University of Auckland Ground Station One

Sean Richards

Department of Physics
The University of Auckland

Supervisor: Dr Nicholas Rattenbury

A dissertation submitted in partial fulfilment of the requirements for the degree of PGDipSci in Physics, The University of Auckland, 2019.

Abstract

The increasing number of private companies launching satellite-bearing rockets into space, combined with the decreasing cost of development has introduced the ability for individuals, private organisations and research institutes to launch small satellites into orbit. In order to properly facilitate operation, one must be able to both communicate to and hear from the satellite during its orbit, via a ground station operation. We discuss the current state of the field in terms of terrestrial ground stations, ways that we may integrate off-the-shelf components to an existing ground station, and present the methodology for student operators to informally monitor satellites, in the context of the upcoming The Auckland Programme for Space Systems (APSS) satellites, in particular APSS-1 as an example. We give a detailed overview of how the University of Auckland ground station works from both a hardware and software point of view, show the challenges faced, and discuss the current status of the ground station, providing recommendations for the next operators in terms of finishing the in-progress upgrade.

Acknowledgements

Ehara taku toa, he takitahi, he toa takitini.
My success should not be bestowed onto me alone, as it was not individual success but success of a collective.

Māori Whakataukī

I would like to thank Dr Nicholas Rattenbury and Dr John Cater for their invaluable help in reassembling the Ground Station, even when all seemed broken.

Mr Francis Moynihan Lavey, whose experience both in building APSS-1 and his own personal Ground Station proved immensely helpful in designing and carrying out the building of the new rotator circuitry.

Mr Dimitrios Tsifakis of the Australian National University – for his immense help in learning the ropes of satellite tracking, and for being a willing outside influence to listen to ideas.

And my mother – for listening to my endless rambles on the project, for being a sounding board when everything was going wrong, and for never letting me lose sight of why I chose to study physics.

Contents

List of Figures

List of Tables

Glossary

Notation	Description
AM	Amplitude Modulation.
AOS	Acquisition of Signal.
APSS	The Auckland Programme for Space Systems.
DTA	Defence Technology Agency.
FM	Frequency Modulation.
LEO	Low-Earth Orbit.
LOS	Loss of Signal.
RAAN	Right Ascention of the Ascending Node.
SDR	Software Defined Radio.
TLE	Two Line Element set.
UHF	Ultra-High Frequency.
VHF	Very-High Frequency.

1

Introduction & Background

We choose to go to the Moon! We choose to go to the Moon... We choose to go to the Moon in this decade and do the other things, not because they are easy, but because they are hard; because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one we intend to win, and the others, too.

John F. Kennedy

Ever since the dawn of the Space Age, both space missions and satellite communications have become simpler and more accessible. What was once the realm of Governments and Space Agencies is increasingly more accessible to private individuals and organisations. In 2016, the University of Auckland established The Auckland Programme for Space Systems (APSS), aiming to produce CubeSats (Loff 2015), small 1U satellites which carry student missions into Low-Earth Orbit (LEO). Naturally, in order to receive data from, and transmit to, these satellites, a communications scheme must be employed, of which the ground station is a critical piece of technology.

In 2019, the University launched Te Pūnaha Ātea – Auckland Space Institute to facilitate research in Aerospace Engineering and Science, with a view towards improving the ground station to be able to informally monitor the satellites launched under the APSS. This ground station is located on the roof of building 303, belonging to the Faculty of Science, and is physically accessible to those working on it. However, practical requirements necessitate that the control computer should be accessible via remote control, originally realised through the Windows "Remote Desktop Connection" software. The major upgrade of the ground station – the core of my project – involved a shift of Operating Systems from Windows 10 to Ubuntu 19.04, at the time of upgrade, the most recently released version of Ubuntu.

1.1 Scope of the project

The aim of the project was twofold:

(1) To gain familiarity with the process of satellite tracking, and

(2) To complete a major upgrade of the ground station system.

Figure 1.1: The ground station, as viewed from the roof platform. The antenna is pointing roughly west.

The first point links closely with APSS; in developing the methodology for students to be able to track satellites, we are able to train future operators to be able to informally monitor APSS missions such as *APSS-2* and *APSS-3*, notwithstanding the fact that actual satellite communications with the APSS-satellites will be completed externally to the University, using the resources of the Defence Technology Agency (DTA).

1.2 Theoretical Background

1.2.1 Data Modulation

All satellites must employ a communications scheme onboard the craft to modulate a signal; without such a scheme, the satellite will not be able to transmit nor receive telemetry information. The quintessential examples are Frequency Modulation (FM) and Amplitude Modulation (AM) which underpin traditional terrestrial radio – and even optical – signals. Of course, any given satellite may employ any one of these or other schemes, though by convention the modulation schema of at least the beacon is published on the satellite's webpage.

The electric field component of the electromagnetic wave is

$$A(1 + \Delta_{\mathrm{a}} m(t)) \cos(\omega_{\mathrm{c}} t + \phi) \qquad \text{for AM} \tag{1.1}$$

$$A \cos\left(\omega_{\mathrm{c}} t + 2\pi\Delta_{\mathrm{f}} \int m(t)\,\mathrm{d}t + \phi\right) \qquad \text{for FM} \tag{1.2}$$

Where $m(t)$ denotes the arbitrary information wave and Δ is the *degree of modulation* coefficient (informally, a weight to assign the modulation). The specific form of the information wave $m(t)$ depends again on the satellite, however it is most commonly a sequence of square waves representing an "on bit" (when the wave is high) and an "off bit" (when the wave is low). When $m(t)$ is digital, the process is renamed *Amplitude/Frequency Shift Keying* (ASK/FSK for short). In this way we may encode a binary string into the carrier wave, which may then be demodulated at the ground station. The process of Frequency and Amplitude Modulation for realistic digital signals (reading 10101... each) is shown in Fig. 1.2.

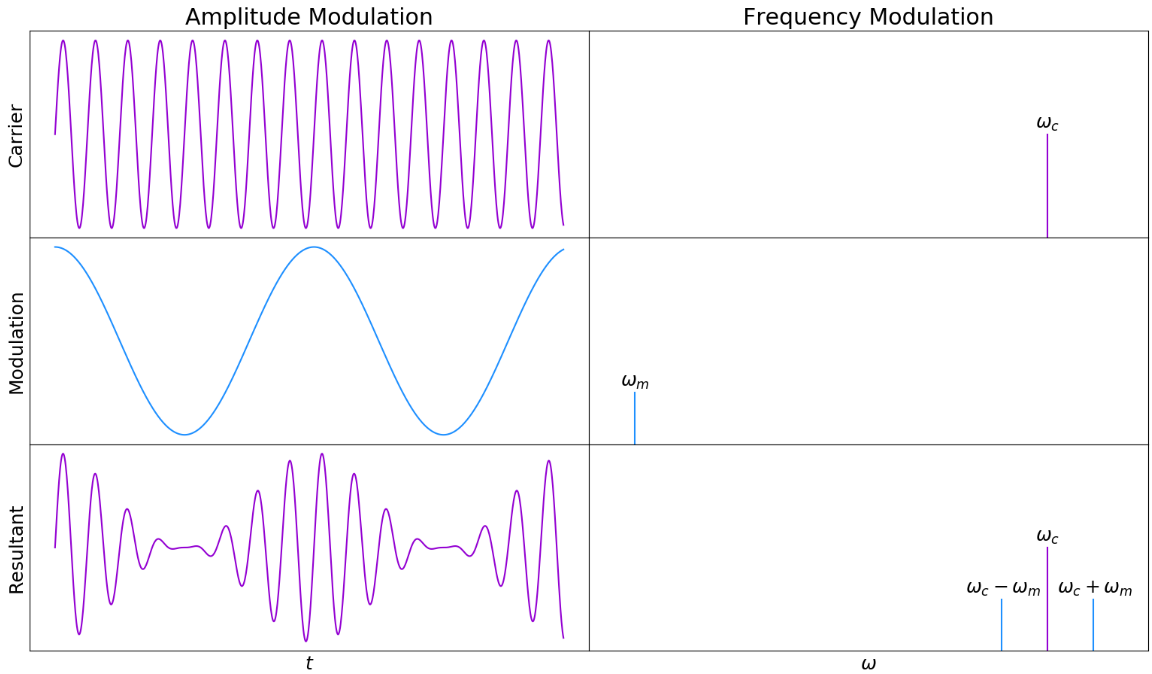


Figure 1.2: Amplitude and Frequency Shift Keying for a simple binary signal.

Ideally a data signal would be a sequence of square waves, with the signal going high for a 1 bit and low for a 0 bit. However in practice the signal has some *rise time* t_{rise} to effectively change between off and on (strictly, between 10% of the signal and 90% of the signal – see Fig. 1.3), which has been exaggerated in the Amplitude Modulation example in Fig. 1.2 for clarity. Conventionally, we denote the length of one bit as t_{B}. Then the rise time of the signal (as a function of t_{B}) is bounded from above by

$$t_{\text{rise}} < \begin{cases} 0.7 t_{\mathrm{B}}, & \text{if the signal does not return to zero between successive on bits} \\ 0.35 t_{\mathrm{B}}, & \text{if the signal does return to zero between successive on bits} \end{cases} \tag{1.3}$$

If the rise time is lower than this, we would observe bits "bleeding into" one another, and we would not be able to distinguish successive bits reliably.

This idealisation does not support a multiplexed signal, as in the case of an antenna array consisting of a Ultra-High Frequency (UHF) and Very-High Frequency (VHF) antenna system. The current ground

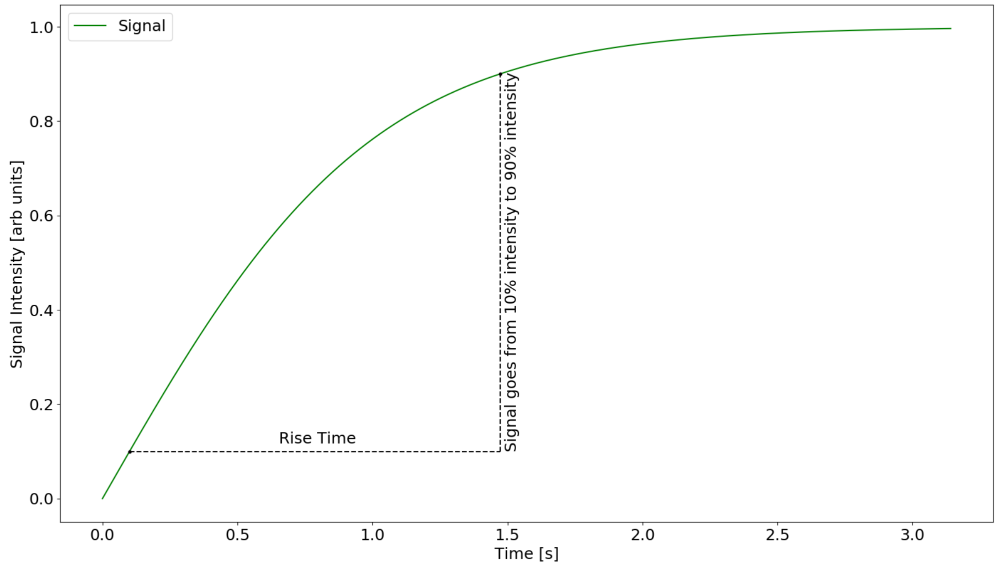


Figure 1.3: The rise time of a hypothetical signal. The signal starts at 0% intensity and climbs through to 100% intensity. The rise time is the time between 10% intensity, and 90% intensity.

station does not support VHF signals, however this is the next major step of the upgrade project, which will take place over the summer of 2019/2020 (see Section 6).

1.2.2 Data Encoding

The most common encoding schematic for data among amateur radio enthusiasts is the *AX.25 packet format* (Beech, Nielsen & Taylor 1998). Indeed, the APSS satellites use the AX.25 format. This format consists of 7 fields, each an integer number of bytes long. A sample AX.25 packet is shown in Table 1.1.

Flag	Address	Control	PID	Information	FCS	Flag
01111110	14/70 bytes	1 byte	1 byte	N bytes	2 bytes	01111110

Table 1.1: Sample AX.25 packet frame.

What follows is a superficial discussion of each field. For the complete technical specification of the AX.25 format, see (Beech et al. 1998).

(1) A *flag* to indicate the beginning of a packet.

(2) An *address* field, which contains the callsign of the transmitter and receiver. The callsign is a six-character alphanumeric string uniquely identifying an individual. There are two subfields of this field, each with six bytes for a callsign (callsigns shorter than six characters have ASCII spaces to lengthen them), and one byte for a Secondary Station Identifier or SSID (in case an operator uses more than one station with the same callsign).

(3) A *control* field identifying the type of frame being packet being sent (information, supervisory, or unnumbered).

(4) A *protocol identifier* field, identifying which kind of layer 3 (the network layer in the Open Systems Interconnection model in networking) encoding is being used, if any.

(5) An *information field* which contains a binary encoding of the data we wish to transmit (for instance commands to the satellite, or science tasks).

(6) A *frame checksum* field, which is computed using the CRC-CCITT polynomial (discussed in (Beech et al. 1998)).

(7) Another *flag* field indicating the beginning of a packet.

Notice that the *flag* field is the binary string 01111110, which corresponds to the ASCII "tilde" character ~. This character, whilst uncommon, may appear in the Information field. This is not desirable as a packet receiver will interpret the binary representation of this as the end of the packet. To compensate for this, the process of *bit stuffing* is employed, where after a string of five consecutive 1 bits, a 0 bit is artificially inserted. Then, when a station receives a signal, when a sequence of five 1 bits is received, any 0 bit that appears immediately afterwards is removed, preserving the message integrity.

2

Literature Review

Most research ground stations are oriented towards support for a multi-mission ground station, capable of full management of satellite activity. Canonical examples include the Virginia Tech ground station (Hitefield, Leffke, Fowler & McGwier 2016), which is an "almost-AMSAT" station, insofar as the components that form the station – with the notable exception of the parabolic dishes in their infrastructure – are available from any reputable HAM Radio outlet. Similarly, the Vienna Institute of Technology station (Fischer & Scholtz 2010) was built to support the (then) future missions of their institute.

An overarching theme that exists across papers in the field is that of separation of abilities. Virginia Tech (Hitefield et al. 2016) for example partitions their ground station into four colour-coded components; *Red* for the untrusted interface, *green* for the trusted local network, *blue* for an additional trusted segment, and *orange* for the "demilitarized zone". The effect of such a network setup is the idea that no one person, outside of global system administrators, has full control of the station, and each only has access to the functions required to complete their tasks. This reduces the potential damage any one person can do, and improves the resilience of the system to attack.

Multi-mission support, however, is beyond the requirements of the University ground station *at this time*. As discussed briefly in Section 1.1, the actual communication with the APSS satellites will be realised by the communications array at DTA. Therefore, the ground station at the University conforms to an almost unique requirement: whilst it is used for simple research currently, our experience with the station informs the process by which we interact with DTA. That said, we do wish to in future support multiple missions. As pointed out in (Choi, Stevenson & Lightsey 2017), a University wishing to support satellite missions must be flexible and capable of reusing hardware without much modification to the station itself. The trend among institutions in this regard is to use open-source software and off-the-shelf components wherever possible, something we have integrated into our ground station as best we can.

