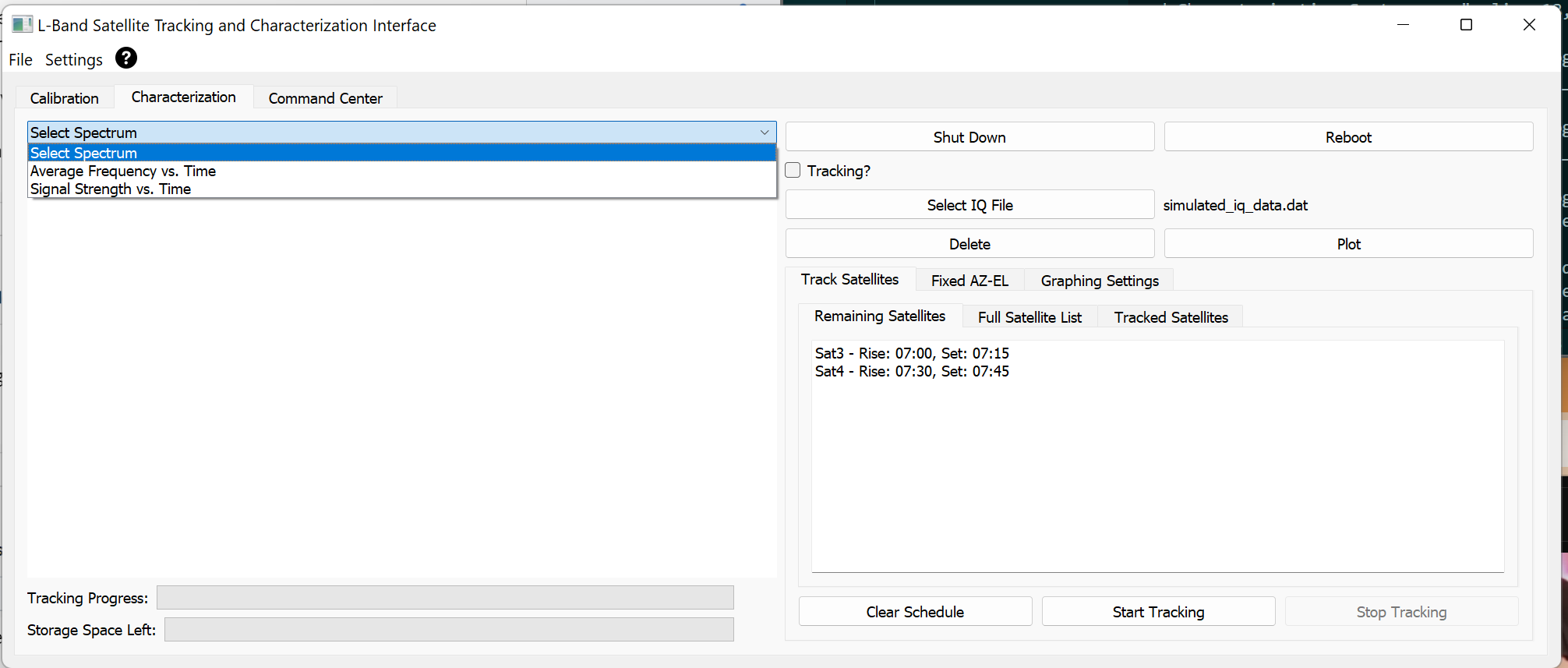
* Once you have recorded your data, or while you are recording data. You can again Navigate to File -> Open -> Open IQ . From here, you can either choose to select a file from the system or if you have already offloaded the files via USB.
  + If you choose to select a file off of the system, a list of files will be displayed, and you will have to type out the directory of the file, which will then start a file transfer.
    - Warning: Offloading wirelessly, is quite slow, so offloading via USB is highly recommended
* Once you have the offloaded IQ Data, You can then navigate to the Select IQ File button, shown below



* Once a file is selected the text field to the right of the button will be populated with the name of the selected file.

