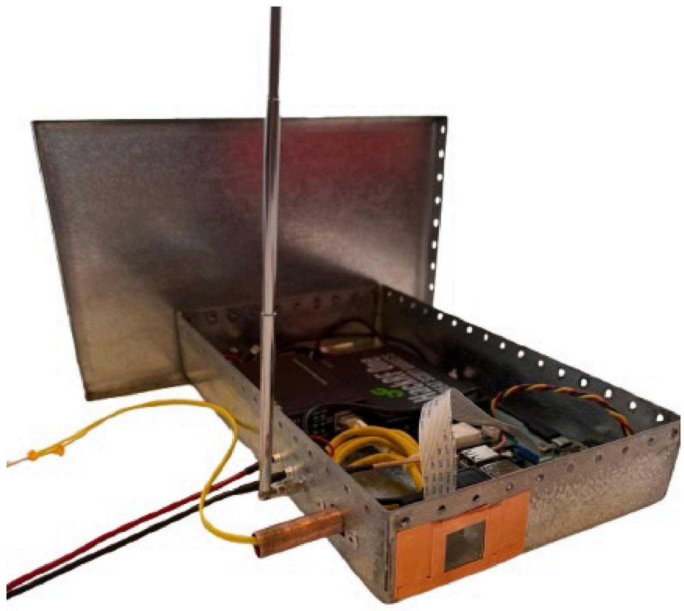
Radio Frequency Noise Monitor

for the Dominion Radio Astrophysical Observatory



By,
Felipe Garavelli, Sasan Ghasaei
William Gibbs, Mahdi Shakouri

Project Sponsors:
Dr. Mark Halpern, Dr. Gary Hinshaw
Mr. Parham Zarei

Project 2510
Engineering Physics 459
The University of British Columbia
8 April, 2025

Executive Summary

The Experimental Cosmology Lab at UBC launched this project to investigate whether vehicles near the CHIME radio telescope at DRAO were generating radio frequency interference (RFI) that could compromise its extremely sensitive astronomical observations.

We developed a fully autonomous, electromagnetically silent detection system that combines a motion-triggered camera with a software-defined radio to detect vehicles and capture their spectral footprint. The system is housed in a custom shielded enclosure (>100 dB attenuation in CHIME's band), communicates via fiber optics, and can operate unattended in harsh environments.

Testing confirmed that vehicles do emit detectable RFI in the CHIME band, and that our system can reliably identify and record these events without introducing additional interference.

Recommendations: (1) Mount conductive glass to the system, (2) Improve the system's frequency detection capabilities.

Table of Contents

Table of Figures and Tables

1 Introduction

The Canadian Hydrogen Intensity Mapping Experiment (CHIME), located at the Dominion Radio Astrophysical Observatory (DRAO) in British Columbia, is one of the most sensitive radio telescopes in the world. As part of the global effort to deepen our understanding of the universe, CHIME collects cosmological signals in the 400–800 MHz frequency band with such precision that even the faintest sources of terrestrial radio frequency interference (RFI) can significantly degrade the quality of the data [1]. To address this challenge, our project, sponsored by the Experimental Cosmology Lab at the University of British Columbia (UBC), aims to develop a system capable of detecting and correlating vehicle presence near the CHIME site. This solution will help the cosmology lab verify the source of RFI they are encountering, so later they can improve the quality of their data.

1.1 Sponsor Context and Motivation

The Experimental Cosmology Lab at the University of British Columbia is focused on building instruments to measure the formation history of the Universe. In particular, the lab is interested in measuring the properties of the Cosmic Microwave Background (CMB) and in measuring the star formation history (SFH) of the Universe [2]. Run by Prof. Mark Halpern and Prof. Gary Hinshaw, the Experimental Cosmology Lab is a leading collaborator in CHIME. Specifically, our immediate project sponsors are Prof. Mark Halpern and Prof. Gary Hinshaw, and their PhD candidate, Mr. Parham Zarei. Their support for this project is rooted in a practical need to mitigate local RFI sources, especially those linked to human activity on and around the observatory premises.

1.2 Background and Significance

CHIME is a state-of-the-art telescope, known for mapping the large-scale structure of the universe using hydrogen line intensity measurements. Its innovative design has enabled it to detect extragalactic Fast Radio Bursts (FRBs) and trace the universe's expansion history. However, the same sensitivity that allows CHIME to detect signals a thousand times fainter than a mobile phone broadcasting from the Moon [3] also makes it susceptible to contamination from terrestrial RFI sources. Vehicle traffic near the observatory, especially along the service and access road, has been identified as a potential source of interference. The lack of an automated system to track and correlate such events limits CHIME's ability to filter or correct for this noise in post-processing, motivating the need for a solution.

Figure 1. The CHIME Telescope. Left: wire-mesh semi-cylindrical reflectors;
Right: cloverleaf-shaped antennas at the focal line.

While shielding techniques and post-processing algorithms offer partial mitigation, real-time detection of RFI sources, especially mobile and transient ones such as cars, is both more effective and scalable. This project, therefore, fills a critical gap in existing operational protocols by introducing an autonomous vehicle detection and RFI logging system. Comparable efforts in radio observatories worldwide tend to rely on more static or manual means of monitoring local interference, making this approach a potentially valuable contribution to the field.

1.3 Problem Statement and Project Objectives

With CHIME capturing signals with such high sensitivity as described in the last section, ensuring minimal disruption from terrestrial sources is essential for maintaining the accuracy and reliability of CHIME's scientific measurements.

Figure 2: CHIME location with respect to nearby road

To address this, the objectives of the project are to determine problematic sources of RFI and to design and implement a fully autonomous system that detects the presence of vehicles near CHIME, and then correlate these events with measurements of their radio spectrum in the 400-800 MHz range. The system is required to operate continuously and reliably in remote and harsh environmental conditions without contributing any additional RFI [2]. Key design features include real-time triggering based on motion detection, spectrum acquisition using a software-defined radio (SDR), and storing data remotely for subsequent analysis.

1.4 Scope and Limitations

The scope of the project encompasses the development of a hardware-software system comprising a camera-based triggering module, integration of an SDR for spectrum capture, and the design of a weather-resistant, RFI-shielded enclosure for outdoor deployment capable of providing 100dB of isolation and surviving -25°C to 50°C. The solution also includes algorithms for frequency sweeping and spectral stitching to overcome the SDR's limited 20 MHz instantaneous bandwidth.

However, several limitations must be acknowledged. The reliance on SDR-based spectrum capture imposes temporal constraints, as full-spectrum data acquisition cannot occur instantaneously. Additionally, the system must endure challenging environmental conditions, including wide temperature fluctuations and exposure to precipitation, which places stringent requirements on the enclosure.

2 Detection System

2.1 Overview

Before developing a full detection system, we first verified that vehicles do emit detectable RFI within CHIME's operational band by using a low-cost Software Defined Radio (SDR) to observe spectral differences during vehicle pass-by events. This preliminary validation confirmed the hypothesis posed by our sponsors and justified further system development.

Our approach utilizes two parallel detection mechanisms:

1. **Vehicle detection** using a Raspberry Pi camera and classical computer vision techniques for triggering event recording based on motion.
2. **Spectral detection** using an SDR to record the electromagnetic spectrum coinciding with the presence of vehicles.

Together, these systems provide multimodal detection, allowing us to correlate motion-based triggers with a spectral reading.

2.2 Vehicle Detection

2.2.1 Approaches Considered

At the beginning of the project, several commonly used vehicle detection techniques were evaluated for feasibility:

- **Infrared Beam Triggering**
  Infrared (IR) detection was dismissed early in the design process due to the strict RFI constraints of the CHIME site.
- **Pneumatic Compression Tubes**
  The need to open and reseal the shielded enclosure for each maintenance action would compromise the RF integrity of the system.
- **Computer Vision Techniques**
  Computer vision offers a flexible detection approach. It can be fully contained within the shielded box and requires no external signal emission. We considered both classical image processing and machine learning-based methods. Due to the computational constraints of the Raspberry Pi, we discarded the machine learning-based approach.

