

System Integration of a Remotely Accessible Satellite Tracker

by

Rushil Kisoorn

(216004605)

Submitted in fulfilment of the academic requirements for the degree of

Master of Science in Mechatronics Engineering

College of Agriculture, Engineering and Science

University of KwaZulu-Natal

Durban

South Africa

January 2022

Supervisor: Professor R. Stopforth

Declaration – Plagiarism

I, **Rushil Kisoan**, declare that:

1. The research reported in this dissertation, except where otherwise indicated, is my original research.
2. This dissertation has not been submitted for any degree or examination at any other university.
3. This dissertation does not contain other persons' data, pictures, graphs, or other information, unless specifically acknowledged as being sourced from other persons.
4. This dissertation does not contain other persons' writing unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
 - a. Their words have been re-written, but the general information attributed to them has been referenced
 - b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks and referenced.
5. This dissertation does not contain text, graphics or tables copied and pasted from the Internet unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

Signature: ......

As the candidate's Supervisor, I agree/do not agree to the submission of this dissertation.

Signature:

Acknowledgements

- My supervisor: Professor R. Stopforth, for his assistance and guidance.
- ICS for setting up a new wireless network.
- The workshop technicians for fabricating the required components.
- The 2019 and 2020 UKZN final year mechanical engineering groups for creating the frame structure and antenna.
- My family for their encouragement, support, and advice.

Abstract

The recent rise in space missions and satellite launches has piqued the interest of many. As a result, the growing number of objects in orbit has led to an increase in space monitoring. While large-scale tracking systems exist, many enthusiasts utilise rudimentary methods for satellite monitoring. Some have created do-it-yourself rotators or radio reception stations. Subsequently, a remotely accessible system was required to track satellites and receive transmissions.

A small-scale system was designed to fulfil the requirements. Aluminium extrusions were used to create the frame, while microprocessors, sensors and other components were integrated to complete the rotator assembly. A circuit layout was drawn up to coordinate the flow of power and data between components. Low-powered electronics allow the system to run off-grid via a 12 V battery. Many features of the Raspberry Pi lent well to the required application, such as wireless network connectivity and a remote access service. Gpredict, an advanced satellite tracking software, was used to calculate the coordinates for rotation. The RTL-SDR, a software-controlled tuner, provided control to receive radio frequencies. Features such as a slow scan image decoder and frequency identification toolkit were also integrated. The user interface program called TrackGUI was coded, which allows easy access to tracking features. A login system was created with ‘admin’ and ‘user’ access levels. Initially, the presence of GlobalProtect on the UKZN network posed an issue, but methods were implemented to solve this problem. Various aspects of the system were tested and iterated until they met the required parameters.

Test results showed that the system could operate within 5 degrees of the intended target. Radio transmissions and slow-scan images could be received and recorded. Therefore, the aim of integrating hardware and software components to create a remotely accessible satellite tracking system was achieved.

The demonstration video is available at: https://www.youtube.com/watch?v=zwEyT_j5G54

The sound clips are available at: https://stuukznac-my.sharepoint.com/:f/g/personal/216004605_stu_ukzn_ac_za/Er5PZoVWia9MnTGqSic6URMBsZw0kscVa9VPwY57ORLMfg?e=d4ExBv or https://drive.google.com/drive/folders/1yCP_EO-uUbk21xzCndKR4dIRhPjShVAi?usp=sharing

Table of Contents

List of Figures	xiii
List of Tables	xviii
List of Acronyms	xix
Nomenclature - Symbols.....	xxii
Nomenclature – Greek Symbols.....	xxii
Nomenclature.....	xxiii
1. Introduction	1
1.1. Research Background.....	1
1.2. Research Purpose & Impact	2
1.3. Research Aims.....	2
1.4. Research Objectives	2
1.5. Methodology	3
1.6. Research Contributions	4
1.7. Overview of the Dissertation	5
1.8. Chapter Summary.....	6
2. Literature Review	7
2.1. Introduction	7
2.2. Satellite Tracking History.....	7
2.3. Current Methods.....	8
2.4. Space Domain Awareness	9

2.5.	Existing Systems	9
2.5.1.	Large-scale Systems	9
2.5.1.1.	Space Fence	9
2.5.1.2.	Grand Réseau Adapté à la Veille Spatiale (GRAVES)	10
2.5.2.	Small-scale Systems	10
2.5.2.1.	NyanSat	11
2.5.2.2.	Satellite Networked Open Ground Stations (SatNOGS)	11
2.5.2.3.	SATRAN Antenna Rotator	12
2.5.2.4.	ABB Rotator	13
2.6.	Tracking Technologies	13
2.6.1.	Radio	13
2.6.1.1.	RTL-SDR Array	14
2.6.1.2.	Radio Frequency Sensing	14
2.6.1.3.	KrakenSDR	14
2.6.1.4.	TRANET Doppler Tracking System.....	15
2.6.2.	Visual	16
2.6.2.1.	FireOPAL	16
2.6.3.	Radar	17
2.6.4.	Laser.....	17
2.7.	Orbital Mechanics	17
2.7.1.	Orbit Types.....	18

2.7.2.	Two-Line Element (TLE).....	19
2.7.3.	Orbital Positions	20
2.8.	Satellite Data Download.....	20
2.8.1.	Radio	20
2.8.2.	Slow-Scan Television (SSTV).....	21
2.9.	Chapter Summary.....	22
3.	System Design.....	23
3.1.	Introduction	23
3.2.	Mechatronic System	23
3.3.	Mechanical Design	24
3.3.1.	Mechanical Design Considerations.....	24
3.3.2.	Mechanical System Decomposition.....	24
3.3.3.	Mechanical Components	25
3.3.3.1.	Frame.....	25
3.3.3.2.	Drive.....	26
a.	Spur Gears	26
b.	Gear Shaft.....	27
c.	Motor Mount	27
3.3.3.3.	Azimuth Axis.....	28
a.	Stands	28
3.3.3.4.	Elevation Axis	29

3.3.3.5.	Mounts.....	30
3.3.3.6.	Hole Reducers	30
3.3.3.7.	IMU Bracket.....	31
3.3.4.	Mechanical Part Selection	32
3.3.5.	Dimensions.....	32
3.4.	Electronic Integration	33
3.4.1.	Electronic Integration Considerations.....	33
3.4.2.	Electronic Components.....	33
3.4.2.1.	Control Computer.....	33
a.	Raspberry Pi 4	33
b.	Rock Pi 4	34
c.	Banana Pi M5	34
3.4.2.2.	Microcontroller.....	35
a.	Arduino Pro Micro	35
b.	Teensy	35
c.	ESP32.....	36
3.4.2.3.	SDR.....	36
a.	RTL-SDR	36
b.	Airspy R2	36
3.4.2.4.	Global Positioning System (GPS).....	37
3.4.2.5.	Inertial Measurement Unit (IMU).....	37

3.4.2.6.	Motors	37
a.	Stepper Motor.....	37
b.	Wiper Motor.....	37
3.4.3.	Electronic Component Selection	37
3.4.4.	Circuit Design.....	39
3.4.4.1.	Power.....	39
3.4.4.2.	Signal.....	40
a.	Raspberry Pi.....	40
b.	Arduino	40
3.4.4.3.	Inter-device.....	41
3.4.5.	Internet Connectivity	41
3.5.	Software Design	42
3.5.1.	Software Design Considerations.....	42
3.5.2.	Programming Languages.....	42
3.5.2.1.	C++.....	42
3.5.2.2.	Python.....	42
3.5.3.	Tracking Platforms	43
3.5.3.1.	Gpredict.....	43
3.5.3.2.	PREDICT	44
3.5.3.3.	Other Tracking Software	44
3.5.4.	Rotator Control.....	44

3.5.4.1.	Hamlib.....	45
3.5.4.2.	Rotctl.....	45
3.5.4.3.	EasyComm	45
3.5.4.4.	SARCNET Rotator.....	45
3.5.5.	Radio Control	46
3.5.5.1.	GQRX.....	47
3.5.5.2.	SDR++.....	47
3.5.6.	GPS	47
3.5.1.	Frequency Identification.....	48
3.5.2.	SSTV	48
3.5.3.	Graphical User Interface (GUI)	48
3.5.3.1.	Design Parameters	48
3.5.3.2.	Design Planning.....	48
3.5.3.3.	Digital Design.....	50
3.6.	Assembly and Setup	53
3.6.1.	Mechanical Setup	53
3.6.2.	Electronic Setup	54
3.6.2.1.	Power Circuit.....	54
3.6.2.2.	Signal Circuit.....	55
3.6.2.3.	Antenna	56
3.6.3.	Software Setup	57

3.6.3.1.	Raspberry Pi	57
3.6.3.2.	Remote Access	57
3.6.3.3.	Gpredict.....	58
3.6.3.4.	GQRX.....	59
3.6.3.5.	Rotator.....	59
3.6.3.6.	Network Connectivity.....	60
a.	Wi-Fi	60
b.	GlobalProtect.....	61
3.6.3.7.	IMU	61
3.6.3.8.	GPS	62
3.6.4.	Iteration and Improvements	63
3.7.	Verification	67
3.7.1.	Software Validation.....	67
3.7.1.1.	Experimental procedure.....	67
a.	Login Window Results	68
b.	Main Menu Results.....	70
3.7.2.	GPS Verification	72
3.7.2.1.	Experimental Procedure.....	72
3.7.2.2.	GPS Results	73
3.8.	Chapter Summary.....	75
4.	Testing and Results.....	76

4.1.	Introduction	76
4.1.1.	Apparatus	76
4.2.	Orientation and Feedback Test	76
4.2.1.	Experimental Procedure	76
4.2.2.	Orientation and Feedback Results	78
4.2.2.1.	Elevation 15°, 30° and 45° Tests	78
4.2.2.2.	Elevation 60°, 75 ° and 90° Tests	79
4.2.2.3.	Azimuth Test	80
4.2.3.	Orientation and Feedback Discussion.....	81
4.3.	Satellite Tracking Test.....	82
4.3.1.	Experimental Procedure	82
4.3.2.	Satellite Tracking Results	83
4.3.3.	Satellite Tracking Discussion	86
4.4.	Radio Reception Test	87
4.4.1.	Experimental Procedure	87
4.4.2.	Radio Reception Results.....	87
4.4.3.	Radio Reception Discussion	89
4.5.	SSTV Test	90
4.5.1.	Experimental Procedure	90
4.5.1.	SSTV Results	91
4.5.1.	SSTV Test Discussion	92

4.6.	Frequency ID Test.....	93
4.6.1.	Experimental Procedure	93
4.6.2.	Frequency ID Results	94
4.6.1.	Frequency ID Discussion.....	95
4.7.	Chapter summary	97
5.	Conclusion and Recommendations.....	98
5.1.	Conclusion	98
5.2.	Recommendations	100
6.	Appendices	101
6.1.	Appendix A – Calculations.....	101
6.1.1.	Power Draw.....	101
6.1.2.	Elevation Torque	102
6.2.	Appendix B – Drawings	103
6.3.	Appendix C – Results.....	105
6.3.1.	Orientation and Feedback Test Data.....	105
6.3.2.	Satellite Tracking Test Data	109
6.4.	Appendix D – Code.....	112
6.4.1.	Doppler Shift Algorithm.....	112
6.4.2.	TrackGUI Code	113
7.	References	122

List of Figures

Figure 1-1: Methodology Overview	4
Figure 2-1: The first Baker-Nunn camera	7
Figure 2-2: Space Fence Diagram	9
Figure 2-3: GRAVES Bistatic Radar	10
Figure 2-4: Antenny PCB V2.....	11
Figure 2-5: SatNOGS Rotator	12
Figure 2-6: SATRAN Antenna Rotator.....	12
Figure 2-7: ABB Robot with Antenna.....	13
Figure 2-8: 3 x 7 Antenna Array	14
Figure 2-9: KrakenSDR	15
Figure 2-10: TRANET Network	15
Figure 2-11: Deployment of two FireOPAL observatories.....	16
Figure 2-12: Diagram showing Azimuth and Elevation	17
Figure 2-13: Comparison of Orbit Types	18
Figure 2-14: TLE Description	19
Figure 2-15: ISS SSTV Image from August 2021.....	21
Figure 3-1: Mechatronics Venn Diagram	23
Figure 3-2: Mechanical System Decomposition Diagram	24
Figure 3-3: PG30 Profile	25
Figure 3-4: Frame subassembly	25

Figure 3-5: Drive assembly	26
Figure 3-6: Spur Gears	26
Figure 3-7: Gear Shaft.....	27
Figure 3-8: Motor Mount	27
Figure 3-9: Azimuth Shaft.....	28
Figure 3-10: Frame with the Small Stand.....	28
Figure 3-11: Large Stand.....	29
Figure 3-12: Elevation Shaft	29
Figure 3-13: Shaft Mount.....	30
Figure 3-14: Elevation Axis	30
Figure 3-15: Hole Reducer	31
Figure 3-16: IMU Bracket.....	31
Figure 3-17: Raspberry Pi 4B Board	34
Figure 3-18: Rock Pi 4 Board.....	34
Figure 3-19: Banana Pi M5 Board.....	35
Figure 3-20: Arduino Pro Micro	35
Figure 3-21: Teensy 4.0	36
Figure 3-22: ESP32-WROOM	36
Figure 3-23: Circuit Power Diagram.....	39
Figure 3-24: Signal Connection Diagram.....	40
Figure 3-25: Serial Communication via GPIO	41

Figure 3-26: Gpredict.....	43
Figure 3-27: PREDICT	44
Figure 3-28: Hamlib API	45
Figure 3-29: Flowchart of Rotator Process.....	46
Figure 3-30: NMEA Sentence Description.....	47
Figure 3-31: UI Design Concept	49
Figure 3-32: Login Window Concept.....	49
Figure 3-33: First UI Iteration	50
Figure 3-34: Second UI Iteration.....	51
Figure 3-35: Final Login Window.....	51
Figure 3-36: User Level Pop-up.....	51
Figure 3-37: Admin-level Iinterface.....	52
Figure 3-38: Edit User Pop-up	52
Figure 3-39: Hierarchy of System	53
Figure 3-40: Power Connections	54
Figure 3-41: Signal Circuit.....	55
Figure 3-42: Antenna Verification	56
Figure 3-43: Raspberry Pi Imager	57
Figure 3-44: RealVNC Selection Window	58
Figure 3-45: Gpredict Monitoring Window	58
Figure 3-46: Initial GQRX Test	59

Figure 3-47: Serial Link	59
Figure 3-48: IMU Calibration	62
Figure 3-49: GPS Setup	62
Figure 3-50: GPS Serial Output	63
Figure 3-51: Wiper Motor Pinout.....	64
Figure 3-52: Neatened Wiring with Ferrite Rings Added.....	64
Figure 3-53: Azimuth Motor Shaft.....	65
Figure 3-54: Frame Misalignment.....	65
Figure 3-55: System Rebuild.....	66
Figure 3-56: Screenshot Showing the Login Window and CPU Monitor.....	67
Figure 3-57: Graph of Login Menu CPU Usage vs Time	68
Figure 3-58: Graph of Login Menu CPU Temperature vs Time.....	69
Figure 3-59: Graph of Main Menu CPU Usage vs Time	70
Figure 3-60: Graph of Main Menu CPU Temperature vs Time.....	71
Figure 3-61: GPS Output	72
Figure 3-62: GPS Phone Location.....	73
Figure 3-63: Scatter Plot of GPS Coordinates	74
Figure 4-1: Elevation Test.....	77
Figure 4-2: Azimuth Test	77
Figure 4-3: Box and Whisker of 15°, 30° and 45° Test	78
Figure 4-4: Box and Whisker of 60°, 75° and 90° Test	79

Figure 4-5: Box and Whisker of Azimuth Test	80
Figure 4-6: System Outside.....	82
Figure 4-7: Phone Verification of Position.....	83
Figure 4-8: Satellite Tracking.....	83
Figure 4-9: Meteor M2 Tracking.....	84
Figure 4-10: NOAA 18 Tracking	84
Figure 4-11: NOAA 19 Tracking	85
Figure 4-12: ISS Tracking.....	85
Figure 4-13: Radio Reception Waterfall and Gpredict	87
Figure 4-14: QSSTV Decoder.....	90
Figure 4-15: ISS SSTV Radio Transmission.....	91
Figure 4-16: ISS SSTV Decoding, 1	91
Figure 4-17: ISS SSTV Decoding, 2	92
Figure 4-18: ISS SSTV Decoding, 3	92
Figure 4-19: IQ Files.....	93
Figure 4-20: Frequency Recording.....	94
Figure 4-21: Frequency Analysis	94
Figure 4-22: Orbit Determination.....	95
Figure 4-23: Satellite ID.....	95
Figure 4-24: 75 Satellites Added.....	96

List of Tables

Table 3-1: Mechanical Design Requirements.....	24
Table 3-2: Mechanical COTS Components.....	32
Table 3-3: Electronic Integration Requirements.....	33
Table 3-4: Electronic COTS Components.....	38
Table 3-5: Software Design Requirements.....	42
Table 3-6: Login Menu CPU Usage.....	68
Table 3-7: Login Menu CPU Temperature.....	69
Table 3-8: Main Menu CPU Usage.....	70
Table 3-9: Main Menu CPU Temperature.....	71
Table 3-10: GPS Verification Results	73
Table 4-1: Waveforms of recordings.....	88
Table 5-1: System Comparison	99
Table 6-1: 15° Test Results	105
Table 6-2: 30° Test Results	105
Table 6-3: 45° Test Results	106
Table 6-4: 60° Test Results	106
Table 6-5: 75° Test Results	107
Table 6-6: 90° Test Results	107
Table 6-7: Azimuth Test Results.....	108
Table 6-8: Satellite Tracking Results	109

List of Acronyms

A-GPS	Assisted Global Positioning System
API	Application Programming Interface
CAD	Computer-Aided Design
COTS	Commercial Off the Shelf
DC	Direct Current
DDS	Defense Digital Service.
DFA	Design for Assembly
DFM	Design for Manufacture
DIY	Do-It-Yourself
DOA	Direction of Arrival
DoF	Degrees of Freedom
EMI	Electromagnetic Interference
GEO	Geosynchronous Equatorial Orbit
GPS	Global Positioning System
GRAVES	<i>Grand Réseau Adapté à la Veille Spatiale</i>
GUI	Graphical User Interface
HEO	High Earth Orbit
I/O	Input and Output
IBEX	Interstellar Boundary Explorer
ICRA	International Conference on Robotics and Automation
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronic Engineers
IMU	Inertial Measurement Unit
IQ	Quadrature signal
ISS	International Space Station
ITU	International Telecommunication Union
LCD	Liquid Crystal Display
LED	Light Emitting Diode

LEO	Low Earth Orbit
MEO	Medium Earth Orbit
MNASSA	Monthly Notes of the Astronomical Society of South Africa
MSFN	Mercury Space Flight Network
NASA	National Aeronautics and Space Administration
NMEA	National Marine Electronics Association
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defence Command
NSA	National Security Agency
OLED	Organic Light-Emitting Diode
ONERA	<i>Office National d'Etudes et de Recherches Aéronautiques</i>
PCB	Printed Circuit Board
PETG	Polyethylene Terephthalate Glycol
PWM	Pulse Width Modulation
RADAR	Radio Detection and Ranging
RDF	Radio Direction Finding
RPM	Revolutions per minute
RTC	Real-Time Clock
SANSA	South African National Space Agency
SatNOGS	Satellite Networked Open Ground Stations
SBC	Single Board Computer
SBSS	Space-Based Space Surveillance
SCL	Serial Clock
SDA	Serial Data
SDK	Software Development Kit
SDR	Software Defined Radio
SHF	Super High Frequency
SSA	Space Situational Awareness
SSI	Slow Scan Image
SSN	Space Surveillance Network

SSO	Sun-synchronous Orbit
SSTV	Slow Scan Television
TDRS	Tracking and Data Relay Satellite
TESS	Transiting Exoplanet Survey Satellite
TLE	Two-Line Element
UCS	Union of Concerned Scientists
UDP	User Datagram Protocol
UI	User Interface
UKZN	University of KwaZulu-Natal
USB	Universal Serial Bus
USSF	United States Space Force
VFO	Variable Frequency Oscillator
VNC	Virtual Network Computing
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity

Nomenclature - Symbols

Variable	Description	Unit
F	Force	Newton
g	Gravitational Constant (9.81 m/s ²)	Metres per second squared
I	Current	Ampere
m	Mass	Kilogram
P	Power	Watt
$P_{Arduino}$	Power, Arduino	Watt
P_{minor}	Power, Minor Losses	Watt
P_{motor}	Power, Motor	Watt
P_{RPi}	Power, Raspberry Pi	Watt
P_{total}	Power, Total	Watt
r	Lever Arm	Metre
R	Resistance	Ohm
U	Gear Ratio	-
V	Voltage	Volt
Z	Number of Gear Teeth	-
Z_1	Number of Gear Teeth, 1	-
Z_2	Number of Gear Teeth, 2	-

Nomenclature – Greek Symbols

τ	Torque	Newton-metre
τ_1	Torque, 1	Newton-metre
τ_2	Torque, 2	Newton-metre

Nomenclature

Description	Unit
Megapixel	MP
Megabyte	MB
Metre	m
Millimetre	mm
Centimetre	cm
Azimuth	AZ
Elevation	EL
Hertz	Hz
Degrees (Angle)	°
Degrees Celsius	°C
Terabyte	TB

1. Introduction

The introduction chapter provides insight into the research topic. A brief overview of satellite tracking is provided as well as the significance of this research. The aims and objectives are listed, with the methodology and overview of the dissertation being discussed.

1.1. Research Background

Since the launch of Sputnik 1 in 1957, keeping track of objects in orbit has become a critical part of space research. Early tracking methods relied on visual confirmation via telescope. Modern technology such as radio detection and ranging (radar), optical sensors, and surveillance satellites are used to automatically detect and document objects in space. This data is uploaded to a database such as *Space-Track*, or *CelesTrak* (CelesTrak, 2022). Making this information publicly accessible helps guide future launches while keeping the skies safe.

Approximately 19 700 of 46 000 objects being tracked are debris (Space-Track, 2022). National Aeronautics and Space Administration (NASA) scientists Kessler & Cour-Palais (1978) stated that due to the high density of objects in space, any collision has the potential to trigger a chain reaction of impacts. Kessler (1991) wrote that while fragments may be as small as paint flecks, the average size of debris in space is 1 kg. Experimental testing showed that a 1 kg object, orbiting at a speed of 10 km/s, could destroy a 1 000 kg spacecraft (National Research Council, 1995). Subsequently, various space agencies, scientists and hobbyists have become more active in monitoring the movement of both natural and artificial satellites.

The “billionaire space race” and initiatives such as SpaceX Starlink have highlighted the importance of satellites and space to laypeople (The New University, 2022). Many have taken an interest in satellite monitoring, with some hobbyists creating do-it-yourself (DIY) rotators and tracking systems (Tysonpower, 2017). Very few systems exist that incorporate the ability to track satellites and receive transmissions while being remotely accessible and easy to use.

Therefore, the primary focus of this research aims to design and build a low-cost, remotely accessible satellite tracker. Using knowledge learnt from past and present methods, the most suitable hardware can be integrated to create an effective system capable of tracking and receiving transmissions from satellites.

1.2. Research Purpose & Impact

South Africa is in a geographically significant location on Earth. This position in the southern hemisphere makes it ideal for observing satellites within their first few hours of launch (Astronomical Society of Southern Africa, 2022). Various space agencies have taken an interest in South Africa as many orbital paths pass over the area (University of Cape Town, 2022).

Creating a low-cost tracking and monitoring system would enable many more base stations to be set up for hobbyist and professional use. Including radio reception capabilities will allow transmissions to be received and decoded. By increasing the number of base stations, the likelihood of detecting something that was previously undiscovered is also increased.

1.3. Research Aims

This study involves both design and research aspects to develop a satellite tracking system. The aims are as follows:

- To create a system that is capable of accurately monitoring and tracking satellites.
- To design a system that works off-grid via battery.
- To incorporate radio reception and decoding functionality.
- To develop a remotely accessible user interface (UI).

1.4. Research Objectives

To achieve the aims listed above, the objectives mentioned below would need to be accomplished:

- Review literature on past and present tracking methods and the technologies used.
- Identify suitable tracking methods and possible improvements.
- Develop the mechanical rotator system.
- Research the electronic controllers and sensors that are often used for this application.
- Develop the electronic system.
- Research software design and UI development.
- Develop a remotely accessible UI, implementing tracking and radio reception software.
- Assemble and set up the various subsystems.
- Evaluate all aspects of the system to determine if modifications are required.
- Iteratively improve subsystems to meet the required parameters.
- Perform experimental testing to validate system performance.
- Analyse results and draw conclusions.

1.5. Methodology

This research focused on designing and developing a remotely accessible satellite tracking system. A design and research task relies on an iterative approach to solve the task at hand. Aspects of the system are validated and improved until they meet specified parameters.

The literature review was used to determine the significance of the research by analysing relevant content. Evaluation of past systems helped gather information to meet the aims and objectives.

Careful consideration of the required specifications helped drive the design decisions. The mechatronic system was broken down into mechanical, electronic and software subsystems. Once the necessary subsystems were designed and developed, the appropriate parts were purchased and fabricated. Once the mechanical and electronic systems were assembled, the software was installed.

Various aspects of the system were verified and validated. Modifications and improvements were made until the preliminary results met the required parameters.

Experimental testing was used to gather quantitative data about various aspects of the system. Two smartphones were used to record data and verify certain tests due to their inclusion of sensors such as a gyroscope, accelerometer, and Global Positioning System (GPS). Mobile applications such as *Commander Compass Go* and *Phyphox* were used to extract data from the phone sensors.

Data points were recorded manually and via the Arduino serial monitor. The use of multiple devices helped improve the validity of the results. Some tests were repeated at least three times to increase the reliability of the results. Multiple recordings were taken to ensure the accuracy of the data. In some cases, the average was taken, while other tests required the relationship between recorded values to be determined.

The data was tabulated and represented on appropriate charts to visualise the relationship between the results. Radio recordings were decoded to evaluate the quality of slow-scan images.

Analysis of the results helped determine the performance and effectiveness of the system. Discussion of the outcomes evaluated if the aims and objectives of this research were met.

Figure 1-1 outlines the structure of the methodological approach.

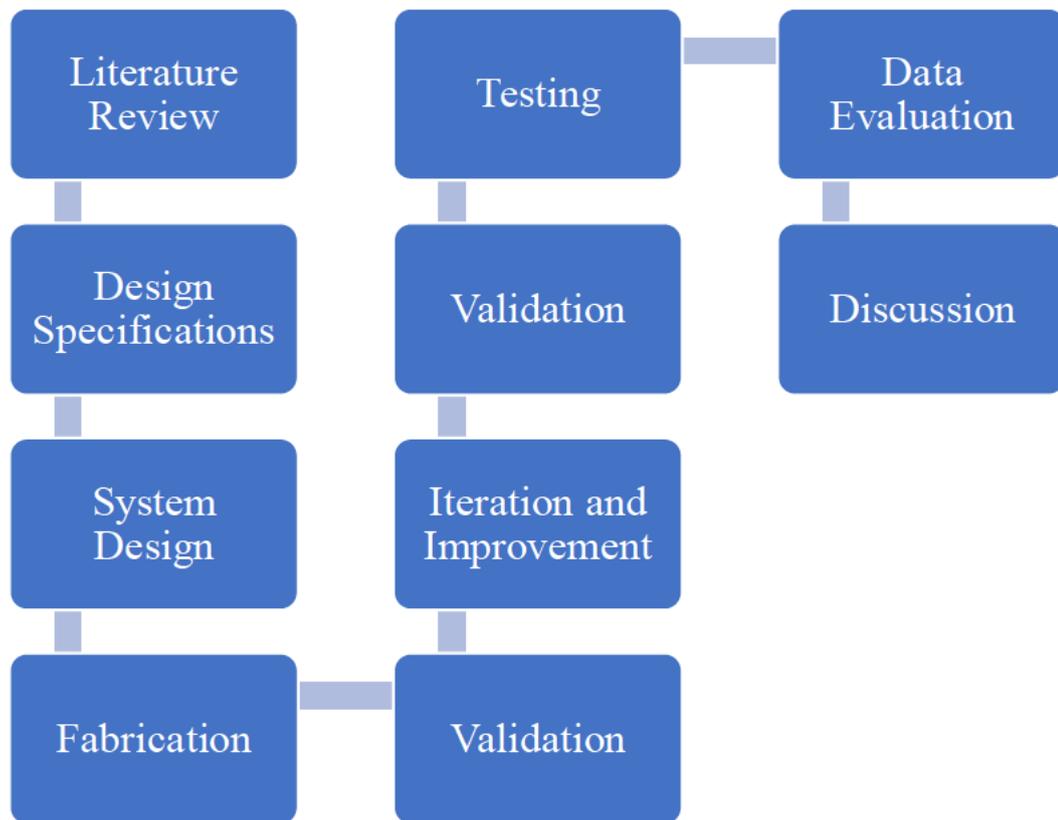


Figure 1-1: Methodology Overview

1.6. Research Contributions

The following contributions were made:

- Integration of hardware and electronic components to create a system used to track satellites and listen in on their radio transmissions.
- Implementation of low-powered electronic components which enables remote operation and access via a 12 V battery.
- A remotely accessible graphical user interface for various tracking tools, with an admin and user access hierarchy.

1.7. Overview of the Dissertation

This overview provides a guideline pertaining to the dissertation chapters:

Chapter 1 - Introduction: This chapter introduces the topic by highlighting the background, purpose, aims and objectives. A description of the methodology allows the reader to better understand the research and development process of the tracking system.

Chapter 2 - Literature Review: The literature review chapter provides a brief history of satellite tracking. Relevant systems, technologies and methods were reviewed. Analysing this information helped formulate a solution to the tasks at hand.

Chapter 3 - System Design: The design chapter provides insight into the development process of the various tracking subsystems.

- **Mechanical Design:** Overview and improvement of the existing mechanical assembly.
- **Electronic Design:** Review and selection of electronic components. Design of electronic circuits for both power and signal flow.
- **Software Design:** Review of programming languages and relevant software. Implementation of software methods to integrate rotator components. Design of a user interface program.
- **Assembly and Setup:** Construction and configuration of the system.
- **Verification:** Validation of various aspects within the system.
- **Iteration and Improvements:** Details steps taken to improve the system after initial tests. Issues that arose were debugged and fixed before final testing.

Chapter 4 – Testing and Results: This chapter details the procedures required to perform each test. The results obtained from each test were analysed to evaluate the system's performance. The outcome of each test was discussed.

Chapter 5 - Conclusion and Recommendations: The conclusion explains how the objectives were met. The final system was compared to similar trackers, with suggestions being made for future improvements.

Chapter 6 - Appendices: A supplementary chapter containing calculations, drawings, results and code used in this research.

- **Appendix A – Calculations:** Details the calculations performed.
- **Appendix B – Drawings:** Presents the technical drawings for designed parts.
- **Appendix C – Results:** Lists the raw data values and results from testing.
- **Appendix D – Code:** Lists the code created for the system.

1.8. Chapter Summary

The background for this research was understood. The aims and objectives were outlined, and the methodology required to accomplish them. An overview of each chapter was detailed.

The following chapter presents the literature review, which provides insight into the past and present methods used for satellite tracking. Suitable technology was investigated to provide solutions to meet the objectives of the research.

2. Literature Review

2.1. Introduction

The following chapter details the research required to knowledge areas required to develop a satellite tracking system. Previous systems were explored to understand the developments in this field. Current systems were analysed to determine the most effective technological methods which could be implemented to create a tracking system.

2.2. Satellite Tracking History

The origins of satellite tracking stem from military research into missile tracking during the 1940s. Linda Neuman Ezell was a NASA author who published many books for the agency. The *NASA Historical Data Book* mentioned that the organisation took over many projects from the United States military in 1958 and further developed them for satellite tracking and data acquisition (Ezell, 1988).

The earliest forms of satellite tracking utilised optical methods. The Smithsonian Astrophysical Observatory operated Baker-Nunn cameras from 1958 to 1970. These cameras were a derivative of the Schmidt camera, designed to track satellites. Figure 2-1 shows the first Baker-Nunn camera that was built for the Smithsonian Astrophysical Observatory (Boller & Chivens, 2011).



Figure 2-1: The first Baker-Nunn camera

In a paper from the November 1962 *Photogrammetric Engineering & Remote Sensing* journal, Brandenberger (1962) mentioned that the camera required skilled operation as the assembly was 2.5 meters long, 1.5 meters wide, and three meters high.

The Naval Research Laboratory explored radio tracking through their Minitrack project. Radio interferometry was used to determine orbital data (Ezell, 1988). When NASA acquired the project in 1958, there were a series of 12 base stations with precise clock synchronisation to record data for satellite orbit determination.

By July 1961, NASA had completed the Mercury Space Flight Network (MSFN) (Ezell, 1988). A series of 18 ground stations and two ships utilised radar to provide instantaneous tracking of spacecraft used for human spaceflight (NASA, 2017).

Advancements in technology meant that by the 1980s, satellites were able to track other satellites. The family of Tracking and Data Relay Satellites (TDRS) were launched into orbit from 1983 (NASA, 2021). Each generation brought new improvements, but all implementations of the satellite featured radar and communication antennas at their core (NASA, 2012).

The rapid development of satellite technology saw the launch of the Hubble Space Telescope and the ISS in the 1990s. With the introduction of CubeSats, many small research and development missions became a reality due to their standardised design parameters and low cost. As of February 2022, the Nanosats Database has a record of over 3300 nanosatellite launches (Kulu, 2022). The skies are rapidly populating with various satellites, and the ever-increasing density requires modern methods to keep track of their whereabouts.

2.3. Current Methods

Many satellite tracking systems today still rely on optical and radar technology as well as Space-Based Space Surveillance (SBSS) satellites.

With thousands of new satellites launched each year, it has become impractical to track each object in space manually. The US Space Surveillance Network (SSN) uses an array of radar, telescopes, and satellites to detect and document objects in the skies (Secure World Foundation, 2019). They aim to avert potential catastrophes and improve space safety by making the catalogue publicly accessible via www.space-track.org (Space-Track, 2022).

2.4. Space Domain Awareness

Space Domain Awareness is a phrase used to describe the detection, tracking and identification of objects in the Earth's orbit. It superseded the term space situational awareness (SSA). Sophisticated systems exist to monitor, track, and predict the movement of satellites (ExoAnalytic Solutions, 2021). Studying an object's orbit allows for a greater understanding of its behaviour.

2.5. Existing Systems

Researching large and small tracking systems provides an understanding of the technology in use.

2.5.1. Large-scale Systems

Various agencies have created state-of-the-art detection systems to monitor the objects in space.

2.5.1.1. Space Fence

The Space Fence was developed for the United States Space Force (USSF) by Lockheed Martin (Lockheed Martin Corporation, 2021). It tracks over 25 000 objects in low Earth orbit via S-band radar, with selected data being available from www.space-track.org and www.celestrak.com. The primary radar system, located on the Marshall Islands, was deemed operational in March 2020 (Shimkus, 2020). Figure 2-2 illustrates the coverage and capabilities of the system (Director Operational Test and Evaluation, 2019).

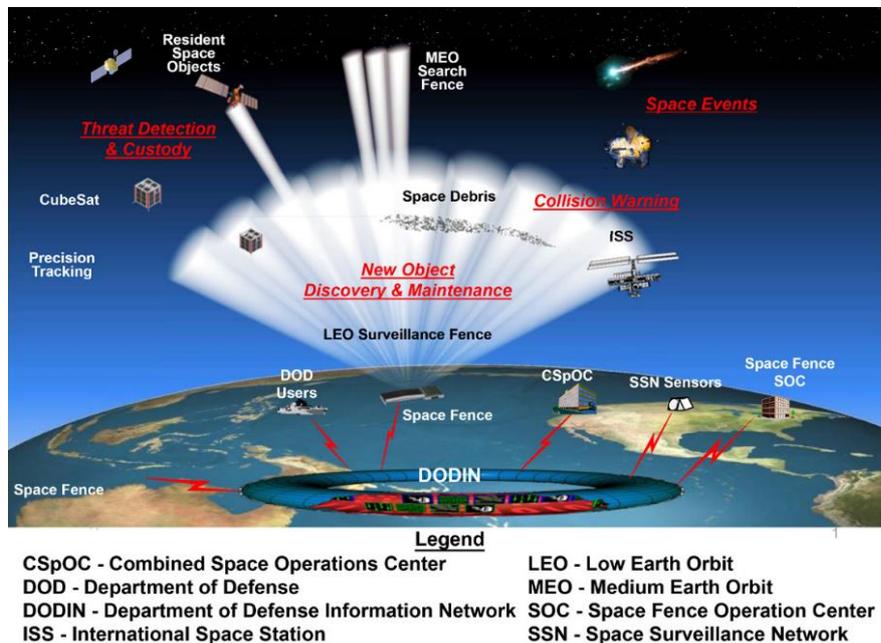


Figure 2-2: Space Fence Diagram

2.5.1.2. Grand Réseau Adapté à la Veille Spatiale (GRAVES)

The French-based GRAVES radar surveillance system aids the European Space Agency track debris and satellites (VINCI Energies, 2017). Bistatic Doppler radar allows the emitter and receiver to be in different locations. The emitters broadcast over a large area of Europe, allowing many to set up a receiver.

Users require an antenna and a Software Defined Radio (SDR) tuned to 143.050 MHz to listen in on the signals being reflected (RTL-SDR, 2017). Jouadé & Barka (2019) presented Figure 2-3 to better visualise the operation of the GRAVES radar as part of their study into its operation (Jouadé & Barka, 2019).

After 15 years of development by *Office National d'Etudes et de Recherches Aéropatiales* (ONERA), it was deemed operational in 2005. A 5-year long upgrade was commissioned in 2017, which will allow for smaller objects to be detected.

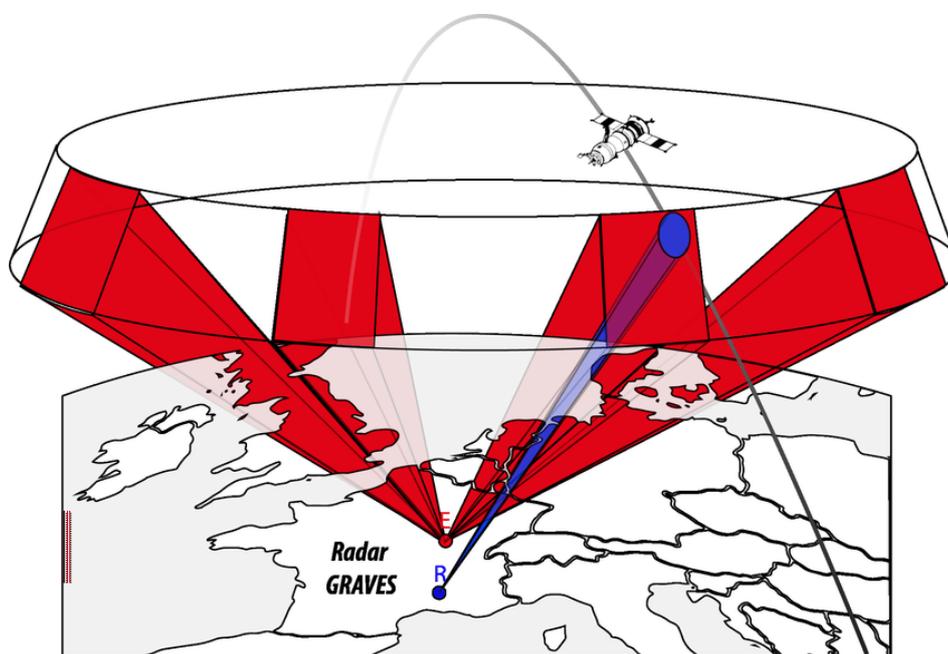


Figure 2-3: GRAVES Bistatic Radar

2.5.2. Small-scale Systems

Hobbyists have created many small systems to track satellites and listen in on transmissions. The hardware varies from commercial off-the-shelf (COTS) components to custom-designed printed circuit boards (PCB). Since small-scale tracking systems are a recent creation, published work on their developments is limited. Trackers which are available off-the-shelf are often proprietary or make use of closed-source hardware and software.