Papers such as (Zufelt & Aarested 2018) and (González, Cabrera & Calderón 2016) explore the architecture and standardisation required in ground stations for CubeSat-based missions. They argue for a single unified technique; in the case of (Zufelt & Aarested 2018), it is for a cloud-based *Internet of Things* approach, whereas in (González et al. 2016) they remark that as academic aerospace research is an emerging field, there does not exist sufficient standards in place to ensure interoperability between different stations. Given that passes occur on average three times per day for a typical LEO satellite, and even then there is typically one "good" pass per day lasting no more than 15 minutes, interoperability between academic institutions is of paramount importance. An institute operating in Oceania would do well to partner with an institute in Europe, for example, to increase the effective number of times per day that the satellite can be heard.

Moreover, the wider amateur radio in space community organises systems to track the vital details of satellites. Groups such as AMSAT in the United States (*AMSAT OSCAR Satellite Status* n.d.) and Mike Rupprecht (DK3WN) in Germany (Rupprecht n.d.) track individual satellites and crowd-source data as to whether or not individual satellites are receiving/transmitting signals. The matrix provided by AMSAT, shown in Fig. 2.1, is of particular use, and something that we anticipate being able to contribute to when the ground station is fully operational. On a daily basis, AMSAT appears to attract around 180-200 individual operators submitting data with remarkably few conflicting reports.

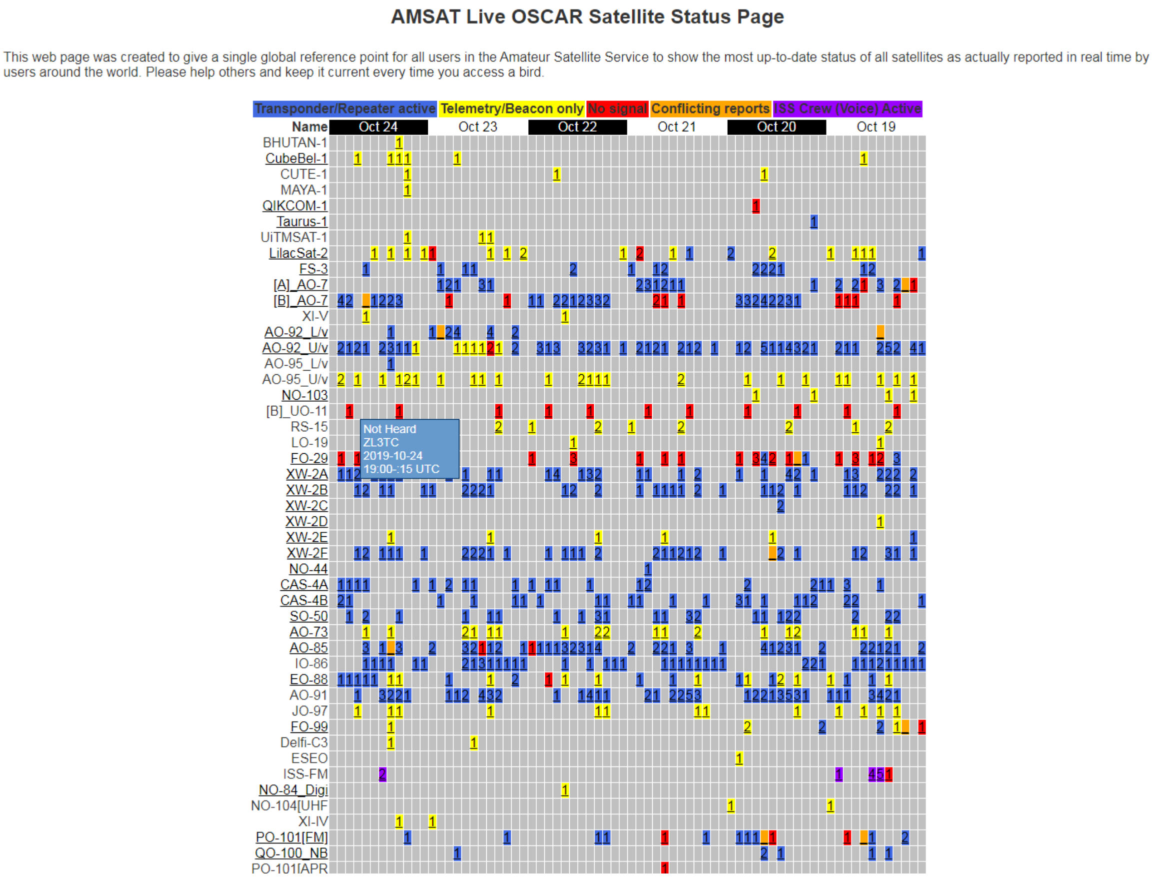


Figure 2.1: The AMSAT tracking matrix. The tracking entries shown here are colour-coded according to the legend at the top, and the number associated with the matrix entry represents the number of passes logged. The tooltip shows that the selected pass (leftmost [B]_UO-11) was tracked by a New Zealand-based operator (from the ZL at the beginning of the callsign ZL3TC), and was not successful. This is a satellite that may be dead, compared to say CAS-4A which is still transmitting.

Conversely, Rupprecht's work appears to focus more on the technical side of tracking – monitoring craft and decoding telemetry. In particular, his work collates information about the data modulation schematic aboard every craft. This is of particular use to us, as it allows us an easy way to determine the right demodulation suite to employ when attempting to decode the telemetry of a given satellite during testing.

3
The Ground Station Overview

3.1 Satellite Tracking

The process of Satellite Tracking can broadly be cast into four main categories:

(1) Pass Determination: Determining the position of a satellite, and when it will pass overhead,

(2) Signal Reception: Receiving a signal via the UHF or VHF antenna,

(3) Signal Interpretation: Interpretation by the SDR viewer (i.e., GQRX or SDR#) to identify signals within the frequency band, and

(4) Signal Demodulation: Demodulation via software like SoundModem or MixW.

The ground station is capable of performing each of these functions, partially with the aid of a human operator. In future, we aim for the ground station to work semi-autonomously, removing the need for a human operator to perform certain repetitive tasks. These are discussed in more detail in Appendix 6.

3.1.1 Pass Determination

As discussed in Appendix A, the motion of satellites is governed by a set of six elements, called the *Keplerian Elements*. Together with information about when they were computed (the so-called *Epoch Time*), one can solve the two-body problem to extrapolate the position of a satellite at any given time, provided that the most recent Keplerian Elements are used.[1] The complete set of Keplerian Elements, along with the Epoch information and satellite metadata, forms the TLE data, which is conventionally paired with the satellite name to give a three-line set of elements. These are typically composed by the United States Air Force, cleaned to remove sensitive military information, and then distributed via services like CelesTrak (Kelso n.d.) to the public. For a brief discussion of the TLE format, see Appendix B

Computing a solution to the two-body problem is beyond the scope of this project – though explored in Section A. The computation is performed by the software GPredict (Csete 2019*a*), which we have configured to update automatically TLE files when it detects they are out of date. A sample GPredict

[1]Keplerian Elements are computed periodically and assume an idealised elliptical orbit – as satellites are susceptible to atmospheric drag and the gravitational field of the Earth is nonuniform, the element set periodically becomes outdated and must be recomputed for accurate predictions.

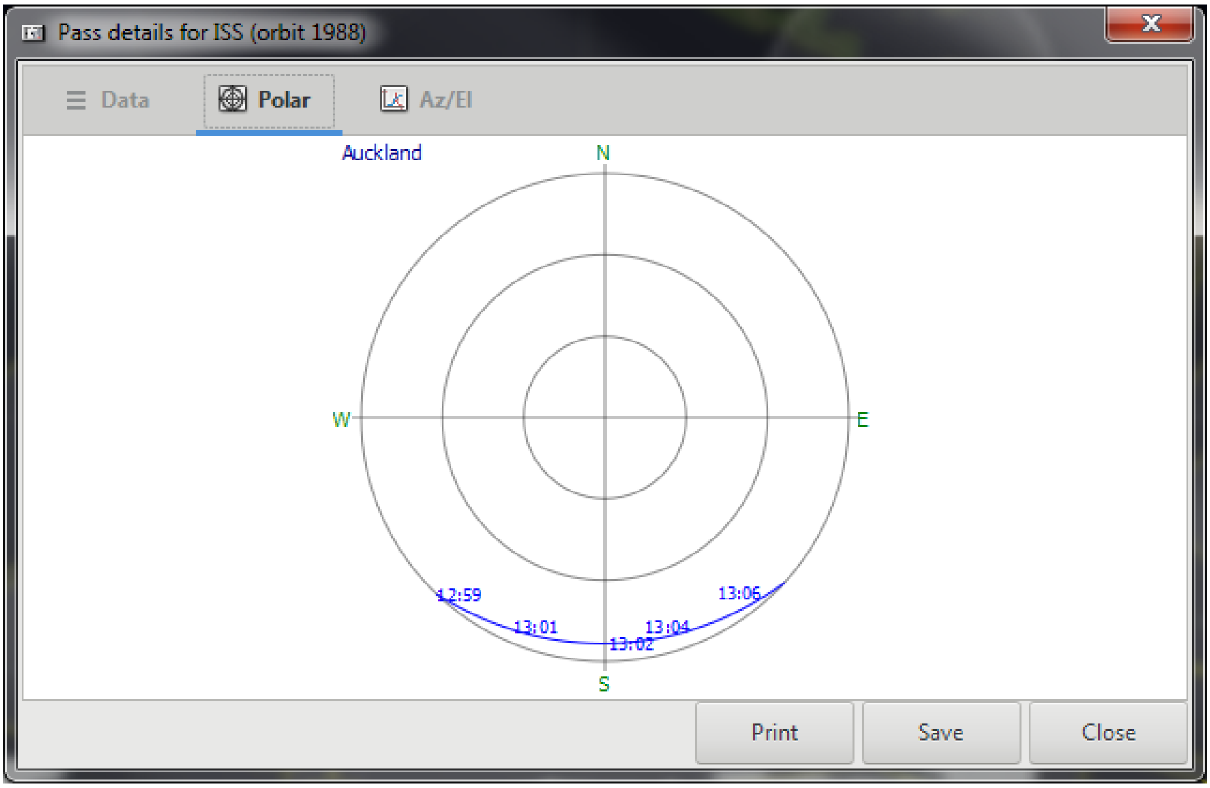


Figure 3.1: A GPredict Prediction for the next pass of the ISS.

prediction is shown in Fig. 3.1. From this, we can see that Acquisition of Signal (AOS) occurs at 12:59 local time, and Loss of Signal (LOS) occurs at 13:06 local time. The pass shown is not suitable for tracking – GPredict computes that the maximum elevation is $\sim 6°$, when the ideal pass would be at zenith. In practice, we ignore passes that are less than 10°, though this tolerance can be relaxed if one is not afraid of signal noise or can compensate for it. The restriction here is the so-called Signal-to-Noise Ratio (SNR), which in ideal conditions is maximised when the satellite barely crosses the horizon, and minimised when the satellite passes through zenith (the origin of the polar plot – corresponding to 90°).

3.1.2 Signal Reception

Whilst it is rather trivial with the aid of software to predict passes by satellites, it is more difficult to receive any signal the satellite may be emitting. Most satellites are launched by spacefaring nations such as the United States and the People's Republic of China. Indeed, APSS-1 is on track to become the first instrumented research satellite launched by a New Zealand institution. There is no benefit to the operators of these stations to transmit to terrestrial stations when they are over New Zealand – any information that they may emit will typically not be heard, given the relative time zone difference between New Zealand and these nations. Therefore, the action of a satellite passing overhead is not necessarily a good indicator that a signal will actually be transmitted.

Moreover satellites have a fixed lifespan. Although some satellites may last for decades, not all do. The CubeSat *FO-29* (JAS 2) launched in 1996 and was pronounced dead by the AMSAT community in 2019, whereas NASA originally defined a CubeSat mission to be successful if it lasted in Space for over 60 days (Tolmasoff & Venturini 2017). Indeed, it can be a small miracle finding a known good satellite, which is why organisations such as *Amateur Radio in Space* in the United States collate information about transmissions received from satellites and publish it on the internet (*AMSAT OSCAR Satellite Status* n.d.). This enables operators to find reliably transmitting satellites that they can test station hardware against.

For satellites that do transmit a signal as they pass overhead in Auckland, the ground station is equipped with a Yagi Y6U UHF antenna (*Hi-Tec Aerials UHF Folded Dipoles & Yagis - Professional Series* n.d.). When circumstances permit, this will be accompanied by a 2MCP14 VHF antenna (*2MCP14, 143-148 MHz* n.d.) to form an antenna array. These antennae capture signals transmitted in their respective frequency bands, and propagate them to an Airspy R2 Software Defined Radio (SDR) (*Airspy R2 - airspy.com* n.d.), which converts the received Radio Frequency (RF) signals into binary data which is processed by the software demodulation suite. The Airspy was chosen at the inception of the ground station as it offers "plug-and-play" functionality – once plugged in to both a computer and an antenna, it can run without any further configuration. For a further example of research applications of the Airspy, see (Snjegota & Rattenbury 2017).

Figure 3.2: The Yagi Y6U UHF antenna currently situated in the ground station. When circumstances permit, this will be complemented by a VHF antenna to form a full antenna array (see Section 6).

3.1.3 Signal Interpretation

Signal interpretation is achieved by using software such as SDR# (Windows-based) (*SDR# and Airspy Downloads - airspy.com* n.d.) or GQRX (Linux based) (Csete 2019*b*). The operator is capable of viewing a range of frequencies at once, with the aim of capturing a signal. The specific form of the signal depends on the satellite — in Fig. 3.3, we are observing the CW beacon of FO-29, which is continuously emitting. This made it an ideal candidate for training, and ground station re-calibration post-upgrade, until it was declared dead by the community.

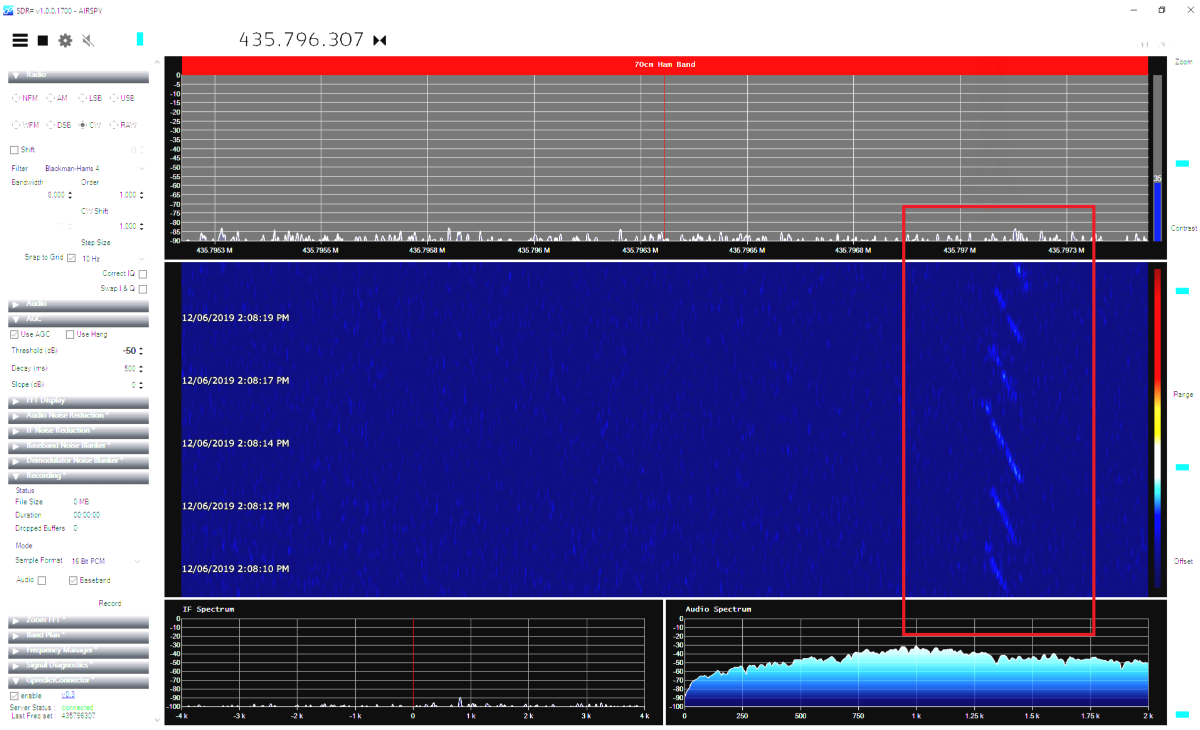


Figure 3.3: A sample signal from the satellite FO-29 (contrast adjusted). The inset within the red box is shown in Fig. 3.4.