2.2.2 Approach Used

We decided to use classical image processing to detect motion.

The camera, mounted behind a shielded RF glass window, captures continuous footage. A circular buffer allows pre-event recording, so footage is saved from several seconds before motion is detected. The car-detection algorithm works by repeatedly comparing frames to the previous (with a delay), and finding the difference. If the difference is above a certain threshold, the device knows there is motion in the frame and will begin saving data. A visual representation of motion can be seen in Figure 3.

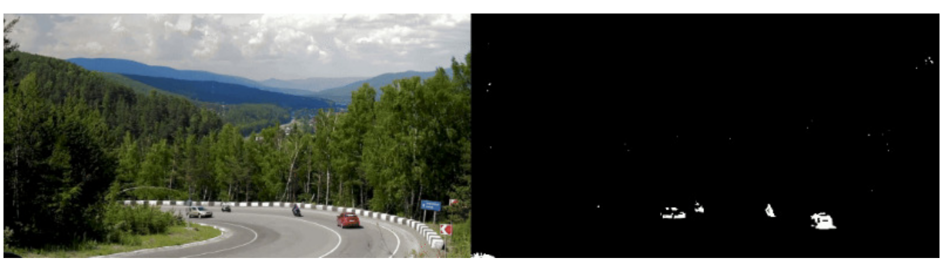


Figure 3: Cars are highlighted in white

2.3 Frequency Detection

2.3.1 Initial Detection of Car RFI

To first confirm our sponsors' hypothesis that vehicles produce RFI, we utilized an RTL Software Defined Radio (SDR). We conducted sweeps through several 100 MHz ranges.

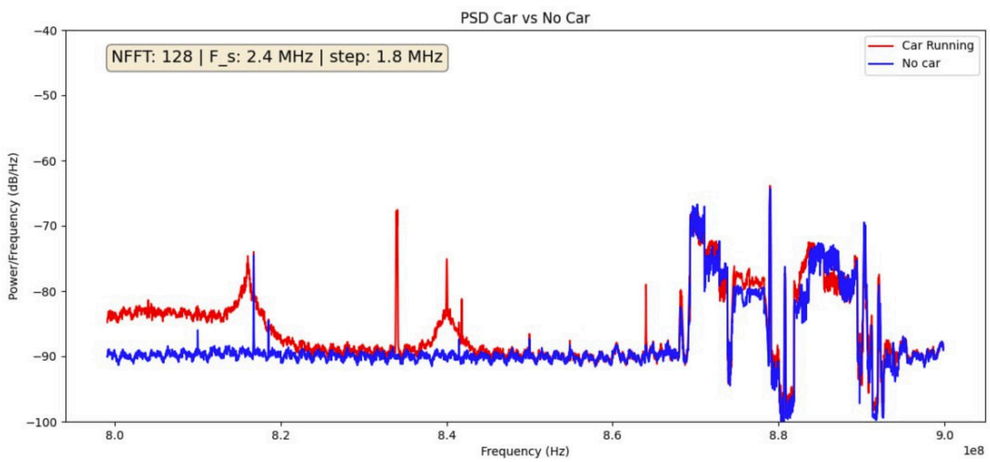


Figure 4: RFI signature of a car running w.r.t. the background RFI at the location

We can observe a car's RFI clearly in Figure 4, where the car is emitting a significant amount of RFI in the 800-850 MHz range. These are likely from spark plugs and onboard electronics.

2.3.2 Spectrum Detection Approaches Considered

We considered the following RF detection methods:

- **Commercial Spectrum Analyzers**
  They were ruled out because their integration with an autonomous system is impractical.
- **Custom FPGA-based Solution**
  This approach was abandoned due to design complexity, time constraints, and the maturity of existing SDR toolchains.
- **SDRs**
  We began testing with an RTL-SDR, a popular low-cost USB SDR, to validate whether vehicle RFI was detectable at all. While the RTL-SDR served well for

initial experiments, it lacked the bandwidth and dynamic range required for CHIME's operational frequencies.

Ultimately, we selected the HackRF One, which supports real-time capture and sweeping across the 400–800 MHz band with sufficient dynamic range and bandwidth control.

2.3.3 Frequency Sweeping

Sweeping was necessary due to the limited instantaneous bandwidth of our SDR, which was insufficient for capturing the full frequency range (400–800 MHz) used by the CHIME telescope. Without employing sweeping, the SDR would be unable to comprehensively measure and characterize the RFI emitted by vehicles across the entire relevant spectrum, thus limiting the system's effectiveness in protecting the telescope's sensitive observations.

To address this limitation, we divided the 400 MHz wide spectrum into 20 MHz segments. The SDR was tuned to each of these smaller segments, capturing RF data for each one. After collecting the measurements across all partitions, we used digital signal processing techniques to stitch them together, resulting in a continuous and spectral representation of vehicular RFI. This sweeping approach allowed complete coverage of the required bandwidth, ensuring accurate monitoring and analysis.

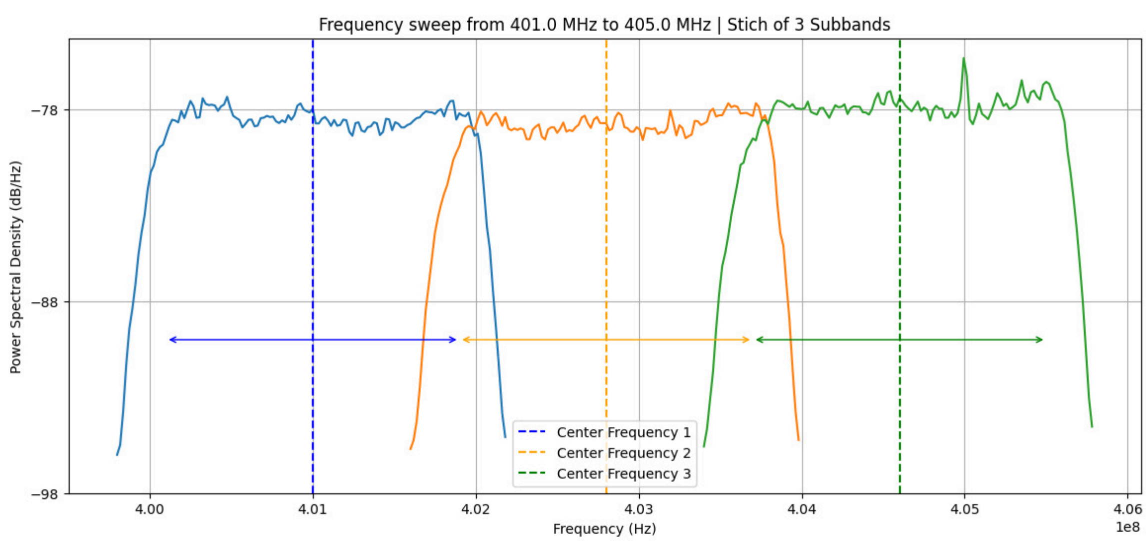


Figure 5: Sweeping through a larger bandwidth in three smaller sub-bands

Figure 5 illustrates the frequency sweep process by stitching together three separate sub-bands, each measured by the SDR around distinct center frequencies.

To do so, we select N distinct center frequencies that uniformly span our entire desired bandwidth, B.

$$f_{c,n} = f_{\text{start}} + \left(n - \frac{1}{2}\right) \cdot B, \quad n = 1, 2, 3$$

For each center frequency, the SDR collects signal data separately by a convolution of the frequency response of the signal X(f) and shifts the frequencies to the center frequency.

$$S_n(f) = \int_{f_{c,n}-\frac{B}{2}}^{f_{c,n}+\frac{B}{2}} X(f) H(f - f_{c,n}) \, df$$

Finally, the stitched signal is the union of all sub-bands

$$X_{\text{stitched}}(f) = S_1(f) \cup S_2(f) \cup S_3(f), \quad f_{\text{start}} \leq f \leq f_{\text{end}}$$

To quantify the strength of the signal at each frequency bin, we used a quantity known as Power Spectral Density, defined as

$$PSD(f) = 10 \cdot \log_{10} \left(\frac{|X_{\text{stitched}}(f)|^2}{\Delta f} \right)$$

where Δf is the width of our frequency bins selected and can be adjusted depending on the desired resolution of the output [4].

