Cislunar Explorers Semester Report

Tyler King

December 10, 2021

Contents

1 Introduction

I joined Cislunar in early September 2021 as part of the Communications team. Initially most of the work I completed was with I&T, helping train my technical hardware skills (soldering, wire cutting/stripping, cleanroom training, etc) that would be relevant throughout this semester. Over time, my focus shifted towards communications, and particularly the ground station. I helped set up documentation for uplinking/downlinking along with documenting all ground station components and utilizing software to interface with our hardware. Additionally, under the guidance of Michael Hojnowski (radio club president and systems engineer for Cornell), I was able to set up tests with the FlatSat to test downlinking results.

2 I&T EDU Integration

Initially, my heavy focus on hardware meant I was tasked with checking CAD diagrams and following them to confirm clarity of instructions. A big part of this was setting up the ebox, confirming that the specifications in past CAD diagrams worked when fitting the relevant hardware.

When trying to place the rest of the ebox on the battery for the satellite, we ran into spacing problems due to the limited washer height (3mm). These changes were then fixed by using taller washers and the CAD design was updated.

I also helped set up pin housing connectors in the ebox, numbering the pins and following CAD procedures.

3 Documentation

3.1 Flowchart Documentation

I worked closely with Paul to sift through past documents on Box along with online parts documentation to establish formal flowcharts for uplinking and downlinking processes. These flowcharts (figures 1, 2) were based on similar ones from NASA and incorporated all known components and connections. This was one of the several pieces Toby insisted on to help clean up documentation (particularly in communications) for future students working on the project.

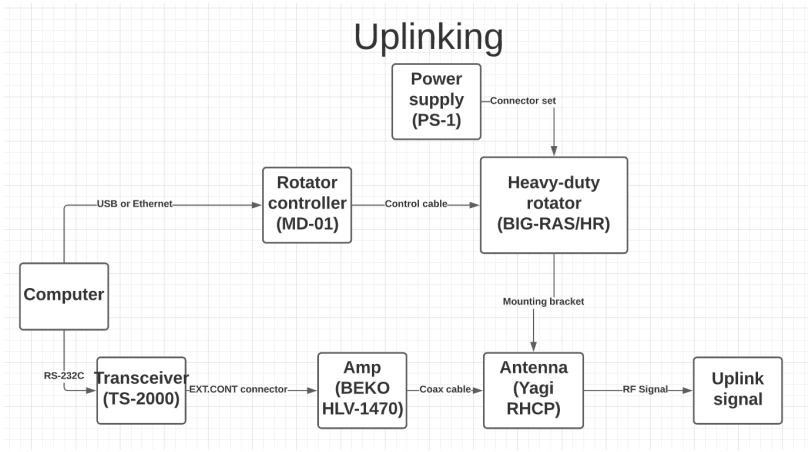

Figure 1: Uplink Flowchart

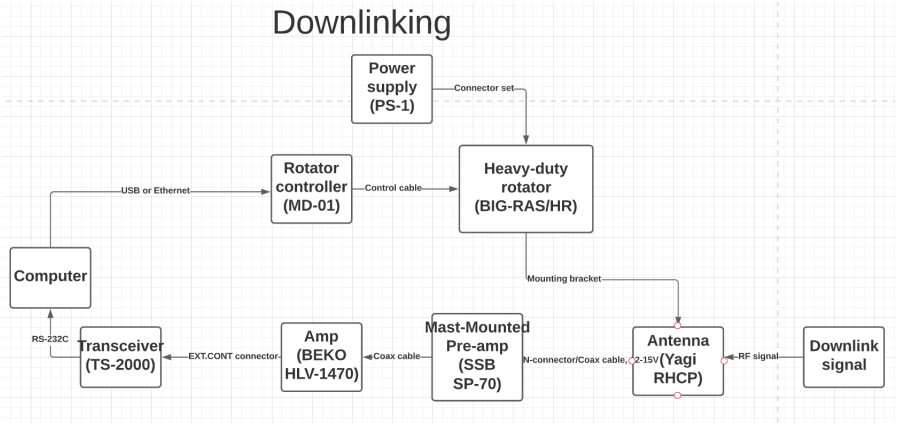

Figure 2: Downlink Flowchart

3.2 Parts Documentation

After finalizing the flowcharts, I began looking for the individual components of our ground station to make sure that there were no inconsistencies. We were able to find all the components we expected to see and documented this information accordingly (Parts Documentation.gdoc contains images of all components along with other crucial information). The location of all parts is as follows:

- Power Supply (PS-01) - Rhodes Penthouse
- Rotator Controller (MD-01) - Rhodes Penthouse
- Heavy-duty Rotator (BIG-RAS/HR) - Rhodes Roof
- Transceiver (TS-2000) - Rhodes Penthouse
- Amp (BEKO HLV-1470) - Upson 450
  - Due to lab space changes in the incoming semester, we may have to relocate the Amp.
- Mast-Mounted Pre-amp (SSB SP-70) - Rhodes Roof

- Antenna (Yagi RHCP) - Rhodes Roof

Thanks to Mike, me and Paul were able to access the Rhodes rooftop, allowing us to check out all the individual components of the ground station.

4 Transceivers

Even though most of our past work in communications had been based on the use of the TS-2000 as our transceiver of choice, Mike was hesitant against us making too much headway with this because it has historically been inconsistent at receiving signals during the downlinking process.

4.1 TS-2000

As mentioned earlier, this was the transceiver we had planned to use throughout most of Cislunar's history, mainly because of its high power output that would pass the minimum wattage threshold for the BEKO HLV-1470. However, as mentioned earlier, the inaccuracy with receiving was a major concern and this led us to running all of our downlinking tests with the HackRF (which was borrowed from PAN). This was worsened by the TS-2000's filters to corrode and degrade over time, limiting their relative sensitivity. Although the receive sensitivity isn't horrible (Mike offered to tune it, which he says can get us to around -126dbm), it is below what our link budget needs at the current time.

Furthermore, we had great difficulty with moving the TS-2000 to run tests. Since the distance from the basement to the Penthouse was too far to obtain any noticeable results and the TS-2000 was bolted down in the Rhodes Penthouse, it was difficult to run tests or even obtain access to the transceiver.

Figure 3: Kenwood TS-2000

4.2 HackRF

In part because of ease of maneuverability, the HackRF became our go-to transceiver when testing FlatSat downlinking signals. For ease of testing, we chose to forgo a lot of our heavy-duty ground station components, using only the HackRF and the built-in antenna for testing. This allowed us to use various software on personal laptops, allowing us to try out various programs that we discuss further in section 6.

The largest issue with the HackRF was the low driving power (between 1 and 30 mW) that would not be enough to supply out HLV-1470 amp. We decided that the best course of action would thus be to use an RF switch (one which we could easily borrow from Mike or some other source).

4.3 RF Switch

The greatest advantage of an RF switch means that we could effectively swap between the TS-2000 and the HackRF, taking advantage of their individual strengths. Due to the TS-2000's driving power, we could use this for uplinking and the HackRF for downlinking with the help of an RF switch. This would allow us to take advantage of the strengths of both while mitigating their weaknesses.

Note that the TS-2000 is untested, so it may still be possible. In the future, I hope to move around the TS-2000 and conduct more thorough tests. Especially with the link budget documentation being updated, it is worth seeing whether or not the TS-2000 is viable for downlinking.

5 Coax Cables

A majority of our connections between hardware components on both the ground station and the satellite utilize coax cables. This meant that there were several nuances that I observed and analyzed that I felt were worth discussing in my report.

5.1 Coax Loop

HAM radio enthusiasts tend to encourage looping exposed coax cables that are connected to antenna arrays. This was something that has not been implemented for our Yagi RHCP on the rooftop of Rhodes and may be worth considering in the future. Adding a loop helps prevent water flowing down the sides of the coax cable and into the connector (this technique is also known as a drip loop). It can also help serve as a coaxial RF choke, something that I discuss in greater detail in 5.2 since a similar technique was implemented for the coax cables on the satellite.

5.2 Collins Balun

Having loops in coax cables can also increase the effectiveness on an antenna's performance without having any direct negative results. While analyzing the cables on the satellite, we noticed that one coax cable was around 3 times longer than the other, and after talking with Mike, we concluded that it was

due to the Collins balun. The effectiveness of this technique depends on various characteristics of the loop, including the diameter and number of turns.