The signal interpretation software reconstitutes the input digital signal into a graphical display for the operator. It shows a waterfall plot and a frequency-domain signal spectrogram, enabling easy identification of signals. The output of the interpreter can be piped into signal demodulation software to facilitate the decoding of telemetry of the satellite. In the particular case of this ground station, the signal interpretation software is GQRX (Csete 2019*b*), and the TCP throughput from the physical ground station to the analysis machine is provided by SpyServer (*SDR# and Airspy Downloads - airspy.com* n.d.).

The ground station currently lacks this functionality. Support for GQRX to connect to SpyServer is through a patched version of the gr-osmosdr package (Teske 2019). This is a part of the underlying libaries which allow GQRX to be receiver-agnostic. The package appears to have some defects which mean that the connection between GQRX and SpyServer does not activate, despite SpyServer being ready to transmit data and GQRX being ready to receive it. Rectifying this is the first step in the future development of the ground station, as will be discussed in Section 6.

3.1.4 Signal Demodulation

In principle, signal demodulation can be accomplished by the software soundmodem (*UZ7HO Personal page - Packet-Radio - English version* n.d.). Ubuntu uses PulseAudio as an audio driver which allows us an inbuilt way to redirect the sound output of GQRX to the microphone input of soundmodem. In principle this means that all we need to do to configure soundmodem is to allow it to input through the microphone. Since the ground station currently is not able to visualise SDR signals in GQRX, we are not able to test this functionality; it shall be implemented as soon as we are able.

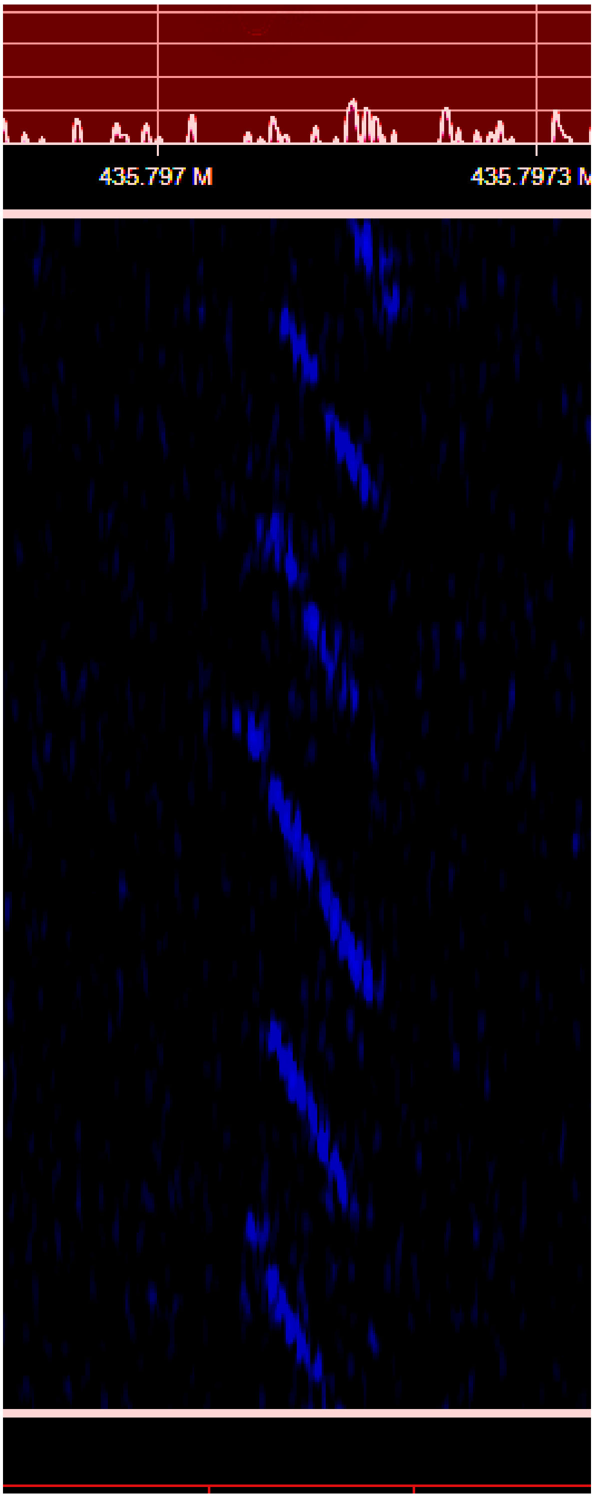


Figure 3.4: Color-corrected inset from Fig. 3.3. The periodic repeating lines are the signal, and show it appearing to move (due to the Doppler effect) from the higher frequency to the lower one, before repeating when the beacon starts emitting a new signal. As FO-29 transmitted its beacon continuously, this repeats for as long as the craft is sufficiently high above the horizon.

4

The Ground Station Upgrade

4.1 The Ground Station before the upgrade

The original ground station was designed and prototyped by Anand Thirumalai (Thirumalai 2017) in 2017 (see Fig. 4.2) before being packaged into a weatherproof container by Dr Nicholas Rattenbury, and the upgrade has changed few of the hardware components. The largest part of the project was the shift from an old LabJack U12-based rotator controller (Fig. 4.1) to a more modern Arduino-based circuit, coupled with the shift of Operating Systems from Windows 10 to Ubuntu 19.04. The firmware for the LabJack had not been updated in some twenty years, and was forcing us to use software written in 1993 – Nova for Windows. Whilst Nova is certainly a capable tool, the extensibility that comes with utilising an Arduino-based circuit is of vital importance for the future of the ground station.

Secondly, and more pressingly, the ground station was beginning to exhibit issues with spurious signals, an example of which is shown in Fig. 4.3. These "signals" swept through approx 1MHz at their strongest, corresponding to an object travelling at more than 2,000 times the speed of sound. This is not a physical scenario, hence we concluded these "signals" are likely SDR artefacts, corresponding to either strange RF signals in the ground station container, faulty cabling, or to a malfunctioning component. This was causing false positives, as shown in the tracking log in Fig. 5.9 on page 28.

4.2 The Ground Station after the upgrade

4.2.1 The Ground Station

The rotator controller is based on an Arduino UNO rev3 (*Arduino Uno Rev3* n.d.). The circuit (Figs. 4.4 and 4.7) runs a script that has been slightly modified from one written by Anthony Good (Good 2019), and is designed to emulate a Yaesu GS-232A interface (*Yaesu GS-232 Rotor Computer Interface, GS232A, gs232b* n.d.). The Y6U is rotated by a Yaesu 5500-G (*Welcome to Yaesu.com* n.d.) which in principle outputs between 2 and 4.5 VDC on DIN pins 1 and 6 (see Table 4.1) corresponding to 0 to 450° and 0 to 180°, corresponding to Azimuth and Elevation, respectively. In practice we found this to not be the case, and it output between 0 and 6 VDC corresponding to those same ranges. We thus installed a voltage divider into the circuit to compensate. Provision of the signals is much easier – the Arduino pins D6, D8, D10, D12, go high to represent a rotation Clockwise, Up, Counter-Clockwise, and Down, respectively. Any combination of these will cause the rotator to rotate in both directions at once,

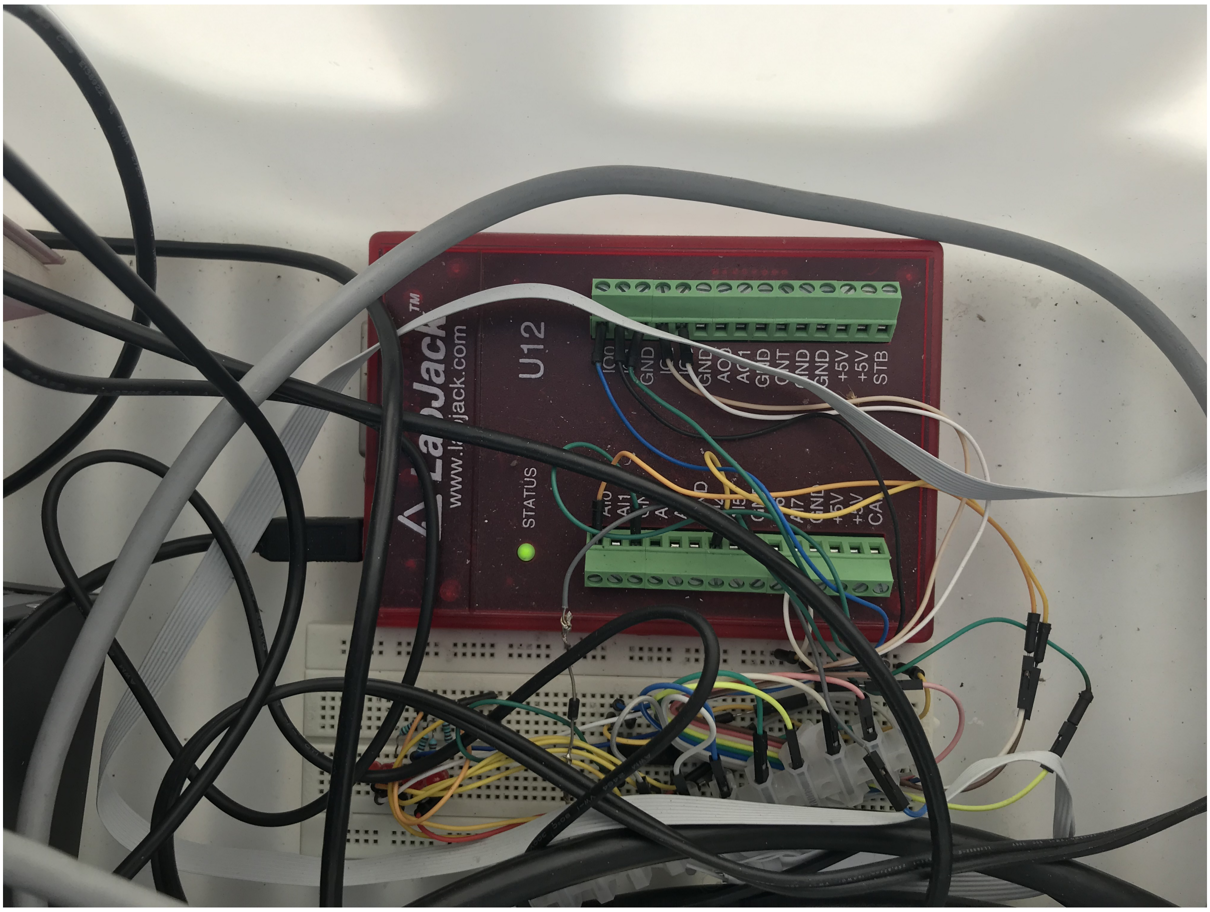


Figure 4.1: The LabJack-based circuit that was replaced.

if able.

The prototype rotator (Fig. 4.4) was built using 2N7000 MOSFETs to act as digital switches, and was designed on a breadboard in order for us to test compatibility with the Yaesu 5500G. The final version (Fig. 4.7) was built on Vero board to be able to be used as an Arduino header.

The next stage of the upgrade was to move away from a Windows-based station towards a Linux-based one. Whilst Windows is the operating system of choice for amateur satellite operators, it faces some drawbacks when compared to Linux:

(1) The Windows operating system is closed-source, meaning we are not able to make modifications to crucial components if required,

(2) HAM software is typically written for Linux distributions *first* and then ported to Windows,

(3) Linux software has a wider support community and is typically supported for longer than Windows versions, and

(4) Linux is the software of choice for embedded devices and so-called "Internet of Things" devices.

We use a Raspberry Pi running Raspbian (*Buy a Raspberry Pi 3 Model B – Raspberry Pi* n.d.) to control the Arduino. The Pi is configured to run two important daemons (background processes) on startup: rotctld (*Hamlib/Hamlib* 2019) and SpyServer (*SDR# and Airspy Downloads - airspy.com* n.d.). rotctld is a daemon allowing GPredict to interface with our rotator and control it; SpyServer is an SDR server which allows antenna signals received by the Raspberry Pi to be streamed over SSH to a remote control computer. The net effect of this is that a ground station operator may control the ground