The HackRF One is capable of transmitting and receiving signals from 1MHz to 6GHz with a sample rate of 2-20 MHz. This sampling rate is almost 10 times faster than RTL-SDR, which is a huge advantage. (See architecture of HackRF in Appendix A)

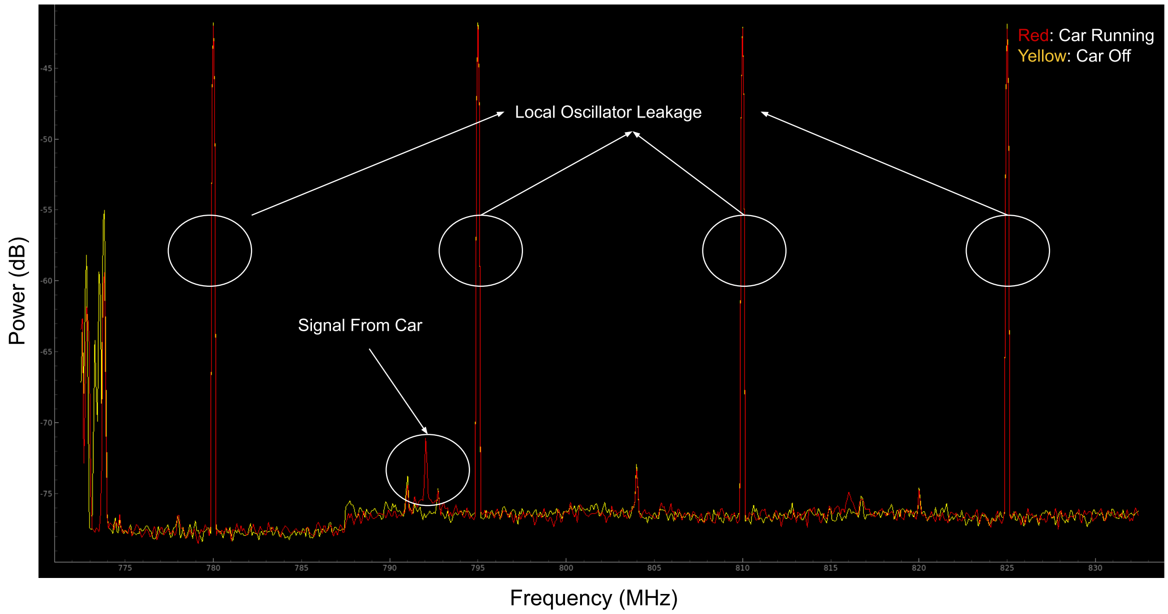


Figure 6: Detection of Car Frequencies with HackRF

2.3.4 Removal Of Local Oscillator Leakage

Local oscillator (LO) leakage happens when the LO signal (used for mixing RF down to baseband or vice versa) leaks into the signal path. This causes a spike at 0 Hz (DC) in the baseband spectrum or a spike at the LO frequency in the RF spectrum. The diagram below shows the significance of local oscillator leakage (LOL) when the gain of the SDRis set to zero with no antenna connected.

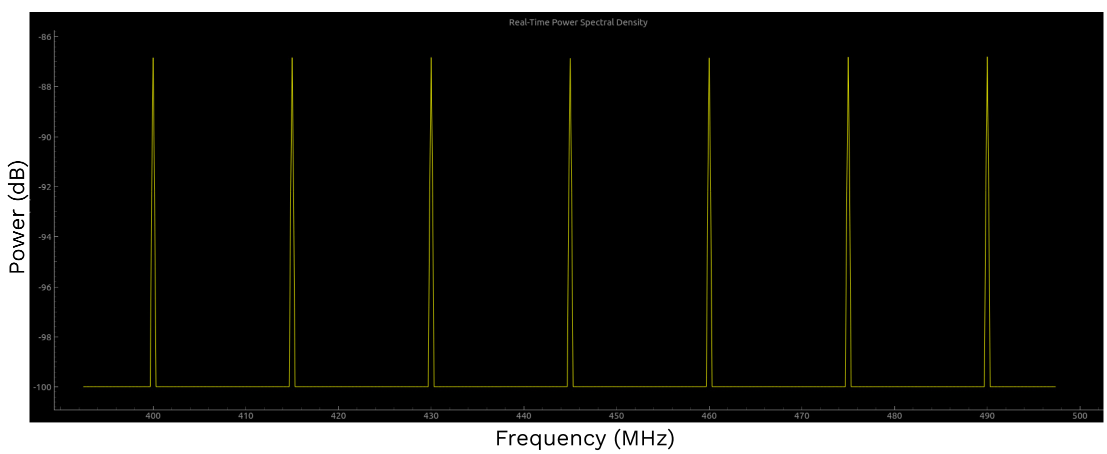


Figure 7: Local Oscillator Leakage present at Center Frequencies

To remove the LOL, we move the center frequency of each scan in a way that the center frequency lies in a desirable location. We let F_c be the center frequency of the current scan and F_c' be the center frequency of the next scan. F_s is the sampling frequency of the SDR, which for the HackRF is a maximum of 20 MHz. Because the baseband bandpass filter of the SDR is set to 75% by default, we remove 25% of the spectrum to avoid potential attenuations on either end. This means that the usable spectrum is reduced to frequencies in the range

$$[F_c - \frac{0.75}{2}F_s\ ,\ \ F_c + \frac{0.75}{2}F_s].$$

Furthermore, we assume that the 4 frequency bins at the center are corrupted with local oscillator leakage. Letting N_{FFT} be the number of frequency bins performed in Fast Fourier Transform, we know that our bin resolution is given by

$$bin\ width\ =\ \frac{F_s}{N_{FFT}}.$$

Therefore, the goal is to move F_c' such that it is aligned with the end of the previous scan, or equivalently, we have

$$F_c'\ +\ 2\ \times\ bin\ width\ =\ F_c + \frac{0.75}{2}F_s.$$

And since $F_c'\ =\ F_c\ +\ step\ frequency$ we have

$$step\ frequency\ =\ \frac{0.75}{2}F_s\ -\ 2\ \times\ bin\ width$$

The diagram below illustrates the entire process of removing LOL in more detail. The dotted blue line shows the location where F_c' is moved to. The shaded region shows the final usable section of the spectrum after removal of local oscillator leakage.

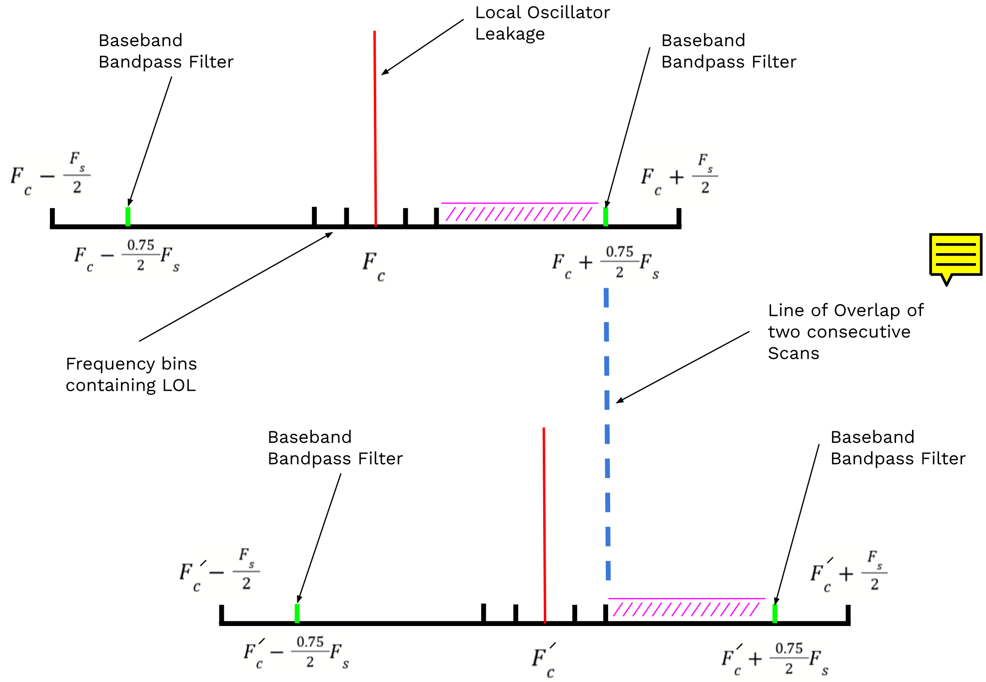


Figure 8: Diagram illustrating methodology for removing Local Oscillator Leakage

After removal of local oscillator leakage, the power spectral density for a sweep ranging from 400 MHz to 500 MHz with no antenna and gain is shown below:

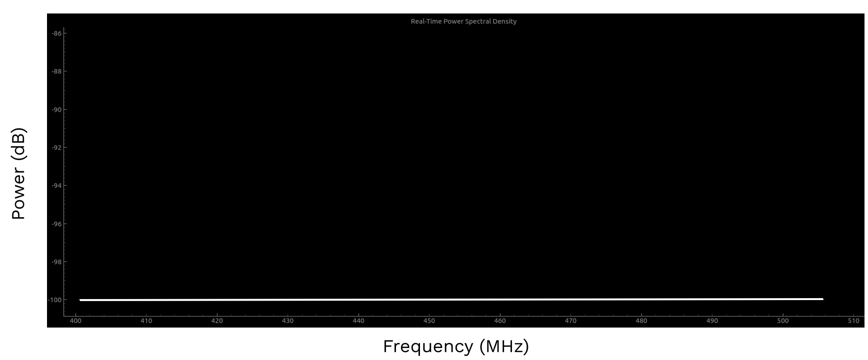


Figure 9: Local Oscillator Leakage removed

As shown in the diagram above, the DC spikes at center frequencies are removed, and the signal looks like a flat line at -100 dB.

One downside of this method is that the number of scans for any given range is almost double, and hence, the time of acquisition and processing of data is doubled too.

Data Acquisition and Processing with HackRF One for a 100 MHz Sweep		
Method	**Time of Acquisition (s)**	**Time of Processing (s)**
No Removal Of LOL	0.2	0.1
Removal of LOL	0.3	0.4

Table 1: Comparing acquisition and processing time for two methods of sweeping.

3 Physical Enclosure

3.1. Physical Enclosure Design

3.1.1 Overview

A key part of this project was the need to construct a shielded system that could operate in proximity to the CHIME telescope without contributing to RFI. The physical enclosure was therefore not just a housing but a critical component of the system's function and integrity. The box needed to suppress signals across a wide frequency band (400–800 MHz) to meet CHIME's stringent electromagnetic compatibility requirements. This required a rigorously designed and tested Faraday enclosure, built for long-term deployment under adverse environmental conditions.

3.1.2 Rationale for Shielding

DRAO operates under tight restrictions for electromagnetic emissions due to the extreme sensitivity of its instruments. Shielded boxes are commonly used at DRAO to isolate experimental or diagnostic equipment from the environment, allowing use of electronics near telescopes. These enclosures typically achieve this by functioning as Faraday cages, providing continuous conductive surfaces to attenuate incoming and outgoing electromagnetic radiation.

Our design followed this tradition but incorporated several unique features. Most notably, we introduced a lid to enable iterative testing and reconfiguration, whereas conventional boxes at the site are sealed permanently. This feature was essential during development, where repeated access was necessary to verify performance, troubleshoot failures, and update internal configurations. It will continue to be useful in the future, for the same reasons mentioned previously.

3.1.3 Mechanical Construction

The enclosure was constructed using 1 mm galvanized steel panels, chosen over aluminum for their superior mechanical robustness, excellent shielding properties, and ease of welding shut. Galvanized steel also improves corrosion resistance, essential for outdoor deployment. The lid was fastened to the box with 54 sheet metal screws, and conductive continuity was maintained by sanding off the zinc coating at contact interfaces. This ensured bare steel-to-steel contact at all joints to minimize resistive discontinuities.

Copper tape was used extensively at seams and potential leak points, including around the box lid and corners, to enhance conductivity and minimize RF leakage. A lead-based solder was used where necessary to strengthen electrical continuity in critical junctions such as the bent corners. Unlike traditional DRAO boxes, which use heavy electromagnetic interference (EMI) gaskets and are sealed indefinitely, our lid-based design permitted reopening the box without performance loss due to using sheet metal screws.

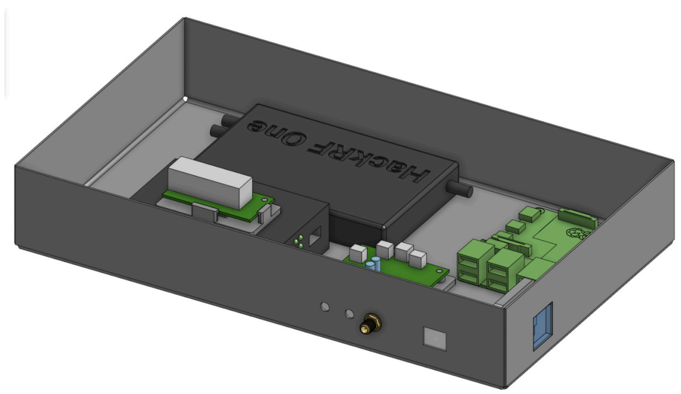


Figure 10: CAD design of the system without the lid

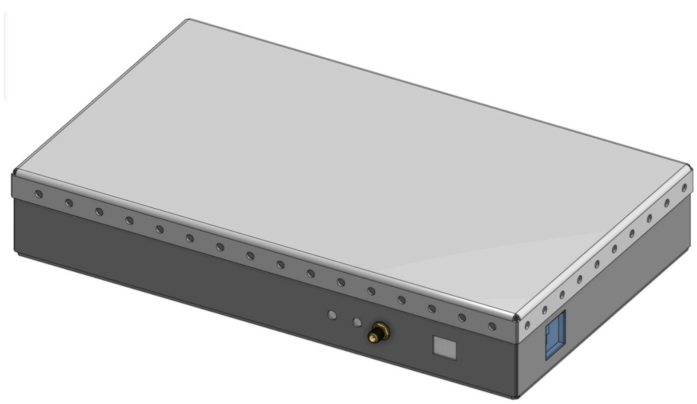


Figure 11: CAD design of the system with the lid

3.1.4 Internal Components

The final configuration of the box included:

- **Raspberry Pi 4**: For control, data logging, and sensor interface
- **HackRF One**: A software-defined radio (SDR) to capture spectral data
- **Raspberry Pi Camera with fisheye lens**: For visual motion detection
- **Media Converter**: For data transmission via optical fiber
- **Power Distribution Board (PDB)**: For regulated and filtered voltage delivery

The interior layout was guided by thermal management as well as by ease of assembly. This layout can be seen in the figure above.

3.1.5 Entry Panel

The shielded enclosure includes four entry points, each engineered to maintain electromagnetic integrity while allowing essential connectivity:

1. **Dual Power Entry Ports**
   Two openings are dedicated to inline power filters, which allow DC power to enter the enclosure without introducing electromagnetic noise. These filters are mounted directly onto the entry panel.
2. **SMA Connector Port**
   A shielded SMA connector is used to route signals to and from an external antenna while maintaining a secure RF-tight interface. The connector is mounted flush to the box wall and grounded to the enclosure.
3. **Fiber Optic Feedthrough**
   A machined copper waveguide tube houses the fiber optic line, designed with dimensions below the cutoff wavelength to attenuate RF leakage. This allows for reliable data transfer via optical link while preserving the shielding effectiveness of the enclosure. A typical LC-LC coupler would not work since the plastic casing would not provide enough attenuation.