2.5.2.1. NyanSat

The NyanSat challenge was created to help individuals better understand how satellites are tracked by building their own ground station (Nyansat, 2020). It was developed in partnership with the US Airforce and Defense Digital Service (DDS). A kit was provided with an Antenny PCB seen in Figure 2-4 (Nyansat, 2020). The system has an IMU and servo motor driver, with provisions to add a GPS and organic light-emitting diode (OLED) display. The user was required to add a few items, such as servo motors, an SDR and an antenna. The system can track satellites, as well as receive radio transmissions. A demonstration of the system was streamed from the DefCon conference in August 2020 (Red Balloon Security, 2020). A panel of judges from organisations like the National Security Agency (NSA), US Air Force and NASA decided on the best creations.

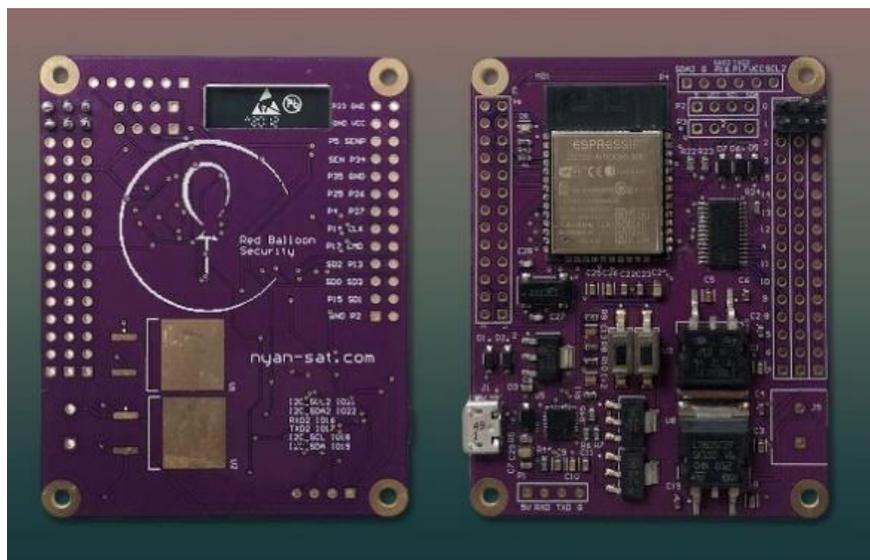


Figure 2-4: Antenny PCB V2

2.5.2.2. Satellite Networked Open Ground Stations (SatNOGS)

Dan White presented the SatNOGS rotator and controller at the 2018 Conference on Small Satellites (White, 2018). The aluminium extrusion-based frame resembles the existing system, seen below in Figure 2-5 (SatNOGS, 2020). The rotator base allows various antenna types to be mounted to the system. The azimuth and elevation axes are controlled using gears, direct current (DC) motors, and various 3D printed parts. It uses the rotctl and rigctl control daemons for both the rotator and SDR, respectively.



Figure 2-5: SatNOGS Rotator

2.5.2.3. SATRAN Antenna Rotator

The SATRAN rotator is a small, Wi-Fi controlled satellite antenna rotator. The device in Figure 2-6 (SATRAN, 2021), can be ordered as a kit and has an enclosure with parts that can be 3D printed. The custom controller board allows for app control via a local Wi-Fi connection. The user can select a satellite to track via the app. The coordinates for the selected satellite are updated relative to the built-in GPS of the connected smartphone. The universal mounting base allows various antenna types to be attached to the rotator.

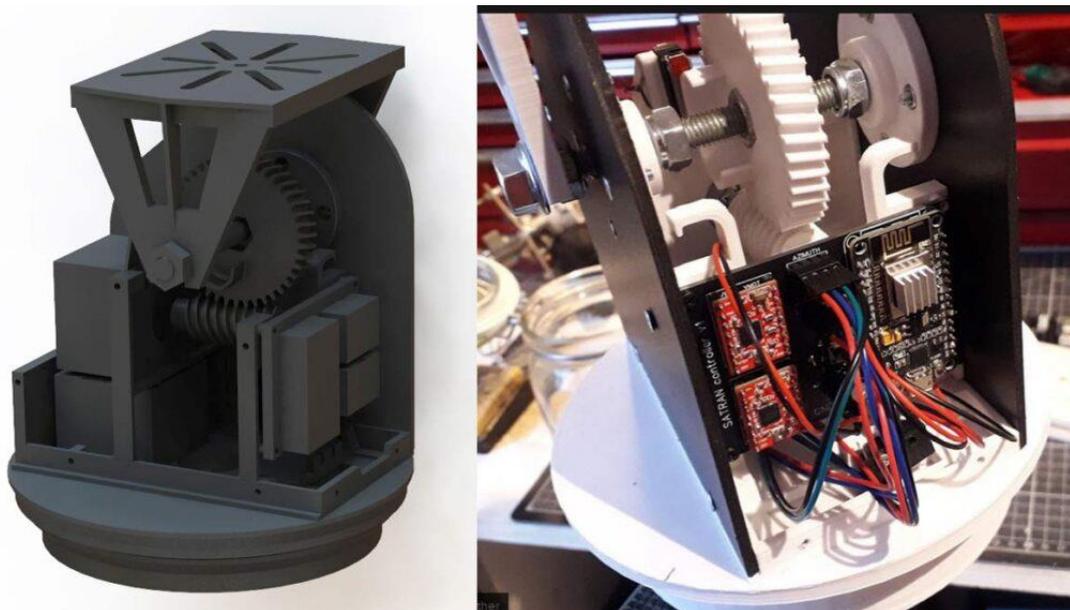


Figure 2-6: SATRAN Antenna Rotator

2.5.2.4. ABB Rotator

An IEEE publication by Reza Chalak Qazani, et al. (2021) detailed the process of creating an antenna rotator using an ABB IRB 6600 industrial robot. The robot has 6-degrees-of-freedom (DoF), enabling precise motion and position control. The team utilised MATLAB and Simulink to interact with the IRC5 controller to send motion commands. An IMU was used to provide accuracy by analysing the orientation of the arm and providing feedback. Figure 2-7 shows the ABB robot with an antenna attached to the tool head (Reza Chalak Qazani, et al., 2021).



Figure 2-7: ABB Robot with Antenna

2.6. Tracking Technologies

Whilst many methods exist to track objects, some technologies are better suited to tracking and monitoring objects in space. The various types are explored in the following subsections.

2.6.1. Radio

Radiolocation is a term used to describe the tracking of an object using radio waves. Using an array of three or more antennas allows a signal to be triangulated. Authorities often use these systems to locate malfunctioning equipment or rogue devices disrupting the spectrum.

2.6.1.1. RTL-SDR Array

A recent publication from Aalto University shows a system capable of determining the direction of arrival (DOA) (M. Laakso, 2021). The array of 21 antennas can be seen in Figure 2-8. The system can detect radio waves within a range of 6 m and a 180° area in both the azimuth and elevation (RTL-SDR, 2021). The Multiple Signal Classification (MUSIC) algorithm is combined with this custom multichannel SDR to provide radio direction finding (RDF) capabilities.

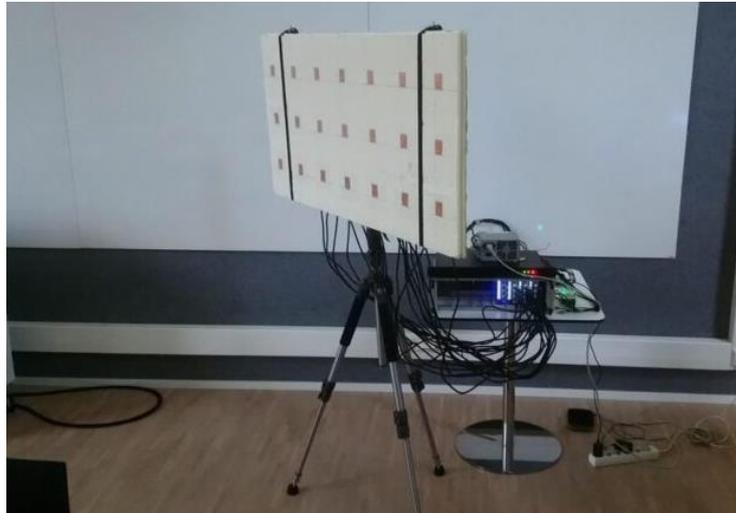


Figure 2-8: 3 x 7 Antenna Array

2.6.1.2. Radio Frequency Sensing

A publication in the 2020 Military Communications and Information Systems Conference overviews the development of a radio frequency sensing system (Islam, et al., 2020). A four-channel SDR and Yagi antenna allows quadrature (IQ) data to be recorded from satellites such as NOAA 18 and AISat-1. These satellites were used for analysis due to their continuous broadcast of frequency transmissions.

The Doppler shift and angle of arrival for low Earth-orbiting satellites can be determined using the algorithm in Appendix D – Code - Doppler Shift (Islam, et al., 2020).

2.6.1.3. KrakenSDR

The KrakenSDR is a crowdfunded device capable of RDF. The system utilises five RTL-SDR dongles and five antennas to create a beamforming array, as visualised in Figure 2-9 (Crowdsupply, 2021). The approximate direction and distance of the signal source are determined using Doppler shift analysis. An Android application can overlay the signal data onto a map, allowing for much easier navigation toward the source of the signals.



Figure 2-9: KrakenSDR

2.6.1.4. TRANET Doppler Tracking System

The US Navy created TRANET in 1959. It was a tracking system that utilised radio signals from satellites. Figure 2-10 (Dunnell, 1967), shows the sites where various base stations were positioned. Each station used high-stability oscillators to ensure that the data was accurate and in sync to calculate the doppler frequencies. The Doppler data and other information were recorded on tapes for further processing (Dunnell, 1967).

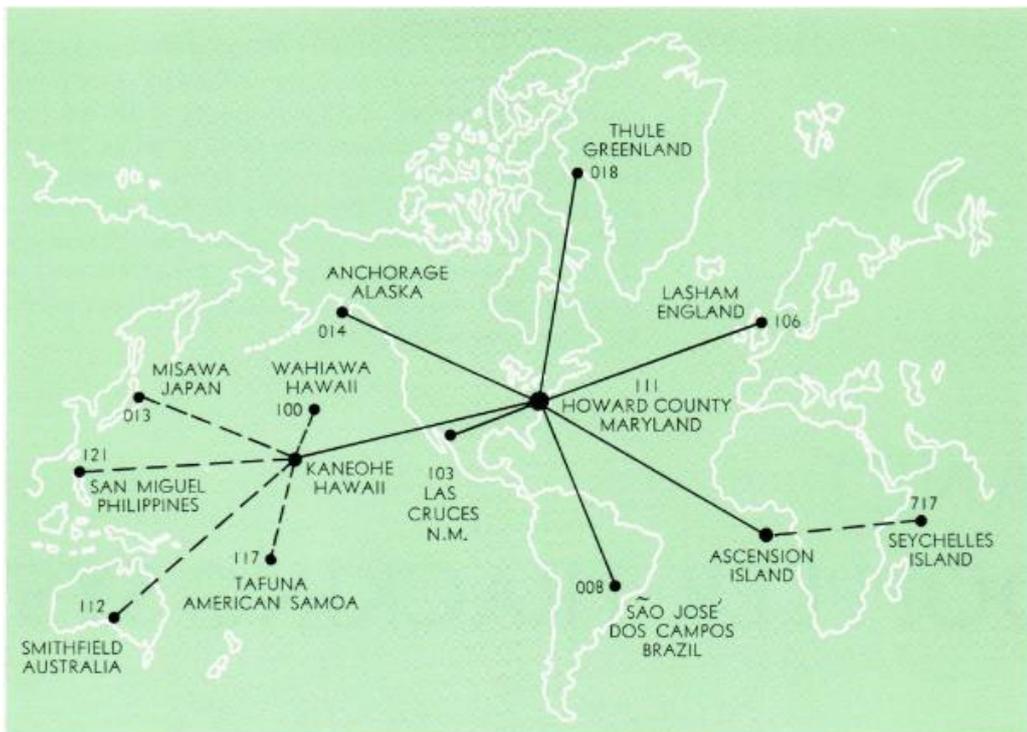


Figure 2-10: TRANET Network

2.6.2. Visual

In a *Monthly Notes of the Astronomical Society of South Africa* (MNASSA) journal, Greg Roberts stated that the best time to observe satellites by eye is just after twilight or before dawn. These conditions make it ideal as the dark sky provides contrast against the bright satellites reflecting sunlight (Roberts, 2007). The addition of light-emitting diodes (LEDs) or reflectors can also aid in identifying and tracking certain satellites (Kovo, 2020).

2.6.2.1. FireOPAL

FireOPAL is a low-cost optical tracking system. It was presented at the *Advanced Maui Optical and Space Surveillance Technologies Conference* (Bland, et al., 2018). It was developed in partnership with Lockheed Martin Space and Curtin University, Australia. Each observatory is equipped with a multi-core computer, 36 MP camera, 3G/4G, 30 TB hard drive, solar panel, and backup battery. This array of features allows a station to be set up wherever required, as shown in Figure 2-11 (Bland, et al., 2018).

The only reliance is a network connection to communicate with the server. Eleven observatories were deployed for testing throughout Australia, with plans to increase the production if required. At night, an exposure is captured every five seconds before being processed onboard or uploaded to the server. The images are synced and analysed to determine spacecraft and their orbits.



Figure 2-11: Deployment of two FireOPAL observatories

2.6.3. Radar

Ground-based radar, with frequencies in the S-band, between 2.0 and 4.0 GHz, is often used for air surveillance. Pulses of electromagnetic waves are transmitted in the direction of operation. When these waves encounter an object, they are reflected. Receivers detect the direction and speed of incoming waves that are reflected off the object (Pun, 2021).

The Space Fence radar system uses S-band radar. The GRAVES system operates at 143.050 MHz. With an appropriate antenna and SDR, anyone in Europe can detect the reflected signals from passing satellites.

2.6.4. Laser

Objects in space are detected via a ranging technique using a high-powered laser. The time taken for the laser beam to reflect is used to determine the location. Recent developments allow non-reflective objects to be detected with a 60% accuracy (European Space Agency, 2019).

2.7. Orbital Mechanics

Understanding how satellites are referenced in space relies on knowledge of the coordinate systems used. Figure 2-12 (Celestis, Inc., 2021) visually represents the azimuth and elevation axes. Elevation can be described as the height above the horizon, while azimuth can be described as a position along the horizon.

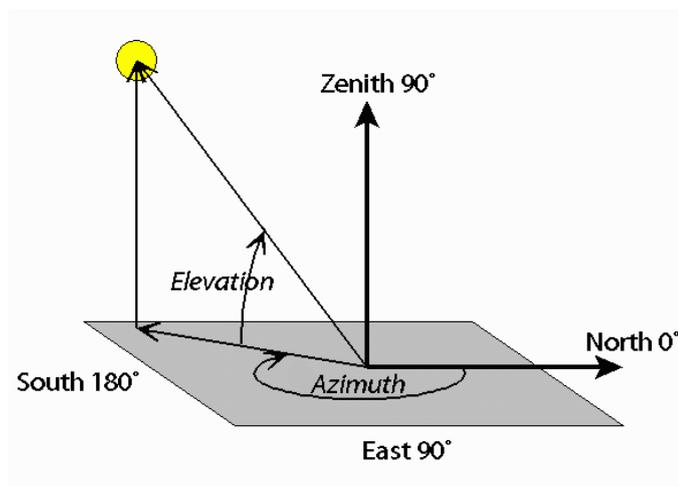


Figure 2-12: Diagram showing Azimuth and Elevation

2.7.1. Orbit Types

Satellites are put into various orbits depending on their use, as seen in Figure 2-13 (cmglee, 2021). While all orbits serve a significant purpose, a select few are important in the realm of space tracking and reducing space debris.

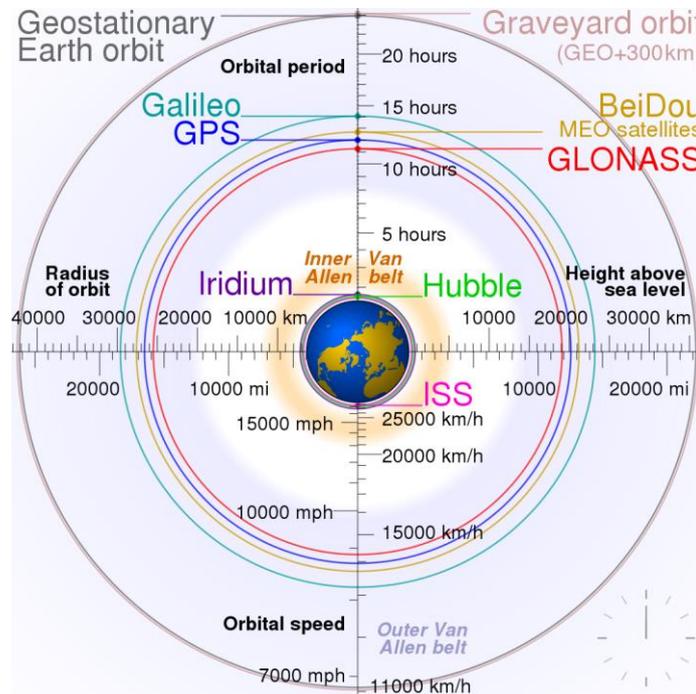


Figure 2-13: Comparison of Orbit Types

Objects in Low Earth Orbit (LEO) are at an altitude between 160 and 1000 km. Imaging satellites use this orbit to take high-resolution photos of the topography. This orbit also allows for lower latency in transmissions. However, due to the low altitude and increased speed, the range of usable coverage is reduced when compared to satellites in higher orbits (European Space Agency, 2020)

Many communications, television and meteorological satellites orbit the Earth using Geosynchronous equatorial orbit (GEO). Orbiting at an altitude of 35 786 km above the equator results in a rotational period of 23 hours and 56 minutes. This allows satellites to rotate with the Earth and provide coverage to the same areas (European Space Agency, 2020).

Once a satellite exceeds its usable lifespan, it is decommissioned. Objects in LEO can be brought down into the ocean or burned up in the Earth's atmosphere. However, objects in GEO are too high to recover. Therefore, satellites launched into GEO after 18 March 2002 are required to move to the graveyard orbit at the end of their mission (United Nations, 2010). Sufficient energy and fuel are required to facilitate the final move of 300 km upwards into the graveyard zone. This allows the desirable GEO zone to remain uncluttered and reduces space debris (E-Signals, 2020).

2.7.2. Two-Line Element (TLE)

Orbital data for a satellite is often represented in a format created by NASA and North American Aerospace Defence Command (NORAD), known as a two-line element. This format documents various parameters about a satellite. Figure 2-14 details the decoding table (CelesTrak, 2019). A TLE from National Oceanic and Atmospheric Administration (NOAA) 14 is shown below.

```
NOAA 14
1 23455U 94089A 97320.90946019 .00000140 00000-0 10191-3 0 2621
2 23455 99.0090 272.6745 0008546 223.1686 136.8816 14.11711747148495
```

Line 1	
Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (Leading decimal point assumed)
54-61	BSTAR drag term (Leading decimal point assumed)
63	Ephemeris type
65-68	Element number
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Line 2	
Column	Description
01	Line Number of Element Data
03-07	Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (Leading decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63	Mean Motion [Revs per day]
64-68	Revolution number at epoch [Revs]
69	Checksum (Modulo 10)

Figure 2-14: TLE Description

2.7.3. Orbital Positions

Due to gravitational attraction, atmospheric drag and drift, satellites rarely remain in a set orbit. These phenomena are perturbations that cause satellites to drift from their orbit between 1-3 km per day. The United States uses military and surveillance satellites and radar systems like the Space Fence to continually map the skies and their objects (Grush, 2019).

The updated TLEs are uploaded to databases on www.space-track.org and www.celestrak.com. The inconsistent behaviour of objects in space means that TLE data is only accurate for about a week before it needs to be updated (AMSAT, 2021).

Converting TLE data into vectors relative to the Earth's coordinate system relies on simplified general perturbation models. Five models exist (SGP, SGP4, SDP4, SGP8 and SDP8), with SGP4 and SDP4 being the most used.

The 1988 report: *Spacetrack Report Number 3: Models for Propagation of NORAD Element Sets* (Kelso, 2021) included the source code for the SGP4 and SDP4 algorithms. This public release of information is what allowed many tracking programs to exist. Using this algorithm allows the future orbital path and location of a satellite to be predicted.

2.8. Satellite Data Download

Satellites use radio waves to communicate to and from Earth. Various types of information are broadcast via these signals, such as voice and images. Frequencies from 136 MHz to 9 GHz are used for various satellite communications (Australian Space Academy, 2021).

2.8.1. Radio

Most satellite transponders have a frequency around the 145 MHz or 435 MHz range. The International Telecommunication Union (ITU) Radio Regulations have approved these bands for use by amateur radio operators (International Telecommunication Union, 2020). These frequencies are used for specific broadcasts, allowing many to listen in on transmissions or download data.

With the frequency of each satellite varying by a few hertz, an SDR is often used for precise tuning whilst allowing coverage from 500 kHz to 1.75 GHz (RTL-SDR, 2021).

2.8.2. Slow-Scan Television (SSTV)

SSTV is a method used to send images via voice frequencies. Various transmission modes allow for colour and larger image sizes. Frequency modulation of the tone is used to create the colour and brightness for each scan line. Based on the mode, it can take from 8 seconds to 240 seconds to render a single frame (Bhattacharya, 2020).

Russian cosmonauts onboard the ISS broadcasted a series of SSTV images in August 2021, seen in Figure 2-15 (The National Association for Amateur Radio, 2021)



Figure 2-15: ISS SSTV Image from August 2021

2.9. Chapter Summary

Analysing the present satellite tracking systems shows significant developments compared to those of the past. Implementation of modern technology, such as the Space Fence, allows for greater accuracy in cataloguing space objects.

Since small-scale tracking systems are a recent development, most of the advancements in this field are made by hobbyists. Subsequently, some useful research and innovations go unpublished. However, due to forums and blogs, these developments are documented for all to see.

Due to the cost and complexity required to create a large-scale system, it is far more effective to use their data to implement a small-scale tracker. The Space-Track database makes precise tracking a possibility due to the regularly updated TLE data. The general perturbation models allow for the azimuth and elevation path of a satellite to be calculated from TLE data. Satellite transmissions can be received by tuning an SDR to the frequency of the transponder.

Subsequently, it was decided to incorporate aspects from the small-scale systems into the proposed system. Radio waves from space are not a reliable way to track satellites due to weather interference and the overlap of satellites using the same downlink frequency. Broadcasting radio, radar, or laser signals to reflect off a satellite would require extremely high-powered hardware and regulatory approval. Therefore, the most reliable method of tracking and monitoring a satellite is through the use of TLE data. Components such as a microcomputer, IMU and GPS are required to gather the necessary positional data. An SDR and antenna are needed to receive and decode radio transmissions. Appropriate software is also required to control the system and interpret data.

Details about the design and development of the system are presented in the following chapter.

3. System Design

3.1. Introduction

This chapter provides insight into the development process of the tracking system. The definition of mechatronics was used as a guide to divide the research into subsystems, while the knowledge gained from the literature review helped to drive design decisions.

3.2. Mechatronic System

The Venn diagram in Figure 3-1 shows the relationship between the various fields that make up a mechatronic system (Western Piedmont Community College, 2022). With each subsystem being interdependent, a change to one aspect will influence the others.

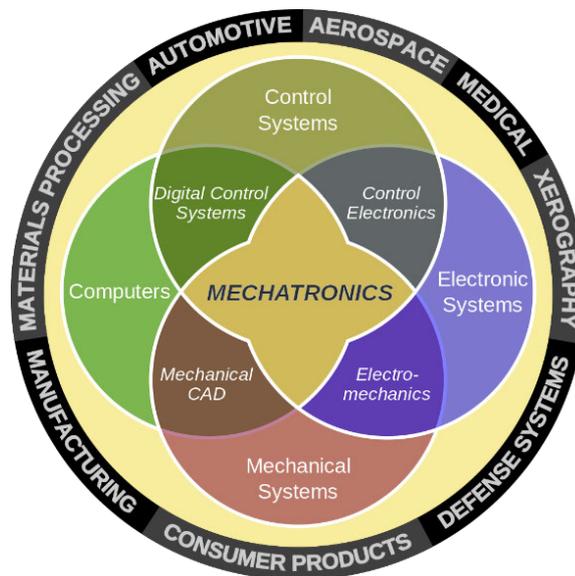


Figure 3-1: Mechatronics Venn Diagram

The definition of mechatronics was used as a reference when dividing the proposed system into smaller subsystems. This segmentation allows for more precise design decisions to be made for each aspect. The system breakdown is as follows:

- Mechanical system – Frame and motion mechanism.
- Computer system – Selection of controllers, components, and sensors.
- Electronic system – Wiring and connection layout for the components.
- Control system – Design and program the UI elements for the tracking system.

3.3. Mechanical Design

The mechanical frame structure was previously created as part of an ongoing research project. Further improvements were made to optimise the capabilities of this subsystem.

3.3.1. Mechanical Design Considerations

The design requirements for the mechanical system are listed in Table 3-1. The existing frame dimensions are 330 mm x 330 mm x 660 mm. The size of the frame was a constraint but not a limitation, as seen from the results in chapter 4.

Table 3-1: Mechanical Design Requirements

Aspect	Description
Movement	360° azimuth rotation, 90° elevation rotation.
Cost	Economical design and component choices.
Electronics	Allocation of space to house the electronics.
Manufacturability	Maximum use of off-the-shelf components.

3.3.2. Mechanical System Decomposition

Figure 3-2 details the subassemblies and components required to construct the mechanical system.

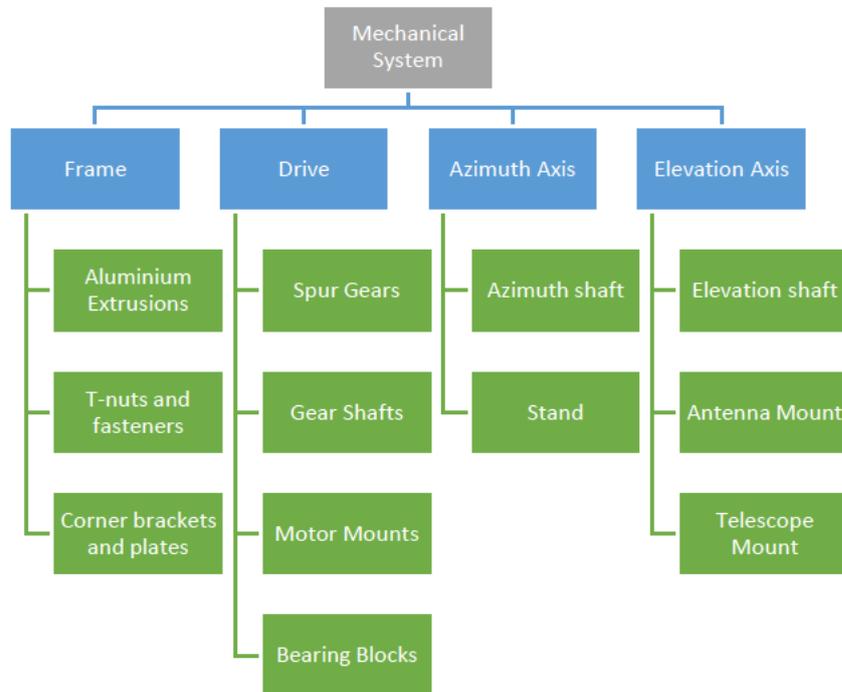


Figure 3-2: Mechanical System Decomposition Diagram

3.3.3. Mechanical Components

There were many subassemblies used to construct the mechanical structure of the tracking system. A breakdown of the subassemblies and their components is mentioned below.

3.3.3.1. Frame

Aluminium extrusions with a 30 x 30 mm profile width were used to create the frame. Figure 3-3 illustrates the dimensions of the profile (Modular Assembly Technology SA, 2022). Using the provided computer-aided design (CAD) models, the system was modelled using SolidWorks.

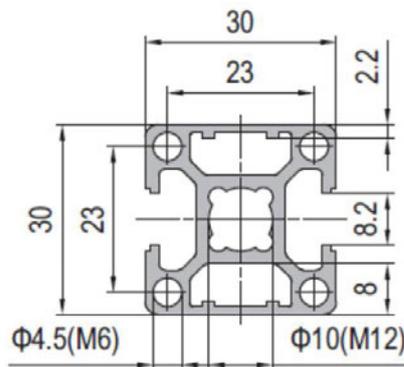


Figure 3-3: PG30 Profile

Twenty-nine extrusion pieces were cut to a length of 330 mm and arranged to create the frame structure. T-nuts, corner joints and aluminium plates were used to secure and align the extrusions. The construction of two cube-shaped enclosures in Figure 3-4 allowed the azimuth and elevation mechanisms to be independent, ensuring that they do not interfere.

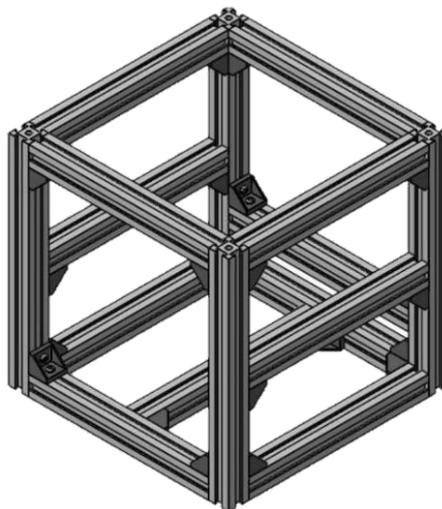


Figure 3-4: Frame subassembly

3.3.3.2. Drive

The drive mechanism was created to facilitate the system's motion. The wiper motor was bolted to a mount fabricated by the workshop. The bearing blocks were pressed onto the shaft after the small spur gear was attached. The workshop also machined a coupling to attach the gear shaft to the motor. Figure 3-5 shows the completed subassembly that is then secured to the frame.

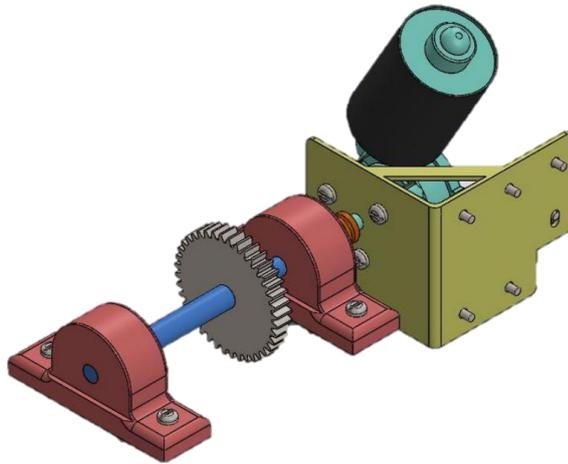


Figure 3-5: Drive assembly

a. Spur Gears

A pair of gears were used to increase the motor torque whilst decreasing the revolutions per minute (RPM). Figure 3-6 shows the 40-tooth pinion spur gear and the 60-tooth driven spur gear. This combination results in a gear ratio of 1.5. The face width is 20 mm with a module of 2. The elevation torque calculations are shown in 6.1.2 - Elevation Torque.

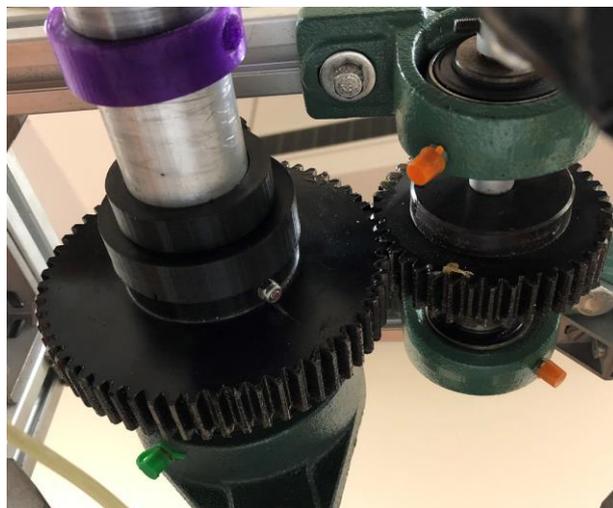


Figure 3-6: Spur Gears

b. Gear Shaft

Sections of the gear shaft were turned to various diameters to facilitate mounting the bearings, gear, and coupling. The diameter of the shaft starts at 7 mm on the left to accommodate the motor coupling. The next step has a 12 mm diameter, which fits into the bearing block. The shaft then steps up to a diameter of 15 mm, to accommodate the gear before reducing to 12 mm to secure the other bearing block. Figure 3-7 shows the shaft profile.



Figure 3-7: Gear Shaft

c. Motor Mount

A steel plate was used to create the motor mount. A laser cutter ensured that the fabricated piece had the required holes in the correct positions before being bent into its final shape, as shown in Figure 3-8.

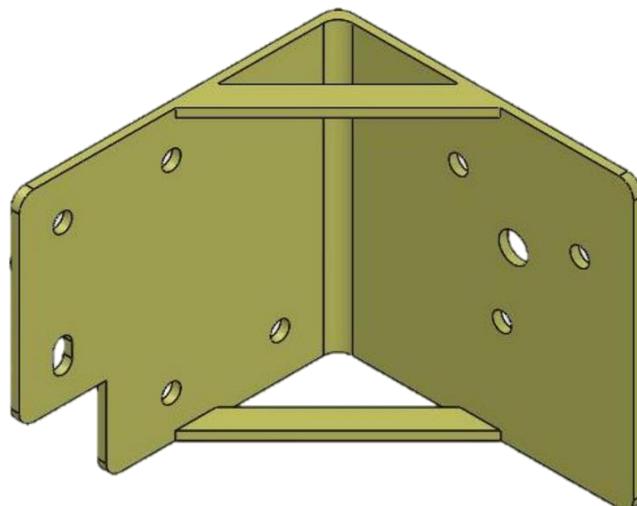


Figure 3-8: Motor Mount

3.3.3.3. Azimuth Axis

The pinion gear was attached to a hollow 38 mm aluminium tube. The bearings were secured to the shaft, allowing the assembly shown in Figure 3-9 to bolt to the aluminium frame.

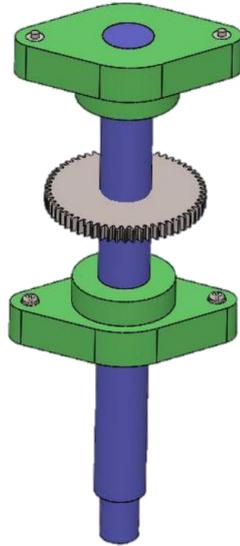


Figure 3-9: Azimuth Shaft

a. Stands

Two steel stands were fabricated to support the frame assembly. The small stand was welded together using L-profiled steel. A 38 mm mount was also welded in to secure the azimuth axis, allowing for 360° rotation. The complete assembly is shown in Figure 3-10.



Figure 3-10: Frame with the Small Stand

Figure 3-11 shows the large stand, which was also welded together using steel L-profiles. Both stands were painted to prevent corrosion. Holes at the top of the large stand allow the small stand to bolt onto it. The larger stand allows the assembly to be further elevated if required.



Figure 3-11: Large Stand

3.3.3.4. Elevation Axis

The elevation axis was also constructed using a spur gear mounted to a hollow 38 mm aluminium tube. Bearing blocks were used to secure the assembly to the frame in Figure 3-12.

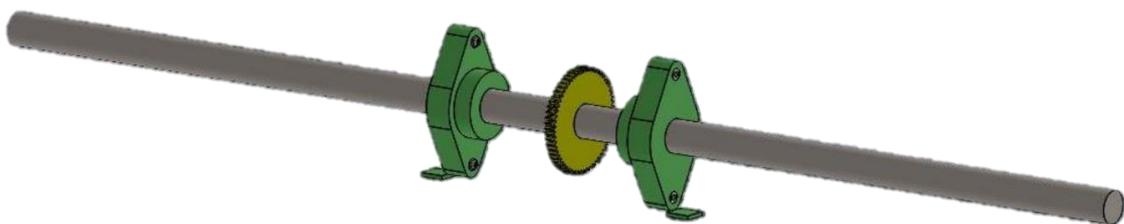


Figure 3-12: Elevation Shaft

3.3.3.5. Mounts

A pair of mounts were machined from aluminium to secure the telescope and antenna to the elevation axis. Bolt holes were threaded to allow the mount to be fastened to the shaft to prevent any movement. Figure 3-13 shows the model of the mount on a shaft.

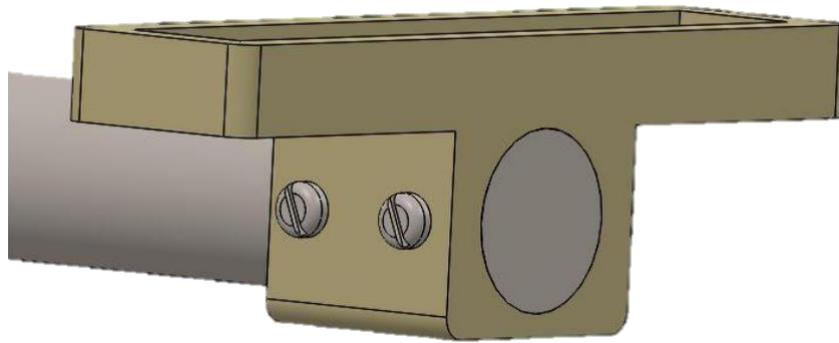


Figure 3-13: Shaft Mount

Fastening the mounts to the antenna and telescope, and attaching them to the elevation shaft, creates the elevation axis seen in Figure 3-14.