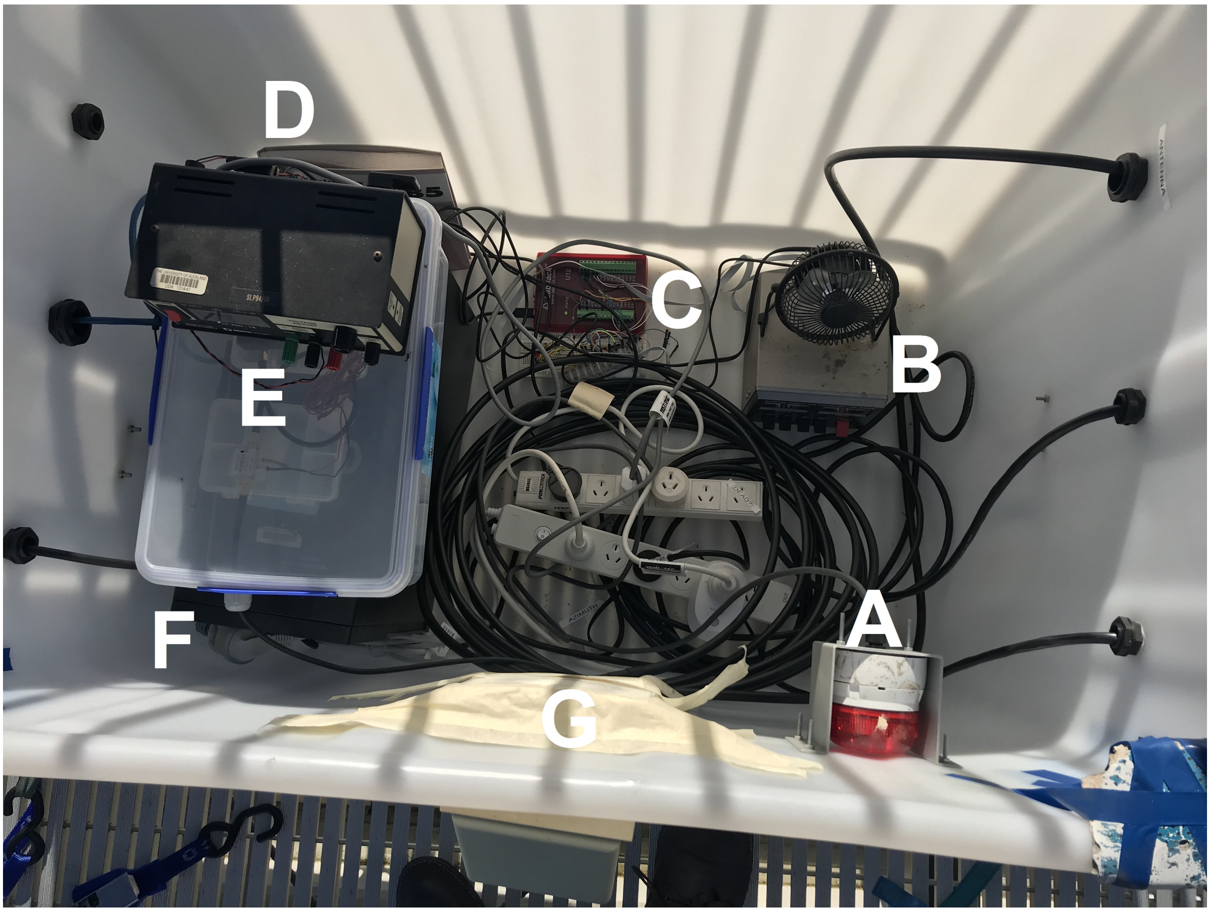


Figure 4.2: The ground station prior to the upgrade. **A**: A "power-on" light. **B**: A small fan sitting atop the Yaesu-G5500 rotator controller. **C**: The old LabJack circuit (see Fig. 4.1). **D**: The maintenance log for the ground station. **E**: A power supply for the Low Noise Amplifier (LNA), which sits in the tupperware container along with the SDR. **F**: The original analysis machine. **G**: A ventilation fan to reduce heat in the ground station.

station from any device at any place from within the University, so long as they have an SDR client (i.e., GQRX) and GPredict installed on that local machine. See Fig. 4.6 for a functional diagram of the upgraded station.

4.2.2 The Analysis Machine

The main analysis machine of the ground station is an Ubuntu 19.04 machine which has soundmodem, GPredict, and GQRX installed on it. It is currently located on the 8th floor of building 303 – inside the "shack" – but will eventually be moved to a more appropriate place such as the APSS laboratory on campus.

We use SpyServer on the ground station because it is the most widely supported daemon for remote streaming for the Airspy. However, it is fully closed source and therefore is not natively supported by GQRX. This has the unfortunate side effect of needing to use a patched version of gr-osmosdr, which is a sublibrary of osmocom (*Open Source Mobile Communications* 2019) which provides the abstraction between GQRX and the radio receiver (in this instance, the TCP connection). The patch is not widely supported and is suspect to some bugs and idiosyncrasies; consequently the analysis machine has a severe bug where GQRX appears to connect to SpyServer (insofar as it doesn't *fail* when connecting), however the spectrum displayed is most assuredly not what the antenna is receiving; upon testing with a

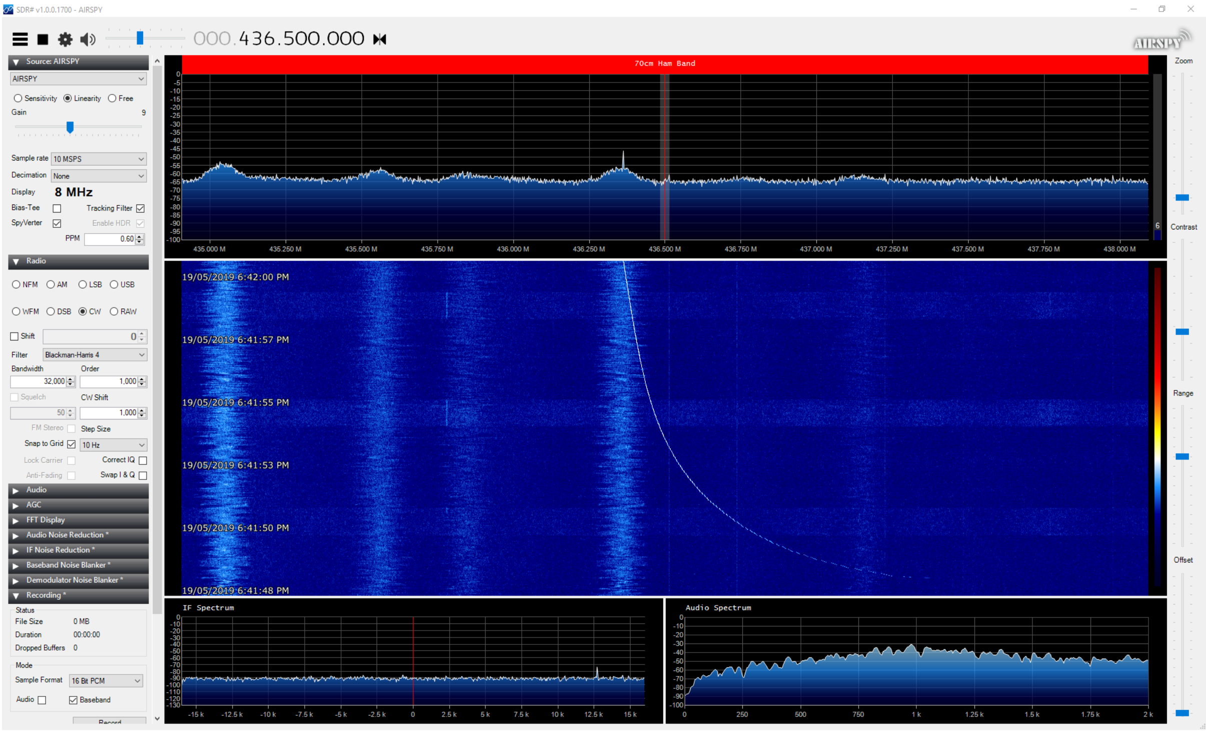


Figure 4.3: An example of a spurious signal that was observed prior to the upgrade.

stationary radio we did not see the signal on GQRX until an SDR was connected directly to the analysis machine. As discussed in Section 6, this is an issue that must be resolved over Summer.

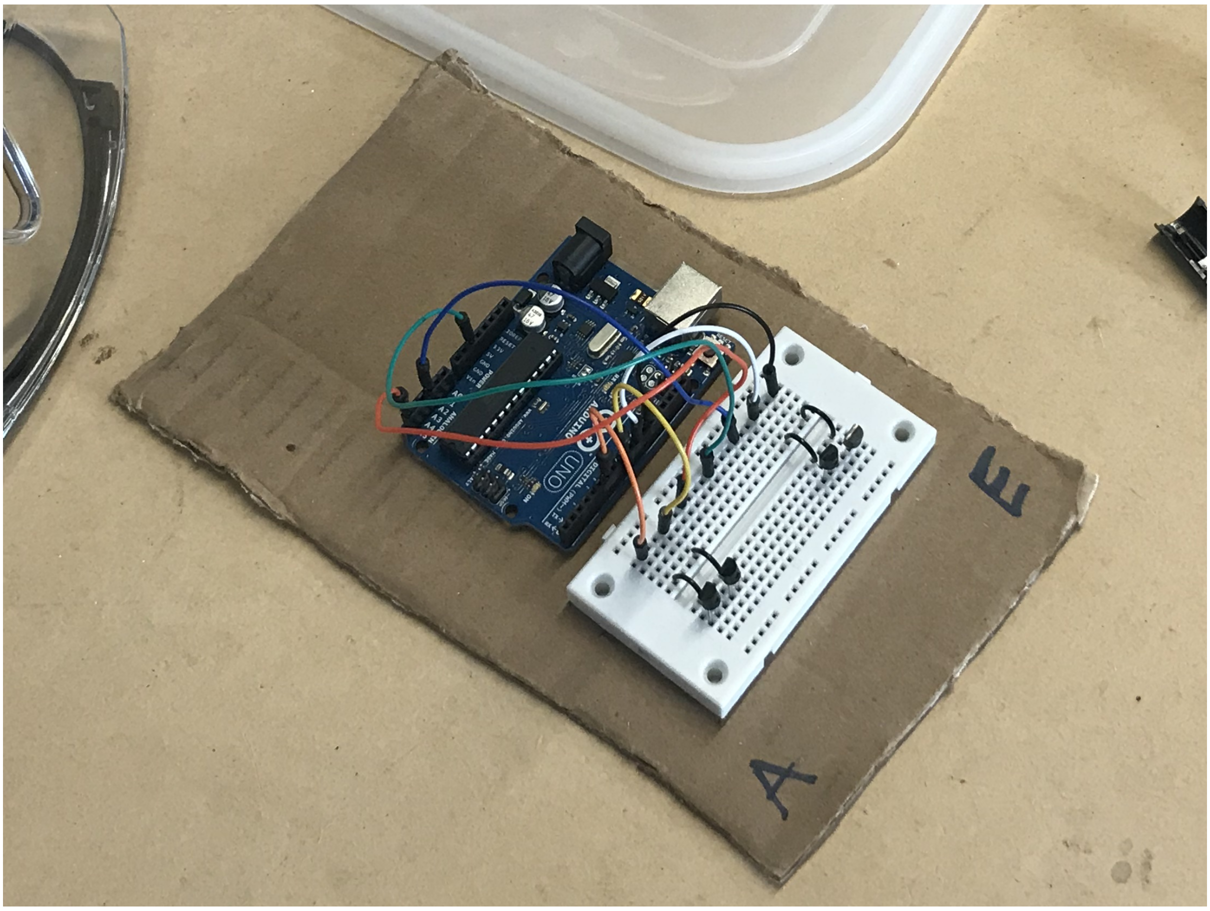


Figure 4.4: A prototype version of the new rotator circuit.

Arduino Pin	DIN Pin	Function
D6	2	Rotate Right (CW)
D8	3	Rotate Up
D10	4	Rotate Left (CCW)
D12	5	Rotate Down
A2	1	Elevation Readout
A5	6	Azimuth Readout
-	7	Dead pin
GND	-	Ground
-	8	Ground

Table 4.1: The configuration of the DIN pins.

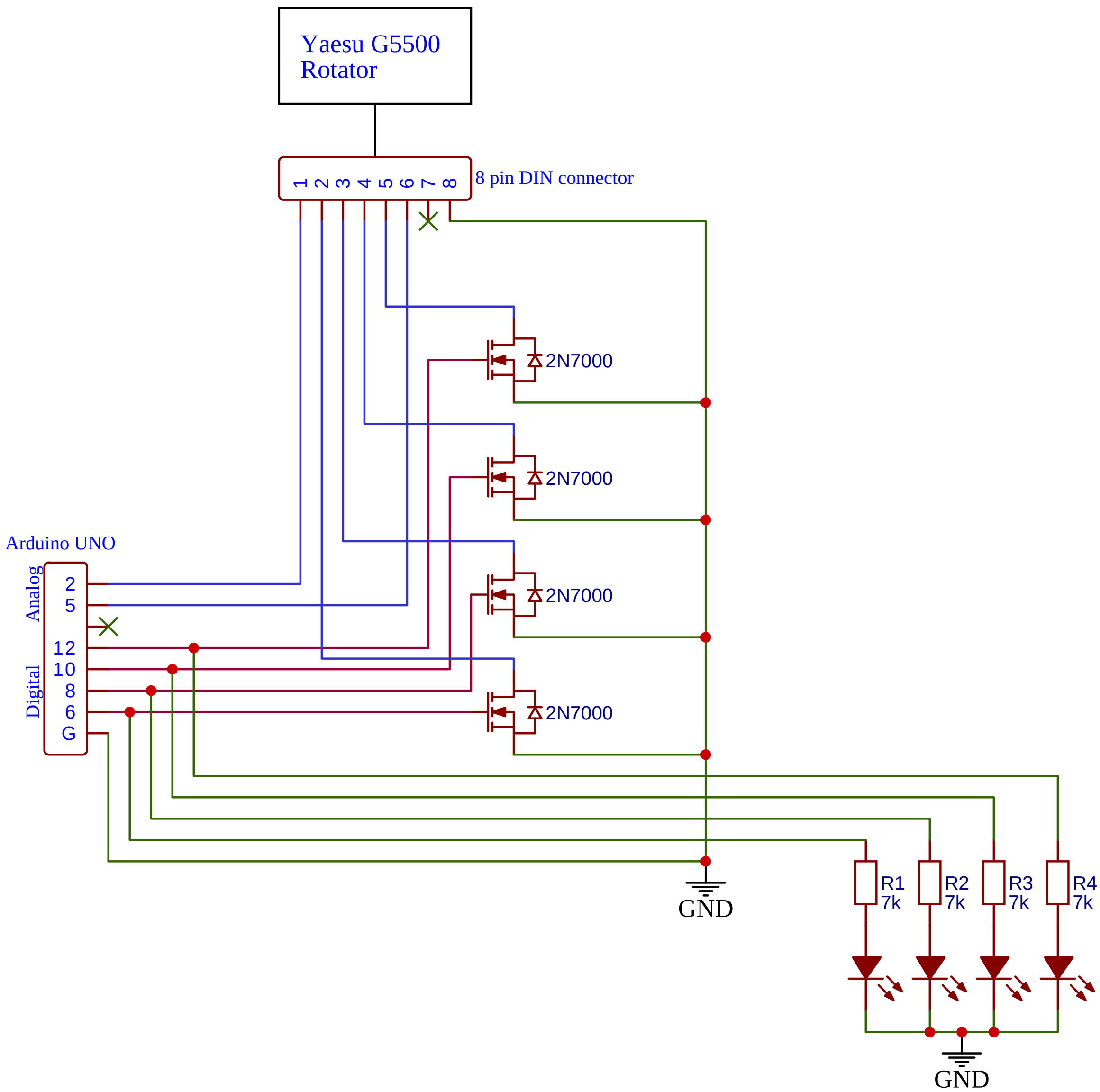


Figure 4.5: Circuit diagram of the rotator circuit. The green cross marks represent unused (i.e., dead) pins.