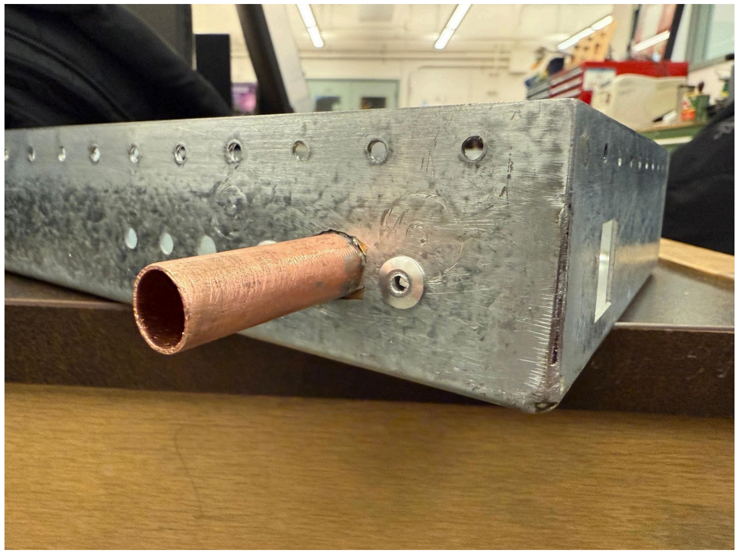


Figure 12: Fiber optic feedthrough tube

3.2. Shielding Performance Testing

3.2.1 Methodology

Shielding effectiveness was evaluated in collaboration with Parham Zarei using a calibrated vector network analyzer (VNA). The test setup included:

- A monopole antenna inside the box, connected to a VNA port configured as a signal source.
- An external omnidirectional antenna, connected to the second port of the VNA for reception.
- Measurement of signal strength with the box open and then closed and fully sealed with copper tape.

This setup allowed us to characterize attenuation across a wide range of frequencies by measuring the difference in received power with and without the shielding in place.

3.2.2 Results

With the box fully sealed using screws and copper tape, we measured:

- **100 dB of attenuation from 0–4 GHz**, which was the noise floor of the VNA.
- **70 dB of attenuation from 4–8 GHz**, showing some degradation in higher frequency suppression.

Given CHIME's operational band of 400–800 MHz, the system met and exceeded the required shielding performance, with 100 dB isolation in the relevant range. These results demonstrate the effectiveness of our shielding and that it won't interfere with CHIME. Some further revisions might be required for better performance in the 4-8 GHz range, which other telescopes in DRAO operate in.

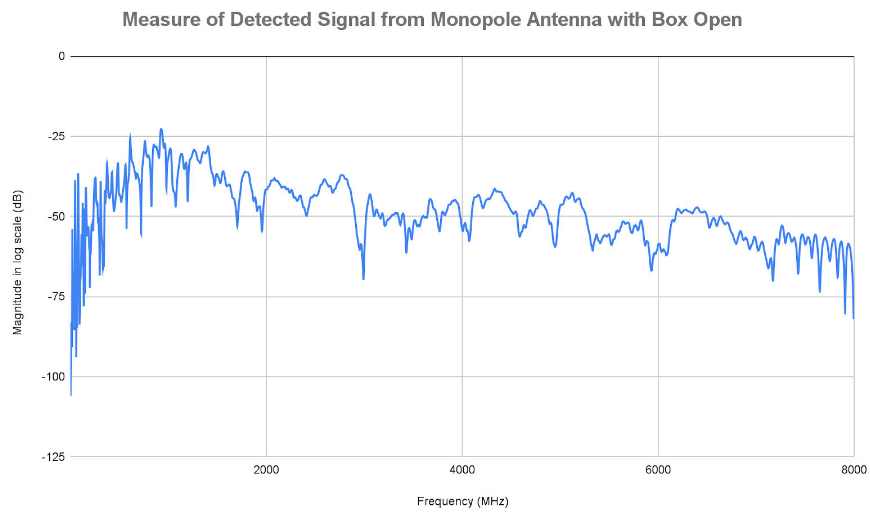


Figure 13: Detected Signal from Monopole Antenna with Box Open

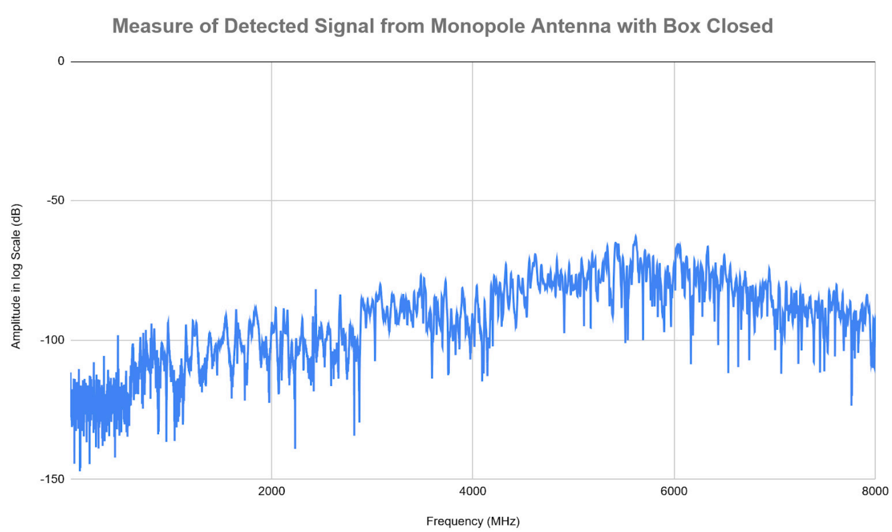


Figure 14: Detected Signal from Monopole Antenna with Box Closed

3.3. Thermal Management and Environmental Sensing

3.3.1 Temperature and Humidity Monitoring

To monitor internal environmental conditions, we integrated a DHT22 sensor wired to the Raspberry Pi. The sensor provides temperature and humidity data, logged continuously using the Adafruit-DHT library. Monitoring is essential not only for diagnostics but also for ensuring safe operating conditions for the SDR and media converter, both of which are sensitive to thermal extremes.

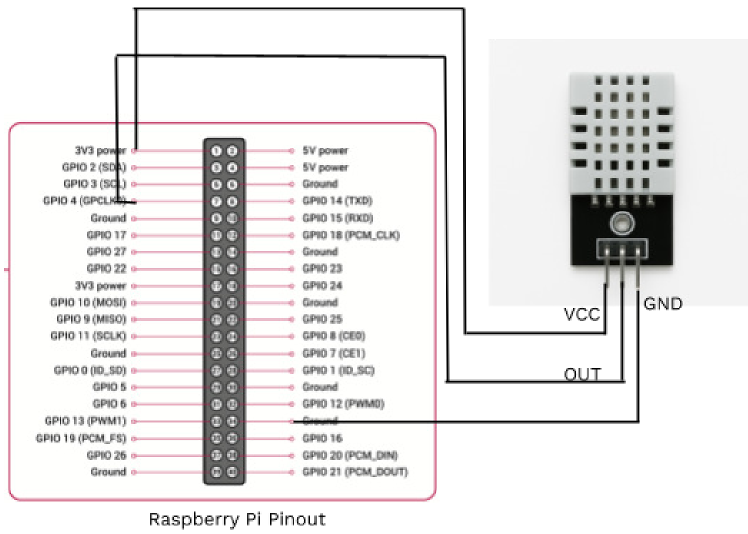


Figure 15: Wiring of Temperature Sensor to the Raspberry Pi

3.3.2 Thermal Stress Test

Before building the metal enclosure, we tested to see if all components would remain in an accepted operational temperature range. If the test were to fail, active cooling or ventilation on the box would need to be implemented into our design.

We evaluated heat buildup within an enclosure by replicating its thermal behavior using a sealed cardboard prototype. All components were powered and placed inside, and the box was sealed using electrical tape. We maximized Raspberry Pi CPU activity to simulate worst-case thermal conditions. Temperature readings were logged every 5 seconds using the DHT22 sensor.