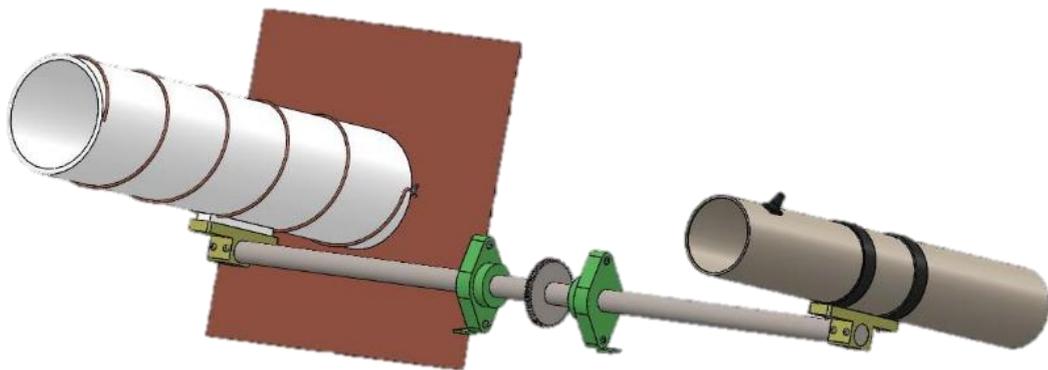


Figure 3-14: Elevation Axis

3.3.3.6. Hole Reducers

The t-nuts used to secure the bearing blocks to the frame require an M6 bolt. However, the bearing block holes are manufactured to accommodate an M12 bolt. Figure 3-15 shows the simple reducer that was created to allow M6 bolts to fit in the M12 sized holes. The addition of a washer and spring washer secured the bolt to the t-nut.

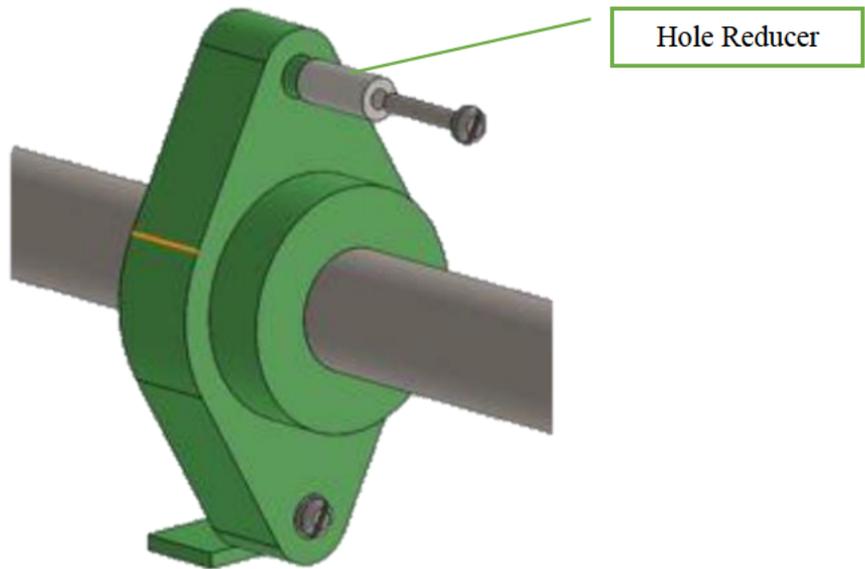


Figure 3-15: Hole Reducer

3.3.3.7. IMU Bracket

A bracket was designed to mount the IMU to the elevation shaft, Drawing B1. Figure 3-16 displays the 3D printed mount. Polyethylene terephthalate glycol (PETG) was used to prevent any interference that may occur with a metallic mount.

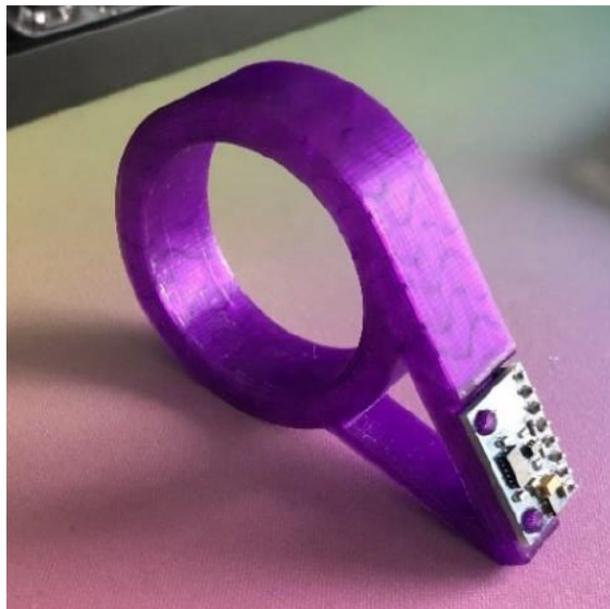


Figure 3-16: IMU Bracket

3.3.4. Mechanical Part Selection

The major COTS components used in the mechanical system are listed in Table 3-2:

Table 3-2: Mechanical COTS Components

Component	Supplier	Quantity
40 tooth spur gear	RS Components	2
60 tooth spur gear	RS Components	2
40 mm bearing block	Zhous Enterprise	4
12 mm bearing block	Zhous Enterprise	4
PG30 aluminium extrusions, 330 mm	Modular Assembly South Africa	29
PG30 90° bracket	Modular Assembly South Africa	52
M6 t-nut	Modular Assembly South Africa	170
M5 x 25 bolt	SA Fasteners	170
M6 x 10 bolt	SA Fasteners	8
M6 x 20 bolt	SA Fasteners	8
M6 x 35 bolt	SA Fasteners	16

3.3.5. Dimensions

The assembled parameters of the system are as follows:

- Frame width: 330 mm
- Frame depth: 330 mm
- Frame height: 660 mm
- Total width (with antenna): 1.5 m
- Total depth (with antenna): 1.1 m
- Total height (small stand): 1.8 m
- Required torque: 17.92 Nm

3.4. Electronic Integration

The satellite tracking mechanism required automation and control. A microcontroller was needed to manage the operation of various systems, with suitable sensors providing the required data.

3.4.1. Electronic Integration Considerations

Using information gained from the literature review, design requirements for the electronic system were drawn up. The details are listed in Table 3-3.

Table 3-3: Electronic Integration Requirements

Aspect	Requirement
Control computer	Allow for remote access, control serial devices, and support a UI.
Microcontroller	Control motor drivers, serial devices, and sensors.
SDR	Software controlled radio tuning.
GPS	Provide latitude and longitude.
IMU	Orientation feedback.
Motors	Accurately move each axis.

3.4.2. Electronic Components

Various components and devices need to interface with each other to provide control. The following subsections explore the components that may be suitable for use.

3.4.2.1. Control Computer

A device was required to process the information for the TLE data, allow remote access to the system, and have capabilities to allow the SDR and microcontroller to interface with it.

a. Raspberry Pi 4

The Raspberry Pi is a multipurpose Single Board Computer (SBC). It has 40 general-purpose input-output (GPIO) pins capable of Pulse Width Modulation (PWM), I2C or serial. The ARM-based processor is highly capable and supports many types of operating systems. The large community of developers and creators are a major source of information. An array of ports allows for components such as Universal Serial Bus (USB) devices, cameras and displays to be connected. Network communication can be established through Bluetooth, Wireless Fidelity (Wi-Fi) or ethernet connections, as seen in Figure 3-17 (Raspberry Pi Foundation, 2021).

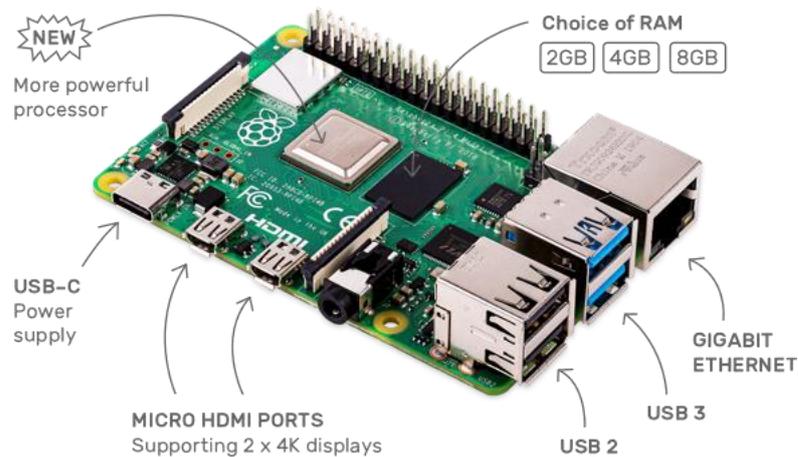


Figure 3-17: Raspberry Pi 4B Board

b. Rock Pi 4

Another Raspberry Pi inspired SBC is the Rock Pi 4. It features the same ports as the Raspberry Pi, with built-in Wi-Fi and Bluetooth, as seen in Figure 3-18 (Radxa Limited, 2019). The addition of a more powerful RK3399 64-bit six-core processor makes it more capable in both CPU and GPU intensive tasks (Radxa Limited, 2019).



Figure 3-18: Rock Pi 4 Board

c. Banana Pi M5

Figure 3-19 shows the Banana Pi; an SBC built with inspiration from the Raspberry Pi (Sinovoip, 2018). The GPIO header is identical to that of the Raspberry Pi. It shares the same footprint as the Raspberry Pi, and while it has built-in flash storage, it lacks a camera port. The main difference is that Wi-Fi and Bluetooth connectivity requires a separate add-on board (Sinovoip, 2018).

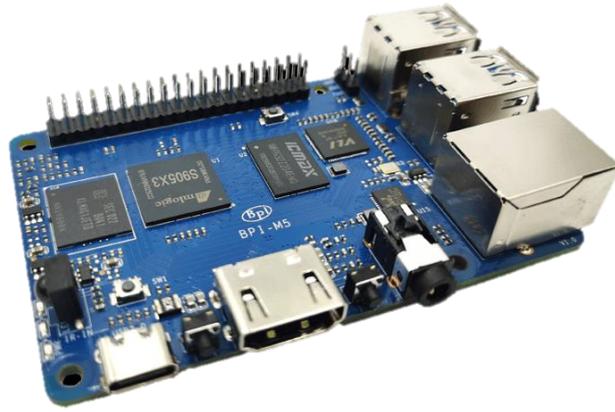


Figure 3-19: Banana Pi M5 Board

3.4.2.2. Microcontroller

Interfacing between components required a controller to manage the PWM, analogue data, and digital data.

a. Arduino Pro Micro

The Arduino is designed to make prototyping simple and easy (Arduino, 2020). Its versatility makes it an ideal platform for managing the input and output (I/O) from each component. The Pro Micro has a small 330 x 178 mm footprint. Figure 3-20 shows the 8-bit ATmega32U4 processor, 18 I/O pins and a built-in USB interface (Robotistan, 2022).

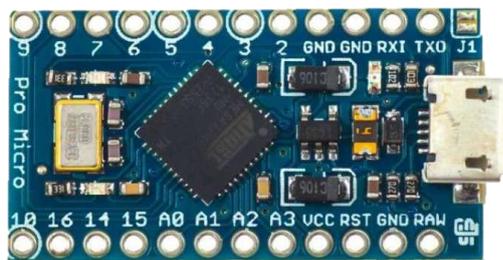


Figure 3-20: Arduino Pro Micro

b. Teensy

With most Arduino models utilising a 10-year-old Atmel ATmega328P processor, there is a growing market for faster and more powerful microcontrollers. Teensy is based on the 32-bit ARM-Cortex M4 to M7 processor. It is highly capable with 42 I/O pins, 512 kB of storage, and a small 15 x 50 mm form factor. Figure 3-21 illustrates the similarity of the board to the Arduino (PJRC, 2022). Using the Teensy software add-on allows most Arduino projects to be translated for use without modification (Sanghvi, 2021).



Figure 3-21: Teensy 4.0

c. ESP32

The ESP32 is a low powered microcontroller focused on the Internet of Things (IoT) integration. Built-in Wi-Fi, Bluetooth, touch, temperature and hall-effect sensors complete the feature set, which makes it a versatile board. The 32-bit Xtensa dual-core microprocessor and 34 I/O pins can be seen in Figure 3-22 (Espressif Systems, 2021).

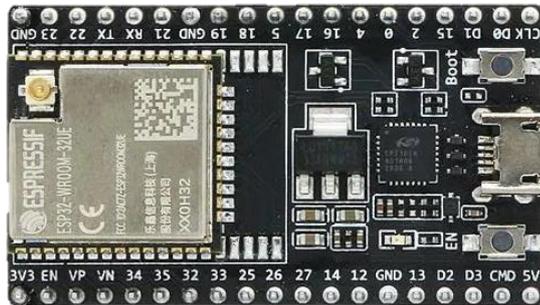


Figure 3-22: ESP32-WROOM

3.4.2.3. SDR

A radio receiver was required to obtain the transmissions broadcasted from various satellites.

a. RTL-SDR

The RTL-SDR is a software-defined radio tuner. Software control of digital components such as modulators and tuners allows for frequencies from 500 kHz to 1.75 GHz to be received and processed (RTL-SDR, 2021). This makes it ideal for receiving broadcasts from various satellites across all frequency ranges.

b. Airspy R2

The Airspy R2 is an SDR with a wide range of features. It can receive super high frequency (SHF) signals, with the built-in real-time clock (RTC) allowing the synchronisation of data from multiple antennas (Airspy, 2021).

3.4.2.4. Global Positioning System (GPS)

The geographical location of the satellite tracker needs to be known to calculate the new orbital paths from updated TLE data. By utilising a GPS sensor, precise longitude and latitude values can be obtained. Most GPS modules require a signal lock with four satellites to determine location coordinates. A direct line of sight is critical to obtaining a fix with GPS satellites, making it unsuitable for indoor use. With minimal variation between the technology used in available modules, most will be suitable for use (Day, 2020).

3.4.2.5. Inertial Measurement Unit (IMU)

The IMU is a sensor used to determine the orientation of an object. Depending on the IMU type, it can have various sensor chips, such as an accelerometer, gyroscope, and magnetometer MPU-6050 or LSM303DLHC. With the addition of a barometer or thermometer on the MPU-9250 sensor, 10 DoF can be measured (SparkFun Electronics, 2021). This motion data aids in determining the speed and position of the azimuth and elevation axes.

3.4.2.6. Motors

The motors were required to have significant torque to move the axes whilst providing the ability to maintain a position.

a. Stepper Motor

Stepper motors rotate in steps using multiple phases. A stepper can achieve high levels of precision depending on the number of phases to achieve step angles from 0.72° to 1.8° . A stepper driver is required to provide voltage and current control of each winding. Energising all the coils enables the motor shaft to lock in place (RS Components (SA), 2022).

b. Wiper Motor

A wiper motor is comprised of a DC motor attached to a gearbox. The worm gears provide significant torque whilst preventing the output shaft from rotating when the motor is not powered. Their speed and direction can be controlled with a simple H-bridge driver (Terry, 2019).

3.4.3. Electronic Component Selection

While there were many suitable items for each task, cost and stock availability were significant factors affecting the final selection.

Table 3-4 lists the COTS parts chosen for the electronic system and the justification for their use.

Table 3-4: Electronic COTS Components

Component	Specification	Justification
Microcomputer	Raspberry Pi 4B	<ul style="list-style-type: none"> • Built-in Wi-Fi and IO control pins. • Camera port. • Reasonable price to performance. • Ongoing software updates and support.
Microcontroller	Arduino Pro Micro	<ul style="list-style-type: none"> • Known compatibility with Raspberry Pi. • Interfaces with many sensors. • Widely used platform with many tutorials.
Motor Driver	Pololu md07a	<ul style="list-style-type: none"> • Rated for 15A loads. • Overcurrent protection. • Overtemperature protection.
Motor	12 V wiper motor	<ul style="list-style-type: none"> • Cost-effective. • High torque. • Simple control. • Does not require power to hold its position.
Voltage Regulator	12 V to 5 V	<ul style="list-style-type: none"> • Used to provide 5V to selected components.
Fuse Box	4 Way	<ul style="list-style-type: none"> • Isolates the four circuits in the system.
IMU	LSM303DLHC	<ul style="list-style-type: none"> • Provides the desired 6-DoF.
GPS	Adafruit Ultimate GPS	<ul style="list-style-type: none"> • 10 Hz update rate. • Serial interface.
Radio receiver	RTL-SDR V3	<ul style="list-style-type: none"> • Economical. • USB powered and controlled. • External antenna port.
Fan	12V Axial	<ul style="list-style-type: none"> • Provides airflow to cool components via convection.

3.4.4. Circuit Design

To better understand the power and signal connections of the system, two circuits were drawn up. A basic power flow diagram was created using the Scheme-it circuit designer from Digi-Key Electronics, shown in Figure 3-23 (Digi-Key Electronics, 2021).

Figure 3-24, created using Fritzing, shows the signal connections (Fritzing, 2021).

3.4.4.1. Power

A calculation of each component's power requirements in Appendices

The following appendices contain calculations, drawings, results and code used in this report.

Appendix A – Calculations indicated that the system could draw a maximum of 145W from the 12V battery. Since a substantial amount of current was to be drawn, heavy gauge wiring was needed for some parts of the system. The motors require 12V, while the microcontrollers use 5V. Subsequently, regulators were used to lower the voltage. Fuses were placed on each circuit to provide isolation and protection from shorts. Figure 3-23 shows the layout of the connection between these components.

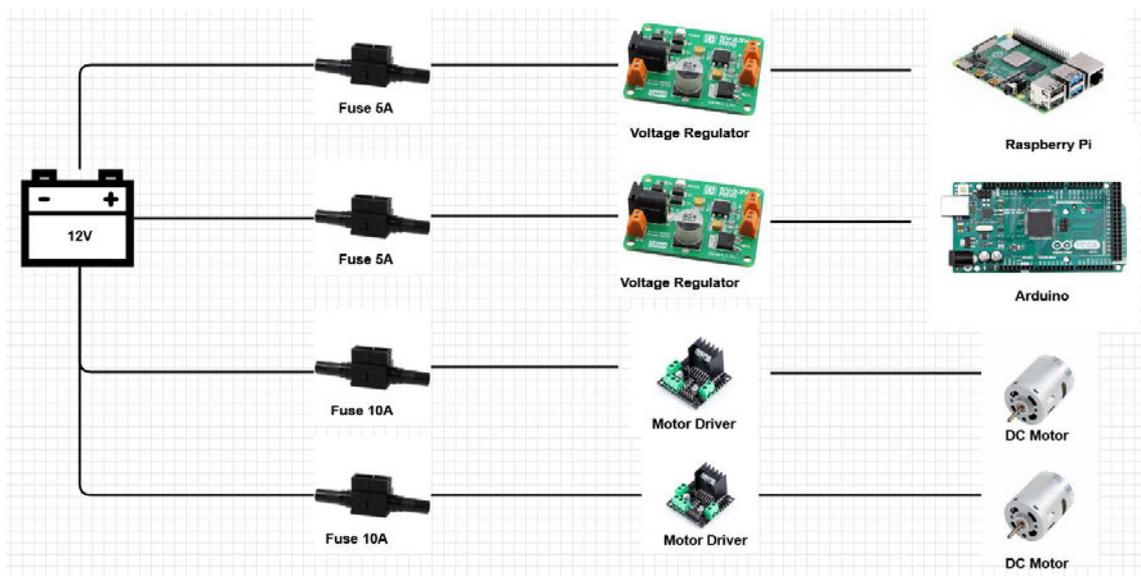


Figure 3-23: Circuit Power Diagram

3.4.4.2. Signal

The Raspberry Pi acts as the central control computer with the other devices connecting to it. Both the Raspberry Pi and Arduino have a single serial input port. This limitation meant that the IMU and GPS would have to be on separate boards. With the Arduino as the rotator controller, the IMU would be best suited to it, with the GPS being connected to the Raspberry Pi. Figure 3-24 illustrates the proposed wiring diagram.

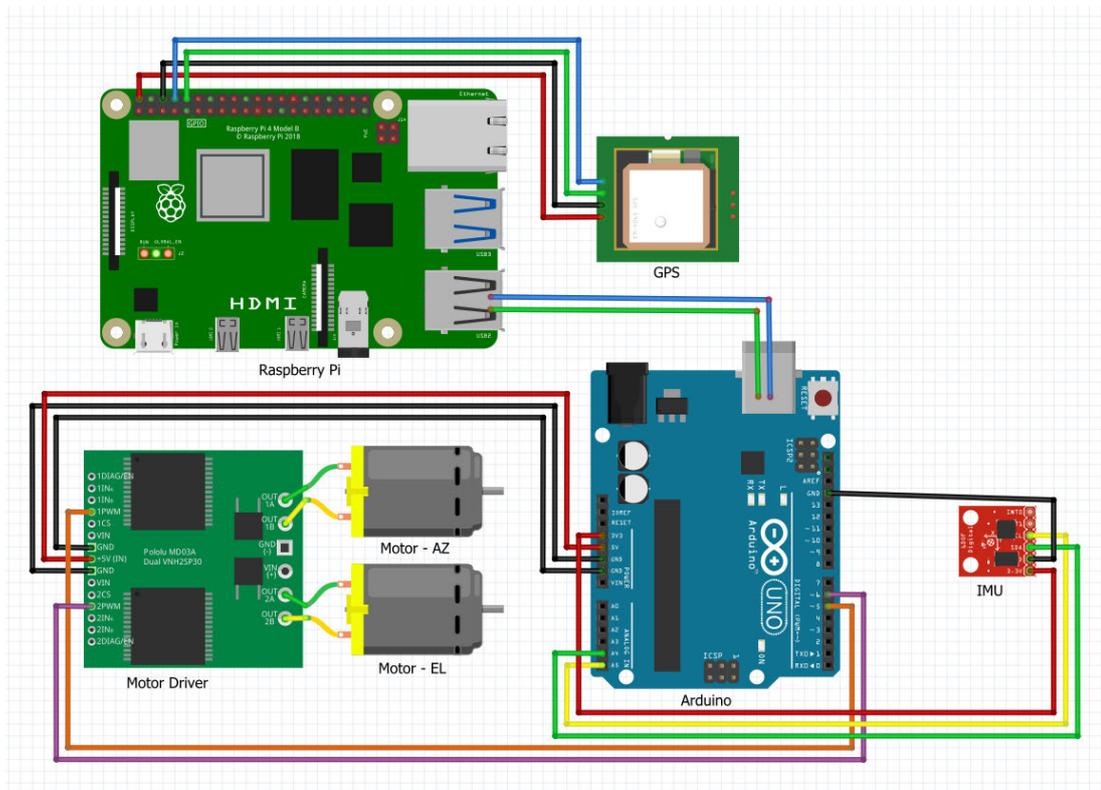


Figure 3-24: Signal Connection Diagram

a. Raspberry Pi

The GPS module can be connected to the Raspberry Pi using the GPIO pins. Since the device is to be used via serial communication, the Rx and Tx pins must be crossed when connecting it up.

b. Arduino

The motor drivers required a direction and speed signal from the Arduino. The IMU transmits data from the gyroscope and magnetometer to the Arduino.

3.4.4.3. Inter-device

To exchange information between the Raspberry Pi and Arduino, there needed to be a method of communication with each other. Data telling the motors to move a specific position needed to be sent from the Raspberry Pi to the Arduino, while information from sensors such as the IMU needed to be sent from the Arduino to the Raspberry Pi for analysis.

Figure 3-25 was created in Fritzing and shows the possible communication link via the serial pins on both devices. However, this method would require a logic level converter as the Raspberry Pi pins operate at 3.3V, whilst the Arduino pins are 5V. A much easier and more reliable way of accomplishing the same task was to use a USB cable between them.

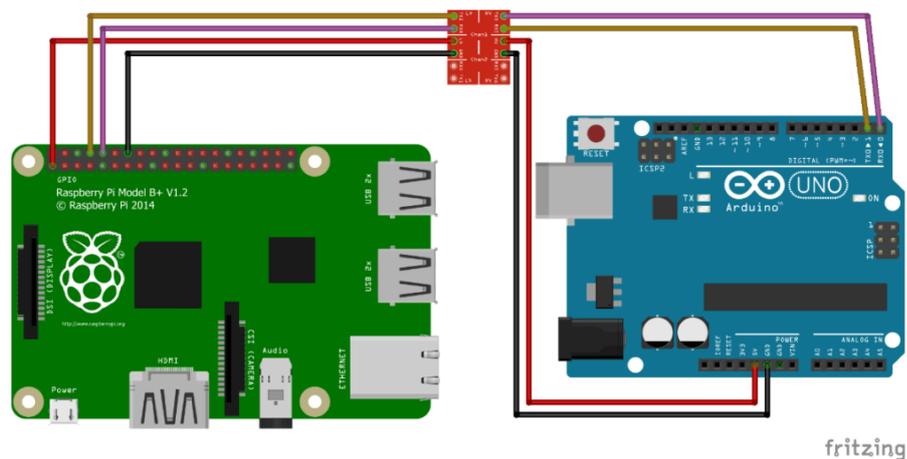


Figure 3-25: Serial Communication via GPIO

3.4.5. Internet Connectivity

The Raspberry Pi has both Wi-Fi and an Ethernet network port. The addition of two methods to connect to the Internet allows for greater flexibility. The Ethernet port is rated at 1 Gigabit. The wireless radio is rated at the Institute of Electrical and Electronic Engineers (IEEE) 802.11ac standard (Raspberry Pi Foundation, 2021). Both methods provide sufficient bandwidth for remote access and streaming.

3.5. Software Design

Various sensors and components were required to interface with each other, so a method of integrating them into one system needed to be developed. A user interface was needed for certain aspects of the system.

3.5.1. Software Design Considerations

The requirements for the software system were drawn up. The details are listed in Table 3-5.

Table 3-5: Software Design Requirements

Aspect	Requirement
Programming language	A language that is the most compatible.
Tracking Platforms	Software that provides the necessary TLE data and computations.
Rotator Control	Send and receive commands to move the rotator.
Radio Control	Software control of SDR tuning.
GPS	Provide the necessary output of coordinates.
SSTV	Decode slow-scan image transmissions.
GUI	Easy to use interface for all users.

3.5.2. Programming Languages

The Raspberry Pi requires a graphical user interface (GUI) application to be created for its various systems. Many coding languages are supported by the Raspberry Pi, such as C++, Java, HTML and Python (Sean McManus, 2021).

The Arduino makes use of C or C++ for interacting with its microprocessor.

3.5.2.1. C++

The Arduino integrated development environment (IDE) requires either C or C++ to create sketches for deployment on the Arduino. Prior knowledge of programming and using an Arduino allows for a greater understanding when setting up this system.

3.5.2.2. Python

While C or C++ can be used to create a GUI, the compatibility between Windows and Linux has been known to cause issues (poilkjewq, 2018). Subsequently, the decision was made to use Python to create the interface that the user initially interacts with.

The good interoperability of Python libraries between Windows and Raspberry Pi OS streamlines the development and deployment from one platform to another.

3.5.3. Tracking Platforms

TLE data for objects in space are accessible via online databases. Tracking platforms have combined access to these sources, allowing for satellites to be tracked.

3.5.3.1. Gpredict

Gpredict has been the recommended tracking program for many years. The automatic TLE updates allow satellite passes to remain accurate. An informative GUI visualises the selected satellite paths on a world map in Figure 3-26.

The radio and antenna controller make it the most versatile solution for this application. The SGP4 and SDP4 algorithms allow for future pass predictions (Csete, 2021).

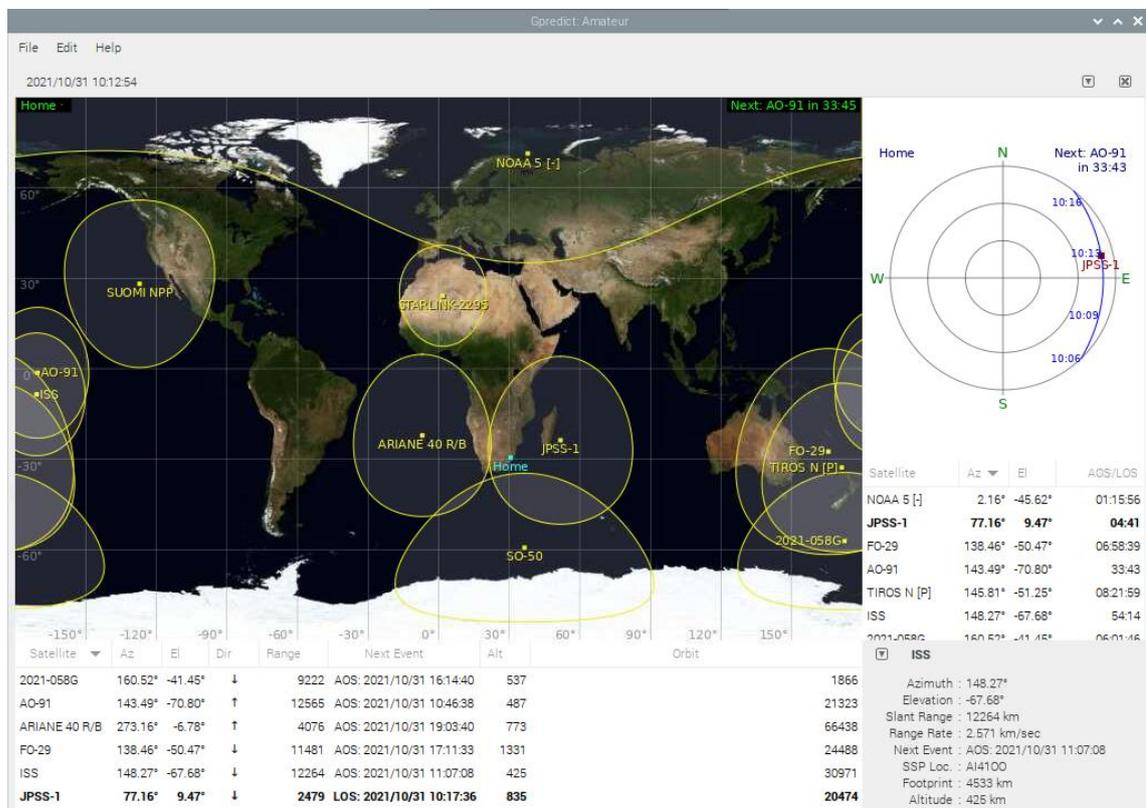


Figure 3-26: Gpredict

3.5.3.2. PREDICT

PREDICT is a satellite tracker with a minimal UI (Magliacane, 2021). It does provide support for radio control and pass prediction. However, many intensive graphical elements are limited to the full-featured Linux version, as seen in Figure 3-27.

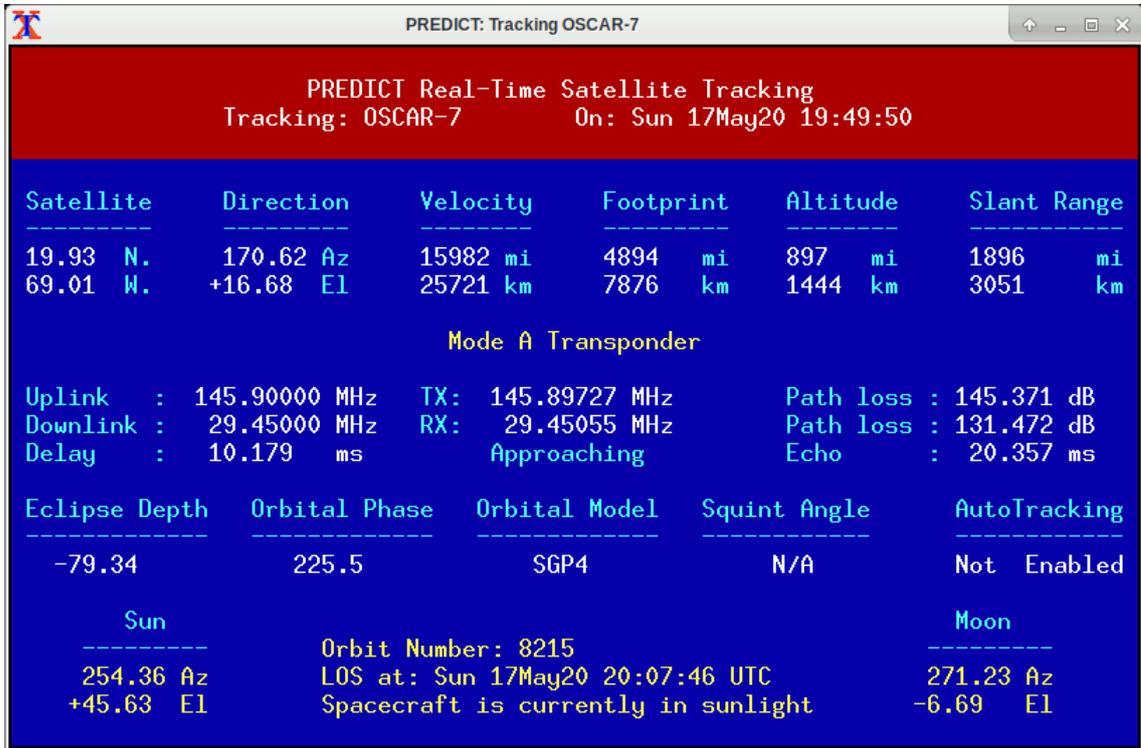


Figure 3-27: PREDICT

3.5.3.3. Other Tracking Software

Many tracking programs work on various operating systems from Windows, macOS, Android and iOS. However, only a few have support for Linux and Raspberry Pi OS. The main limitation is the ARM-based processor. Most intensive calculations and graphics features are not possible on a low-powered mobile-class processor.

3.5.4. Rotator Control

Moving the azimuth and elevation axes to the desired position required control of the motors and sensors such as the IMU.

3.5.4.1. Hamlib

Hamlib is a library used to control both radios and rotators. The Hamlib Application Programming Interface (API) is an intermediary software layer that interfaces with many radios and rotators. Figure 3-28 illustrates the position of the API in a system (Hamlib, 2021).

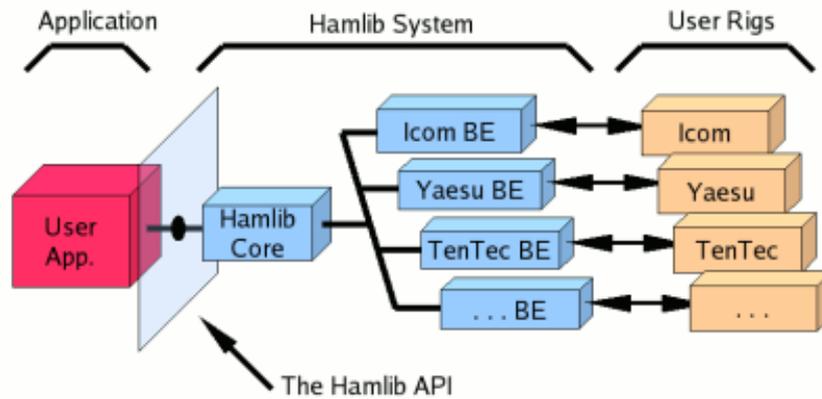


Figure 3-28: Hamlib API

3.5.4.2. Rotctl

Rotctl is a Hamlib rotator control daemon. It sends and receives commands between the software and rotator hardware. Details such as the azimuth and elevation can be sent to the rotator, with positional information from the IMU being sent back (Hamlib, 2021).

3.5.4.3. EasyComm

Many off-the-shelf rotators from Yaesu and Celestron have native support for interfacing with Hamlib. The current rotator system utilises non-standard hardware, so a custom protocol is required to facilitate device communication. EasyComm II is a protocol that allows control and feedback from a custom rotator system. Once implemented, rotator control software such as Gpredict will be able to recognise the rotator as an EasyComm II compatible device and take control of the system (Jackson, 2021).

3.5.4.4. SARCNET Rotator

A method was required to get the two systems working. The team from SARCNET created a rotator controller using an Arduino (Gonzales & Gonzales, 2018). The software was modified to suit the current hardware configuration of the system. It implements the EasyCommII protocol, allowing the Arduino and subsequent hardware to be recognised as a controllable rotator device. Figure 3-29 shows the simplified process of the rotator control.

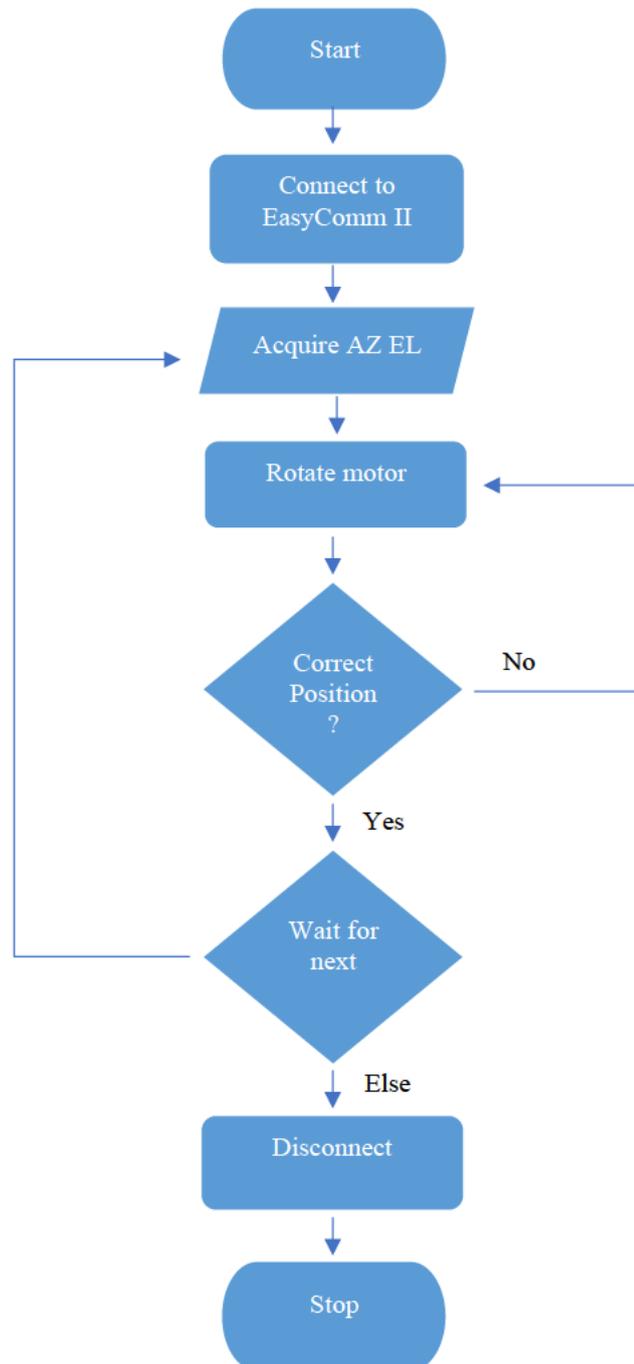


Figure 3-29: Flowchart of Rotator Process

3.5.5. Radio Control

The RTL-SDR is a USB device, allowing it to be controlled similarly to the rotator. Since the device is designed to be a software-controlled receiver, the driver and a software tuner were required to control the frequency. With the transmission of signals by an unlicensed radio operator being illegal, the RTL-SDR would only be allowed to act as a receiver.

Many SDR tuning software packages are available, but only a handful work with Linux. Being able to record the transmission, as well as providing remote control, is of importance.

3.5.5.1. GQRX

GQRX is a lightweight SDR controller. It features an intuitive UI allowing the signal spectrum waterfall to be visualised (Csete, 2021). The remote-control feature allows other software to change the frequency automatically. Streaming audio across a local network via user datagram protocol (UDP) is a useful way to listen in and record incoming transmissions.