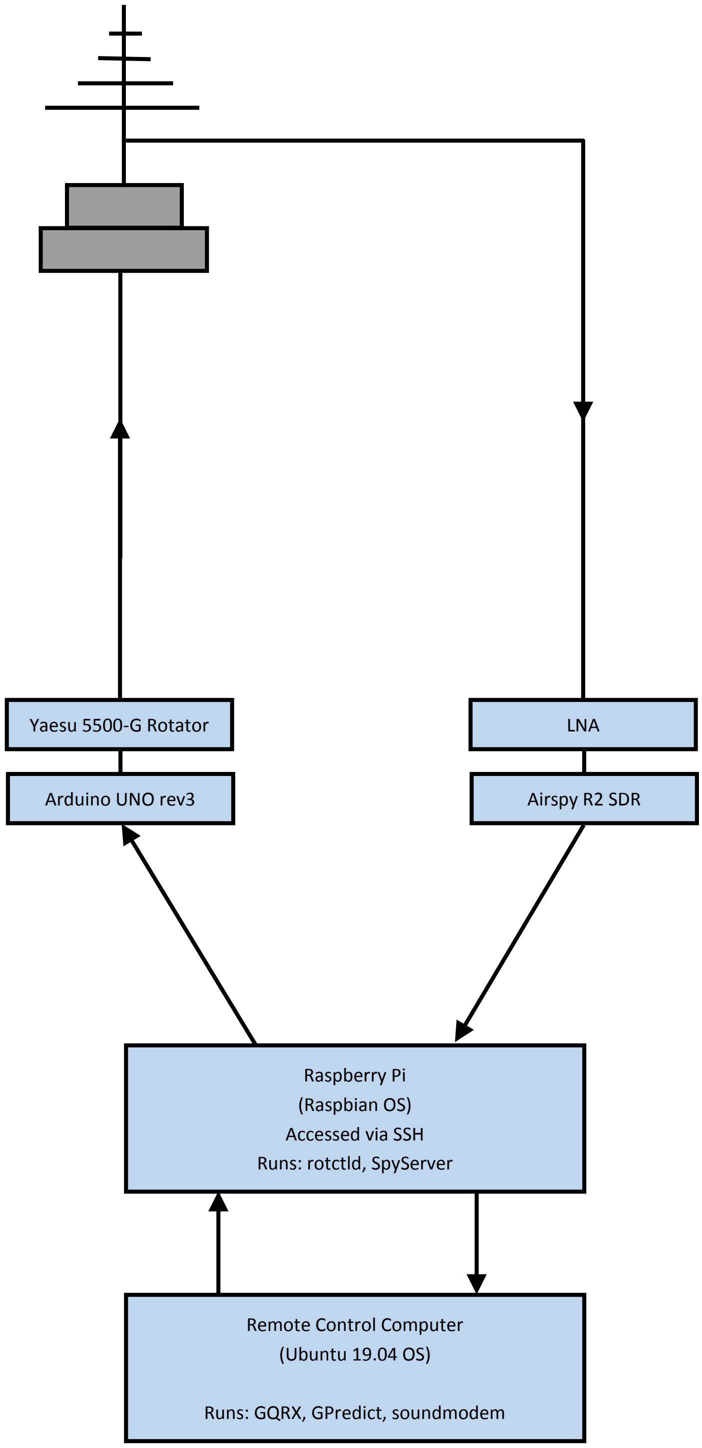


Figure 4.6: Functional Diagram for the upgraded ground station.

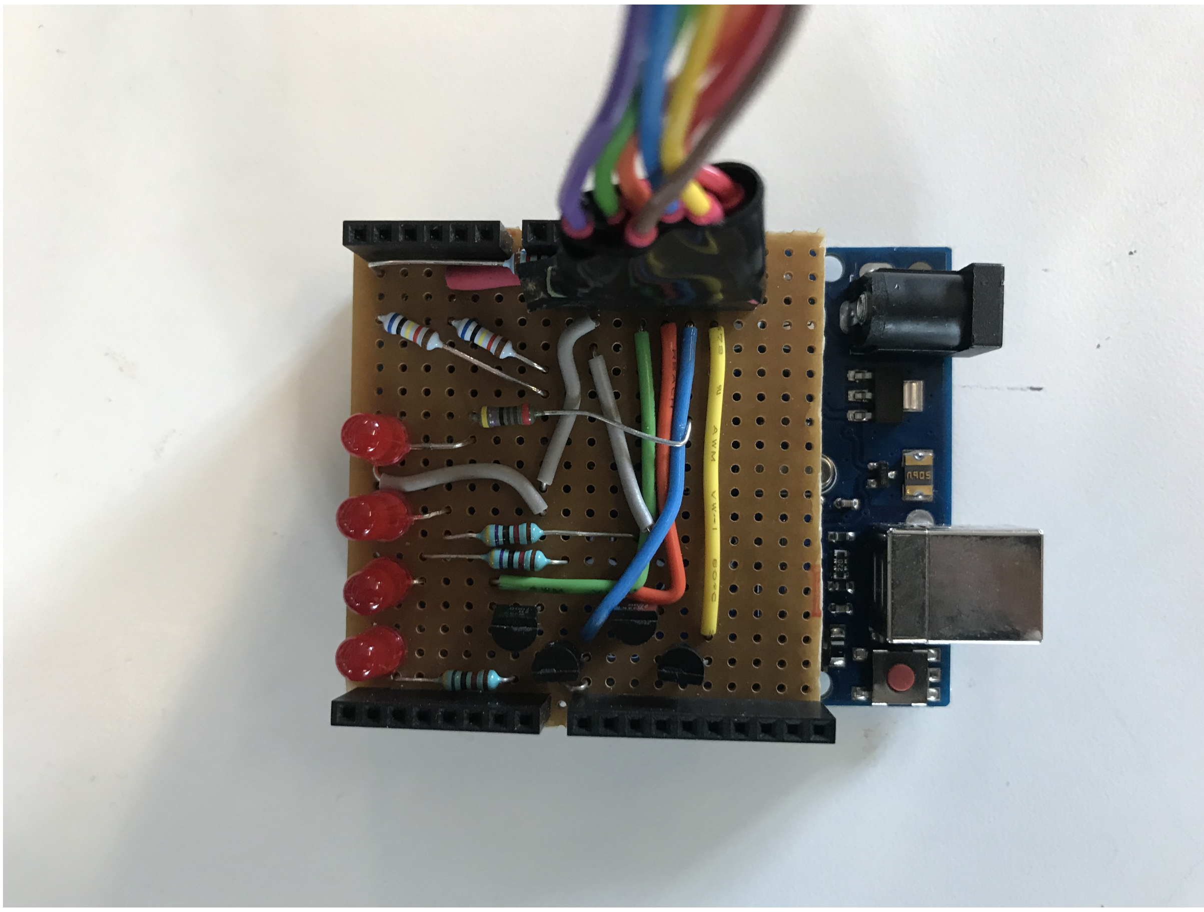


Figure 4.7: The final assembled rotator circuit.

5

Ground Station Operation

Having completed a discussion on the theory of operation, we now cover the process of tracking a satellite on the local analysis machine.

5.1 Pass Determination

The home screen of GPredict is shown in Fig. 5.1. The primary display on GPredict is a Mercator projection of the Earth (shown in red), which shows a configurable list of satellites in the sky at any given time. Beneath this is a list of each satellite shown (in blue), with its current Azimuth, Elevation, Orbital Direction, Slant Range, the time of the next event (AOS/LOS), and Altitude. The fields shown are customisable, however, so the operator is free to adjust this however they require. This satellite list is repeated on the right hand side of the main GPredict window and is highlighted again in blue there.

To the right of the map and list of satellites, is a polar plot (in yellow). The origin of the plot is the location of the ground station, and the satellites appear as red dots moving along a blue path. CAS-4A, shown in Fig. 5.1, appears to the North, flies barely above the horizon, and disappears to the East. These "sky tracks" appear as necessary and are not visible for satellites not currently within view of the observer. Underneath the plot is a repeat of the list underneath the map, with fewer fields.

Underneath this is a so-called "single-sat display" (in green). This display shows the parameters of a selected satellite, enabling the Operator to see potentially important information.

Right-clicking on any satellite in the list allows the operator to see the Satellite Information and Transponders of said satellite, as shown in Fig. 5.2. "Orbit Information" shows essentially a decoding of the first line of the Two-Line Element data set corresponding to the satellite. Transponders shows a list of all known active transponders of the satellite, allowing the Operator to see what they may listen to.

To identify a viable pass, one must right-click on the satellite (in map view, or single-sat view, or list view) and opt to see future passes. We shall consider the satellite SWISSCUBE, and its future passes are listed in Fig. 5.3. Identification of a viable pass is a reasonably simple process – we identify any pass with a maximum elevation of higher than $\sim 10°$. Once we have found such a pass, we may double-click on the pass in the window to show the pass in greater detail.

The two plots of interest are a polar plot and an Azimuth/Elevation plot for the pass. Firstly, the polar plot. The ground station currently has reasonably free access to the sky in the Northeastern quadrant, and in all quadrants above 10-20° or so. Any pass must conform to at least one of these two parameters to be viable. Shown in Fig. 5.4 is a sample pass that passes almost through zenith and would

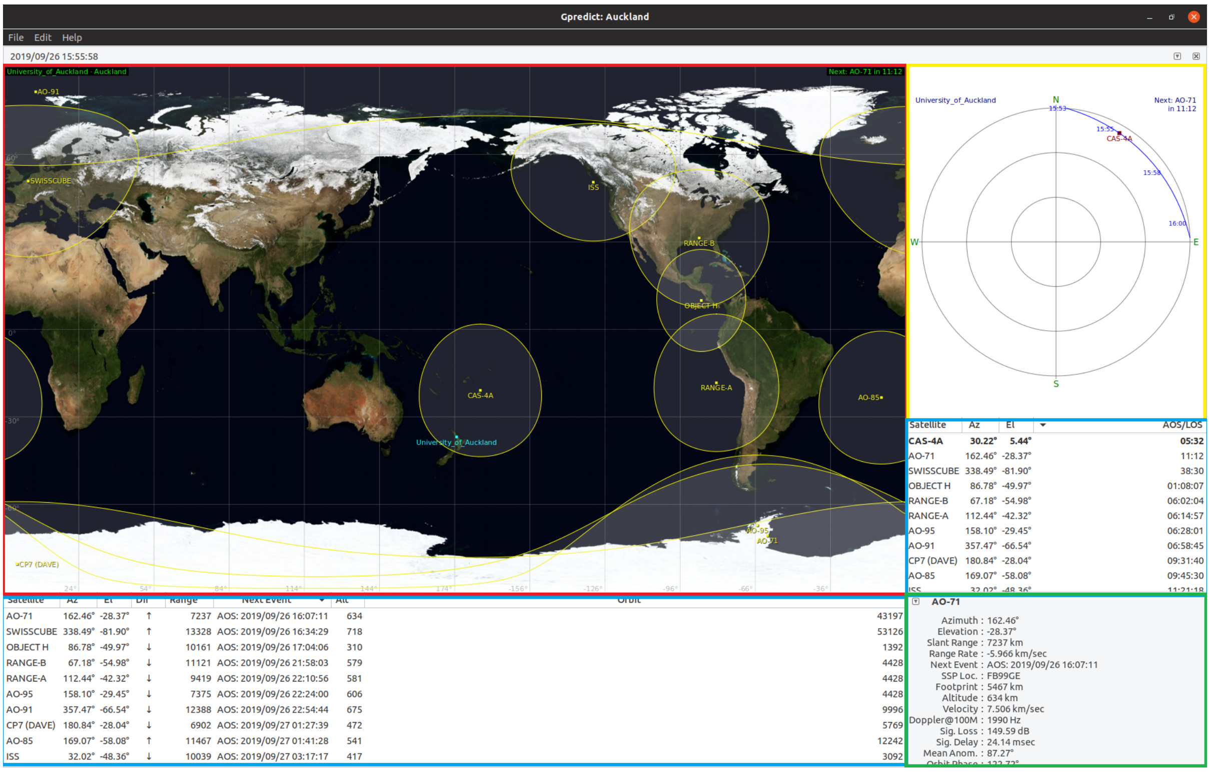


Figure 5.1: The home screen of GPredict. See the beginning of Section 5.1 for a description of each major subwindow.

be an ideal pass to track.

As well as being the prediction tool, GPredict is the main rotator control system. By clicking the arrow in the top right of the main display, we can open the Rotator and Radio Control windows, as shown in Fig. 5.6 (rotator control) and Fig. 5.5 (radio control).

5.1.1 Radio Control

First, we must ensure that the Target (highlighted in red in Fig. 5.5) is successfully configured. The top field is the satellite we are tracking, and underneath it is the transponder we wish to tune to. The UHF antenna is a Yagi Y6U, with a frequency range of 400-500 MHz, and a bandwidth of 10 MHz. Any transponders that transmit at frequencies within these parameters are able to be tracked.

Next, in blue, is the Settings pane. The “Device” refers to the specific radio we wish to connect to. For our purposes, this is GQRX (which is what receives the signal from the Airspy SDR).

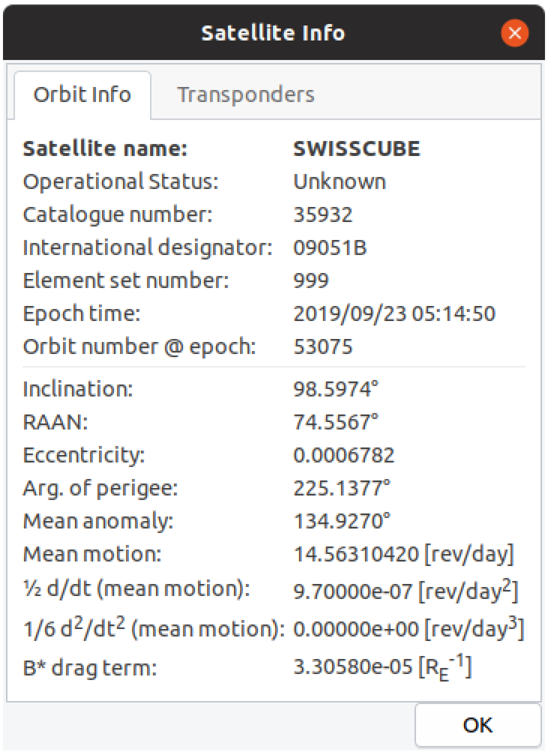
Now this done, we may click *Track*, which enables Doppler shift correction, *T* (for *Tune*) which tunes the radio to the desired frequency, and *Engage* which enables the throughput to GQRX. The use of GQRX is detailed in Section 5.1.3.

Finally, in green is the countdown timer to AOS/LOS. This enables us to see at any point when we will gain or lose the signal.

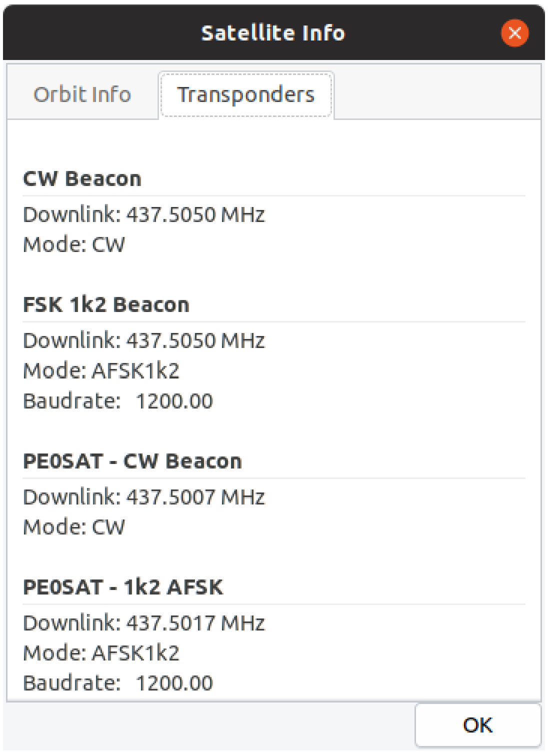
5.1.2 Rotator Control

The Rotator controller in GPredict – shown in Fig. 5.6 – is a simple, yet powerful, tool. It is split into five main sections, which are (clockwise from left):

(1) A recreation of the polar plot from the main screen of GPredict. This also provides a graphical display of where the antenna is pointing and where it is moving towards, if it is moving.