Results showed a minor internal temperature increase, remaining well within safe operating limits for all components. This validated the box’s passive thermal dissipation capability since a metal-cased box would perform better in dissipating the heat. No ventilation or active cooling solution would need to be implemented for the box.

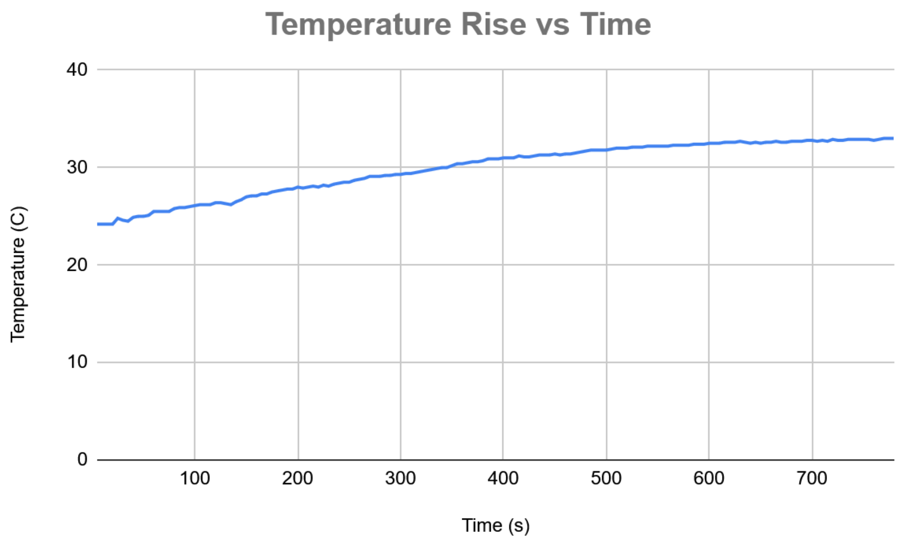


Figure 16: Temperature Rise Inside of the Cardboard Box During Testing

3.4. Power Supply and Protection

3.4.1 Power Source and Filtering

To avoid RFI emissions, we selected a linear power supply, which operates without the high-frequency switching present in most modern switch-mode power supplies. This was a critical decision in maintaining low electromagnetic emissions. Power is supplied through a standard XT-60 connector, with in-line filters (inserted before entry to the enclosure) to suppress conducted emissions.

3.4.2 Undervoltage Lockout (UVLO) Circuit

Due to our requirement of releasing no electromagnetic noise, we opted to use a linear power supply. Linear power supplies inherently produce less noise than the more common switching power supplies because they regulate voltage using a continuous, linear process, avoiding the high-frequency switching. For this reason alone, the choice was obvious. We put a switch on the power line, and attached the power (5V) and ground (0V) output from the power supply to a XT-60 connector.

When disconnected from the power line, our linear power supply slowly decays from 5V to 0V. This is a problem, as our electrical components (media converter and pi in particular) can be damaged by running off-spec voltages - this is known as “browning out”. Ideally, the switch will always be turned off first before the power supply is ever disconnected. However, this depends on a correct procedure, and is unreliable in many cases, such as:

- Power Outage
- Somebody on-site is not properly turning it off.

For this reason, we built an undervoltage lockout (UVLO) circuit to cancel this effect. This diagram shows the circuit:

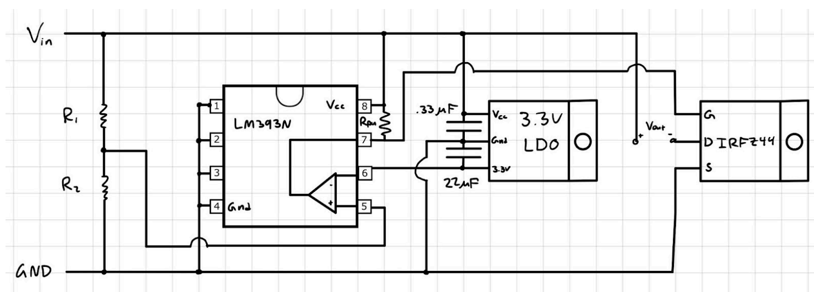


Figure 17: Circuit diagram for the UVLO

This circuit takes input from the power supply at the V_{in} and GND nodes. Essentially, this works by comparing the V_{in} with a stable reference voltage generated by our low-dropout regulator (referred to as LDO in the diagram) which consistently outputs $\sim 3.3V$. The threshold voltage for lockout is set by R_1 and R_2. We set these resistor values of this voltage divider such that the input to the comparator is lower than the reference at $4.7V$. When this condition is met, the comparator output is cut, and therefore the MOSFET gate is LOW, cutting the current to the components. This solves our problem, by giving a hard edge on the output. Here is a voltage scan illustrating this cut.

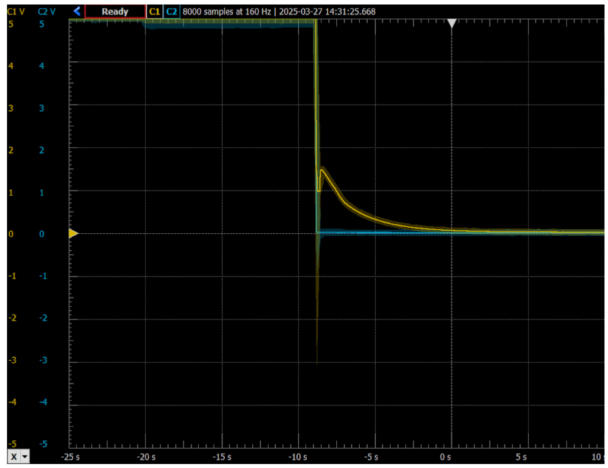


Figure 18: Power supply voltage (yellow) vs. UVLO Output Voltage (Blue)

3.5 Data Transfer

Given the need for complete RF shielding, all data communications from the box to the office on site (approximately 400m away) had to be optical. We used media converters to transform Ethernet data into optical data and vice versa. One media converter is placed inside the box, connected to optical cables that exit the enclosure. These cables are then connected to a second media converter outside the box, which interfaces with an external computer acting as a server. Through this setup, the server provides the Raspberry Pi with access to external networks and internet connectivity, all while maintaining the electromagnetic shielding of the system.

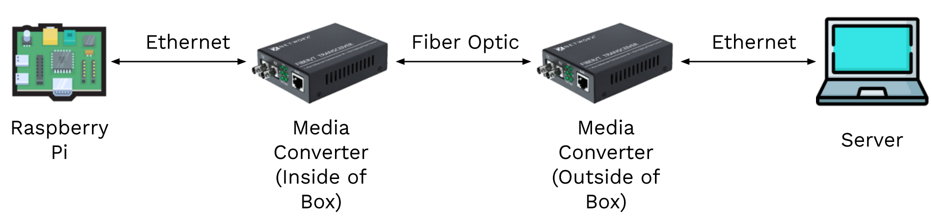


Figure 19: Diagram of Data Pipeline from the Raspberry Pi to the External Computer (Server)

4 Software System Architecture

4.1 Overview

The software system developed for this project was designed with modularity, robustness, and autonomy in mind. The overarching goal was to integrate multiple hardware components—including a Raspberry Pi, a HackRF SDR, and a fisheye camera—into a coherent software framework that could detect vehicle motion, capture and process capture and process frequency sweeps, and log and transfer data autonomously. To accomplish this, the software was built using a class-based architecture in Python with a state-machine-based execution model.