3.5.5.2. SDR++

SDR++ is an SDR tuner with a UI based on the popular Windows-only SDR# (SDR Sharp) (Rouma, 2021). It has advanced features such as a multi-variable frequency oscillator (VFO), allowing the RTL-SDR to receive multiple signals simultaneously. However, the lack of a remote-control feature makes it less desirable for this application.

3.5.6. GPS

GPS modules are receive-only devices. Their location is determined using satellite triangulation. Data is provided in National Marine Electronics Association (NMEA) sentences using serial communication. Using Figure 3-30 (RF Wireless World, 2012), these sentences can be translated for a more straightforward interpretation of the information.

NMEA Sentence	Meaning
GPGGA	Global positioning system fix data (time, position, fix type data)
GPGLL	Geographic position, latitude, longitude
GPVTG	Course and speed information relative to the ground
GPRMC	Time, date, position, course, and speed data
GPGSA	GPS receiver operating mode, satellites used in the position solution, and DOP values.
GPGSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
GPTRF	Transit fix data
GPSTN	Multiple data ID
GPXTE	cross track error, measured
GPZDA	Date and time (PPS timing message, synchronised to PPS).

Figure 3-30: NMEA Sentence Description

3.5.1. Frequency Identification

Locating the source of a radio signal requires, at minimum, a trio of antennas and receivers (Vitaliy, et al., 2011). Frequency path identification is an alternative that is possible with the current system, which consists of a single antenna and receiver.

STRF is a toolkit that matches an orbiting satellite to a radio frequency (Bassa, 2021). IQ data for the desired frequency is recorded via the RTL-SDR and is processed using the rffft tool to produce a spectrogram. The rffplot tool allows the target signal to be visualised and selected. Finally, the rffit tool is used to analyse the isolated signal. Aspects such as the Doppler shift are compared to determine the best fitting orbit and satellite for that recording frequency.

This implements a similar approach as the previously mentioned paper: *Doppler Estimation for Passive RF Sensing Method in Space Domain Awareness* (Islam, et al., 2020).

3.5.2. SSTV

Certain satellite transmissions contain slow scan image broadcasts. An image is transmitted using frequency modulation. Once the audio is received, it can be decoded by a utility such as QSSTV (Everberg, 2021). Depending on the encoding method and complexity of the broadcasted image, a transmission can be a few seconds to minutes for a single frame.

3.5.3. Graphical User Interface (GUI)

Creating a GUI will aid the user in navigating through the software, allowing easy access to the required elements of the system.

3.5.3.1. Design Parameters

- Login system with user hierarchy.
- Run at start-up.
- Easy to use.
- Appealing design.

3.5.3.2. Design Planning

The main design goal was to keep the interface as simple as possible. A simple layout will eliminate any confusion, making the UI easy to use. The use of icons will make the user experience more inviting and intuitive by providing a visual representation of each feature.

A rough layout was created in Figure 3-31 to better visualise the conceptual design of the main window.

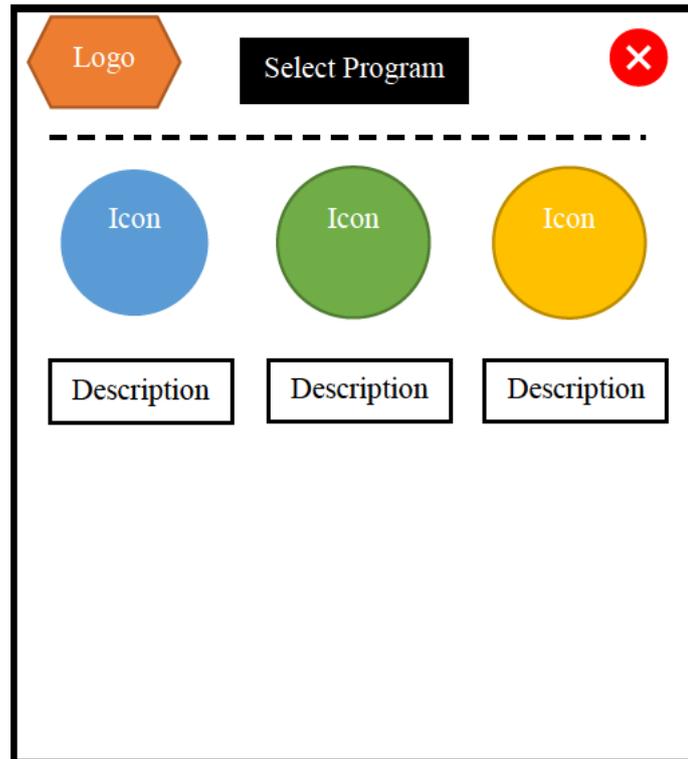


Figure 3-31: UI Design Concept

A concept for the login window was created in Figure 3-32. It follows the same design language as the main window, with the left icon and dividing line, making it recognisable.

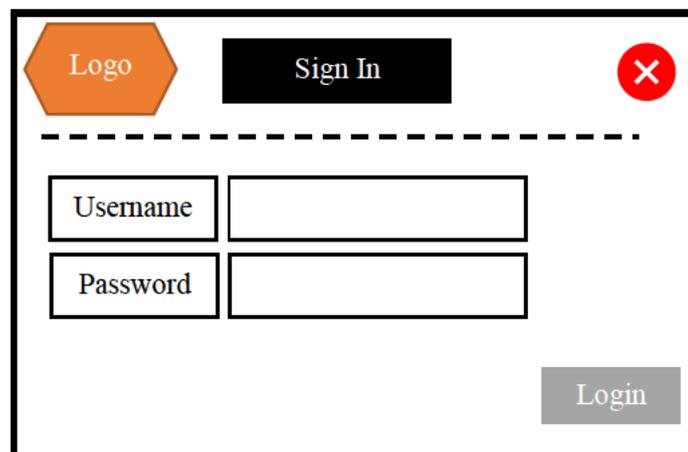


Figure 3-32: Login Window Concept

3.5.3.3. Digital Design

A grid layout was used to accurately align the various elements on the window. This was accomplished using Python and the guizero Python module (guizero, 2021). An application was created to fulfil the requirements, named TrackGUI.

After some use, the background was changed from white to black to minimise eye strain. This also increased the legibility of text and icons by minimising the range of contrast, as seen in Figure 3-33.

An icon was placed inside a large push button to make the selection process easy. A title and description were placed below each icon to indicate the purpose of the button.

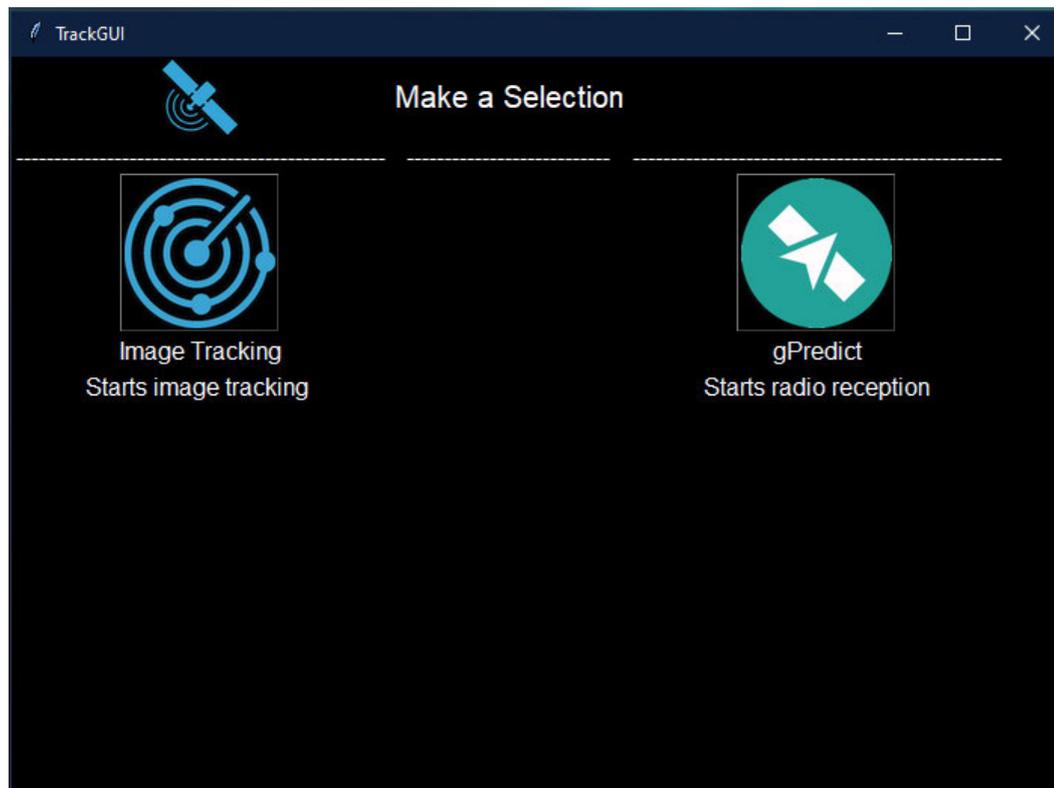


Figure 3-33: First UI iteration

Figure 3-34 shows the second iteration of the UI. Additional features were added, such as the GPS and SSTV options. Tweaks were made to incorporate colour into the header dividing line.

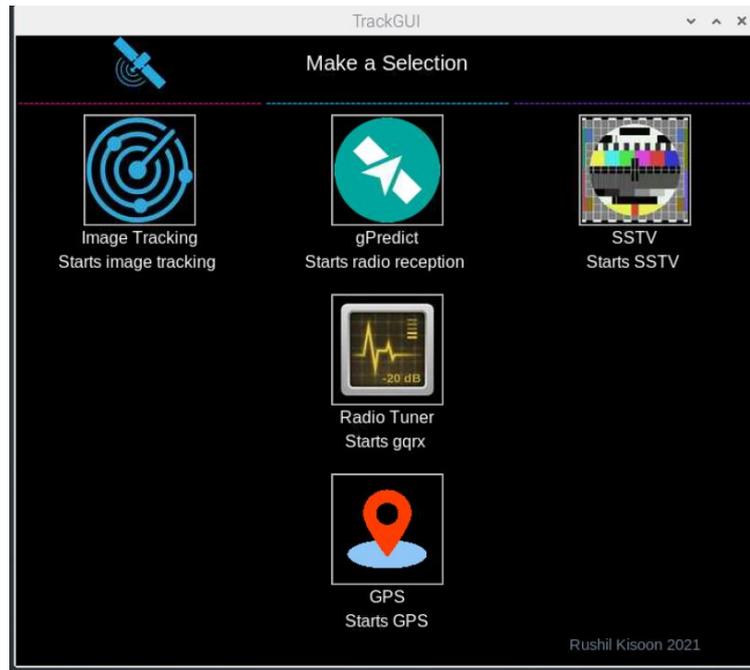


Figure 3-34: Second UI Iteration

The login window was created using the same aesthetic as the main window, as seen in Figure 3-35. The dividing line colours were replicated to tie both windows to the TrackGUI program.

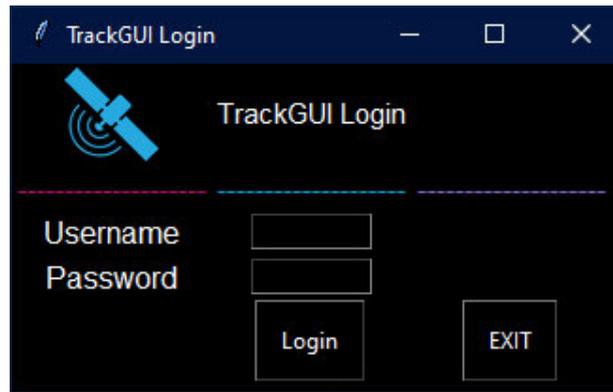


Figure 3-35: Final Login Window

Figure 3-36 shows a pop-up, which indicates the user access level once successfully logged on.

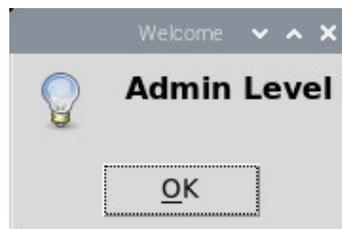


Figure 3-36: User Level Pop-up

Final changes and improvements to the UI included the creation of an executable desktop icon and exit button. A “User” file was created to store the username and passwords. It is only accessible through the admin level of the software. Figure 3-37 shows that the “Edit Users” button is viewable in admin mode, while regular users have the button removed.

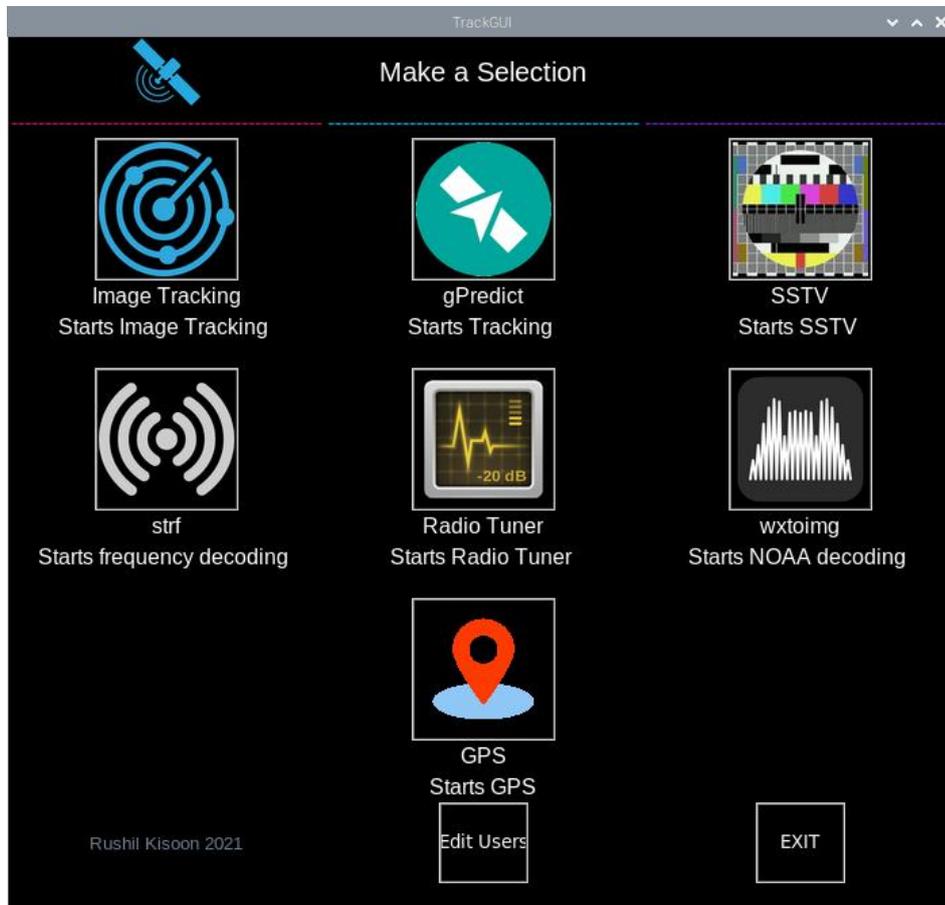


Figure 3-37: Admin-level Interface

Upon selecting “Edit Users”, a warning pop-up alerts the user as to the format required when adding additional persons to the system, illustrated in Figure 3-38.



Figure 3-38: Edit User Pop-up

3.6. Assembly and Setup

The processes involving system assembly and setup are detailed in the hierarchy of Figure 3-39.

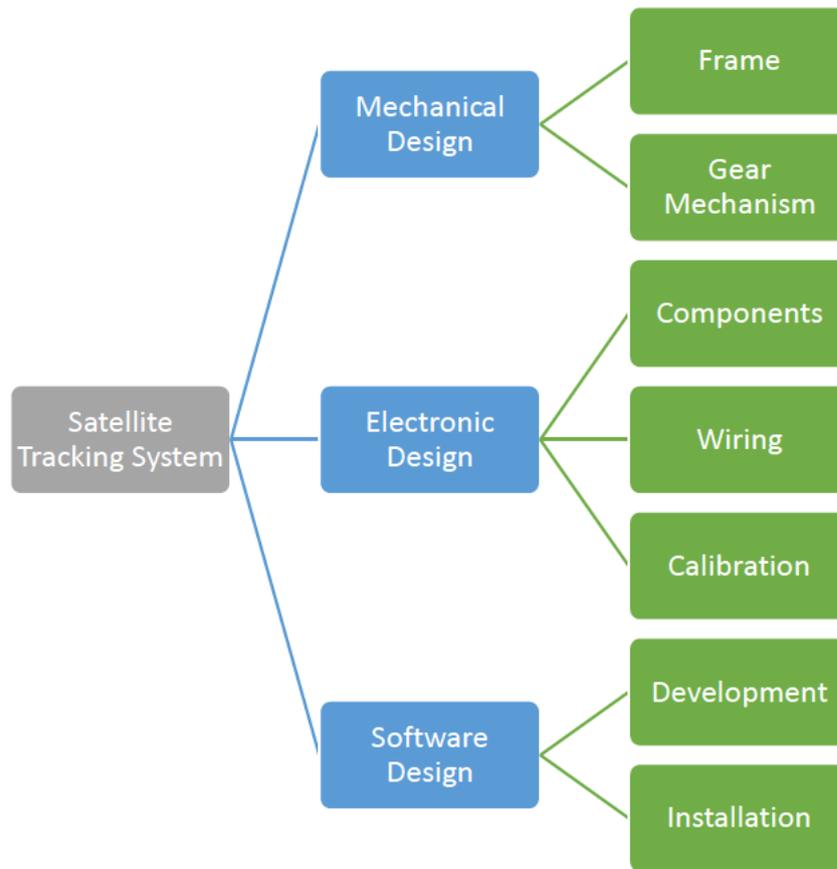


Figure 3-39: Hierarchy of System

3.6.1. Mechanical Setup

Since the system was previously constructed, little was required to complete the mechanical setup. Minor modifications were completed to improve the system.

- The bolts and other hardware were checked and tightened.
- A solid stainless-steel shaft with a wire slot was previously used for the azimuth axis. The new power wires were too large to fit in this narrow slot, so the shaft was replaced by a hollow aluminium tube of equal length and diameter. This allowed free movement of the wire and rotation of the slip ring.
- A steel plate was initially used as a surface to mount the electronics. This was replaced by an Acrylonitrile butadiene styrene (ABS) sheet to prevent short circuits.
- The antenna and telescope were secured to the new mounts before being attached to the elevation axis.

3.6.2. Electronic Setup

The wiring from the previous system was removed, allowing the components to be rewired from scratch. The electronic system was assembled using Figure 3-23 and Figure 3-24 as a guide.

3.6.2.1. Power Circuit

The steps followed to connect the power circuit in Figure 3-40 are listed below:

- Two strands of 2.5 mm² wire were used for the main power.
- A fuse was attached near the positive battery terminal.
- The power wires were run through the azimuth shaft.
- Two of the four slip ring wires were soldered to the positive (+) power cable, and the other two to the negative (–) power cable.
- Each positive wire was first connected to the fuse box to provide circuit protection.
- A positive and negative wire was run to each of the motor drivers.
- The motor wires were connected to the driver outputs.
- Another pair of power wires were connected to the voltage regulator.
- A male barrel jack was connected to the voltage regulator to power the Arduino.
- A USB Type-C cable was attached to the regulator to power the Raspberry Pi.
- The final fuse was used to power a cooling fan.

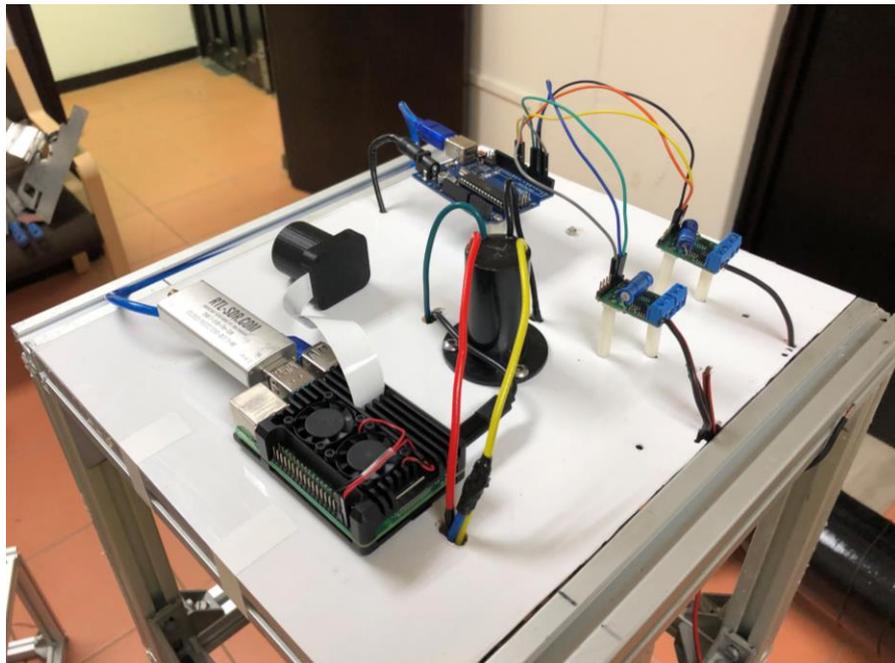


Figure 3-40: Power Connections

3.6.2.2. Signal Circuit

The steps followed to assemble the signal circuit in Figure 3-41 are listed below:

- Pins 5 and 6 were used for the azimuth motor driver.
- Pins 9 and 10 were used for the elevation motor driver.
- The IMU was connected to the 5V and GND pins.
 - Pins A4, Serial Data (SDA), and A5, Serial Clock (SCL), were used for serial communication.
- The Arduino was connected to the Raspberry Pi using the USB type-B cable.
- Pins 4 and 6 on the Raspberry Pi were used to power the case cooling fans.
- Pins 2 and 9 were used to power the GPS.
 - Pins 8 and 10 were used for serial communication with the GPS.
- The antenna cable was secured to the RTL-SDR dongle before it was connected to a USB port on the Raspberry Pi.

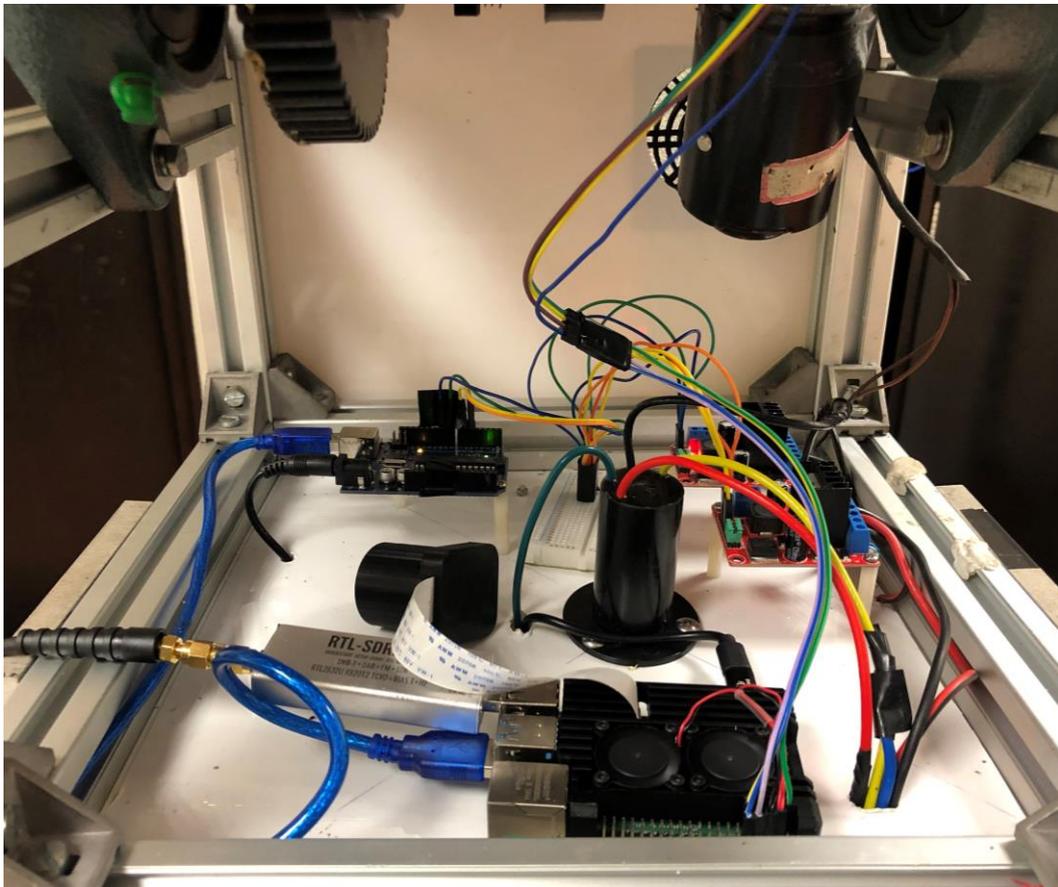


Figure 3-41: Signal Circuit

3.6.2.3. Antenna

The antenna parameters were checked using the specifications outlined in the 2020 report. RG58 coaxial cable was used for the antenna wire. This selection was due to its suitability for use with high-frequency signals (Belden, 2012). The antenna wire was spaced 172 mm apart, creating eight turns.

Once verified, the wire was secured using Pratley epoxy and fibre-reinforced tape, as shown in Figure 3-42.



Figure 3-42: Antenna Verification

3.6.3. Software Setup

The processes and methods used to install the various software utilities are detailed. Component setup and verification tests were also performed.

3.6.3.1. Raspberry Pi

The Raspberry Pi is supplied as a barebones device, without peripherals such as a keyboard, mouse, or display. It requires a microSD card to store files and the operating system. Whilst many operating systems run on the ARM processor, Raspberry Pi OS was chosen. The constant development and support for the Raspberry Pi ensure maximum compatibility with the device. Using the Raspberry Pi Imager, seen in Figure 3-43, the Linux-based Raspberry Pi OS was flashed onto the microSD card (Raspberry Pi Foundation, 2021).

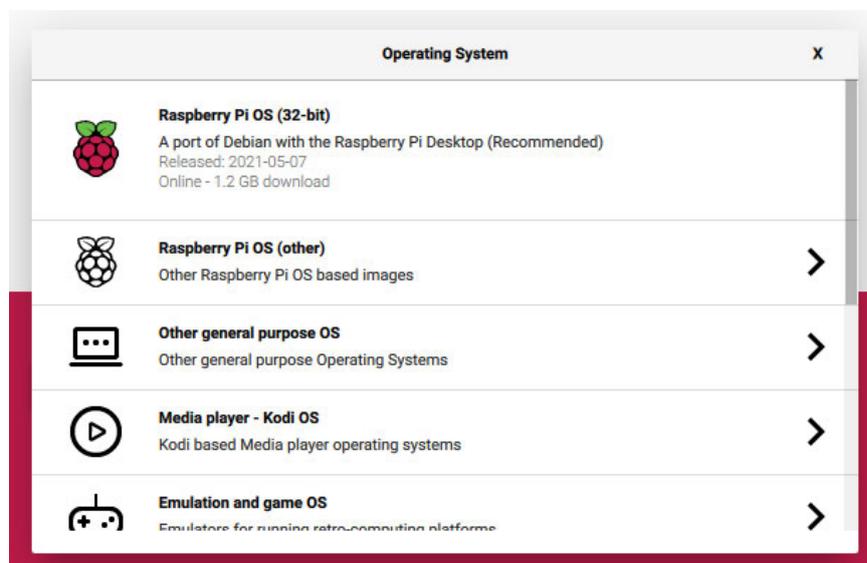


Figure 3-43: Raspberry Pi Imager

3.6.3.2. Remote Access

The Raspberry Pi has both Wi-Fi and ethernet capabilities, allowing it to be accessed from wherever there is an internet connection. Traditionally, remote access to a computer would involve setting up a firewall and port forwarding through the router, but with RealVNC, the task is much easier. Since the framework for RealVNC is built into Raspberry Pi OS, it allows greater functionality than the traditional command-line access.

Using a Virtual Network Computing (VNC) service allows complete control of a system as if you were interacting with it directly. Fortunately, due to an agreement between the Raspberry Pi Foundation and RealVNC, this service is free when used non-commercially (RealVNC, 2021).

Using the RealVNC Viewer app allows the Raspberry Pi to be accessed remotely from anywhere. The desktop can be quickly accessed via a mobile device or computer, as seen in Figure 3-44.

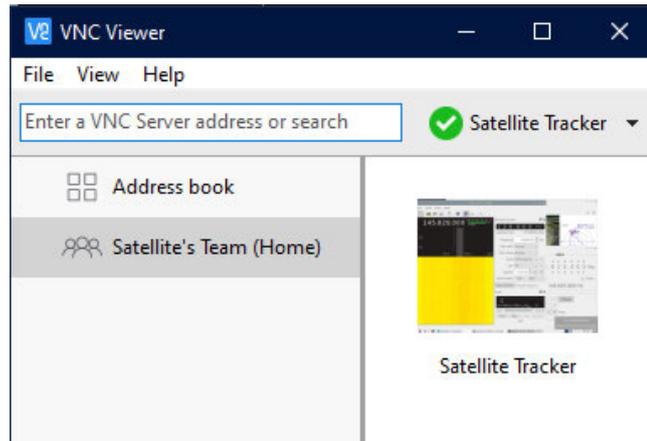


Figure 3-44: RealVNC Selection Window

3.6.3.3. Gpredict

Using the “Add / Remove Software” option in the main Raspberry Pi menu allowed Gpredict to be installed. The location of the base station was set up using the coordinates from the GPS. Satellites were added to the map to be tracked, as seen in Figure 3-45.

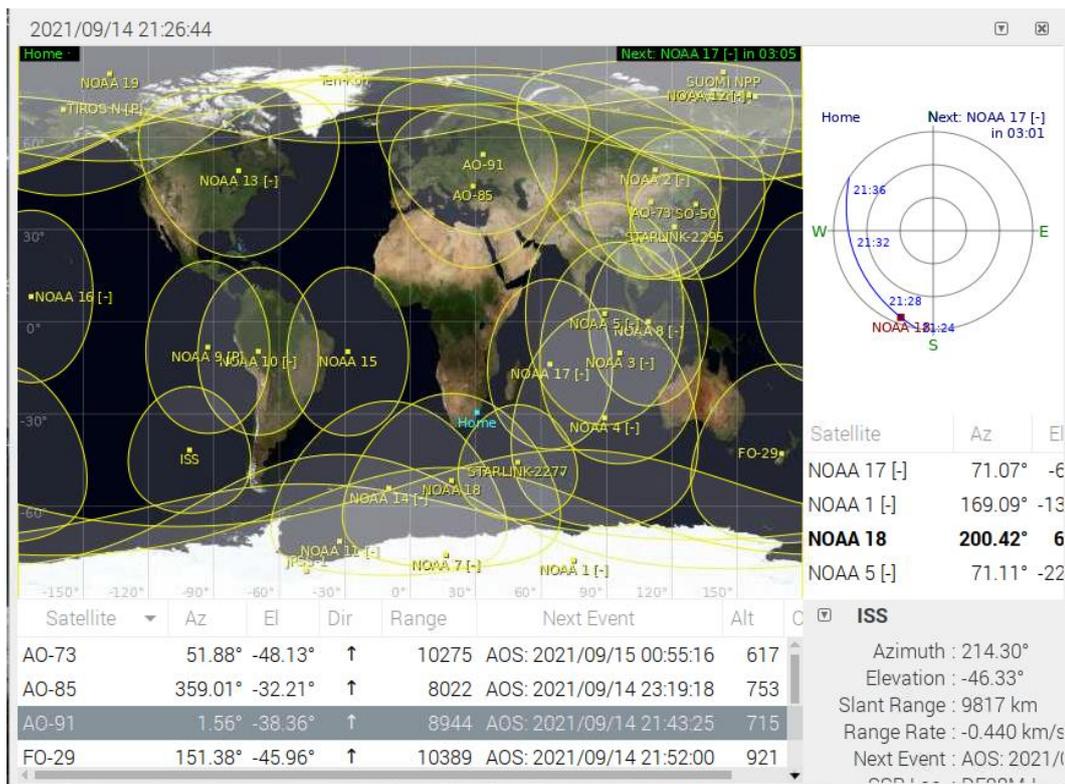


Figure 3-45: Gpredict Monitoring Window

3.6.3.4. GQRX

GQRX was installed using the “Add / Remove Software” feature. The red waterfall line in Figure 3-46 indicated that the tuner and antenna were functional when tested on a local radio station. The appropriate driver for the RTL-SDR dongle was required and installed using:

```
sudo apt-get install rtl-sdr
```

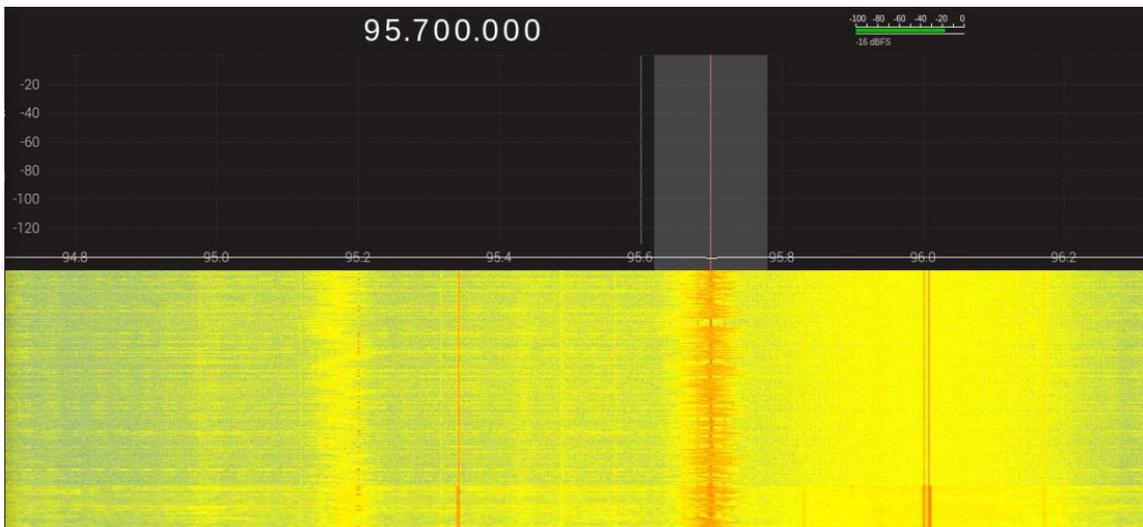


Figure 3-46: Initial GQRX Test

3.6.3.5. Rotator

The operation of the rotator software was verified. The serial communication between the Raspberry Pi and Arduino was operational. Figure 3-47 shows the series of commands between the rotcld, and hamlib protocols. The setup was recognised as an EasycommII rotator controller.

```
pi@raspberrypi:~$ rotcld -m 202 -r /dev/ttyACM0 -s 19200 -C timeout=500 -vvvv
rotcld, Hamlib 3.3
Report bugs to <hamlib-developer@lists.sourceforge.net>

rot_init called
initrots3_easycomm called
rot_register (201)
rot_register (202)
rot_register (204)
rot_token_lookup called
rot_confparam_lookup called
rot_set_conf called
rot_confparam_lookup called
rot_set_conf: timeout='500'
frontrot_set_conf called
rot_open called
serial_open called
serial_setup called
Opened rot model 202, 'EasycommII'
rig_strstatus called
Backend version: 0.4, Status: Beta
```

Figure 3-47: Serial Link

3.6.3.6. Network Connectivity

Raspberry Pi OS does not have native support for the WPA2-Enterprise network protocol used by the university. The GlobalProtect virtual private network (VPN) was also unsupported, thus preventing network access.

a. Wi-Fi

The file storing the wireless network configuration had to be manually edited to include the details of the network. The configuration file was accessed using:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

The following lines were added, pertaining to the UKZN network:

```
Network = {  
  
    ssid="UKZNDATA"  
  
    proto=RSN  
  
    key_mgmt=WPA-EAP  
  
    pairwise=CCMP  
  
    auth_alg=OPEN  
  
    eap=PEAP  
  
    identity="*****"  
  
    password="*****"  
  
    phase1="Peaplabel=0"  
  
    phase2="auth=MSCHAPV2"  
  
    priority=1  
  
}
```

b. GlobalProtect

Implementing the GlobalProtect VPN at UKZN created many network issues. Since the Raspberry Pi and other IoT devices make up a small market share compared to Android, iOS, Mac and Windows systems, there is limited VPN support from the developer of GlobalProtect, Palo Alto Networks.

The utility operates through the command line, which makes the setup challenging. With the assistance of ICS Howard at UKZN, the initial installation of GlobalProtect was completed.

The main hurdle was that the system would be inaccessible if the VPN did not connect upon start-up. With the Raspberry Pi being remotely operated in headless mode, any form of user input relies on an internet connection to relay the commands.

The following command is required to connect to the VPN:

```
globalprotect connect -u USERNAME
```

A short script, GP.py, was created to run at start up, allowing the Raspberry Pi to connect to the GlobalProtect service automatically.

```
#!/usr/bin/env python

import os

os.system("globalprotect connect -u USERNAME")
```

3.6.3.7. IMU

Figure 3-48 shows the IMU calibration. Holding the Arduino and not the sensor reduces the chance of interference and errors. The calibration procedure requires that the face of the IMU be pointed in all orientations and compass directions. The LEDs on the Arduino indicate the calibration status. A flashing light shows that the specific orientation is being calibrated. When the sensor is calibrated, the LEDs turn off.

Once the sensor was tested and placed on the system, recalibration ensured that it was adjusted for that position.

If any magnetic objects placed are within 100 mm of the sensor, it would permanently damage the orientation of the IMU sensor chips.

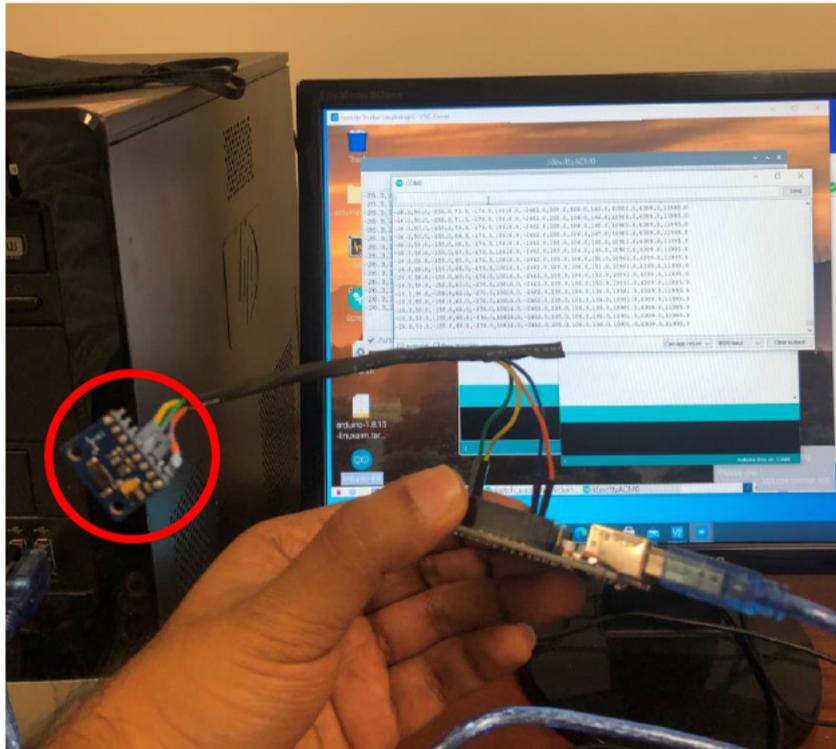


Figure 3-48: IMU Calibration

3.6.3.8. GPS

The GPS sensor was connected and placed on the top of the frame. After 15 minutes, the red “FIX” LED illuminated, indicating that a GPS signal was found, as seen in Figure 3-49.