(a) Satellite Information



(b) Transponders window

Figure 5.2: The Satellite Information and Transponders Windows of the SWISSCUBE satellite.

(2) A manual controller for Azimuth.

(3) A manual controller for Elevation.

(4) A *settings* window, allowing us to change which antenna we are rotating (in the case of a system with multiple rotators), the tolerance, and whether or not we wish to lock the rotator so it cannot move (the *Monitor* checkbox).

(5) A *target* window, which allows us to change which satellite we are tracking.

The philosophy here is simple; if one selects the right target (in this instance, SWISSCUBE), the right device (Antenna_Array), and confirm a *Tolerance* (the maximum discrepancy we can allow between where the antenna should be pointing and where it is *actually* pointing – typically around 2° is plenty), then all one has to do to start the antenna moving is to click *Track* to start the tracking algorithm, and *Engage* to turn on the Arduino.

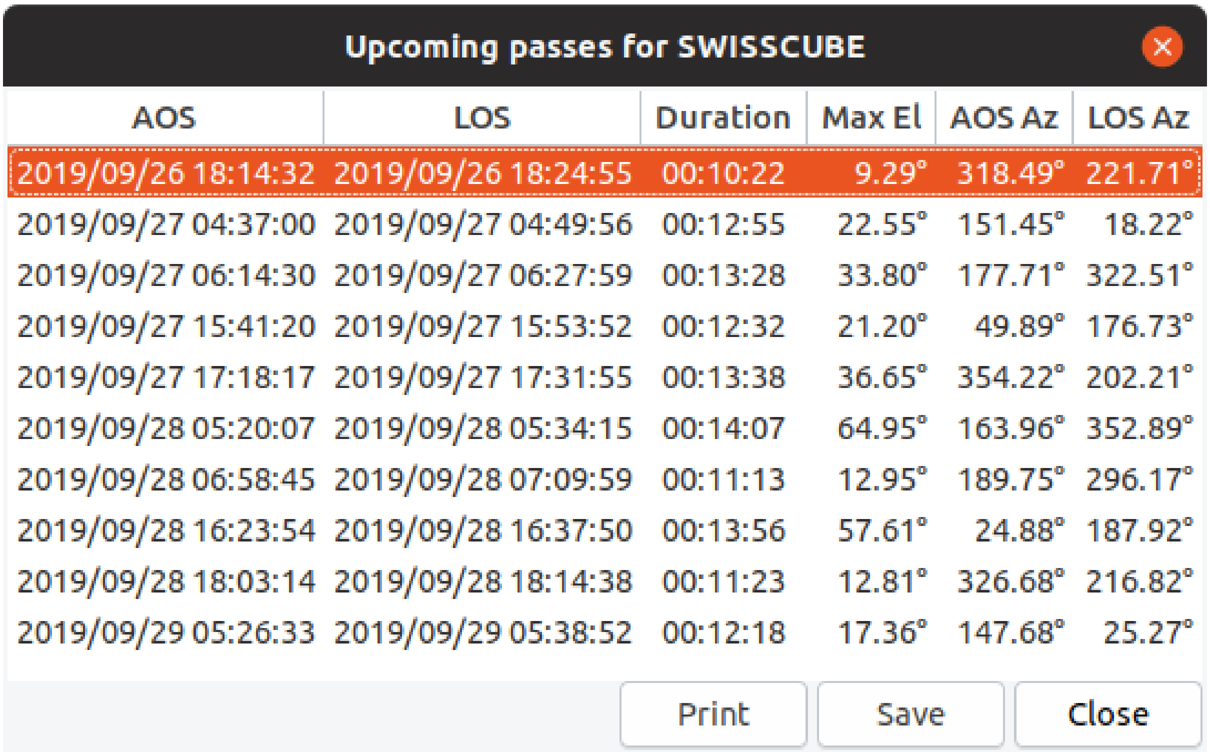
On the polar plot, one notices three important features when tracking is engaged;

(1) A red circle, which is where GPredict is attempting to move the antenna to,

(2) A red crosshairs, which represents where the antenna is presently pointing to, and

(3) A red square, which represents the position of the satellite on the sky.

If all three are aligned – that indicates the antenna is tracking the satellite.



Upcoming passes for SWISSCUBE

AOS	LOS	Duration	Max El	AOS Az	LOS Az
2019/09/26 18:14:32	2019/09/26 18:24:55	00:10:22	9.29°	318.49°	221.71°
2019/09/27 04:37:00	2019/09/27 04:49:56	00:12:55	22.55°	151.45°	18.22°
2019/09/27 06:14:30	2019/09/27 06:27:59	00:13:28	33.80°	177.71°	322.51°
2019/09/27 15:41:20	2019/09/27 15:53:52	00:12:32	21.20°	49.89°	176.73°
2019/09/27 17:18:17	2019/09/27 17:31:55	00:13:38	36.65°	354.22°	202.21°
2019/09/28 05:20:07	2019/09/28 05:34:15	00:14:07	64.95°	163.96°	352.89°
2019/09/28 06:58:45	2019/09/28 07:09:59	00:11:13	12.95°	189.75°	296.17°
2019/09/28 16:23:54	2019/09/28 16:37:50	00:13:56	57.61°	24.88°	187.92°
2019/09/28 18:03:14	2019/09/28 18:14:38	00:11:23	12.81°	326.68°	216.82°
2019/09/29 05:26:33	2019/09/29 05:38:52	00:12:18	17.36°	147.68°	25.27°

Figure 5.3: Upcoming passes for the CubeSat SWISSCUBE.

5.1.3 Signal Reception and Interpretation

Once we have determined when a pass is to occur, begun the antenna rotation, and enabled the radio, we must turn to the software GQRX to begin listening. The GQRX main display is shown in Fig. 5.7.

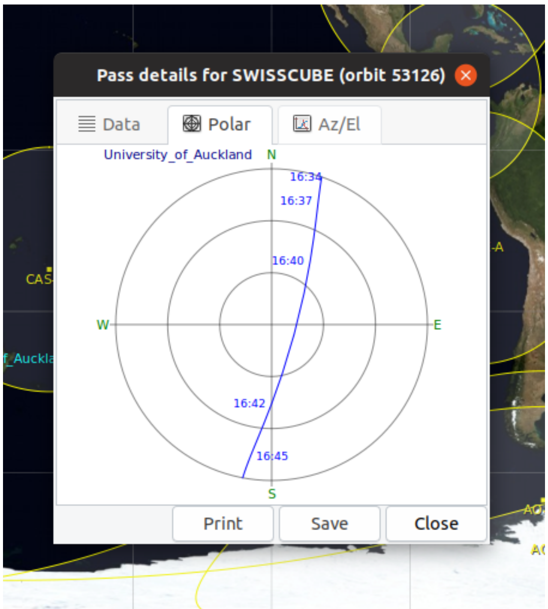
To turn on the SDR, there are two steps we must undertake; firstly, we must enable the GQRX to GPredict throughput by pressing the button highlighed in red in Fig. 5.7, and secondly we must press the button highlighted in blue to turn on the SDR. At this point, GPredict may disengage the connection, so we must switch back to GPredict and re-engage the connection in Radio Control. The orange line indicates a connection between GQRX (running locally) and SpyServer, which is the software on the raspberry pi which allows the SDR to transmit over the internet.

The GPredict to GQRX throughput is of vital importance. GPredict is what computes the Doppler Shift correction, however GQRX is what implements that correction. Therefore, we require the connection in order to distinguish satellites from stationary sources.

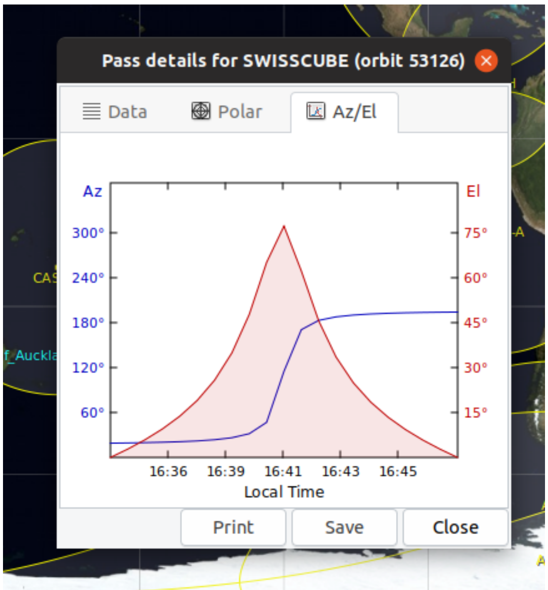
When GQRX is enabled, the Fourier Transform window and Waterfall plots activate, as shown in Fig. 5.8. The throughput functionality causes the frequency displayed at the top of the FFT display to shift as the satellite passes, which in turn causes the center frequency to sweep through the spectrum in order to listen for satellites. A satellite passing overhead is visualised in this way as a *signal* that sweeps across the waterfall; this enables us to rapidly identify candidate signals.

5.2 Pass Logging

All tracking attempts are logged on a communal Google Spreadsheet, as shown in Fig. 5.9. This is colour-coded for successful attempts (green) and unsuccessful attempts (red). We also mark important milestones in the tracking log history, such as when spurious signal issues arise and are resolved, and when the upgrade was completed. These are shown as markings on the log. The log establishes accountability



(a) Polar Plot



(b) Az/El Plot

Figure 5.4: The pass parameters of SWISSCUBE. Particular attention is drawn in (a) to the fact that the blue curve (the sky track of SWISSCUBE) passes very near to the origin, being zenith, and in (b) to the height of the red peak, indicating a high ($\simeq 75°$) maximum elevation.

for the ground station and allows independent verification of signals. More obscure in GPredict is the ability to control the flow of time, allowing us to reverse back and view which satellites were in the sky at a given time in the event of a queried signal. There are thirteen fields in the log:

(1) **Date**, corresponding to the date of the pass,

(2) **Time**, corresponding to the time (local time, though this may be configured to UTC if desired) of the pass,

(3) **Orbit**, corresponding to the orbit number of the satellite, as reported by GPredict,

(4) **Success**, which is a simple Yes/No field representing a successful/failed pass respectively,

(5) **Satellite**, the human-readable name of the satellite,

(6) **Status**, as reported by Mike Rupprecht in Germany (Rupprecht n.d.),

(7) **NORAD**, representing the NORAD identifier of the satellite where known,

(8) **Uplink**, representing the uplink frequency of the satellite in MHz,

(9) **Downlink**, representing the downlink frequency of the satellite in MHz,

(10) **Beacon**, representing the beacon frequency of the satellite in MHz,

(11) **Mode**, detailing what modulation schematic is employed onboard the craft,

(12) **Tracked by**, logging the identity of the person who tracked the satellite, and

(13) **Notes**, where notes about the signal are stored.

The references to "false positives" in the log refer to the spurious signals we saw, and were discussed in Section 4.1. The thick black line indicates where the spurious signals were resolved, and the thinner black line indicates when the upgrade occurred.

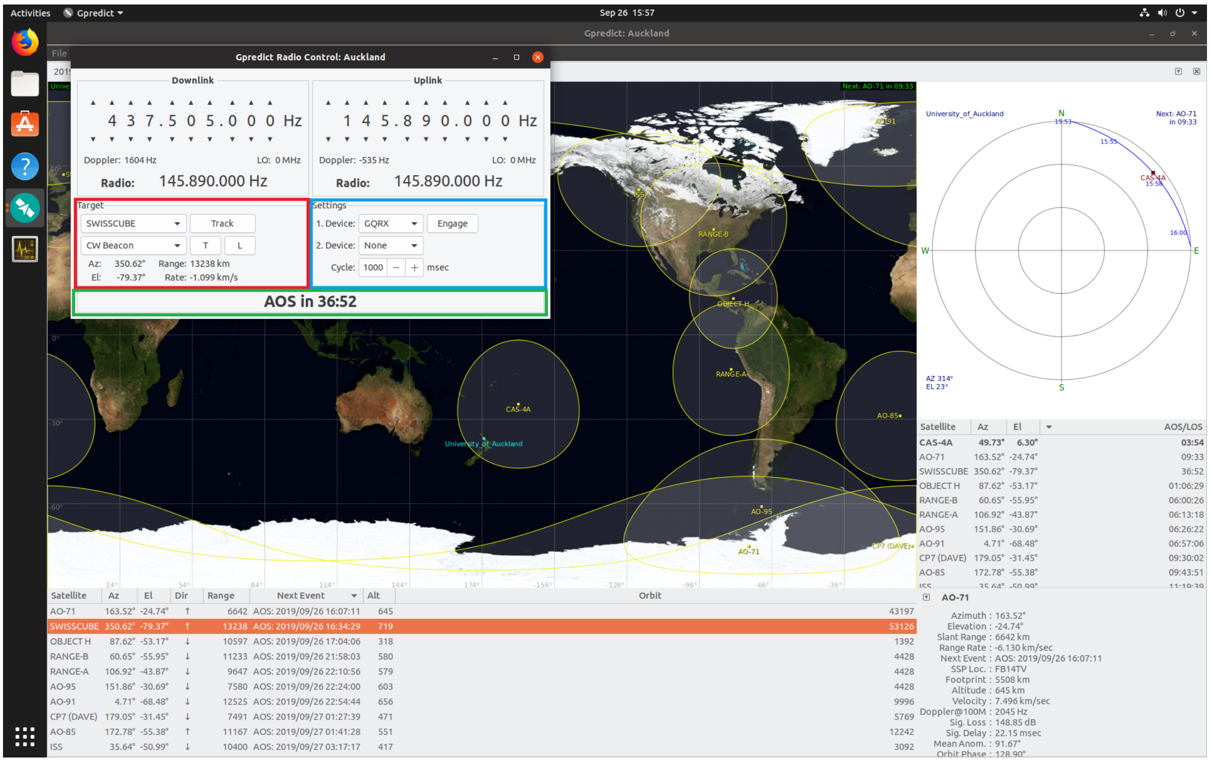


Figure 5.5: Radio Control Window. See Section 5.1.1 for an explanation of the fields.