Additionally, the longer coax cable can help correct the shielding effect on the dipole antenna.

6 Software

Our various attempts to downlink effectively from our ax5043 radio chip to the HackRF were contingent on effective software reading the transmission. Thanks to the help of Mike and various members of the Radio Club, I was able to grasp the fundamentals fairly quickly.

6.1 SDRSharp

Initially, we chose to use SDRSharp since it effectively interfaced with the HackRF (along with the TS-2000 if we so desired further down the road). While trying to learn the software, I experimented with various radio settings, allowing me to listen in to FM radio and other broad spectrum. After getting these results, we then moved on to running tests with the FlatSat, trying to transmit PSK (phase-shift keying) modulation.

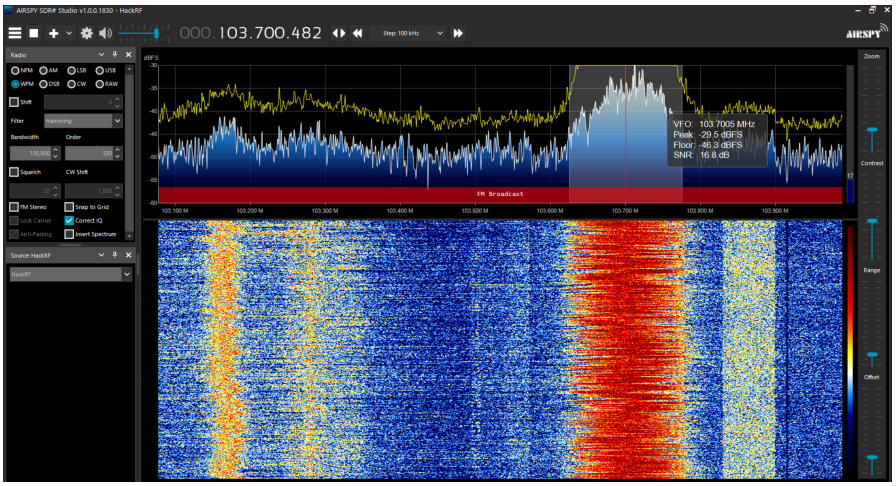

Figure 4: Listening to 103.7 FM with SDRSharp and a HackRF

Some advantages of SDRSharp include a simple UI that allows for easy changes on bandwidth and target frequency while also allowing for the utilization of virtual audio cables.

6.2 Virtual Audio Cables

The one largest limitation with SDRSharp is they do not provide the tools to directly decrypt PSK signals. This meant that we would need to port the output from SDRSharp over to a different software. To make this transition, we

installed VB-Audio as our VAC (virtual audio cable), allowing us to port the detected audio from SDRSharp to Fldigi.

6.3 Fldigi

With Fldigi, it was possible to decrypt various PSK encryptions (PSK31, PSK125, PSK500, etc). When we initially ran our tests, we used PSK31 decryption on Fldigi because this was the default setting for the ax5043 radio chip. However, upon closer visual inspection we noticed that the baud rate of our transmissions was 500 Hz instead of the expected 31 Hz. This variation in baud rate could lead to dramatic changes in our link budget, something that we would have to consider if we do plan to go through with the varying PSK transmissions in the future.

We also noticed that, when Fldigi was decoding the signal, the output would show up in varicode instead of the expected ascii result. This is an area that I am still looking into and hope to continue to improve upon during my next semester. My hope is that by changing around the register keys, we will be able to have the output in ascii instead of varicode.