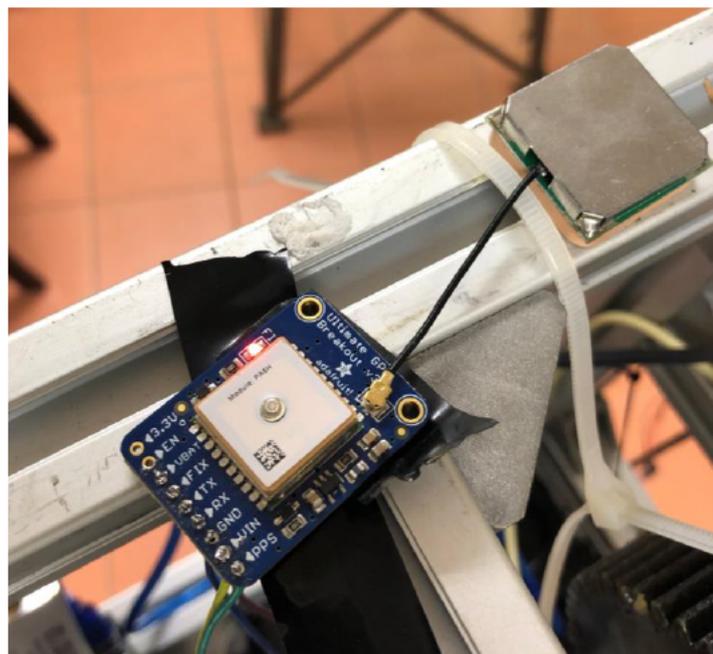


Figure 3-49: GPS Setup

The serial output from the GPS can be seen in Figure 3-50. As mentioned in Figure 3-30, the expected NMEA sentences are visible in the screenshot.

```

pi@raspberrypi:~$ sudo cat /dev/serial0
!$GPGGA,165734.800,,,,,0,00,,M,M,,*72

$GPGSA,A,1,,,,,,,,,,,,,*1E

$GPRMC,165734.800,V,,,,,0.00,0.00,120180,,,N*4D

$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32

$GPGSA,A,1,,,,,,,,,,,,,*1E
^X^O^_L yy$GPGGA,202505.799,,,,,0,00,,M,^Z^A^X^O^_L !$GPGGA,165734.800,,,,,
0,00,,M,M,,*72
$GPRMC,165734.800,V,,,,,0.00,0.00,120180,,,N*4D
202505.799,,,,,0,00,,M,^Z^A^X^O^_L !$GPGGA,165734.800,,,,,0,00,,M,M,,*72
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32
0180,,,N*4D
202505.799,,,,,0,00,,M,^Z^A^X^O^_L !$GPGGA,165734.800,,,,,0,00,,M,M,,*72
$GPGSA,A,1,,,,,,,,,,,,,*1E
K,N*32
0180,,,N*4D
202505.799,,,,,0,00,,M,^Z^A^X^O^_L !$GPGGA,165734.800,,,,,0,00,,M,M,,*72
$GPRMC,165734.800,V,,,,,0.00,0.00,120180,,,N*4D
PGGA,165734.800,,,,,0,00,,M,M,,*72
!$GPGGA,165734.800,,,,,0,00,,M,M,,*72
$GPGGA,165734.800,,,,,0,00,,M,M,,*72
0,,,N*4D
PGGA,165734.800,,,,,0,00,,M,M,,*72
!$GPGGA,165734.800,,,,,0,00,,M,M,,*72
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32
72

```

Figure 3-50: GPS Serial Output

3.6.4. Iteration and Improvements

Certain aspects need to be addressed and improved to solve issues that arose.

- Further research revealed that the wiper motors could be operated in two modes, depending on which pins are powered. Figure 3-51 shows the position of the low-speed, high-speed and ground pins (Terry, 2019). Using the low-speed pin allows for an average RPM of 40, with a torque of 23 Nm, while the high-speed pin increases the speed to 70 RPM but decreases the torque to 18 Nm (Terry, 2019).

Preliminary testing showed that the high-speed mode was too fast to control accurately. Once the initial stall was overcome, the rapid movement meant that the required position was quickly overshoot. Subsequently, the low-speed setting was selected on both motors.

The Elevation Torque was calculated in Appendix A – Calculations. The results showed that the system was able to generate 30 Nm of torque after passing through the gears. Only 17.92 Nm was needed to move the elevation axis, which resulted in 1.67 times more torque being produced than required.

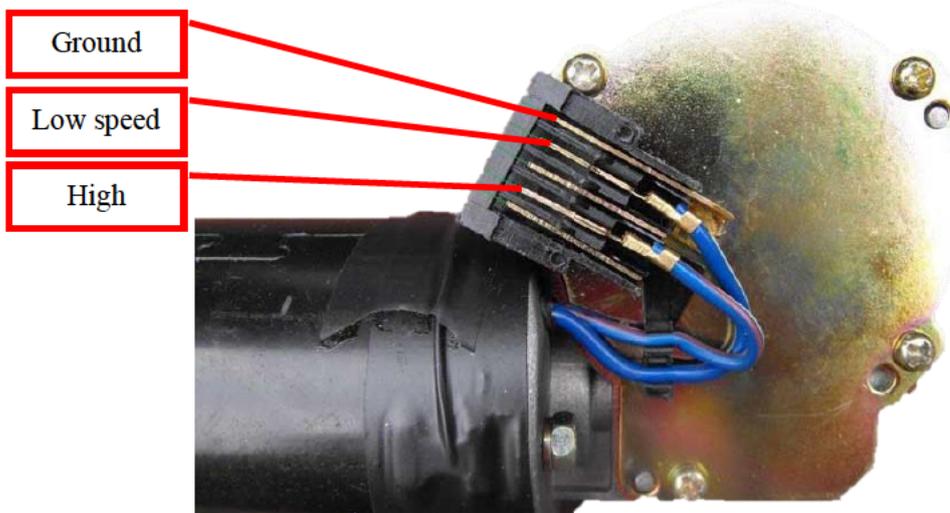


Figure 3-51: Wiper Motor Pinout

- The wiring was neatened, as seen in Figure 3-52. All components were secured and adhered to the base plate. Ferrite rings (yellow) were added to the GPS and IMU serial lines. They help to reduce electromagnetic interference (EMI),

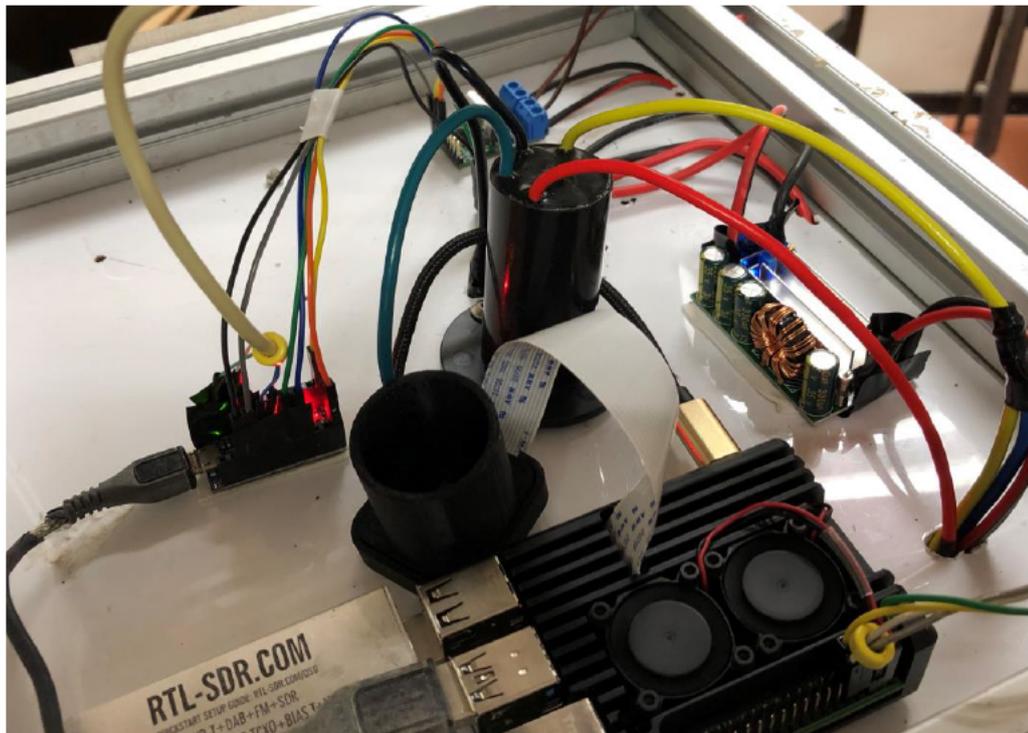


Figure 3-52: Neatened Wiring with Ferrite Rings Added

- The current draw and speed of the azimuth motor would spike once per revolution. Inspection of the motion showed that the motor shaft was not spinning true. Disassembly of the motor showed wear and tear that resulted in the output shaft fusing to the bushing of the motor housing, as seen in Figure 3-53.

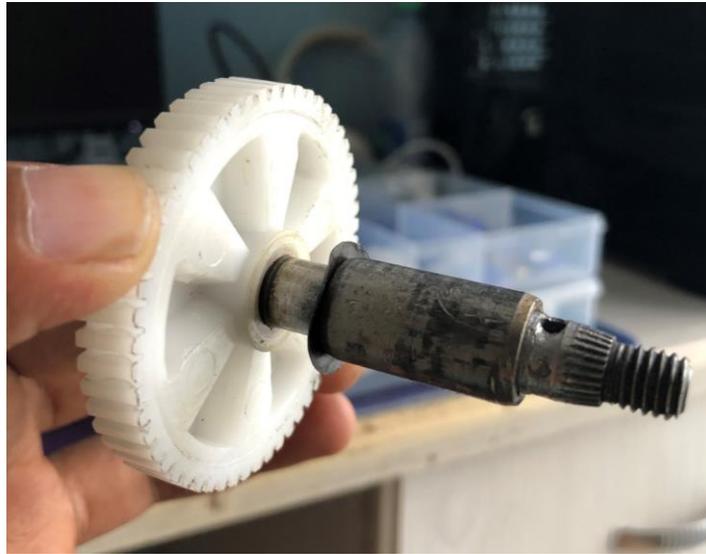


Figure 3-53: Azimuth Motor Shaft

- Some of the frame extrusions were not correctly aligned, leading to stress being placed on the mechanism and fixtures, as seen in Figure 3-54. Subsequently, the system was disassembled, realigned, and secured, as shown in Figure 3-55.



Figure 3-54: Frame Misalignment

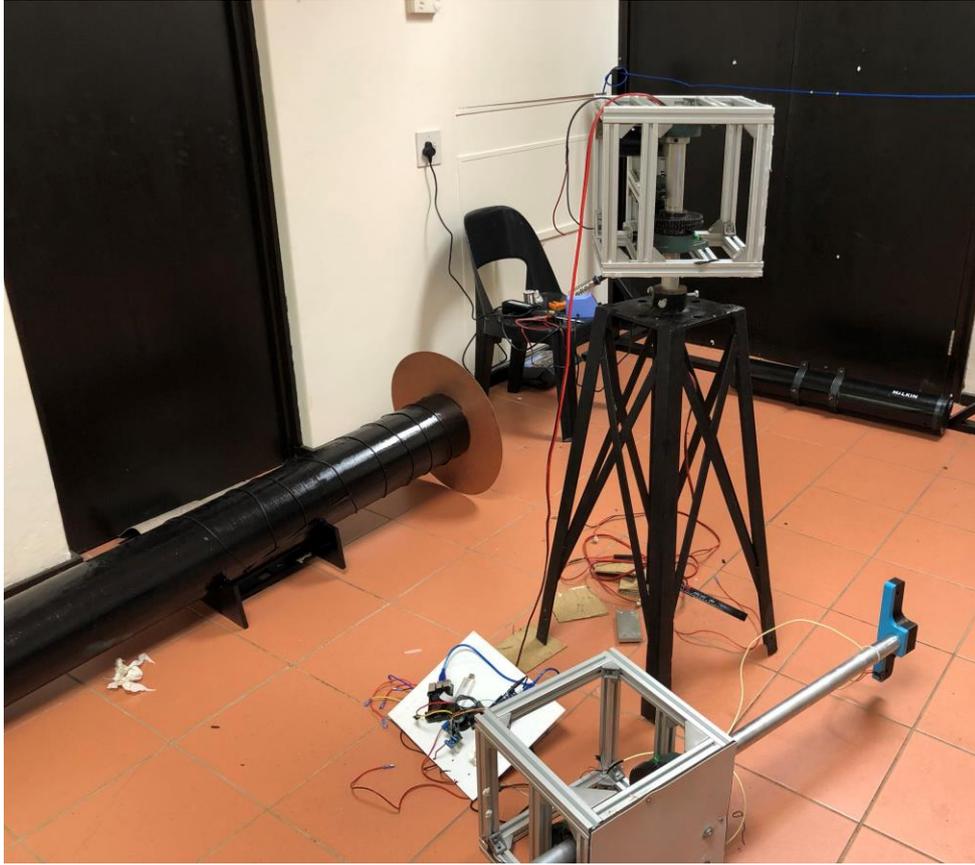


Figure 3-55: System Rebuild

- The IMU bracket was initially placed within the frame. Preliminary testing showed that the magnetic field from the elevation motor caused interference at certain orientations. The IMU bracket and sensor were moved outside the frame to prevent false readings and interference.

3.7. Verification

3.7.1. Software Validation

The TrackGUI program was run to determine CPU usage and temperature to evaluate if further optimisation was required.

3.7.1.1. Experimental procedure

The steps below were followed to perform the verification:

1. The Raspberry Pi was booted up.
2. The system was given 5 minutes to reach an idle state whilst the start-up tasks and processes were executed.
3. The air-conditioning in the lab was turned on to 20°C to provide a constant temperature.
4. The initial temperature and usage of the processor were recorded using the built-in CPU monitor, seen in Figure 3-56.
5. The TrackGUI software was launched.
6. The CPU temperature and usage were recorded for a period of 10 seconds.
7. The software was closed, and the system was allowed to 'rest' for 1 minute.
8. Steps 3 to 6 were repeated to obtain more data points.
9. The test was repeated to evaluate the performance upon successful login and launching of the main user interface.

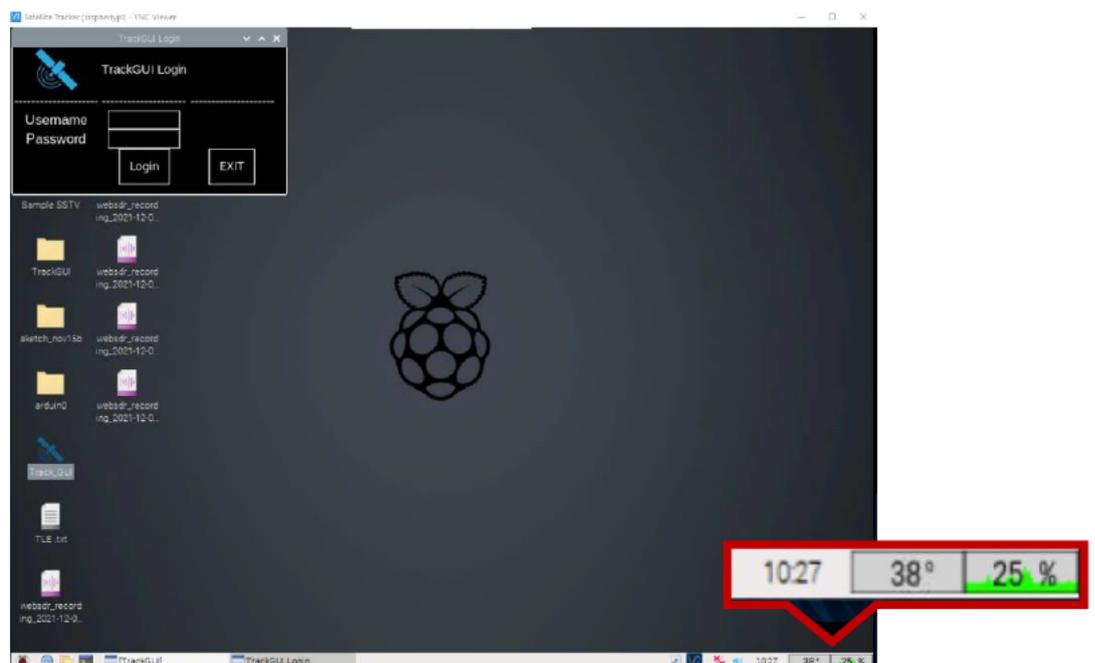


Figure 3-56: Screenshot Showing the Login Window and CPU Monitor

a. Login Window Results

The CPU usage data for the login menu was recorded in Table 3-6 and plotted in Figure 3-57. The trends showed a spike in CPU usage to approximately 60% after the program was launched. After approximately 6 seconds, the usage returned to an idle value of around 25%.

Table 3-6: Login Menu CPU Usage

Time (s)	CPU usage (%)		
	Test 1	Test 2	Test 3
0	25	25	25
1	26	27	26
2	46	48	45
3	61	59	60
4	51	51	51
5	45	40	47
6	27	25	25
7	25	26	26
8	26	25	25
9	25	25	25
10	25	26	25

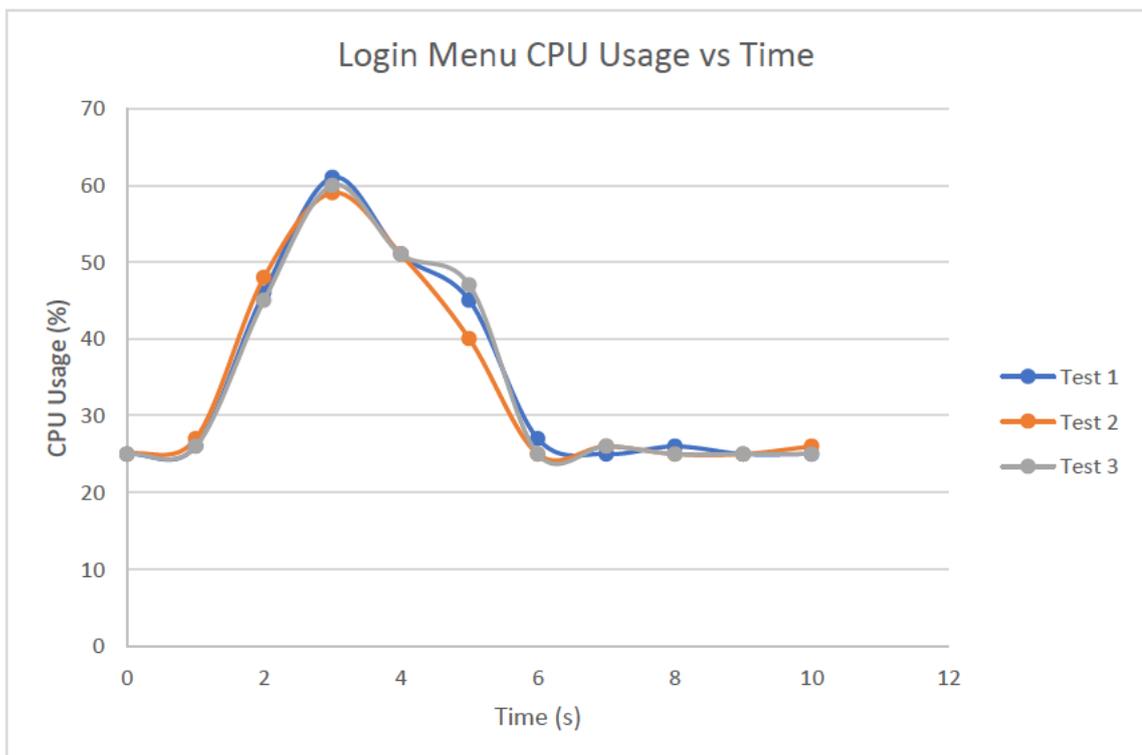


Figure 3-57: Graph of Login Menu CPU Usage vs Time

The login menu CPU temperature data was recorded in Table 3-7 and plotted in Figure 3-58. After the launch of the program, the CPU temperature rose to a maximum of 42 °C before returning to an idle of 40 °C. This showed that the spike in CPU usage upon program launch was not sustained for long enough to overheat the CPU.

Table 3-7: Login Menu CPU Temperature

Time (s)	CPU Temperature (°C)		
	Test 1	Test 2	Test 3
0	39	39	39
1	41	39	40
2	41	41	41
3	42	41	42
4	41	42	42
5	39	41	39
6	40	40	40
7	40	40	39
8	40	39	39
9	40	39	39
10	40	40	40

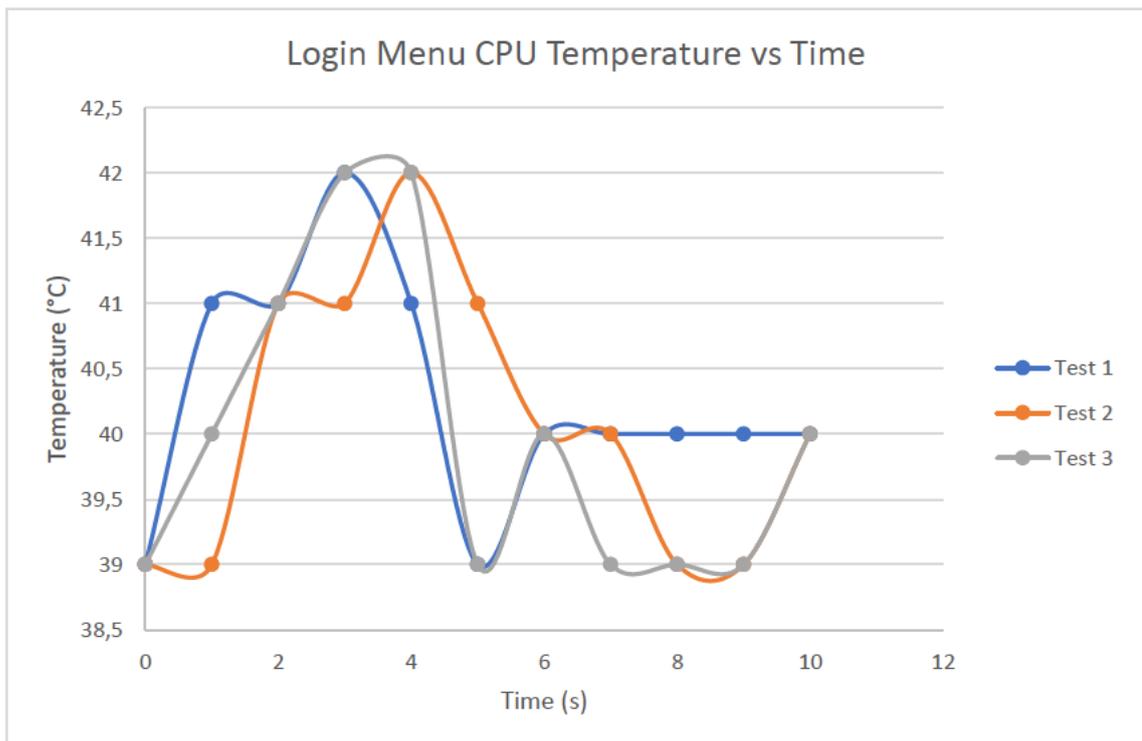


Figure 3-58: Graph of Login Menu CPU Temperature vs Time

b. Main Menu Results

The main menu CPU usage data was recorded in Table 3-8 and plotted in Figure 3-59. Upon successful login, the CPU usage only increased by 2% to generate the main menu window and load the required assets. The usage normalised after 2 seconds.

Table 3-8: Main Menu CPU Usage

Time (s)	CPU usage (%)		
	Test 1	Test 2	Test 3
0	25	25	25
1	25	26	26
2	26	26	27
3	26	25	27
4	27	25	26
5	25	25	25
6	25	24	25
7	25	24	26
8	24	25	25
9	25	25	25
10	25	24	25

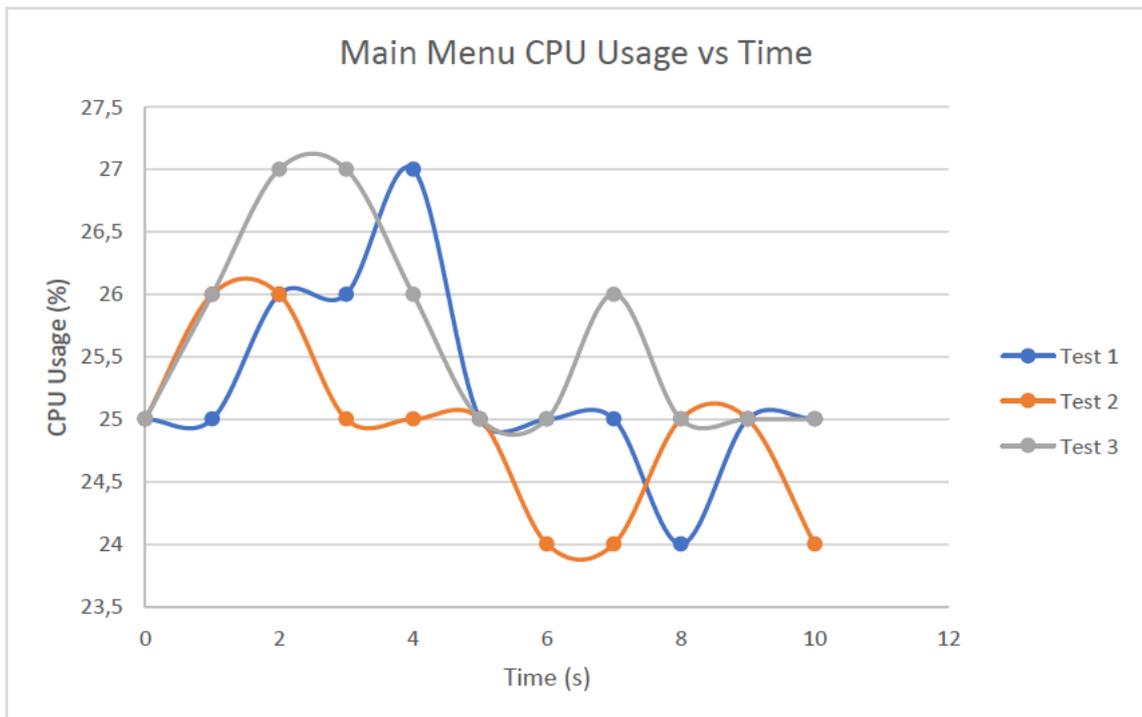


Figure 3-59: Graph of Main Menu CPU Usage vs Time

The main menu CPU temperature was recorded in Table 3-9 and plotted in Figure 3-60. The CPU temperature increased by 2 °C upon launch of the main menu and normalised after 10 seconds.

Table 3-9: Main Menu CPU Temperature

Time (s)	CPU Temperature (°C)		
	Test 1	Test 2	Test 3
0	39	39	39
1	41	40	39
2	41	41	40
3	39	38	39
4	40	39	39
5	39	39	39
6	40	39	39
7	40	39	39
8	40	39	38
9	40	38	39
10	39	39	39

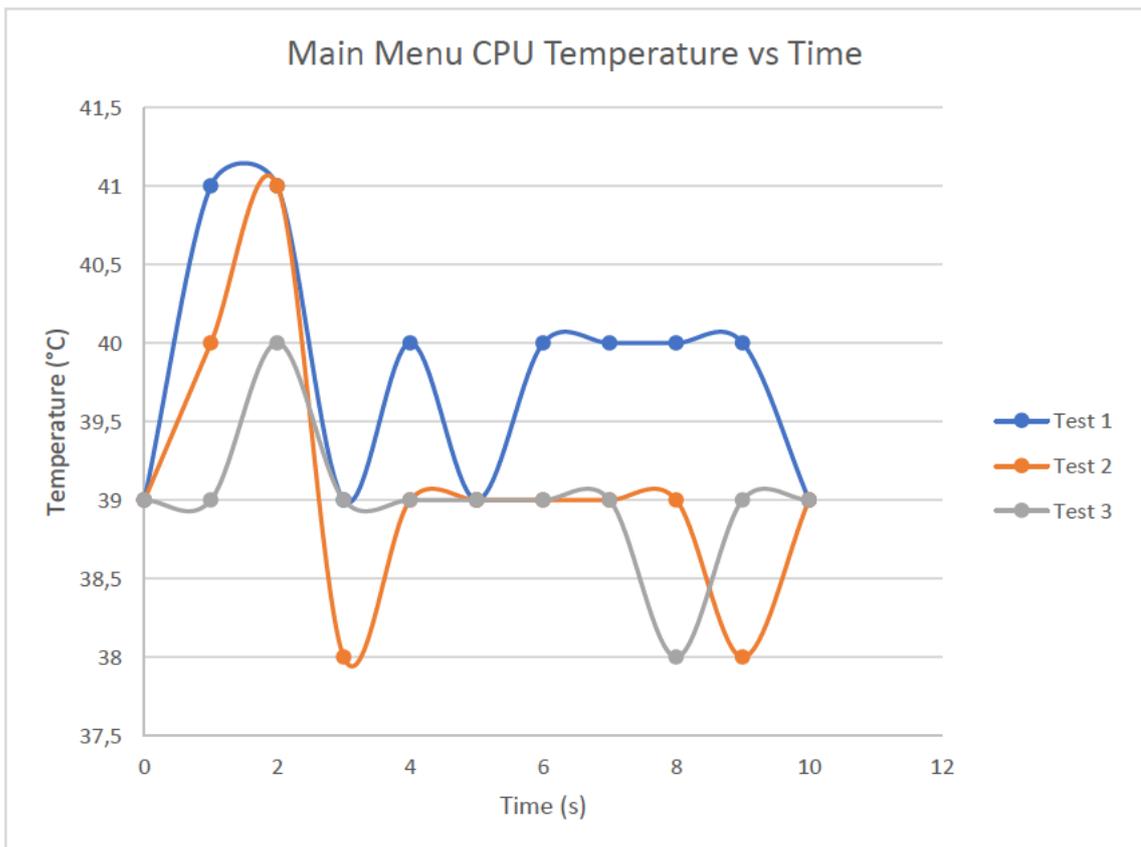


Figure 3-60: Graph of Main Menu CPU Temperature vs Time

Verification of the software measured the impact of the created UI on both the CPU usage and CPU temperature. Analysing the graph in Figure 3-57 showed that initially launching the program increased CPU usage by as much as 36%. The required assets and libraries are loaded in at launch, so the CPU usage rises by only 2%, Figure 3-59 when the main window is rendered after login.

Throughout this verification, the CPU temperature stayed below the 80 °C limit, which is the value at which the system would throttle the CPU usage to reduce temperatures (Anderson, 2019).

Figure 3-58 and Figure 3-60 confirmed that along with the Raspberry Pi cooling case and heatsink, the code was effective and efficient to not cause any hangs to harm the processor.

3.7.2. GPS Verification

The precision and accuracy of the GPS module were verified.

3.7.2.1. Experimental Procedure

The steps below were followed to perform the verification:

1. The GPS was accessed using the GPS button on the TrackGUI software.
2. A period of 10 minutes was given to allow the GPS module to obtain a fix.
3. The longitude and latitude of the GPS module were obtained in Figure 3-61.
4. The phones were placed below the GPS module to get an accurate reading.
5. Using the Commander Compass Go app on both phones, the GPS location of each device was determined, shown in Figure 3-62.
6. The latitude and longitude of the phone and the GPS module were recorded.

```

raspberrypi:/dev/serial0 9600 NMEA0183> 0A0A0A0A[A[A[A[A[A[A[A[A[A[A[A[A[B
Time: 2021-10-22T15:08:57.000Z Lat: 29 52' 14.00273" Son: 30 58' 34.08471" E
Cooked TPV
GPGGA GPGSA GPRMC GPVTG GPGSV
Sentences
Ch PRN Az El S/N Time: 150857.000 Time: 150858.000
0 2 101 67 19 Latitude: 2952.1573 S Latitude: 2952.1578
1 20 165 57 28 Longitude: 03058.2683 E Longitude: 03058.2682
2 5 231 55 31 Speed: 0.41 Altitude: 69.1
3 13 358 54 0 Course: 355.01 Quality: 1 Sats: 04
4 29 229 32 32 Status: A FAA: A HDOP: 4.26
5 15 326 32 0 MagVar: RMC GGA
6 6 71 29 0
7 30 83 19 0
8 7 115 11 0 Mode: A3 Sats: 2 20 5 29 UTC: RMS:
9 12 307 6 0 DOP: H=3.72 V=0.99 P=3.85 MAJ: MIN:
10 19 22 4 0 TOFF: -0.076960631 ORI: LAT:
11 25 279 3 0 PPS: LON: ALT:
GSV GSA + PPS GST

```

Figure 3-61: GPS Output



Figure 3-62: GPS Phone Location

3.7.2.2. GPS Results

Values were recorded in Table 3-10 at both the front and back of the lab. The results were plotted in Figure 3-63. The scatter plot shows that the data gathered from the GPS module and phone GPS did correlate; however, the GPS module has a lower output resolution.

Table 3-10: GPS Verification Results

GPS Module		Phone GPS		Deviance	
Front of Lab					
Latitude	Longitude	Latitude	Longitude	Latitude	Longitude
-29.8704	30.9760	-29.870431	30.976076	0.000031	0.000076
-29.8704	30.9761	-29.870402	30.976089	0.000002	0.000011
-29.8704	30.9760	-29.870388	30.976049	0.000012	0.000049
Back of Lab					
-29.8706	30.9762	-29.870608	30.976256	0.000008	0.000056
-29.8705	30.9762	-29.870586	30.976281	0.000086	0.000081
-29.8706	30.9762	-29.870623	30.976237	0.000023	0.000037

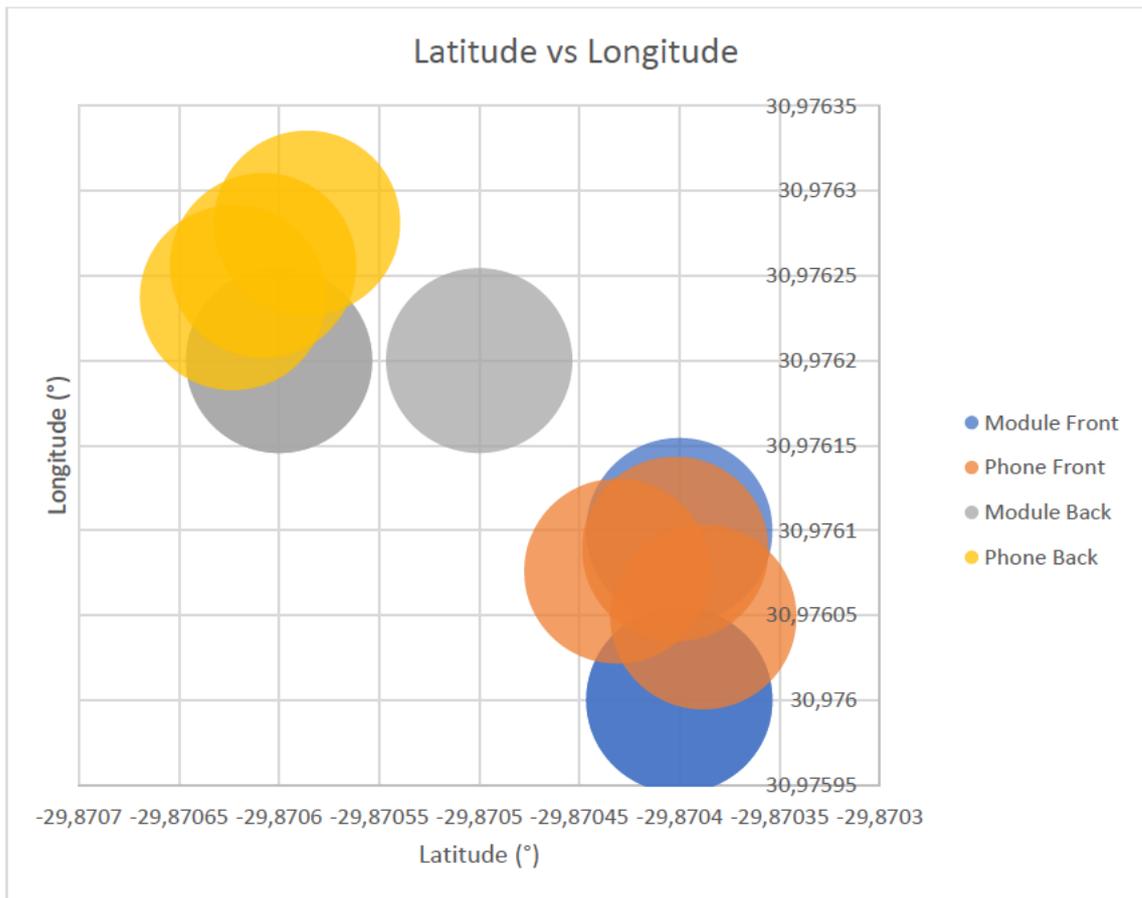


Figure 3-63: Scatter Plot of GPS Coordinates

The GPS operation was validated. Recordings were taken in two locations, at the front and at the back of the lab. The outdoor testing range on the Raspberry Pi was limited to the front and back of the lab due to the coverage provided by the Wi-Fi access point. Table 3-10 shows that the values recorded by the GPS module had a resolution of 4 decimal points, whilst the phone sensor had six decimal points. GPS values with four decimals have an accuracy of 3-5 m, whilst six decimals indicate an exact position (Schmidt, 2021). The deviance values were calculated. Since the difference in results was due to the lower resolution of the GPS module, it could be classified as a type of instrumental error.

A notable difference between a phone GPS and a GPS module is that a phone uses an assisted global positioning system (A-GPS). A combination of satellite signals and triangulation data from cell towers helps to determine its position more accurately. In Figure 3-63, the variance in phone GPS values is due to its greater accuracy when compared to that of the GPS module. The position of the GPS module compared to the phone is acceptable, given the hardware differences in the technology.

3.8. Chapter Summary

Aspects of the mechanical design were previously completed. Minor modifications were required to refine certain parts and add functionality. The frame is heavy-duty aluminium, thus providing a rigid structure and resistance to wear. The system was able to rotate 360° in the azimuth direction and 90° in the elevation direction. Cost and availability affected the selection of parts. Items that were previously purchased were reused wherever possible.