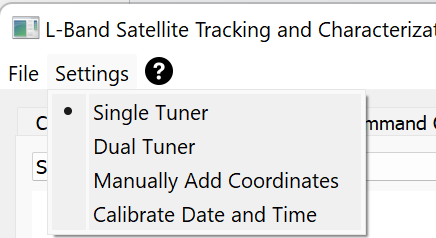


* Once you have selected a file you should then select the type of plot you would like to display, from here you will have two options: Averaged Spectrum Vs. Time or Signal Strength Vs time



* Once you have selected the graph you would like to plot, proceed with Clicking the “Plot” button to the right of the screen. This will then populate the blank space underneath the dropdown box you just selected your plot from with the plot of your choice.
* If you would like to clear the displayed graph you can click on the “Delete” button

Miscellaneous Settings:



* Single Tuner mode Vs Dual Tuner Mode
  + In the settings menu in the top left you can choose to swap the system between Single Tuner and Dual Tuner Mode
* Calibrate Date and Time
  + If for some reason the Raspberry Pi has faced some sort of issue in regards to its date and time, clicking this button will set the Pis time to the time on the computer

System Power Options



* Shut Down
  + When the shutdown button is clicked, a pop-up will display asking the user to confirm that they want to shut the system down.
  + When the user clicks yes, a second pop-up is displayed explaining that if you shut the system down, then there is no way for you to turn it back on from the software side and you have to manually turn the system back on.
* Reboot
  + When the user clicks on Reboot, a similar pop-up will display, asking the user to confirm they want to reboot the system.
  + Note: There is no second pop-up here as the system will reboot itself.

## Raspberry Pi Installation and Setup (Optional)

(Optional: The Raspberry Pi image, complete with all necessary settings and code to fully function, is provided to the client pre-installed on the device, along with a backup. This installation guide is in case the client wishes to set up the raspberry pi independently or if the client wishes to change the USB drive that is being used.)

All Raspberry Pi images are available in the following google drive link: <https://drive.google.com/drive/folders/1JRJSMsTya5UHqm25miUP2O2MPQko55_x?usp=sharing>

If the user opts for manual setup of the Raspberry Pi, ensure that the provided Raspberry Pi image named "sdr\_image" is used. This image has SDRplay preloaded for subsequent steps. Should the user choose to reinstall the latest backup that includes all preinstalled applications, they must install the image called "latest\_backup\_complete". It is important to note that the Raspberry Pi image should not need to be altered unless the client decides to use a different USB than the one provided, or if the data becomes corrupted in any way.

Preparing the Raspberry Pi Environment

1. Clone the project repository. cd Desktop && git clone <https://github.com/samueltv250/MDE.git>
2. Install required Python packages. pip install --break-system-packages skyfield && pip install --break-system-packages timezonefinder && pip install --break-system-packages pyserial && pip install --break-system-packages adafruit-circuitpython-gps

Configuring Raspberry Pi as a Hotspot with Static IP Address

[Update and Upgrade DietPi](https://github.com/samueltv250/MDE#update-and-upgrade-dietpi):

sudo apt-get update && sudo apt-get upgrade -y

[Install Exfatprogs, Hostapd and Dnsmasq](https://github.com/samueltv250/MDE#install-exfatprogs-hostapd-and-dnsmasq):

sudo apt-get install hostapd dnsmasq -y sudo apt-get install exfatprogs

[Stop Services to Configure Them](https://github.com/samueltv250/MDE#stop-services-to-configure-them)

sudo systemctl stop hostapd && sudo systemctl stop dnsmasq

[Configure Hostapd](https://github.com/samueltv250/MDE#configure-hostapd)

1. Edit the configuration file. sudo nano /etc/hostapd/hostapd.conf Add the following lines to the file: interface=wlan0 driver=nl80211 ssid=SatelliteTrackingSystem hw\_mode=g channel=7 wmm\_enabled=0 auth\_algs=1 wpa=2 wpa\_passphrase=12345678 wpa\_key\_mgmt=WPA-PSK wpa\_pairwise=TKIP rsn\_pairwise=CCMP
2. Point Hostapd to the Configuration File. sudo nano /etc/default/hostapd Replace the #DAEMON\_CONF line with: DAEMON\_CONF="/etc/hostapd/hostapd.conf"

[Configure Dnsmasq](https://github.com/samueltv250/MDE#configure-dnsmasq)