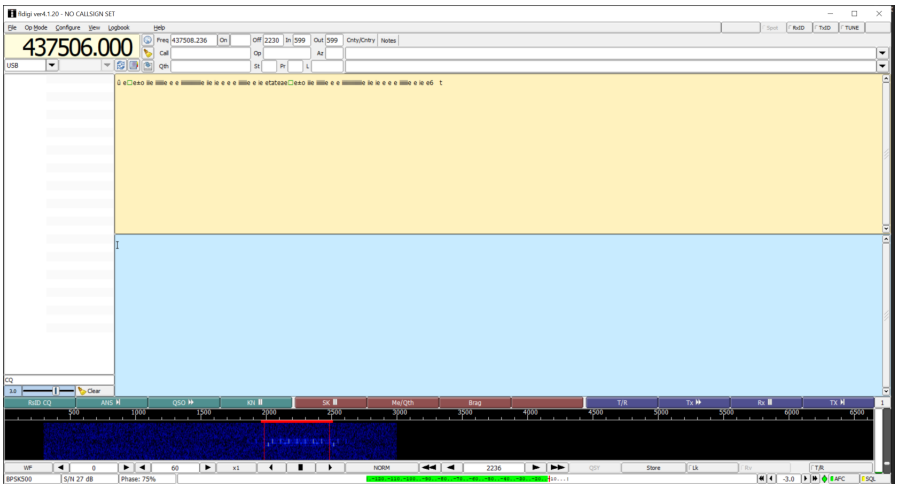

Figure 5: Example output in Fldigi with PSK500

It is worth noting that Abhinav (in his 2019 Spring semester report) had success with receiving PSK125 back when we still used FPrime. Due to the differences between Python and FPrime along with the lack of thorough documentation in the past, it is difficult to replicate the results that have been observed before.

Additionally, Abhinav used SDRConsole (in conjunction with the HackRF) to both receive and process his signals. Although different than my combination of SDRSharp+Fldigi, the result remains the same. I downloaded SDRConsole to help confirm this.

6.4 GNU Radio

To try and test whether the output from Fldigi was flawed or not I decided to download GNU Radio and compare the results. Due to errors with the window installation, I was forced to utilize Ubuntu in a VM. Even with GNU Radio installed several packages that were necessary to run tests with the HackRF had been deprecated and as a result, I was forced to abandon this plan. However, several packages still remain actively updated (including the ones to produce radio output), so it could be an effective way in the future to test uplinking as necessary.

7 FlatSat

Due to my close work with downlinking, I frequently was involved with the FlatSat (and the related ax5043 chip). To make it easier for future members of Cislunar, I helped write down documentation for certain FlatSat processes that were relevant to what I was doing, along with general usage (particularly in relationship to WSL - Windows Subsystem for Linux).

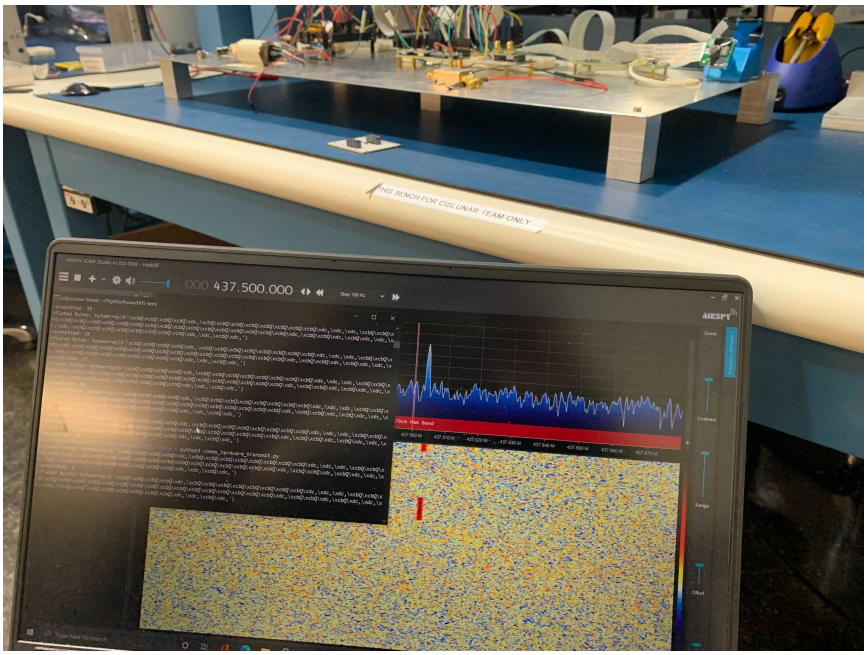

Figure 6: Running tests with the FlatSat in the background

8 Software

As my interests in working with register keys grew, I became more involved on the software side of Cislunar. In particular, I worked closely with both Toby and Stephen to build up my understanding of integral software skills including

git and other command line features. This allowed me to begin making contributions in the flight software repository, where I created a branch that hosted my proposed edits to the register keys