Electronic components were chosen based on their applicability to perform the task at hand. Many components were chosen as they were low-powered, which lent well to the system operating on a 12 V battery. Wiper motors were also selected as they operate at 12 V instead of the 48 V required for equivalent stepper motors.

Connection diagrams were drawn up to allow for the layout of both the power and signal circuits to be better visualised. It was discovered that communication between the Raspberry Pi and Arduino could be established via USB, eliminating a logic level converter.

Python and C were chosen to form the basis of the Raspberry Pi and Arduino software, respectively. An easy-to-use GUI was created, with a user login interface and user hierarchy. Various iterations added and altered aspects of the appearance and functionality to produce the final TrackGUI program. Gpredict and GQRX were chosen to aid in tracking and radio reception of signals from satellites. Getting the Arduino to interface with the Raspberry Pi, a rotator device, relied on the Hamlib, Rotctl and EasyComm II protocols. The SARCNET rotator code was modified and uploaded. Frequency identification was implemented using the STRF toolkit, with SSTV decoding added as an additional feature.

Assembly and setup procedures were listed for each aspect of the system. The power and signal lines were wired up and secured using the designed circuit diagrams. Initial testing and calibration of the sensors were performed. The antenna was verified to ensure that it met the previously designed parameters. Verification tests were performed to evaluate the system.

As the preliminary tests were conducted, certain aspects needed to be addressed and improved. Various iterations were made to the system wherever required.

Once the system was finalised, experimental testing was performed. Chapter 4 details the procedure and results of each test that was carried out.

4. Testing and Results

4.1. Introduction

The following section analyses the operating parameters of the system. This will enable further optimisation to produce a better operating machine. Quantifiable results are needed to verify the system, components and processes used. These investigations aim to assess the suitability and performance of various aspects of the system.

4.1.1. Apparatus

The following equipment was required to perform the various tests:

- Satellite tracking system
- Raspberry Pi
- RTL-SDR
- Arduino
- GPS sensor
- IMU sensor
- iPhone 6S
- iPhone 8 Plus
- iOS Apps: Commander Compass Go (Ahafonau, 2021)
- : Phyphox (University, RWTH Aachen, 2021)
- : Satellite Pointer (TouchDown Apps, 2021)

4.2. Orientation and Feedback Test

The movement of the azimuth and elevation axis were tested to determine their accuracy. The system was moved to various positions and recordings were taken to verify the coordinates.

4.2.1. Experimental Procedure

1. The IMU was recalibrated as per the method explained in Section 3.6.3.7.
2. The sensors in both phones were calibrated by enabling the *Compass Calibration* and *Motion Calibration & Distance* setting as mentioned by Lifewire (Chandler, 2021).
3. The rotator axis that was being tested was set to 0°.
4. One phone was secured to the antenna bracket and oriented to EL 0° (The phone sensor is orientated such that the elevation value is 0° when held upwards. When placed flat down, it displays a value of -90°, as shown in Figure 4-1).
5. A command was sent to the Arduino to move the axis to various degrees.

6. The actual axis position could be determined and recorded using the IMU output via the serial monitor, with the phone sensor as feedback.
7. The test was repeated, for the AZ axis, with the phone secured to the frame to determine the azimuth values, shown in Figure 4-2



Figure 4-1: Elevation Test



Figure 4-2: Azimuth Test

4.2.2. Orientation and Feedback Results

The motion behaviour of each axis was recorded and analysed from various positions. The results from the test are presented below.

4.2.2.1. Elevation 15°, 30° and 45° Tests

Movement values of 15°, 30° and 45° were recorded in Appendix C – Results, Table 6-1, Table 6-2, and Table 6-3, respectively. The results are presented in Figure 4-3 as a box and whisker diagram. When the axis was commanded to move from 0°, it often overshoot the required value. When moving from a previous angle, it tended to undershoot the position.

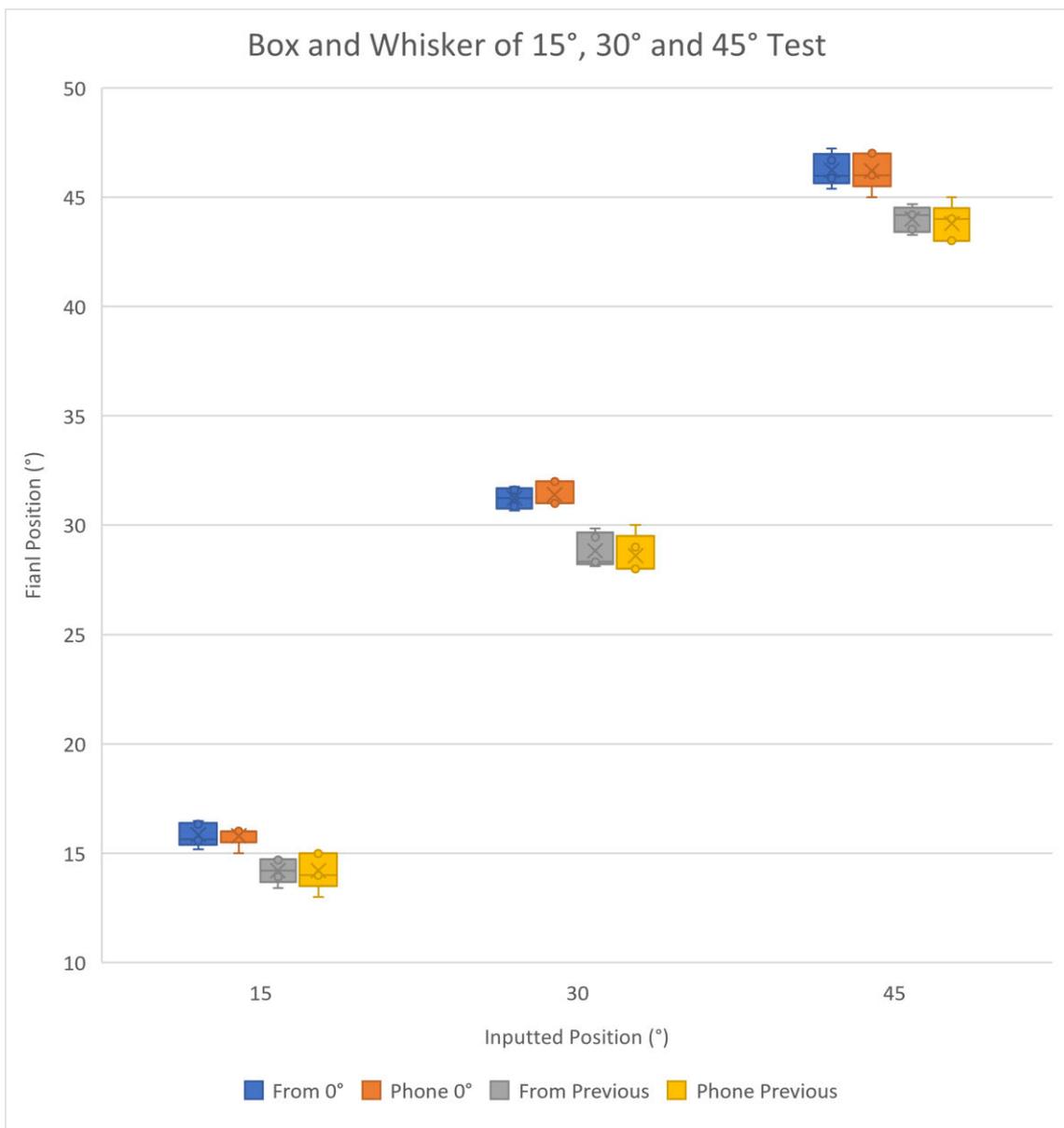


Figure 4-3: Box and Whisker of 15°, 30° and 45° Test

4.2.2.2. Elevation 60°, 75° and 90° Tests

Movement values of 60°, 75° and 90° were recorded in Appendix C – Results, Table 6-4,

Table 6-5, and Table 6-6, respectively. The results were represented in Figure 4-4 as a box and whisker diagram. A similar trend was observed whereby the elevation axis would overshoot the position when moving from 0° and undershoot when moving from a previous angle.

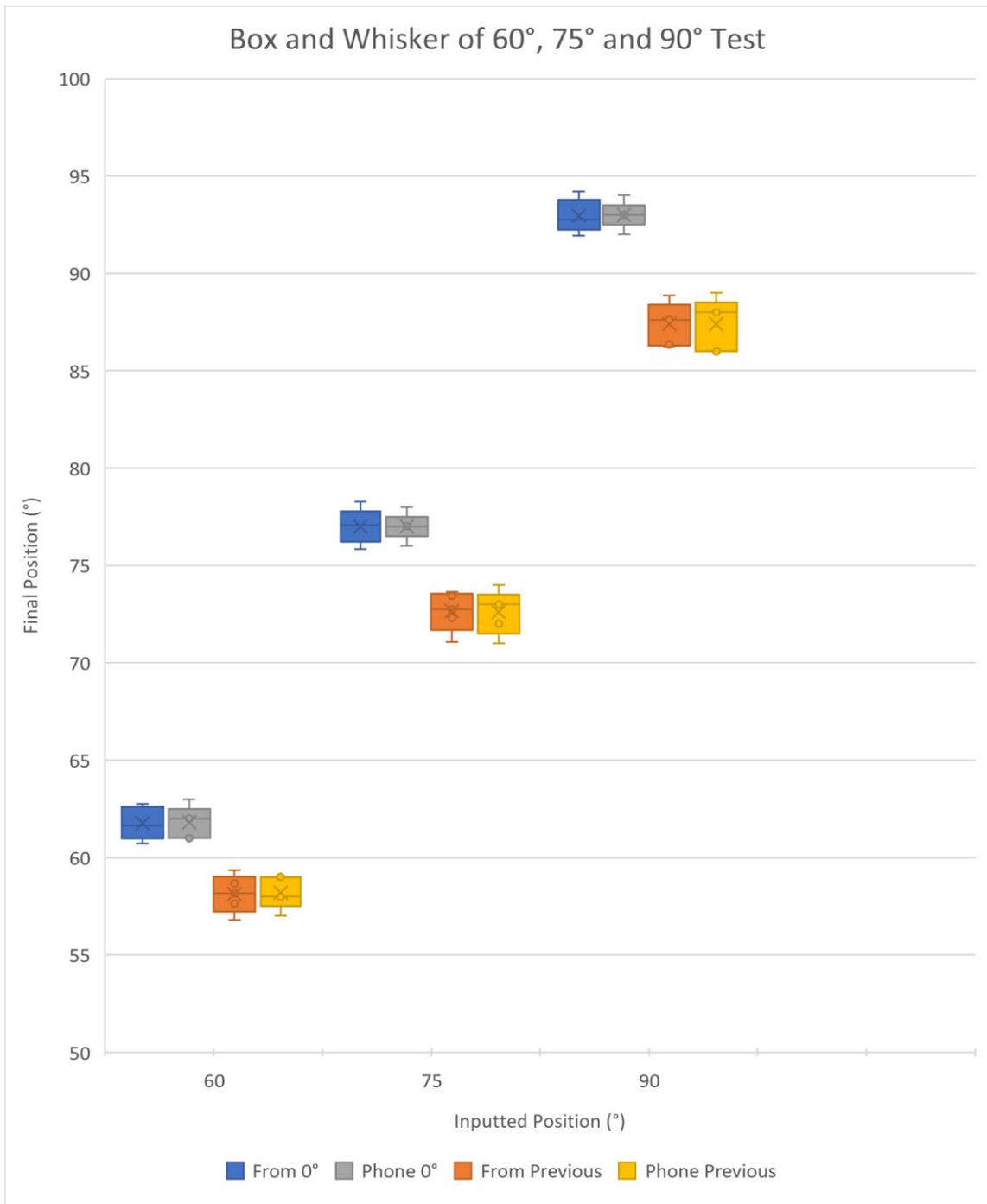


Figure 4-4: Box and Whisker of 60°, 75° and 90° Test

4.2.2.3. Azimuth Test

The azimuth axis has uniform mass acting in its direction of rotation, allowing the mechanics of motion to remain constant throughout its range. Values were recorded in Appendix C – Results, Table 6-7. The system was rotated to 15° starting at various intervals and plotted in Figure 4-5. As the starting position neared 15°, the range of rotated values gradually increased.

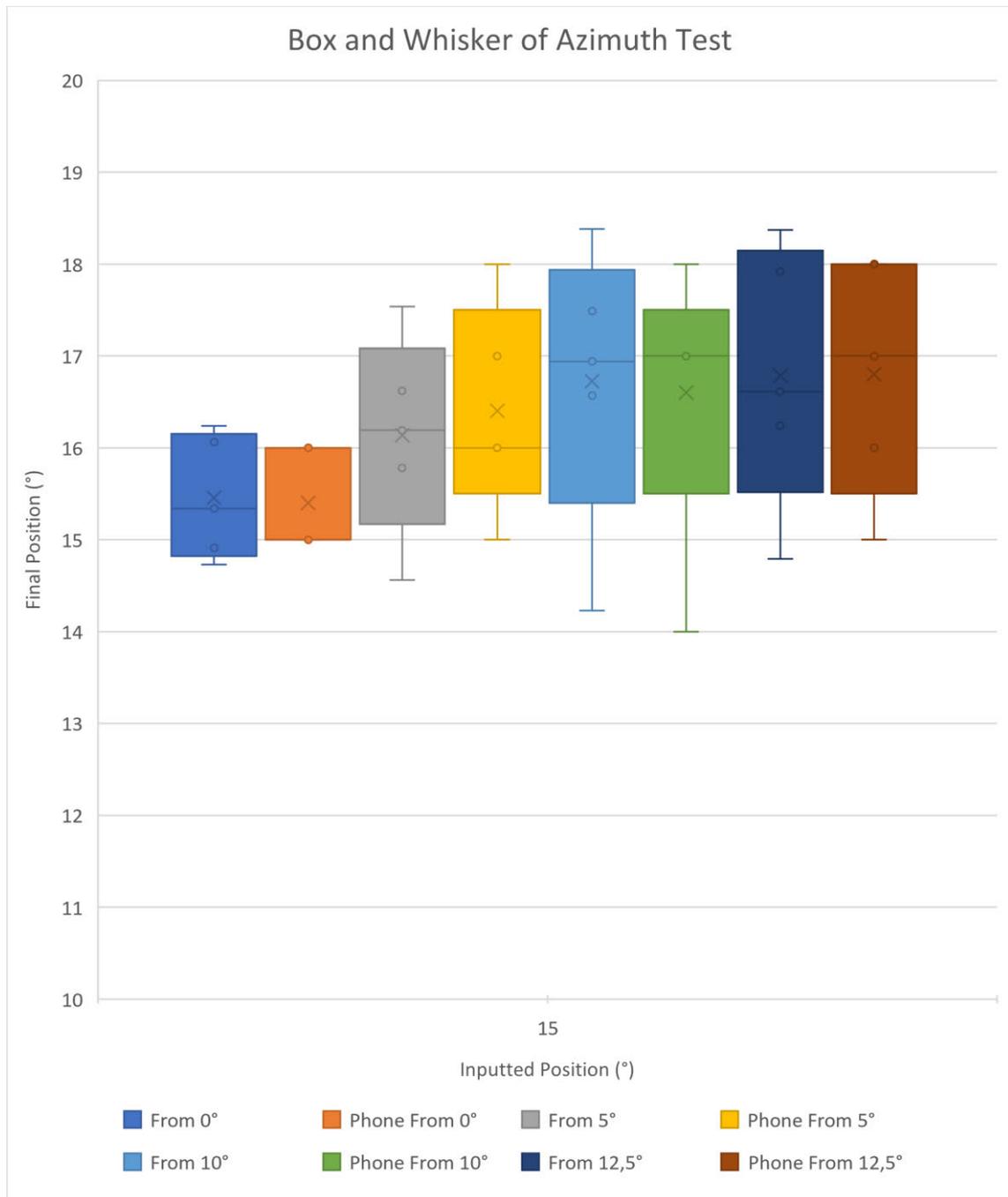


Figure 4-5: Box and Whisker of Azimuth Test

4.2.3. Orientation and Feedback Discussion

This test evaluated the motion of the system. Comparing the results and deviance from each test showed a few trends in the system's behaviour. Analysing the values in Table 6-1, Table 6-2 and Table 6-3 indicated that when the elevation axis moves from 0° , it tends to overshoot the desired value. When moving from a previous angle, it tends to undershoot the position. This is also evident in Table 6-4, Table 6-5 and Table 6-6 for larger movement angles.

The azimuth axis has a uniform mass acting upon its axis of rotation. This allowed tests from 0° to 15° to apply to other increments. Table 6-7 shows that the overshoot value increases as the starting value nears the target value. Comparing the values from the phone sensor to the IMU showed that the results were accurate. Due to the phone sensor displaying values to the nearest whole number, the resulting precision is lower when compared to the two decimal points of the IMU. Most of the results obtained were within the ± 2 -degree accuracy of the IMU sensor, as stated by the manufacturer (Pololu, 2021).

The few outliers visible in Figure 4-3, Figure 4-4 and Figure 4-5 resulted from the mechanism. The slight backlash meant that the axis would only stop moving when a pair of gear teeth were perfectly meshed. External forces such as gravity, inertia, and momentum play a significant role in the overall mechanics of the motion. Despite being balanced, once in motion, the inertia of the antenna and telescope put a significant load on the mechanism when stopping. The system does fulfil the requirement to be able to accurately rotate 360° in the azimuth direction and 90° in the elevation direction.

4.3. Satellite Tracking Test

A range of satellites were chosen, and the system was used to track them. Satellites that broadcasted radio signals were selected for monitoring.

4.3.1. Experimental Procedure

The steps below were followed to perform the test:

1. The system was moved outside and powered on, as shown in Figure 4-6.
2. Visible satellites were determined using the iOS app, Satellite Pointer.
3. Gpredict was opened on the Raspberry Pi to select a satellite to track.
4. The phone was secured to the antenna mount to verify the direction, seen in Figure 4-7.
5. The selected satellite was tracked for the period.
6. Azimuth and elevation data of both the satellite and system were recorded at 1-minute intervals.
7. The test was repeated for other satellites.



Figure 4-6: System Outside

Table 6-8 in Appendix C – Results, lists the recorded tracking values. The positions for Meteor M2 were plotted in Figure 4-9 and NOAA 18 in Figure 4-10. These charts show the relationship between the expected satellite position and the actual value of the system.

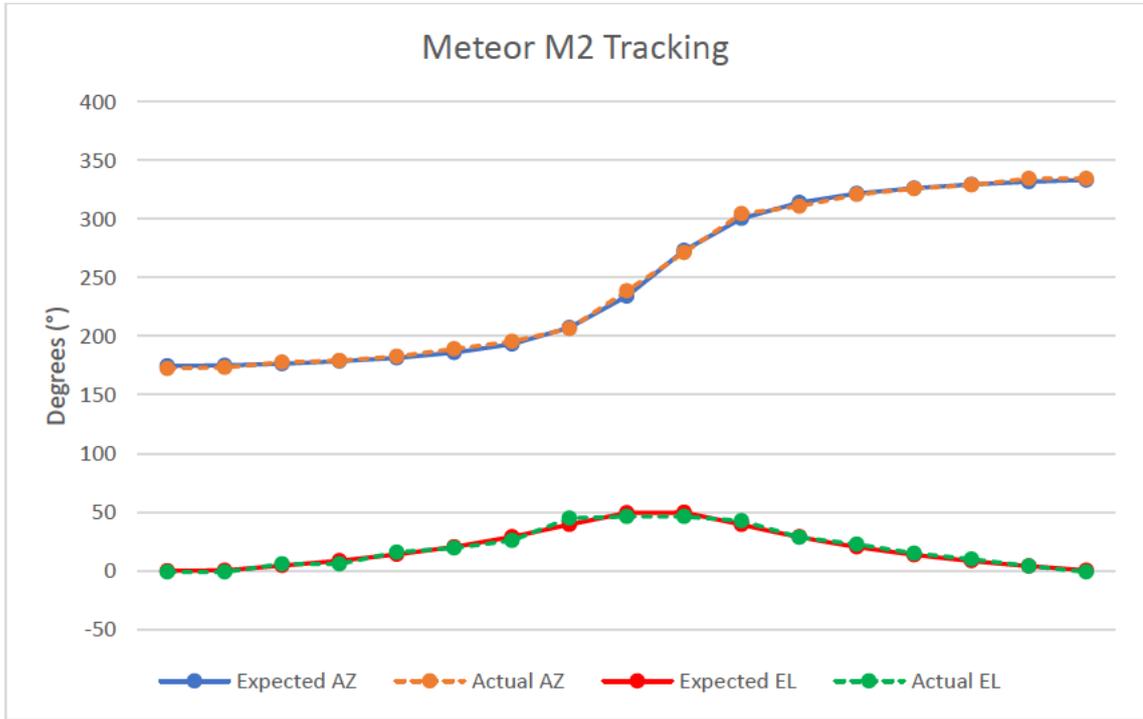


Figure 4-9: Meteor M2 Tracking

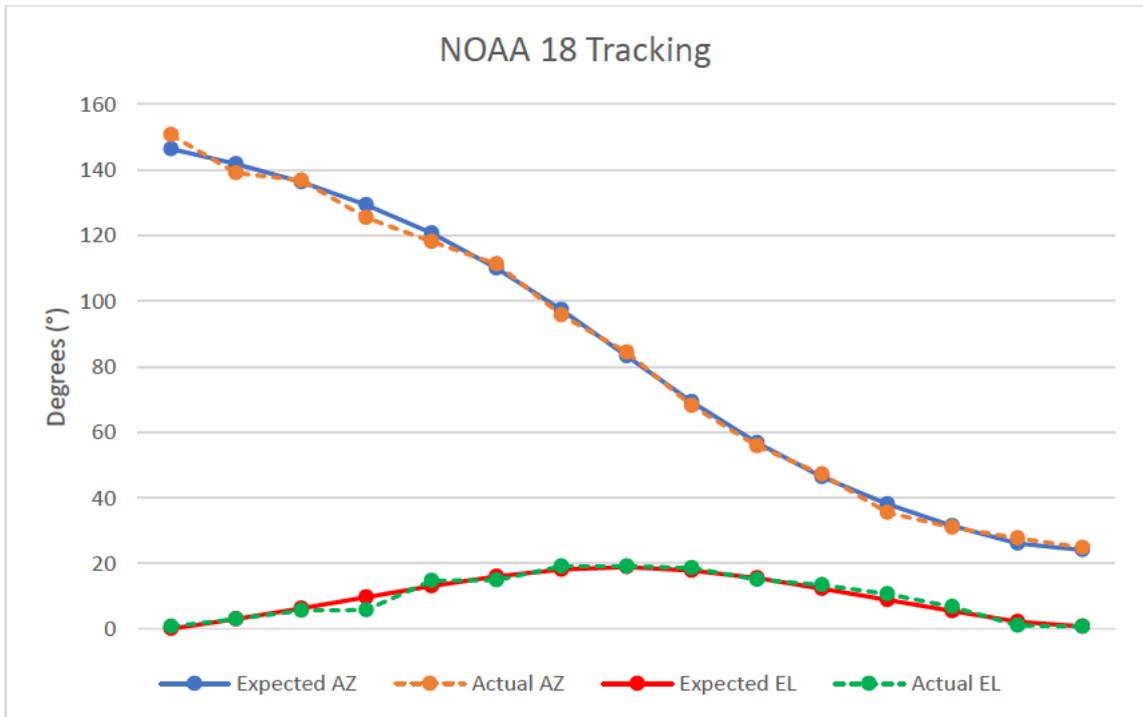


Figure 4-10: NOAA 18 Tracking

Data for NOAA 19 is presented in Figure 4-11, and the ISS in Figure 4-12. These charts also show the relationship between the expected satellite location and the actual value that the system moved to.

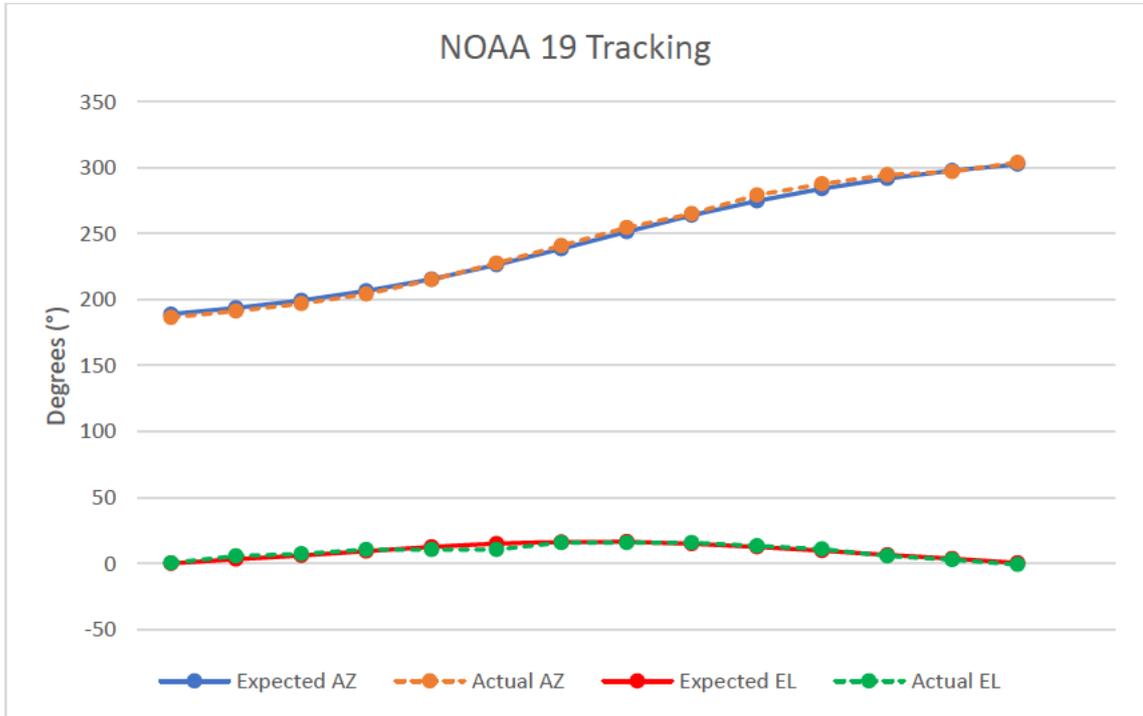


Figure 4-11: NOAA 19 Tracking

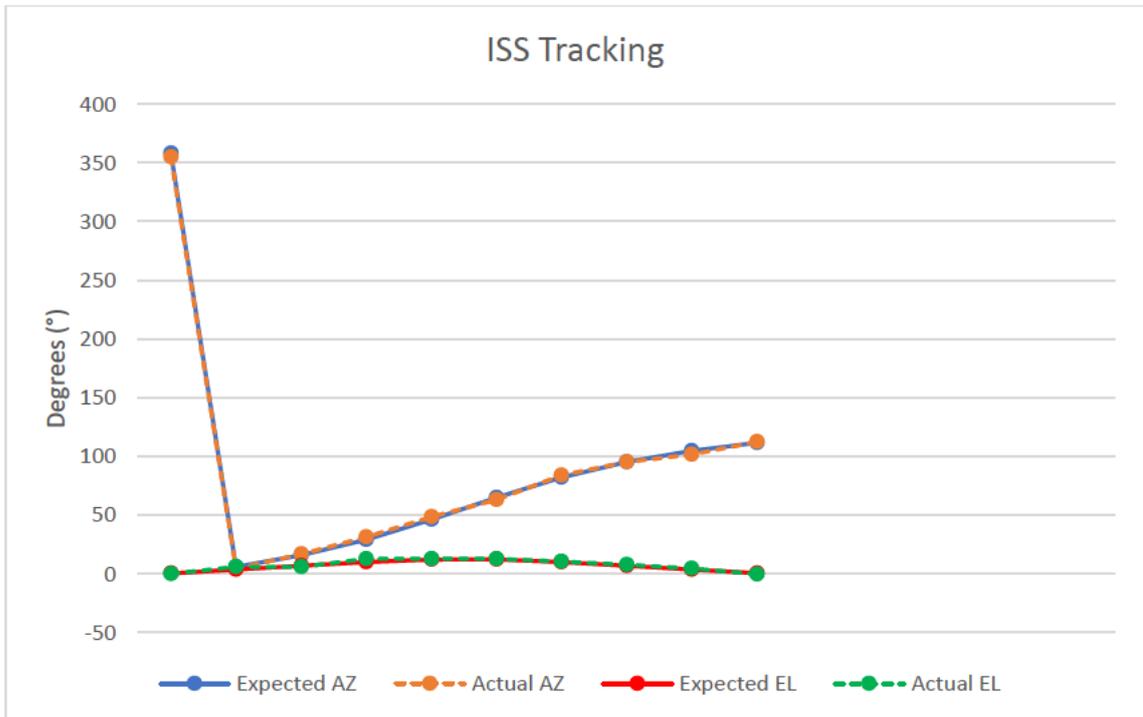


Figure 4-12: ISS Tracking

4.3.3. Satellite Tracking Discussion

This test demonstrated the tracking capability of the overall system. The system was moved to the required points indicated by the satellite position overhead.

The results were recorded in Table 6-8. Figure 4-9, Figure 4-10, Figure 4-11, and Figure 4-12 exhibited a trend where the elevation did not move upwards if the step size was too small. This is partially due to the stall torque being too large to overcome when the motion required is minuscule. This behaviour is rectified when the elevation arm is moving downwards and the motor is acting in conjunction with gravity.

The motion in the azimuth direction was consistent. The values showed a similar accuracy to that of the previous test. The angles moved by each axis were still within the 5° range.

The large difference in azimuth values seen in Figure 4-12 was due to the system rotating past 360° and back to 0°

Satellites that broadcast signals, such as Meteor M2, NOAA 18, NOAA 19 and the ISS, were selected for this test. This was done to allow for transmissions to be downloaded to further confirm the tracker orientation in the Radio Reception Test.

4.4. Radio Reception Test

The audible broadcasts and sounds were recorded to verify the position of certain satellites and validate the operation of the system to receive radio signals.

4.4.1. Experimental Procedure

The steps below were followed to perform the test:

1. A satellite from the Satellite Tracking Test was selected on Gpredict.
2. The frequency was tuned to the broadcasting downlink transponder.
3. A clip of the transmission was recorded and shown in Figure 4-13.
4. Using an online waveform creation tool, WaveVisual (Tunca, 2021), the clip was converted to “visualise the sound” for better representation on paper.

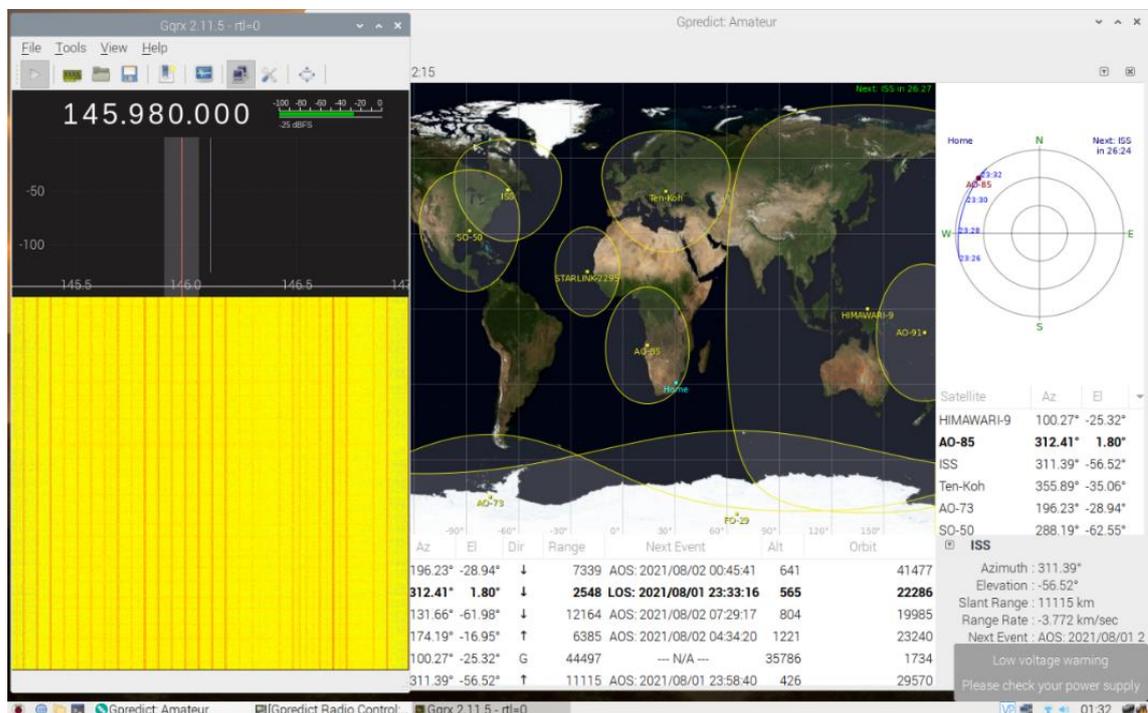


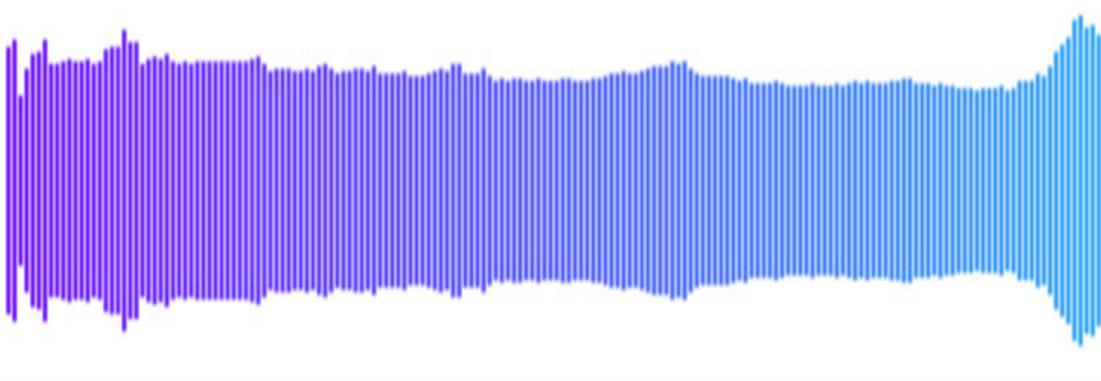
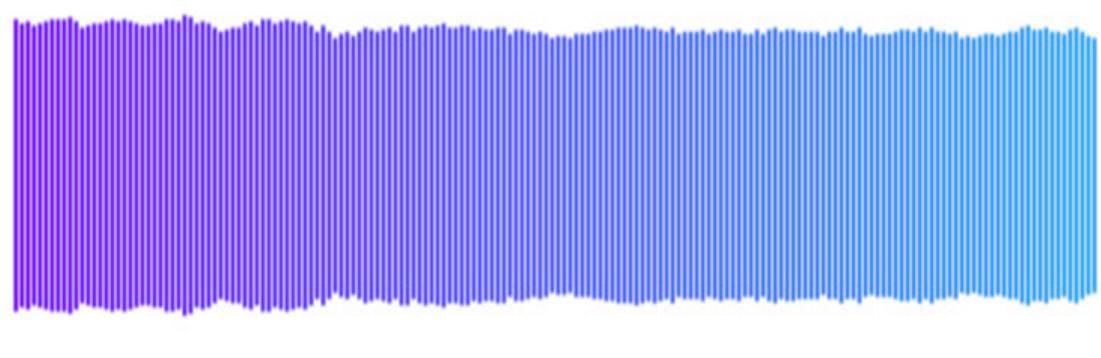
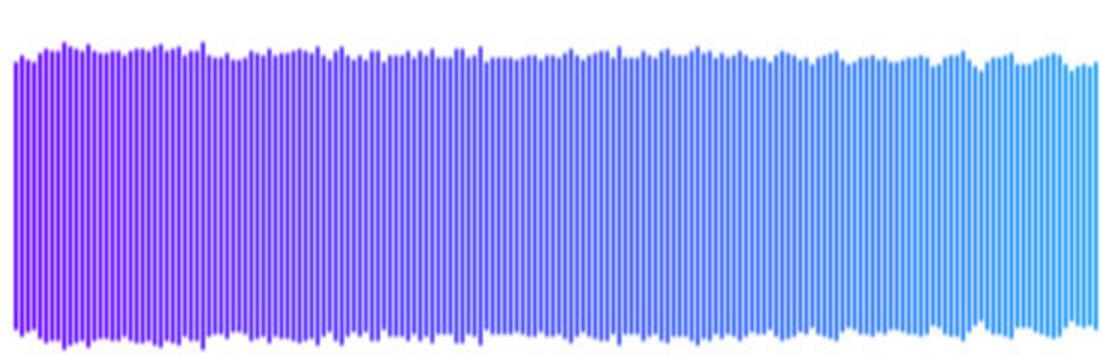
Figure 4-13: Radio Reception Waterfall and Gpredict

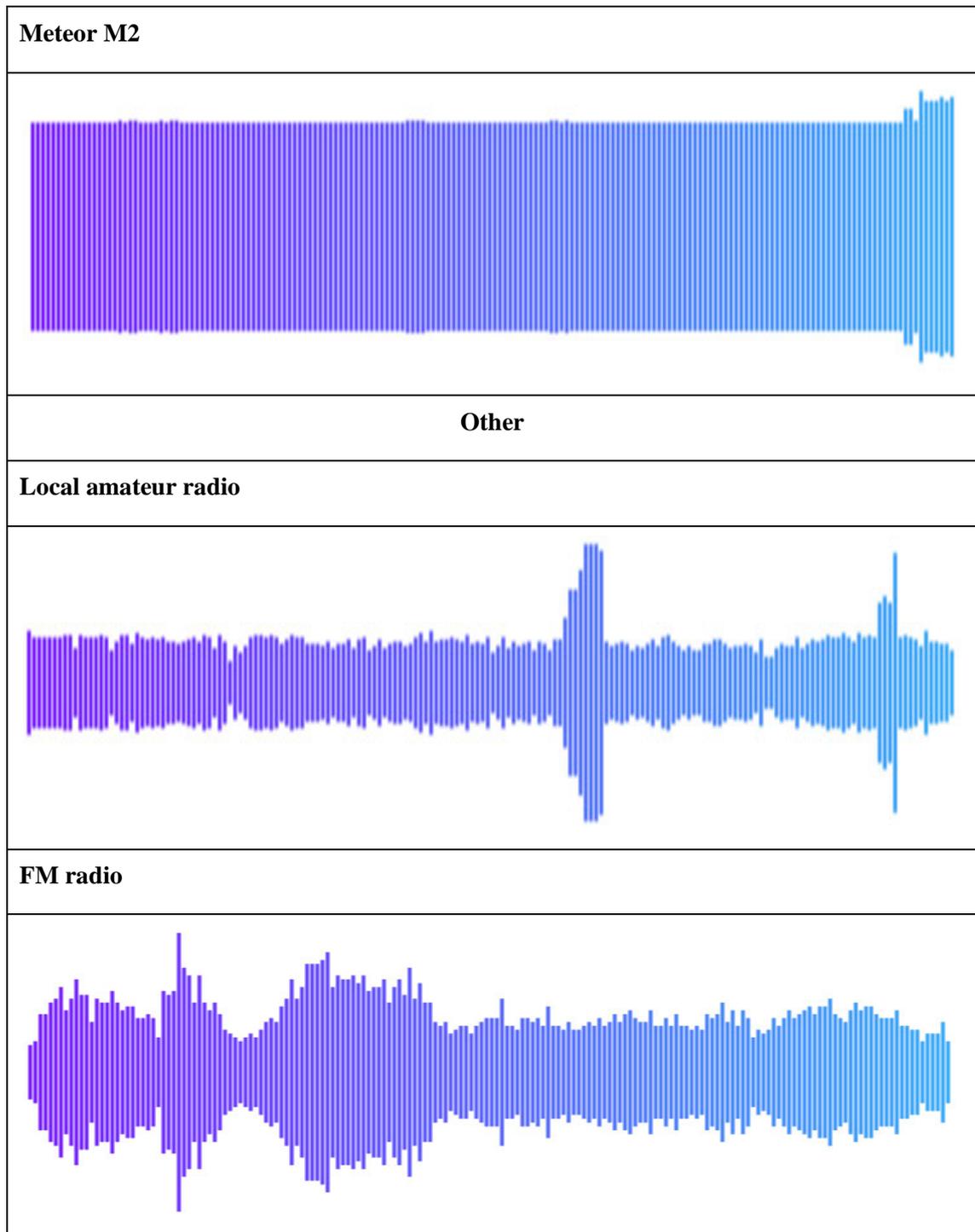
4.4.2. Radio Reception Results

Transmissions from the satellites tracked in the previous test were recorded. The documented satellites did not pass overhead within the same test period, so additional recordings were taken at a later stage. Local FM radio and amateur radio broadcasts were also recorded. The audio clips were converted into their waveforms to visualise the transmission, as seen in Table 4-1.