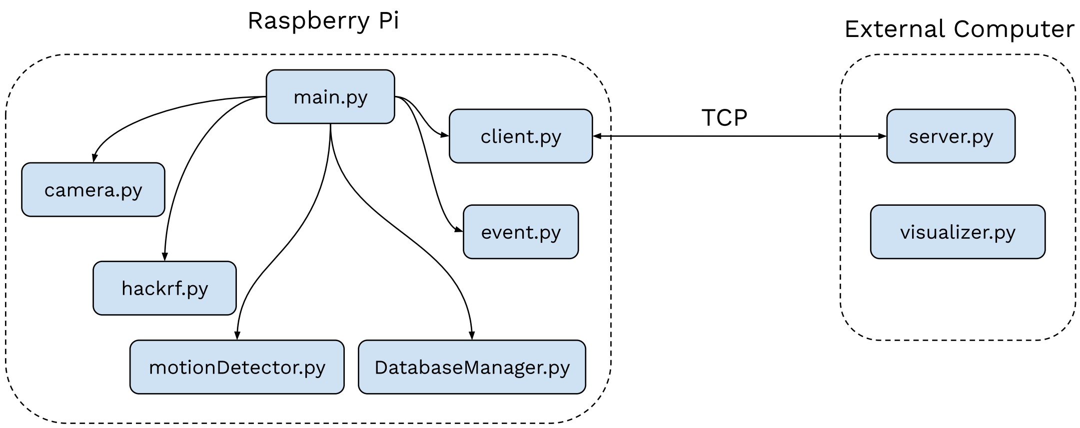


Figure 20: System Level Diagram of class-based Architecture in Python

4.2 Software Architecture and Class Structure

The codebase is composed of several interdependent classes that encapsulate distinct functionalities:

- **Camera:** Interfaces with the Raspberry Pi camera using the picamera2 library. It maintains a circular video buffer to retain footage preceding motion-triggered events.
- **HackRF**: Manages interaction with the HackRF SDR, performing spectrum sweeps and configuring SDR parameters (gain, frequency steps, etc.).
- **MotionDetection**: Implements the motion detection algorithm using real-time image differencing.
- **Event** and **Sweep**: Data structures to record event-level metadata (e.g., timestamp, environmental conditions) and SDR sweep-specific data, respectively.
- **DatabaseManager**: Interfaces with an SQLite database to persist and index events for efficient retrieval and logging.
- **Client**: Handles communication with an external server over TCP sockets using Python's socket module.

- **VirtualSensors**: Allows for hardware abstraction during local testing, enabling us to emulate Pi-specific peripherals on a general-purpose computer.

All file and directory operations are performed relative to a defined PROJECT_ROOT to avoid dependency on system-specific absolute paths. This decision was critical for ensuring reproducibility and portability between the Raspberry Pi (in deployment) and external server computers.

4.3 Logging and Configuration Management

Logging is implemented through Python's logging module. A dedicated logger labeled "DRAO" is instantiated and accessed across all files. The logging system prints console messages at WARNING level and stores log files at DEBUG level, ensuring that detailed diagnostics are persistently available without overwhelming the console during runtime. Logs are stored under PROJECT_ROOT/log_files.

Configuration parameters such as camera buffer size, HackRF gain and sampling rate, and networking port numbers are stored in a configurations.conf file. This facilitates rapid tuning and maintains consistent operational behavior across deployments.

4.4 Execution Model: State Machine

The system follows a state-machine model implemented in the main.py script. This design allows for conditional actions based on system state.

4.4.3 State Descriptions:

1. **Initialize Devices**: Instantiates key objects (camera, HackRF, motion detection, database manager), and establishes communication with the server.
2. **Waiting for Car**: Continuously captures video frames and analyzes them for motion. Reference frames are updated until motion is detected.
3. **Car in Frame:** Upon detection, initializes an Event object, logs timestamp along with the box's temperature and humidity, and prompts the server to prepare data storage directories. It also begins recording the event with the camera.
4. **Sweep:** Initiates a frequency sweep from 400–800 MHz in discrete steps, avoiding local oscillator leakage. Raw IQ data and metadata (in .json format) are saved locally.

5. **Send Sweep Data to Server:** Transfers sweep files and accompanying metadata to the server.
6. **Send Video Recording to Server:** Transfers buffered video files corresponding to the event.
7. **Send Metadata to Server:** Transfers the event's metadata JSON file.
8. **Update Database:** Inserts new records into the local SQLite database and sends the updated database to the server
9. **Send Logs and Delete Local Data:** At the end of each operational day, logs are transferred to the server and the day's data is purged from local storage to free up space.

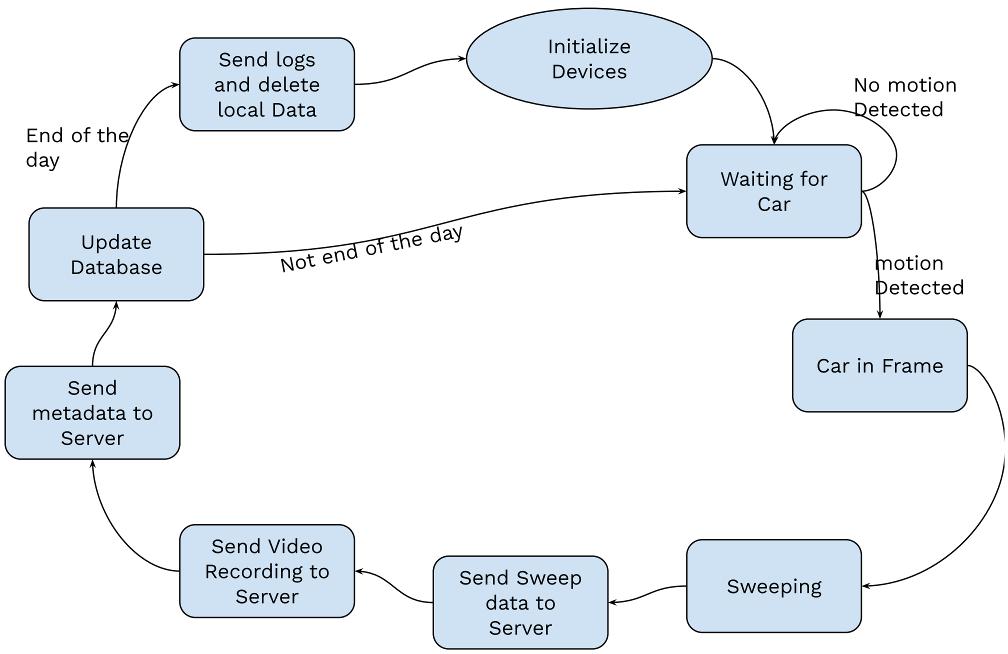


Figure 21: Flow Chart of State Machine Architecture

4.5 Client-Server Communication

Reliable communication with an external server is a key component of the system. Server.py file is run on an external computer connected to the raspberry pi. It listens on port 5000 and waits for TCP requests from the client running on the Raspberry Pi.

The choice of TCP over UDP was deliberate: while TCP introduces some latency, it ensures data integrity through acknowledgements and retransmissions—features critical for our use case involving large file transfers (IQ data, video) and configuration messages.