1. Rename the original configuration file. sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
2. Create a new configuration file. sudo nano /etc/dnsmasq.conf Add the following lines to the file: interface=wlan0 dhcp-range=192.168.220.10,192.168.220.50,255.255.255.0,24h

[Set Static IP Address](https://github.com/samueltv250/MDE#set-static-ip-address)

1. Create a systemd service file. sudo nano /etc/systemd/system/static-ip-wlan0.service Add the following lines to the file: [Unit] Description=Set static IP address for wlan0 Wants=network.target Before=network.target  
   [Service] Type=oneshot ExecStart=/sbin/ip addr add 192.168.220.1/24 dev wlan0 ExecStart=/sbin/ip link set wlan0 up RemainAfterExit=yes  
   [Install] WantedBy=multi-user.target
2. Enable and start the service. sudo systemctl daemon-reload && sudo systemctl enable static-ip-wlan0 && sudo systemctl start static-ip-wlan0

[Start Hostapd and Dnsmasq](https://github.com/samueltv250/MDE#start-hostapd-and-dnsmasq)

sudo systemctl unmask hostapd && sudo systemctl enable hostapd && sudo systemctl start hostapd && sudo systemctl enable dnsmasq && sudo systemctl start dnsmasq

[Reboot the Raspberry Pi](https://github.com/samueltv250/MDE#reboot-the-raspberry-pi)

sudo reboot

[Improve Wireless Configuration](https://github.com/samueltv250/MDE#improve-wireless-configuration)

1. Call sudo dietpi-config.
2. Enable 802.11n/ac/ax.
3. Change Frequency to 5 GHz.

[Enabling slave.py at Boot](https://github.com/samueltv250/MDE#enabling-slavepy-at-boot)

1. Create a New Service File. sudo nano /etc/systemd/system/slave.service Add the following lines to the file: [Unit] Description=Python Script Service After=network.target  
   [Service] Type=simple User=dietpi WorkingDirectory=/home/dietpi/Desktop/MDE ExecStart=/usr/bin/python3 /home/dietpi/Desktop/MDE/slave.py Restart=on-failure  
   [Install] WantedBy=multi-user.target
2. Reload the System Daemon. sudo systemctl daemon-reload
3. Enable the Service. sudo systemctl enable slave.service
4. Start the Service. sudo systemctl start slave.service
5. Check the Service Status. sudo systemctl status slave.service

[Instructions to mount a USB drive on boot with the dietpi user as the owner:](https://github.com/samueltv250/MDE#instructions-to-mount-a-usb-drive-on-boot-with-the-dietpi-user-as-the-owner)

1. Find the UUID of the USB Drive:
   * Run the blkid command to find the UUID of your USB drive:sudo blkid
   * Look for the line that corresponds to your USB drive (/dev/sda1 in this case) and note the UUID.
2. Create a Mount Point:
   * Choose a location for your mount point, such as /mnt/usbdrive, and create it:sudo mkdir /mnt/usbdrive
3. Edit the /etc/fstab File:
   * Open the /etc/fstab file in a text editor with root privileges:sudo nano /etc/fstab
   * Add the following line at the end of the file:UUID=your-uuid /mnt/usbdrive exfat uid=1000,gid=1000 0 2
   * Replace your-uuid with the UUID you noted earlier. If your USB drive is not exfat, replace exfat with the actual filesystem type (e.g., vfat for FAT32, ntfs-3g for NTFS).
4. Mount the Drive:
   * Now you can mount all drives with:sudo mount -a
   * This will apply all the new configurations set in the /etc/fstab file.
5. Set Ownership:
   * Change the ownership of the mount point to dietpi:
   * sudo chown dietpi:dietpi /mnt/usbdrive
   * This step is only necessary once, as the fstab entry will ensure the correct ownership on boot.
6. Test the Configuration:
   * Reboot your Raspberry Pi to ensure the drive mounts automatically with the correct ownership:sudo reboot
   * After rebooting, verify that the drive is mounted and the ownership is correct with:ls -l /mnt/usbdrive
7. Troubleshooting:
   * If you encounter any issues, you can revert the changes in /etc/fstab by commenting out the added line and rebooting the system. The backend will continue to operate without a drive but it will be limited to wireless offloading and less space.