The sound clips are available at: https://stuukznac-my.sharepoint.com/:f/g/personal/216004605_stu_ukzn_ac_za/Er5PZoVWia9MnTGqSic6URMBsZw0kscVa9VPwY57ORLMfg?e=d4ExBv or https://drive.google.com/drive/folders/1yCP_EO-uUbk21xzCndKR4dIRhPjShVAi?usp=sharing

Table 4-1: Waveforms of recordings

Satellites
ISS

NOAA 18

NOAA 19




4.4.3. Radio Reception Discussion

The test captured recordings of various satellites and other broadcasts. Since audible media cannot be displayed on paper, the transmissions were converted into waveforms to be visualised in Table 4-1, as demonstrated by White (2018). A monotonous tone is visible in the ISS, NOAA 18, NOAA 19, and Meteor M2 waveform, which coincides with the sound in the audio recorded. The peaks and troughs from a local amateur radio broadcast and FM radio depict the presence of a voice that was heard in the recording.

4.5. SSTV Test

A transmission from a satellite broadcasting a slow-scan image was to be recorded and decoded. Finding a satellite that was scheduled to broadcast a transmission was critical for this test

4.5.1. Experimental Procedure

The steps below were followed to perform the test:

1. A period at which an SSTV broadcast was to be scheduled to be transmitted was required.
2. The ISS was scheduled to broadcast between the 26th and 31st of December 2021.
3. Pass predictions were checked on Gpredict to determine when the satellite was due to pass over the sky above.
4. The RTL-SDR was tuned to 145,800 MHz.
5. Analysing the waterfall in GQRX allowed the frequency to be adjusted to compensate for the Doppler shift of the signal.
6. The satellite transmission was recorded.
7. Using the QSSTV decoder, Figure 4-14, the image was decoded using the PD120 protocol.
8. The received SSTV image was analysed and assessed.

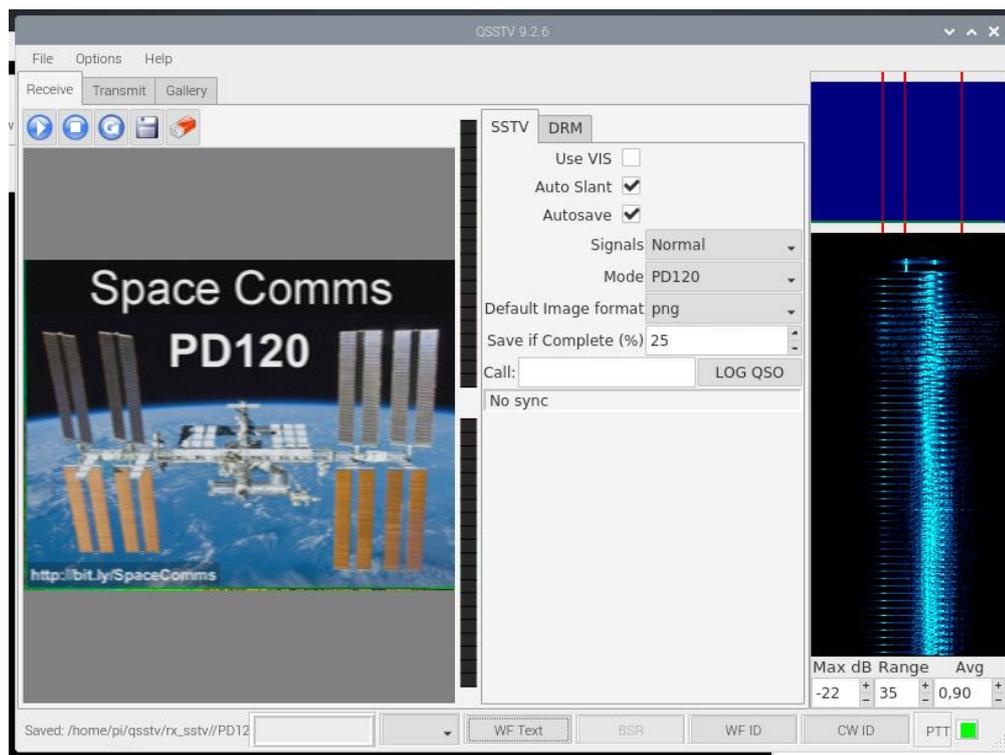


Figure 4-14: QSSTV Decoder

4.5.1. SSTV Results

An ISS SSTV transmission was recorded on the 27th, 28th and 30th of December, seen in Figure 4-15. The broadcast was decoded in Figure 4-16, Figure 4-17, and Figure 4-18.

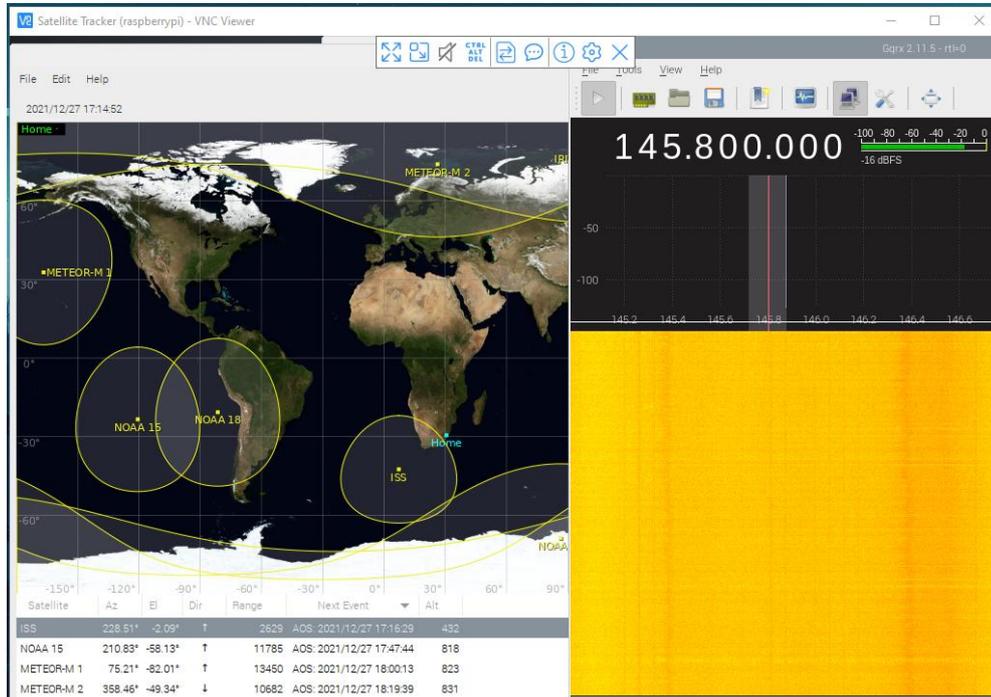


Figure 4-15: ISS SSTV Radio Transmission

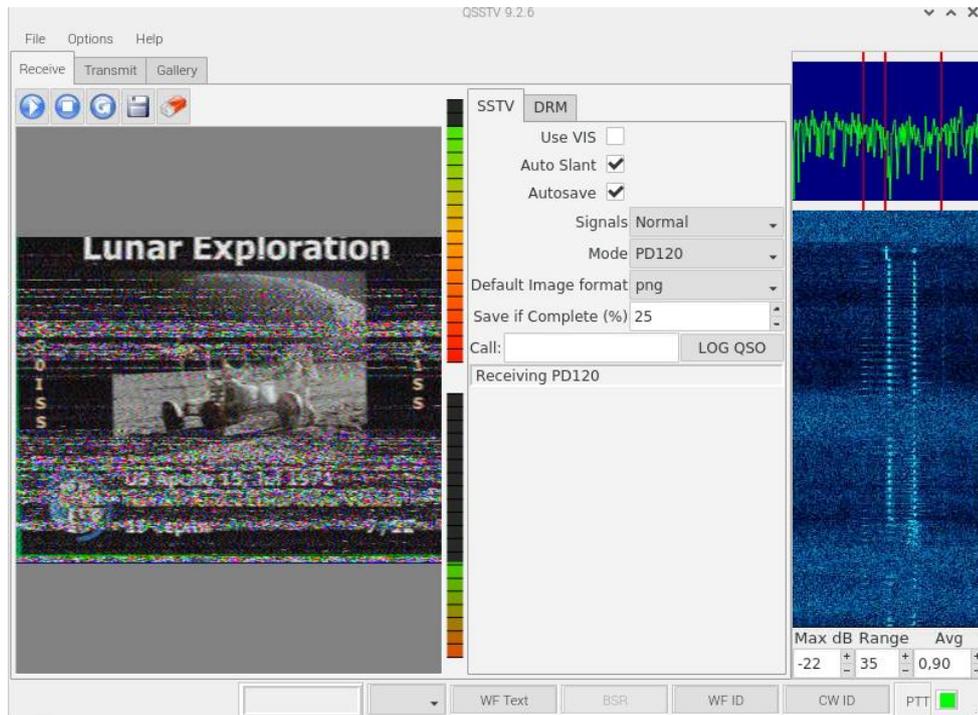


Figure 4-16: ISS SSTV Decoding, 1

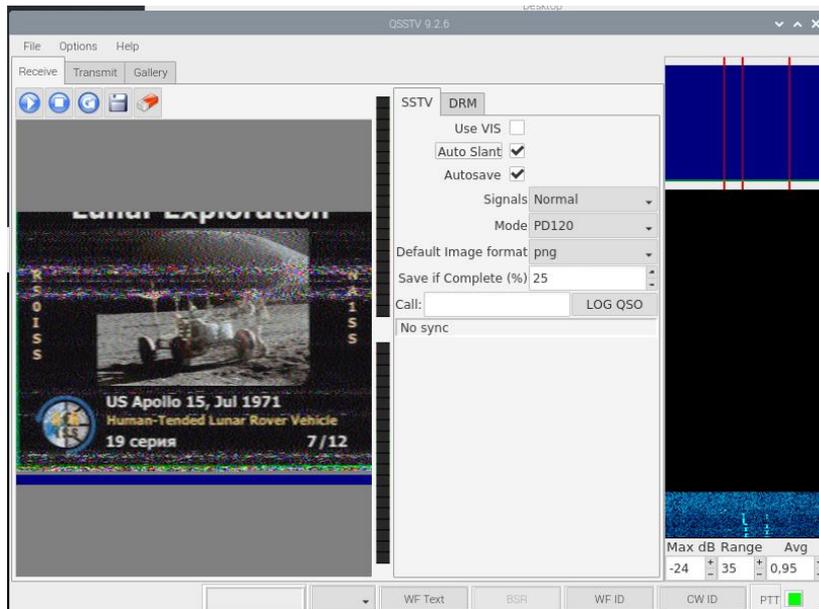


Figure 4-17: ISS SSTV Decoding, 2

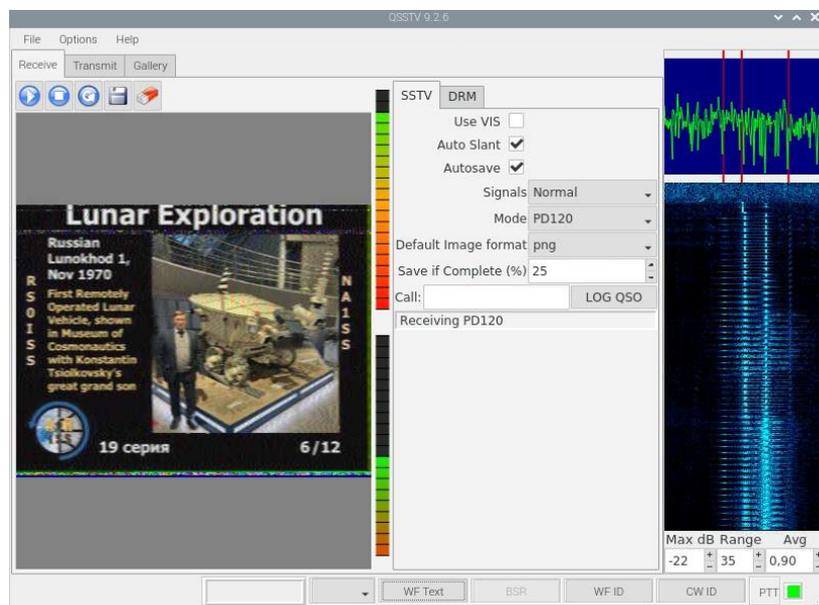


Figure 4-18: ISS SSTV Decoding, 3

4.5.1. SSTV Test Discussion

The SSTV test assessed the ability to decode SSTV images. The ISS recording from the previous test was a broadcast containing a slow scan image. Figure 4-16 was the first recording taken. Although the image is visible, there was fuzziness around the middle and bottom, resulting from electrical interference or signal obstruction from clouds. Figure 4-17 and Figure 4-18 were recorded on subsequent days. The Lunar Rover image is much clearer when compared to the previous recording. On the third pass of the ISS, the image being broadcasted was changed.

4.6. Frequency ID Test

Using the STRF program, frequency identification was performed. Recordings of IQ data were taken to further analyse and ultimately determine the satellite being tracked.

4.6.1. Experimental Procedure

The steps below were followed to perform the test:

1. The system was set up, and the rotator was disabled.
2. The strf program was run.
3. An IQ recording was taken for a period of 10 minutes to generate .bin files, as seen in Figure 4-19.
4. The recording was then processed and visualised by rfplot.
5. Frequencies within the recorded range were analysed by rffit, to determine the closest satellite match, identified by its NORAD ID.

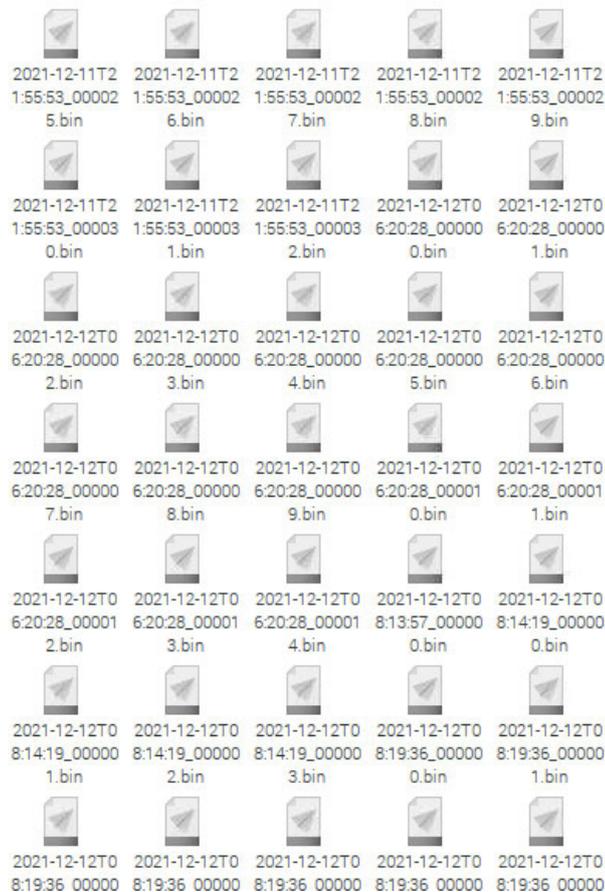


Figure 4-19: IQ Files

4.6.2. Frequency ID Results

The frequency of 137.900 MHz was selected for analysis as it was a known frequency at which NOAA 18 broadcasts. The STRF program was run for 10 minutes, allowing IQ data to be recorded. The data was viewed using rfplot, in Figure 4-20, with analysis of the frequency performed by rffit, in Figure 4-21.

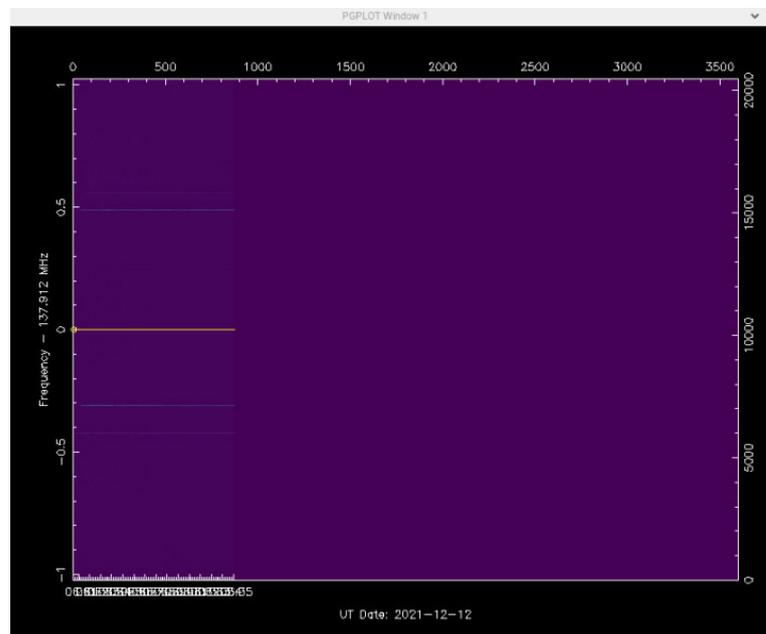


Figure 4-20: Frequency Recording

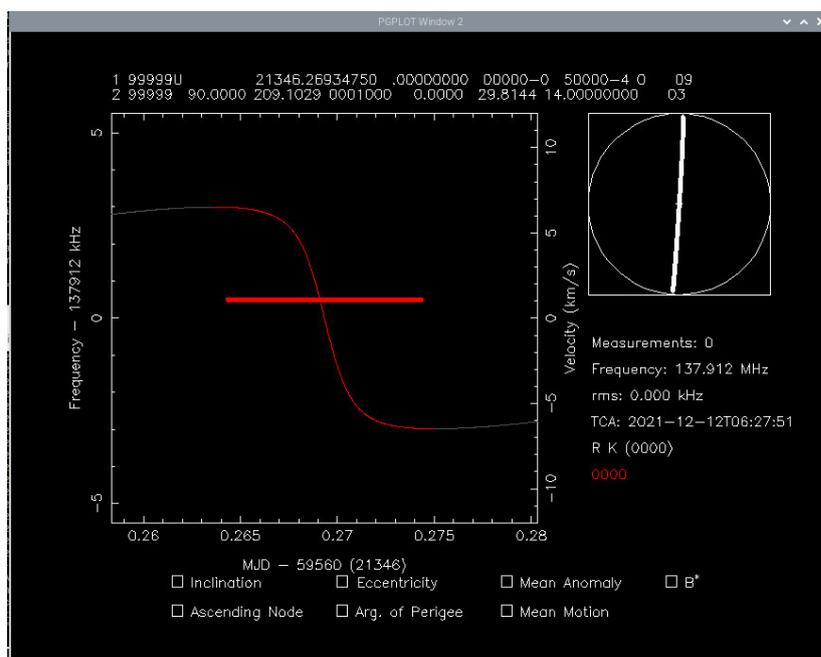


Figure 4-21: Frequency Analysis

The final satellite determination was accomplished in Figure 4-22. Searching the database for the given NORAD ID confirms the satellite to be NOAA 18, in Figure 4-23 (N2YO, 2021).

```

pi@raspberrypi ~
File Edit Tabs Help
26936 0.000 kHz 137.912526 MHz
26939 0.000 kHz 137.912521 MHz
26985 0.000 kHz 137.912502 MHz
26987 0.001 kHz 137.912587 MHz
26988 0.002 kHz 137.912357 MHz
26989 0.000 kHz 137.912690 MHz
26992 0.001 kHz 137.912564 MHz
27025 0.023 kHz 137.913536 MHz
27026 0.020 kHz 137.912513 MHz
27027 0.026 kHz 137.913176 MHz
27029 0.019 kHz 137.912521 MHz
27030 0.019 kHz 137.912433 MHz
27032 0.020 kHz 137.912268 MHz
27033 0.019 kHz 137.912583 MHz
27040 0.021 kHz 137.913361 MHz
27041 0.021 kHz 137.912157 MHz
27043 0.023 kHz 137.911171 MHz
27044 0.019 kHz 137.912829 MHz
27046 0.036 kHz 137.911332 MHz
27047 0.023 kHz 137.912834 MHz
27048 0.036 kHz 137.912205 MHz
27049 0.023 kHz 137.912974 MHz
27051 0.027 kHz 137.913117 MHz
Identified 163 candidate(s), best fitting satellite is 28654

```

Figure 4-22: Orbit Determination

1 match(es)

Name	NORAD ID	Int'l Code	Launch date (YYYY-MM-DD)	Status	Action
NOAA 18	28654	2005-018A	2005-05-20	IN ORBIT	TRACK IT

Figure 4-23: Satellite ID

4.6.1. Frequency ID Discussion

The frequency identification test used the STRF tool. The program was run for 10 minutes, collecting approximately 500 MB of IQ data. The data was viewed in Figure 4-20, analysed in Figure 4-21, and then processed in Figure 4-22.

This test followed a similar verification process as demonstrated in the *Doppler Estimation for Passive RF Sensing Method in Space Domain Awareness* (Islam, et al., 2020). Selecting a known frequency of 137.900 MHz (NOAA 18) ensured that a valid broadcast could be evaluated.

Searching the given NORAD ID indicated that the frequency signature was NOAA 18, Figure 4-23. The result ties in with the hypothesis frequency since NOAA 18 was the only satellite passing over and broadcasting near the selected frequency when the test was performed.

Analysing the IQ data to predict the satellite that it belongs to is a CPU intensive process. A period of 30 minutes was required to determine the result of the test performed, which is a limitation of the mobile-grade processor on the Raspberry Pi.

The main obstacle is “space traffic”. Seventy-five active satellites were added to Gpredict to understand the congestion of the skies above, shown in Figure 4-24. Ten satellites were in the system's line of sight when the screenshot was taken.

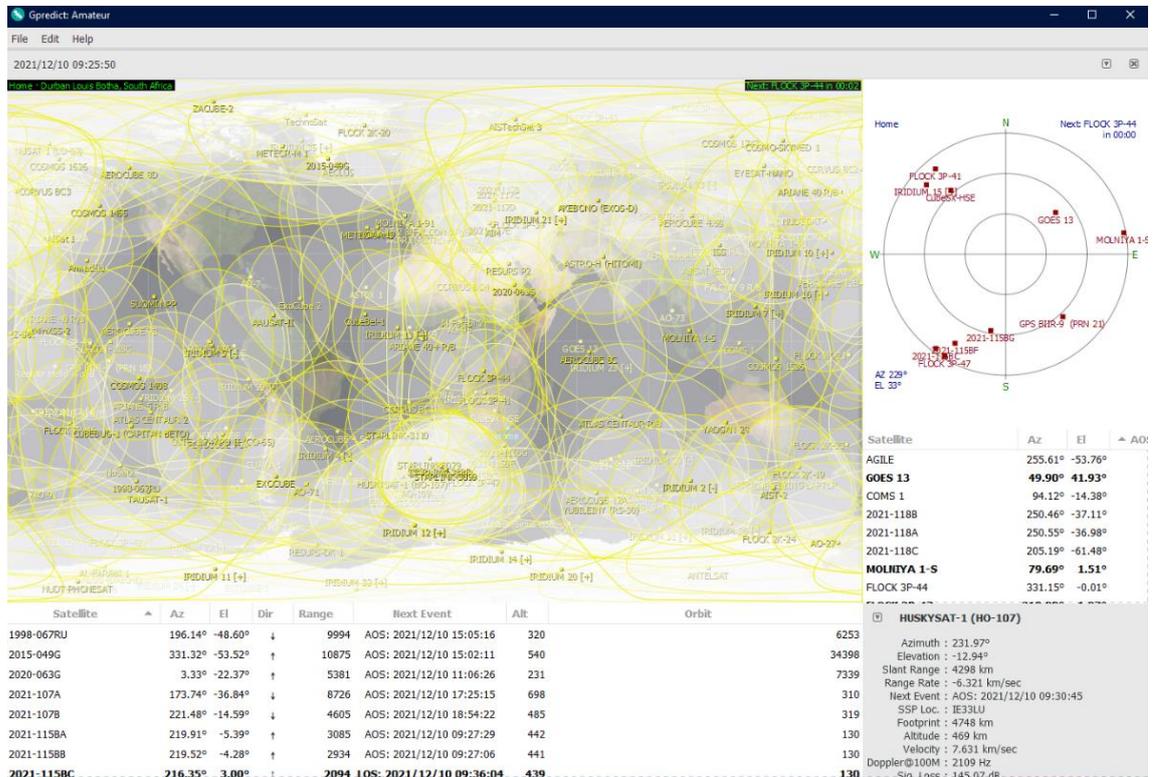


Figure 4-24: 75 Satellites Added

4.7. Chapter summary

The results and data show that all aspects of the system work as expected. The motion and tracking were accurate to within 5° of a chosen target. Analysing the motion data showed characteristic trends. The elevation axis tended to overshoot the desired value when moving from 0° while moving from the previous angle led to an undershoot of the position. The slight backlash in the mechanism meant that the axis would only stop moving once a pair of gear teeth were perfectly meshed. Planetary gearboxes were explored, but the cost and size of a compatible unit was unfeasible. The system does fulfil the requirement to be able to accurately rotate 360° in the azimuth direction and 90° in the elevation direction.

Given the propagation of signals from a satellite, and the results obtained in the radio tests, the accuracy was acceptable. Transmissions from satellites such as the ISS, NOAA 18, NOAA 19, and Meteor M2 were recorded. An ISS transmission that was recorded contained a slow scan image when decoded. This confirms the operation of the radio and SSTV functions.

While frequency identification does work, only certain satellites broadcast transmissions. This limits the usage of some weather satellites and broadcasts from the ISS or similar voice downlinks. The large size of the data files produced and long processing times to evaluate the signal limit the use of the tool.

The system's overall performance was as expected, accomplishing the tasks that were required from it. While the elevation axis struggled to move between small increments, the overall operation was acceptable. The selection and use of specific electronic components were also validated.

Tests were repeated at least to increase the reliability of the results. Multiple recordings were taken to ensure the accuracy of the data. Finally, the correct use of the apparatus and test equipment ensured that the methods used and the results obtained were valid.

The next chapter presents a further evaluation of the results in relation to the aims and objectives of this study. A brief comparison is made between similar systems, allowing for conclusions to be drawn. Recommendations for further developing and improving this system are also mentioned.

5. Conclusion and Recommendations

Many aspects were successfully integrated to create a remotely accessible satellite tracking system. A brief comparison is made between similar systems, with recommendations to further this research mentioned below.

5.1. Conclusion

Creating a remotely accessible satellite tracking system was the motivation behind this research. Many hardware and software systems were researched and selected to develop an optimised solution to the task at hand. Calculations were done to determine the power draw of the system from the battery, and thus an appropriate circuit was designed to connect the power and signal lines for the chosen components. The required elevation torque value was calculated to determine the motor suitability. The entire system was disassembled and reconstructed, with certain parts being replaced to ensure that it was fully operational. Verification of device functionality was performed before software integration to facilitate communication. Various aspects of the system were iterated and improved on until they met the designed operating parameters.

Table 3-1 listed the mechanical design requirements of the system. The Orientation and Feedback Test confirmed the ability of the system to rotate 360° in the azimuth direction and 90° in the elevation direction. Costs were kept to a minimum by reusing and recycling parts from older projects wherever possible. Ample space was allocated for the electronics and wiring, as seen in Figure 3-40 and Figure 3-41. Almost all the parts selected for use were off the shelf items. A few components, such as the motor mount brackets, were custom fabricated.

The electronic design requirements are listed in Table 3-3. The Raspberry Pi 4 was chosen for use as the control computer. Both wired and wireless network support enabled easy remote access through RealVNC. The Arduino Pro Micro was used to control the motor drivers and IMU. A 12 V wiper motor was used to move each axis. The GPS module was connected to the serial pins on Raspberry Pi to provide location coordinates. The electronics were connected to a 12 V battery, with regulators to supply the required voltage to components.

The software design was led by parameters detailed in Table 3-5. TrackGUI was created using Python to allow users to easily log in and operate commonly used tracking and monitoring functions. Rotator control was integrated using C++ and the EasyCommII protocol. The ability to receive radio transmissions was incorporated using the RTL-SDR, with results shown in the Radio Reception Test. The SSTV Test demonstrated the reception and decoding functionality. The Frequency ID Test displayed the operation of the frequency identification utility.

Testing and results were shown in Chapter 4. The results confirmed that the system works well and effectively achieves the aims and objectives specified.

Finally, similar antenna rotator systems are compared in Table 5-1.

Table 5-1: System Comparison

	Current System	SatNOGS v3	SATRAN v1
Tracking	Yes	Yes	Yes
Accuracy	5°	1°	1.8°
Telescope Mount	Yes	No	No
Material	Aluminium	3D printed + Aluminium	3D printed
Power Source	12 V battery	DC 48V adaptor	DC 9V adaptor
Remote access	VNC	Website	Android App
Controller	Raspberry Pi	Raspberry Pi + Custom PCB	Custom PCB
Frequency ID	Yes	No	No
SSTV	Yes	Yes	Yes
Size	660 mm x 330mm x 330 mm	280 mm x 140 mm x 140 mm	180 mm x 150 mm x 150 mm
Cost	US\$ 1195	US\$ 500	US\$ 172

The designed system is larger than other systems, but it offers unrivalled stability and strength due to the metal construction. The ability to support a telescope is something that the other systems do not offer. Since the SatNOGS and SATRAN systems utilise 3D printed parts for their structure and mechanism, it reduces the overall rigidity and introduces points of failure.

A 12 V car battery powering the current system makes it portable for remote use, whilst the other systems require an AC-DC adaptor to provide power.

The additional cost is due to the prices of materials and components in South Africa. Revisions and modifications could be made to reduce the cost of the assembly. Since the other systems are commercially available products, supply chain discounts greatly reduce manufacturing costs.

Numerous changes, modifications and improvements were made throughout the system to ensure that it functions as required, allowing for the following to be achieved:

- Integration of hardware and electronic components to create a system used to track satellites and listen in on their radio transmissions.
- Implementation of low-powered electronic components which enables remote operation and access via a 12 V battery.
- A remotely accessible graphical user interface for various tracking tools, with an admin and user access hierarchy.

5.2. Recommendations

The system is limited to operating within an area with a usable Wi-Fi network. A 3G/4G mobile Wi-Fi router would expand the usable range of the system to wherever there is cellular coverage.

Similarly, implementing a solar charger would enable the system to be independent of electricity from the power grid.

A small touchscreen LCD could be connected to the Raspberry Pi to allow direct access to the system. It will allow an operator to edit Wi-Fi configurations or alter operating parameters without the need for a network connection and a laptop or phone.

Despite being a mobile system, the mass and size of the steel stand make it difficult to move. A smaller, lighter structure would make transportation and usability much easier.

Users requiring a password to access the Raspberry Pi VNC and TrackGUI add a layer of security. Data security could be further researched to encrypt the TrackGUI program dependencies and libraries.

The other systems use stepper motors to rotate their axes and obtain a high level of accuracy. Torque demands mean that a suitable stepper for this system would require 48V to maintain a safe current value, like the SatNOGS system.

6. Appendices

The following appendices contain calculations, drawings, results and code used in this report.

6.1. Appendix A – Calculations

Determining operating values and expected results allow for more informed decisions to be made when components are selected, and parts are designed.

6.1.1. Power Draw

The voltage and current requirements of each component in the system will contribute to the overall power draw of the system. Devices such as the Arduino and Raspberry Pi operate on 5V, with the two motors requiring 12V. The total power can be calculated using the maximum current values as follows:

An Arduino requires 5V and can draw a maximum of 1A (Arduino, 2020).

$$P_{Arduino} = V \times I \quad (6.1.1-1)$$

$$P_{Arduino} = 5 \times 1$$

$$P_{Arduino} = 5 \text{ W}$$

The Raspberry Pi requires 5V and can draw a maximum of 3A (Raspberry Pi Foundation, 2021).

$$P_{RPi} = V \times I \quad (6.1.1-2)$$

$$P_{RPi} = 5 \times 3$$

$$P_{RPi} = 15 \text{ W}$$

A 12 V wiper motor can draw 10 A of current when starting.

$$P_{motor} = V \times I \quad (6.1.1-3)$$

$$P_{motor} = 12 \times 10$$

$$P_{motor} = 120 \text{ W}$$

Accounting for minor loads such as the GPS and IMU: 5W

$$P_{total} = P_{Arduino} + P_{RPi} + P_{motor} + P_{minor} \quad (6.1.1-4)$$

$$P_{total} = 5 + 15 + 120 + 5$$

$$P_{total} = 145 \text{ W}$$

Therefore, the system could draw, at maximum, 145W from the battery.

6.1.2. Elevation Torque

Previous groups were unsuccessful at moving the antenna and telescope assembly. After reducing the mass of the antenna, telescope, and mounts to 3.48 kg, it was rebalanced, allowing the mount to be secured at the centre of mass.

$$F = m \times g \quad (6.1.2-1)$$

$$F = 3.48 \times 9.81$$

$$F = 34.1388 \text{ N}$$

The antenna was secured to the mounting bracket. The resulting moment arm, from the end of the antenna to the shaft centre, was reduced to 525 mm.

$$\tau = F \times r \quad (6.1.2-2)$$

$$\tau = 34.1388 \times 0.525$$

$$\tau = 17.92 \text{ Nm}$$

Calculating the gear ratio using the number of teeth:

$$U = \frac{Z_2}{Z_1} \quad (6.1.2-3)$$

$$U = \frac{60}{40}$$

$$U = 1.5$$

Although the low-speed torque is approximately 23 Nm, using a value of 20 Nm will allow for more conservative values to be determined (Terry, 2019):

$$U = \frac{\tau_2}{\tau_1} \quad (6.1.2-4)$$

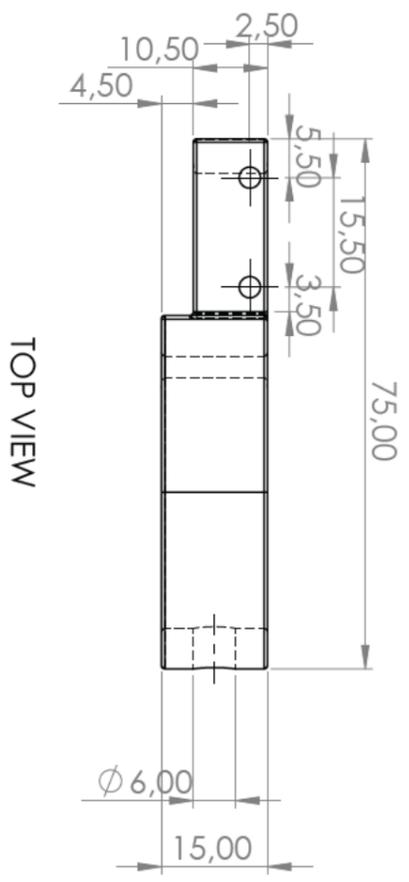
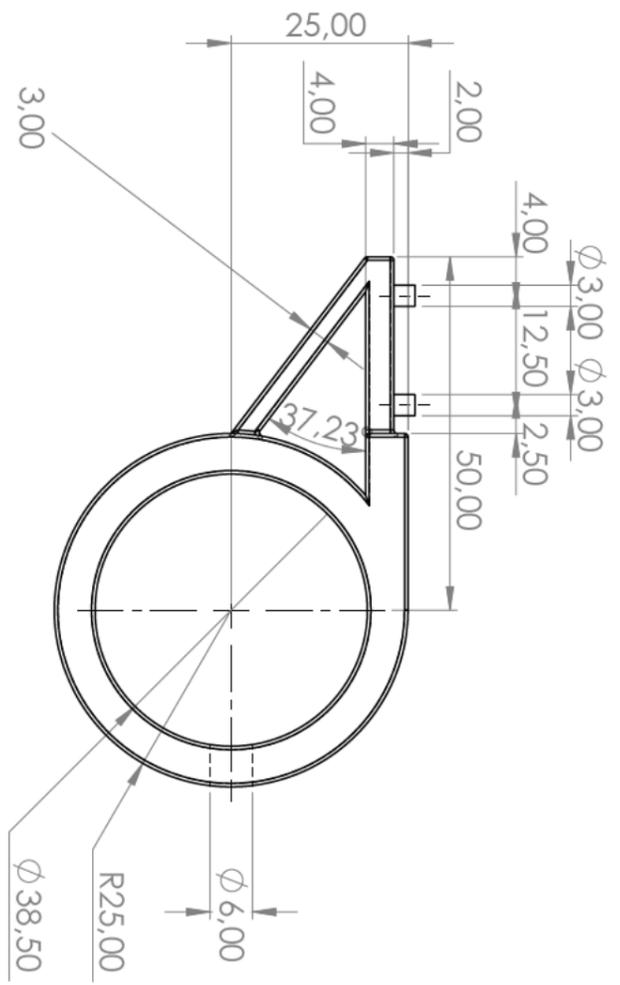
$$\tau_2 = U \times \tau_1$$

$$\tau_2 = 1.5 \times 20$$

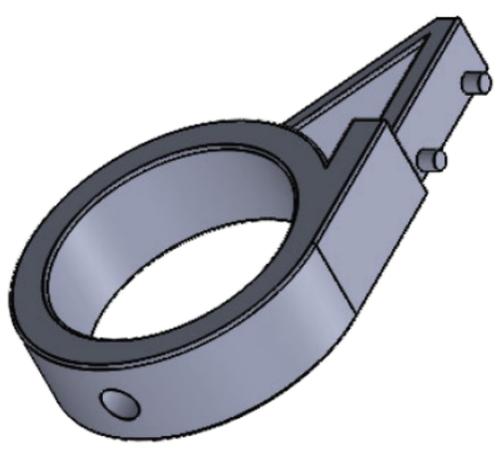
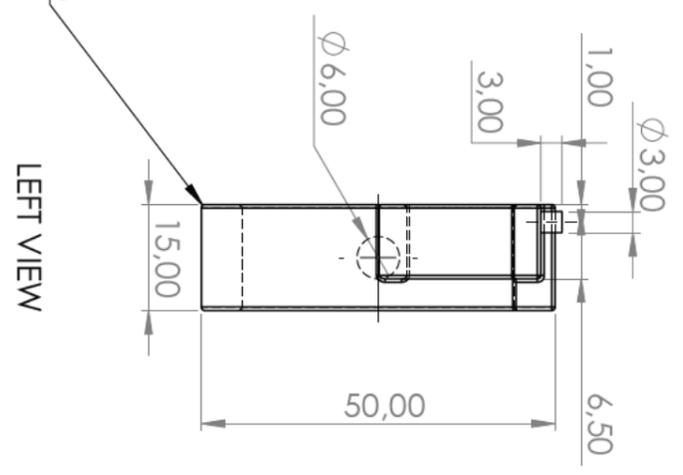
$$\tau_2 = 30 \text{ Nm}$$

Comparing the required and expected torque values indicated that an additional 12.08 Nm of torque is produced.