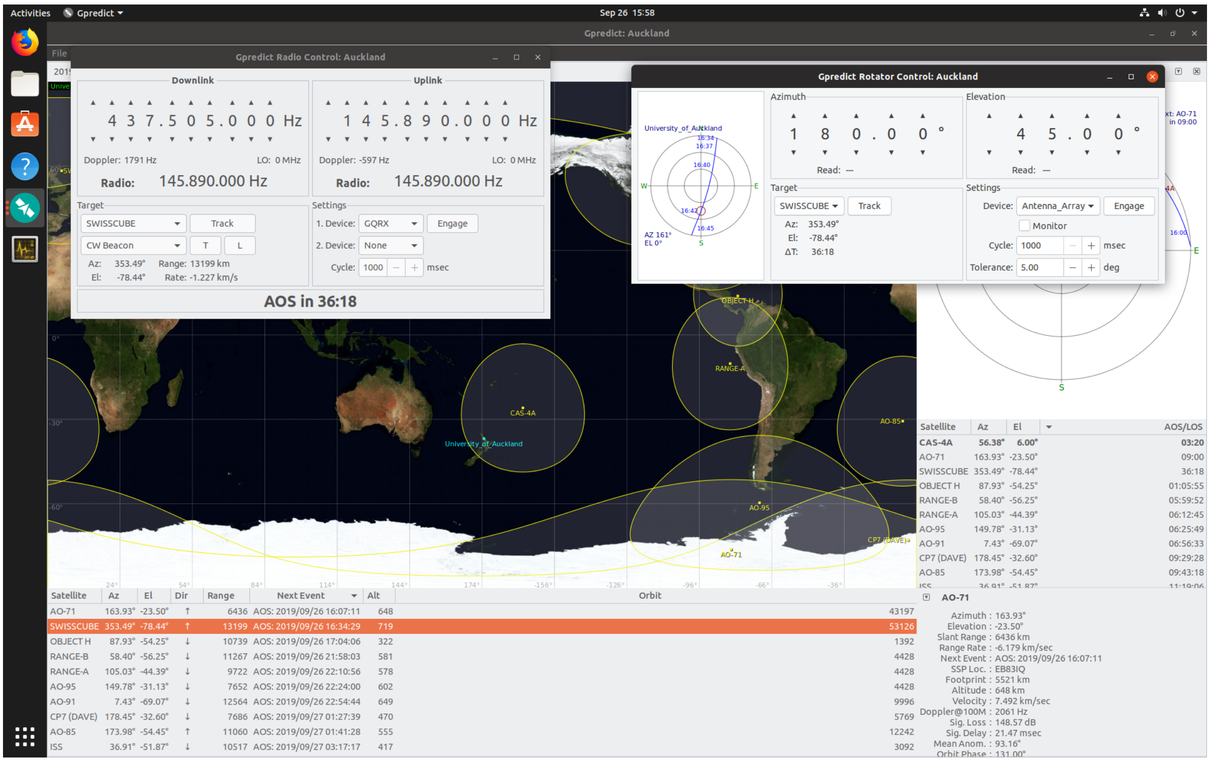


Figure 5.6: Rotator Control window. See Section 5.1.2 for an explanation of the fields.

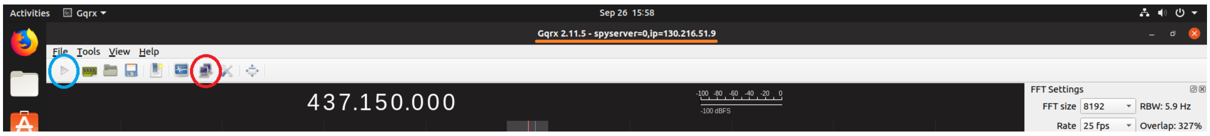


Figure 5.7: The main control bar of GQRX. The “play” icon circled in blue starts the SDR, and the “computer” icon circled in red launches the TCP server.

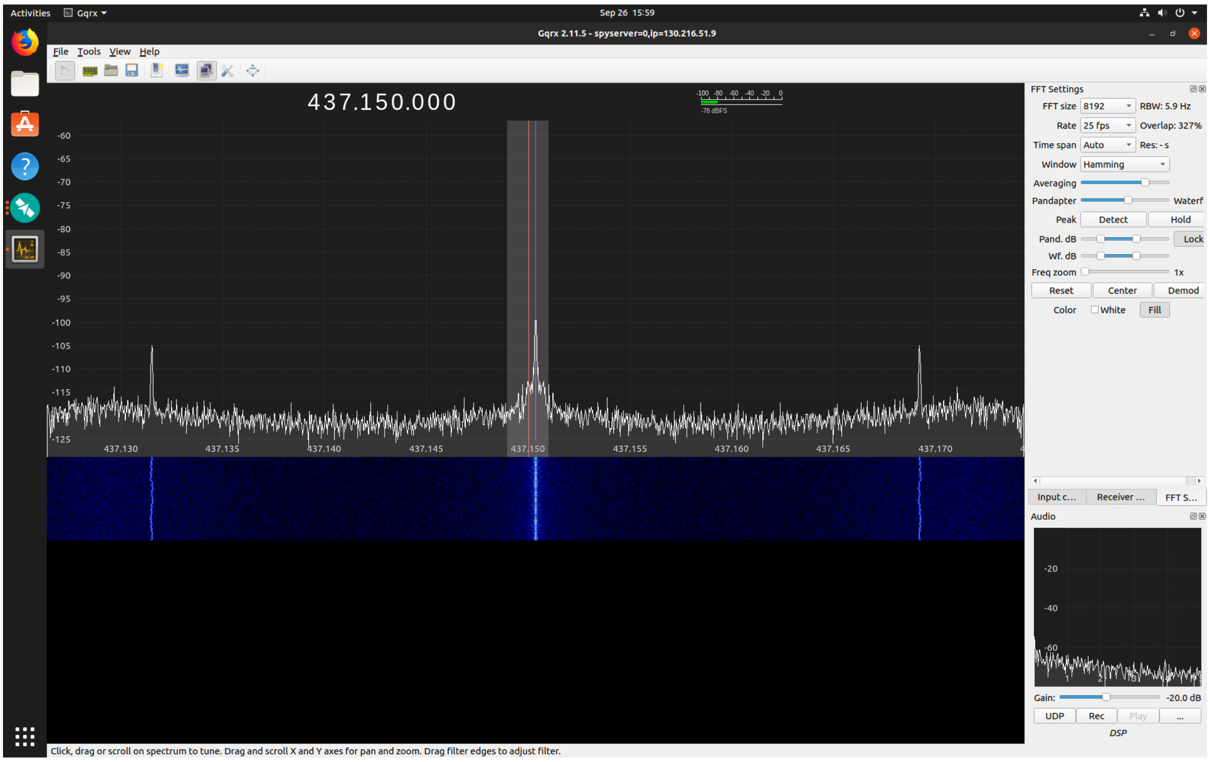
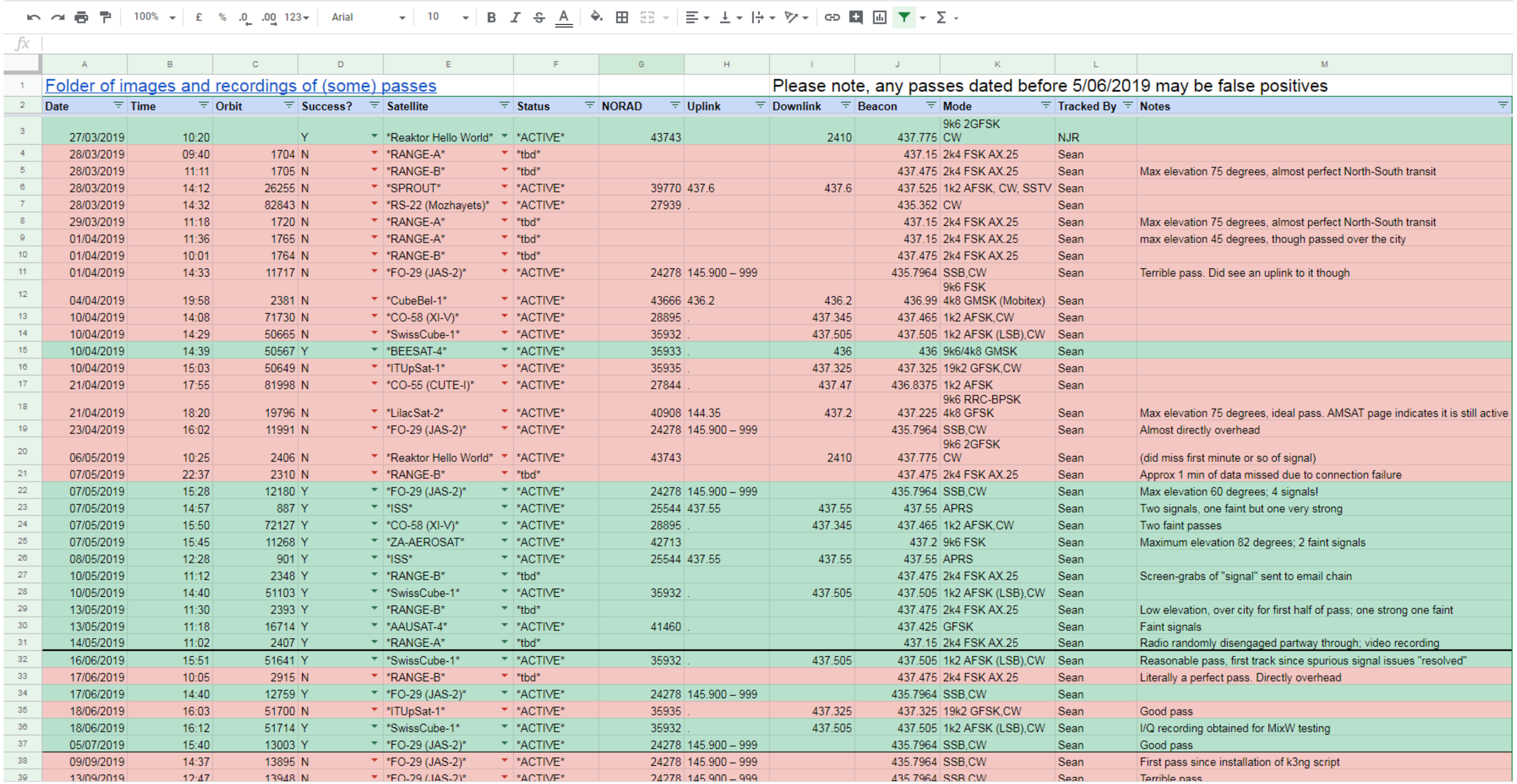


Figure 5.8: The main GQRX window when GQRX is running. The peaks are due to stationary sources of noise and are shown as examples of signals.



[Folder of images and recordings of (some) passes](#)

Please note, any passes dated before 5/06/2019 may be false positives

Date	Time	Orbit	Success?	Satellite	Status	NORAD	Uplink	Downlink	Beacon	Mode	Tracked By	Notes
27/03/2019	10:20		Y	*Reaktor Hello World*	*ACTIVE*	43743		2410	437.775	9k6 2GFSK CW	NJR	
28/03/2019	09:40	1704	N	*RANGE-A*	*tbd*				437.15	2k4 FSK AX.25	Sean	
28/03/2019	11:11	1705	N	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	Max elevation 75 degrees, almost perfect North-South transit
28/03/2019	14:12	26255	N	*SPROUT*	*ACTIVE*	39770	437.6	437.6	437.525	1k2 AFSK, CW, SSTV	Sean	
28/03/2019	14:32	82843	N	*RS-22 (Mozhayets)*	*ACTIVE*	27939	.		435.352	CW	Sean	
29/03/2019	11:18	1720	N	*RANGE-A*	*tbd*				437.15	2k4 FSK AX.25	Sean	Max elevation 75 degrees, almost perfect North-South transit
01/04/2019	11:36	1765	N	*RANGE-A*	*tbd*				437.15	2k4 FSK AX.25	Sean	max elevation 45 degrees, though passed over the city
01/04/2019	10:01	1764	N	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	
01/04/2019	14:33	11717	N	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	Terrible pass. Did see an uplink to it though
04/04/2019	19:58	2381	N	*CubeBel-1*	*ACTIVE*	43666	436.2	436.2	436.99	9k6 FSK 4k8 GMSK (Mobitex)	Sean	
10/04/2019	14:08	71730	N	*CO-58 (XI-V)*	*ACTIVE*	28895	.	437.345	437.465	1k2 AFSK,CW	Sean	
10/04/2019	14:29	50665	N	*SwissCube-1*	*ACTIVE*	35932	.	437.505	437.505	1k2 AFSK (LSB),CW	Sean	
10/04/2019	14:39	50567	Y	*BEESAT-4*	*ACTIVE*	35933	.	436	436	9k6/4k8 GMSK	Sean	
10/04/2019	15:03	50649	N	*ITUpSat-1*	*ACTIVE*	35935	.	437.325	437.325	19k2 GFSK,CW	Sean	
21/04/2019	17:55	81998	N	*CO-55 (CUTE-I)*	*ACTIVE*	27844	.	437.47	436.8375	1k2 AFSK	Sean	
21/04/2019	18:20	19796	N	*LilacSat-2*	*ACTIVE*	40908	144.35	437.2	437.225	9k6 RRC-BPSK 4k8 GFSK	Sean	Max elevation 75 degrees, ideal pass. AMSAT page indicates it is still active
23/04/2019	16:02	11991	N	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	Almost directly overhead
06/05/2019	10:25	2406	N	*Reaktor Hello World*	*ACTIVE*	43743		2410	437.775	9k6 2GFSK CW	Sean	(did miss first minute or so of signal)
07/05/2019	22:37	2310	N	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	Approx 1 min of data missed due to connection failure
07/05/2019	15:28	12180	Y	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	Max elevation 60 degrees; 4 signals!
07/05/2019	14:57	887	Y	*ISS*	*ACTIVE*	25544	437.55	437.55	437.55	APRS	Sean	Two signals, one faint but one very strong
07/05/2019	15:50	72127	Y	*CO-58 (XI-V)*	*ACTIVE*	28895	.	437.345	437.465	1k2 AFSK,CW	Sean	Two faint passes
07/05/2019	15:45	11268	Y	*ZA-AEROSAT*	*ACTIVE*	42713			437.2	9k6 FSK	Sean	Maximum elevation 82 degrees; 2 faint signals
08/05/2019	12:28	901	Y	*ISS*	*ACTIVE*	25544	437.55	437.55	437.55	APRS	Sean	
10/05/2019	11:12	2348	Y	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	Screen-grabs of "signal" sent to email chain
10/05/2019	14:40	51103	Y	*SwissCube-1*	*ACTIVE*	35932	.	437.505	437.505	1k2 AFSK (LSB),CW	Sean	
13/05/2019	11:30	2393	Y	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	Low elevation, over city for first half of pass; one strong one faint
13/05/2019	11:18	16714	Y	*AAUSAT-4*	*ACTIVE*	41460	.		437.425	GFSK	Sean	Faint signals
14/05/2019	11:02	2407	Y	*RANGE-A*	*tbd*				437.15	2k4 FSK AX.25	Sean	Radio randomly disengaged partway through; video recording
16/06/2019	15:51	51641	Y	*SwissCube-1*	*ACTIVE*	35932	.	437.505	437.505	1k2 AFSK (LSB),CW	Sean	Reasonable pass, first track since spurious signal issues "resolved"
17/06/2019	10:05	2915	N	*RANGE-B*	*tbd*				437.475	2k4 FSK AX.25	Sean	Literally a perfect pass. Directly overhead
17/06/2019	14:40	12759	Y	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	
18/06/2019	16:03	51700	N	*ITUpSat-1*	*ACTIVE*	35935	.	437.325	437.325	19k2 GFSK,CW	Sean	Good pass
18/06/2019	16:12	51714	Y	*SwissCube-1*	*ACTIVE*	35932	.	437.505	437.505	1k2 AFSK (LSB),CW	Sean	I/Q recording obtained for MixW testing
05/07/2019	15:40	13003	Y	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	Good pass
09/09/2019	14:37	13895	N	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	First pass since installation of k3ng script
13/09/2019	12:47	13948	N	*FO-29 (JAS-2)*	*ACTIVE*	24278	145.900 – 999		435.7964	SSB,CW	Sean	Terrible pass

Figure 5.9: The tracking log for pass attempts.