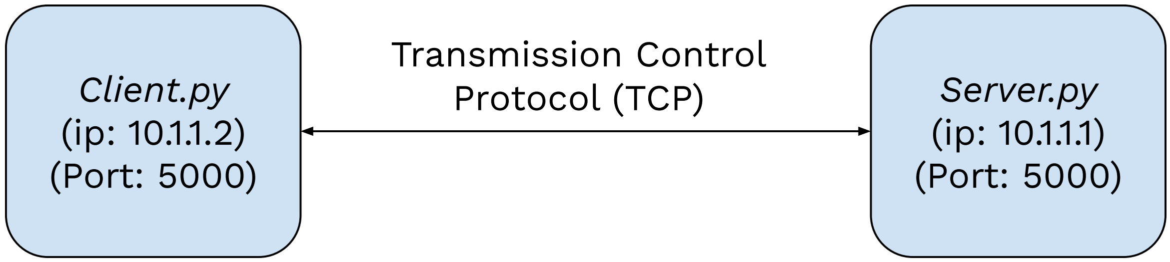


Figure 22: Client-Server Communication

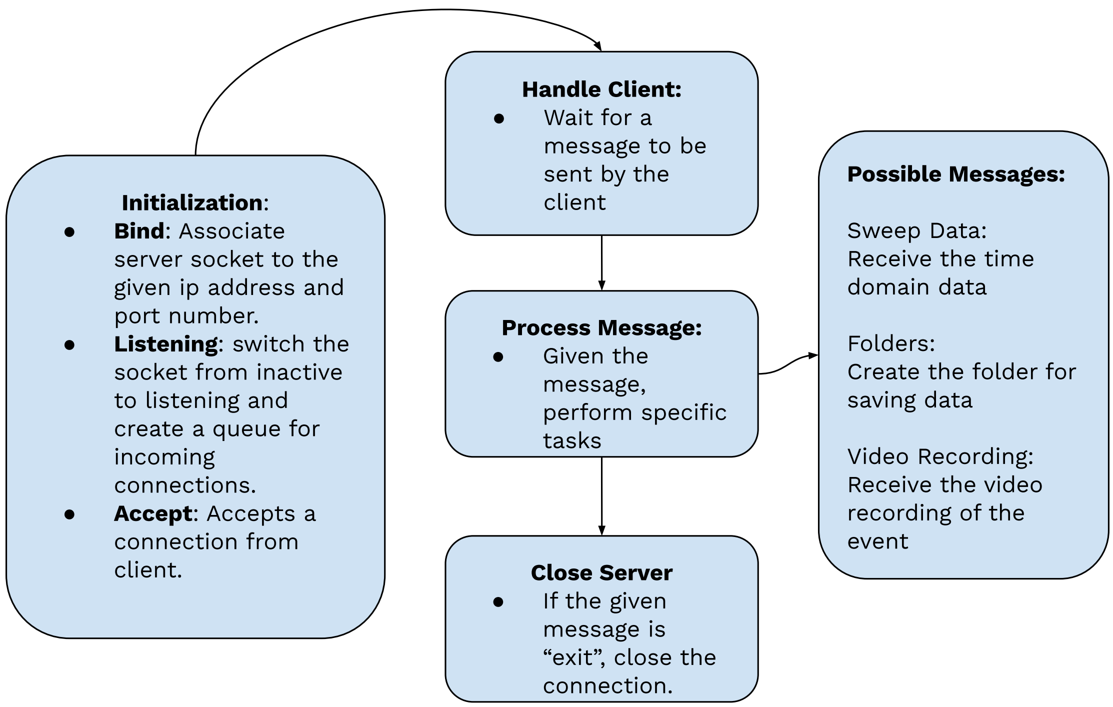


Figure 23: Flow Diagram of Implementation of Server.py and how Messages are Received and Processed from Raspberry Pi

4.6 Graphical User Interface (GUI)

To facilitate data inspection and remote visualization, a GUI was developed using PyQt5. This interface is designed for our sponsor to use and allows real-time and retrospective review of collected data, including:

- Historical event logs (video, spectrum)
- Live video feed from the Raspberry Pi camera
- Real-time SDR spectrum visualization

Live functionality is supported by launching a persistent listener (RpiListener.py) on the Raspberry Pi. The GUI, running on the external computer, connects to this service to pull live frames and spectral data. Users can also adjust parameters like gain and sample rate directly from the GUI.

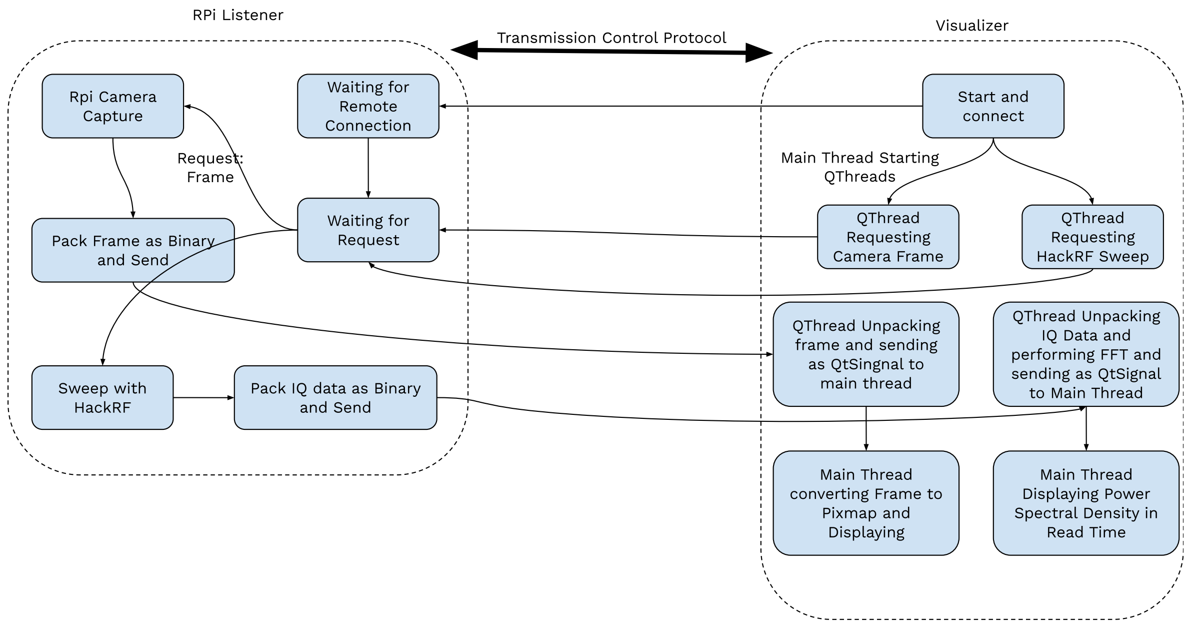


Figure 24: Flow Diagram of how the RPi Listener receives requests from the visualizer and sends back data for real-time monitoring

5 Conclusion

The original motivation for this project stemmed from the need to protect the integrity of radio astronomy data at the DRAO, primarily those conducted by the CHIME radio telescope. As one of the most sensitive instruments of its kind, CHIME operates in such a way that even minor sources of terrestrial RFI can degrade its data quality. Our sponsors hypothesized that vehicle traffic near the CHIME site was a contributor to such interference. The project was then initiated to verify if that hypothesis was true, and if so, to develop a system capable of detecting vehicles and simultaneously recording their associated RFI signatures in the CHIME band (400–800 MHz).

We successfully concluded that vehicles do generate RFI. Additionally, the box successfully met shielding requirements, achieving 100 dB attenuation up to 4 GHz, exceeding the sensitivity needs of CHIME’s operating band. From a software perspective, a modular and maintainable framework was successfully developed, incorporating a custom state machine to manage device states and data acquisition events. Data is securely transferred to an external server via fiber optics and is made accessible through a PyQt5-based GUI. Vehicle presence is reliably detected using classical computer vision techniques, and the SDR subsystem captures correlated spectral data accurately.

Importantly, this solution introduces no additional RFI and is aligned with site operating constraints. The successful construction and demonstration of this system reduces a key operational uncertainty at CHIME and opens new pathways for proactive RFI management.

6 Recommendations

While the SDR subsystem was effective in capturing wideband spectral information through frequency sweeping, further work is required to increase the accuracy of these measurements to the high standards of the DRAO. Moreover, long-term thermal reliability under extreme outdoor conditions has not yet been validated in deployment.

To finalize the work done on this project, we recommend:

1. Purchase a shielding glass window for this box, and properly mount it to the case. This must be done before deployment.
2. Improve the system's frequency detection capabilities. One promising approach would be to use a high-performance FPGA in conjunction with an ADC with a high sampling rate.
3. Finalize outdoor testing for certainty of quality.
4. Find ways to further improve the shielding so that higher frequencies have higher isolation.
5. Build upon the software stack created to fit the specific needs of the DRAO. This can be done while the box is deployed, as remote access and changes are possible.

References

[1] Lee Billings (9 July 2013). "A Brilliant Flash, Then Nothing: New 'Fast Radio Bursts' Mystify Astronomers". Scientific American.

[2] [2201.07869] An Overview of CHIME, the Canadian Hydrogen Intensity Mapping Experiment

[3] [2201.11822] Using the Sun to Measure the Primary Beam Response of the Canadian Hydrogen Intensity Mapping Experiment

[4] UBC Experimental Cosmology: home

Appendix A