6.2. Appendix B – Drawings



0.5 mm FILLET ON ALL EDGES



UNLESS OTHERWISE STATED GENERAL TOLERANCES: ±0.5 mm ANGLES: ± 0.5°

UNIVERSITY OF KWAZULU-NATAL		MATERIALS: PETG		No. REQ.: 1		SCALE: 1:1		UNITS: mm		PROJECT: SYSTEM INTEGRATION OF A REMOTELY ACCESSIBLE SATELLITE TRACKER		No.:	
SCHOOL OF ENGINEERING		PROJECT SUPERVISOR		CHECKED		STUDENT NAME: RUSHIL KISOON				TITLE: IMU BRACKET		B1	
MECHANICAL ENGINEERING		WORKSHOP TECHNICIAN		TECHNICAL OFFICER		STUDENT No.: 216004605		E-MAIL: 216004605@stu.ujkn.ac.za				TEL. No.:	
SOLIDWORKS Educational Product. For Instructional Use Only.													

6.3. Appendix C – Results

The data used to produce the charts presented in chapter 4, Testing and Results, are shown below.

6.3.1. Orientation and Feedback Test Data

Elevation movement values of 15°, 30° and 45° were recorded in Table 6-1, Table 6-2, and Table 6-3, respectively. The results were plotted in Figure 4-3.

Table 6-1: 15° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	15	16.32	+ 1.32	16
2			15.57	+ 0.57	16
3			15.19	+ 0.19	15
4			16.47	+ 1.47	16
5			15.63	+ 0.63	16
1	7		14.69	- 0.31	15
2			13.42	- 1.58	13
3			14.76	- 0.24	15
4			13.93	- 1.07	14
5			14.21	- 0.79	14

Table 6-2: 30° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	30	30.87	+ 0.87	31
2			31.62	+ 1.62	32
3			30.66	+ 0.66	31
4			31.24	+ 1.24	31
5			31.76	+ 1.76	32
1	15		28.32	- 1.68	28
2			29.86	- 0.14	30
3			28.33	- 1.67	28
4			28.13	- 1.87	28
5			29.46	- 0.54	29

Table 6-3: 45° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	45	45.97	+ 0.97	46
2			45.39	+ 0.39	45
3			45.87	+ 0.87	46
4			46.69	+ 1.69	47
5			47.23	+ 1.23	47
1	30		44.18	- 0.82	44
2			44.67	- 0.33	45
3			43.53	- 1.47	43
4			43.26	- 1.74	43
5			44.35	- 0.65	44

Additional elevation values of 60°, 75° and 90° were recorded in Table 6-4,

Table 6-5, and Table 6-6, respectively. The results were represented in Figure 4-4 as a box and whisker diagram.

Table 6-4: 60° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	60	62.75	+ 2.75	63
2			61.64	+ 1.64	62
3			60.73	+ 0.73	61
4			62.45	+ 2.45	62
5			61.23	+ 1.23	61
1	45		57.64	- 2.36	58
2			59.35	- 0.65	59
3			56.79	- 3.21	57
4			58.16	- 1.84	58
5			58.67	- 1.33	59

Table 6-5: 75° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	75	77.28	+ 2.28	77
2			75.84	+ 0.84	76
3			77.06	+ 2.06	77
4			76.59	+ 1.59	77
5			78.27	+ 3.27	78
1	60		72.32	- 2.68	72
2			73.44	- 1.56	73
3			73.65	- 1.35	74
4			72.74	- 2.26	73
5			71.07	- 3.93	71

Table 6-6: 90° Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	90	92.76	+ 2.76	93
2			93.34	+ 3.34	93
3			92.55	+ 2.55	93
4			94.21	+ 4.21	94
5			91.94	+ 1.94	92
1	75		87.61	- 2.39	88
2			87.95	- 2.05	88
3			86.34	- 3.66	86
4			88.85	- 1.15	89
5			86.21	- 3.79	86

The azimuth axis has uniform mass acting in its direction of rotation, allowing the mechanics of motion to remain constant throughout its range. Azimuth values were recorded in Table 6-7, starting at various intervals and plotted in Figure 4-5.

Table 6-7: Azimuth Test Results

Run	Initial Position (°)	Input (°)	Final Position (°)	Deviance (°)	Phone Position (°)
1	0	15	14.73	- 0.27	15
2			15.34	+ 0.34	15
3			16.06	+ 1.06	16
4			16.24	+ 1.24	16
5			14.91	- 0.09	15
1	5		14.56	- 0.44	15
2			16.19	+ 1.19	16
3			17.54	+ 2.54	18
4			15.78	+ 0.78	16
5			16.62	+ 1.62	17
1	10		16.57	+ 1.57	17
2			17.49	+ 2.49	17
3			14.23	- 0.77	14
4			16.94	+ 1.94	17
5			18.38	+ 3.38	18
1	12.5		16.61	+ 1.61	17
2			18.37	+ 3.37	18
3			14.79	- 0.21	15
4			17.92	+ 2.92	18
5			16.24	+ 1.24	16

6.3.2. Satellite Tracking Test Data

Table 6-8 lists the tracking values for the selected satellites. Due to the small increments at which a satellite moves, the decision was made to record results at 1-minute intervals. The positions for Meteor M2 were plotted in Figure 4-9, NOAA 18 in Figure 4-10, NOAA 19 in Figure 4-11, and the ISS in Figure 4-12.

Table 6-8: Satellite Tracking Results

Azimuth (°)	Elevation (°)	Azimuth (°)	Elevation (°)
Expected		Actual	
Meteor M2			
174.81	0.00	172.62	-1.09
174.99	0.50	173.25	-1.09
176.50	4.30	177.87	5.68
178.58	8.69	179.41	5.68
181.54	13.95	182.94	15.78
186.03	20.51	189.10	19.47
193.46	28.97	195.68	25.78
207.18	39.61	206.49	45.06
234.21	49.61	238.64	46.21
273.02	49.66	271.23	46.22
300.32	39.63	304.34	42.74
314.21	28.86	310.64	28.34
321.76	20.31	320.91	22.67
326.35	13.69	325.67	14.85
329.42	8.40	329.10	9.78
331.61	3.98	334.57	4.14
333.26	0.16	334.58	-0.94

NOAA 19			
Expected		Actual	
Azimuth (°)	Elevation (°)	Azimuth (°)	Elevation (°)
189.01	0.03	186.41	0.34
193.67	3.05	191.11	5.74
199.43	6.18	196.74	7.45
206.60	9.34	204.32	10.37
215.47	12.34	214.98	10.38
226.18	14.82	227.69	10.38
238.45	16.31	240.87	15.87
251.42	16.42	254.46	15.85
263.88	15.10	265.24	15.82
274.90	12.72	279.18	13.35
284.10	9.77	287.78	10.97
291.58	6.61	294.35	5.51
297.62	3.46	296.78	2.68
302.52	0.43	303.97	-0.87
NOAA 18			
Expected		Actual	
Azimuth (°)	Elevation (°)	Azimuth (°)	Elevation (°)
146.43	0.03	150.74	0.81
141.88	3.08	139.08	2.98
136.31	6.31	136.87	5.67
129.39	9.66	125.46	5.69
120.73	13.01	118.19	14.74
110.02	16.02	111.34	14.89
97.28	18.17	95.75	19.12
83.25	18.85	84.42	19.12
69.32	17.84	68.07	18.64
56.83	15.44	55.77	15.04
46.43	12.26	47.16	13.34
38.07	8.82	35.41	10.62
31.42	5.41	30.97	6.71
26.10	2.15	27.73	1.09
24.04	0.74	24.78	0.71

ISS			
Expected		Actual	
Azimuth (°)	Elevation (°)	Azimuth (°)	Elevation (°)
358.40	0.02	355.34	-0.14
5.60	3.24	4.05	5.78
15.47	6.61	16.41	5.84
28.92	9.79	30.97	12.66
46.07	11.87	48.17	12.67
64.84	11.84	62.77	12.66
81.75	9.73	83.91	10.13
94.91	6.58	95.05	7.41
104.52	3.24	101.58	4.37
111.51	0.06	112.35	-0.64

6.4. Appendix D – Code

The algorithms and code that are referenced and used in the report are listed below.

6.4.1. Doppler Shift Algorithm

The following algorithm determines and angle of arrival using Doppler frequency for satellites in low Earth orbit (Islam, et al., 2020).

```
Input:  $X$ : IQ signal of length  $t$  seconds,  $F_s$ : sample rate,  
 $f_c$ : carrier frequency in Hz.  
Output:  $f_d$ : Doppler frequency  
Initialization:  $j = 1, k = 0, f_o$ : Observed frequency = 0.  
1: for  $i = 1$  to  $t$  do  
2:    $Y = \text{FFT}(X(i))$   
3:    $N_p = \frac{1}{8 \times 10^3} \left( \sum_{q=f_c-4 \times 10^3}^{f_c+4 \times 10^3} Y(q) \right)$   
4:   Find  $f_m$  at  $\max\{Y(q)\}$   
5:    $S_p = \frac{1}{400} \left( \sum_{q=f_m-200}^{f_m+200} Y(q) \right)$   
6:   if  $S_p/N_p > 1.2$  then  
7:     if  $j = 1$  then  
8:        $f_o(j) = f_m$   
9:     else if  $j > 1$  and  $k > 10$  and  $f_o(j-1) > f_o(j)$  then  
10:       $f_o(j) = f_m; k = 0$   
11:    else  
12:      Update  $f_m$  to at  $\max\{Y(q); q = \{f_o(j-1)-200:$   
       $f_o(j-1)+200\}\}$   
13:       $f_o(j) = f_m$   
14:    end if  
15:     $f_d(i) = f_o(j) - f_c$   
16:     $j = j+1$   
17:  else  
18:     $f_d(i) = \text{NaN}$   
19:     $k = k+1$   
20:  end if  
21: end for
```

6.4.2. TrackGUI Code

```
#!/usr/bin/env python3

from guizero import App, Text, Picture, PushButton, Window,
TextBox

import sys

import os

import subprocess

import threading

def open1():

    print("image tracking")

    exec(open("/home/pi/Desktop/Image tracking/Image
tracking/trackingCamera.py").read())

def open2():

    print("gpredict")

    subprocess.Popen(["\xterminal", "-e", "rotctld -m 202 -r
/dev/ttyACM0 -s 9600 -C timeout=500 -vvv"])

    subprocess.Popen(["gqrx"])

    subprocess.run(["gpredict"])

def open3():

    print("SSTV")
```

```

        subprocess.run(["qsstv"])

def open4():
    print("gps")
    subprocess.Popen(["\xterminal", "-e", "sudo gpsmon
/dev/serial0"])

def open5():
    print("radio tuner")
    subprocess.run(["gqrx"])

def open7():
    print("edit users")
    app.warn("Edit Users", "Add each user on a new line,
separate password with a colon(:)")
    subprocess.call(["\xterminal", "-e", "sudo nano
/home/pi/Desktop/TrackGUI/.users.txt"])

def open8():
    print("NOAA")
    subprocess.run(["wxtoimg"])

def open9():
    print("FRQ")

```

```
frqwindow.show()
```

```
def open10():
```

```
    frqwindow.warn("Input Commands","Perform the terminal steps  
mentioned in the lauched window")
```

```
    subprocess.Popen(["\xterminal","-e","sudo nano  
/home/pi/Desktop/TrackGUI/FRQ1"])
```

```
    subprocess.run(["\xterminal","-e","\xterminal"])
```

```
def open11():
```

```
    frqwindow.warn("Input Commands","Perform the terminal steps  
mentioned in the lauched window")
```

```
    subprocess.Popen(["\xterminal","-e","sudo nano  
/home/pi/Desktop/TrackGUI/FRQ2"])
```

```
    subprocess.run(["\xterminal","-e","\xterminal"])
```

```
def open12():
```

```
    frqwindow.warn("Input Commands","Perform the terminal steps  
mentioned in the lauched window")
```

```
    subprocess.Popen(["\xterminal","-e","sudo nano  
/home/pi/Desktop/TrackGUI/FRQ3"])
```

```
    subprocess.run(["\xterminal","-e","\xterminal"])
```

```
def verifylogin(username_box, password_box):
```

```
    with open("/home/pi/Desktop/TrackGUI/.users.txt", "r") as f:
```

```
        for line in f:
```

```

        loginInfo = line.strip().split(":")

        if username_box.value == loginInfo[0] and
password_box.value == loginInfo[1]:

            if username_box.value == "admin":

                app.show()

                button7.show()

                window.destroy()

                app.info("welcome", "Admin Level")

                break

            else:

                app.show()

                window.destroy()

                app.info("welcome", "User Level")

                break

        else:

            app.warn("Incorrect Details", "Try Again")

def exitApp():

    app.destroy()

    sys.exit()

app = App(layout="grid",title="TrackGUI", height=650, width=700)

app.bg = "black"

app.text_color = "white"

```

```

window = window(app, title = "TrackGUI Login", height=175,
width=325, layout="grid")

window.bg = "black"

frqwindow = window(app, title = "Frequency ID", height = 200,
width = 500, layout = "grid")

app.hide()

frqwindow.hide()

loginpicture = Picture(window,
image="/home/pi/Desktop/TrackGUI/mainlogo.png",width=50, height
=50, grid=[0,0])

login_message = Text(window, text="TrackGUI Login", size=11,
grid=[1,0])

wline1 = Text(window, text="-----", grid=[0,3])
wline2 = Text(window, text="-----", grid=[1,3])
wline3 = Text(window, text="-----", grid=[2,3])

wline1.text_color = "deeppink3"
wline2.text_color = "deepskyblue2"
wline3.text_color = "medium purple"

window.show()

user = Text(window,text="Username", grid=[0,4])

```

```

username_box = TextBox(window, grid=[1,4])

passw = Text(window,text="Password", grid=[0,5])

password_box = TextBox(window, hide_text=True, grid=[1,5])

loginButton = PushButton(window, text="Login",
command=verifylogin, args=[username_box, password_box],
grid=[1,6])

exitbutton = PushButton(window, text="EXIT", command=exitApp,
grid=[2,6])

picture = Picture(app,
image="/home/pi/Desktop/TrackGUI/mainlogo.png",width=50, height
=50, grid=[1,0])

welcome_message = Text(app, text="Make a Selection", size=15,
grid=[2,0])

app.bg = "black"

app.text_color = "white"

line1 = Text(app, text="-----"
-----", grid=[1,3])

line2 = Text(app, text="-----"
-----", grid=[2,3])

line3 = Text(app, text="-----"
-----", grid=[3,3])

```

```

line1.text_color = "deeppink3"

line2.text_color = "deepskyblue2"

line3.text_color = "purple3"

buttonf1 = PushButton(frqwindow,
image="/home/pi/Desktop/TrackGUI/frqicon.png", command=open10,
width=100, height =100, grid=[1,4])

textf1 = Text(frqwindow, text="1. IQ Data Download", grid=[1,5])

infotextf1 = Text(frqwindow, text="Starts IQ Reception \n",
grid=[1,6])

buttonf2 = PushButton(frqwindow,
image="/home/pi/Desktop/TrackGUI/specicon.png", command=open11,
width=100, height =100, grid=[2,4])

textf2 = Text(frqwindow, text="2. Spectrogram", grid=[2,5])

infotextf2 = Text(frqwindow, text="Views IQ Data \n",
grid=[2,6])

buttonf2 = PushButton(frqwindow,
image="/home/pi/Desktop/TrackGUI/orbiticon.png", command=open12,
width=100, height =100, grid=[3,4])

textf2 = Text(frqwindow, text="3. IQ Analysis", grid=[3,5])

infotextf2 = Text(frqwindow, text="Frequency Analysis \n",
grid=[3,6])

button1 = PushButton(app,
image="/home/pi/Desktop/TrackGUI/tracklogo.png", command=open1,
width=100, height =100, grid=[1,4])

```

```
text1 = Text(app, text="Image Tracking", grid=[1,5])

infotext1 = Text(app, text="Starts Image Tracking \n",
grid=[1,6])

button2 = PushButton(app,
image="/home/pi/Desktop/TrackGUI/gpredictlogo.png",command=open2
, width=100, height =100, grid=[2,4])

text2 = Text(app, text="gPredict", grid=[2,5])

infotext2 = Text(app, text="Starts Tracking \n", grid=[2,6])

button3 = PushButton(app,
image="/home/pi/Desktop/TrackGUI/sstvlogo.png", command=open3,
width=100, height =100, grid=[3,4])

text3 = Text(app, text="SSTV", grid=[3,5])

infotext3 = Text(app, text="Starts SSTV \n", grid=[3,6])

button4 = PushButton(app,
image="/home/pi/Desktop/TrackGUI/gpslogo.png", command=open4,
width=100, height =100, grid=[2,10])

text4 = Text(app, text="GPS", grid=[2,11])

infotext4 = Text(app, text="Starts GPS", grid=[2,12])

button5 = PushButton(app,
image="/home/pi/Desktop/TrackGUI/gqrxlogo.png", command=open5,
width=100, height =100, grid=[2,7])

text5 = Text(app, text="Radio Tuner", grid=[2,8])

infotext5 = Text(app, text="Starts Radio Tuner \n", grid=[2,9])
```

```
button6 = PushButton(app, text="EXIT", command=exitApp, width=5,  
height=2, grid=[3,13])
```

```
button7 = PushButton(app, text="Edit Users", command=open7,  
width=5, height=2, grid=[2,13])
```

```
button7.hide()
```

```
button8 = PushButton(app,  
image="/home/pi/Desktop/TrackGUI/NOAA.png", command=open8,  
width=100, height =100, grid=[3,7])
```

```
text8 = Text(app, text="wxtoimg", grid=[3,8])
```

```
infotext8 = Text(app, text="Starts NOAA decoding \n",  
grid=[3,9])
```

```
button9 = PushButton(app,  
image="/home/pi/Desktop/TrackGUI/frqicon.png", command=open9,  
width=100, height =100, grid=[1,7])
```

```
text9 = Text(app, text="strf", grid=[1,8])
```

```
infotext9 = Text(app, text="Starts frequency decoding \n",  
grid=[1,9])
```

```
nametext = Text(app, text="Rushil Kisoan 2021", grid=[1,13],  
size=10)
```

```
nametext.text_color = "slate gray"
```

```
app.display()
```

7. References

Ahafonau, P., 2021. *Commander Compass Go*. [Online]

Available at: <https://apps.apple.com/us/app/commander-compass-go/id340268949>

[Accessed 15 December 2021].

Airspy, 2021. *Airspy R2*. [Online]

Available at: <https://airspy.com/airspy-r2/>

[Accessed 3 June 2021].

AMSAT, 2021. *TLE/Keplerian Element Resources*. [Online]

Available at: <https://www.amsat.org/keplerian-elements-resources/>

[Accessed 6 June 2021].

Anderson, T., 2019. *Too hot to handle? Raspberry Pi 4 fans left wondering if kit should come with a heatsink*. [Online]

Available at: https://www.theregister.com/2019/07/22/raspberry_pi_4_too_hot_to_handle/

[Accessed 24 December 2021].

Arduino, 2020. *Arduino*. [Online]

Available at: <https://create.arduino.cc>

[Accessed 10 May 2021].

Astronomical Society of Southern Africa, 2022. *Observing satellites*. [Online]

Available at: <https://assa.saao.ac.za/how-to-observe/satellites/>

[Accessed 25 May 2022].

Australian Space Academy, 2021. *RADIO FREQUENCIES FOR SPACE COMMUNICATION*.

[Online]

Available at: <https://www.spaceacademy.net.au/spacelink/radiospace.htm>

[Accessed 18 June 2021].

Bassa, C., 2021. *STRF*. [Online]

Available at: <https://github.com/cbassa/strf>

[Accessed 18 November 2021].

Belden, 2012. *RG Coaxial and Triaxial Reference Guide*. [Online]

Available at:

<https://web.archive.org/web/20120511024343/http://belden.com/resourcecenter/tools/cablefinde>

[r/upload/06-3_15.pdf](#)

[Accessed 5 June 2022].

Bhattacharya, B., 2020. *Slow Scan TV (SSTV) – How it works*. [Online]

Available at: <https://vu2nsb.com/cw-digital-radio/slow-scan-tv-sstv/>

[Accessed 24 July 2021].

Bland, P. et al., 2018. *FireOPAL: Toward a Low-Cost, Global, Coordinated Network of Optical Sensors for SSA*. Maui, Hawaii, The Maui Economic Development Board.

Boller & Chivens, 2011. *Baker Nunn Satellite Tracking Cameras*. [Online]

Available at: <http://bollerandchivens.com/?p=561>

[Accessed 22 May 2022].

Brandenberger, A. J., 1962. The Use of Baker-Nunn Cameras for Tracking of Artificial Earth Satellites. *Photogrammetric Engineering & Remote Sensing*, 28(5), pp. 727-735.

Celestis, Inc., 2021. *What are the "azimuth and elevation" of a satellite?*. [Online]

Available at: <https://www.celestis.com/resources/faq/what-are-the-azimuth-and-elevation-of-a-satellite/>

[Accessed 29 March 2021].

CelesTrak, 2019. *NORAD Two-Line Element Set Format*. [Online]

Available at: <https://www.celestrak.com/NORAD/documentation/tle-fmt.php>

[Accessed 25 May 2021].

CelesTrak, 2022. *Satellite Catalog (SATCAT)*. [Online]

Available at: <https://celestrak.com/satcat/search.php>

[Accessed 25 May 2022].

Chandler, S., 2021. *How to Calibrate iPhone Motion Sensors and Compass*. [Online]

Available at: <https://www.lifewire.com/how-to-calibrate-an-iphone-4172146>

[Accessed 2 June 2022].

cmglee, 2021. *Comparison satellite navigation orbits*. [Online]

Available at: <https://commons.wikimedia.org/w/index.php?curid=16891766>

[Accessed 3 August 2021].

Crowdsupply, 2021. *KrakenSDR*. [Online]

Available at: <https://www.crowdsupply.com/krakenrf/krakensdr>

[Accessed 18 May 2021].

Csete, A., 2021. *About Gpredict*. [Online]

Available at: <http://gpredict.oz9aec.net/>

[Accessed 26 March 2021].

Csete, A., 2021. *Gqrx SDR*. [Online]

Available at: <https://gqrx.dk/>

[Accessed 14 May 2021].

Day, L., 2020. *How To Choose The Right GPS Module For Your Project*. [Online]

Available at: <https://hackaday.com/2020/08/14/how-to-choose-the-right-gps-module-for-your-project/>

[Accessed 25 July 2021].

Digi-Key Electronics, 2021. *Scheme-it*. [Online]

Available at: <https://www.digikey.com/schemeit/project>

[Accessed 15 June 2021].

Director Operational Test and Evaluation, 2019. *DOT&E FY 2019 ANNUAL REPORT*, United States: Director Operational Test and Evaluation.

Dunnell, C. A., 1967. TRANET DOPPLER TRACKING SYSTEM. *Johns Hopkins APL Technical Digest*, 6(4), pp. 17-23.

E-Signals, 2020. *How safe are we from falling satellites or space junk?*. [Online]

Available at: <https://esignals.fi/en/category-en/sustainability/how-safe-are-we-from-falling-satellites-or-space-junk/>

[Accessed 19 June 2021].

Espressif Systems, 2021. *ESP32*. [Online]

Available at: <https://www.espressif.com/en/products/socs/esp32>

[Accessed 18 May 2021].

European Space Agency, 2019. *Using lasers to track space debris*. [Online]

Available at:

https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Shaping_the_Future/Us

ing lasers to track space debris

[Accessed 2 July 2021].

European Space Agency, 2020. *Types of orbits*. [Online]

Available at: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits

[Accessed 18 May 2021].

Everberg, J., 2021. *QSSTV*. [Online]

Available at: <http://users.telenet.be/on4qz/index.html>

[Accessed 3 July 2021].

ExoAnalytic Solutions, 2021. *Space Domain Awareness Solutions*. [Online]

Available at: <https://exoanalytic.com/space-domain-awareness/>

[Accessed 25 March 2021].

Ezell, L. N., 1988. SP-4012 NASA HISTORICAL DATA BOOK: VOLUME III. In: N. A. a. S.

Administration, ed. *PROGRAMS AND PROJECTS 1969-1978*. 1 ed. Washington, D.C.:

National Aeronautics and Space Administration, pp. 385-428.

Fritzing, 2021. *Fritzing*. [Online]

Available at: <https://fritzing.org/>

[Accessed 10 June 2021].

Gonzales, J. & Gonzales, J., 2018. *Mini Satellite-Antenna Rotator Mk1*. [Online]

Available at: <https://www.sarcnet.org/rotator-mk1.html>

[Accessed 19 April 2021].

Gorman, A., 2021. *Starlink satellites: SpaceX is becoming the iPhone of low-Earth orbit*.

[Online]

Available at: <https://www.inverse.com/innovation/spacex-starlink-satellites-apple-iphone>

[Accessed 5 June 2021].

Grush, L., 2019. *Why the Air Force still cannot identify more than a dozen satellites from one December launch*. [Online]

Available at: <https://www.theverge.com/2019/4/2/18277344/space-situational-awareness-air-force-tracking-sso-a-spaceflight-cubesats>

[Accessed 17 June 2021].

guizero, 2021. *guizero*. [Online]

Available at: <https://lawsie.github.io/guizero/>

[Accessed 5 May 2021].

Hamlib, 2021. *Ham Radio Control Libraries*. [Online]

Available at: <https://github.com/Hamlib/Hamlib/wiki>

[Accessed 26 August 2021].

International Telecommunication Union, 2020. *Radio Regulations*, Geneva: International Telecommunication Union.

Islam, M. . N., Spitzer, T., Hallett, J. & Asikuzzaman, M., 2020. Doppler Estimation for Passive RF Sensing Method in Space Domain Awareness. *2020 Military Communications and Information Systems Conference (MilCIS)*, pp. 1-4.

Jackson, C., 2021. *EASYCOMM II Standard*. [Online]

Available at: <https://www.mustbeart.com/software/easycomm.txt>

[Accessed 27 May 2021].

Jouade, A. a. B. A., 2019. Massively Parallel Implementation of FETI-2LM Methods for the Simulation of the Sparse Receiving Array Evolution of the GRAVES Radar System for Space Surveillance and Tracking. *IEEE Access*, Volume PP, pp. 1-1.

Jouadé, A. & Barka, A., 2019. Massively Parallel Implementation of FETI-2LM Methods for the Simulation of the Sparse Receiving Array Evolution of the GRAVES Radar System for Space Surveillance and Tracking. *IEEE Access*, 7(1), pp. 128968-128979.

Kelso, T. S., 2021. *How does one find source code for the SGP4/SDP4 orbital models?*.

[Online]

Available at: <https://celestrak.com/columns/v04n05/#FAQ11>

[Accessed 12 July 2021].

Kessler, D. J., 1991. Collisional Cascading: The Limits of Population Growth in Low Earth Orbit. *Advances in Space Research*, 11(12), pp. 63-66.

Kessler, D. J. & Cour-Palais, B. G., 1978. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6), pp. 2637-2646.

- Kovo, Y., 2020. *Identification and Tracking Systems*. [Online]
Available at: <https://www.nasa.gov/smallsat-institute/sst-soa-2020/identification-and-tracking-systems>
[Accessed 5 May 2021].
- Kulu, E., 2022. *Nanosats Database*. [Online]
Available at: <https://www.nanosats.eu/>
[Accessed 22 May 2022].
- Lockheed Martin Corporation, 2021. *Space Fence*. [Online]
Available at: <https://www.lockheedmartin.com/en-us/products/space-fence.html>
[Accessed 15 May 2021].
- M. Laakso, R. R. R. W. a. V. K., 2021. Phase-coherent multichannel SDR - Sparse array beamforming,. *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 1856-1860.
- Magliacane, J., 2021. *PREDICT*. [Online]
Available at: <https://www.qsl.net/kd2bd/predict.html>
[Accessed 24 May 2021].
- Modular Assembly Technology SA, 2022. *STRUT PROFILE PG30*. [Online]
Available at: <https://moduasm.co.za/product/strut-profile-pg30-30x30-2-slots-type-b/>
[Accessed 09 January 2022].
- N2YO, 2021. *SEARCH SATELLITE DATABASE*. [Online]
Available at: <https://www.n2yo.com/database/>
[Accessed 31 December 2021].
- NASA, 2012. *Tracking and Data Relay Satellite (TDRS) First Generation Capabilities*. [Online]
Available at: https://www.nasa.gov/directorates/heo/scan/services/networks/tdrs_first_gen
[Accessed 22 May 2022].
- NASA, 2017. *July 1961 - Mercury Space Flight Network (MSFN) Completed*. [Online]
Available at: https://www.nasa.gov/directorates/heo/scan/images/history/July1961_2.html
[Accessed 22 May 2022].

NASA, 2021. *Tracking and Data Relay Satellite (TDRS)*. [Online]
Available at: https://www.nasa.gov/directorates/heo/scan/services/networks/tdrs_main
[Accessed 22 May 2022].

National Research Council, 1995. *Orbital Debris: A Technical Assessment*. 1 ed. Washington, DC: The National Academies Press.

Nyansat, 2020. *Nyansat*. [Online]
Available at: <https://nyan-sat.com/>
[Accessed 24 March 2021].

PJRC, 2022. *Teensy® 4.0 Development Board*. [Online]
Available at: <https://www.pjrc.com/store/teensy40.html>
[Accessed 6 January 2022].

poilkjewq, 2018. *How to create GUI in C++*. [Online]
Available at: <https://forums.raspberrypi.com/viewtopic.php?t=217975>
[Accessed 28 May 2021].

Pololu, 2021. *LSM303DLHC 3D Compass and Accelerometer Carrier with Voltage Regulator*. [Online]
Available at: <https://www.pololu.com/product/2124>
[Accessed 24 December 2021].

Pun, S., 2021. How Radar Technology Changed the Course of the World after World War II - Science and Technology. *Unity Journal*, Volume 2, pp. 243-250.

Radxa Limited, 2019. *Introduce ROCK Pi 4*. [Online]
Available at: <http://rockpi.org/rockpi4>
[Accessed 1 July 2021].

Raspberry Pi Foundation, 2021. *Raspberry Pi 4 Tech Specs*. [Online]
Available at: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>
[Accessed 25 May 2021].

Raspberry Pi Foundation, 2021. *Raspberry Pi OS*. [Online]
Available at: <https://www.raspberrypi.org/software/>
[Accessed 17 May 2021].

RealVNC, 2021. *VNC® Connect is included with Raspberry Pi OS*. [Online]

Available at: <https://www.realvnc.com/en/raspberrypi/>

[Accessed 14 May 2021].

Red Balloon Security, 2020. *Red Balloon Security and NyanSat DEF CON 28 Talk 2020*.

[Online]

Available at: <https://youtu.be/ga44gGe6iok>

[Accessed 28 May 2022].

Reza Chalak Qazani, M. et al., 2021. A Real-Time Motion Control Tracking Mechanism for Satellite Tracking Antenna Using Serial Robot. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1049-1055.

RF Wireless World, 2012. *GPS Sentences / NMEA Sentences /*. [Online]

Available at: <https://www.rfwireless-world.com/Terminology/GPS-sentences-or-NMEA-sentences.html>

[Accessed 09 July 2021].

Roberts, G., 2007. How To ... Visually Observe Satellites. *Monthly Notes of the Astronomical Society of Southern Africa*, 66(9 & 10), pp. 205-209.

Robotistan, 2022. *Pro Micro Clone 5V 16 Mhz For Arduino*. [Online]

Available at: <https://www.robotistan.com/pro-micro-clone-5v-16-mhz-for-arduino>

[Accessed 06 January 2022].

Rouma, A., 2021. *SDRPlusPlus*. [Online]

Available at: <https://github.com/AlexandreRouma/SDRPlusPlus>

[Accessed 23 May 2021].

RS Components (SA), 2022. *The Complete Guide to Stepper Motors*. [Online]

Available at: <https://za.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/stepper-motors-guide>

[Accessed 09 January 2022].

RTL-SDR, 2017. *Using the GRAVES Radar to Listen to Reflections from Meteors, Planes and Spacecraft*. [Online]

Available at: <https://www.rtl-sdr.com/using-the-graves-radar-to-listen-to-reflections-from-meteors-planes-and-spacecraft/>

[Accessed 16 July 2021].

RTL-SDR, 2021. *About RTL-SDR*. [Online]

Available at: <https://www.rtl-sdr.com/about-rtl-sdr/>

[Accessed 28 May 2021].

RTL-SDR, 2021. *Sparse Array Beamforming with a Phase Coherent 21-Channel RTL-SDR Array*. [Online]

Available at: <https://www.rtl-sdr.com/tag/direction-finding/>

[Accessed 27 April 2021].

Sanghvi, Y., 2021. *Arduino Uno vs Teensy*. [Online]

Available at: <https://www.tutorialspoint.com/arduino-uno-vs-teensy>

[Accessed 10 August 2021].

SatNOGS, 2020. *SatNOGS Rotator v3*. [Online]

Available at: https://wiki.satnogs.org/SatNOGS_Rotator_v3

[Accessed 23 March 2021].

SATRAN, 2021. *SATRAN Antenna Rotator*. [Online]

Available at: <https://www.satran.io/>

[Accessed 27 June 2021].

Schmidt, J. R., 2021. *GPS Coordinates: How Many Decimal Places do You Need?*. [Online]

Available at: <https://www.forensicdjs.com/blog/gps-coordinates-many-decimal-places-need/>

[Accessed 30 December 2021].

Sean McManus, M. C., 2021. *Top 10 Programming Languages Ported to the Raspberry Pi*.

[Online]

Available at: <https://www.dummies.com/computers/raspberry-pi/top-10-programming-languages-ported-to-the-raspberry-pi/>

[Accessed 20 May 2021].

Secure World Foundation, 2019. *US Policy and Capabilities on SSA*. [Online]

Available at: <https://swfound.org/media/206348/weeden-us-policy-and-capabilities-for-ssa.pdf>

[Accessed 24 May 2022].

Shimkus, K., 2020. *USSF announces initial operational capability and operational acceptance of Space Fence*. [Online]

Available at: <https://www.spaceforce.mil/News/Article/2129325/ussf-announces-initial->

operational-capability-and-operational-acceptance-of-spa/

[Accessed 15 May 2021].

Sinovoip, 2018. *BPI-M5*. [Online]

Available at: <http://www.banana-pi.org/m5.html>

[Accessed 1 July 2021].

Space Foundation, 2021. *Space Briefing Book*. [Online]

Available at: https://www.spacefoundation.org/space_brief/types-of-orbits/

[Accessed 23 May 2021].

Space-Track, 2022. *Space-track.org - Frequently Asked Questions (FAQ)*. [Online]

Available at: <https://www.space-track.org/documentation#/faq>

[Accessed 23 May 2022].

SparkFun Electronics, 2021. *Accelerometer, Gyro and IMU Buying Guide*. [Online]

Available at: https://www.sparkfun.com/pages/accel_gyro_guide

[Accessed 28 May 2021].

Terry, S., 2019. *Using a wiper motor in your Halloween projects*. [Online]

Available at: <http://www.scary-terry.com/wipmtr/wipmtr2.htm>

[Accessed 20 April 2021].

The National Association for Amateur Radio, 2021. *SSTV Transmissions Scheduled from ISS*.

[Online]

Available at: <https://www.arrl.org/news/sstv-transmissions-scheduled-from-iss>

[Accessed 18 September 2021].

The New University, 2022. *The Billionaire Space Race: Should We Be Worried?*. [Online]

Available at: <https://www.newuniversity.org/2022/05/03/the-billionaire-space-race-should-we-be-worried/>

[Accessed 26 May 2022].

TouchDown Apps, 2021. *Satellite Pointer - Dish Setup*. [Online]

Available at: <https://apps.apple.com/us/app/satellite-pointer-dish-setup/id994565490>

[Accessed 15 December 2021].

Tunca, O., 2021. *WaveVisual*. [Online]

Available at: <https://wavevisual.com/>

[Accessed 28 December 2021].

Tysonpower, 2017. *Building a DIY AZ/EL Antenna Rotator for Satellite Tracking*. [Online]

Available at: <https://www.rtl-sdr.com/building-a-diy-azel-antenna-rotator-for-satellite-tracking/>

[Accessed 26 May 2022].

Union of Concerned Scientists, 2021. *UCS Satellite Database*. [Online]

Available at: <https://www.ucsusa.org/resources/satellite-database>

[Accessed 4 June 2021].

United Nations, 2010. *Space Debris Mitigation Guidelines of the Committee*. Vienna, United Nations.

University of Cape Town, 2022. *Space in South Africa*. [Online]

Available at: <http://www.spacelab.uct.ac.za/space-south-africa-0>

[Accessed 26 May 2022].

University, RWTH Aachen, 2021. *phyphox*. [Online]

Available at: <https://apps.apple.com/us/app/phyphox/id1127319693>

[Accessed 15 December 2021].

VINCI Energies, 2017. *GRAVES radar system upgrade programme*. [Online]

Available at: <https://www.theagilityeffect.com/en/article/graves-radar-system-upgrade-programme/>

[Accessed 15 May 2021].

Vitaliy, . K., Marcos, T., Jafar, S. & Erdal, O., 2011. Direction of Arrival Estimation and Localization Using Acoustic Sensor Arrays. *Journal of Sensor Technology*, 1(3), pp. 71-80.

Western Piedmont Community College, 2022. *mechatronics-venn-diagram*. [Online]

Available at: <https://www.wpcc.edu/mechatronics/mechatronics-venn-diagram/>

[Accessed 24 May 2022].

White, D., 2018. *Overview of the Satellite Networked Open Ground Stations (SatNOGS) Project*. Utah, Small Satellite Conference .