6
Future Steps

Life, forever dying to be born afresh, forever young and eager, will presently stand upon this Earth as upon a footstool, and stretch out its realm amidst the stars.

H. G. Wells

6.1 Future Missions

The overarching aim of the ground station upgrade is to allow automation of the process of satellite tracking, by way of enabling authorised users to submit tracking requests and allowing the system to complete this tracking automatically. Additionally, Te Pūnaha Ātea wants to launch New Zealand's first optical space telescope on a 3U CubeSat, and to launch a CubeSat to study the atmosphere of Venus. These missions will sit alongside the ground station, and as such, the system will need to be extended to truly support both multiple concurrent missions and the ability to autonomously track any required passes.

6.2 Hardware Changes

In May 2019, we received a new 2MCP14 VHF antenna (*2MCP14, 143-148 MHz* n.d.). This antenna will both double the satellites the ground station can hear, and allow us entry into the world of satellite uplink communication, which in principle would allow us to transmit commands to craft under our control. The installation of this antenna will be the next major hardware upgrade of the ground station. Furthermore there is scope for the ground station to become autonomously operating with only supervision from a human operator to queue jobs according to a priority queue.

Moreover, the current location of the ground station will soon become restricted access – over and above the current access restriction – so it will need to move to a more permanent location. The station is also bounded to the west by the city skyline, to the east by the Chemistry building, and to the south by "the shack", the control room. The southwestern quadrant also has multiple metal poles (and a lightning rod) which unfortunately shield signals from access to the antenna. In contrast, the ideal station would be one with unrestricted access to the sky in all directions, to minimise signal loss.

6.3 Limitations

As discussed previously (predominantly in Section 3.1.3, and again in 4.2.2), the ground station is presently handicapped by its inability to display signals on local analysis computers. Our working hypothesis is that this is due to the gr-osmocom patch that we employ to allow GQRX to connect to SpyServer, however we are still in the process of fully diagnosing this issue. Future developers of the ground station have three possibilities to consider to rectify this:

(1) Removing the dependency on GQRX, and replacing it with an SDR that natively supports SpyServer connectivity. The downside is that we would lose the automatic GQRX-GPredict throughput connection which, as we have discussed previously, is a critical piece of the ground station. We would need to therefore either port forward between the Raspberry Pi and analysis machine, or write some sort of code to communicate Doppler connection over the internet.

(2) Replacing the Airspy R2 with an RTL-SDR (*Buy RTL-SDR Dongles (RTL2832U)* 2013), which would allow us to then replace SpyServer with the rtl_tcp package (*Ubuntu Manpage: rtl_tcp - an I/Q spectrum server for RTL2832 based DVB-T receivers* n.d.), which is natively supported by GQRX.

(3) Identifying and resolving the precise gr-osmosdr issue which is causing the GQRX-SpyServer connection to be nonfunctioning.

7 Conclusion

We presented a detailed discussion of the upgrade of the University of Auckland's ground station, and provided a deep methodology for future ground station operators to follow. The ground station faces difficulties at the moment, but that is due to the in-progress upgrade. We approached the problem of upgrading the ground station from the point of view of moving from older, closed-source software to a more modern, Linux-based software. Consequently, we are able to achieve a finer level of customisation, at the expense of losing the existing hardware.

We discussed the problems that we are currently facing, and detailed how over summer 2019/2020 we will be moving forward with the final part of the upgrade. The following steps of the upgrade were undertaken;

(1) Design of an Arduino-based rotator controller to replace an older LabJack U12-based circuit,

(2) Shift of the ground station from a windows-based to a Ubuntu-based analysis machine, and

(3) Replacement of Nova for Windows with gpredict, to achieve a more modern and user-friendly control system.

We identified that the next steps of the upgrade would be:

(1) To repair the issue in our patched version of gr-osmocom,

(2) Installation of the VHF antenna and diplexer,

(3) Eventually, automatic operation of the ground station.

In future, we hope that the working ground station will be able to support multiple missions, and be in a more convenient place to access for maintenance. The future of space communication at the University is bright.

Appendices

A

The Two-Body Problem

Predicting the orbital trajectory of a satellite is a two-body problem, and thus requires six unique orbital elements to fully describe the trajectory. These six elements are called the *Keplerian Elements* of the system (after Johannes Kepler), and they are described in detail in Appendix B. The six Keplerian elements of an orbit are:

(1) Orbital inclination i,

(2) Right Ascention of the Ascending Node (RAAN) (Ω),

(3) Eccentricity e,

(4) Argument of Perigee ω,

(5) Mean Anomaly M, and

(6) Mean Motion n.

These are often complemented by a reference time, called the *Epoch Time* (denoted t_0), which is a timestamp referring to when the Keplerian Elements were measured. Five of the six elements – all except Mean Anomaly – are constants of the motion, and the propagation of M obeys the equation

$$M = M_0(t + t_0) \tag{A.1}$$

where M_0 is the Mean Anomaly as measured at the Epoch Time. The prediction of an orbit is tantamount to determining the position vector $\vec{\mathbf{r}}$ and velocity vector $\vec{\mathbf{v}}$. We present the method for finding $\vec{\mathbf{r}}$ as it is the most pertinent part of the problem for predicting a pass. The full solution is detailed in (Prussing & Conway 2013).

We work in the so-called *Earth-Centered Earth-Fixed* frame (or ECEF frame), which places the origin of the Cartesian Coordinate system at the center of the Earth, and furthermore we assume an idealised system – one in which atmospheric drag is negligible and the satellite orbits in a uniform gravitational field.

Moreover we define a unit vector $\hat{\mathbf{I}}$, which is the vector from the origin to the *First Point of Aries* (Υ), a celestial reference point. We also define the Euler angle $\theta = \omega + f$, i.e., the sum of the Argument of Perigee with the *True Anomaly* f.

As discussed in (Prussing & Conway 2013), the general approach to this problem, given a complete set of Keplerian Elements, is:

(1) Compute the Eccentric Anomaly E by numerically solving Kepler's Equation:

$$M = E - e \sin E \tag{A.2}$$

(2) Compute the true anomaly f by solving

$$\cos f = \frac{\cos E - e}{1 - e \cos E} \tag{A.3}$$

(3) Compute the Euler angle θ by adding

$$\theta = \omega + f \tag{A.4}$$

(4) Solve for the semi-major axis of the orbit a by solving

$$n = \sqrt{\frac{G(M_{\text{earth}} + m_{\text{satellite}})}{a^3}} \tag{A.5}$$

(5) Solve for the magnitude of the position vector r in

$$r = \frac{a(1 - e^2)}{1 + e \cos f} \tag{A.6}$$

(6) Find the unit vector $\hat{\mathbf{r}}$ by rotating the unit vector $\hat{\mathbf{I}}$ through the angles Ω, i, θ successively. This can be accomplished by multiplying $\hat{\mathbf{I}}$ by the rotation matrix

$$\boldsymbol{R} = \begin{pmatrix} \cos\theta\cos\Omega - \cos i \sin\Omega\sin\theta & -\sin\theta\cos\Omega - \cos i \sin\Omega\cos\theta & \sin i \sin\Omega \\ \cos\theta\sin\Omega + \cos I \cos\Omega\sin\theta & -\sin\theta\sin\Omega + \cos i \cos\Omega\cos\theta & -\sin i \cos\Omega \\ \sin i \sin\theta & \sin i \cos\theta & \cos i \end{pmatrix} \tag{A.7}$$

Once the position vector $\mathbf{r} = r\hat{\mathbf{r}}$ is known, we may project that onto the surface of the Earth to determine the suborbital point. Moreover, every satellite has an area on the Earth (a circle projected onto the surface of the Earth), which can be computed. If Auckland lies within this area, then the Ground Station is able to track the satellite.

B
Two-Line Element set format

The TLE format encodes three lines of data:

(1) The satellite name,

(2) General information about the satellite, and

(3) The *Keplerian elements* of the satellite.

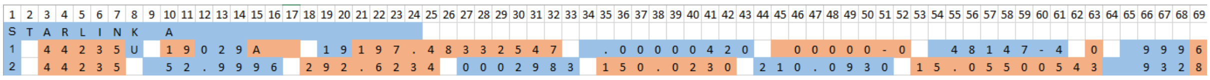


Figure B.1: A TLE file for STARLINK-A. Colours indicate successive elements in the set.

A sample TLE for the satellite STARLINK-A is shown in Fig. B.1, and an interpretation is given in Tables B.1 and B.2. Of note, the *Ephemeris Type* is an obsolete field which now conventionally reads zero. Furthermore, some fields are written in a modified form of scientific notation. In the example that follows, the BSTAR drag term (decimal assumed) is $48147 - 4$, which is shorthand for 0.48147×10^{-4}.

Column(s)	Content	Example
1	Line Number	1
3-7	Satellite Number	44235
8	Classification	U
10-11	International Designator (Year)	19
12-14	International Designator (Launch in Year)	029
15-17	International Designator (Piece of Launch)	A
19-20	Epoch Year	19
21-31	Epoch (fractional days)	197.48332547
34-43	$\frac{1}{2}\frac{\mathrm{d}n}{\mathrm{d}t}$	0.00000420
45-52	$\frac{1}{6}\frac{\mathrm{d}^2n}{\mathrm{d}t^2}$ (decimal assumed)	00000-0
54-61	BSTAR drag term (decimal assumed)	48147-4
63	Ephemeris Type	0
65-68	Element Set Number	999
69	Checksum (mod 10)	6

Table B.1: TLE format (Line 1).

Column(s)	Content	Example
2	Line Number	1
3-7	Satellite Number	44235
9-16	Orbital Inclination (i)	52.9996
18-25	RAAN (Ω)	292.6234
27-33	Eccentricity (e) (decimal assumed)	0002983
35-42	Argument of Perigee (ω)	150.0230
44-51	Mean Anomaly (M)	210.0930
53-63	Mean Motion (n)	15.0550543
64-68	Revolution Number at Epoch	932
69	Checksum (mod 10)	8

Table B.2: TLE format (Line 2).

Bibliography

2MCP14, 143-148 MHz (n.d.).
URL: *https://www.m2inc.com/FG2MCP14*

Airspy R2 - airspy.com (n.d.).
URL: *https://airspy.com/airspy-r2/*

AMSAT OSCAR Satellite Status (n.d.).
URL: *https://www.amsat.org/status/*

Arduino Uno Rev3 (n.d.).
URL: *https://store.arduino.cc/usa/arduino-uno-rev3*

Beech, W. A., Nielsen, D. E. & Taylor, J. (1998), 'AX.25 Link Access Protocol for Amateur Packet Radio', p. 143.
URL: *https://www.tapr.org/pdf/AX25.2.2.pdf*

Buy a Raspberry Pi 3 Model B – Raspberry Pi (n.d.).
URL: *https://www.raspberrypi.org*

Buy RTL-SDR Dongles (RTL2832U) (2013).
URL: *https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/*

Choi, T., Stevenson, T. & Lightsey, E. G. (2017), Reference Ground Station Design for University Satellite Missions with Varying Communication Requirements, *in* '55th AIAA Aerospace Sciences Meeting', American Institute of Aeronautics and Astronautics, Grapevine, Texas.
URL: *http://arc.aiaa.org/doi/10.2514/6.2017-1334*

Csete, A. (2019*a*), 'csete/gpredict'. original-date: 2014-08-06T20:03:51Z.
URL: *https://github.com/csete/gpredict*

Csete, A. (2019*b*), 'csete/gqrx'. original-date: 2010-10-28T21:41:07Z.
URL: *https://github.com/csete/gqrx*

Fischer, M. & Scholtz, A. L. (2010), Design of a Multi-mission Satellite Ground Station for Education and Research, *in* '2010 Second International Conference on Advances in Satellite and Space Communications', IEEE, Athens, Greece, pp. 58–63.
URL: *http://ieeexplore.ieee.org/document/5502585/*

González, F. A. D., Cabrera, P. R. P. & Calderón, C. M. H. (2016), 'Design of a Nanosatellite Ground Monitoring and Control Software – a Case Study', *Journal of Aerospace Technology and Management* **8**(2), 211–231.
URL: *http://dx.doi.org/10.5028/jatm.v8i2.553*

Good, A. (2019), 'k3ng/k3ng_rotator_controller'. original-date: 2015-03-07T00:21:05Z.
URL: *https://github.com/k3ng/k3ng_rotator_controller*

Hamlib/Hamlib (2019). original-date: 2017-08-07T21:44:54Z.
URL: *https://github.com/Hamlib/Hamlib*

Hi-Tec Aerials UHF Folded Dipoles & Yagis - Professional Series (n.d.).
URL: *http://bit.ly/2M2rCDP*

Hitefield, S., Leffke, Z., Fowler, M. & McGwier, R. W. (2016), System overview of the Virginia Tech Ground Station, *in* '2016 IEEE Aerospace Conference', pp. 1–13.
URL: *https://ieeexplore.ieee.org/abstract/document/7500716*

Kelso, T. (n.d.), 'CelesTrak Active Satellite TLE Files'.
URL: *http://celestrak.com/NORAD/elements/active.txt*

Loff, S. (2015), 'CubeSats Overview'.
URL: *https://go.nasa.gov/2nvJgXl*

Open Source Mobile Communications (2019).
URL: *https://osmocom.org/*

Prussing, J. & Conway, B. (2013), *Orbital Mechanics*, 2 edn, Oxford University Press, New York.

Rupprecht, M. (n.d.), 'Status - DK3wn SatBlog'.
URL: *http://www.dk3wn.info/p/?page_id=29535*

SDR# and Airspy Downloads - airspy.com (n.d.).
URL: *https://airspy.com/download/*

Snjegota, A. & Rattenbury, N. J. (2017), 'Detecting Forward-Scattered Radio Signals from Atmospheric Meteors Using Low-Cost Software Defined Radio', *The Physics Teacher* **55**(2), 115–117.
URL: *http://aapt.scitation.org/doi/10.1119/1.4974127*

Teske, L. (2019), 'racerxdl/gr-osmosdr'. original-date: 2016-10-26T02:30:28Z.
URL: *https://github.com/racerxdl/gr-osmosdr*

Thirumalai, A. (2017), CubeSat Ground Station Development - An Automated Tracking System for Amateur Satellites, PhD thesis, The University of Auckland, Auckland.

Tolmasoff, M. & Venturini, C. (2017), 'Improving Mission Success of CubeSats'.
URL: *https://go.nasa.gov/2mIqgnD*

Ubuntu Manpage: rtl_tcp - an I/Q spectrum server for RTL2832 based DVB-T receivers (n.d.).
URL: *http://manpages.ubuntu.com/manpages/trusty/man1/rtl_tcp.1.html*

UZ7HO Personal page - Packet-Radio - English version (n.d.).
URL: *http://uz7.ho.ua/packetradio.htm*

Welcome to Yaesu.com (n.d.).
URL: *http://bit.ly/35o6j7g*

Yaesu GS-232 Rotor Computer Interface, GS232A, gs232b (n.d.).
URL: *https://www.universal-radio.com/catalog/hamrot/4228.html*

Zufelt, B. & Aarested, J. (2018), CloudSat: Centralized-Adaptable Ground Station Architecture utilizing an IoT Design Methodology, *in* '2018 World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC)', pp. 1–6.
URL: *https://ieeexplore.ieee.org/abstract/document/8629747*